Zenbakizko metodoak MATLAB® erabiliz

2. edizioa

Eugenio Juan Mijangos Fernández

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

UPV/EHUko Euskara Zerbitzuak sustatua eta zuzendua, Euskarazko ikasmaterialgintza sustatzeko deialdiaren bitartez.

eISBN: 978-84-9860-545-7

Zenbakizko metodoak

MATLAB®

erabiliz

2. edizioa

Eugenio Juan Mijangos Fernández

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Eugenio Juan Mijangos Fernández

Matematika Saila

Zientzia eta Teknologia Fakultatea

eugenio.mijangos@ehu.eus

Gaien Aurkibidea

Irudien Zerrenda

Taulen Zerrenda

Hitzaurrea

Zenbakizko metodoak I *irakasgaian Matematika Graduko 2. mailako euskarazko ikasleek duten hutsune bibliografikoa liburu honen bidez betetzea da testugile honen helburu nagusia. Egia esan, dakidanez, gai honetaz ez dago liburu bat ere euskaraz. Liburu hau oso erabilgarria izan daiteke bai Ingeniaritzako ikasketetan, eta baita Zientziako beste ikasketa batzuetan ere; hau da, oso erabilgarria izan daiteke.*

Liburu honen bidez zenbakizko metodoen oinarriak eskainiko dira, bai ikuspuntu teoriko batetik, bai ikuspuntu aplikatu batetik. Iraganean, zenbakizko analisian eta teorian oinarritzen zen zenbakizko metodoei buruzko ikasturte bat, baina, gaur egun, irakasgai horren edukia garatzean, kontuan hartu behar dugu ordenagailu ahaltsuak eta kalkulu-software eraginkorrak eskura izatea. Oraingo joera, gero eta gehiago, aplikazioetara eta inplementazioetara joatea da, jadanik prest dagoen kalkulu-tresneria erabiliz. Ikasturte batean, ikasleek zenbakizko metodoen oinarriak ikasiko dituzte. Gainera, ordenagailu-lengoaia batean programatzen ikasiko dute, eta software aurreratua erabiliko problemak ebazteko tresna gisa. MATLAB da horrelako softwarearen adibide egoki bat. Ikasleek beren programak idazteko erabil dezakete pakete informatiko hori, eta haren funtzioak problemak ebazteko tresna gisa.

Dezimalekin erabili beharreko puntuazioari dagokionez, testuan koma eta programazio-adibideetan puntua ez erabiltzearren, puntuz bereiziko dira dezimalak, nahiz eta euskaraz dezimalak komaz banandu.

Bigarren argitalpeneko oharrak:

Argitalpen honetan akats batzuk zuzendu ditut, eta hobekuntza arin batzuk sartu.

Nire gurasoei eskaintzen diet.

1. kapitulua

Sarrera

1.1. Problema bat duzu

Demagun zubi-jauzi enpresa batek zure zerbitzuak kontratatu nahi dituela; hau da zure lana: jauzilariaren erortzeko abiadura, denboraren funtzioan (erorketa librean), aurretik jakitea. Informazio hori erabiliko da zehazteko (beste analisi handiago batean) masa desberdineko jauzilarientzako kordaren luzera eta erresistentzia.

Zuk badakizu, fisika-ikasketei esker, Newtonen bigarren legearen arabera $a = F/m$ erlazioa bete behar dela. Baina, fluidoen mekanikari buruzko ezagupenen arabera, eredu matematiko hau garatzen duzu:

$$\frac{dv}{dt} = g - \frac{c_d}{m}v^2, \tag{1.1}$$

non v = abiadura bertikala baita, t = denbora (s), g = grabitate-azelerazioa (≈ 9.81 m/s^2), c_d = airearen erresistentzia-koefizientea (kg/m) eta m = jauzilariaren masa (kg).

Ekuazio diferentzial horren $v = v(t)$ soluzio analitiko edo zehatz bat kalkula dezakegu eta, $t = 0$ denean $v = 0$ dela hartzen badugu, hau da soluzio analitikoa:

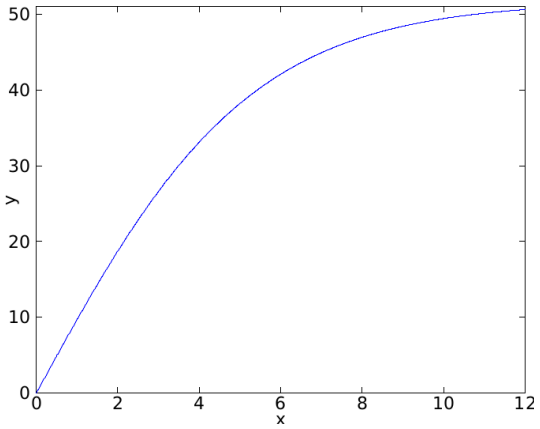
$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right), \tag{1.2}$$

non tangente hiperbolikoa (tanh) zuzen kalkula baitezakegu, edo funtzio esponentzialak erabiliz, adierazpen honetan:

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

1.1. adibidea. *68.1 kg-ko zubi-jauzilari batek zubi batetik salto egiten badu, erabili (1.2)*

t (s)	v (m/s)
0	0
2	18.7292
4	33.1118
6	42.0762
8	46.9575
10	49.4214
12	50.6175
∞	51.6938



1.1. irudia. Abiadura/denbora grafikoa.

adierazpena, bere abiadura erorketa librean 12 s-ko unean kalkulatzeko. Zein izango da azken abiadura, kordaren luzera infinitua bada? Erabili $c_d = 0.25$ *kg/m.*

Ebazpena. Parametroen balioak (1.2) adierazpenean ordezkatzen baditugu hau ematen du:

$$v(t) = \sqrt{\frac{9.81 \cdot 68.1}{0.25}} \tanh\left(\sqrt{\frac{9.81 \cdot 0.25}{68.1}} t\right) = 51.6938 \tanh(0.18977t),$$

eta hori erabil dezakegu taula hau kalkulatzeko:

Eredu honen arabera, jauzilariak azkar azeleratzen du; 49.4214 m/s-ko abiadura hartzen du 10. segundoan. Kontuan izan, baita ere, denbora nahiko handia denean, 51.6983 m/s-ko azken abiadura hartzera jotzen duela. Abiadurak konstante izatera jotzen du, zeren grabitatearen indarra orekatu egingo baita airearen erresistentziarekin. Beraz, indar garbia zero izango da eta azelerazioa gelditu egingo da. □

(1.2) ekuazioari *soluzio analitikoa* (edo *zehatza*) deritzogu, zehazki betetzen duelako jatorrizko ekuazio diferentziala. Zoritxarrez, badaude zehazki ebatzi ezin diren eredu matematikoak. Kasu askotan, dagoen aukera bakarra *zenbakizko soluzio* bat garatzea da, eta horrek soluzio zehatza hurbiltzen du.

Zenbakizko metodoetan, problema matematiko bat birformulatzen da eragiketa aritmetikoen bidez ebazteko moduan. Hori argitu daiteke, (1.1) ekuaziorako honako hau kontuan hartuz:

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}, \tag{1.3}$$

Kontuan izan $dv/dt \approx \Delta v/\Delta t$ hurbilpena dugula Δt finitua delako. Gogoratu kalkulu infinitesimalean hau betetzen dela:

$$\frac{dv}{dt} = \lim_{\Delta t \to 0} \frac{\Delta v}{\Delta t}.$$

(1.3) ekuazioari t uneko deribatuaren *diferentzia finituen hurbilpena* deritzo. Hura ordezka dezakegu, (1.1) ekuazioan hau emateko:

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c_d}{m} v(t_i)^2.$$

Ekuazio hori honela ordena dezakegu:

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c_d}{m} v(t_i)^2 \right] (t_{i+1} - t_i). \tag{1.4}$$

Kontuan izan makoen arteko gaia (1.1) ekuazio diferentzialaren eskuineko gaia dela. Hau da, ekuazio horrek ematen digu bide bat v-ren malda kalkulatzeko. Ondorioz, ekuazioa honela berridatz dezakegu:

$$v(t_{i+1}) = v(t_i) + \frac{dv}{dt}(t_i) \cdot \Delta t. \tag{1.5}$$

Orain, ekuazio diferentzialaren bidez beste ekuazio bat lortu dugu, zeina t_{i+1} uneko abiadura kalkulatzeko erabil baitezakegu aurreko t eta v-ren balioak erabiliz. Eta lortutako balioak, t_{i+1} eta v_{i+1}, erabil ditzakegu v_{i+2} kalkulatzeko, elkarren segidan. Hau da:

Balio berria = balio zaharra + malda × urratsaren luzera.

Metodo horri *Eulerren metodoa* deritzogu.

1.2. adibidea. *Aurreko adibidean, orain (1.5) ekuazioa erabiliko dugu abiadura kalkulatzeko. Urratsaren luzera 2 s izango da.*

Ebazpena. Hasieran $t_0 = 0$ hartuko dugu; orduan, jauzilariaren abiadura 0 da. Aurreko adibideko parametroen balioak erabiliz, $t_1 = 2$ s unean abiadura kalkulatuko dugu:

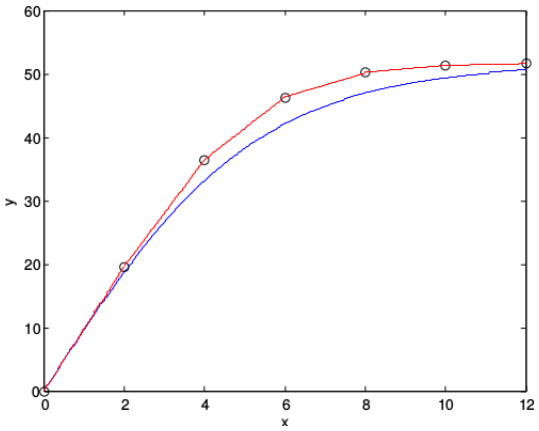
$$v = 0 + \left[9.81 - \frac{0.25}{68.1} 0^2 \right] \times 2 = 19.62 \text{ m/s}.$$

Hurrengo tarterako ($t = 2$-tik $t = 4$-ra), kalkuluak errepikatu eta hau lortuko dugu:

$$v = 19.62 + \left[9.81 - \frac{0.25}{68.1} 19.62^2 \right] \times 2 = 36.4137 \text{ m/s}.$$

Kalkuluaz horrela jarraituz, taula hau lortuko dugu:

t (s)	v (m/s)
0	0
2	19.6200
4	36.4137
6	46.2983
8	50.1802
10	51.3123
12	50.6008
∞	51.6938



1.2. irudia. Zenbakizko soluzioa eta soluzio analitikoa.

2. kapitulua

MATLABi buruzko oinarrizko nozioak

Kapitulu honetan ikusiko dugu MATLAB paketea programa matematikoen multzo bat dela eta matrizeen erabileran oinarritzen dela. MATLAB izenak MATrix LABoratory esan nahi du, zeren bere oinarrizko datu-gaia matrize bat baita. MATLAB kalkulu teknikorako lengoaia ahaltsua da, eta oso erabilia da unibertsitateetan, Matematika, Zientzia eta Ingeniaritza ikasketetan. Pakete horrek zenbakizko programen eta marrazketa bidimentsionala eta tridimentsionala egiteko programa grafikoen bilduma zabal bat dauka. Gainera, goi-mailako lengoaia erabiliz, programa gehigarriak idaztea onartzen du.

MATLAB paketearen instrukzioak `idazmakina-letraz` idatziko dira. Adibideetan ikusiko dugu MATLABeko lan-leiho batek (ingelesez: *command window*) erakusten diguna. Honelako >> ikur baten jarraian agertzen da sartutako datua edo instrukzioa. Nahi duguna idatzi eta gero, «sartu» tekla sakatu behar dugu; orduan, ordenagailuak eragiketa egingo du eta emaitza erakutsiko du, honela: `ans =`. Programarekin erabiltzeko gidak eta erreferentziako gidak datoz; eta laguntza-leihoan ere aurki daitezke instrukzioei buruzko eta haien aukerei buruzko informazio gehiago eta adibideak.

2.1. Eragiketa aritmetikoak

+	Batuketa
−	Kenketa
∗	Biderketa
/	Eskuineko zatiketa
\	Ezkerreko zatiketa
^	Berreketa
pi, e, i	Konstanteak

2.1. adibidea.

```
>>(2+3*pi)/2
ans =
      5.7124
```

2.2. Programari gehitutako funtzioak

Jarraian, MATLAB paketeko funtzio erabilgarri batzuen zerrenda labur bat eskaintzen da:

`abs(#)`	`cos(#)`	`exp(#)`	`log(#)`	`log10(#)`	`cosh(#)`	`sum(#)`	`length(#)`
`sin(#)`	`tan(#)`	`sqrt(#)`	`acos(#)`	`tanh(#)`	`floor(#)`	`ceil(#)`	`round(#)`

Programak berak funtzio erabilgarriei buruzko informazioa eskaintzen du. Funtzio horien zerrenda izateko, idatzi «`help elfun`». Honako adibide honek argitzen du nola erabiltzen eta nola konbinatzen diren eragiketa aritmetikoak eta funtzioak.

2.2. adibidea.

```
>>3*cos(sqrt(4.7))
ans=
     -1.6868
```

Normalean, emaitzan bost zifra hamartar esanguratsu erakusten dira; `format long` instrukzioari esker, hamabost zifra hamartar esanguratsu arte lor ditzakegu.

2.3. adibidea.

```
>> format long
>> 3*cos(sqrt(4.7))
ans=
     -1.68686892236893
```

Jarraian, formatu-instrukzioekin agertzen da taula bat. Haiek `format mota` sintaxiarekin idazten dira.

Mota	Emaitza	Adibidea
`short`	Puntu finkoa 5 digiturekin	3.1416
`long`	Puntu finkoa 15 digiturekin	3.14159265358979
`short e`	Puntu higikorraren formatuan 5 digiturekin	3.1416e+000
`long e`	Puntu higikorraren formatuan 15 digiturekin	3.14159265358979e+000

2.3. Esleipen-instrukzioak

Berdintza ikurraren bidez, adierazpen baten kalkuluaren emaitzari izen bat eslei diezaiokegu.

2.4. adibidea.

```
>>a=3-floor(exp(2.9))
a=
     -15
```

Adierazpen baten bukaeran puntu eta koma idazten dugunean, konputagailuak dagozkion eragiketak egiten ditu, eta bere emaitza gordetzen du guk esleitutako izenpean. Hurrengo adibidean, ez da erakusten `b`-ren balioa.

2.5. adibidea.

```
>>b=sin(a);
>>2*b^2
ans=
     0.8457
```

2.4. Matrizeak

MATLAB paketean, aldagai guztiak matrizeak dira. Matrizeak zuzen sartzen dira.

2.6. adibidea.

```
>>A=[1 2 3;4 5 6;7 8 9]
A=
   1 2 3
   4 5 6
   7 8 9
```

«;» ikurrak matrizeko lerroak bereizten ditu, eta lerro bateko gaiak hutsune-espazio baten bidez (edo koma batez) bereizi behar ditugu. Matrizeak lerroz lerro ere sar ditzakegu.

2.7. adibidea.

```
>>A=[1 2 3
     4 5 6
     7 8 9]
A =
   1 2 3
   4 5 6
   7 8 9
```

Gehitutako funtzio batzuk erabiliz, matrize berezi batzuk sor ditzakegu.

2.8. adibidea.

```
>>Z=zeros(3,5);              (zerozko 3 x 5 dimentsioko matrize bat sortzen du)
>>X=ones(3,5);               (batezko 3 x 5 dimentsioko matrize bat sortzen du)
>>Y=0:0.5:2                  (1 x 5 dimentsioko matrize hau sortzen du:)
Y=
0 0.5000 1.0000 1.5000 2.0000

>>sin(Y)                     (Y matrizeko gai bakoitzaren sinua hartuz, 1 x 5
                             dimentsioko matrize hau sortzen du:)
ans=
     1.0000 0.8776 0.5403 0.0707 -0.4161
```

`linspace(a,b,n)` instrukzioak a eta b puntuen artean tarte berak bereizitako n puntu sortzen ditu, muturrak sartuta. Aurreko adibideko `Y` bektore berdina ematen du, hau idazten badugu:

```
» linspace(0,2,5)
```

Ez badugu jartzen `n`, instrukzioak 100 puntu sortuko ditu automatikoki.

`logspace(a,b,n)` instrukzioak `linspace(10^a,10^b,n)` instrukzioaren berdina ematen du.

Ez badugu jartzen `n`, instrukzioak 50 puntu sortuko ditu automatikoki.

Matrize bateko gaiekin hainbat eratara joka dezakegu.

2.9. adibidea.

```
>>A(2,3)                    (A-ren (2,3) lekuko gaia aukeratzen du)
ans=
     6
>>A(1:2,2:3)                (A-ren azpimatrize bat aukeratzen du)
ans=
     2 3
     5 6
>>A([1 3],[1 3])            (beste era bat A-ren azpimatrize bat aukeratzeko)
ans=
     1 3
     7 9
>>A(2,2)=tan(7.8);          (beste balio bat esleitzen dio A-ren (2,2) lekuan
                            dagoen gaiari)
>>A(2,:)
ans=
     4 5 6
>>A(:,3)
ans=
    3
    6
    9
```

Laguntza-leihoak beste funtzio matrizialei buruzko informazioa eskaintzen digu.

2.5. Eragiketa matrizialak

+	Batuketa
–	Kenketa
/	Eskuineko zatiketa
\	Ezkerreko zatiketa
*	Biderketa
^	Berreketa
'	Irauli konjugatua

2.10. adibidea.

```
>>B=[1 2;3 4];
>>C=B'                          (C B-ren iraulia da)
C=
    1 3
    2 4
>>3*(B*C)^3
ans=
     13080 29568
     29568 66840
```

Azken hori $3(BC)^3$ da.

2.11. adibidea.

```
>>a=[1 2 3];
>>b=[2;4;6];
>>a*b
ans=
     28
>>b*a
ans=
    2   4   6
    4   8  12
    6  12  18
>>a*A
ans=
```

```
    30 36 42
>>c=[4 5 6]';
>>A*c
ans=
    32
    77
   122
>>A*a
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>A/pi
ans=
   0.3183  0.6366  0.9549
   1.2732  1.5915  1.9099
   2.2282  2.5465  2.8648
>>2\[1 2;3 4]
ans=
    0.5  1
    1.5  2
```

`A\B` idazten badugu, $A^{-1}B$ emango digu, eta `B/A` idatziz gero, BA^{-1}. Ondorioz, `4\1`-ek eta `1/4`-ek emaitza bera dute: 0.25.

Zer emango du MATLABek `2/[1 2;3 4]` idazten badugu?

2.6. Gaiez gai egiten diren eragiketak

MATLAB paketearen ezaugarri erabilgarrienetariko bat da matrize batez gaiez gai eragiten duen funtzio kopuru handi bat daukala. Aurreko adibide batean ikusi dugu 1×5 matrize baten gai bakoitzeko sinua kalkulatzen duela. Gaiez gai egiten dira batuketa, kenketa eta eskalar bateko biderketa eragiketa matrizialak; aldiz, hori ez da gertatzen biderketa, zatiketa eta berreketa eragiketa matrizialekin. Azken hiru eragiketak gaiez gai egin ditzakegu eragiketaren aurrean puntu bat idazten badugu, hau da: $.*$, $./$, $.\backslash$ eta $.\hat{}$. Oso garrantzitsua da jakitea nola eta noiz erabili behar ditugun eragiketa horiek, zeren gaiez gaiko eragiketak oso garrantzitsuak baitira zenbakizko programak eta grafiko programak MATLAB paketearekin eraginkorki diseinatzeko eta inplementatzeko orduan.

2.12. adibidea.

```
>>A=[1 2;3 4];
```

```
>>A^2              (AA biderketa kalkulatzen du.)
ans=
     7 10
     15 22
>>A.^2                          (A-ren gai bakoitza karratura jasotzen du.)
ans=
     1 4
     9 16
>>2.\A
ans=
    0.5000 1.0000
    1.5000 2.0000
>>cos(A./2)                     (A-ren gai bakoitza 2rekin zatitu eta gero,
                                kosinua kalkulatzen du.)
ans=
     0.8776  0.5403
     0.0707 -0.4161
```

2.6.1. MATLABek emandako funtzioak

Ikus 2.2. atalean MATLABeko oinarrizko funtzioak. Funtzio horietako propietate garrantzitsu bat da haietako gehienek bektoreen eta matrizeen gainean eragiten dutela, aurreko puntua jarri gabe.

2.13. adibidea.

```
>>log(A)
ans=
    0.0000   0.6931
    1.0986   1.3863
>>B=sqrt(A)
B=
    1.0000   1.4142
    1.7321   2.0000
>>round(B)   (B-ren gaiak zenbaki oso hurbilenera biribiltzen ditu.)
ans=
    1   1
    2   2
>>ceil(B)  (B-ren gaiak goiko zenbaki oso hurbilenera biribiltzen ditu.)
```

```
ans =
    1   2
    2   2
>>floor(B) (B-ren gaiak beheko zenbaki oso hurbilenera biribiltzen ditu.)
ans=
    1   1
    1   2
>>F=[3 5 4 6 1];
>>sum(F)
ans =
    19
>>min(F),max(F),mean(F),prod(F),sort(F)
ans =
    1
ans =
    6
ans =
    3.8000
ans =
    360
ans =
    1   3   4   5   6
```

2.7. Grafikoak

MATLAB paketeak kurben eta gainazalen marrazketa bidimentsionalak eta tridimentsionalak egin ditzake. Laguntza-paketean kontsulta daitezke instrukzio-grafikoen aukera eta alderdi gehigarriak.

`plot` instrukzioaz, kurba lauen grafikoak sor daitezke. Adibide honetan, $[0, \pi]$ tarteko $y = \cos(x)$ eta $y = \cos^2(x)$ funtzioen grafikoak lor ditzakegu.

2.14. adibidea.

```
>>x=0:0.1:pi;
>>y=cos(x);
>>z=cos(x).^2;
>>plot(x,y,x,z,'o')
```

Lehenengo lerroan zehazten dira eremua eta 0.1 urratseko tamaina. Hurrengo bi lerroetan funtzioak definitzen dira. Kontuan izan lehenengo hiru lerroak puntu eta koma batez bukatzen direla; puntu eta koma hori erabiltzen da ez agertzeko pantailan `x`, `y` eta `z` matrize bakoitzeko 32 gaiak. Laugarren lerroak grafikoak ematen duen marrazketa-instrukzioa dauka. Lehenengo bi gaiek, `x` eta `y`, $y = \cos(x)$ funtzioa marrazten dute. Hirugarren eta laugarren gaiek, `x` eta `z`, $z = \cos^2(x)$ funtzioa marrazten dute. Azken gaiak, `'o'`, behartzen du marraztera 'o' puntu hauetan: (x_k, z_k), non $z_k = \cos^2(x_k)$.

Hirugarren lerroan, oinarrizkoa da «`.^`» gaiez gaiko eragiketa-adierazlea erabiltzea; izan ere, lehenik gai bakoitzaren kosinua kalkulatzen da, eta, gero, `cos(x)` matrizearen gai bakoitza karratura jasotzen da `.^` instrukzioa erabiliz.

`fplot` marrazketa-instrukzioa `plot` instrukziorako aukera erabilgarria da. Instrukzio horren sintaxia `fplot('izena',[a,b],n)` da. Honek `izena.m` funtzioaren grafikoa ematen digu, haren balioa zehaztuz $[a, b]$ tarteko n puntutan. Ez bada ematen n-ren balioa, $n = 25$ da.

2.15. adibidea. *Honek* $[-2, 2]$ *tartean* $y = \tanh$ *marrazten du:*

```
>>fplot('tanh',[-2,2])
```

`plot` eta `plot3` instrukzioak erabiltzen dira kurba parametrizatu bidimentsionalak eta tridimentsionalak marrazteko.

2.16. adibidea. $c(t) = (2\cos(t), 3\sin(t))$ *elipsearen marrazketa,* $0 \leq t \leq 2\pi$*, instrukzio hauekin lortzen da:*

```
>>t=0:0.2:2*pi;
>>plot(2*cos(t),3*sin(t))
```

2.17. adibidea. $c(t) = (2\cos(t), t^2, 1/t)$ *kurbaren marrazketa,* $0.1 \leq t \leq 4\pi$*, instrukzio hauekin lortzen da:*

```
>>t=0.1:0.1:4*pi;
>>plot3(2*cos(t),t.^2,1./t)
```

`meshgrid` instrukzioaz marrazketa tridimentsionalak lortzeko funtzioaren eremu angeluzuzen bat zehaztu behar dugu, eta, gero, grafikoa lortu `mesh` edo `surf` instrukzioekin.

2.18. adibidea.

```
>>x=-pi:0.1:pi;
>>y=x;
>>[x,y]=meshgrid(x,y);
>>z=sin(cos(x+y));
>>mesh(z)
```

`hold on` instrukzioak datu eta propietate guztiekin gordetzen ditu marrazkiak, eta, orain, beste instrukzio batzuk gehi daitezke.

`hold off` instrukzioak leiho grafiko hori bukatzen du.

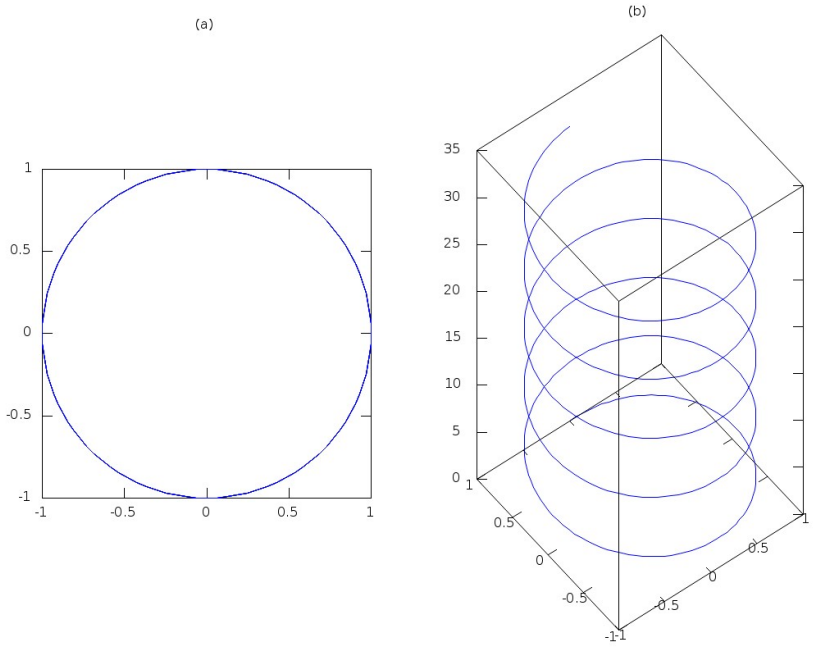
`subplot(m,n,p)` instrukzioak leiho grafikoa zatitzen du m $\times$ n zatitan, eta p. azpileihoa (ezkerretik eskuinera eta goitik behera zenbatuta) aukeratzen du, oraingo grafikoa marrazteko.

2.19. adibidea.

```
>>t=0:pi/50:10*pi;
>>subplot(1,2,1);
>>plot(sin(t),cos(t));
>>axis square
>>title('(a)')
>>subplot(1,2,2);
>>plot3(sin(t),cos(t),t);
>>title('(b)')
```

Irudia gorde dezakegu pdf eta jpg formatuetan zera idatziz:

```
>>print -dpdf irudiizena.pdf;
>>print -djpg irudiizena.jpg;
```



2.1. irudia. Zirkulu bidimentsional bat eta helize tridimentsional bat, bi zatitako irudi batean.

Agindua	Ezaugarria
plot	$x - y$ grafiko soila
loglog	Grafikoa logaritmikoki eskalatutako ardatzekin
semilogx	Grafikoa logaritmikoki eskalatutako x ardatzarekin
semilogy	Grafikoa logaritmikoki eskalatutako y ardatzarekin
polar	Grafikoa koordenatu polarretan
fplot	Funtzio marratzailea
plot3	$x - y - z$ grafiko soila
contour	Sestra-kurben grafikoa
mesh	Sare-begizko gainazala
meshc	Sare-begizko gainazala sestra-kurbekin
surf	Gainazal solidoa
surfc	Gainazal solidoa sestra-kurbekin

2.1. taula. Funtzio marratzaileen laburpen bat.

Beste aukera batzuk ere badaude; begiratu laguntza-leihoan.

2.8. MATLABez programatzen: M fitxategiak

2.8.1. Instrukzioen fitxategiak

MATLABen instrukzioen segida bat fitxategi batean gordetzen dugunean, horrelako instrukzioen M fitxategi bat dugu. Fitxategi horiek erabilgarriak dira, behin baino gehiagotan erabiltzen badugu instrukzioen segida hori. Era hauetan egikari daiteke:

- Aginduen leihoan fitxategiaren izena idatziz.
- Fitxategia editatuz eta editorearen leihotik exekutatuz.

2.20. adibidea. *Garatu instrukzioen fitxategi bat, puenting-jauzilariaren abiadura kalkulatzeko.*

Ebazpena:

```
g=9.81; m=68.1; t=12; cd=0.25;
v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)
```

Hau `instrukadib.m` izeneko fitxategi batean gordetzen badugu eta, gero, aginduen leihoan hau idazten badugu:

```
>>instrukadib
```

emaitza hau izango da:

```
v=
   50.6175
```

Parametro baten balioa jakin nahi badugu, adibidez `g`-rena, hau idatziko dugu:

```
>>g
g=
 9.8100
```

2.8.2. Funtzio-fitxategiak

Guk funtzio berriak defini ditzakegu MATLAB paketearekin erabiltzeko. Horretarako, `m` luzapeneko testu-fitxategi bat idatzi behar dugu MATLABek berak duen editorea erabiliz: M fitxategi bat. Behin definituz gero, beste edozein funtzio bezala erabil daiteke.

2.21. adibidea. *$f(x) = -x^2/4 + x + 1$ funtzioa definituko dugu, eta* `fun.m` *izeneko M fitxategi batean gordeko dugu. Horretarako, testu-editoreaz hau idatziko dugu:*

Ebazpena:

```
function y=fun(x)
% Funtzio honen bidez f(x)=a*x^2+b*x+c polinomioa kalkula dezakegu.
a=-1/4; b=1; c=1;
y=a*x^2+b*x+c
```

Geroago, funtzio hori baliozta dezakegu aginduen leihoan `x` argumentuaren balio baterako. Hots, $f(2)$ kalkulatzeko, nahikoa da honako hau idaztea:

```
>>fun(2)
ans=
    2.0000
```

Gainera, funtzio honi buruz laguntza eska dezakegu:

```
>>help fun
Funtzio honen bidez f(x)=a*x^2+b*x+c polinomioa kalkula dezakegu.
```

Aldagaietarako letra desberdinak erabil ditzakegu eta funtzioari izen desberdina eman diezaiokegu, baina formatu berdina izan behar dute. Funtzio hori `fun.m` fitxategi batean gorde eta gero, MATLABeko beste edozein funtzio bezala erabil dezakegu.

```
>>cos(fun(3))
ans=
     -0.1782
```

Bestalde, `feval` instrukzioaz baliozta ditzakegu funtzioak era erabilgarri eta eraginkor batean.

```
>>feval('fun',4)
ans=
     1
```

Geroago, funtzioaren izena ahazten badugu, guk honela erabil dezakegu `lookfor` instrukzioa:

```
>>lookfor polinomioa
```

eta honako informazio hau jasoko dugu:

```
 fun.m: % Funtzio honen bidez f(x)=a*x^2+b*x+c polinomioa kalkula dezakegu.
```

Aldiz, hau gertatuko da:

```
>>a
??? Undefined function or variable 'a'.
```

Horren arrazoia zera da: `a` aldagaiak -1/4 balioa hartzen du M fitxategiaren barnean, hots, aldagai *lokala* da, eta aldagai lokalak ezabatu egiten dira fitxategia exekutatu eta gero. Aldiz, instrukzio M fitxategi baterako aldagaiak lan-espazioan gordetzen dira, banan-banan instrukzio bakoitza aginduen leihoan eskuz sartuko bazenu bezala; hori gertatzen da, adibidez, `instrukadib.m` M fitxategiarekin. Ariketa gisa, egiazta ezazu hori.

M fitxategiek emaitza bat baino gehiago itzul ditzakete. Adibidez, `estatiskoak.m` fitxategi honek bektore bati dagozkion gaien batezbestekoa eta desbideratze estandarra kalkulatzen ditu:

```
function [bbest,dest]=estatistikoak(x)
n=length(x);
bbest=sum(x)/n;
dest=sqrt(sum((x-bbest).^2/n));
```

Orduan, honela joka dezakegu:

```
>>y=[8 5 10 12 6 7.5 4];
>>[b,d]=estatistikoak(y)
b =
  7.5000
d =
  2.6049
```

Instrukzio M fitxategiak gutxi erabiltzen dira; funtzio M fitxategiak erabiliko ditugu batez ere eta, ondorioz, M fitxategiak izen laburtuarekin adieraziko ditugu hemendik aurrera horiek.

2.8.3. Azpifuntzioak

Funtzio batek beste funtzio batzuk barnera ditzake. Horrelako funtzioak M fitxategi berezitu moduan existitu arren, M fitxategi bakar batean sartuta egon daitezke.

Har dezagun, berriro, jauzilariaren 2.20. adibidea; kalkula dezagun abiadura ere M fitxategi honen bitartez:

```
function v=jauzilari(t,m,cd)
v=abi(t,m,cd);
end

function v=abi(t,m,cd)
g=9.81;
v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)
end
```

`end` funtzio bakoitzaren muga adierazteko erabiliko dugu, baina ez da beharrezkoa. Fitxategi hori `jauzilari.m` izenarekin gordeko dugu. Lehenengo funtzioari *funtzio nagusia* (*main function*) deritzogu. Funtzio hori aginduen leihoan bakarrik eskura dezakegu. Beste funtzioak, kasu honetan `abi`, *azpifuntzioak* (*subfunctions*) dira. Azpifuntzio bat funtzio nagusirako M fitxategiaren barnean bakarrik da eskuragarria. Aginduen leihotik exekutatuz, hau dugu:

```
>>jauzilari(12,68.1,0.25)
ans=
   50.6175
```

Hala ere, `abi` azpifuntzioa exekutatzen saiatzen bagara, honelako errore-mezu bat aterako da:

```
>>abi(12,68.1,0.25)
??? Undefined command/function 'abi'.
```

2.8.4. Input-output

Funtzioan, aginduen leihoaren bidez soilik ez dira sartzen edo ateratzen parametroen balioak edo beste informazio bat; badaude beste bi bide ere lan hori egiteko.

- `input` funtzioa. Funtzio honen bidez, erabiltzaileak balio batzuk zuzenean eman ditzake; adibidez, honela idatziz M fitxategian:

  ```
  m=input('Masa (kg): ')
  ```

  Lerro hau exekutatzen denean, monitorean hau agertuko da:

  ```
  Masa (kg):
  ```

  Erabiltzaileak balio bat sartuko du, eta orduan hura esleituko zaio `m` aldagaiari.

  Funtzio honen bidez hitzak (*strings*) ere sar ditzakegu; horretarako, `'s'` gehituko diogu funtzioaren argumentuen zerrendari. Adibidez,

  ```
  izena=input('Sartu zure izena: ','s')
  ```

- `disp` funtzioa. Funtzio honek era praktiko bat ematen du parametro baten balioa edo hitza (*string*) ikusi ahal izateko. Bere sintaxia `disp(parametroa)` da.

2.22. adibidea.

```
function jauzilaria
% jauzilaria: jauzilariaren abiadura kalkulatzeko era interaktibo bat
g=9.81;
m=input('Masa (kg): ');
cd=input('Erresistentzia-koefizientea (kg/m): ');
t=input('Denbora (s): ');
disp('  ')
disp('Abiadura (m/s):')
disp(sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t))
```

`jauzilaria.m` izeneko fitxategian gordez gero, eta hau idatziz:

```
>> jauzilariia

Masa (kg): 68.1
Erresistentzia-koefizientea (kg/m): 0.25
Denbora (s): 12

Abiadura (m/s):
    50.6175
```

- `fprintf` funtzioa. Funtzio honen bidez kontrolatzen dugu monitorean ateratzen den informazioa. Haren sintaxia, labur emanda, hau da:

  `fprintf('formatua',x,...)` non `formatua`-ren bidez adierazten baitugu nola ikusi nahi dugun `x`-ren balioa. Adibidez,

```
 >>fprintf('Abiadura %8.4f m/s da\n',abiadura)
Abiadura   50.6175 m/s da
```

  Normalean, hauek dira `fprintf` funtzioarekin erabiltzen diren formatuen eta kontrolen kodeak:

Formatuen kodeak	**Deskripzioa**
%d	Zenbaki osoaren formatuan
%e	Formatu zientifikoa e minuskularekin
%E	Formatu zientifikoa e maiuskularekin
%f	Formatu hamarrenarekin
%g	%e edo %f formatuen trinkoena
Kontrolen kodea	**Deskripzioa**
\n	Hasi lerro berri bat
\t	Tabuladorea

  Ikus ditzagun, orain, bi adibide funtzio honen erabilera hobe ulertzeko.

2.23. adibidea.

```
>> fprintf('%5d %10.3f %8.5e\n',100,2*pi,pi);
  100      6.283  3.14159e+000
```

2.24. adibidea.

```
function fprintfdemo
x=[1 2 3 4 5];
y=[20.4 12.6 17.8 88.7 120.4];
z=[x;y];
fprintf('     x        y\n');
fprintf('%5d %10.3f\n',z);
```

Hau da horren emaitza:

```
>> fprintfdemo

    x       y
    1    20.400
    2    12.600
    3    17.800
    4    88.700
    5   120.400
```

2.8.5. Fitxategiak sortu eta fitxategietan sartu

MATLABek datu-fitxategiak irakur eta idatz ditzake. Modu errazenak fitxategi bitar berezi bat erabiltzen du, MAT fitxategi deritzona; hori espresuki gauzatzen da MATLABen barnean inplementatzeko. Horrelako fitxategiak `save` instrukzioaz sortzen dira; `load` instrukzioaz fitxategian sar daitezke, eta datuak irakurri. `fitxategiizena.mat` datu-fitxategia sortzeko edo irakurtzeko, sintaxi hau erabiliko dugu:

```
save fitxategiizena var1 var2 ...  varn

load fitxategiizena var1 var2 ...  varn
```

Ez baditugu `var1 var2 ...  varn` aldagaiak idazten, lan-espazioko aldagai guztiak gordeko dira (`save` kasuan) edo kargatuko dira (`load` kasuan).

Adibidez,

```
>> g=9.81;m=80;t=5;
>> cd=[.25 .267 .1245 .28 .273]';
>> v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
```

Orduan, abiadura- eta erresistentzia-koefizienteak gorde ditzakegu `abikoef.mat` fitxategian, hau idatziz:

```
>> save abikoef v cd
```

`clear` instrukzioa idazten badugu, lan-espazioko aldagai guztiak ezabatuko dira. Aldagai horiek berreskuratzeko, nahikoa da hau idaztea:

```
>> load abikoef v cd
```

Lan-espazioan zein aldagai ditugun jakiteko, hau idatziko dugu:

```
>> who

Your variables are:
cd   v
```

Horrelako fitxategiak oso erabilgarriak dira MATLABekin lan egiteko; baina, agian, beste programa batzuekin hobe izango da testu-fitxategi bat izatea, hau da, ASCII fitxategi bat. Horretarako, nahikoa da `save abikoef v cd -ascii` idaztea, eta `abikoef.txt` fitxategi batean gordeko du. Datuak zehaztasun bikoitzarekin gorde nahi baditugu, `-ascii -double` idatziko dugu. Adibidez,

```
>> A=[5 7 9 2;3 6 3 9];
>> save simpmatrix.txt -ascii -double
```

Horrelako fitxategi bat beste programa batzuek irakur dezakete; esate baterako, Excelek eta Word-ek. Bestalde, `simpmatrix.txt` ez denez MAT fitxategi bat, behin MATLABek irakurriz gero, MATLABek zehaztasun bikoitzeko matrize bat sortuko du, honela:

```
>> load simpmatrix.txt
>> simpmatrix
simpmatrix =
     5    7    9    2
     3    6    3    9
```

Gainera, `load` instrukzioa funtzio gisa erabil dezakegu. Adibidez,

```
>> A = load(simpmatrix.txt)
```

2.8.6. Programazio egituratua

Begiztak eta adarkadurak

Erlazio eragileak:

Eragilea	Erlazioa	Adibidea
==	Zerbaiten berdin	`x == 0`
∼=	Zerbaiten desberdin	`unitate ∼= 'm'`
<	Zerbait baino txikiago	`a < 0`
>	Zerbait baino handiago	`s > t`
<=	Zerbait baino txikiago edo berdin	`2.8 <= a/3`
>=	Zerbait baino handiago edo berdin	`r >= 0`

Eragile logikoak:

$\sim$	Ez	(Egiazkoa, baldin eta soilik baldin, proposizioa faltsua bada)
&	Eta	(Egiazkoa, baldin bi proposizioak egiazkoak badira)
|	Edo	(Egiazkoa, baldin bi proposizioetako bat egiazkoa bada)

Booleko balioak:

1	Egiazkoa
0	Faltsua

`for`, `if`, `switch` eta `while` egiturek MATLAB paketean eragiten dute beste programazio-lengoaia batzuetan egiten duten antzeko era batean. Instrukzio horiek oinarrizko sintaxi hau hartzen dute:

`for` (begiztaren aldagaia=begiztaren heina)

(adierazpenak)

`end`

Baina MATLABen, batzuetan, ez dugu `for` begizta erabiltzeko beharrik. Adibidez,

```
i=0;
for t=0:0.02:50
   i=i+1;
   y(i)=cos(t);
end
```

adieraz daiteke *era bektorizatu* batean, honela:

```
t=0:0.02:50
y=cos(t);
```

MATLABek automatikoki handitzen du bektoreen eta matrizeen luzera. Adibidez,

```
t=0:0.01:5;
for i=1:length(t)
  if t(i)>1
    y(i)=1/t(i);
```

```
  else
    y(i)=1;
  end
end
```

```
if (baldintza)
        (adierazpenak)
end
```

```
if (baldintza)
        (adierazpenak)
    else
        (adierazpenak)
end
```

Baldintza bat baino gehiago daudenean, `if...elseif...else...end` egitura erabiltzen dugu:

```
if (baldintza)
        (adierazpenak)
    elseif (baldintza)
        (adierazpenak)
    elseif (baldintza)
        (adierazpenak)
            .
            .
            .
    else
        (adierazpenak)
end
```

"`error`" funtzioa erabiltzeko, `if` oso egokia da. Adibidez, hau dugu:

```
function f=errortest(x)
if x==0
   error('zero balioa aurkitu du');
end
f=1/x
```

Hau da, zatiketa kalkulatuko da, argumentua zero ez denean bakarrik.

```
>> errortest(10)
ans =
    0.1000
```

Bestela, mezu hau emango digu:

```
 >> errortest(0)
??? Error using ==> errortest
zero balioa aurkitu du
```

`switch` egitura `if...elseif...else...end` egituraren antzekoa da. Hala ere, banakako baldintzak jarri beharrean, adarkatzea test-adierazpen txikiagoetan oinarritzen da:

`switch` (test-adierazpena)

`case` (balioa);

(adierazpenak)

`case` (balioa);

(adierazpenak)

.

.

.

`otherwise`

(adierazpenak)

```
   end
```

Adibidez,

```
kalifikazioa = 'B';
switch kalifikazioa
  case 'A'
    disp('Bikain')
  case 'B'
    disp('Oso ondo')
  case 'C'
    disp('Ondo')
  case 'D'
    disp('Nahiko')
  case 'E'
    disp('Txarto')
  otherwise
    disp('Oso txarto')
end
```

Kode hori exekutatzen badugu, «`Oso ondo`» emango digu.

`while` egituran begizta bat errepikatzen da, baldintza logikoa egia bada, hots:

`while` (baldintza)

(adierazpenak)

`end`

Adibidez,

```
x=8
while x>0
  x=x-3;
  disp(x)
end
```

Kode hori exekutatzen denean, emaitza hau da:

```
x=
    8
```

```
    5
    2
   -1
```

Hurrengo adibidean erakusten da nola sar ditzakegun begizta batzuk elkarren gainka, matrize bat sortzeko. Testu-lerroak `habia.m` fitxategi batean gordez, orduan, MATLAB paketeko lan-leihoan `habia` idazten dugun bakoitzean, `A` matrizea lortuko dugu. Ohartaraziko dugu `A` matrizeko gaiek, goiko ezkerreko izkinatik hasiz, Pascalen triangelua sortzen dutela.

2.25. adibidea.

```
for i=1:5
  A(i,1)=1;
  A(1,i)=1;
end
for i=2:5
  for j=2:5
    A(i,j)=A(i,j-1)+A(i-1,j);
  end
end
A
```

`break` instrukzioa azken (`for` edo `while`) begiztatik ateratzeko, hau bete baino lehen erabiliko dugu.

2.26. adibidea.

```
for k=1:100
  x=sqrt(k);
  if ((k>10)&(x-floor(x)==0))
    break
  end
end
k
```

`disp` instrukzioa erabiliko dugu testu-lerro bat edo matrize bat erakusteko.

2.27. adibidea.

```
n=10;
k=0;
while k<=n
  x=k/3;
  disp([x x^2 x^3]);
  k=k+1;
end
```

`pause` instrukzioa erabiltzen da, programa baten exekuzioa gelditu nahi badugu leku egoki batera heltzen denean. `pause(n)` idazten badugu, **n** segundotan zehar geldi egongo da. `tic` instrukzioak oraingo denbora gordetzen du, eta `toc` instrukzioak monitorean erabilitako denbora emango digu. `beep` instrukzioak, berriz, ordenagailuaren «beep» soinu bat emango digu. Adibidez,

```
tic
beep
pause(5)
beep
toc
```

Kode hau exekutatzen denean, «beep» bat emango digu. Bost segundo geroago beste «beep» bat entzungo dugu, eta monitorean mezu hau izango dugu:

```
Elapsed time is 5.006306 seconds.
```

Programa baten exekuzioa moztu nahi badugu, nahikoa da **Ctrl+C** jotzea.

2.28. adibidea. Egitura habiaratuak. *Adierazpen honen bidez kalkula ditzakegu* $f(x) = ax^2 + bx + c$ *ekuazio koadratikoaren erroak:*

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Garatu funtzio bat, formula hori inplementatzeko koefizienteen balioak finkatu eta gero.

Ebazpena:

```
function errokoad(a,b,c)
% errokoad: ekuazio koadratikoaren erroak
%     errokoad(a,b,c): ekuazio koadratikoaren erro erreal eta konplexuak
% input:
%   a=2. mailako koefizientea
%   b=1. mailako koefizientea
%   c=0. mailako koefizientea
% output:
%   r1=1. erroaren zati erreala
%   i1=1. erroaren zati irudikaria
%   r2=2. erroaren zati erreala
%   i2=2. erroaren zati irudikaria
if a==0
        % kasu bereziak
        if b~=0
                % erro erreal bakuna
                r1=-c/b
        else
                % soluzio nabaria
                error('Soluzio nabaria. Saiatu berriro')
        end
else
        % formula koadratikoa
        d=b^2-4*a*c;                    % diskriminantea
        if d>=0
                % erro errealak
                r1=(-b+sqrt(d))/(2*a)
                r2=(-b-sqrt(d))/(2*a)
        else
                % erro konplexuak
                r1=-b/(2*a)
                i1=sqrt(abs(d))/(2*a)
                r2=r1
                i1=-i1
        end
end
```

Orain, kalkulu hauek egin ditzakegu:

```
>> errokoad(1,1,1)
r1=
  -0.5000
i1=
```

```
   0.8660
r2=
  -0.5000
i2=
  -0.8660
>> errokoad(1,5,1)
r1=
  -0.2087
r2=
  -4.7913
>> errokoad(0,5,1)
r1=
  -0.2000
>> errokoad(0,0,0)
??? Error using ==> errokoad
Soluzio nabaria. Saiatu berriro
```

2.8.7. M fitxategiei funtzioak igorri

M funtzio bat garatu daiteke ekuazio berri bakoitzerako, baina aukera hobea da funtzio generiko bat diseinatzea eta analizatu nahi dugun ekuazioa argumentu gisa pasatzea. MATLABen hizkeran, horrelako funtzioek `funtzio-funtzioak` izena dute.

Funtzio anonimoak

Funtzio anonimoak erabiliz funtzio soil bat sor dezakegu, M fitxategi bat sortu gabe. Aginduen leiho baten barnean defini daitezke haiek, honelako sintaxi baten bidez:

```
feskuleku = @(argumzerrenda) adierazpena
```

non `feskuleku =` zuk erabil dezakezun funtzio-eskulekua baita, `argumzerrenda` funtzioaren argumentuen zerrenda, komaz bereizia, eta `adierazpena` MATLABeko adierazpen bat. Adibidez,

```
>> f1=@(x,y) x^2 + y^2;
>> f1(3,4)
ans =
    25
```

Beste adibide bat:

```
>> a=4;
>> b=2;
>> f2=@(x) a*x^b;
>> f2(3)
ans =
    36
```

`a` eta `b`-rako balio berriak sartzen baditugu, ez da aldatzen funtzio anonimoaren balioa:

```
>> a=3;
>> f2(3)
ans =
    36
```

Baina, guk birsortzen badugu funtzio bera, balioa aldatuko da:

```
>> f2=@(x) a*x^b;
>> f2(3)
ans =
    27
```

Antzeko emaitzak lortzeko beste bide bat da `inline` funtzioa erabiltzea. Hau da:

```
>> f1=inline('x^2 + y^2','x','y');
```

Funtzio-funtzioak

Funtzio-funtzioak beste funtzio batzuetan eragiten duten funtzioak dira, eta funtzioak argumentu gisa pasatzen dira. Adibide argi bat `fplot` funtzioarena da; honek funtzioen grafikoak marrazten ditu. Hau da bere idazkera labur bat:

`fplot`(*fun,lims*)

non *fun* funtzio matematikoa marraztuko baita *lims*=[*xmin,xmax*] mugen artean. Adibidez:

```
>> vel=@(t) ...
sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t)
>> fplot(vel,[0 12])
```

Honek $t = 0$tik $t = 12$rako marrazki bat sortuko du.

2.29. adibidea. *Sortu funtzio-funtzioaren M fitxategi bat, tarte bateko funtzio baten balioen batezbestekoa kalkulatzeko. Hau ikusiko dugu $t >= 0$tik $t = 12$rako tarteko puenting-jauzilariaren abiadura erabiliz.*

Ebazpena. Hau da hori lortzeko bide bat:

```
>> t=linspace(0,12);
>> v=sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> mean(v)
ans =
   36.0870
```

Beste bide bat:

```
function fbb = funcbb(a,b,n)
%
% funcbb: batezbestekoa kalkulatzen du
% input:
%    a = tartearen ezker muturra
%    b = tartearen eskuin muturra
%    n = tartearen puntuen kopurua
% output:
%    fbb = funtzioaren balioen batezbestekoa
x = linspace(a,b,n);
y = func(x);
fbb = mean(y);
end

function f = func(t)
f=sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
end
```

eta gero,

```
>> funcbb (0,12,60)
ans =
   36.0127
```

Funtzio argumentua

Emaitza berdina lortzeko beste era bat da abiadura kalkulatzeko *funtzioa argumentu moduan* sartzea; hau da:

```
function fbb = funcbb(f,a,b,n)
%
% funcbb: batezbestekoa kalkulatzen du
% input:
%    f = balioztatu nahi dugun funtzioa
%    a = tartearen ezker muturra
%    b = tartearen eskuin muturra
%    n = tartearen zatien kopurua
% output:
%    fbb = funtzioaren balioen batezbestekoa
x = linspace(a,b,n);
y = f(x);
fbb = mean(y);
end
```

Gero, hau idatziko dugu:

```
>> abi=@(t) ...
sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> funcbb(abi,0,12,60)
ans =
   36.0127
```

Funtzioen parametroak aldatzea

Ikertzeko orduan oso arrunta da ikustea nola aldatzen den menpeko aldagaiaren balioa funtzioen parametroak aldatzen ditugunean. Parametroei buruzko ikerketa horri ereduaren *sentsibilitatearen ikerketa* deritzogu.

Aurreko 2.29. adibideko `funcbb` funtzioaren sentsibilitate-analisi bat egin nahi badugu, M-file hau idatz dezakegu:

```
function fbb=funcbb(f,a,b,n,varargin)
x = linspace(a,b,n);
y = f(x,varargin(:));
fbb = mean(y);
```

Geroago, funtzio anonimo hau idatziko dugu:

```
>> abi=@(t,m,cd) sqrt(9.81*m/cd)*tanh(sqrt(9.81*cd/m)*t);
```

eta gero, `m` eta `cd` parametroen balio desberdinetarako abiadura kalkula dezakegu. Adibidez, $m = 68.1$ eta $c_d = 0.25$ badira, hau ematen du:

```
>> funcbb(abi,0,12,60,68.1,0.25)
ans =
   36.0127
```

Gero, $m = 100$ eta $c_d = 0.28$ badira, hau ematen du:

```
>> funcbb(abi,0,12,60,100,0.28)
ans =
   38.9345
```

2.9. Problemak

1. Defini ditzagun a, b, c eta d aldagaiak, honela: $a = 14.75$, $b = -5.92$, $c = 61.4$ eta $d = 0.6(ab - c)$. MATLABen bidez kalkula itzazu:

$$(a) \qquad a + \frac{ab(a+d)^2}{c\sqrt{|ab|}} \qquad\qquad (b) \qquad de^{d/2} + \frac{(ad+cd)/\left(\frac{25}{a} + \frac{35}{b}\right)}{a+b+c+d}.$$

2. Sortu tarte berak bereizitako lerro-bektore bat 16 gai dauzkana, non lehenengo gaia 4 baita eta azkena 61.

3. Sortu zutabe-bektore bat non lehenengo gaia 31 baita, hurrengo gaiak txikiagoak egiten baitira -4 gehituz, eta azkena -9 baita. (Zutabe-bektore bat sor daiteke lerro-bektore bat irauliz).

4. Sortu jarraian ematen den matrizea. Lerroak sartzean, erabili bektoreen notazioa tarte berak bereizitako bektoreak sortzeko. (Hots, ez sartu gaiak banan-banan.)

$$A = \begin{bmatrix} 0 & 1.0000 & 2.0000 & 3.0000 & 4.0000 & 5.0000 & 6.0000 \\ 3.0000 & 9.1667 & 15.3333 & 21.5000 & 27.6667 & 33.8333 & 40.0000 \\ 28.0000 & 27.7500 & 27.5000 & 27.2500 & 27.0000 & 26.7500 & 26.5000 \\ 6.0000 & 5.0000 & 4.0000 & 3.0000 & 2.0000 & 1.0000 & 0 \end{bmatrix}.$$

   Lehenengo bi ariketetan bi puntuen ikurra (:) erabiliz, egin hau:

   (a) Sortu 4 gaiko lerro-bektore bat, `va` izenekoa, A-ren bigarren lerroko azken lau gaiak dauzkana.

   (b) Sortu 4 gaiko zutabe-bektore bat, `vb` izenekoa, A-ren seigarren zutabeko gaiak dauzkana.

   (c) Sortu 3×4 matrizea, `B` izenekoa, A matrizearen 1., 2. eta 4. lerroetako eta 1., 2. 4. eta 7. zutabeetako gaiak erabiliz.

   (d) Sortu 2×3 matrizea, `C` izenekoa, A matrizearen 2. eta 4. lerroetako eta 2., 5. eta 6. zutabeetako gaiak erabiliz.

5. Demagun $y = \dfrac{(x^3+1)^2}{x^2+2}$ funtzioa. Kalkula ezazu y-ren balioa x-ren balio hauetarako: -1.6, -1.2, -0.8, -0.4, 0, 0.4, 0.8, 1.2. Ebatzi problema hau x bektore bat sortuz eta gero y bektore bat sortuz, gaiez gaiko kalkuluak erabiliz. Bikote horietarako egin grafiko bat, non puntuak izartxo batez adierazita agertzen baitira, eta puntuak lotuta lerro beltzez. Etiketatu ardatzak.

6. Defini dezagun $a = 0.8$ eskalarra eta $x = -3, -2.8, -2.6, \ldots, 2.6, 2.8, 3$. Orduan, erabili aldagai horiek honela, y kalkulatzeko: $y = \dfrac{8a^2}{x^2+4a^2}$. Marraztu y, x-rekiko.

7. Erabili MATLAB frogatzeko $\sum_{n=0}^{\infty}(-1)^n\frac{1}{2n+1}$ serie infinituaren batura, $\pi/4$ra jotzen duena. Egin batura, hori kalkulatuz, n-ren balio hauetarako:

   (a) $n = 100$

   (b) $n = 1000$

   (c) $n = 5000$.

   Sortu `n` izeneko bektore bat alde bakoitzean, non lehenengo gaia 0 baita, gehikuntza 1, eta azken gaia 100, 1000 edo 5000 baita. Orduan, erabili gaiez gaiko kalkulua bektore bat sortzeko, non gaiak $(-1)^n\frac{1}{2n+1}$ baitira. Azkenik, erabili `sum` funtzioa seriearen gaiak batzeko. Konparatu (a), (b) eta (c) ataletan lortutako balioak $\pi/4$ balioarekin.

8. St. Louis-eko Sarrerako Arkua osatzen da ekuazio honen arabera:

   $$y = 693.8 - 68.8\cosh\left(\frac{x}{99.7}\right)$$

   (oinetan neurtuta). Egin Arkuaren marrazki bat, non $-299.25 \leq x \leq 299.95$ oin.

Daniel Schwen CC BY-SA 3.0

9. Bektore bat honela emango dugu: `x=[15 85 72 59 100 80 44 60 91 38]`. Baldintzazko egiturak eta begiztak erabiliz, idatzi programa bat 59 baino handiagoak diren `x`-ko gaien batezbestekoa kalkulatzeko.

10. Kosinu-funtzioaren balioa kalkula dezakegu serie infinitu honen bidez:

    $$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots.$$

    Sortu M fitxategi bat, formula hori inplementatuz, hau egiteko: $\cos x$-ren balioak kalkulatu, eta monitorean seriearen gai bakoitza erakutsi gehitzen dituen neurrian. Alegia, kalkulatu balio hauek elkarren segidan eta erakutsi monitorean:

    $$\begin{aligned}\cos x &= 1\\ \cos x &= 1 - \frac{x^2}{2!}\\ \cos x &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!}\\ &\vdots\end{aligned}$$

eta segi horrela zuk aukeratutako ordenaraino. Aurreko balio hurbildu bakoitzerako, kalkulatu ehuneko errore erlatiboa, eta erakutsi monitorean hau kontuan hartuz:

$$errorea = \frac{egiazkoa - seriearen\ hurbilpena}{egiazkoa} \times 100.$$

Proba bezala, erabili programa cos(1.5) kalkulatzeko, batuketan 8 batugai sartuta; hau da, seriea garatuz $x^{14}/14!$ batugairaino.

11. Planoko puntu bat kokatzeko, bi koordenatu behar ditugu:

    - Koordenatu kartesiarretan, (x, y) ardatzekiko distantzia horizontala eta bertikala.
    - Koordenatu polarretan, (r, θ) erradioa eta angelua.

    Koordenatu polarrak ezagutuz gero, nahiko erraza da koordenatu kartesiarrak kalkulatzea. Aldiz, alderantziz ez da hain erraza. Erradioa adierazpen honen bidez kalkulatzen dugu:

    $$r = \sqrt{x^2 + y^2}.$$

    Puntua lehenengo edo laugarren koadrantean dagoenean (hots, $x > 0$), erraza da θ kalkulatzeko formula:

    $$\theta = \arctan\left(\frac{y}{x}\right).$$

    Beste kasuetarako agertzen da zailtasuna. Honako taula honek laburtzen ditu aukerak:

x	y	θ
<0	>0	$\arctan(y/x) + \pi$
<0	<0	$\arctan(y/x) - \pi$
<0	$=0$	π
$=0$	>0	$\pi/2$
$=0$	<0	$-\pi/2$
$=0$	$=0$	0

    Idatzi M fitxategi ondo egituratu bat r eta θ kalkulatzeko, x-ren eta y-ren funtzioan. Adierazi θ-rako azken emaitzak, gradutan. Proba ezazu zure programa, kasu hauek balioztatuz:

x	y	r	θ
1	0		
1	1		
0	1		
-1	1		
-1	0		
-1	-1		
0	0		
0	-1		
1	-1		

12. Garatu M fitxategi bat, non 0tik 100erako zenbakizko balio bat igortzen badiogu, hark taula honi jarraituz letrazko kategoria bat bueltatuko baitigu:

Letra	**Irizpidea**
A	$90 \leq$ zenbakizko balioa ≤ 100
B	$80 \leq$ zenbakizko balioa < 90
C	$70 \leq$ zenbakizko balioa < 80
D	$60 \leq$ zenbakizko balioa < 70
E	zenbakizko balioa < 60

13. Manning-en ekuazio hau erabil dezakegu uraren abiadura kalkulatzeko ubide ireki laukizuzen batean:

$$U = \frac{\sqrt{S}}{n}\left(\frac{BH}{B+2H}\right)^{2/3},$$

non $U =$ abiadura (m/s), $S =$ ubidearen malda, $n =$ marruskadura koefizientea, $B =$ zabalera (m) eta $H =$ sakonera (m). Datu hauek eskuragarriak dira bost ubidetarako:

n	S	B	H
0.035	0.0001	10	2
0.020	0.0002	8	1
0.015	0.0010	20	1.5
0.030	0.0007	24	3
0.022	0.0003	15	2.5

Idatzi M fitxategi bat ubide horietarako abiadura kalkulatzen duena. Sartu balio horiek matrize batean, non zutabe bakoitzak parametro bat adierazten baitu eta lerro bakoitzak ubide bat. Monitorean taula itxura izan behar du M fitxategiaren irteerak; baina handituta, bosgarren zutabe batekin, non ubide bakoitzari dagokion abiadura agertzen baita, zutabeak etiketatzeko taularen goiburuak sartuta.

14. Habe baten desplazamendua honelako funtzio batek neurtzen du:

$$y(x) = \frac{-5}{6}[< x-0 >^4 - < x-5 >^4] + \frac{15}{6} < x-8 >^3 + 75 < x-7 >^2 + \frac{57}{6}x^3 - 238.25x,$$

non x habearen zeharkako distantzia baita, eta *singulartasun-funtzioa* honela definitzen baita:

$$< x-a >^n = \left\{ \begin{array}{ll} (x-a)^n, & x > a \text{ denean} \\ 0, & x \leq a \text{ denean} \end{array} \right\}$$

Garatu M fitxategi bat, x abszisa eta y ordenatua dituen grafiko bat egiteko.

15. Likido baten B bolumena, r erradiodun eta L luzeradun zilindro horizontal baten barnean, likidoaren h sakoneraren menpean dago formula honen bidez:

$$B = \left[r^2 \arccos\left(\frac{r-h}{r}\right) - (r-h)\sqrt{2rh-h^2}\right] L.$$

Garatu M fitxategi bat, bolumena/sakonera grafiko bat marrazteko. Proba ezazu programa, $r = 2$ m eta $L = 5$ m denean.

16. «Zatitu eta erdibanatu» metodoa, a zenbaki positibo baten erro karratua hurbiltzeko metodo zaharra, honela formula daiteke:

$$x = \frac{x + a/x}{2}.$$

Idatzi `while ... break` begiztako egituran oinarritutako M fitxategi ondo egituratu bat, algoritmo hori inplementatzeko. Erabili koska egoki bat, egitura argi bat izateko moduan. Urrats bakoitzean, balioetsi zure hurbilpenaren errorea adierazpen honen bidez:

$$\varepsilon = \left| \frac{x_{berria} - x_{zaharra}}{x_{berria}} \right|.$$

Errepikatu begizta ε-ren balioa, zuk emandako ε_e tolerantzia bat baino txikiago izan arte. Diseinatu zure programa, emaitza eta errorea emateko. Ziurta ezazu zuk kalkula ditzakezula zero edo zero baino txikiagoak diren zenbaki guztien erro karratuak. Azken kasu horretan, erakutsi emaitza zenbaki irudikari bat bezala. Esate baterako, monitorean -4ren erro karratuaren itzultzeak $2i$ izan behar du. Proba ezazu zure programa $a = 0, 2, 4$ eta -9 balioztatuz, $\varepsilon_e = 10^{-4}$ hartuz.

17. *Zatikako funtzioak*, batzuetan, oso erabilgarriak dira menpeko aldagaiaren eta aldagai askearen arteko erlazioa ekuazio bakar batean adierazi ezin dugunean. Adibidez, honela deskriba liteke espazio-suziri baten abiadura:

$$v(t) = \begin{cases} 11t^2 - 5t, & 0 \leq t \leq 10 \\ 1100 - 5t, & 10 < t \leq 20 \\ 50t + 2(t-20)^2, & 20 < t \leq 30 \\ 1520e^{-0.2(t-30)}, & t > 30 \\ 0, & \text{beste kasu batean.} \end{cases}$$

Garatu M fitxategi bat v kalkulatzeko t-ren funtzioan. Orduan, erabili $v = v(t)$ funtzio hori, grafiko bat sortzeko $t = -5$etik $t = 50$ arte (t abszisa eta v ordenatua erakutsiz).

3. kapitulua

Ordenagailuaren aritmetika eta errorearen analisia

Soluzio bat lortzeko erabiltzen dugun ordenagailua tresna inperfektua da. Izan ere, mugatuta dago hark duen gaitasuna zenbakiak zehaztasunez adierazteko. Ondorioz, makinak berak lortzen dituen emaitzek erroreak dituzte.

Bestalde, metodo hurbilduak erabiltzen baditugu, erroreak sortzen dira. Adibidez, abiaduraren deribatua kalkulatzeko (puenting-jauzilariaren adibidean bezala) diferentzia finituak erabiltzean:

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}.$$

Auzia da, beraz, nola maneiatu horrelako ziurgabetasuna.

3.1. Algoritmoak eta erroreak

Ordenagailu bat erabiltzen denean problema baten zenbakizko soluzio bat lortzeko, programak gauzatzen ditu erabilitako zenbakizko metodoari elkartutako eragiketak. Zenbakizko metodo batzuk errazak dira inplementatzeko, baina batzuetan zenbakizko prozedurak zailak dira programatzeko.

Zenbakizko metodo bat programatu baino lehen, oso onuragarria da zenbakizko metodoa inplementatzeko jarraitu behar ditugun urrats guztiak planifikatzea. Horrelako plan bati *algoritmo* deritzogu, eta soluziora heltzeko urratsez urratseko instrukzioen bilduma da. Algoritmoak xehetasun-maila batzuetan idatz daitezke. Adibidez, jo dezagun $ax^2 + bx + c = 0$

ekuazio koadratikoa ebatzi nahi dugula, soluzio hauek kalkulatuz, algoritmo batez:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

3.1. algoritmoa. Ekuazio koadratiko baten erro errealak ebazteko algoritmoa.

Ekuazio koadratikoaren a, b eta c konstanteak emanda daude.

1. Kalkulatu diskriminantearen balioa, $D = b^2 - 4ac$.
2. $D > 0$ bada, kalkulatu bi erroak goiko adierazpenak erabiliz.
3. $D = 0$ bada, kalkulatu $x = -b/2a$ erroa, eta erakutsi mezu hau: «Ekuazioak erro bakar bat du».
4. $D < 0$ bada, erakutsi mezu hau: «Ekuazioak ez du erro errealik».

Behin algoritmoa asmatuz gero, inplementa daiteke ordenagailuko programa batean.

3.1.1. Erroreak

Zenbakizko kalkuluaren praktikan, kontuan hartu behar dugu ordenagailuak lortutako emaitzak ez direla soluzio matematiko zehatzak. Hainbat faktorerengatik, zenbakizko soluzioaren zehaztasuna txikitu daiteke, eta zailtasun hori ulertzeak maiz gida gaitzake zenbakizko algoritmo egokiak eraikitzera.

3.1. Definizioa. *Demagun $\widehat{p}$ balioa p-ren hurbilpen bat dela. Hurbilpenaren* ***errore absolutua*** *$E_p = |p - \widehat{p}|$ da eta* ***errore erlatiboa*** *$R_p = |p - \widehat{p}|/|p|$, $p \neq 0$.*

3.1. adibidea. *Izan bitez $x = 3.141592$ eta $\widehat{x} = 3.14$; orduan, errore absolutua hau da:*

$$E_x = |x - \widehat{x}| = |3.141592 - 3.14| = 0.001592$$

eta errore erlatiboa hau da:

$$R_x = \frac{|x - \widehat{x}|}{|x|} = \frac{0.001592}{3.141592} = 0.00507.$$

Puntu higikorreko adierazpenetan, nahiago da errore erlatiboaz lan egitea, hau mantisarekin zuzen erlazionatuta baitago.

Errore motak

Atal honetan aztertuko dugu zein diren erroreen iturriak.

1. **Ebatzi behar den problemako erroreak.**

   Eredu matematikoaren hurbiltze-erroreak izan daitezke. Adibidez, sarritan zeruko gorputzak hurbiltzen ditugu esferen bidez haien propietateak kalkulatzean; esate baterako, jakiteko asteroide batek talka egingo duen planeta batekin emandako data bat baino lehen.

   Problema bateko beste ohiko errore-iturri bat *datuetako erroreak* dira. Hau neurri fisikoetatik etor daiteke, horiek ez baitira inoiz zeharo zehatzak.

2. **Hurbiltze-erroreak.**

   Horrelako erroreak sortzen dira benetako funtzio bat balioztatzeko adierazpen hurbildu bat erabiltzen denean. Sarritan, bi errore mota hauek aurkitzen ditugu:

   - *Diskretizazio-erroreak* diskretizatzeko prozesu batetik sortzen dira; esate baterako, interpolazioa, diferentziazioa eta integrazioa.
   - *Konbergentzia-erroreak* metodo iteratiboetan sortzen dira. Esate baterako, problema ez-linealak orokorki prozesu iteratibo baten bidez ebazten dira. Horrelako prozesuak konbergitu egingo dira infinitu iteraziotan, baina moztu egiten da iterazio kopuru finitu bat egin ondoren. Metodo iteratiboak aljebra linealean sortzen dira.

3. **Biribiltze-erroreak.**

   Ordenagailu batez (edo kalkulagailu batez) zenbaki errealekin egindako edozein kalkuluk biribiltze-errore bat izaten du. Hori gertatzen da edozein ordenagailutan zenbaki errealen zehaztasun finituko adierazpenagatik; horrek eragiten du bai datuen adierazpenetan, bai ordenagailuaren aritmetikan.

Diskretizazio- eta konbergentzia-erroreak balioztatu behar dira erabilitako metodoaren analisi batez. Aldiz, biribiltze-erroreek egitura leunago bat dute, eta hori, batzuetan, ustiatu daiteke. Gure oinarrizko hipotesia izango da hurbiltze-erroreen tamaina biribiltze-erroreena baino handiagoa dela kalkulu erreal arrakastatsuetan.

3.1.2. Diskretizazio-erroreak

Ikus dezagun, adibide bat argitzeko, diskretizazio-erroreen portaera.

3.2. adibidea. *Izan bedi $x = x_0$ puntu bateko $f(x)$ funtzio baten $f'(x_0)$ deribatua hurbiltzearen problema. Esate baterako, izan bedi $f(x) = \cos(x)$ eta $x_0 = 0.5$. Orduan, $f(x_0) = \cos(0.5) = 0.87758\ldots$.*

Jo dezagun egoera berezi batean ezin dugula $f'(x_0)$ kalkulatu, edo bere kalkulua konputazionalki oso garestia dela, baina $f(x)$ balioa kalkula dezakegula x_0-tik x hurbil badago.

Ebazpena. Algoritmo erraz bat eraiki dezakegu Taylorren seriea erabiliz. Hots, izan bedi $h > 0$ balio txiki bat; orduan:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f''''(x_0) + \ldots.$$

Beraz,

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \left(\frac{h}{2}f''(x_0) + \frac{h^2}{6}f'''(x_0) + \frac{h^3}{24}f''''(x_0) + \ldots\right).$$

Gure algoritmo $f'(x_0)$ hurbiltzeko, hau kalkulatzean datza:

$$\frac{f(x_0 + h) - f(x_0)}{h}.$$

Lortutako hurbilpenak errore hau dauka:

$$\left|f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h}\right| = \left|\frac{h}{2}f''(x_0) + \frac{h^2}{6}f'''(x_0) + \frac{h^3}{24}f''''(x_0) + \ldots\right|.$$

Baldin ezagutzen badugu $f''(x_0)$ eta hura ez bada zero, orduan h txiki baterako errore hori, *diskretizazio-errorea*, hurbil dezakegu adierazpen honen bidez:

$$\left|f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h}\right| \approx \left|\frac{h}{2}f''(x_0)\right|.$$

Hurbiltzearen errorea taula honetan ikus daiteke, h-ren balio hauetarako:

h	Errorea
0.1	4.3044e-2
0.01	4.3799e-3
0.001	4.3871e-4
1.e-4	4.3878e-5
1.e-5	4.3879e-6
1.e-6	4.3887e-7
1.e-7	4.3963e-8

Ikus daitekeenez, ematen du diskretizazio errorea txikiago egiten dela h-rekin batera. Bigarren deribatua kalkulatuz, $\frac{1}{2}f''(x_0) \approx -0.43879$. Hots, $0.43879h$-k erroreak zehaztasunez hurbiltzen dituela ematen du. □

3.1.3. Biribiltze-errorearen eragin kaltegarria

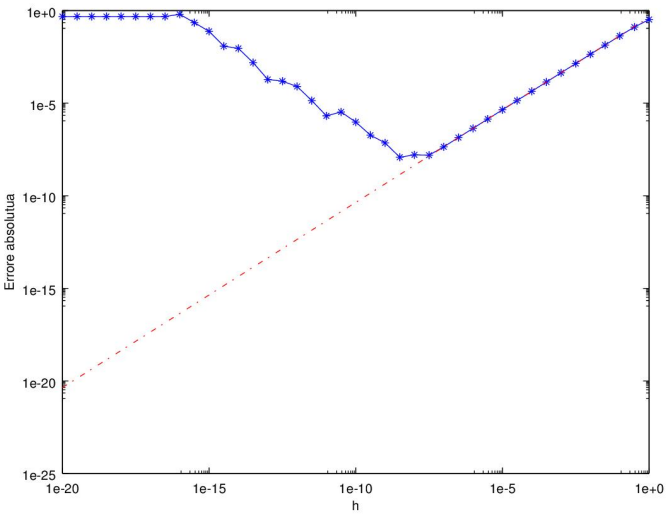
Itzul gaitezen 3.2. adibidera, ikusteko zelako ondorioak ekar ditzakeen biribiltze-erroreak.

3.3. adibidea. *3.2. adibideko taulako zenbakiek iradokitzen dute algoritmoak edozein zehaztasun lor dezakeela, betiere h nahiko txiki bat hartuz. Esate baterako, demagun hau nahi dugula:*

$$\left| -\sin(0.5) - \frac{\cos(0.5+h) - \cos(0.5)}{h} \right| < 10^{-10}.$$

Ebazpena. Adibide haren arabera, ematen du nahikoa dela $h \leq 10^{-10}/0.43879$ hartzea. Baina honako taula honen arabera, zer gertatzen den ikus dezakegu:

h	Errorea
1.e-8	1.6207e-08
1.e-9	7.1718e-08
1.e-10	9.5990e-07
1.e-11	2.0701e-06
1.e-12	7.9786e-05
1.e-13	1.9081e-04
1.e-15	0.075686
1.e-16	0.63080



3.1. irudia. Kurba urdin jarraituak diskretizazio-errorea ematen du, eta puntu-marra zuzen gorriak biribiltze-errorerik gabeko errore hori (MATLABeko `loglog` funtzioak egindako grafikoa).

Diskretizazio-errorea txikiago egiten da era ordenatu batean h txikiago egiten den ahala, eta biribiltze-errorea menperatzen du h handi samarra den bitartean. Baina, h 10^{-8} baliora gutxi gorabehera jaisten denean, diskretizazioaren erroreak oso balio txikia hartzen du, eta biribiltze-errorea diskretizazioarena menperatzen hasten da (errorerik handiena bihurtzen baita). □

3.1.4. Algoritmo bat ebaluatzeko irizpideak

Algoritmo baten kalitatearen eta erabilgarritasunaren ebaluazio bat irizpide batzuetan oinarri dezakegu.

Zehaztasuna

Hau elkarri lotuta dago errore motekin (geroago aztertuko dugu). Algoritmo baten ebaluazioan zenbakizko algoritmo baten zehaztasuna parametro erabakigarria dela kontuan hartzea oso garrantzitsua da, eta zenbakizko algoritmo bat diseinatzean beharrezkoa da kalkulu bat burutzean zelako errore tamaina espero dezakegun jakitea; ikus 3.2. adibidea.

Eraginkortasuna

Propietate teoriko handiak dituen zenbakizko algoritmo bat alferra da, bere kalkulua egiteko denbora gehiegi erabiltzen badu. Eraginkortasuna CPU denboraren eta memoria-espazioaren eskakizunen menpe dago. Algoritmo baten inplementazio xehetasun batzuek garrantzi handia izan dezakete eraginkortasunean. Beste propietate teoriko batzuk ere eraginkortasunaren adierazle izan daitezke; esate baterako, konbergentzia-ordena (ratioa).

Sarritan egin behar diren eragiketa aritmetikoen estimazio batek algoritmoaren eraginkortasunaren informazioa ematen digu. Adibidez, kalkulu arruntean, n mailako polinomio baten balioa kalkulatzeko n^2 ordenako eragiketak egiten dira:

$$p_n(x) = c_0 + c_1 x + \ldots + c_n x^n.$$

Aldiz, Horner-en metodoa erabiltzen badugu, n ordenako eragiketak:

$$p_n(x) = (\ldots((c_n x + c_{n-1})x + c_{n-2})x \ldots)x + c_0.$$

Sendotasuna

Sarritan, zenbakizko softwarea idazteko ahalegin handiena ez da gastatzen algoritmo baten mamia inplementatzeko, baizik eta ziurtatzeko berak lan egingo duela baldintza ahul guztien

menpean. Hots, errutinak, tolerantzia-maila baten barne, emaitza zuzen bat eman beharko luke, edo, huts egitekotan, dotoreziaz (esate baterako, abisu batez) egin beharko luke.

3.2. Puntu higikorreko sistemak

Euskaraz dezimalak komaz banantzen dira, baina testu honetan koma eta programazio-adibideetan puntua ez erabiltzearren puntuz bereiziko dira dezimalak. Hori dela eta, ingelesez *floating point* dena *puntu higikor* deituko dugu hemendik aurrera testu honetan.

Puntu higikorreko sistema bat lau baliok definitzen dute: (β, t, L, U), non

$$\begin{aligned} \beta &= \text{zenbaki-sistemaren oinarria;} \\ t &= \text{zehaztasuna (digitu kopurua);} \\ L &= e \text{ berretzailearen behe-bornea;} \\ U &= e \text{ berretzailearen goi-bornea.} \end{aligned}$$

Orokorrean, zera dugu:

$$\mathrm{fl}(x) = \pm \left(\frac{\widetilde{d}_0}{\beta^0} + \frac{\widetilde{d}_1}{\beta^1} + \ldots + \frac{\widetilde{d}_{t-1}}{\beta^{t-1}} \right) \times \beta^e,$$

β *oinarria* 1 baino handiagoa den zenbaki oso bat da, eta $\widetilde{d}_i$ zenbaki osoak $0 \leq \widetilde{d}_i \leq \beta - 1$ tartean daude. $\mathrm{fl}(x)$ zenbakia x-ren hurbilpen bat da.

Adierazpen horren bakartasuna ziurtatzeko, $\widetilde{d}_0 \neq 0$ betetzeko normalizatzen dugu, e berretzailea doituz aurreko zeroak kentzeko. Kontuan hartu, $\beta = 2$ kasuan izan ezik, $\widetilde{d}_0$ finkatu gabe dagoela, eta, ondorioz, gorde egin behar da, baita ere. Gainera, e berretzaileak $L \leq e \leq U$ bete behar du.

3.2.1. Inaustea eta biribiltzea

Nola gorde $x = \pm(d_0.d_1d_2d_3 \ldots d_{t-1}d_td_{t+1} \ldots) \times \beta^e$ zenbakia t digitu bakarrik erabiliz? Bi estrategia hauek erabil daitezke:

- inaustea: $d_t, d_{t+1}, d_{t+2}, d_{t+3} \ldots$, digituak baztertzea, $\widetilde{d}_i = d_i$ izanik, eta
$$\mathrm{fl}(x) = \pm(d_0.d_1d_2d_3 \ldots d_{t-1}) \times \beta^e;$$
- biribiltzea: aztertu d_t hurbilpena zehazteko:
$$\mathrm{fl}(x) = \begin{cases} \pm(d_0.d_1d_2d_3 \ldots d_{t-1}) \times \beta^e, & d_t < \beta/2 \text{ bada,} \\ \pm(d_0.d_1d_2d_3 \ldots d_{t-1} + \beta^{1-t}) \times \beta^e, & \text{bestela.} \end{cases}$$

3.4. adibidea. *Inaustearen eta biribiltzearen emaitzak, $\beta = 10$ eta $t = 3$ direnean.*

x	inausia	biribildua
6.552	6.55	6.55
-6.552	-6.55	-6.55
6.557	6.55	6.56
-6.557	-6.55	-6.56
6.592	6.59	6.59
6.595	6.59	6.60

3.5. adibidea. *Izan bedi sistema hamartarra, hots, $\beta = 10$, eta sistema horretako zenbaki hau:*

$$\frac{8}{3} = 2.6666\ldots = \left(\frac{2}{10^0} + \frac{6}{10^1} + \frac{6}{10^2} + \frac{6}{10^3} + \frac{6}{10^4} + \ldots\right) \times 10^0.$$

Zenbaki horrek 10 oinarriko berretura-serie infinitu bat ematen du, nahiz eta $\beta = 3$ oinarrian seriea finitua izan (hain zuzen, $2 \cdot 3^0 + 2 \cdot 3^{-1}$).

Adierazi zenbaki hamartar hori $t = 4$ erabiliz.

Ebazpena. Inausiz, zera dugu:

$$\frac{8}{3} \approx \left(\frac{2}{10^0} + \frac{6}{10^1} + \frac{6}{10^2} + \frac{6}{10^3}\right) \times 10^0 = 2.666 \times 10^0.$$

Bestalde, biribiltzean $d_t = 6 \geq \beta/2 = 5$ denez, zenbaki inausiari $\beta^{1-t} = 10^{-3}$ gehituta 2.667×10^0 ematen du.

Puntu higikorreko adierazpen hau ez da bakarra; adibidez, hau dugu:

$$2.667 \times 10^0 = 0.2667 \times 10^1.$$

Beraz, adierazpena normalizatzen dugu $d_0 \neq 0$ eta $1 \leq d_0 \leq 9$, $0 \leq d_i \leq 9$, $i = 1, \ldots, t-1$ nahitaez hartuz, edozein anbiguetate ezabatuz. Horrela, $x = 2.6666\ldots$ biribilduz, $\mathrm{fl}(x) = 2.667 \times 10^0$ dugu. □

0 zenbakia ezin da adierazi era normalizatu batean, eta $\pm\infty$ adierazten dira biten konbinazio berezi batzuen bidez, zenbaki-sistemaren arabera.

3.6. adibidea. *Izan bedi (jostailuzko) puntu higikorreko sistema hamartar bat $t = 4$, $U = 1$ eta $L = -2$ hartuz. Zein dira zenbaki handienak eta txikienak? Zenbat zenbaki ditu?*

Ebazpena. Horrela, hain zuzen, 2.666 zenbaki hamartarra adierazgarria da bere mantisak lau digitu dituelako eta $L \leq e \leq U$.

Kasu honetan, zenbaki handiena $99.99 < 10^2 = 100$ da, txikiena $-99.99 > -100$, eta zenbaki positibo txikiena $10^{-2} = 0.01$.

Zenbat zenbaki desberdin ditugu? Lehenengo digituak 9 balio desberdin har ditzake, beste digituetako bakoitzak 10 balio. Hortaz, $9 \times 10 \times 10 \times 10 = 9000$ zatiki normalizatu desberdin posible daude. Berretzaileak har ditzake $U - L + 1 = 4$ balio desberdin, beraz, guztira $4 \times 9000 = 36000$ zenbaki positibo desberdin daude. Beste horrenbeste zenbaki negatibo daude, eta, gainera, 0 zenbakia dugu. Ondorioz, 72001 zenbaki desberdin daude puntu higikorreko sistema honetan. □

3.2.2. Biribiltzearen unitatea

Puntu higikorreko adierazpenari dagokion errore erlatiboa hau da:

$$\frac{|\,\mathrm{fl}(x) - x|}{|x|}.$$

Zenbaki batetik hurrengorako mantisen arteko distantzia β^{1-t} da. Askotan, $t-1$ zenbakiari *digitu esanguratsuen kopurua* izendatzen diogu.

Ondorioz, (β, t, L, U) sistema orokor baterako *biribiltzearen unitatea* $\eta = \frac{1}{2}\beta^{1-t}$ da eta $\frac{|\,\mathrm{fl}(x) - x|}{|x|} \leq \eta$ betetzen da. Adibidez, $\beta = 10$, $t = 4$ eta $x = 12743.25$ badira, $\mathrm{fl}(x) = 1.274 \times 10^4$ dugu eta, ondorioz,

$$\frac{|\,\mathrm{fl}(x) - x|}{|x|} = \frac{|1.274 \times 10^4 - 1.274325 \times 10^4|}{|1.274325 \times 10^4|} = \frac{0.000325 \times 10^4}{1.274325 \times 10^4} = \frac{0.000325}{1.274325} < \frac{10^{-3}}{2} = \eta.$$

Zehaztasun bikoitzeko sistemaren kasuan, non $t - 1 = 52$ baita, $\eta = \frac{1}{2}2^{-52} = 1.1 \times 10^{-16}$ dugu.

3.3. Zenbaki bitarrak

Nahiz eta gizakiok zenbaki-sistema hamartarra erabili kalkulu aritmetikoetan, ordenagailu gehienek zenbaki-sistema bitarra erabiltzen dute. Ordenagailuak datu guztiak zenbaki bitar bihurtzen ditu. Kalkulu aritmetikoak 2 oinarrian egiten ditu, eta gero 10 oinarrira itzultzen ditu. Zehaztasuneko bederatzi zifra dezimal dauzkan ordenagailu batek emaitza hau eman zuen:

$$\sum_{i=1}^{100000} 0.1 = 9999.99447.$$

Batura 10000 izan behar zenez, gure lanetariko bat izango da hura gertatzeko arrazoia jakitea.

3.3.1. Zenbaki oso bitarrak

Sistema hamartarrean, 1563 zenbakia era garatuan idatzita hau da:

$$1563 = (1 \times 10^3) + (5 \times 10^2) + (6 \times 10^1) + (3 \times 10^0).$$

Baina, zenbaki bera sistema bitarrean idatzita hau da:

$$\begin{aligned} 1563 \;=\; & (1 \times 2^{10}) + (1 \times 2^9) + (0 \times 2^8) + (0 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) \\ & +(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0). \end{aligned}$$

Beraz, $1563{=}11000011011_{bi}$ dugu.

Hau da beste adibide bat:

$$(111\ldots11)_2 = 2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2^1 + 2^0 = (2^n - 1).$$

Azken berdintza lortzeko, kontuan hartu behar dugu $1{+}x{+}x^2{+}\ldots{+}x^n = \dfrac{x^{n+1}-1}{x-1}$ betetzen dela, $x \neq 1$ denean.

Nola lor dezakegu sistema bitarrera igarotzea?

1563 zenbakia birekin zatitzean, hau ateratzen dugu:

$$1563 = 2 \times 781 + 1.$$

Orain, 781 zenbakia birekin zatitzean, hau ateratzen dugu:

$$781 = 2 \times 390 + 1.$$

Eragiketa hori errepikatuz, zera lortzen da:

$$\begin{aligned} 390 &= 2 \times 195 + 0 \\ 195 &= 2 \times 97 + 1 \\ 97 &= 2 \times 48 + 1 \\ 48 &= 2 \times 24 + 0 \\ 24 &= 2 \times 12 + 0 \\ 12 &= 2 \times 6 + 0 \\ 6 &= 2 \times 3 + 0 \\ 3 &= 2 \times 1 + 1 \\ 1 &= 2 \times 0 + 1 \end{aligned}$$

eta behetik gorako hondarrak hartuz, zenbaki bitarra eraikitzen dugu: $1563{=}11000011011_{bi}$.

3.3.2. Zatiki bitarrak

R zenbaki erreal bat bada, non $0 < R < 1$, orduan $\{0, 1\}$ multzoan badago $d_1, d_2, \ldots, d_n, \ldots$ zifra segida bat hau betetzen duena:

$$R = (d_1 \times 2^{-1}) + (d_2 \times 2^{-2}) + \ldots + (d_n \times 2^{-n}) + \ldots . \tag{3.1}$$

Eskuineko adierazpena laburki emanda, hots, notazio zatikiar bitarrean, honela da:

$$R = 0.d_1 d_2 \ldots d_n \ldots_{bi} .$$

Askotan, zenbaki errealek 1 zifraren kopuru infinitu bat behar dute adierazpen bitarrean. Adibidez, 7/10, oinarri hamartarrean 0.7ren bidez adierazten da, baina bere adierazpen bitarrak 1 zifraren kopuru infinitu bat behar du:

$$\frac{7}{10} = 0.1\overline{0110}_{bi}.$$

Zatiki hori periodikoa da: 0110 lau zifrako taldea errepikatu egiten da, bukaerarik gabe.

Orain, algoritmo eraginkor bat garatu dezakegu 2 oinarriko adierazpenak aurkitzeko. Birekin (3.1) adierazpena biderkatzen badugu, hau dugu:

$$2R = d_1 + (d_2 \times 2^{-1}) + \ldots + (d_n \times 2^{-n+1}) + \ldots ,$$

non d_1 $2R$-ren zati osoa baita, $d_1 = \lfloor 2R \rfloor$. Adierazpen horren zatikia, $zat(2R)$, hau da:

$$Z_1 = zat(2R) = (d_2 \times 2^{-1}) + \ldots + (d_n \times 2^{-n+1}) + \ldots$$

eta (0,1) tartean dago. Azken adierazpen hori birekin biderkatuz, zera lortzen dugu:

$$2Z_1 = d_2 + (d_3 \times 2^{-1}) + \ldots + (d_n \times 2^{-n+2}) + \ldots .$$

Berdintza horren zati osoa hartuz, hots, $d_2 = \lfloor 2Z_1 \rfloor$. Prozesua aurrera joango da, seguraski bukaerarik gabe (R-ren adierazpena 2 oinarrian ez bada finitua, ezta periodikoa ere), eta era errepikarian honela sortzen ditu $\{d_k\}$ eta $\{Z_k\}$ bi segidak:

$$\begin{aligned} d_k &= \lfloor 2Z_{k-1} \rfloor \\ Z_k &= zat(2Z_{k-1}), \end{aligned}$$

non $d_1 = \lfloor 2R \rfloor$ eta $Z_1 = zat(2R)$ baitira. Ondorioz, R-ren adierazpen bitarra serie konbergente honek ematen digu:

$$R = \sum_{i=1}^{\infty} d_i (2)^{-i}$$

eta hau 1/2 arrazoiko serie geometriko baten azpiseriea da.

3.7. adibidea. *Kalkulatu 7/10en adierazpen bitarra.*

Ebazpena. $R = 7/10 = 0.7$ denez, honela kalkulatuko dugu:

$$\begin{array}{lll} 2R = 1.4 & d_1 = \lfloor 1.4 \rfloor = 1 & Z_1 = zat(1.4) = 0.4 \\ 2Z_1 = 0.8 & d_2 = \lfloor 0.8 \rfloor = 0 & Z_2 = zat(0.8) = 0.8 \\ 2Z_2 = 1.6 & d_3 = \lfloor 1.6 \rfloor = 1 & Z_3 = zat(1.6) = 0.6 \\ 2Z_3 = 1.2 & d_4 = \lfloor 1.2 \rfloor = 1 & Z_4 = zat(1.2) = 0.2 \\ 2Z_4 = 0.4 & d_5 = \lfloor 0.4 \rfloor = 0 & Z_5 = zat(0.4) = 0.4 \\ 2Z_5 = 0.8 & d_6 = \lfloor 0.8 \rfloor = 0 & Z_6 = zat(0.8) = 0.8 \\ 2Z_6 = 1.6 & d_7 = \lfloor 1.6 \rfloor = 1 & Z_7 = zat(1.6) = 0.6. \end{array}$$

Kontuan izan $2Z_2 = 1.6 = 2Z_6$ dela. Beraz, $d_k = d_{k+4}$ eta $Z_k = Z_{k+4}$ patroiak errepikatuko dira $k = 2, 3, 4, \ldots$-tarako. Ondorioz, $7/10 = 0.1\overline{0110}_{bi}$. □

Serie geometrikoa erabil dezakegu, adierazpen bitar bati dagokion zenbaki arrazional hamartarra kalkulatzeko.

3.8. adibidea. *Izan bedi $0.\overline{01}_{bi}$ zenbaki bitarra. Aurkitu zenbaki horri dagokion zenbaki arrazional hamartarra.*

Ebazpena. $0.\overline{01}_{bi}$ era garatuan idatziz, zera dugu:

$$\begin{aligned} 0.\overline{01}_{bi} &= (0 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4}) + \ldots \\ &= \sum_{i=1}^{\infty} (2^{-2})^i = -1 + \sum_{i=0}^{\infty} (2^{-2})^i \\ &= -1 + \frac{1}{1 - \frac{1}{4}} = -1 + \frac{4}{3} = \frac{1}{3}. \quad \square \end{aligned}$$

3.9. adibidea. Desplazamendu bitarra. *Demagun $S = 0.00000\overline{11000}_{bi}$ dela. Kalkula ezazu S zenbakia sistema hamartarrean.*

Ebazpena. Hau dugu:

$$2^5 S = 32S = 0.\overline{11000}_{bi}$$

Bestalde,

$$2^{10} S = 1024S = 11000.\overline{11000}_{bi}$$

Bigarren berdintzari lehenengoa kenduz:

$$1024S - 32S = 11000.\overline{11000}_{bi} - 0.\overline{11000}_{bi} = 11000_{bi} = 1 \times 2^4 + 1 \times 2^3 = 24,$$

eta orduan $992S = 24$, azkenik $S = 24/992 = 8/33$. □

3.3.3. Ordenagailuko zenbakiak

Ordenagailuek zenbaki errealetarako puntu higikorraren adierazpen bitarra erabiltzen dute. Zenbaki bitarrak 0 eta 1 digituek osatuta daudenez, puntuaren ezkerraldeko zenbakia 1 izateko normalizatzen dira beti. Beraz, *bit* (0 edo 1 digitu bitar bakoitza) hori ez da gorde behar (beti 1 baita). Ondorioz, zero ez diren zenbaki bitarrak honela gordeko dira:

$$\mathrm{fl}(x) = zeinu(x)(1+f)\times 2^e. \tag{3.2}$$

f zenbakiari *mantisa* deritzogu eta adierazpen bitar finitua da; e zenbakiari *berretzaile* deritzogu. Ordenagailuek zenbaki errealen azpimultzo txiki bat soilik erabiltzen dute, zeren f eta e izan ditzaketen zifra bitarren kopurua murriztea beharrezkoa baita. Adibidez, demagun era honetako zenbaki erreal positiboen multzoa:

$$1.d_1d_2d_3d_4\ _{bi}\times 2^e,$$

non $d_1, d_2, d_3, d_4 \in \{0,1\}$ eta $e \in \{-3,-2,-1,0,1,2,3,4\}$. Mantisarako $2^4 = 16$ aukera dugu, eta berretzailerako 8 aukera ere bai, horrek 128 zenbakien multzo bat ematen digu:

$$\{1.0000_{bi}\times 2^{-3}, 1.0001_{bi}\times 2^{-3}, \ldots, 1.1110_{bi}\times 2^4, 1.1111_{bi}\times 2^4\},$$

non, esate baterako:

$$\begin{aligned} 1.0000_{bi}\times 2^{-3} &= 1\times 2^{-3} = 0.1250\\ 1.1111_{bi}\times 2^4 &= (1+1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}+1\times 2^{-4})\times 2^4\\ &= 2^4+2^3+2^2+2^1+2^0 = 16+8+4+2+1 = 31. \end{aligned}$$

Ordenagailu batek 4 zifrako mantisa batez soilik $(1/10+1/5)+1/6$ eragiketa egin beharko balu, ordenagailuak zenbaki bitar hurbilenari biribilduko dio zenbaki erreal bakoitza; kasu honetan, hauek batzen ditu:

$$\begin{array}{lll} \frac{1}{10} = 0.1 = \frac{1.6}{2^4} = 1.6\times 2^{-4} = 1.\overline{1001}_{bi}\times 2^{-4} & \approx 1.1010_{bi}\times 2^{-4} & = 0.11010_{bi}\times 2^{-3}\\ \frac{1}{5} = 0.2 = \frac{1.6}{2^3} = 1.6\times 2^{-3} = 1.\overline{1001}_{bi}\times 2^{-3} & \approx 1.1010_{bi}\times 2^{-3} & = 1.1010_{bi}\times 2^{-3}, \end{array}$$

batuz hau lortzen da:

$$\frac{3}{10} \approx 10.01110_{bi}\times 2^{-3}.$$

Ordenagailuak erabakitzen du nola gorde behar duen $10.01110_{bi}\times 2^{-3} = 1.001110_{bi}\times 2^{-2}$ zenbakia; demagun $1.0100_{bi}\times 2^{-2}$ zenbakian biribiltzen duela. Hurrengo urratsa hau da:

$$\begin{array}{lll} \frac{3}{10} = 0.3 & \approx 1.0100_{bi}\times 2^{-2} & = 1.0100_{bi}\times 2^{-2}\\ \frac{1}{6} = \frac{1.\overline{3}}{2^3} = 1.\overline{3}\times 2^{-3} = 1.\overline{01}_{bi}\times 2^{-3} & \approx 1.0101_{bi}\times 2^{-3} & = 0.10101_{bi}\times 2^{-2}, \end{array}$$

batuz hau lortzen da:

$$\frac{7}{15} \approx 1.11101_{bi}\times 2^{-2}.$$

Ordenagailuak erabakitzen du nola gorde behar duen $1.11101_{bi}\times 2^{-2}$. Biribiltzen du $1.1111_{bi}\times 2^{-2}$ gordez. Beraz, hau da ordenagailuak batuketa problemari ematen dion soluzioa:

$$\frac{7}{15} \approx 1.1111_{bi}\times 2^{-2}.$$

Ordenagailuak sortutako errorea hau izan da:

$$\frac{7}{15} - 1.1111_{bi} \times 2^{-2} \approx -0.017708,$$

eta hori 7/15en %3.79 da.

3.3.4. Ordenagailu baten zehaztasuna

Mantisak 32 zifra baditu, 9 zifrara arteko zenbakiak gorde daitezke. Itzul gaitezen atalaren hasierara, ordenagailu baten bidez 1/10 elkarren ondoan 100000 bider batu nahi genuen lekura.

Demagun f mantisak 32 zifra bitar dauzkala. Beraz,

$$1 + f = 1.d_1 d_2 d_3 \ldots d_{31} d_{32bi}.$$

Zatiki bat era bitarrean adierazten badugu, adierazpen hori periodikoa izango da; adibidez:

$$\frac{1}{10} = 0.0\overline{0011}_{bi}.$$

Baina 32 zifrako mantisa bat erabiltzen dugunean, ordenagailuak trunkatzen du, eta barneko hurbilpen gisa hau erabiltzen du:

$$\frac{1}{10} = 1.10011001100110011001100110011001_{bi} \times 2^{-4},$$

eta horren errorea (hau da, aurreko bi zenbaki bitarren arteko diferentzia) hau da:

$$0.\overline{1001}_{bi} \times 2^{-36} \approx 8.731149137 \times 10^{-12}.$$

Arrazoi horregatik, ordenagailuak errore bat egin behar du, 1/10 zenbakia 100000 bider batzea eskatzen diogunean. Errore hori $(100000)(8.731149137 \times 10^{-12}) = 8.731149137 \times 10^{-7}$ izan behar zen gutxienez. Batura handitzen doan heinean, batura partzialak ere biribiltzen dira, eta batugaiak txikiak dira une horretan dagoen batura partzialarekin konparatuta. Ondorioz, trunkaketa gogorragoa da batugaiaren ekarpenerako. Errore horien guztien eragin konposatuak azken errore hau ematen du: $10000 - 9999.99447 = 5.53 \times 10^{-3}$.

3.3.5. Ordenagailuko puntu higikorreko zenbakiak

IEEE 754 puntu higikorreko aritmetikarako IEEEko estandarra da. Ordenagailuek bi modu dituzte zenbakiak adierazteko: *modu osoa* eta *puntu higikorreko modua*. Modu osoa erabiltzen da emaitza osoa izateko ziurtasuna duten eragiketak egiteko. Baina, normalean,

zientzietan eta ingeniaritzan, puntu higikorraren adierazpenak erabiltzen dira; eta (3.2) adierazpena erabiltzeak mugak jartzen ditu f mantisako zifren kopuruaren eta e berretzailearen heinaren gainean.

Ordenagailuak *zehaztasun bikoitzarekin* zenbaki errealak adierazteko 64 zifra erabiltzen ditu; orduan, lehenengo bita zenbakiaren zeinua gordetzeko izango da (0 + da eta 1 - da), hurrengo 11 bitak berretzailerako, eta azken 52 bitak mantisarako. Beraz, esanguratsutasunak bit inplizitu 1 dauka (finko alde osoan) eta 52 bit esplizitu, guztira 53 bit; ondorioz, zehaztasun osoak 53 bit dauzka, hau da ≈ 16 digitu dezimal (16 digitu esanguratsu), $\log_{10}(2^{53})$. Balio handiena, `realmax`, hau da:

$$+1.1111\ldots1111_{bi} \times 2^{+1023} = (2 - 2^{-52}) \times 2^{+1023} = 1.7977 \times 10^{308}.$$

Eta balio positibo txikiena, `realmin`, hau da:

$$+1.0000\ldots0000_{bi} \times 2^{-1022} = 2.2251 \times 10^{-308}.$$

Zenbakien arteko tartearen luzera haien tamainaren mendekoa da. Mantisaren balio txikiena $2^{-52} = 2.2204 \times 10^{-16}$. Balio horrek makinaren zehaztasuna, `eps`, (edo makinaren errorea, ε_M) ematen du 1 tamainako zenbakietarako; hori da bi zenbakien artean egon daitekeen diferentzia txikiena. Zenbakiaren tamaina handiagoa denean, diferentzia hori handitu egiten da.

MATLABek, berez, zehaztasun bikoitza erabiltzen du; hots, $(\beta, t, L, U) = (2, 52, -1022, 1023)$ puntu higikorreko sistema (IEEE sisteman, zenbakiaren 1 lehenengo digitua ez da zenbatzen t emateko).

Puntu higikorreko adierazpenari dagokion errore erlatiboa hau da:

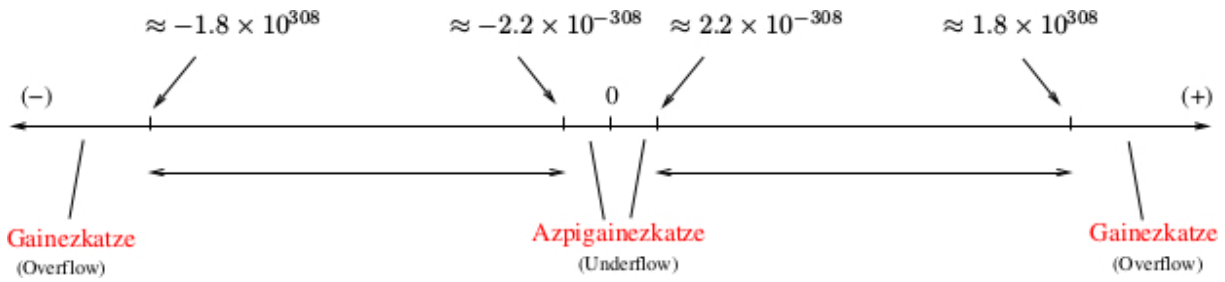
$$\frac{|\,\mathrm{fl}(x) - x|}{|x|}.$$

Gaurko puntu higikorreko sistemek, IEEEko estandarrek bezala, bermatzen dute errore erlatibo hori bornatuta dagoela zenbaki honetaz:

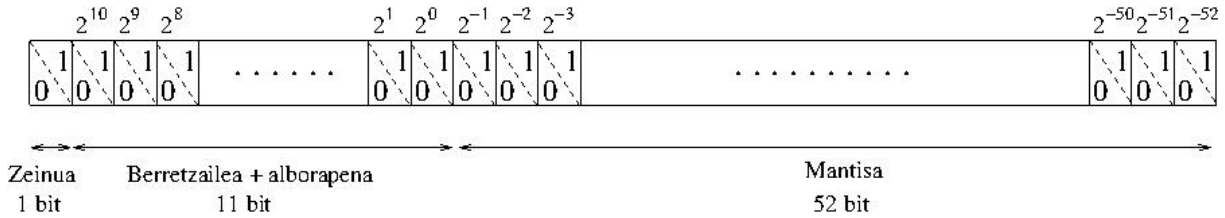
$$\eta = \frac{1}{2} \times 2^{-t}.$$

Zenbaki horri *biribiltzearen unitatea* deritzogu. Zehaztasun bikoitzaren kasuan, $\eta = \frac{1}{2} \times 2^{-52} \approx 1.1 \times 10^{-16}$ (beraz, 52 digitu bitar esanguratsu daude, edo 16 digitu hamartar).

Mantisaren balioa era bitarrean sartzen da. Berretzailearen balioa *alborapen* batekin sartzen da. Hemen, alborapenak esan nahi du berretzailearen balioari konstante bat batzen zaiola. Alborapena erabiltzen da berretzailearen zeinuan bit bat ez gastatzeko. Zehaztasun bikoitzean, notazio bitarrean 11 bitekin idatz dezakegun zenbaki handiena 2047 da (hots, 11 digituak 1 direnean). Erabiltzen den alborapena 1023 da (berretzaile txikiena 1 da). Hots, berretzailea 4 bada, gordetako balioa 4+1023=1027 izango da (1 gordeta badago, benetako



3.2. irudia. Zehaztasun bikoitzean adieraz daitezkeen zenbakien heina



3.3. irudia. Zenbaki baten biltzea, notazio bitarrean, IEEE-754 estandarrean.

berretzailea 1-1023=-1022 dugu). Beraz, ordenagailuan gordetako berretzaile txikiena -1022 izango da (1 bezala gordez), eta handiena 1023 (2046 bezala gordez).

Zehaztasun bakunarekin zenbaki errealak adierazteko, 32 zifra bitar (bit) erabiltzen dituzte ordenagailuek. Lehenengo bitak zenbakiaren zeinua gordetzen du. Hurrengo 8 bitak berretzailea gordetzeko erabiltzen dira. Eta azken 23 bitak mantisa gordetzeko erabiltzen dira. Horrek, zeroaz gain, 1.1755×10^{-38}tik 3.4028×10^{38}ra arteko zenbaki errealak adierazten uzten du; hots, 2^{-126}tik $(2 - 2^{-23})2^{127}$ra arte. Zenbakien arteko tartearen luzera haien tamainaren mendekoa da. Gorde dezakegun mantisaren balio txikiena $2^{-23} = 1.1921E-7$ da. Ondorioz, zehaztasun bakunak $(\beta, t, L, U) = (2, 23, -126, 127)$ puntu higikorreko sistema da.

Zehaztasun bakunean, non 8 bit erabiltzen baititugu, haiekin idatz dezakegun zenbaki handiena 255 da (hots, 11111111_{bi}) eta txikiena 1; ondorioz, 127 da berretzailearen balioa gordetzeko alborapena.

Zehaztasun bikoitzaren kasuan, *biribiltzearen unitatea* $\eta = \frac{1}{2} \times 2^{-23} \approx 6.0 \times 10^{-8}$ da (beraz, 23 digitu bitar esanguratsu daude, edo 7 digitu hamartar).

IEEEren biribiltze zehatza

Nahiz eta gure puntu higikorreko sisteman zenbakizko adierazpenak zehatzak izan, zenbaki horien eragiketa aritmetikoek biribiltze-erroreak sartzen dituzte. Errore horiek erlatiboki

nahiko handiak izan daitezke, *eskolta-digituak* erabiltzea izan ezik. Aparteko digitu horiek interim kalkuluetan erabiltzen dira. IEEE estandarrak *biribiltze zehatza* behar du, eta, horri esker, eragiketa aritmetiko bakoitzean errore erlatiboa bornatuta dago η balioaz.

3.10. adibidea. *Izan bedi puntu higikorreko sistema bat, non $\beta = 10$ oinarria eta $t = 4$ digituak baitira. Ondorioz, biribiltze unitatea $\eta = \frac{1}{2} \times 10^{-3}$ da. Aztertu biribiltze zehatzaren eragina.*

Ebazpena: Izan bitez

$$x = 0.1103 = 1.103 \times 10^{-1}, \quad y = 9.963 \times 10^{-3}.$$

Bi zenbaki horiek kenduz, balio zehatza $d = x - y = 0.100337$ da. Beraz, biribiltze zehatzak $\widehat{d} = 0.1003$ ematen du. Jakina, errore erlatiboa hau da:

$$\frac{|d - \widehat{d}|}{|d|} = \frac{|(x - y) - \mathrm{fl}(x - y)|}{|x - y|} = \frac{|0.100337 - 0.1003|}{0.100337} = 0.37 \times 10^{-3} < \eta.$$

Aldiz, baldin guk kentzen baditugu bi zenbaki horiek eskolta-digiturik gabe, $\overline{d} = 0.1103 - 0.0099 = 0.1004$ lortuko genuke. Orain, lortutako errore erlatiboa ez da η baino txikiago, zeren

$$\frac{|d - \overline{d}|}{|d|} = \frac{|0.100337 - 0.1004|}{0.100337} = 0.63 \times 10^{-3} > \eta.$$

Hortaz, eskolta-digituak erabili behar dira biribiltze zehatza lortzeko. □

Balio bereziak

Ez dira erabiltzen berretzaile posible guztiak. Zehaztasun bikoitzerako $2^{11} = 2048$ berretzaile desberdin ditugu, eta zehaztasun bakunerako $2^8 = 256$. Baina, 2046 eta 254 bakarrik, hurrenez hurren, erabiltzen ditugu. Hori da IEEE estandarrak berretzaileen muturreko puntuak ($b = 00000000000_{bi} = 0$ eta $b = 11111111111_{bi} = 2047$ zehaztasun bikoitzean eta $b = 00000000_{bi} = 0$ eta $b = 11111111_{bi} = 255$ zehaztasun bakunean) helburu berezietarako erreserbatzen dituelako.

Nola gordetzen dira 0 eta ∞? Hemen erabiltzen dira berretzailearen zenbaki berezi horiek.

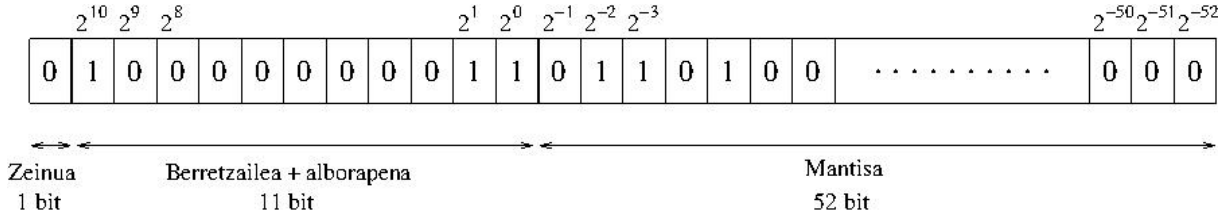
- 0-rako, $b = 0, f = 0$ jartzen dugu, zeinua edozein izanik.
- $\pm\infty$-rako, $b = 1 \ldots 1, f = 0$.

- $b = 1 \ldots 1, f \neq 0$ patroia konbentzioz `NaN` da (Not a Number); emaitza bat kalkula ezina denean agertzen da $(1/0, 0/0, \ldots)$.

3.11. adibidea. *Ikus dezagun 22.5 zenbakia nola gordetzen den zehaztasun bikoitzean, IEEE-754 estandarraren arabera. Lehenengo, zenbakia normalizatzen da (hots, alde osoan 1 zenbakia jartzen da) honela:*

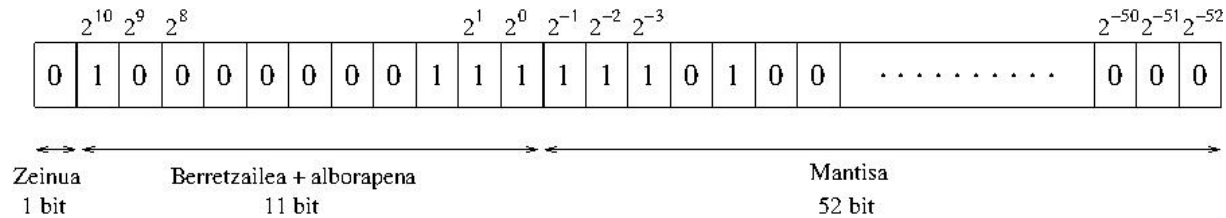
$$\frac{22.5}{2^4} 2^4 = 1.40625 \times 2^4.$$

Zehaztasun bikoitzean, berretzailea alborapenarekin 4+1023=1027 da, zeina era bitarrean 10000000011_{bi} *gordetzen baita. Mantisa 0.40625 da, zeina era bitarrean* $.01101000_{bi} \ldots 000$ *gordetzen baita. Zenbaki horren biltzea hau da:*



3.4. irudia. 22.5 zenbakiaren biltegia, notazio bitarrean, IEEE-754 estandarrean.

3.12. adibidea. *Zein zenbaki hamartar dago gordeta zehaztasun bikoitzeko biltegi honetan:*



Ebazpena. $b = 10000000111_{bi} = 1031$, beraz, $e = 1031 - 1023 = 8$.

Bestalde, $f = 1110100 \ldots 0_{bi} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} = 0.90625$. Eta zeinua + da.

Ondorioz, $+(1+f)2^e = +1.90625 \cdot 2^8 = 488$. □

3.4. Errorearen analisia

3.2. Definizioa. *$\widehat{p}$ zenbakia d zifra dezimal esanguratsuko p-ren hurbilpen bat dela esango dugu, hau betetzen duen zenbaki arrunt handiena d zenbakia bada:*

$$\frac{|p - \widehat{p}|}{|p|} < \frac{10^{-d}}{2}.$$

3.13. adibidea. $x = 3.141592$ *eta* $\widehat{x} = 3.14$ *badira, orduan* $|x - \widehat{x}|/|x| = 0.000507 < 10^{-2}/2$. *Beraz,* $\widehat{x}$ *bi zifra esanguratsuko* x*-ren hurbilpena da.*

Gainezkatze bat lortzen da zenbaki bat handiegia denean adierazteko erabilitako puntu higikor sisteman; esate baterako, $e > U$ denean. *Azpigainezkatze* bat lortzen da $e < L$ denean. Kalkulu bat egitean gainezkatze bat gertatzen denean, hori oso txarra da. Aldiz, hori ez da gertatzen azpigainezkatzearekin; sistemak orokorki jartzen du zenbakia 0-ra eta jarraitu egiten du (MATLABek hori egiten du, ezer esan gabe).

3.4.1. Zifra esanguratsuen ezeztapena

Errore hau gertatzen da bi zenbaki ia berdinak kentzen direnean.

Demagun $z = x - y$, non $x \approx y$. Orduan

$$|z - \mathrm{fl}(z)| \leq |x - \mathrm{fl}(x)| + |y - \mathrm{fl}(y)|,$$

eta, hortaz, errore erlatiboak hau betetzen du:

$$\frac{|z - \mathrm{fl}(z)|}{|z|} \leq \frac{|x - \mathrm{fl}(x)| + |y - \mathrm{fl}(y)|}{|x - y|}.$$

Zenbakitzailearekin ez da egon behar inolako arazorik puntu higikorreko sistemak x eta y ondo adierazten baditu. Baina, izendatzailea zerotik oso hurbil dago $x \approx y$ bada, eta orduan z-ren errore erlatiboa oso handia bihur daiteke.

Jo dezagun $p = 3.1415926536$ eta $q = 3.1415957341$ zenbakiak berdintsuak direla, eta 11 zifra dezimaleko zehaztasunez adierazita daudela. Haien kendura kalkulatzen badugu, $p - q = -0.0000030805$, ikusiko dugu p-ren eta q-ren lehenengo sei zifrak berdinak direla; bere diferentziak $p - q$ bost zifra dezimal bakarrik dauzka; fenomeno horri ***zifra esanguratsuen ezeztapena*** edo ***galera*** deritzogu, eta kontuz ibili behar dugu, zeren konturatu gabe kalkulatutako azken emaitzaren zehaztasuna txikiagotu egin baitezake.

3.14. adibidea. *Izan bitez* $f(x) = x(\sqrt{x+1} - \sqrt{x})$ *eta* $g(x) = x/(\sqrt{x+1} + \sqrt{x})$ *algebraikoki funtzio baliokideak (egiaztatu). Konparatuko ditugu* $f(500)$ *eta* $g(500)$*, sei zifra esanguratsuko biribiltzea erabiliz.*

Ebazpena. Lehenengo funtzioarekin hau lortzen dugu:

$$f(500) = 500(\sqrt{501} - \sqrt{500}) = 500(22.3830 - 22.3607) = 500(0.0223) = 11.1500.$$

Orain, $g(x)$-rekin zera dugu:

$$g(500) = \frac{500}{\sqrt{501} + \sqrt{500}} = \frac{500}{22.3830 + 22.3607} = \frac{500}{44.7437} = 11.1748.$$

Bigarrenaren errore absolutua txikiagoa da. Hori da lortuko genukeena 11.174755300747198 ... emaitza zehatza sei zifra esanguratsuetara biribilduz. □

3.15. adibidea. *Demagun* $y = \sqrt{x+1} - \sqrt{x}$ *kalkulatu nahi dugula* $x = 100000$ *bost digituzko aritmetika hamartarrean (*$\beta = 10, t = 5$*).*

Ebazpena. Bistan dago 100001 zenbakia ezin dela zehazki adierazi puntu higikorreko sistema horretan, eta bere adierazpena (inausiz edo biribilduz) 100000 da. Beste hitzez, x-ren balio horretarako sistema horretan $x+1 = x$ dugu, eta kalkuluak sistema horretan $\sqrt{x+1}-\sqrt{x} = 0$ ematen du.

Askoz hobe egin dezakegu identitate hau erabiltzen badugu:

$$\frac{(\sqrt{x+1} - \sqrt{x})(\sqrt{x+1} + \sqrt{x})}{(\sqrt{x+1} + \sqrt{x})} = \frac{1}{\sqrt{x+1} + \sqrt{x}}.$$

Formula horren eskuineko adierazpena erabiliz 5 digitu esanguratsurekin, 1.5811×10^{-3}, eta hori da hain zuzen, balio zuzena 5 digitu hamartarrekin. □

Badago beste era bat ere ezeztatze-erroreak saihesteko, eta teknika ezagun bat Taylorren garapena erabiltzea da.

3.16. adibidea. *Demagun hau kalkulatu nahi dugula:*

$$y = \sinh(x) = \frac{1}{2}(e^x - e^{-x}).$$

Guk ziurtatu nahi dugu x *guztietarako emaitza zehatz (doi) bat ematen duela (ez derrigorrez zuzena).*

Ebazpena. Goiko formula zuzenean aplikatu ahal izan arren, guk lor dezakegu formula sendoago bat x ia zero den kasurako, eta Taylorren garapen honetatik lortzen da:

$$\sinh(x) = x + \frac{x^3}{6} + \frac{\psi^5}{120},$$

ψ baterako, non $|\psi| \leq |x|$. Orduan, $x + x^3/6$ formulak hurbilpen efikaza emango luke $|x|$ nahiko txikia denean, zeren trunkatze-errorea borna baitaiteke $\frac{x^5}{120}$ balioaz, eta biribiltze-errorea jadanik ez da arazo bat. □

Horner-en metodoa

Polinomio bat balioztatzeko orduan, emaitza hobeak lortuko ditugu Hornerren metodoa (biderkadura ahokatua) erabiltzen badugu.

Izan bedi $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots + a_nx^n$ polinomioa; demagun haren balioa x_0 baliorako, $p(x_0)$, kalkulatu nahi dugula. Hori lortzeko, segida hau definituko dugu:

$$\begin{aligned} b_n &= a_n \\ b_{n-1} &= a_{n-1} + b_n x_0 \\ b_{n-2} &= a_{n-2} + b_{n-1} x_0 \\ &\vdots \\ b_1 &= a_1 + b_2 x_0 \\ b_0 &= a_0 + b_1 x_0, \end{aligned}$$

orduan, $p(x_0)$-ren balioa b_0 da.

Metodoa ulertzeko, kontuan izan polinomioa biderkadura ahokatu hau bezala idatz dezakegula:

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \ldots + x(a_{n-1} + a_n x)\ldots))).$$

Gero, b_i banan-banan ordezkatuz, zera dugu:

$$\begin{aligned} p(x_0) &= a_0 + x_0(a_1 + x_0(a_2 + x_0(a_3 + \ldots + x_0(a_{n-1} + b_n x_0)\ldots))) \\ &= a_0 + x_0(a_1 + x_0(a_2 + x_0(a_3 + \ldots + x_0(b_{n-1})\ldots))) \\ &\vdots \\ &= a_0 + x_0(a_1 + x_0 b_2) \\ &= a_0 + x_0 b_1 \\ &= b_0. \end{aligned}$$

3.17. adibidea. *Izan bitez* $P(x) = -1 + 3x - 3x^2 + x^3$ *eta* $Q(x) = -1 + x(3 + x(-3 + x))$. *Hiru zifra esanguratsutarako biribiltzea erabiliz,* $P(2.19)$ *eta* $Q(2.19)$ *kalkulatuko ditugu, eta bi hurbilpen horiek konparatuko ditugu benetako balioekin,* $P(2.19) = Q(2.19) = 1.685159$.

Ebazpena:

$$\begin{aligned} P(2.19) &= (2.19)^3 - 3(2.19)^2 + 3(2.19) - 1 \\ &\approx -1 + 6.57 - 14.4 + 10.5 = 1.67. \end{aligned}$$

Hornerren metodoaz, honela kalkulatzen da:

$$\begin{aligned} b_3 &= a_3 = 1 \\ b_2 &= a_2 + b_3 x_0 = -3 + 1(2.19) = -0.81 \\ b_1 &= a_1 + b_2 x_0 = 3 - 0.81(2.19) \approx 1.23 \\ b_0 &= a_0 + b_1 x_0 \approx -1 + 1.23(2.19) \approx 1.69, \end{aligned}$$

hots, $Q(2.19) \approx 1.69$.

Errore absolutuak 0.015159 eta 0.004841 dira, hurrenez hurren. Beraz, $Q(2.19) \approx 1.69$ hurbilpena hobea da. □

3.4.2. Trunkatze-errorea

Trunkatze-errorearen nozioa orokorki erlazionatuta dago adierazpen zail bat beste formula errazago batez ordezkatzearekin. Adibidez, funtzio bat Taylorren polinomio batez ordezkatzen dugunean; esate baterako, adierazpen honen kasuan:

$$e^{x^2} = 1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \frac{x^8}{4!} + \dots,$$

bere integrala zenbakizko kalkuluen bitartez aurkitzeko orduan, lehenengo bost gaien baturarekin ordezka dezakegu: $1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \frac{x^8}{4!}$.

3.4.3. $O(h^n)$ hurbiltze-ordena

Jakin badakigu $\left\{\frac{1}{n^2}\right\}_{n=1}^{\infty}$ eta $\left\{\frac{1}{n}\right\}_{n=1}^{\infty}$ segidak konbergenteak direla; hala ere, lehenengoak bigarrenak baino azkarrago jotzen du zerora. Jarraian, segida baten konbergentziaren azkartasuna deskribatzeko beharrezko terminologia eta notazioa sartuko ditugu.

3.3. Definizioa. *$f(h)$ funtzioa $h \to 0$ denean, $g(h)$ ordenakoa dela esango dugu, eta $f(h) = O(g(h))$ idatziko, C eta c konstanteak existitzen direnean, non hau betetzen baita:*

$$|f(h)| \leq C|g(h)|, \qquad |h| \leq c \ \ bada.$$

3.18. adibidea. *Izan bitez $f(x) = x^3 + 2x^2$ eta $g(x) = x^2$. Izan ere, $|x| \leq 1$-erako $x^3 \leq x^2$ da; $|x| \leq 1$-erako $x^3 + 2x^2 \leq 3x^2$ lortzen dugu. Beraz, $f(x) = O(g(x))$.* □

Landau-ren $O(\cdot)$ notazioa deritzo, eta infinituetarako limiteetan ere erabiltzen da. Oso erabilgarria da, funtzio baten handitzearen abiadura deskribatzeko, funtzio ezagun batekin konparatzea (x^n, $x^{1/n}$, a^x, $\log_a x$, eta abar).

Segiden konbergentziaren abiadura ere deskriba dezakegu antzeko era batean.

3.4. Definizioa. *Izan bitez $\{x_n\}_{n=1}^{\infty}$ eta $\{y_n\}_{n=1}^{\infty}$ segidak. $\{x_n\}$ segida $\{y_n\}$ ordenakoa dela esaten da, eta $x_n = O(y_n)$ idazten, C eta N konstanteak existitzen badira, non hau betetzen baita:*

$$|x_n| \leq C|y_n|, \qquad n \geq N \ \ bada.$$

3.19. adibidea. $\frac{n^2-1}{n^3} = O\left(\frac{1}{n}\right)$*; izan ere,* $\frac{n^2-1}{n^3} \leq \frac{n^2}{n^3} = \frac{1}{n}$, $n \geq 1$ *bada.* □

3.5. Definizioa. *Demagun $p(h)$ funtzioak beste $f(h)$ funtzio bat hurbiltzen duela, eta existitzen direla $M > 0$ zenbaki erreal bat eta n zenbaki arrunt bat non hau betetzen baita:*

$$\frac{|f(h) - p(h)|}{|h^n|} \leq M, \qquad h \quad \textit{nahiko txiki baterako.}$$

Orduan, $p(h)$ funtzioak $f(h)$ hurbiltzen duela esaten da $O(h^n)$ hurbiltze-ordenarekin, eta honela idazten da:

$$f(h) = p(h) + O(h^n).$$

Goiko desberdintza $|f(h) - p(h)| \leq M|h^n|$ bezala idatziz, $O(h^n)$ adierazpenak $M|h^n|$ errorebornearen lekua betetzen duela dakusagu.

3.1. teorema. *Demagun $f(h) = p(h) + O(h^n)$ eta $g(h) = q(h) + O(h^m)$, eta izan bedi $r = \min\{m, n\}$. Orduan:*

$$\begin{aligned} f(h) + g(h) &= p(h) + q(h) + O(h^r), \\ f(h)g(h) &= p(h)q(h) + O(h^r), \\ f(h)/g(h) &= p(h)/q(h) + O(h^r), \qquad g(h) \neq 0 \ \textit{eta}\ q(h) \neq 0 \ \ \textit{badira.} \end{aligned}$$

Oso interesgarria da aintzat hartzea $f(x)$ funtzioaren Taylorren polinomioen bidezko n-garren hurbilpena $p(x)$ den kasua. Orduan Taylorren formularen hondarra $O(h^{n+1})$-ren bidez adierazten da, eta idatzi gabeko gai guztiak ordezkatzen ditu, hau da, h^{n+1} berreketa daukana eta goi-ordenakoak. Taylorren formularen hondarrak jotzen du zerora $h \to 0$ denean, h^{n+1}-en abiadura berdinarekin, adierazpen honek erakusten duen bezala:

$$O(h^{n+1}) \approx Mh^{n+1} \approx \frac{f^{(n+1)}(c)}{(n+1)!}h^{n+1},$$

zeina baliozkoa baita h nahiko txikia denean. Beste hitz batzuetan esanda, $O(h^{n+1})$ gaiak Mh^{n+1} kopurua ordezkatzen du, non M konstantea baita.

3.2. Taylorren teorema. *Demagun $f \in C^{n+1}[a, b]$. Baldin x_0 eta $x = x_0 + h$ $[a, b]$ tartean badaude, orduan, hau betetzen da:*

$$f(x_0 + h) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}h^i + O(h^{n+1}).$$

Kalkuluetan propietate hauek erabiltzen dira:

(i) $O(h^p) + O(h^p) = O(h^p)$.

(ii) $O(h^p) + O(h^q) = O(h^r)$, non $r = \min\{p, q\}$.

(iii) $O(h^p)O(h^q) = O(h^s)$, non $s = p + q$.

Honako adibide honek 3.1. teorema argitzen du.

3.20. adibidea. *Izan bitez Taylorren garapen hauek:*

$$e^h = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + O(h^4) \quad eta \quad \cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6).$$

Aurki itzazu baturaren eta biderkaduraren hurbiltze-ordenak.

Ebazpena:

$$\begin{aligned} e^h + \cos(h) &= 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + O(h^4) + 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6) \\ &= 2 + h + \frac{h^3}{3!} + O(h^4) + \frac{h^4}{4!} + O(h^6). \end{aligned}$$

Eta $O(h^4) + \dfrac{h^4}{4!} = O(h^4)$ eta $O(h^4) + O(h^6) = O(h^4)$ direnez, zera dugu:

$$e^h + \cos(h) = 2 + h + \frac{h^3}{3!} + O(h^4),$$

eta, ondorioz, $O(h^4)$ hurbiltze-ordena da.

Biderkaduran, antzeko eran egiten da.

$$\begin{aligned} e^h \cos(h) &= \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + O(h^4)\right)\left(1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6)\right) \\ &= \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!}\right)\left(1 - \frac{h^2}{2!} + \frac{h^4}{4!}\right) \\ &\quad + \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!}\right)O(h^6) + \left(1 - \frac{h^2}{2!} + \frac{h^4}{4!}\right)O(h^4) \\ &\quad + O(h^4)O(h^6) \\ &= 1 + h - \frac{h^3}{3} - \frac{5h^4}{24} - \frac{h^5}{24} + \frac{h^6}{48} + \frac{h^7}{144} \\ &\quad + O(h^6) + O(h^4) + O(h^4)O(h^6). \end{aligned}$$

Izan ere, $O(h^4)O(h^6) = O(h^{10})$ eta hau betetzen da:

$$-\frac{5h^4}{24} - \frac{h^5}{24} + \frac{h^6}{48} + \frac{h^7}{144} + O(h^6) + O(h^4) + O(h^{10}) = O(h^4),$$

honako erlazio hau lortzen da:

$$e^h \cos(h) = 1 + h - \frac{h^3}{3} + O(h^4).$$

Ondorioz, biderkadura $O(h^4)$ hurbiltze-ordenakoa da. □

3.4.4. Segida baten hurbiltze-ordena

3.6. Definizioa. *Demagun* $\lim_{n\to\infty} x_n = x$ *dugula eta* $\{r_n\}_{n=1}^{\infty}$ *segida bat dela, non* $\lim_{n\to\infty} r_n = 0$. *Orduan, esango dugu* $\{x_n\}_{n=1}^{\infty}$ *segidak* x*-ra jotzen duela* $O(r_n)$ *hurbiltze-ordenarekin,* $K > 0$ *konstante bat existitzen bada hau betetzen duena:*

$$\frac{|x_n - x|}{|r_n|} \leq K \quad n \text{ nahiko handi baterako.}$$

Hori $x_n = x + O(r_n)$ *idatziz adieraziko dugu, edo* $x_n \to x$ $O(r_n)$ *hurbiltze-ordenarekin.*

3.21. adibidea. *Izan bitez* $x_n = \cos(n)/n^2$ *eta* $r_n = 1/n^2$, *orduan* $\lim_{n\to\infty} x_n = 0$ *dugu* $O(1/n^2)$ *hurbiltze-ordenarekin. Hori erlazio honetatik ateratzen da:*

$$\frac{|\cos(n)/n^2|}{|1/n^2|} = |\cos(n)| \leq 1 \quad n \text{ guztietarako.} \quad \square$$

3.4.5. Errorearen hedapena

Atal honetan aztertuko dugu nola heda daitezkeen erroreak segidako eragiketen kate batean. Jo ditzagun p eta q zenbakien batura (balio zehatzak izanik), haien $\widehat{p}$ eta $\widehat{q}$ balio hurbilduak, eta azken horien erroreak, ε_p eta ε_q, hurrenez hurren. Hots, $p = \widehat{p} + \varepsilon_p$ eta $q = \widehat{q} + \varepsilon_q$. Batura horiek gaiez gai batuz, hau dugu:

$$p + q = (\widehat{p} + \varepsilon_p) + (\widehat{q} + \varepsilon_q) = (\widehat{p} + \widehat{q}) + (\varepsilon_p + \varepsilon_q).$$

Beraz, batuketa batean, errorea da batugaien erroreen batura.

Biderketa batean, errorearen hedapena konplexuagoa da. Biderkadura hau da:

$$pq = (\widehat{p} + \varepsilon_p)(\widehat{q} + \varepsilon_q) = \widehat{p}\widehat{q} + \widehat{p}\varepsilon_q + \widehat{q}\varepsilon_p + \varepsilon_p\varepsilon_q.$$

Beraz, $|\widehat{p}|$ eta $|\widehat{q}|$ 1 baino handiagoak badira, $\widehat{p}\varepsilon_q$ eta $\widehat{q}\varepsilon_p$ gaiek handitu ditzakete jatorrizko ε_p eta ε_q erroreak. Berrordenatuz, hau lortuko dugu:

$$pq - \widehat{p}\widehat{q} = \widehat{p}\varepsilon_q + \widehat{q}\varepsilon_p + \varepsilon_p\varepsilon_q.$$

Demagun $p \neq 0$ eta $q \neq 0$; orduan, zera betetzen da:

$$R_{pq} = \frac{pq - \widehat{p}\widehat{q}}{pq} = \frac{\widehat{p}\varepsilon_q + \widehat{q}\varepsilon_p + \varepsilon_p\varepsilon_q}{pq} = \frac{\widehat{p}\varepsilon_q}{pq} + \frac{\widehat{q}\varepsilon_p}{pq} + \frac{\varepsilon_p\varepsilon_q}{pq}. \tag{3.3}$$

Gainera, suposatzen badugu $\widehat{p}$ eta $\widehat{q}$ p-ren eta q-ren hurbilpen onak direla, orduan, $\widehat{p}/p \approx 1$, $\widehat{q}/q \approx 1$ eta $R_pR_q = (\varepsilon_p/p)(\varepsilon_q/q) \approx 0$ (R_p eta R_q $\widehat{p}$-ren eta $\widehat{q}$-ren errore erlatiboak dira). Hurbilpen horiek (3.3) adierazpenean ordezkatuz, hau dugu:

$$R_{pq} = \frac{pq - \widehat{p}\widehat{q}}{pq} \approx \frac{\varepsilon_q}{q} + \frac{\varepsilon_p}{p} = R_q + R_p.$$

Beraz, $\widehat{p}\widehat{q}$ biderkadurari dagokion errore erlatiboa, gutxi gorabehera, $\widehat{p}$ eta $\widehat{q}$ faktoreei dagozkien errore erlatiboen batura da.

Normalean, datuen hasierako erroreak hedatzen dira eragiketen kate batean zehar. Edozein zenbakizko prozesutan, kualitate desiragarria da hasierako baldintzetako errore txiki batek azken emaitzan errore txikiak eragitea. Algoritmo batek propietate hori badauka, *egonkorra* dela esango dugu; bestela, *ezegonkorra* deritzo. Ahal den neurrian, metodo egonkorrak aukeratuko ditugu.

3.7. Definizioa. *Demagun ε hasierako errore bat dela, eta $\varepsilon(n)$ adierazpenak n eragiketa ondoren errore horren hazkuntza adierazten duela. Baldin $|\varepsilon(n)| \approx n\varepsilon$ bada, hazkuntza lineala dela esaten da. Baldin $|\varepsilon(n)| \approx K^n\varepsilon$ bada, hazkuntza esponentziala dela esaten da. Baldin $K > 1$ bada, orduan, $n \to \infty$ denean, errore esponentziala bornerik gabe hazten da; baina, $0 < K < 1$ bada, orduan, $n \to \infty$ denean, errore esponentzialak zerora jotzen du.*

Ondorengo bi adibideen bitartez ikusten da hasierako errore bat modu egonkorrean edo ezegonkorrean heda daitekeela. Lehenengoan hiru algoritmo aurkezten dira, eta ikusiko dugu hiru horien bidez segida berdina lortuko genukeela, eragiketak era zehatzean egingo bagenitu. Bigarrenean, hastapen-baldintzetako errore txikien hedapena aztertuko dugu.

3.22. adibidea. *Erakutsiko dugu $\{1/3^n\}_{n=0}^{\infty}$ segidaren gaiak hiru eskema desberdin erabiliz gara ditzakegula.*

Ebazpena. Izan bitez hiru adierazpen hauek:

$$r_0 = 1 \qquad r_n = \frac{1}{3}r_{n-1} \qquad n = 1, 2, \ldots$$

$$p_0 = 1, \quad p_1 = \frac{1}{3} \quad p_n = \frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2} \quad n = 2, 3, \ldots$$

$$q_0 = 1, \quad q_1 = \frac{1}{3} \quad q_n = \frac{10}{3}q_{n-1} - q_{n-2} \quad n = 2, 3 \ldots,$$

$\{r_n\}$-rena bistan dago. Jarraian egiaztatuko dugu $\{p_n\}$ segidan diferentzien kendurak $p_n = A(1/3^n) + B$ soluzio orokorra daukala.

$$\begin{aligned}\frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2} &= \frac{4}{3}\left(\frac{A}{3^{n-1}} + B\right) - \frac{1}{3}\left(\frac{A}{3^{n-2}} + B\right) \\ &= \left(\frac{4}{3^n} - \frac{3}{3^n}\right)A + \left(\frac{4}{3} - \frac{1}{3}\right)B = A\frac{1}{3^n} + B = p_n.\end{aligned}$$

$A = 1$ eta $B = 0$ hartzen baditugu, $p_0 = 1$ eta $p_1 = 1/3$ ditugu, eta hori da gure segida.

Jarraian egiaztatuko dugu $\{q_n\}$ segidan diferentzien kendurak $q_n = A(1/3^n)+B3^n$ soluzio orokorra daukala.

$$\begin{aligned}\frac{10}{3}q_{n-1} - q_{n-2} &= \frac{10}{3}\left(\frac{A}{3^{n-1}} + B3^{n-1}\right) - \left(\frac{A}{3^{n-2}} + B3^{n-2}\right)\\ &= \left(\frac{10}{3^n} - \frac{9}{3^n}\right)A + (10-1)3^{n-2}B\\ &= A\frac{1}{3^n} + B3^n = q_n.\end{aligned}$$

$A = 1$ eta $B = 0$ hartzen baditugu, $q_0 = 1$ eta $q_1 = 1/3$ ditugu, eta horrek gure segida garatzen du. □

3.23. adibidea. $\{x_n\} = \{1/3^n\}$ *segidaren hurbilpenak garatuko ditugu adierazpen hauek erabiliz:*

$$r_0 = 0.99996 \qquad r_n = \frac{1}{3}r_{n-1} \qquad n = 1, 2, \ldots$$

$$p_0 = 1, \quad p_1 = 0.33332 \quad p_n = \frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2} \quad n = 2, 3, \ldots$$

$$q_0 = 1, \quad q_1 = 0.33332 \quad q_n = \frac{10}{3}q_{n-1} - q_{n-2} \quad n = 2, 3 \ldots,$$

$\{r_n\}$ *kasuan* r_0*-ren errorea 0.00004 da.* $\{p_n\}$ *eta* $\{q_n\}$ *kasuetan,* p_1*-en eta* q_1*-en errorea* $0.00001\overline{3}$ *da. Hastapen-baldintzetako errore txiki horien hedapena aztertuko dugu.*

n	x_n	r_n	p_n	q_n
0	1.0000000000	0.9999600000	1.0000000000	1.0000000000
1	0.3333333333	0.3333200000	0.3333200000	0.3333200000
2	0.1111111111	0.1111066667	0.1110933330	0.1110666667
3	0.0370370370	0.0370355556	0.0370177778	0.0369022222
4	0.0123456790	0.0123451852	0.0123259259	0.0119407407
5	0.0041152263	0.0041150617	0.0040953086	0.0029002469
6	0.0013717421	0.0013716872	0.0013517695	-0.0022732510
7	0.0004572474	0.0004572291	0.0004372565	-0.0104777503
8	0.0001524158	0.0001524097	0.0001324188	-0.0326525834
9	0.0000508053	0.0000508032	0.0000308063	-0.0983641945
10	0.0000169351	0.0000169344	-0.0000030646	-0.2952280648

3.1. taula. $\{x_n\} = \{1/3^n\}$ segida eta bere hurbilpenak.

n	$x_n - r_n$	$x_n - p_n$	$x_n - q_n$
0	0.0000400000	0.0000000000	0.0000000000
1	0.0000133333	0.0000133333	0.0000133333
2	0.0000044444	0.0000177778	0.0000444444
3	0.0000014815	0.0000192593	0.0001348148
4	0.0000004938	0.0000197531	0.0004049383
5	0.0000001646	0.0000199177	0.0012149794
6	0.0000000549	0.0000199726	0.0036449931
7	0.0000000183	0.0000199909	0.0109349977
8	0.0000000061	0.0000199970	0.0328049992
9	0.0000000020	0.0000199990	0.0984149998
10	0.0000000007	0.0000199997	0.2952449999

3.2. taula. Erroreen segidak.

$\{r_n\}$-ren errorea egonkorra da eta era esponentzialean txikitzen da. $\{p_n\}$-ren errorea egonkorra da. $\{q_n\}$-ren errorea ezegonkorra da eta abiadura esponentzialarekin handitzen da. Nahiz eta $\{p_n\}$-ren errorea egonkorra izan, haren gaiek $n \to \infty$ denean $p_n \to 0$ betetzen dutenez, epe luzean errorea menperatu egiten da, eta p_8-tik aurrerako zifra esanguratsuak ez daude ados x_n-ri dagozkienekin. Ikus 3.4.5. eta 3.4.5. taulak.

3.4.6. Datuen ziurgabetasuna

Errealitatean agertzen diren problemen datuek ziurgabetasunak edo erroreak dauzkate. Errore mota hori *zarata* izenarekin ezagutzen da, eta datu horietan oinarritzen den edozein zenbakizko kalkuluren zehaztasunari eragiten dio. Ezin dugu hobetu kalkuluen zehaztasuna, zaratak erasandako datuekin eragiketak gauzatzen baditugu. Beraz, d zifra esanguratsu dauzkaten datuekin hasten bagara, orduan, datu horien bidez lortutako emaitzak d zifra esanguratsuekin erakutsi beharko lirateke. Adibidez, demagun $p_1 = 4.152$ eta $p_2 = 0.07931$ datuek lau zifrako zehaztasuna daukatela; orduan, tentagarria izango litzateke kalkulagailu baten pantailan agertzen diren zifra guztiak ematea, esate baterako, haien batuketa egitean: $p_1 + p_2 = 4.23131$. Baina hori ez da zuzena, ez genituzke emaitzak erabili beharko, horiek jatorrizko datuek baino zifra esanguratsu gehiago baldin badituzte. Hau izango litzateke emaitza egokia egoera horretan: $p_1 + p_2 = 4.231$.

3.5. Problemak

Eskuz ebazteko problemak:

1. Egin 3.3. adibidearen pareko kalkuluak $x_0 = 0.5$ puntuko $f(x) = e^{-2x}$ funtzioaren deribatua hurbiltzeko. Behatu antzekotasunak eta desberdintasunak konparatzean zure taula adibideko taularekin.

2. Egin 3.2. adibidearen antzeko kalkuluak $f'(x_0)$ hurbiltzeko, adierazpen hau erabiliz:

$$\frac{f(x_0 + h) - f(x_0 - h)}{2h}.$$

   Erakutsi errorea $O(h^2)$ ordenakoa dela. Hain zuzen, errorearen gai nagusia (h nahiko txikia denean) $-\frac{h^2}{3!}f'''(x_0)$ dela $f'''(x_0) \neq 0$ bada.

3. Egin 3.3. adibidearen pareko kalkuluak 2. problemaren hurbilpena erabiliz. Behatu antzekotasunak eta desberdintasunak konparatzean zure taula adibideko taularekin.

4. Bihurtu zenbaki bitar hauek zenbaki dezimal:

   (a) 10101_{bi}
   (b) 11111110_{bi}
   (c) 0.11011_{bi}
   (d) 0.1010101_{bi}
   (e) 1.0110101_{bi}
   (f) 11000101.101_{bi}.

5. Bihurtu zenbaki dezimal hauek zenbaki bitar:

   (a) 23
   (b) 378
   (c) 0.6
   (d) 7/16
   (e) 23/32
   (f) 1/7.

6. Idatzi 81, 66.25 eta -0.625 zenbakiak era hauetan:

   (a) Era bitarrean.
   (b) Puntu higikorreko era normalizatuan.
   (c) 32 biteko zehaztasun bakuneko kate baten bidez (IEEE-754 estandarrean).

7. Idatzi 256.1875, -30952 eta -0.0032 zenbakiak era hauetan:

(a) Era bitarrean.

(b) Puntu higikorreko era normalizatuan.

(c) 64 biteko zehaztasun bikoitzeko kate baten bidez (IEEE-754 estandarrean).

8. Aurkitu $R - 0.d_1d_2d_3d_4d_5d_6d_{7bi}$ hurbiltze-errorea, zazpi zifra esanguratsuko hurbilpen bitar hauetan:

   (a) $R = 1/10 \approx 0.0001100_{bi}$.

   (b) $R = 1/7 \approx 0.0010010_{bi}$.

9. Serie geometriko konbergenteen baturaren formula erabiliz, frogatu $1/7{=}0.\overline{001}_{bi}$ garapen bitarra eta $\dfrac{1}{7} = \dfrac{1}{8} + \dfrac{1}{64} + \dfrac{1}{512} + \ldots$ baliokideak direla.

10. Serie geometriko konbergenteen baturaren formula erabiliz, frogatu $1/5{=}0.\overline{0011}_{bi}$ garapen bitarra eta $\dfrac{1}{5} = \dfrac{3}{16} + \dfrac{3}{256} + \dfrac{3}{4096} + \ldots$ baliokideak direla.

11. Zehaztu zer gertatzen den, lau zifrako mantisa duen ordenagailu batek eragiketa hauek egiten dituenean:

    (a) $\left(\dfrac{1}{3} + \dfrac{1}{5}\right) + \dfrac{1}{6}$

    (b) $\left(\dfrac{1}{10} + \dfrac{1}{3}\right) + \dfrac{1}{5}$

    (c) $\left(\dfrac{3}{17} + \dfrac{1}{9}\right) + \dfrac{1}{7}$

    (d) $\left(\dfrac{7}{10} + \dfrac{1}{9}\right) + \dfrac{1}{7}$.

12. (a) Zenbat zenbaki positibo desberdin adieraz daitezke $(\beta, t, L, U) = (10, 2, -9, 10)$ puntu higikorreko sisteman?

    (b) Zenbat zenbaki normalizatu desberdin adieraz daitezke (β, t, L, U) puntu higikorreko sistema orokorrean?

13. (a) $8/7 = 1.14285714285714\ldots$ zenbakiak, noski, ez du adierazpen zehatzik sistema hamartarrean. Aurkitu β oinarriko eta t mantisako (zehaztasuneko) puntu higikorreko sistema bat, non zenbaki horrek adierazpen zehatza daukan.

    (b) π zenbakirako al dago horrelako sistema?

14. Aurkitu E_x errore absolutua eta R_x errore erlatiboa, eta zehaztu hurbilpenaren zifra esanguratsuen kopurua kasu hauetan:

    (a) $x = 2.71828182,\ \widehat{x} = 2.7182$.

    (b) $y = 98350,\ \widehat{y} = 98000$.

    (b) $z = 0.000068,\ \widehat{z} = 0.00006$.

15. Bete ezazu kalkulu hau:

$$p = \int_0^{1/4} e^{x^2} dx \approx \int_0^{1/4} \left(1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!}\right) dx = \widehat{p}.$$

Esan zein errore mota gertatzen den, eta konparatu $\widehat{p}$ emaitza egiazko balioarekin: $p = 0.2553074606$.

16. (a) Jo dezagun $p_1 = 1.414$ eta $p_2 = 0.09125$ datuak lau zifra esanguratsuko zehaztasunarekin emanda daudela. Egoera horretan, aurkitu $p_1 + p_2$ eta $p_1 p_2$ eragiketei dagozkien emaitzak.

(b) Jo dezagun $p_1 = 31.415$ eta $p_2 = 0.027182$ datuak bost zifra esanguratsuko zehaztasunarekin emanda daudela. Egoera horretan, aurkitu $p_1 + p_2$ eta $p_1 p_2$ eragiketei dagozkien emaitzak.

17. Bukatu kalkulu hauek eta esan nolako errorea gertatzen den:

(a) $\dfrac{\sin(\pi/4 + 0.00001) - \sin(\pi/4)}{0.00001} = \dfrac{0.70711385222 - 0.70710678119}{0.00001} = \ldots.$

(b) $\dfrac{\ln(2 + 0.00005) - \ln(2)}{0.00005} = \dfrac{0.69317218025 - 0.69314718056}{0.00005} = \ldots.$

18. Hiru zifrako eta biribiltzeko puntu higikorreko aritmetika erabiliz, kalkula itzazu batura hauek (batu adierazten den ordenan):

(a) $\sum_{k=1}^{6} \frac{1}{3^k}$.

(b) $\sum_{k=1}^{6} \frac{1}{3^{7-k}}$.

19. Izan bitez Taylorren garapen hauek:

$$\frac{1}{1-h} = 1 + h + h^2 + h^3 + O(h^4)$$

eta

$$\cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6).$$

Aurkitu horien baturaren eta biderkaduraren hurbiltze-ordenak.

20. Izan bitez Taylorren garapen hauek:

$$\sin(h) = h - \frac{h^3}{3!} + \frac{h^5}{5!} + +O(h^7)$$

eta

$$\cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6).$$

Aurkitu horien baturaren eta biderkaduraren hurbiltze-ordenak.

21. $f_1(x_0, h) = \cos(x_0 + h) - \cos(x_0)$ funtzioa eralda dezakegu beste $f_2(x_0, h)$ bat lortzeko, formula trigonometriko hau erabiliz:

$$\cos(a) - \cos(b) = -2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right).$$

Beraz, f_1 eta f_2 funtzioek balio berdinak dituzte, aritmetika zehatzean, x_0 eta h aldagaien balio guztietarako.

(a) Aurkitu $f_2(x_0, h)$-ri dagokion adierazpena.

(b) $(f(x_0 + h) - f(x_0))/h$ adierazpenaren bidezko $f'(x_0)$-ren kalkulu hurbilduan, iradoki adierazpen bat ezeztapenaren errorea saihesteko. Idatzi MATLABeko programa bat zure adierazpena inplementatzen duena, eta kalkula itzazu $f'(0.5)$-ren hurbilpenak, $h = 1.e-20, 1.e-19, \ldots, 1$ balioetarako.

(c) Azaldu zure emaitzen eta 3.3. adibideko emaitzen artean dagoen diferentzia.

22. $f_1(x, \delta) = \sin(x + \delta) - \sin(x)$ funtzioa eralda dezakegu beste $f_2(x, \delta)$ bat lortzeko, formula trigonometriko hau erabiliz:

$$\sin(a) - \sin(b) = 2 \cos\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right).$$

Beraz, f_1 eta f_2 funtzioek balio berdinak dituzte, aritmetika zehatzean, x eta δ aldagaien balio guztietarako.

(a) Aurkitu $f_2(x, \delta)$-ri dagokion adierazpena.

(b) Froga ezazu, analitikoki, $f_1(x, \delta)/\delta$ eta $f_2(x, \delta)/\delta$ adierazpenak, δ nahiko txikia denean, $\cos(x)$ funtzioaren hurbilpenak direla.

(c) Idatzi MATLABeko funtzio bat $g_1(x, \delta) = f_1(x, \delta)/\delta - \cos(x)$ eta $g_2(x, \delta) = f_2(x, \delta)/\delta - \cos(x)$ kalkulatzeko, $x = 3$ eta $\delta = 1.e-11$ hartuz.

(d) Azaldu bi kalkuluen arteko emaitzen diferentziak.

23. Jo dezagun lehenengo deribatuaren hurbilpena, hots:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

Formula honetarako trunkatze (edo diskretizazio) errorea $O(h)$ da. Demagun f balioztatzean errore absolutua ε-k bornatzen duela, eta baztertu egingo ditugu oinarrizko eragiketa aritmetikoak egitean sortutako erroreak.

(a) Froga ezazu konputazio-errore osoa (trunkatzearen eta biribiltzearen konbinazioa) bornatuta dagoela balio honetaz:

$$\frac{Mh}{2} + \frac{2\varepsilon}{h},$$

non M $|f''(x)|$-ren borne bat baita.

(b) Zein h-ren baliotarako minimizatzen da aurreko bornea?

(c) Erabiltzen dugun biribiltze-errorea, gutxi gorabehera, 10^{-16} da. Erabili aurreko galderaren emaitza, 3.3. adibideko grafikoaren portaera azaltzeko. Grafikoaren itxura azaltzean, azaldu, baita ere, non espero dezakegun minimo hori aurkitzea.

24. Demagun $f(x) = \dfrac{1-\cos(x)}{\sin(x)}$ funtzioa.

(a) Erabili formatu hamartarra, sei digitu esanguratsurekin (biribilduz), $f(0.007)$ kalkulatzeko (kalkulagailua erabiliz).

(b) Erabili MATLAB (`format long` erabiliz) $f(x)$-ren balioa kalkulatzeko, eta benetako errore erlatiboa, biribiltzearen ondorioz, (a) atalean lortutako $f(x)$-ren balioarekiko.

(c) Biderkatu $f(x)$ funtzioa $\dfrac{1+\cos(x)}{1+\cos(x)}$ adierazpenaz, biribiltze-errorea izateko joera gutxiago duen $f(x)$-ren era bat lortzeko. Era berrian, erabili sei digitu esanguratsutako formatu hamartarra (biribilduz) $f(0.007)$ kalkulatzeko (kalkulagailua erabiliz). Konparatu balioa, (a) eta (b) ataletan lorturiko balioekin.

25. Demagun $f(x) = \dfrac{\sqrt{9+x}-3}{x}$ funtzioa.

(a) Erabili formatu hamartarra sei digitu esanguratsurekin (biribilduz) $f(0.005)$ kalkulatzeko (kalkulagailua erabiliz).

(b) Erabili MATLAB (`format long` erabiliz) $f(x)$-ren balioa kalkulatzeko, eta benetako errore erlatiboa, biribiltzearen ondorioz, (a) atalean lortutako $f(x)$-ren balioarekiko.

(c) Biderkatu $f(x)$ funtzioa $\dfrac{\sqrt{9+x}+3}{\sqrt{9+x}+3}$ adierazpenaz, biribiltze-errorea izateko joera gutxiago duen $f(x)$-ren era bat lortzeko. Era berrian, erabili sei digitu esanguratsutako formatu hamartarra (biribilduz) $f(0.005)$ kalkulatzeko (kalkulagailua erabiliz). Konparatu balioa (a) eta (b) ataletan lorturiko balioekin.

26. Honako kasu hauetan, aurkitu formula baliokide bat zifra esanguratsuen galera saihesteko:

(a) $\ln(x+1) - \ln(x)$, x handi baterako.

(b) $\sqrt{x^2+1} - x$, x handi baterako.

(c) $\cos^2(x) - \sin^2(x)$, $x \approx \pi/4$ baterako.

(d) $\sqrt{\dfrac{1+\cos(x)}{2}}$, $x \approx \pi$ baterako.

Kasu bakoitzean, azaldu zergatik aukeratu duzun formula hori, eta eman zenbakizko adibide bat zehaztasunaren diferentzia agerian uzteko.

27. (a) Froga ezazu $\ln(x - \sqrt{x^2-1}) = -\ln(x + \sqrt{x^2-1})$ betetzen dela $|x| \geq 1$ bada.
    (b) Bi formula horietako zein da egokiena zenbakizko kalkulurako? Azaldu zergatik, eta eman zenbakizko adibide bat zehaztasunaren diferentzia agerian jartzeko.

28. Azaldu adierazpen hauetarako ager daitezkeen zailtasunak, eta berridatzi formula horiek zenbakizko kalkulurako era egokiago batean:

    (a) $\sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}}$, $x \gg 1$ denean.

    (b) $\sqrt{\frac{1}{a^2} + \frac{1}{b^2}}$, $a \approx 0$ eta $b \approx 1$ direnean.

    Kasu bakoitzean, eman zenbakizko adibide bat zehaztasunaren diferentzia nabaria izateko.

29. Izan bitez adierazpen hauek:

$$p(x) = x^3 - 3x^2 + 3x - 1, \quad q(x) = ((x-3)x + 3)x - 1, \quad r(x) = (x-1)^3.$$

    (a) Lau zifrako puntu higikorreko aritmetika erabiliz eta biribilduz, kalkula itzazu $p(2.72)$, $q(2.72)$ eta $r(2.72)$. Kontuan izan $p(x)$-ren kalkuluan, aritmetika horrekin, $(2.72)^3 = 20.12$ eta $(2.72)^2 = 7.398$ direla.

    (b) Lau zifrako puntu higikorreko aritmetika erabiliz eta biribilduz, kalkula itzazu $p(0.975)$, $q(0.975)$ eta $r(0.975)$. Kontuan izan $p(x)$-ren kalkuluan, aritmetika horrekin, $(0.975)^3 = 0.9268$ eta $(0.975)^2 = 0.9506$ direla.

30. Demagun $(\beta, t, L, U) = (10, 8, -50, 50)$ puntu higikorreko sistema duen makina bat erabiltzen dugula ekuazio koadratiko honen erroak kalkulatzeko:

$$ax^2 + bx + c = 0,$$

    non a, b eta c zenbaki erreal ezagunak baitira.

    Ondorengo kasu bakoitzerako, esan zelako zenbakizko zailtasunak gerta daitezkeen formula estandarra erabiltzen badugu erroak kalkulatzeko. Azaldu, baita ere, nola gainditu zailtasun horiek (ahal denean):

    (a) $a = 1$; $\quad b = -10^5$; $\quad c = 1$.

    (b) $a = 6 \cdot 10^{30}$; $\quad b = 5 \cdot 10^{30}$; $\quad c = -4 \cdot 10^{30}$.

    (c) $a = 10^{-30}$; $\quad b = -10^{30}$; $\quad c = 10^{30}$.

31. *Bigarren mailako ekuazioaren ebazpenaren formula hobetua.* Demagun $a \neq 0$ eta $b^2 - 4ac > 0$ ditugula, eta jo dezagun $ax^2 + bx + c = 0$ ekuazioa. Haren erroak adierazpen ezagun hauen bidez kalkulatzen dira:

    (a) $x_1 = \dfrac{-b + \sqrt{b^2 - 4ac}}{2a}$ eta $x_2 = \dfrac{-b - \sqrt{b^2 - 4ac}}{2a}$.

    Frogatu erro horiek kalkula daitezkeela adierazpen baliokide hauen bidez:

(b) $x_1 = \dfrac{-2c}{b + \sqrt{b^2 - 4ac}}$ eta $x_2 = \dfrac{-2c}{b - \sqrt{b^2 - 4ac}}$.

Oharra. $|b| \approx \sqrt{b^2 - 4ac}$ denean, kontuz ibili behar dugu ezeztatzearen bidezko zehaztasunaren galera saihesteko. Baldin $b > 0$ bada, x_1 erroa (b) formularen bidez kalkulatu beharko genuke, eta x_2 erroa (a) formulaz. Aldiz, $b < 0$ bada, x_1 erroa (a) formularen bidez kalkulatu beharko genuke, eta x_2 erroa (b) formula erabiliz.

32. Demagun gure zenbaki-sistema hamartarrak 8 digitu esanguratsu erabiltzen dituela (hots, $t = 8$). Erabili formula egokia x_1 eta x_2 kalkulatzeko, aurreko ariketan azaltzen den bezala, bigarren mailako ekuazio hauen erroak aurkitzeko:

    (a) $x^2 - 1000.001x + 1 = 0$.

    (b) $x^2 - 100000.0001x + 1 = 0$.

    (c) $x^2 + 100000.00001x + 1 = 0$.

33. Aztertu eragiketa hauen erroreen hedapena (begira ezazu 3.4.5. azpiatala):

    (a) Hiru zenbakiren batura:

    $$p + q + r = (\widehat{p} + \varepsilon_p) + (\widehat{q} + \varepsilon_q) + (\widehat{r} + \varepsilon_r).$$

    (b) Zero ez den zenbaki baten alderantzizkoa:

    $$\frac{1}{q} = \frac{1}{\widehat{q} + \varepsilon_q}.$$

    (c) Zero ez den zenbaki batez zatitzea:

    $$\frac{p}{q} = \frac{\widehat{p} + \varepsilon_p}{\widehat{q} + \varepsilon_q}.$$

MATLABez ebazteko problemak:

34. (a) Marraztu $(\beta, t, L, U) = (2, 3, -2, 3)$ puntu higikorraren sistemako zenbaki guztien adierazpen hamartarrak. $d_0.d_1d_2$ mantisako zenbaki txikiena 1.00 da eta zenbaki posible handiena 1.11 (sistema hamartarrean 1.75). Hortaz, begizta batean 1etik 1.75era joango gara, eta gehikuntzak $\beta^{1-t} = 2^{-2} = 0.25$ izango dira. Hori da begizta baterako oinarria. Bestalde, berretzailea -2tik 3ra doa; hori da aurreko begiztaren barneko begizta baten tartea. Begizta bikoitz horrek eraikitzen du bektore bat sistema horretako zenbaki positibo guztiekin. Baina, zenbaki negatiboak eta zeroa ere marraztu behar dituzu. Marrazteko orduan, $\boldsymbol{x}$ bektorean zenbaki hamartar horiek guztiak gorde behar dituzu, eta $y_i = 0$ hartu (x_i, y_i) bikote guztietarako.

    (b) Zer gertatzen da sistemako zenbakien arteko distantziarekin? Zergatik?

35. Formula koadratiko klasikoak dio $ax^2 + bx + c = 0$ ekuazioaren bi erroak honela kalkulatzen direla:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{eta} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

(a) Sortu MATLABeko M fitxategi bat (`[x1,x2]=fkklas(a,b,c)` motakoa) erroak kalkulatzeko, eta erabili funtzio hori bi erroak kalkulatzeko kasu honetan:

$$a = 1, \quad b = -100000000, \quad c = 1.$$

(b) Konpara itzazu emaitza horiek MATLABeko funtzio honek ematen duenarekin:

```
roots([a b c])
```

(c) Zer gertatzen da erroak eskuz edo kalkulagailu arrunt batez kalkulatzen badituzu? Ikusi beharko duzu formula klasikoa ona dela erro bat kalkulatzeko, baina ez bestea. Beraz, erabili ezazu zuk sortutako funtzio hura zehaztasunez erro bat aurkitzeko, eta gero erabili propietate hau:

$$x_1 x_2 = \frac{c}{a}$$

beste erroa kalkulatzeko.

36. *Formula hobetua.* Erabili aurreko 31. problemaren informazioa ariketa hauek egiteko.

(a) Idatzi algoritmo bat erroak era horretan kalkula ditzan eta $|b| \approx \sqrt{b^2 - 4ac}$ denean, kontuz ibil dadila ezeztatzearen bidezko zehaztasunaren galera saihesteko. Alegia, zera kontuan hartuz:

- Baldin $b > 0$ bada, x_1 erroa azken formularen bidez kalkulatu beharko genuke, eta x_2 erroa formula klasikoaz.
- Aldiz, $b < 0$ bada, x_1 erroa formula klasikoaren bidez kalkulatu beharko genuke, eta x_2 erroa azken formula erabiliz.

(b) Gero, inplementatu algoritmo hori MATLABeko M fitxategi bat eraikiz (adibidez, `[x1,x2]=fkhobe(a,b,c)` motakoa) eta probatu ekuazio koadratiko egoki batzuekin (esate baterako, $x^2 - 5x + 6 = 0$ edo $x^2 - 100000000x + 1 = 0$).

37. Formula klasikoa eta formula hobetua erabiliz, kalkula itzazu bigarren mailako ekuazio hauen erroak:

(a) $x^2 - 1000.001x + 1 = 0$.

(b) $x^2 - 100000.0001x + 1 = 0$.

(c) $x^2 - 100000.00001x + 1 = 0$.

(d) $x^2 - 1000000.000001x + 1 = 0$.

(e) $x^2 - 100000000.00000001x + 1 = 0$.

Egiaztatu emaitzak ($x_1 + x_2 = -b/a$ eta $x_1 x_2 = c/a$ bete behar dute), eta konpara itzazu kasu bakoitzeko formula klasikoak eta formula hobetuak lorturiko emaitzak. Zein da metodo egokiena? Zergatik huts egiten du besteak?

38. Hau da $f(x) = e^x$ funtzioaren Taylorren seriearen garapena:

$$f(x) = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots .$$

Erabili ekuazio hori e^{-2} kalkulatzeko, kasu hauetan:

(a) Lehenengo lau batugaiak erabiliz.

(b) Lehenengo sei batugaiak erabiliz.

(c) Lehenengo zortzi batugaiak erabiliz.

Kasu bakoitzean, kalkula ezazu trunkatze-errore absolutua eta errore erlatiboa. Erabili MATLAB, `format long` instrukzioaz, e^{-2}-ren benetako balioa kalkulatzeko. Erabili sei zenbaki esanguratsuko zenbaki hamartarrak (biribilduz), eragiketa guztietan.

39. Hau da $\sin(x)$-rako berretura-seriea:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots .$$

MATLABeko funtzio honek seriea erabiltzen du $\sin(x)$ kalkulatzeko:

```
function s = berresin(x)
% sin(x) berretura seriearen bidez kalkulatzen du.
s = 0;
t = x;
n = 1;
while s+t ~= s;
s = s+t;
t = -x.^2/((n+1)*(n+2)).*t
n = n+2;
end
```

(a) Zergatik bukatuko da `while` begizta?

(b) Aldatu `berresin.m`-ren lehenengo lerroa honela:
`function [s,tmax,tkop]=berresin(x)`
Gero, sartu lerro hauek `while` begizta hasi baino lehen:

```
tmax = abs(t);
tkop = 0;
```

eta lerro hauek begiztaren bukaeran:

```
tmax = max(tmax,abs(t));
tkop = tkop+1;
```

Erantzun galdera hauek $x = \pi/2,\ 11\pi/2,\ 21\pi/2$ eta $31\pi/2$ balioetarako:

- Zein da emaitzaren zehaztasuna?
- Zenbat gai erabili behar izan ditu?
- Zein izan da seriearen tamaina handieneko gaia?

Galdera horiek erantzuteko, komeni da honelako taula bat egitea:

x	$\pi/2$	$11\pi/2$	$21\pi/2$	$31\pi/2$
sin(x)-berresin(x)				
tmax				
tkop				

(c) Zer ondorioztatzen da puntu higikorreko aritmetikaren eta berretura-serieen erabileraz funtzioak balioztatzeko orduan?

40. 3.22. eta 3.23. adibideak kontuan hartuz, frogatu $\{1/2^n\}_{n=1}^{\infty}$ segidaren gaiak garatzeko hiru metodo hauek erabil ditzakegula:

(a) $r_0 = 1$ eta $r_n = \frac{1}{2}r_{n-1}$, $n = 2, 3, \ldots$ izanik.

(b) $p_0 = 1$, $p_1 = 0.5$ eta $p_n = \frac{3}{2}p_{n-1} - p_{n-2}$, $n = 2, 3, \ldots$ izanik.

(c) $q_0 = 1$, $q_1 = 0.5$ eta $q_n = \frac{5}{2}q_{n-1} - q_{n-2}$, $n = 2, 3, \ldots$ izanik.

Orain, kasu bakoitzean hasierako errore txiki bat sortzen da:

(a) $r_0 = 0.994$.

(b) $p_0 = 1$, $p_1 = 0.497$.

(c) $q_0 = 1$, $q_1 = 0.497$.

MATLAB erabiliz, sortu metodo bakoitzerako lehenengo 10 iterazioak, eta aurkeztu emaitzak 3.4.5. eta 3.4.5. tauletan bezala (MATLABen bidez, hori ere).

Nolako errorea (egonkorra/ezegonkorra) gertatzen da kasu bakoitzean? Non txikitzen da errore hori?

4. kapitulua

Ekuazio ez-linealen ebazpena

Medikuntza-ikasketek baieztatu dute puenting-jauzilari batek bizkarrezurreko kalte garrantzitsu bat jasateko duen probabilitatea handitu egiten dela baldin erorketa libreko abiadura 36 m/s baino handiagoa bada, erorketa librean 4 s egon ondoren. Orduan, zure nagusiak nahi du puenting-enpresan zuk zehaztea zein masatarako gainditzen den irizpide hori, airearen erresistentzia-koefizientea 0.25 kg/m bada.

Zuk badakizu, aurreko ikasketen bidez, adierazpen hau erabil dezakegula denboraren menpeko erorketaren abiadura jakiteko:

$$v(t) = \sqrt{\frac{g \cdot m}{c_d}} \tanh\left(\sqrt{\frac{g \cdot c_d}{m}} t\right).$$

Ezin da bakandu m. Aukera bat da m aurkitzeko funtzio berri honen *erroa* kalkulatzea:

$$f(m) = \sqrt{\frac{g \cdot m}{c_d}} \tanh\left(\sqrt{\frac{g \cdot c_d}{m}} t\right) - v(t). \tag{4.1}$$

Kapitulu honetan ikusiko dugu nola erabil dezakegun ordenagailua horrelako soluzioak kalkulatzeko.

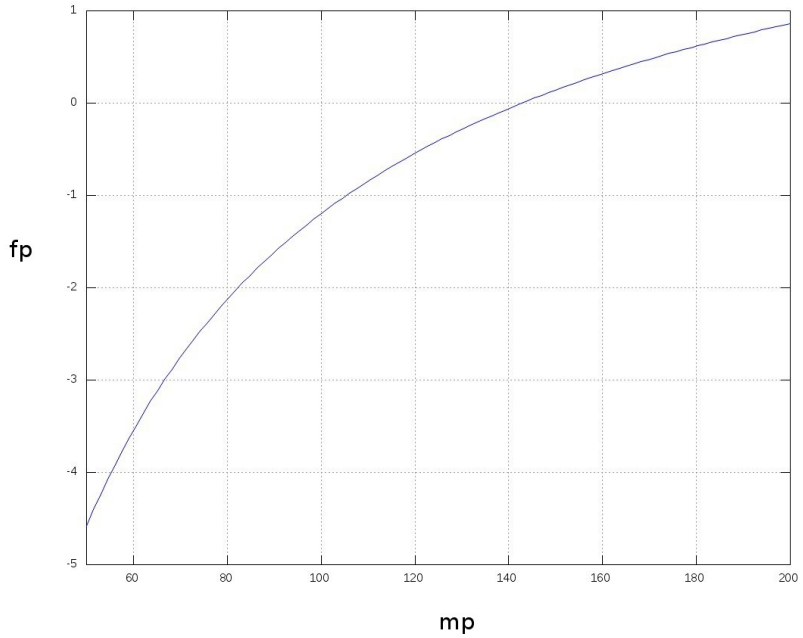
4.1. Metodo grafikoak

Metodo erraz bat da $f(x) = 0$ ekuazioko erroaren hurbilpen bat lortzeko funtzioaren marrazketa egitea, eta aztertzea non zeharkatzen duen x ardatza. Puntu horrek erroaren sasihurbilpen bat ematen digu.

4.1. adibidea. *Hurbilpen grafikoa erabiliz, aurkitu puenting-jauzilariaren masa, $c_d = 0.25$ kg/m bada, erorketa libreko 4 s igaro ondoren abiadura 36 m/s-koa izateko ($g = 9.81$ m/s^2).*

Ebazpena: m-rekiko (4.1) funtzioa marrazteko, MATLAB erabiliz, hau egingo dugu:

```
>> cd=0.25; g=9.81; v=36; t=4;
>> mp=linspace(50,200);
>> fp=sqrt(g*mp/cd).*tanh(sqrt(g*cd./mp)*t)-v;
>> plot(mp,fp), grid
```



4.1. irudia. (mp, fp) grafikoa.

Funtzio horrek 140 eta 150 kg artean zeharkatzen du m ardatza. Marrazketaren azterketak erroaren 145 kg-ko sasihurbilpen bat ematen digu. Hurbilpen grafiko horren baliotasuna probatzeko, (4.1) adierazpenean ordezkatuko dugu:

```
>> sqrt(g*145/cd).*tanh(sqrt(g*cd./145)*t)-v
ans =
    0.0456
```

zeina zerotik hurbil baitago. □

Teknika grafikoek erabilera praktiko mugatua dute, oso zehatzak ez direlako. Hala ere, metodo grafikoak erabil daitezke erroen sasihurbilpenak aurkitzeko, eta horiek zenbakizko metodoetan erabil daitezke, hasierako hurbiltze-puntuak bezala.

4.2. Bakartze-metodoak

Erroari buruzko informazioa lortzeko metodo nagusiak hauek dira:

- *Bakartze-metodoak.* Metodo hauetan, hasierako bi hurbiltze-puntuk erro bakoitza bakartzen du tarte batean.
- *Metodo irekiak.* Metodo hauetan, hasierako hurbiltze-puntu bat (edo gehiago) erabiltzen da, baina ez da beharrezkoa erroa tarte batean bakartzea.

Bakartze-metodoen konbergentzia motela da. Aldiz, metodo irekien konbergentzia, oro har, azkarragoa da; baina haiek dibergenteak izan daitezke.

Bi kasuetan, beharrezkoak dira hasierako hurbiltze-puntuak. Aztertzen ari garen testuinguru fisikotik atera ditzakegu horiek. Hala ere, batzuetan ez da erraza horrelako hastapendatu onak iragartzea.

Ekuazio batek erro anitz baldin badauzka tarte batean, erroak bakartzeko tarte txikiagoak definituko ditugu. Hori funtzioa marraztuz egin dezakegu, edo funtzioa balioztatuz segidako puntuen multzo bat hartuz, eta aztertuz funtzioaren zeinua zein tartetan aldatzen den.

Ondorengo atalean kasu berezi bat ikusiko dugu, non erroak analitikoki bakartu baititzakegu.

4.2.1. Erroak bakartzea

Izan bitez $[a, b]$ tartea eta $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ funtzio ondo definitua eta jarraitua tarte horretan. Funtzioaren erroak bilatzea eta $f(x) = 0$ ekuazioaren soluzioak asmatzea problema bera da. Berdintza hori betetzen duen p soluzioari *funtzioaren erro* esango diogu.

Batzuetan, $f'(x)$ funtzioaren existentzia funtsezkoa izango da helburua lortzeko eta, zenbaitetan, $f''(x)$-rena ere bai. Jo dezagun $f(x) = 0$ ekuazioaren erro guztiak desberdinak direla. Hortaz, erro bakoitzaren ingurune nahiko txiki baterako ez dago beste errorik. Orduan, bi urrats hauek beteko ditugu:

1. Erroen bereizketa: erro bakoitzaren $[\alpha, \beta]$ ingurune ahalik eta txikiena aurkitzea; non tartearen barnean ez baita ekuazioaren beste errorik egon behar.
2. Erro hurbilduen balioak doitzea: emaitza gero eta hobeak hurbiltzea.

Jarraian, kontuan hartu beharko ditugu bi propietate hauek:

(i) *Izan bedi f jarraitua $[a,b]$ tartean eta $[a,b]$-ko muturretan zeinu desberdinekin (hots, $f(a)\cdot f(b)<0$); orduan, f funtzioak gutxienez c erro bat izango du (a,b) tartean; hots, $\exists c\in(a,b)$, non $f(c)=0$ baita (Bolzano-ren teorema).*

(ii) *Baldin f deribagarria bada (a,b) tartean, eta f'-k tarte horretan zeinua aldatzen ez badu (hau da, f monotonoa bada), orduan, f funtzioak ez du $[a,b]$ tartean erro bat baino gehiago izango.*

Ondorioz, $f(x)=0$ ekuazioaren erroak kalkulatzeko urrats hauei jarraituko diegu:

1. Funtzioen erroak tarte desberdinetan bakartuko ditugu (ii) aplikatuz, ahal bada. Lehendabizi, emandako tartearen partizio bat egiten da: $a<\alpha_1<\alpha_2<\ldots<\alpha_n<b$ (adibidez, f deribagarria bada, deribatuaren zeinu-aldaketak erabiliz lor dezakegu partizioa; hau da, $f'(x)$ funtzioaren erroak kalkulatuz).

2. Ondoz ondoko puntu-bikoteak bilatuko ditugu, (i) aplikatuz: hala, f funtzioak zeinu desberdinak hartuko ditu. Hau da, α_k, α_{k+1} izango ditugu, non $f(\alpha_k)\cdot f(\alpha_{k+1})<0$ baita. Horrek adierazten digu ezen tarte horretan $f(x)$-ren erro bat existitzen dela.

4.2. adibidea. *Ebatzi $x^4-4x-1=0$ ekuazioa.*

Ebazpena. Izan bedi $f(x)=x^4-4x-1$. Funtzio horrek, 4. mailako polinomio bat denez, gehienez lau erro izango ditu. Funtzio horren zeinu-aldaketak finkatzeko, aski dira emandako tartearen muturrak eta funtzioaren maximo eta minimo lokalak. Kasu horretan, $f'(x)=4(x^3-1)$ aztertzen da, eta, ondorioz, $x=1$ da $f'(x)$-ren zeinu-aldaketa bakarra. Hain zuzen ere:

$$x\in(-\infty,1)\Rightarrow f'(x)<0$$
$$x\in(1,\infty)\Rightarrow f'(x)>0.$$

Hortaz, $(-\infty,1)$ tartean, $f(x)$ beherakorra da, eta, $(1,\infty)$ tartean, $f(x)$ gorakorra. Gainera, tarte horien muturretan zera gertatzen da:

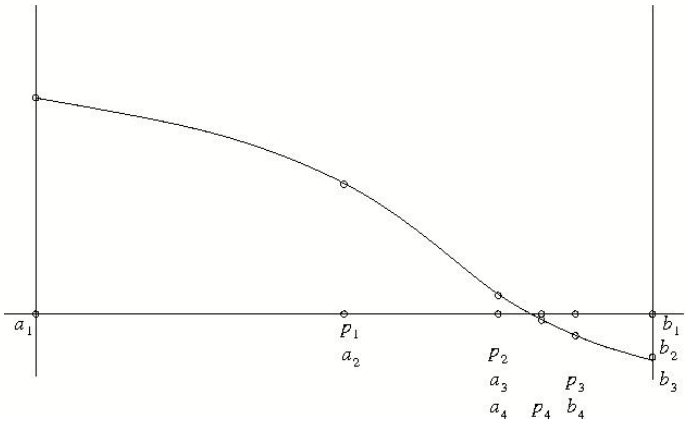
$$f(-\infty)\cdot f(1)<0 \quad \text{eta} \quad f(1)\cdot f(\infty)<0,$$

Ondorioz, bi tarte horietan $f(x)$-k 0 balioa hartzen du (i) aplikatuz. Hau da, tarte bakoitzean erro bat dago. □

Erroak tarte desberdinetan bakartzeko beste metodo bat da funtzioaren grafikoa aztertzea (esate baterako, MATLAB erabiliz). Geroago, erro bat duen tarte bakoitza txikitzen da, adibidez, bisekzio-metodoaren bitartez.

4.2.2. Bisekzio-metodoa

Demagun f funtzioa jarraitua dela $[a,b]$ tartean, eta $f(a)\cdot f(b)<0$ betetzen dela; orduan, (i) propietatearen arabera p balio bat existitzen da (a,b) tartean, non $f(p)=0$ baita. Suposa dezagun, orobat, tarte horretan erro hori baino ez dagoela.



4.2. irudia. Bisekzio-metodoa.

Lehendabizi, $a_1=a$ eta $b_1=b$ izendatuko dira, eta emandako tartea bi zati berdinetan bereiziko da; $[a,b]=[a_1,p_1]\cup[p_1,b_1]$, non $p_1=(a_1+b_1)/2$ erdiguneko balioa baita. Orain, hasierako tartearen luzera-erdiko azpitarte batean egongo da p erroa, eta, zein den jakiteko, $f(a_1)\cdot f(p_1)$ balioztatzen da. Baldin azken balio hori negatiboa bada, orduan lehenengo azpitartean egongo da, eta, ondorioz, prozesua $[a_2,b_2]$ tartean errepikatu behar da, tarte horretarako $a_2=a_1$ eta $b_2=p_1$ hartuz; bestela, $a_2=p_1$ eta $b_2=b_1$ izendatu beharko dira. Iterazio bakoitzean dagoen tartearen luzera erdibitu egiten da, eta, n iterazio gauzatu ondoren, $[a_n,b_n]$-ren luzera $(b-a)/2^{n-1}$ izango da. Noski, $p\in[a_n,b_n]$ dagoenez, errorea $|p_n-p|\le(b-a)/2^n$ izango da. Beraz, n handituz, errorea gero eta txikiagoa egiten da. Hori formaliza daiteke teorema honetan:

4.1. teorema. *Demagun $f\in C[a,b]$ eta $f(a)f(b)<0$ dela. Izan bedi $\{p_n\}_{n=1}^{\infty}$ segida, non $p_n=(a_n+b_n)/2$ baita. Orduan, badago $p\in(a,b)$ puntu bat, non $f(p)=0$, hau betetzen duena:*

$$\varepsilon_n=|p-p_n|\le\frac{b-a}{2^n},\quad n=1,2,\dots \tag{4.2}$$

eta, bereziki, $\lim_{n\to\infty}p_n=p$.

Frogapena. Metodoaren deskripziotik (4.2) lortzen da. Bestalde, hau betetzen da:

$$\lim_{n\to\infty}|p-p_n|\le\lim_{n\to\infty}\frac{b-a}{2^n}=0,$$

eta horrek $\lim_{n\to\infty} p_n = p$ inplikatzen du. □

Aurreko teoremaren arabera, hasierako $\tau > 0$ tolerantzia bat (errorearen goi-borne bat) jar dezakegu, eta $\varepsilon_n < \tau$ denean bukatutzat eman prozesua. Bestalde, prozesua mozteko beti jar dezakegu iterazioen n_{max} kopuru maximo bat; hots, $n = n_{max}$ denean, prozesua bukatzen da eta p_n-rekin geratzen gara.

4.3. adibidea. *Bisekzio-algoritmoa erabiliz, kalkulatu* $e^x = \sin x$ *ekuazioaren soluzioa* $x \in [-4, -3]$ *tartean, errorearen tolerantzia 0.01 izanik.*

Ebazpena. Izan bedi $f(x) = e^x - \sin x$; orain, ekuazioaren soluzio bat f funtzioaren erro bat izango da. Hasteko, $[a_1, b_1] = [-4, -3]$, $f(a_1) = f(-4) < 0$ eta $f(b_1) = f(-3) > 0$ direnez, $[-4, -3]$ tartean gutxienez erro bat dago.

Jarraian, $[-4, -3]$ tartea erdibituko dugu. Erdigunea $p_1 = -3.5$ da; $[-4, -3] = [-4, -3.5] \cup [-3.5, -3]$, $f(p_1) = f(-3.5) < 0$ denez, $[a_2, b_2] = [-3.5, -3]$ tartean egongo da erroa.

Orain, tartearen erdigunea $p_2 = -3.25$ da; $[-3.5, -3] = [-3.5, -3.25] \cup [-3.25, -3]$, $f(p_2) = f(-3.25) < 0$ denez, $[a_3, b_3] = [-3.25, -3]$ tartean egongo da erroa.

Eta horrela jarraituz, nahi dugun bezainbeste hurbilduko gara errorantz.

n	a_n	b_n	p_n	$\lvert f(p_n)\rvert$
1	-4	-3	-3.5	0.3206
2	-3.5	-3	-3.25	0.0694
3	-3.25	-3	-3.125	0.0605
4	-3.25	-3.125	-3.1875	0.0046

4.1. taula. Bisekzio-metodoa.

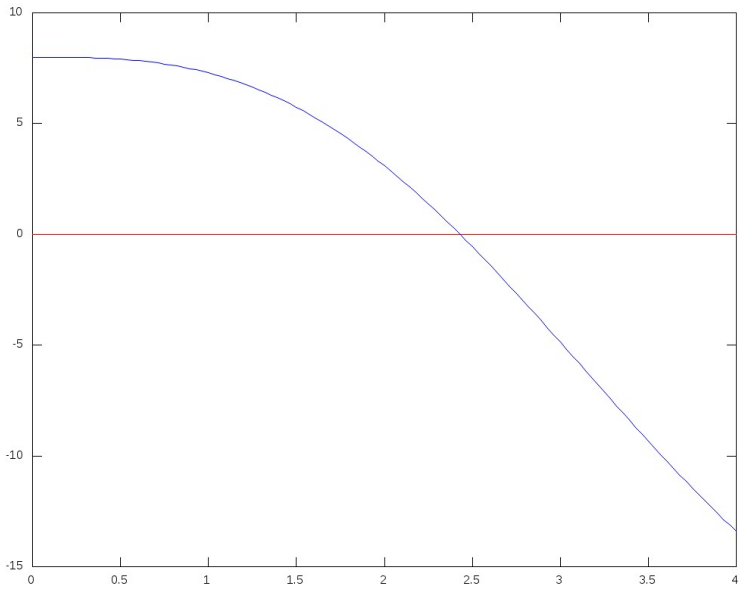
Beraz, lau iterazio egin ondoren lorturiko soluzio hurbildua $x_4 = -3.1875$ da, eta $|f(x_4)| = 0.0046 < 0.01$ sortzen den errorea. □

Bisekzio-metodoari buruzko oharrak

Alde batetik, bisekzio-metodoak bi eragozpen nabari ditu, eta hauexek dira: bata, metodoa astiro hurbiltzen dela soluziorantz, eta bestea, hurbildu bitartean alde batean utz ditzakeela hurbilketa onak. Beste aldetik, metodo honek soluziorantz jotzen du, eta, horregatik, erabilgarria da hasierako estimazio bat asmatzeko, eta, jarraian, beste metodo eraginkorrago bat aplikatu ahal izateko. Metodo horrek huts egin dezake p puntuan, baldin abszisa $y = f(x)$ kurbaren zuzen ukitzailea bada eta kurbak abszisa ez badu zeharkatzen.

4.4. adibidea. *Idatzi MATLAB programa bat, instrukzioen M fitxategi bat,* $8 - 4.5(x - \sin(x)) = 0$ *ekuazioaren soluzioa aurkitzeko, bisekzio-metodoa erabiliz. Soluzioaren errore-tolerantziak 0.001 rad izan behar du (hots, errorea* < 0.001*). Sortu taula bat, non bisekzio-prozesuko iterazio bakoitzean* a*,* b*,* p_n*,* $f(p_n)$ *eta errorea agertzen baitira.*

Ebazpena. Soluzioaren hasierako hurbiltze bat egiteko, $f(x) = 8 - 4.5(x - \sin(x))$ marrazten dugu, MATLABen `fplot` instrukzioa erabiliz. Horrek erakusten digu $x = 2$ren eta $x = 3$ren artean soluzioa dagoela. Beraz, $a = 2$ eta $b = 3$ muturrekin aukeratuko da hasierako tartea.



4.3. irudia. $f(x) = 8 - 4.5(x - \sin(x))$-ren grafikoa.

MATLAB programa honek ebazten du problema hori:

```
clear all
f=inline('8-4.5*(x-sin(x))');
a=2; b=3; nmax=20; tol=0.001;
fa=f(a); fb=f(b);

if fa*fb>0
  disp('Errorea: Funtzioak a eta b puntuetan zeinu berbera du.')
else
  disp('n       a            b          p_n         f(p_n)     errorea')
  disp('-------------------------------------------------------------------')
```

```
  for n=1:nmax
    p_n=(a+b)/2;
    err=(b-a)/2;
    fp_n=f(p_n);
    fprintf('%3i   %11.6f %11.6f %11.6f %11.6f %11.6f\n',n,a,b,p_n,fp_n,err)
    if fp_n==0
      fprintf('p=%11.6f soluzio zehatz bat aurkitu dugu',p_n)
      break
    end
    if err<tol
      fprintf('p=%11.6f soluzio hurbildu bat aurkitu dugu',p_n)
      break
    end
    if i==nmax
      fprintf('Ez dugu lortu soluzioa %i iterazioetan',nmax)
      break
    end
    if fa*fp_n<0
      b=p_n;
    else
      a=p_n;
    end
  end
end
```

4.2.3. *Regula falsi* metodoa

Metodo honen beste izen batzuk dira posizio faltsuaren eta interpolazio linealaren metodoak. Bisekzio-metodoan bezala, $[a,b]$ tartean f jarraitua da, $f(a)\cdot f(b)<0$ eta, beraz, $f(x)=0$-k erro bat dauka tarte horretan. Bisekzio-metodoan, hurrengo urratsa egiteko, $[a,b]$ tartearen erdigunea erabiltzen da. Izan bedi $(a,f(a))$, $(b,f(b))$ puntuetatik igarotzen den $y=f(x)$-ren E zuzen ebakitzailea. Oraingo metodoan hurbilpena hobetzen da, E zuzen horretako $(\overline{p},0)$ puntua erabiliz; hots, E-ren ebakitze-puntua abszisarekin. Orain $\overline{p}$ aurkituko dugu. E-ren m malda kalkulatzeko, bi adierazpen hauek ditugu:

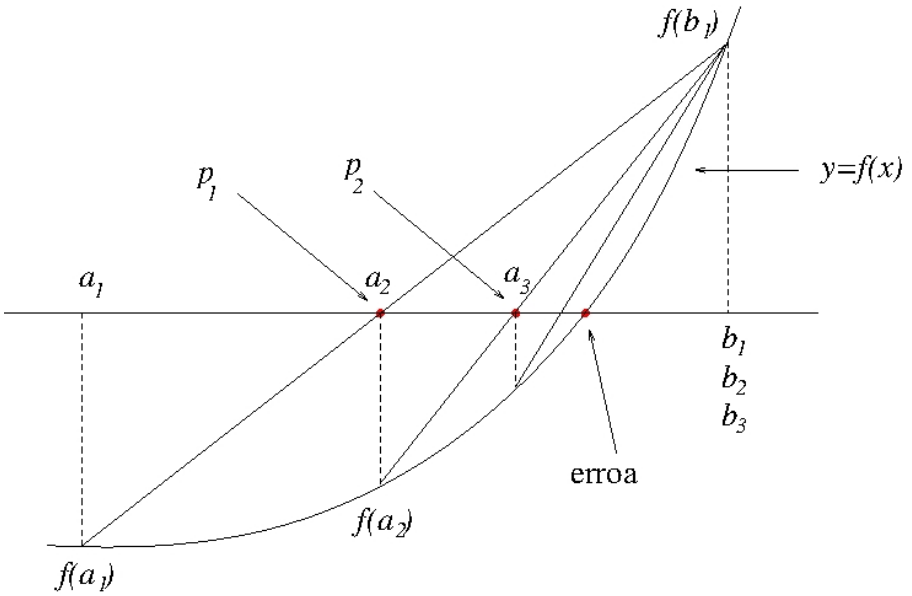
$$m=\frac{f(b)-f(a)}{b-a} \quad \text{eta} \quad m=\frac{0-f(b)}{\overline{p}-b},$$

biak berdinduz,

$$\frac{f(b)-f(a)}{b-a}=\frac{0-f(b)}{\overline{p}-b}$$

eta $\overline{p}$ bakanduz formula hau ateratzen dugu:

$$\overline{p}=b-\frac{(b-a)}{f(b)-f(a)}f(b). \tag{4.3}$$



4.4. irudia. *Regula falsi* metodoa.

Hurrengo iteraziorako, metodo honen aukerak bisekzioaren berdinak dira, alegia:

- $f(a) \cdot f(\overline{p}) < 0$ bada, $[a, \overline{p}]$ tartean erro bat dago.
- $f(\overline{p}) \cdot f(b) < 0$ bada, $[\overline{p}, b]$ tartean erro bat dago.
- $f(\overline{p}) = 0$ bada, $\overline{p}$ da $f(x)$-ren erro bat.

Aurreko (4.3) formula erabil dezakegu, aukera horiekin batera $\{[a_n, b_n]\}$ tarteen segida sortzeko, horietako bakoitzak erro bat edukiz.

Regula falsi metodoari buruzko oharrak

- Iterazio bakoitzean, hau izango da erroaren hurbilpena:
$$p_n = b_n - \frac{(b_n - a_n)}{f(b_n) - f(a_n)} f(b_n) = \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$$
eta egiazta daiteke, goiko baldintzetan, $\{p_n\}$-k jotzen duela funtzioaren p erro batera.
- Iterazioak gelditu dira $\varepsilon_n = |f(p_n)|$ errorea erabiltzaileak finkaturiko tolerantzia bat baino txikiago denean, edo iterazio kopurua ezarritako maximora heltzen denean.
- Sarritan, funtzioa gorako edo beherako ahurra da $[a, b]$ tartean; orduan, tartearen mutur bat finko geratzen da iterazio guztietan, beste muturrak errorantz jotzen duen bitartean. Hots, zenbakizko soluzioa errorantz doa, alde batetik bakarrik.

4.3. Metodo irekiak

4.3.1. Puntu finkoaren metodoa

Metodo irekiek formula bat erabiltzen dute erro bat iragartzeko. Horrelako formula garatu dezakegu *puntu finkoaren iterazioa* erabiliz, $f(x) = 0$ ekuazioa berrordenatuz ekuazioaren ezkerraldean x izateko, alegia:

$$x = F(x). \tag{4.4}$$

Transformazio hori lor dezakegu manipulazio algebraikoaren bidez, edo x batuz jatorrizko ekuazioaren bi aldeetan. Orduan, (4.4) adierazpenaren bidez iragar daiteke x-rentzat balio berri bat, x-ren balio zahar batetik. Hots,

$$p_n = F(p_{n-1}). \tag{4.5}$$

Konbergentziaren analisia

4.2. teorema. *Demagun $F \in C[a,b]$ dela.*

(i) $y = F(x)$ funtzioaren irudiak $y \in [a,b]$ betetzen badu, $x \in [a,b]$ puntu bakoitzeko, orduan, F-k puntu finko bat dauka $[a,b]$ tartean.

(ii) Demagun, gainera, $F'(x)$ definituta dagoela (a,b) tartean, eta $|F'(x)| < 1$ dugula, $x \in (a,b)$ denean; orduan, F-k puntu finko bakar bat dauka $[a,b]$ tartean.

(i)-ren frogapena. $F(a) = a$ edo $F(b) = b$ bada, ondorioa egia da. Demagun, orduan, $F(a) \in (a,b]$ eta $F(b) \in [a,b)$ egiaztatzen direla. $f(x) = x - F(x)$ funtzioak propietate hau dauka:

$$f(a) = a - F(a) < 0 \text{ eta } f(b) = b - F(b) > 0.$$

Bolzanoren teorema aplikatuz, badago $p \in (a,b)$ puntu bat non $f(p) = 0$ baita. Beraz, $p = F(p)$ eta p puntua $F(x)$-ren puntu finkoa da.

(ii)-ren frogapena. Puntu finko bakar bat dagoela frogatu behar dugu. Aldiz, demagun bi puntu finko daudela, p_1 eta p_2 ($p_1 \neq p_2$); orduan, batez besteko balioaren teorema (Lagrangeren teorema) erabiliz, badago $d \in (a,b)$ puntu bat hau betetzen duena:

$$F'(d) = \frac{F(p_2) - F(p_1)}{p_2 - p_1}.$$

Baina bi puntu horiek puntu finkoak direnez, hau bete behar:

$$F'(d) = \frac{p_2 - p_1}{p_2 - p_1} = 1,$$

eta hori $|F'(x)| < 1$ teoremako baldintzaren kontraesana da. Beraz, ezin dira egon bi puntu finko desberdin. □

4.5. adibidea. *Frogatu $F(x) = \cos(x)$ funtzioak puntu finko bakar bat daukala $[0,1]$ tartean.*

Ebazpena. Bistakoa da $F \in C[0,1]$ dela. Gainera, $F(x) = \cos(x)$ funtzioa tarte horretan beherakorra denez, haren irudiak $F([0,1]) = [\cos(1), 1] \subset [0,1]$ betetzen du. Orduan, aurreko teoremaren (i) atala betetzen du. Alegia, F-k puntu finko bat du $[0,1]$ tartean. Azkenik, $x \in (0,1)$ bada, $|F'(x)| = |-\sin(x)| = \sin(x) \leq \sin(1) \approx 0.8415 < 1$. Beraz, aurreko teoremaren (ii) baldintza betetzen da, eta, ondorioz, bakarra da F-ren puntu finkoa $[0,1]$ tartean. □

4.3. teorema. (Puntu finkoaren teorema). *Demagun (a) $F, F' \in C[a,b]$, (b) $K > 0$, (c) $p_0 \in (a,b)$ eta (d) $F(x) \in [a,b]$, $x \in [a,b]$ guztietarako. Orduan, badago F-ren $p \in [a,b]$ puntu finko bat.*

(i) Baldin $|F'(x)| \leq K < 1$, $x \in [a,b]$ guztietarako, orduan, p da F-ren puntu finko bakarra $[a,b]$ tartean, eta $p_n = F(p_{n-1})$ iterazioak p puntura jotzen du. Kasu horretan, p puntu finko erakargarria dela esaten da. Gainera, $e_n = |p_n - p|$ erroreak ez badira zero, $e_{n+1}/e_n \to F'(p)$, $n \to \infty$ denean.

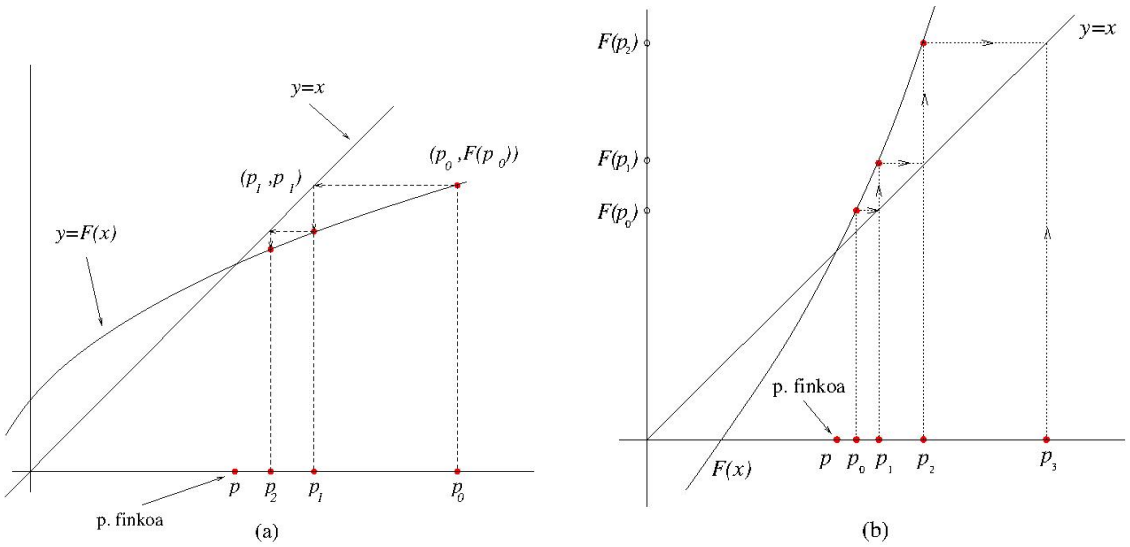
(ii) Baldin $|F'(x)| > 1$ eta $p_0 \neq p$ badira, orduan, $p_n = F(p_{n-1})$ iterazioak ez du jotzen p puntura. Kasu horretan, p puntu finko aldaragarria dela esaten da, eta iterazioak dibergentzia lokala du.

(i)-ren frogapena. 4.2. teoremako (i) atalak eta oraingo teoremaren (a) eta (d) hipotesiek bermatzen dute F-k $[a,b]$ tartean puntu finko bat daukala. Bestalde, $|F'(x)| < 1$ izateak eta 4.2. teoremako (ii) atalak inplikatzen dute F-ren puntu finkoa bakarra dela $[a,b]$ tartean. Bestalde, indukzioz ikusiko dugun bezala, (c) eta (d) baldintzetatik ateratzen da $\{p_n\}_{n=0}^{\infty}$ segidako puntu guztiak $[a,b]$ tartean daudela. Jarraian, $p_0 \in [a,b]$ puntutik hasiz, batez besteko balioaren teorema aplikatuko dugu, p_0 eta p-ren arteko $c_0 \in (a,b)$ existitzen dela ateratzeko, hau egiaztatzen duena:

$$\begin{aligned} |p - p_1| &= |F(p) - F(p_0)| = |F'(c_0)(p - p_0)| = |F'(c_0)| \cdot |p - p_0| \\ &\leq K|p - p_0| < |p - p_0|, \end{aligned} \tag{4.6}$$

Beraz, p_1 p-tik hurbilago dago p_0 baino. Horrela arrazoituz, oro har, hau izango dugu:

$$\begin{aligned} |p - p_n| &= |F(p) - F(p_{n-1})| = |F'(c_{n-1})(p - p_{n-1})| = |F'(c_{n-1})| \cdot |p - p_{n-1}| \\ &\leq K|p - p_{n-1}| < |p - p_{n-1}|. \end{aligned} \tag{4.7}$$



4.5. irudia. (a) P. finko erakargarria ($|F'(x)| < 1$).
(b) P. finko alderagarria ($|F'(x)| > 1$).

Frogapena bukatzeko, hau betetzen dela ikusi behar dugu:

$$\lim_{n\to\infty} |p - p_n| = 0.$$

Lehenik, indukzioz (4.6)-(4.7) erlazioak erabiliz, hau erraz lortzen da:

$$|p - p_n| \leq K^n |p - p_0|. \tag{4.8}$$

Gainera, $0 < K < 1$ denez,

$$0 \leq \lim_{n\to\infty} |p - p_n| \leq \lim_{n\to\infty} K^n |p - p_0| = 0.$$

Ondorioz, $\lim_{n\to\infty} |p - p_n| = 0$, eta, orduan, $\lim_{n\to\infty} p_n = p$.

Azkenik, $e_n = |p_n - p|$ erroreak ez badira zero, deribatuaren definizioaz zera betetzen da:

$$\frac{e_{n+1}}{e_n} = \frac{|p_{n+1} - p|}{|p_n - p|} = \frac{|F(p_n) - F(p)|}{|p_n - p|} \to |F'(p)|.$$

(ii) frogapena ariketa gisa geratzen da. □

Kontuan izan 4.3. teoremaren (i) ataleko baldintzak betetzen direnean (4.8) betetzen dela, eta, ondorioz, puntu finkoaren iteraziorako errorearen borne hau dugula:

$$|p - p_n| \leq K^n |p - p_0|, \quad n \geq 1 \text{ guztietarako.}$$

K horren balio hurbildu on bat $F'(\widetilde{p})$ izan daiteke, $\widetilde{p}$ puntua p puntu finkotik nahiko hurbila bada (p bera sartuta, ezagutzen bada).

4.1. Korolarioa. *Demagun $F, F' \in C(p-r_0, p+r_0)$, non $F(p) = p$. Orduan, hau betetzen bada:*

$$|F'(p)| < 1, \tag{4.9}$$

r positibo bat existitzen da, $r < r_0$, non $p_0 \in [p-r, p+r]$ betetzen bada, $p_n = F(p_{n-1})$ segidako gai guztiak $[p-r, p+r]$ tartean baitaude eta $\{p_n\} \to p$.

Aurreko teoremako (i) atala kontuan hartuz froga daiteke (ariketa gisa geratzen da). □

4.6. adibidea. *Puntu finkoaren metodoa erabiliz, kalkulatu $\cos x - x = 0$ ekuazioaren soluzio hurbildu bat, errore-tolerantzia=0.0001 eta $p_0 = \pi/4 = 0.785398$ hartuz. Gehienez, egin hamar iterazio. (Errorea $|p - p_n|$ da).*

Ebazpena. Kasu honetan, $f(x) = \cos x - x = 0$ ekuazioa dugu, eta $[0, \pi/2]$ tartean p erro bakar bat dauka, zeren eta $F(x) = \cos x$ funtzioak 4.2. teoremaren baldintzak betetzen baititu ($|F'(x)| = |-\sin(x)| < 1$ da $(0, \pi/2)$ tartean). Gainera, 4.3. teoremaren (i) ataleko baldintzak betetzen direnez, p puntu finko hori erakargarria da. Ondorioz, metodo hori konbergentea da. Taula honetan $|p - p_n|$ errorea $|p_{n+1} - p_n| = |F(p_n) - p_n| = |f(p_n)|$ balioaz hurbiltzen da.

n	p_n	$F(p_n)$	$\lvert f(p_n)\rvert$
0	0.785398	0.707107	0.078291
1	0.707107	0.760244	0.053137
2	0.760244	0.724668	0.035576
3	0.724668	0.748720	0.024052
4	0.748720	0.732561	0.016159
5	0.732561	0.743464	0.010903
6	0.743464	0.736128	0.007336
7	0.736128	0.741074	0.004945
8	0.741074	0.737744	0.003330
9	0.737744	0.739988	0.002244

4.2. taula. Puntu finkoaren iterazioak.

Hona hemen puntu finkoaren iterazioa:

$$p_n = F(p_{n-1}) = \cos(p_{n-1}), \quad n \geq 1.$$

Beraz, hamar iterazio egin ondoren, ez dugu lortu nahi genuen soluzio hurbildua. Iterazio gehiago eginez gero, eskatutako soluzio hurbildura helduko da. Zenbat iteraziotan gehienez? □

4.3.2. Konbergentziaren ordena

Segida baten konbergentzia neurtzeko erabiltzen den kontzeptu bat da. Askotan, segida hori algoritmo batek sortutakoa da. Demagun $\{x_n\} \to x^*$ segida konbergentea. Orduan, hau betetzen da:

$$\lim_{k\to\infty} |x_k - x^*| = 0.$$

4.1. Definizioa. *Konbergentzia* ***Q-lineala*** *dela esango dugu $r \in [0,1)$ konstante bat existitzen bada, non hau betetzen baita:*

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|} \leq r, \quad k \text{ nahiko handi baterako.}$$

Horrek esan nahi du iterazio bakoitzean x^*-ra arteko distantzia gutxienez txikitzen dela faktore konstante batean.

Baldintza hori betetzen da hau badugu:

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|} < 1.$$

Adibidez, $\{1+(0.5)^k\} \to 1$ segidak konbergentzia Q-lineala du. Horren frogapena ariketa gisa geratzen da.

4.2. Definizioa. *Konbergentzia* ***Q-superlineala*** *dela esango dugu $\{r_k\} \to 0$ segida bat existitzen bada, non hau betetzen baita:*

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|} \leq r_k.$$

Hots, $\lim_{k\to\infty} \dfrac{|x_{k+1} - x^*|}{|x_k - x^*|} = 0$ *denean.*

Adibidez, $\{1+k^{-k}\} \to 1$ segidak konbergentzia Q-superlineala du. Horren frogapena ariketa gisa geratzen da.

4.3. Definizioa. *Konbergentzia* ***Q-koadratikoa*** *dela esango dugu $r \geq 0$ konstante bat existitzen bada, non hau betetzen baita:*

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} \leq r, \quad k \text{ nahiko handi baterako.}$$

Kasu honetan, r konstantea ez da derrigorrez 1 baino txikiagoa izan behar.

Baldintza hori betetzen da hau badugu:

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} < \infty,$$

hau da, limite hori finitua bada.

Adibidez, $\{1 + (0.5)^{2^k}\} \to 1$ segidak konbergentzia Q-koadratikoa du. Horren frogapena ariketa gisa geratzen da.

4.4. Definizioa. *Oro har, konbergentzia* ***Q-konbergentzia ordena*** p ***gutxienez*** *dela esango dugu $r \geq 0$ konstante bat existitzen bada, non hau betetzen baita:*

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} \leq r, \quad k \text{ nahiko handi baterako.}$$

Konbergentzia koadratikoa bada, superlineala da (frogapena ariketa gisa geratzen da).

Oro har, «Q» ez da idazten (Q «quotient» hitzetik dator). Puntu finkoaren metodoaren konbergentzia (konbergente denean) lineala da (ikus (4.7) adierazpena).

4.3.3. Newton-Raphson-en metodoa

Metodo hau (Newtonen metodoa ere esaten zaio) oso azkarra da, baina badu arazo bat: hasierako puntuak errotik nahiko hurbil egon behar du.

Oinarri analitikoa Taylorren serie-garapenean datza. Izan bedi $f \in C^2[a,b]$, non p erroa baita eta $p_0 \in [a,b]$ haren hurbilketa on bat. Orduan, $|p_0 - p|$ txikia eta $f(p_0) \neq 0$ izango dira. Funtzio horren p_0-ren inguruneko Taylorren garapena hau da:

$$f(x) = f(p_0) + (x - p_0)f'(p_0) + \frac{1}{2}(x - p_0)^2 f''(c),$$

non c balioa p_0 eta x-ren artean baitago. Beraz, $f(p) = 0$ bete behar denez, hau lortuko dugu:

$$f(p_0) + (p - p_0)f'(p_0) + \frac{1}{2}(p - p_0)^2 f''(c) = 0 \Rightarrow f(p_0) + (p - p_0)f'(p_0) \approx 0.$$

Baina, azken emaitza lortzeko, $(p - p_0)^2 f''(c)/2$ batugaia ezabatu dugu, baztergarria dela suposatzen baita; izan ere, $|p - p_0|$ tartea nahiko txikia dela jo dugu. Arrazoi horretan datza, hain zuzen ere, metodoaren baldintza.

Beraz, $f(p_0)+(p-p_0)f'(p_0)\approx 0$ dela joz eta p askatuz, $p\approx p_0-\dfrac{f(p_0)}{f'(p_0)}=p_1$ hurbilketa lortzen dugu. Iterazio hori errepikatuz, metodoaren iterazio-formula hau lortuko dugu:

$$p_n=p_{n-1}-\frac{f(p_{n-1})}{f'(p_{n-1})}. \tag{4.10}$$

Argi dago $\{p_n\}$ segida ondo definituta izateko, $f'(p_n)\neq 0$ bete behar dela n azpiindize guztietarako.

Newtonen iterazioa puntu finkoaren iterazioa da, *iterazio-funtzio* honetarako:

$$F(x)=x-\frac{f(x)}{f'(x)}.$$

Kontuan izan p errorako $p=F(p)$ dela. Kasu honetan, $f'(p)\neq 0$ hartuz, zera dugu:

$$F'(x)=1-\frac{(f'(x))^2-f(x)f''(x)}{(f'(x))^2}=\frac{f(x)f''(x)}{(f'(x))^2}.$$

Hipotesiaren arabera, $f(p)=0$ denez, $F'(p)=0$ dugu. Beraz, $[a,b]$ tartean $F'(x)$ jarraitua denez eta $F'(p)=0$, badago p-ren $(p-\delta,p+\delta)$ ingurune bat non $|F'(x)|<1$ den. Ondorioz, 4.3. teoremako (i) atalaren arabera, iterazio hori konbergentea da p_0 ingurune horretako puntu bat bada. Baina, zein da haren konbergentzia-ordena? Jarraian erantzungo dugu.

4.4. teorema. (Newtonen teorema.) *Izan bitez $f\in C^2[a,b]$ eta $f(x)=0$ ekuazioko $p\in[a,b]$ erroa. Demagun $\rho>0$ baterako $|f'(x)|>\rho$ betetzen dela $x\in[a,b]$ guztietarako. Orduan, $\eta>0$ bat existitzen da, non: $|p_0-p|<\eta$ betetzen bada, (4.10) adierazpenak p_0-tik sortutako $\{p_n\}$ segidak p puntura jotzen baitu konbergentzia koadratikoarekin.*

Frogapena. $n=1$-erako hau dugu:

$$\begin{aligned} p_1-p &= \left(p_0-\frac{f(p_0)}{f'(p_0)}\right)-p=p_0-p-\frac{f(p_0)-f(p)}{f'(p_0)} \\ &= \frac{1}{f'(p_0)}[f(p)-f(p_0)-f'(p_0)(p-p_0)]. \end{aligned}$$

Makoen arteko gaia Taylorren garapenaren $O(|p-p_0|^2)$ hondarra da (ikus 3.4.3. azpiatala); hots, badago $M>0$ zenbaki bat hau betetzen duena:

$$|f(p)-f(p_0)-f'(p_0)(p-p_0)|\leq M|p-p_0|^2.$$

M hori nahiko handia hartuko dugu $|f''(x)/2|<M$ bete dadin $[a,b]$ tartean ere ($f''(x)$ jarraitua denez, bornatua da $[a,b]$ tartean). Ondorioz,

$$|p_1-p|\leq\frac{M}{|f'(p_0)|}|p_0-p|^2\leq\frac{M}{\rho}|p_0-p|^2$$

$n = 2$ denean, hau dugu:

$$|p_2 - p| \leq \frac{M}{|f'(p_1)|}|p_1 - p|^2 \leq \frac{M}{\rho}|p_1 - p|^2.$$

Horrela, $r = M/\rho$ bada, $n = 1, 2, \ldots$ guztietarako emaitza honetara heltzen gara:

$$|p_n - p| \leq r|p_{n-1} - p|^2.$$

Hori dela eta, ondorio hau lortuko dugu:

$$\frac{|p_n - p|}{|p_{n-1} - p|^2} \leq r. \tag{4.11}$$

Beraz, $n = 1, 2, \ldots$ zera dugu.

$$r|p_{n-1} - p| < 1 \;\Rightarrow\; |p_n - p| < |p_{n-1} - p|. \tag{4.12}$$

Izan bedi $\eta \in \mathbb{R}$, non $\eta \leq 1/r$, orduan nahikoa da $p_0 \in (p-\eta, p+\eta)$ hartzea $r|p_0 - p| < 1$ betetzeko. Aurrekoak (4.12)-ren bitartez $r|p_1 - p| < 1$ inplikatzen du, eta (4.12) berriro erabiliz, $|p_2 - p| < |p_1 - p|$ dugu, eta horrela elkarren segidan. Hots:

$$|p_n - p| < |p_{n-1} - p| < \ldots < |p_1 - p| < |p_0 - p|, \quad \forall n \in \mathbb{N}, n > 0.$$

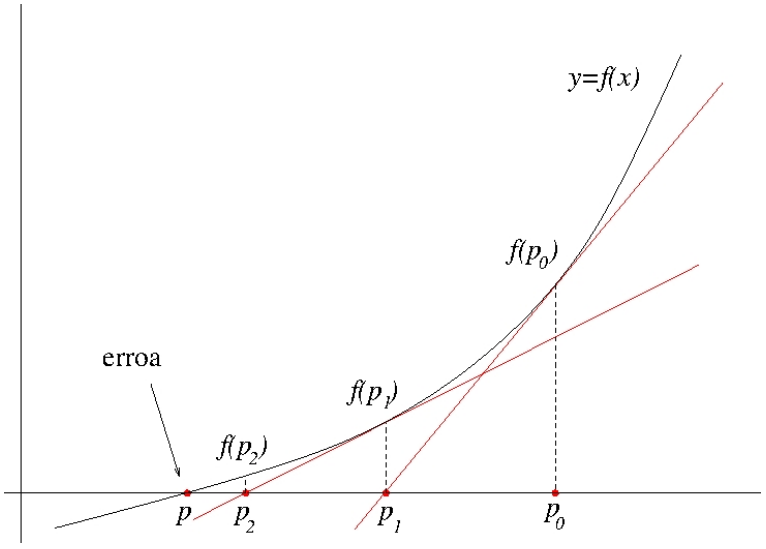
Gainera, $K = r|p_0 - p|$ definituz $r|p_n - p| < K$ da, $n > 0$ azpiindize guztietarako, eta zera dugu:

$$|p_n - p| < K|p_{n-1} - p| < K^2|p_{n-2} - p| < \ldots < K^n|p_0 - p|, \quad K < 1.$$

Ondorioz, p_0 hasierako puntuak $|p_0 - p| < \eta$ betetzen badu, zera dugu:

$$\lim_{n\to\infty} |p_n - p| = 0$$

eta $\{p_n\} \to p$ konbergentzia koadratikoarekin, ikus (4.11). □



4.6. irudia. Newtonen metodoa.

Geometrikoki, metodoak hau egiten du: OX ardatza ebakitzen duen $y = f(x)$ kurbaren ordez, $(p_0, f(p_0))$ puntutik pasatzen den kurbaren zuzen ukitzailea erabiltzen du. Gero, ebakidura OX-rekin ($y = 0$-rekin) kalkulatzen da, eta, hala, p_1 puntua lortzen da. Teknika hori errepikatuz, $(p_n, f(p_n))$ puntutik pasatzen den $f(x)$-ren ukitzailearen ekuazioa lortzen da, hots, $y - f(p_n) = f'(p_n)(x - p_n)$. Orain, OX ardatzarekin ($y = 0$-rekin) ebakiz, x aska daiteke. Ondorioz, $x = p_n - \dfrac{f(p_n)}{f'(p_n)}$ lortzen da, eta $p_{n+1} = x$. Begira ezazu 4.6. irudia.

4.1. algoritmoa. Newtonen algoritmoa.

0 urratsa. SARRERA. Sartu: $f(x)$ funtzioa, $f'(x)$ funtzioa, p_0 hasierako hurbilpena, ε_{max} errorearen tolerantzia, ze_{max} eskatutako zehaztasun erlatiboa eta i_{max} iterazioen kopuru maximoa.

1 urratsa. Jarri $i = 0$ eta definitu:

$$\begin{aligned} fp_0 &= f(p_0);\\ dfp_0 &= f'(p_0);\\ \varepsilon &= |fp_0|;\\ ze &= ze_{max} + 1; \end{aligned}$$

2 urratsa. $i \leq i_{max}$ eta $\varepsilon > \varepsilon_{max}$ eta $ze > ze_{max}$ diren bitartean, egin hau:

(a) Kalkulatu p honela:

$$p = p_0 - fp_0/dfp_0;$$

(b) Egin hau:

$$\begin{aligned} p_{zaharra} &= p_0;\\ p_0 &= p;\\ fp_0 &= f(p_0);\\ dfp_0 &= f'(p_0); \end{aligned}$$

(c) Egin $i = i + 1$;

(d) Kalkulatu zehaztasun erlatiboa: $ze = |p_0 - p_{zaharra}|/|p_0|$;

(e) Kalkulatu errorea: $\varepsilon = |fp_0|$;

3 urratsa. IRTEERA. Emaitzak: p erroa, i erabilitako iterazioen kopurua, ze zehaztasun erlatiboa eta ε errorea.

4.7. adibidea. *Newtonen metodoa erabiliz, kalkulatu $x = \cos x$ ekuazioaren soluzio hurbildu bat, errore-tolerantzia 0.00001 hartuz.*

Ebazpena. Izan bedi $f(x) = \cos x - x$. Orain, $f(x)$-ren erro bat aurkitu behar dugu. Baina, lehendabizi, soluzio bat bakartu behar dugu tarte batean. Horretarako, aldez aurretik ikusitako (ii) propietatea (Bolzanoren teorema) erabiliko dugu. Orduan, $f(\pi/2) = -\pi/2 < 0$ eta $f(0) = 1 > 0$ direnez, funtzioak gutxienez erro bat dauka $(0, \pi/2)$ tartean (grafikoki ikus daiteke erro bakarra dela). Hona hemen Newtonen iterazioa kasu horretan:

$$p_n = p_{n-1} - \frac{\cos(p_{n-1}) - p_{n-1}}{-\sin(p_{n-1}) - 1}, \quad n \geq 1$$

p_0 geometriaren arabera hautatuko dugu. Kasu honetan, $p_0 = \pi/4 = 0.785398$ egokia dela ikus daiteke.

n	p_n	$\lvert f(p_n)\rvert$
0	0.785398	0.078291
1	0.739542	0.000765
2	0.739085	$2.2 \cdot 10^{-7}$

4.3. taula. Newtonen iterazioak.

Beraz, bi iterazio egin ondoren, $x = 0.739085$ dugu ekuazioaren soluzio hurbildua. □

4.8. adibidea. *Newtonen metodoa erabiliz eta $p_0 = -3$ hartuz, kalkulatu $e^x = \sin x$ ekuazioaren soluzio hurbildu bat errore-tolerantzia 0.0001 hartuz.*

Ebazpena. Hemen $f(x) = e^x - \sin x$ dugu eta

$$p_n = p_{n-1} - \frac{e^{p_{n-1}} - \sin(p_{n-1})}{e^{p_{n-1}} - \cos(p_{n-1})}, \quad n \geq 1$$

n	p_n	$\lvert f(p_n)\rvert$
0	-3.0000	0.19090
1	-3.1836	0.00056
2	-3.1831	$3.8 \cdot 10^{-5}$

4.4. taula. Newtonen iterazioak.

Beraz, bi iterazio egin ondoren, $x = -3.1831$ dugu ekuazioaren soluzio hurbildua. □

Newtonen metodoari buruzko oharrak

- Ikusi dugun bezala, metodo hau konbergente izateko, hasierako puntuak p soluziotik nahiko hurbil egon behar du; horrelako metodo bati *metodo lokala* deritzogu.

- Oro har, metodoak konbergentzia arazoak izaten ditu p soluzioaren ingurune batean $f'(x) \approx 0$ denean. Ariketa gisa, aurkitu $x^{10} - 1 = 0$ ekuazioaren erro positibo bat $p_0 = 0.5$ hartuz. Zer gertatzen da?

- Erroa anizkoitza denean, konbergentzia lineala da. Demagun $f(x) = (x-p)^m G(x)$ funtzioa dugula eta $G(p) \neq 0$; orduan, $|p_n - p|/|p_{n-1} - p| \approx 1 - \dfrac{1}{m}$ da n nahiko handi baterako. Beraz, anizkoiztasuna handitzen den heinean, konbergentzia motelago bihurtzen da. Horrelako kasuetan, oso egokia da $f(x)$ funtzioaren ordez $g(x) = f(x)/f'(x)$ funtzioa hartzea; horrek *Newtonen metodo orokortua* ondorioztatzen du. Adibidez, $f(x) = x^3 - 3x + 2$ funtzioak erro bikoitza du $x = 1$ puntuan. Aztertu gertatzen dena Newtonen metodoa erabiltzen badugu erroa kalkulatzeko $p_0 = 1.2$ hartuz. Geroago, erabili Newtonen metodo orokortua.

- Hasierako balioa, p_0, errotik nahiko urrun badago, gerta daiteke metodoa konbergente ez izatea. Adibidez, $f(x) = xe^{-x}$ kasuan, bistan da erroa $p = 0$ dela. Aztertu gertatzen dena $p_0 = 2$ hartzen badugu.

- Beste gertaera berezi bat *periodikotasuna* da; hori gertatzen da $\{p_n\}$ segidaren gaiak periodikoki errepikatzen (edo ia errepikatzen) hasten direnean. Adibidez, aztertu $f(x) = x^3 - x - 3$ eta $p_0 = 0$ hartzen dugunean.

- $|F'(x)| > 1$ betetzen bada p erroa daukan tartean, orduan, oszilazio dibergente bat gerta daiteke. Adibidez, $f(x) = \arctan(x)$ funtzioa $p_0 = 1.45$ hartuz. Aztertu kasu hori.

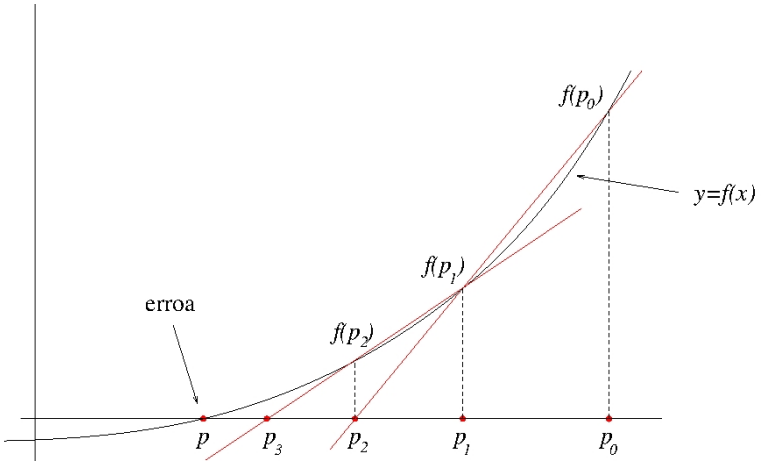
4.3.4. Ebakitzailearen metodoa

Batzuetan, ezin da $f'(x)$ deribatua kalkulatu iterazioaren formulan ordezkatzeko; orduan, zenbakizko metodoak erabiliz malda zehaztu dezakegu. Hots, Newtonen metodoaren ordez *ebakitzailearen metodoa* erabiliko dugu. Kasu horietan, deribatua hurbil dezakegu adierazpen honen bitartez:

$$f'(p_k) \approx \frac{f(p_k) - f(p_{k-1})}{p_k - p_{k-1}},$$

orduan, Newtonen ekuazioan ordezkatuz $k = n - 1$-erako,

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}. \tag{4.13}$$



4.7. irudia. Ebakitzailearen metodoa.

Adierazpen horretan ikusten den bezala, hasierako bi puntu behar ditugu. Hau da, p_0 eta p_1 puntuek errotik nahiko hurbil egon behar dute (Newtonen metodoan bezala); p_2 izango da lortuko duguna 1. iterazioan.

Newtonen metodoan, beste aukera bat da deribatua hurbiltzea oraingo iterazioa zertxobait aldatuz, alegia:

$$f'(p_k) \approx \frac{f(p_k + \delta) - f(p_k)}{\delta},$$

non $\delta > 0$ balio txiki bat baita, eta, orduan, *ebakitzailearen metodo aldatua* izango dugu:

$$p_n = p_{n-1} - \frac{\delta \cdot f(p_{n-1})}{f(p_{n-1} + \delta) - f(p_{n-1})}. \tag{4.14}$$

Ebakitzailearen metodoari buruzko oharrak

- Metodo hau Newtonen metodoaren deribatu gabeko hurbilpen bat denez (batez ere (4.14) iterazioa erabiltzen badugu), konbergentziaren ordena α da, non:

  $$\alpha = \frac{1 + \sqrt{5}}{2} \approx 1.618$$

  urrezko ratioa baita. Bereziki, konbergentzia superlineala da, baina ez da heltzen koadratikoa izatera. Konbergentziaren ordena 1.618 da baldin $f \in C^2$ bada, p erro bakun (hots, anizkoiztasuna=1) bada baten ingurune batean, eta p_0, p_1 hasierako puntuak p-tik nahiko hurbil daudenean.

- Baina, $\overline{p} \in [p_0, p_1]$ puntu baterako $f'(\overline{p}) = 0$ bada, algoritmoa baliteke konbergentea ez izatea.

4.2. algoritmoa. Ebakitzailearen algoritmoa.

0 urratsa. SARRERA. Sartu: $f(x)$ funtzioa, p_0 eta p_1 hasierako hurbilpenak, ε_{max} errorearen tolerantzia, ze_{max} eskatutako zehaztasun erlatiboa, eta i_{max} iterazioen kopuru maximoa.

1 urratsa. Jarri $i = 1$ eta definitu:

$$q_0 = f(p_0);$$
$$q_1 = f(p_1);$$
$$\varepsilon = |q_1|;$$
$$ze = |p_1 - p_0|/|p_1|;$$

2 urratsa. $i \leq i_{max}$ eta $\varepsilon > \varepsilon_{max}$ eta $ze > ze_{max}$ diren bitartean, egin hau:

(a) Kalkulatu p honela:

$$p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0);$$

(b) Egin hau:

$$p_0 = p_1;$$
$$p_1 = p;$$
$$q_0 = q_1;$$
$$q_1 = f(p_1);$$

(c) Egin $i = i + 1$;

(d) Kalkulatu zehaztasun erlatiboa: $ze = |p_1 - p_0|/|p_1|$;

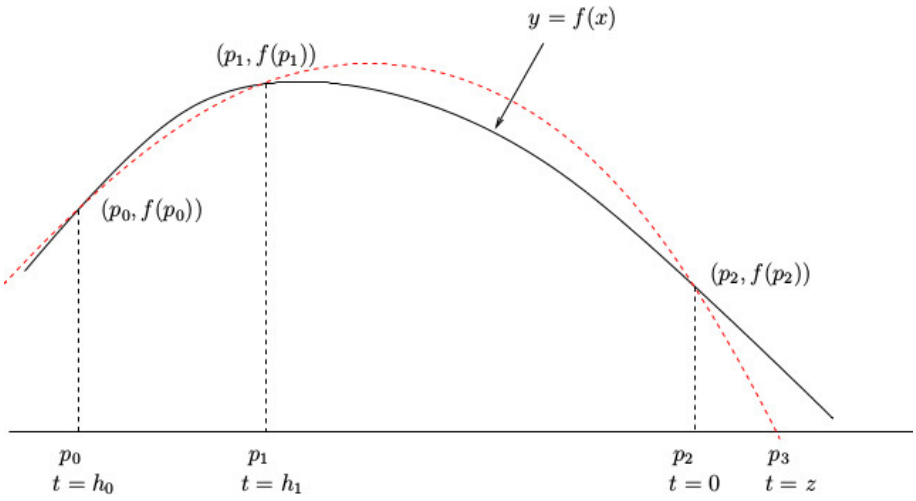
(e) Kalkulatu errorea: $\varepsilon = |q_1|$;

3 urratsa. IRTEERA. Emaitzak: p erroa, i erabilitako iterazioen kopurua, ze zehaztasun erlatiboa eta ε errorea.

4.3.5. Muller-en metodoa

Ebakitzailearen metodoak bi puntu erabiltzen ditu puntu berri bat lortzeko. Jarraian azalduko dugun metodoari *Mullerren metodoa* deritzogu.

Demagun iterazioen hasieran $(p_0, f(p_0))$, $(p_1, f(p_1))$ eta $(p_2, f(p_2))$ hiru puntuak ditugula. Orduan, hiru puntu horietatik pasatzen den parabola bakarra eraikitzen dugu, eta, abszisa-ardatzarekin ebakitze-puntua kalkulatuz, hurrengo p_3 puntua izango dugu. Jo dezagun p_2 erroaren hurbilpen hoberena dela, eta demagun hiru puntuetatik igarotzen den 4.8. irudiko parabola dela.



4.8. irudia. Mullerren metodoa.

Izan bitez $t = x - p_2$ aldagai-aldaketa, eta $h_0 = p_0 - p_2$ eta $h_1 = p_1 - p_2$ diferentziak. Izan bedi $y = at^2 + bt + c$ polinomio koadratikoa. Hiru puntu horietatik igarotzen den parabolaren ekuazioko a, b eta c koefizienteak kalkulatuko ditugu. Puntu bakoitzeko, hau dugu:

$$\begin{array}{lll} p_0 - \text{rako} & t = h_0 \text{ da eta hau bete behar:} & ah_0^2 + bh_0 + c = f(p_0), \\ p_1 - \text{erako} & t = h_1 \text{ da eta hau bete behar:} & ah_1^2 + bh_1 + c = f(p_1), \\ p_2 - \text{rako} & t = 0 \text{ da eta hau bete behar:} & a0^2 + b0 + c = f(p_2). \end{array}$$

Hau da sistema horren soluzioa:

$$c = f(p_2),$$

$$a = \frac{e_0 h_1 - e_1 h_0}{h_1 h_0^2 - h_0 h_1^2},$$

$$b = \frac{e_1 h_0^2 - e_0 h_1^2}{h_1 h_0^2 - h_0 h_1^2},$$

non $e_0 = f(p_0) - c$ eta $e_1 = f(p_1) - c$. Orain, parabola horren $t = z_1, z_2$ erroak formula hau erabiliz lortzen dira:

$$z = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}.$$

Dakigunez, formula hori da bigarren mailako ekuazioen erroak aurkitzeko dugun ohiko formularen baliokidea; baina kasu honetan hobe da hura erabiltzea, jada ezagutzen baitugu $c = f(p_2) \approx 0$ dela eta, ondorioz, $b \approx \sqrt{b^2 - 4ac}$.

Metodoaren egonkortasuna ziurtatzeko, balio absolutu txikiena duen erroa aukeratuko dugu. Beraz, $b > 0$ bada, zeinu positiboa aukeratuko dugu erro karraturako. Aldiz, $b < 0$ bada, zeinu negatiboa aukeratuko dugu erro karraturako. Puntu berria hau izango da:

$$p_3 = p_2 + z.$$

Orduan, p_0 eta p_1 puntu berriak $\{p_0, p_1, p_2\}$ puntu zaharren artean aukeratuko ditugu, eta p_3-tik hurbilenak izango dira; p_2 berria p_3 izango da.

Prozesu hau errepikatuz, iterazio-segida bat lortzen da, eta horrek, baldintza egokietan, $f(x)$ funtzioaren errora joko du. Mullerren metodoak arazo larriak izan ditzake, ez badira desberdinak $f(p_1)$, $f(p_2)$ eta $f(p_3)$. Hala ere, errotik nahiko hurbil badago, oso azkarra da; haren konbergentzia-ordena 1.84 da. Ebakitzailearena 1.62 da eta Newtonen metodoarena 2 (erro bakunen kasurako).

Gerta daiteke sortutako parabolak x ardatza ez ukitzea; orduan, erro konplexuak izango ditugu, non alde irudikaria txikia baita alde errealarekin konparatuta. Kasu horretan, alde irudikaria kentzen da eta prozesuarekin jarraitzen da.

4.3.6. Alderantzizko interpolazio koadratikoa

Metodo honetan, x-rekiko koadratikoa izan beharrean, y-rekiko izango da. Hots, $(a, f(a))$, $(b, f(b))$ eta $(c, f(c))$ hiru puntuak ezagutuz, alderantzizko funtzioarekin arituko gara; lortuko dugun polinomio koadratikoa $P(y)$ motakoa izango da, eta interpolazio-baldintza hauek erabiliz zehaztuko dugu:

$$a = P(f(a)), \quad b = P(f(b)), \quad c = P(f(c)).$$

Parabola honek beti ebakitzen du x ardatza; hots, $y = 0$. Beraz, $x = P(0)$ da hurrengo iterazioa. Metodo honi IQI (Inverse Quadratic Interpolation) izen laburtua emango diogu. MATLABen kodean; hau izan liteke:

```
function x=iqi(f,a,b,c)
   while abs(c-b) > eps*abs(b);
      x = polyinterp([f(a),f(b),f(c)],[a,b,c],0)
      a = b;
      b = c;
      c = x;
   end
end
```

Metodo horren arazoa zera da: interpolazio-polinomioak $f(a)$, $f(b)$ eta $f(c)$ desberdinak izatea eskatzen du. Baina, guk ez dugu berme hori. Adibidez, kalkulatu nahi badugu $\sqrt{2}$ balioa $f(x) = x^2 - 2$ funtzioa erabiliz, eta $a = -2$, $b = 0$ eta $c = 2$, orduan, $f(a) = f(c)$ eta lehenengo urratsa ez dago definituta. Aldiz, $a = -2.001$, $b = 0$ eta $c = 1.999$ balioekin hasten bagara, hurrengo iterazioa $x = 500$ izango da.

4.3.7. Zeroin algoritmoa

Metodo honetan, bisekzioaren konbergentziarekiko fidagarritasuna, ebakitzaile-metodoaren konbergentzia-azkartasuna eta IQI metodoa konbinatzen dira. Dekker-ek eta Brent-ek garatu zuten algoritmo honen lehenengo bertsioa (ikus [5, 8]); geroko bertsioek asko hobetu dute algoritmo hori. Hona hemen algoritmoaren laburpen bat:

4.3. algoritmoa. Zeroin algoritmoa.

a eta b emanda, puntu horietarako $f(a) \cdot f(b) < 0$ betetzen da; beraz, f jarraitua bada $[a, b]$-n, erroa tartearen barnean dago.

- Erabili ebakitzailearen iterazioa, a eta b-ren arteko c lortzeko.
- Gero, errepikatu honako urrats hauek, $|b - a| < \varepsilon|b|$ edo $f(b) = 0$ bete arte:

  1 urratsa. Ordenatu a, b eta c honela:

  ▷ $f(a)$-k eta $f(b)$-k zeinu desberdinak dituzte.
  ▷ $|f(b)| \leq |f(a)|$.
  ▷ c da b-ren aurreko balioa; beraz, $c = a$ izan liteke.

  2 urratsa. $c \neq a$ bada, IQI iterazioa kalkulatuko dugu.

  3 urratsa. $c = a$, ebakitzailearen iterazioa kalkulatuko dugu.

  4 urratsa. IQIren edo ebakitzailearen emaitza $[a, b]$-n badago, hartu egingo dugu; ez bada horrela, bisekzioaren metodoa erabiliko dugu hurrengo iterazioa lortzeko.

Algoritmo honen inplementazioa MATLABeko `fzero` funtzioa da, eta eraginkortasun handia du ekuazio ez-linealen erroen kalkuluan. Kontuan izan ez duela deribaturik erabiltzen. Adibidez, $f(x) = x - 0.5\sin(x) - 0.7$ funtzioaren erroaren kalkulurako, $x_0 = 1$ hartuz, `fzero('x-0.5*sin(x)-0.7',1)` instrukzioak 1.1580 ematen du eta $f(1.1580) = -1.1102 \times 10^{-16}$.

Honela ere idatz daiteke: `fzero('x-0.5*sin(x)-0.7',[1,2])`, x_0-ren ordez [1,2] tartea jarriz, baina bertan erro bat izan behar du, bestela errore-mezu bat emango du.

Badago `roots` funtzioa ere; `r=roots([1,-6,5])` idazten badugu, $p(x) = x^2 - 6x + 5 = 0$ ekuazioaren erroak emango dizkigu, honela: $\boldsymbol{r} = (1, 5)^T$; hots, $\boldsymbol{r}$ bektoreak $p(x)$ polinomioaren erroak gordeko ditu.

4.4. Algoritmoak gelditzeko irizpideak

Hauek izaten dira algoritmo horiek gelditzeko irizpideak:

1. Errorea, ε, aldez aurretik finkatutako $\varepsilon_{max} > 0$ errorearen tolerantzia baino txikiagoa izatea, alegia:
$$\varepsilon = |f(p_n)| < \varepsilon_{max}.$$

2. Ondoz ondoko bi iterazioren arteko diferentzia (zehaztasun absolutua) nahiko txikia izatea; hots, zehaztasun absolutua aldez aurretik finkatutako $za > 0$ tolerantzia baino txikiagoa izatea, alegia:
$$|p_n - p_{n-1}| < za.$$

3. Bi iterazioren arteko zehaztasun erlatiboa nahiko txikia izatea; hots, diferentzia erlatibo hori aldez aurretik finkatutako $ze > 0$ zehaztasun erlatiboa baino txikiagoa izatea, alegia:
$$\frac{|p_n - p_{n-1}|}{|p_n|} < ze.$$
Hau da $p_n \approx 0$-ren arazoa gainditzeko erabil dezakegun beste zehaztasun mota bat:
$$\frac{|p_n - p_{n-1}|}{1 + |p_n|} < ze.$$
Kontuan izan $p_n = 0$ bada, zatidura hori zehaztasun absolutua dela.

4. Iterazioen kopuru maximo bat, n_{max}, finkatzea. Alegia, hau bada:
$$n > n_{max},$$
algoritmoa gelditu egingo da.

4.5. Problemak

Eskuz ebazteko problemak:

1. $e^x - 3x = 0$ ekuazioak erro bat dauka $p \approx 0.61906129$ puntuan. Erabili bisekzio-metodoaren sei iterazio, erro hori aurkitzeko, $[0, 1]$ tartetik hasiz. Zenbat iterazio behar dira erroaren hurbilpena balioztatzeko lau zifra esanguratsurekin (hots, 0.6190 lortzeko)?

2. Bistakoa da $(x - 0.4)(x - 0.6) = x^2 - x + 0.24 = 0$ ekuazioak $x = 0.4$ eta $x = 0.6$ erroak dituela. Kontuan izan $[0, 1]$ tarteko muturrak ez direla onak bisekzioaren metodoa hasteko; zergatik? Funtzioaren marrazketa erabiliz, aurkitu erroen tarte egokiak bisekzio-metodoak erro bakoitzera jotzeko. Bestalde, $[0.5, 1]$ tartetik bisekzio-metodoaz bilaketa hasten badugu, zein da $|p - p_5|$ zehaztasunaren bornea bost iterazio igaro ondoren? Zein da benetako zehaztasuna, bost iterazio igaro ondoren?

3. Erabili bisekzioaren metodoa, ekuazio hauen erro positibo txikiena aurkitzeko:

   (a) $e^x - x - 2 = 0$,

   (b) $x^2 - e^x = 0$,

   (c) $\sin(x) - 2\cos(x) = 0$,

   (d) $x^3 - x^2 - 2x + 1 = 0$,

   (e) $2e^{-x} - \sin x = 0$,

   (f) $3x^3 + 4x^2 - 8x - 1 = 0$.

   Kasu bakoitzean, aurkitu tarte egoki bat eta, gero, kalkulatu erroaren hurbilpena % 0.5 zehaztasun erlatiboarekin. Lehen erabilitako tartearekin, kalkula ezazu *regula falsi* metodoaz erroaren hurbilpena % 0.5 zehaztasun erlatiboarekin. Zein metodok du konbergentzia azkarrena?

4. Izan bedi $f(x) = e^x - x^2 + 3x - 2$ funtzioa.

   (a) Aurkitu, konbergentzia irizpidea erabiliz, iterazioen kopuru maximoa $f(x) = 0$ ekuazioa bisekzio-metodoaz ebazteko, $a = 0$ eta $b = 1$ hartuz, $\tau = 10^{-3}$ zehaztasun absolutuarekin.

   (b) Bisekzio-metodoa erabiliz, aurkitu $f(x) = 0$ ekuazioaren $p \in [0, 1]$ erroaren lehenengo hurbilpena, non $\dfrac{|p_n - p_{n-1}|}{|p_n|} < 10^{-3}$ edo $|f(p_n)| < 10^{-3}$ betetzen baita.

   (c) *Regula falsi* metodoa erabiliz, aurkitu $f(x) = 0$ ekuazioaren $p \in [0, 1]$ erroaren lehenengo hurbilpena, non $\dfrac{|p_n - p_{n-1}|}{|p_n|} < 10^{-3}$ edo $|f(p_n)| < 10^{-3}$ betetzen baita.

   (d) Zein metodok du konbergentzia azkarrena?

5. Izan bedi $f(x) = x^4 + 2x^2 - x - 3$ funtzioa.

(a) Aurkitu, konbergentzia irizpidea erabiliz, iterazioen kopuru maximoa $f(x) = 0$ ekuazioa bisekzio metodoaz ebazteko, $a = 1$ eta $b = 2$ hartuz, $\tau = 10^{-4}$ zehaztasun absolutuarekin.

(b) Bisekzio metodoa erabiliz, aurkitu $f(x) = 0$ ekuazioaren $p \in [1, 2]$ erroaren lehenengo hurbilpena, non $\dfrac{|p_n - p_{n-1}|}{|p_n|} < 10^{-2}$ edo $|f(p_n)| < 5.0 \times 10^{-2}$ betetzen baita.

(c) *Regula falsi* metodoa erabiliz, aurkitu $f(x) = 0$ ekuazioaren $p \in [1, 2]$ erroaren lehenengo hurbilpena, non $\dfrac{|p_n - p_{n-1}|}{|p_n|} < 10^{-2}$ edo $|f(p_n)| < 5.0 \times 10^{-2}$ betetzen baita.

(d) Zein metodok du konbergentzia azkarrena?

6. Zer gertatuko da bisekzio-metodoa erabiltzen badugu $f(x) = 1/(x - 2)$ funtzioarekin tarte hauetan: (a) $[3, 7]$ eta (b) $[1, 7]$.

7. Emandako tarteetan, aztertu funtzio hauek puntu finko bakar bat duten ala ez:

(a) $F(x) = 1 - x^2/4$, $[0, 1]$.

(b) $F(x) = 2^{-x}$, $[0, 1]$.

(c) $F(x) = 1/x$, $[0.5, 5.2]$.

Aintzat hartu 4.5. adibidea.

8. Aztertu puntu finkoko iterazioaren konbergentzia $F(x) = -4 + 4x - \dfrac{1}{2}x^2$ denean.

(a) Ebatzi analitikoki $F(x) = x$, eta frogatu $p = 2$ eta $p = 4$ puntu finkoak direla.

(b) Hartu $p_0 = 1.9$ eta kalkula itzazu p_1, p_2, p_3, eta balio horiei dagozkien zehaztasun absolutuak eta erlatiboak.

(c) Hartu $p_0 = 3.8$ eta kalkula itzazu p_1, p_2, p_3, eta balio horiei dagozkien zehaztasun absolutuak eta erlatiboak.

(d) 4.3. teorema erabiliz, zer ondoriozta daiteke?

9. Izan bedi $F(x) = x^2 + x - 4$. Orduan, $x = F(x)$ ekuazioaren erroak kalkulatzeko puntu finkoko iterazioa erabil dezakegu? Zergatik?

10. Puntu finkoaren metodoan, zergatik da abantaila $F'(p) \approx 0$ izatea?

11. Demagun $F, F' \in C(p - r_0, p + r_0)$, $r_0 > 0$, non $F(p) = p$, eta ez dela n existitzen $F(p_n) = p$ betetzen duenik. Froga ezazu hau betetzen bada:

$$|F'(p)| > 1,$$

p puntu finko aldaragarria dela. Hots, p_n iterazioa p-tik nahiko hurbil badago, $|p_{n+1} - p| > |p_n - p|$ dugula.

12. Demagun 4.1. korolarioaren hipotesiak betetzen direla, $F'(p) = 0$, eta $F''(p)$ existitzen dela. Froga ezazu r positibo bat existitzen dela, $r < r_0$, non $p_0 \in [p-r, p+r]$ betetzen bada, $p_n = F(p_{n-1})$ segidako gai guztiak $[p-r, p+r]$ tartean baitaude eta $\{p_n\} \to p$ konbergentzia koadratikoarekin. (*Iradokizuna*: erabili F-ren bigarren mailako Taylorren polinomioa p-ren ingurunean $F(p_n)$ hurbiltzeko).

13. Izan bedi $e^x - 3x^2 = 0$ ekuazioa; aurkitu $x = -0.5$ eta $x = 4$ puntuetatik gertuko erroak Newton-Raphson metodoaz, sei digituko zehaztasun erlatiboarekin (hots, $ze = 10^{-6}$ izanik).

14. Izan bedi $x^2 - x + 2 = 0$ ekuazioa.

    (a) Zehaztu Newton-Raphsonen $p_n = F(p_{n-1})$ iterazio-funtzioa.

    (b) Hasi $p_0 = -1.5$ puntutik, eta aurkitu $p_1,\ p_2$ eta p_3.

    (c) Zein da $|f(p_3)|$ errorea p_3 puntuan?

    (d) Zergatik gertatzen da hori?

15. Izan bedi $x^2 - x - 3 = 0$ ekuazioa.

    (a) Zehaztu Newton-Raphsonen $p_n = F(p_{n-1})$ iterazio-funtzioa.

    (b) Hasi $p_0 = 1.6$ puntutik, eta aurkitu $p_1,\ p_2,\ p_3$ eta p_4.

    (c) Zein da $|f(p_4)|$ errorea p_4 puntuan?

16. Izan bedi $(x-2)^4 = 0$ ekuazioa.

    (a) Zehaztu Newton-Raphsonen formulako $p_n = F(p_{n-1})$ iterazio-funtzioa.

    (b) Hasi $p_0 = 2.1$ puntutik, eta aurkitu $p_1,\ p_2,\ p_3$ eta p_4.

    (c) Zein da $|f(p_4)|$ errorea p_4 puntuan?

    (d) Aztertu $\{p_n\}$ segidaren konbergentzia-ordena.

17. Izan bedi $x^3 - 3x - 2 = 0$ ekuazioa.

    (a) Zehaztu Newton-Raphsonen formulako $p_n = F(p_{n-1})$ iterazio-funtzioa.

    (b) Hasi $p_0 = 2.1$ puntutik, eta aurkitu $p_1,\ p_2,\ p_3$ eta p_4.

    (c) Zein da $|f(p_4)|$ errorea p_4 puntuan?

    (d) Aztertu $\{p_n\}$ segidaren konbergentzia-ordena.

18. Izan bedi $xe^{-x} = 0$ ekuazioa.

    (a) Zehaztu Newton-Raphsonen formulako $p_n = F(p_{n-1})$ iterazio-funtzioa.

    (b) Hasi $p_0 = 0.2$ puntutik, eta aurkitu $p_1,\ p_2,\ p_3$ eta p_4. Zein da $|f(p_4)|$ errorea p_4 puntuan? Zein da $\lim_{n\to\infty} p_n$? Zein da zehaztasun absolutua p_4 puntuan?

    (c) Hasi $p_0 = 20$ puntutik, eta aurkitu $p_1,\ p_2,\ p_3$ eta p_4. Zein da $|f(p_4)|$ errorea p_4 puntuan? Zein da $\lim_{n\to\infty} p_n$? Zein da zehaztasun absolutua p_4 puntuan?

(d) Azaldu zer gertatzen den (b) eta (c) kasuetan.

19. Ebakitzailearen metodoa erabiliz, aurki itzazu ekuazio hauen p_2 eta p_3 hurbilpenak, emandako hastapen-puntuetatik hasiz:

    (a) $x^2 - 2x - 1 = 0$, non $p_0 = 2.6$ eta $p_1 = 2.5$.

    (b) $x^2 - x - 3 = 0$, non $p_0 = 1.7$ eta $p_1 = 1.67$.

    (c) $x^2 - 2x - 1 = 0$, non $p_0 = -1.5$ eta $p_1 = -1.52$.

20. Erabil dezakegu Newton-Raphsonen metodoa $x^2 - 14x + 50 = 0$ ebazteko? Zergatik?

21. Erabil dezakegu Newton-Raphsonen metodoa $x^{1/3} = 0$ ebazteko? Zergatik?

22. Erabil dezakegu Newton-Raphsonen metodoa $(x - 3)^{1/2} = 0$ ebazteko, $p_0 = 4$ hartuz? Zergatik?

23. Erabili Mullerren metodoa $x^3 - x - 2 = 0$ ekuazioaren erro bat hurbiltzeko. Hasi $p_0 = 1.0$, $p_1 = 1.2$ eta $p_2 = 1.4$ puntuetatik, eta kalkula itzazu p_3 eta p_4. Zein da $|f(p_4)|$ errorea p_4 puntuan?

24. Erabili Mullerren metodoa $4x^2 - e^x = 0$ ekuazioaren erro bat hurbiltzeko. Hasi $p_0 = 4.0$, $p_1 = 4.1$ eta $p_2 = 4.2$ puntuetatik, eta kalkula itzazu p_3 eta p_4. Zein da $|f(p_4)|$ errorea p_4 puntuan?

25. Izan bedi $xe^{0.5x} + 1.2x - 5 = 0$ ekuazioa. $f(x) = xe^{0.5x} + 1.2x - 5$ funtzioa marrazten badugu, $x = 1$ eta $x = 2$ artean erro bat dagoela ikusten dugu. Ekuazio hori $x = F(x)$ era desberdinetan berridatz dezakegu. Honako hiru aukera hauetatik, zein da zure aburuz egokiena? Eman arrazoia funtzioen deribatuak aztertuz:

    (a) $x = \dfrac{5 - xe^{0.5x}}{1.2}$;

    (b) $x = \dfrac{5}{e^{0.5x} + 1.2}$;

    (c) $x = \dfrac{5 - 1.2x}{e^{0.5x}}$.

MATLABez ebazteko problemak:

26. Metodo grafikoa erabiliz, bakartu tarteetan funtzio hauen erroak:

    (a) $f(x) = \sin(x) + 0.8\cos(x)$.

    (b) $f(x) = x^2 - 4x + 3.5 - \ln(x)$.

    (c) $f(x) = (x - 2.1)^2 - 7x\cos(x)$.

    Egiaztatu analitikoki tarte bakoitzean erro bat dagoela (hots, Bolzanoren teorema erabiliz).

27. Inplementatu bisekzio-metodoa honelako MATLABeko funtzio bat garatuz:

    (a) Lehenengo lerroa honela izan behar:
    `function [p,z,e,i]=bisekzioa(f,a,b,ztol,etol,imax)`

    (b) Sarrerak: `f`= funtzioa, `a,b`= funtzio horrek erro bat duen tarte bat, `ztol`= zehaztasunaren tolerantzia, `etol`= errorearen tolerantzia, eta `imax`= iterazio kopuru maximoa.

    (c) Irteerak: `p`= erroaren azken hurbilpena tolerantzia horiekin, `i`= erabilitako iterazio kopurua, `z`= zehaztasun absolutua, eta `e`= errorea azken iterazioan.

    (d) MATLABeko funtzio horrek, datu horiek emateaz gain, ondo eraikitako (formatu egokiak erabiliz) taula bat sortu behar du zutabe desberdinak erabiliz era honetan:

    $i \qquad a_i \qquad b_i \qquad p_i \qquad z_i \qquad e_i$

    i iterazio bakoitzerako a_i eta b_i zutabe horietan agertu behar dira erroa daukan tartearen muturrak, $p_i = (a_i + b_i)/2$ hurbilpen berria, $z_i = |p_i - p_{i-1}|$ zehaztasun absolutua eta $e_i = |f(p_i)|$ errorea.

    (e) Egiaztatu kode hori ondo dabilela, zuk aukeratuz ekuazio egoki bat non soluzio zehatza aurrez ezagutzen duzun eta bisekzio-metodoaren urrats batzuk eginda dauzkazun. Aukera itzazu tarte egoki bat, iterazio kopuru maximoa, eta zehaztasunaren eta errorearen tolerantziak.

28. Bisekzio-metodoaren kodea erabiliz, aurkitu 26. problemako funtzioen erroak.

    (i) Kalkulatu erroa 10^{-4} errore tolerantziarekin, 10^{-4} zehaztasun erlatiboarekin eta gehienez 100 iterazio erabiliz.

    (ii) Kalkula ezazu, arkatzez, kasu bakoitzean zenbat iterazio beharko dituen gehienez metodoak, zehaztasun absolutua 10^{-6} izateko. Orain, MATLABen bidez, kalkulatu metodoak erabiltzen duen benetako iterazio kopurua zehaztasun hori lortzeko. Horretarako, hartu `etol` oso txikia (esate baterako, 10^{-10}).

29. Zure kodean zati batzuk aldatuz, lortu `regula_falsi.m` fitxategi-kode bat; metodo honetan erroaren hurbilpen berria ez da lortzen $p_i = (a_i + b_i)/2$ eginez, honela baizik:

$$p_i = b_i - f(b_i)\frac{b_i - a_i}{f(b_i) - f(a_i)}.$$

    (i) Aurkitu 26. problemako funtzioen erroak M fitxategi berri hau erabiliz eta 10^{-4} errore tolerantziarekin, 10^{-4} zehaztasun erlatiboarekin eta gehienez 100 iterazio erabiliz.

    (ii) Zein du konbergentzia azkarrena? Bisekzioarena edo regula falsirena?

    (iii) Orain, MATLABen bidez, kalkulatu metodoak erabiltzen duen benetako iterazio kopurua zehaztasun absolutua 10^{-6} izateko. Horretarako, hartu `etol` oso txikia (esate baterako, 10^{-10}). Konparatu 28. problemako (ii) atalean lortutako emaitzarekin. Ondorioren bat atera dezakezu?

30. `newton.m` fitxategia lortzeko, inplementa ezazu Newtonen metodoa (4.1. algoritmoa) honelako MATLABeko funtzio bat garatuz:

    (a) Lehenengo lerroa honela izan behar:
    `function [p,ze,e,i]=newton(f,df,p0,zetol,etol,imax)`

    (b) Sarrerak: `f`= funtzioa, `df`= funtzioaren deribatua, `p0`= hasierako puntua, `zetol`= zehaztasun erlatiboaren tolerantzia, `etol`= errorearen tolerantzia eta `imax`= iterazio kopuru maximoa.

    (c) Irteerak: `p`= erroaren hurbilpen hoberena tolerantzia horiekin, `i`= iterazio kopurua, `ze`= zehaztasun erlatiboa eta `e`= errorea.

    (d) MATLABeko funtzio horrek, datu horiek emateaz gain, ondo eraikitako (formatu egokiak erabiliz) taula bat sortu behar du zutabe desberdinak erabiliz era honetan:

    $i \qquad p_i \qquad ze_i \qquad e_i$

    i iterazio bakoitzerako, p_i hurbilpenak, $ze_i = |p_i - p_{i-1}|/|p_i|$ zehaztasun erlatiboa eta $e_i = |f(p_i)|$ errorea.

    (e) Egiaztatu kode hori ondo dabilela, zuk aukeratuz ekuazio egoki bat non soluzio zehatza aurrez ezagutzen duzun eta Newtonen metodoaren urrats batzuk eginda dauzkazun. Aukera itzazu zuk hasierako puntu egoki bat, iterazio kopuru maximoa, eta zehaztasunaren eta errorearen tolerantziak.

31. Aurreko ariketa egin ondoren, ebatzi ekuazio hauek hasierako puntuetatik abiatuz eta Newton-en metodoa erabiliz:

    (a) $f(x) = 2^{-x} - x = 0$, non $p_0 = 1$.

    (b) $f(x) = x^2 - 4x + 3.5 - \ln(x) = 0$, non $p_0 = 1$.

    (c) $f(x) = (x - 2.1)^2 - 7x\cos(x) = 0$, non $p_0 = 1.5$.

    (i) Kalkulu hauek egiteko, erabili lehenengo ariketako `newton.m` fitxategia.

    (ii) Kalkulatu erroa ekuazio bakoitzean 10^{-6} errore-tolerantziarekin, 10^{-6} zehaztasun erlatiboarekin, eta gehienez 100 iterazio erabiliz.

32. Newtonen metodoari dagokion inplementazioa adibide moduan hartuz, inplementa ezazu ebakitzailearen metodoa; ikus ezazu 4.2. algoritmoaren sasikodea. Ebatzi:

    (a) $f(x) = 2^{-x} - x = 0,\ p_0 = 1, p_1 = 1.5$.

    (b) $f(x) = x^2 - 4x - 3.5 - \log(x) = 0,\ p_0 = 1, p_1 = 1.5$.

    (c) $f(x) = (x - 2.1)^2 - 7x\cos(x) = 0,\ p_0 = 1.5, p_1 = 2$.

    Konparatu emaitza horiek MATLABeko `fzero.m` funtzioaz lortutakoekin.

33. (a) Garatu sasikode bat puntu finkoaren metodorako, $x = F(x)$ ekuazioa ebazteko gai dena.

(b) Inplementatu sasikode hori MATLABeko `fpuntufinko.m` fitxategi berri bat bezala.

(c) Ebatzi $x = F(x) = 2^{-x}$ ekuazioa, $p_0 = 1$ puntutik hasiz. Zer gertatzen da? Zergatik?

34. Garatu 4.3.5. ataleko Mullerren metodoaren sasikode bat, eta, gero, inplementatu MATLABeko M fitxategi batean, non sartutako balioak `f,p0,p1,p2,etol,zetol,imax` baitira, eta ateratzen diren balioak `p,err,ze,i` (ikus ezazu 28. ariketako (b) eta (c) ataletan parametro horien esanahia). Kontuan hartu $\sqrt{b^2 - 4ac}$ zenbaki konplexua izan daitekeela, eta zein erro gorde behar dugun.

35. Erabili Mullerren metodoa $p_0 = 1.5$, $p_1 = 1.4$ eta $p_2 = 1.3$ hasierako balioak $f(x) = 1 + 2x - \tan(x)$ funtzioaren erro baten hurbilpena lortzeko, `etol`=10^{-8} errorearen tolerantziarekin, `zetol`=10^{-12} zehaztasun erlatiboarekin, eta `imax`=50 iterazio kopuru maximoarekin.

36. Konparatu Newtonen, ebakitzailearen eta Mulleren metodoen konbergentzia aurreko problemaren funtziorako, Newtonen metodorako $p_0 = 1.5$ hartuz, eta ebakitzailearen metodorako $p_0 = 1.5$ eta $p_1 = 1.4$. Hartu aurreko problemako balio berdinak `etol, zetol` eta `imax` parametroetarako.

5. kapitulua

Sistema linealak: metodo zuzenak eta iteratiboak

Bi (edo hiru) ekuazio eta bi (edo hiru) ezezagun dituen sistema bat ebaztea eskuz egin dezakegu, ordezkapenaz edo beste metodo bat erabiliz (esate baterako, Cramer-en metodoa). Sistema bat horrela ebaztea, praktikan, ezinezko bihurtzen da ekuazioen eta ezezagunen kopurua handiagoa denean.

5.1. Sistema linealen ebazpena

Notazio matrizialarekin, honela idatz daiteke ekuazio linealen sistema bat:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}.$$

Askotan, ekuazioen eta ezezagunen kopurua berdina eta handia da, n ordenako $\boldsymbol{A}$ matrize karratua ezaguna da, eta n dimentsioko $\boldsymbol{b}$ zutabe-bektorea ere bai, eta $\boldsymbol{x}$ ezezagun zutabe-bektorea n dimentsiokoa da.

Jakin badakigu $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$-ren soluzioa $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ idatz daitekeela, non $\boldsymbol{A}^{-1}$ matrizea $\boldsymbol{A}$-ren alderantzizkoa baita. Hala ere, konputazio praktikako problema gehienetan, ez da beharrezkoa, ezta gomendagarri ere, $\boldsymbol{A}^{-1}$ kalkulatzea. Adibide erakusgarri moduan, ekuazio bateko eta ezezagun bateko ekuazio hau hartuko dugu:

$$7x = 21.$$

Sistema hori ebazteko modu hoberena zatiketa da:

$$x = \frac{21}{7} = 3.$$

Alderantzizko matrizea erabiltzeak honetara eramaten gaitu:

$$x = 7^{-1} \times 21 = 0.142857 \times 21 = 2.99997.$$

Alderantzizkoak aritmetika gehiago behar du (zatiketa bat eta biderketa bat, zatiketa bat bakarrik izan beharrean) eta emaitzaren zehaztasuna txikiagoa da. Antzeko zerbait gertatzen da sistema handiagoetan. Ondorioz, ekuazio-sistemen ebazpen zuzenean zentratuko gara, alderantzizkoaren kalkulua egin beharrean.

MATLABeko atzeranzko barra "\" erabiliko dugu $\boldsymbol{AX} = \boldsymbol{B}$ sistema ebazteko, non $\boldsymbol{A}$-ren lerroen kopurua eta $\boldsymbol{B}$-rena berdinak baitira. Orduan, sistema horren soluzioa `X=A\B` da eta horrek $\boldsymbol{X} = \boldsymbol{A}^{-1}\boldsymbol{B}$ ematen du. Alegia, *ezker zatiketa* da.

MATLABeko aurreranzko barra "/" erabiliko dugu $\boldsymbol{XA} = \boldsymbol{B}$ sistema ebazteko, non $\boldsymbol{A}$-ren zutabeen kopurua eta $\boldsymbol{B}$-rena berdinak baitira. Orduan, sistema horren soluzioa `X=B/A` da eta horrek $\boldsymbol{X} = \boldsymbol{B}\boldsymbol{A}^{-1}$ ematen du. Alegia, *eskuin zatiketa* da.

Notazio hori erabiltzen da $\boldsymbol{A}$ karratua ez denean ere; hots, ekuazioen kopurua eta ezezagunen kopurua desberdinak izan arren. Oraingoz, matrize karratuko sistemekin arituko gara.

5.2. 3x3 adibide bat

Sistema lineal baten ebazte-algoritmo bat erakutsiko dugu. Izan bedi $\boldsymbol{Ax} = \boldsymbol{b}$ ekuazio-sistema hau:

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}.$$

Adierazpen hori ekuazio hauen sistemari dagokio:

$$\begin{aligned} E_1: &\quad 10x_1 - 7x_2 = 7, \\ E_2: &\quad -3x_1 + 2x_2 + 6x_3 = 4, \\ E_3: &\quad 5x_1 - x_2 + 5x_3 = 6. \end{aligned}$$

Algoritmoaren lehenengo urratsak lehenengo ekuazioa erabiltzen du beste ekuazioetako x_1 ezabatzeko. Hori lortzeko, $E_2 + 0.3 \cdot E_1$ eta $E_3 - 0.5 \cdot E_1$ egiten da. E_1 ekuazioko x_1-en 10 koefizienteari *pibot* deritzogu, eta *biderkatzaile* deritze beste ekuazioetako x_1-en koefizienteak 10 pibotaz zatituz lortutako -0.3 eta 0.5 kopuruei. Lehenengo urratsak honela aldatzen ditu ekuazioak:

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 6.1 \\ 2.5 \end{bmatrix}.$$

Ohartu hau betetzen dela:

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.3 & 1 & 0 \\ 0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{bmatrix} = \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix}.$$

Bigarren urratsean, bigarren ekuazioa erabil dezakegu hirugarren ekuazioko x_2 ezabatzeko. Baina, bigarren pibota, bigarren ekuazioko x_2-ren koefizientea, -0.1 da, eta hori hurrengo koefizientea baino txikiagoa da. Ondorioz, azken bi ekuazioak trukatzen dira. Horri *pibotatzea* deritzogu. Adibide honetan, egia esan, hori ez da beharrezkoa, ezen ez baitago biribiltze-errorerik; baina, oro har, erabakigarria da. Beraz, pibotatu eta gero, zera dugu:

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & -0.1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 2.5 \\ 6.1 \end{bmatrix}.$$

Orain, bigarren pibota 2.5 da, eta erabil dezakegu hirugarren ekuazioko x_2 ezabatzeko. Hori lortuko dugu hirugarren ekuazioari 0.04 bider bigarrena batuz (hots, -0.04 biderkatzailea da), hau lortuz:

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 2.5 \\ 6.2 \end{bmatrix}.$$

Orain, azken ekuazioa hau da:

$$6.2x_3 = 6.2$$

eta, ondorioz, $x_3 = 1$. Balio hori bigarren ekuazioan ordezka dezakegu:

$$2.5x_2 + (5) \cdot (1) = 2.5$$

eta x_2 askatuz, $x_2 = -1$ dugu. Azkenik, lehenengo ekuazioan ordezkatzen ditugu x_2-ren eta x_3-ren balioak:

$$10x_1 + (-7) \cdot (-1) = 7$$

eta x_1 askatuz, $x_1 = 0$ dugu. Emaitza hau da:

$$\boldsymbol{x} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}.$$

Emaitza hau erraz egiazta daiteke jatorrizko ekuazioak erabiliz:

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}.$$

Ohartu $\boldsymbol{L}_1\boldsymbol{L}_2 = \boldsymbol{L}$ betetzen dela, non

$$\boldsymbol{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & 0 & 1 \end{bmatrix}, \ \boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.04 & 1 \end{bmatrix}, \ \boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{bmatrix}.$$

Bestalde, $\boldsymbol{A}$ matrizean bigarren eta hirugarren lerroak trukatu (permutatu) ditugu; hori $\boldsymbol{P}$ *permutazio-matrize* batez aurrebiderkatuz lortzen da, alegia:

$$\boldsymbol{PA} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} = \begin{bmatrix} 10 & -7 & 0 \\ 5 & -1 & 5 \\ -3 & 2 & 6 \end{bmatrix}.$$

Algoritmo osoa era trinko batean adieraz daiteke, notazio matriziala erabiliz. Adibide honetarako hau dugu:

$$\boldsymbol{LU} = \boldsymbol{PA},$$

non

$$\boldsymbol{U} = \begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{bmatrix}.$$

Hau da, $\boldsymbol{L}$ matrizeak ezabatze-prozesuko biderkatzaileak gordetzen ditu bateko diagonal azpian, $\boldsymbol{U}$ matrizea azken koefiziente-matrizea da, eta $\boldsymbol{P}$ matrizeak pibotatze-prozesua deskribatzen du.

5.3. Permutazio- eta triangelu-matrizeak

Permutazio-matrize bat identitate-matrize bat da, lerroak eta zutabeak trukatuta dauzkana. Hain zuzen ere, 1 bakarra dauka lerro eta zutabe bakoitzean, eta zero dira beste gaiak. Adibidez,

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

$\boldsymbol{A}$ matrizea ezkerretik biderkatzen badugu $\boldsymbol{P}$ permutazio-matrize batez (hots, $\boldsymbol{PA}$), $\boldsymbol{A}$ matrizearen lerroak permutatzen dira. Aldiz, eskuinetik biderkatzen badugu (hots, $\boldsymbol{AP}$), orduan, $\boldsymbol{A}$ matrizearen zutabeak permutatzen ditu.

MATLABek badauka `p` permutazio-bektore bat matrize baten lerroak edo zutabeak berrordenatzeko. Goiko $\boldsymbol{P}$ matrizearen kasuan, `p` hau da:

```
p=[4 1  3 2]
```

Orduan, `P*A` eta `A(p,:)` berdinak dira. Lortutako matrizean 1. lerroa $\boldsymbol{A}$-ren 4.a izango da, 2. lerroa $\boldsymbol{A}$-ren 1.a, 3. lerroa $\boldsymbol{A}$-ren 3.a, eta 4. lerroa $\boldsymbol{A}$-ren 2.a. Halaber, `A*P` eta `A(:,p)` berdinak dira, biek $\boldsymbol{A}$-ren zutabeen permutazio berdina sortzen dute.

$\boldsymbol{Px} = \boldsymbol{b}$ ekuazio-sistemetarako soluzioa kalkulatzeko, hau bakarrik egin behar dugu:

$$\boldsymbol{x} = \boldsymbol{P}^T\boldsymbol{b},$$

zeren $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$ baita; hots, $\boldsymbol{P}$ *ortogonala* da.

Matrize goi-triangeluar batean, zeroak dira diagonal nagusiaren azpiko gai guztiak. *Matrize behe-triangeluar* batean, zeroak dira diagonal nagusiaren goiko gai guztiak. Adibidez, aurreko atalean, $\boldsymbol{L}$ matrizea behe-triangeluarra da eta $\boldsymbol{U}$ matrizea goi-triangeluarra.

Sistema bateko matrizea triangeluarra bada, erraz ebazten da. Izan bedi $\boldsymbol{Ux} = \boldsymbol{b}$ sistema goi-triangeluar hau:

$$\begin{aligned} u_{11}x_1 + u_{12}x_2 + \ldots + u_{1,n-1}x_{n-1} + u_{1n}x_n &= b_1 \\ u_{22}x_2 + \ldots + u_{2,n-1}x_{n-1} + u_{2n}x_n &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots &\quad \ldots\ldots \\ u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n &= b_{n-1} \\ u_{nn}x_n &= b_n. \end{aligned} \tag{5.1}$$

Beraz, soluzioa lortzeko, *atzeranzko ordezkatze-prozesu* honi jarraitu behar diogu (hots, behetik gora):

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}} \\ x_{n-1} &= \frac{b_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}}. \end{aligned} \tag{5.2}$$

Eta, honela, ondoz ondo jardunez, zera dugu:

$$x_i = \frac{b_i - u_{in}x_n - u_{i,n-1}x_{n-1} - \ldots - u_{i,i+1}x_{i+1}}{u_{ii}} = \frac{b_i - \displaystyle\sum_{j=i+1}^{n} u_{ij}x_j}{u_{ii}}, \tag{5.3}$$

non $i = n-1, n-2, \ldots, 3, 2, 1$.

Eragiketa horiek, MATLABen bitartez, honela egin ditzakegu:

```
x = zeros(n,1);
for k = n:-1:1
   x(k) = b(k)/U(k,k);
   i = (1:k-1)';
   b(i) = b(i)-x(k)*U(i,k);
end
```

Bestalde, demagun $\boldsymbol{Lx} = \boldsymbol{b}$ sistema behe-triangeluar hau dugula, eta $\boldsymbol{L}$-ren diagonaleko gaiak batak direla:

$$\begin{aligned} x_1 &= b_1 \\ l_{21}x_1 + x_2 &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots &\quad \ldots\ldots \\ l_{n1}x_1 + l_{n2}x_2 + \ldots + l_{n,n-1}x_{n-1} + x_n &= b_n. \end{aligned} \tag{5.4}$$

Soluzioa lortzeko, *aurreranzko ordezkatze-prozesu* honi jarraitu behar diogu (hots, goitik behera):

$$\begin{aligned} x_1 &= b_1 \\ x_2 &= b_2 - l_{2,1}x_1. \end{aligned} \tag{5.5}$$

Eta, honela, ondoz ondo jardunez, zera dugu:

$$x_i = b_i - l_{i1}x_1 - l_{i,2}x_2 - \ldots - l_{i,i-1}x_{i-1} = b_i - \sum_{j=1}^{i-1} l_{ij}x_j, \tag{5.6}$$

non $i = 2, 3, \ldots, n-1, n$.

Ariketa moduan, sortu MATLABen kode bat eragiketa horiek egiteko.

5.4. Pibotatzearen beharra

U matrizearen diagonaleko gaiei *pibotak* deritzegu. Goiko adibidean, pibotak dira 10, 2.5 eta 6.2. Biderkatzaileen kalkuluak eta atzeranzko ordezkapenak pibotekin zatitu behar dira. Ondorioz, piboten bat zero bada, ezin dugu burutu algoritmoa. Intuizioz, badakigu ez dela ideia ona kalkulua aurrera eramatea piboten bat ia zero bada. Hori erakusteko, pixka bat aldatuko dugu gure goiko adibidea:

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2.099 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 3.901 \\ 6 \end{bmatrix}.$$

Matrizeko $(2,2)$ gaia aldatu egin da 2.000 izatetik 2.099 izatera, eta berdintzaren eskuineko aldea aldatu egin dugu emaitza berbera izateko moduan, $(0,-1,1)^T$. Demagun bost zifra esanguratsutako puntu higikor hamartarra duen makina batean kalkulatu dugula soluzioa.

Ezabapenaren lehenengo urratsak hau ematen du:

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & -0.001 & 6 \\ 0 & 2.5 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 6.001 \\ 2.5 \end{bmatrix}.$$

Orain, $(2,2)$ gaia nahiko txikia da matrizeko beste gaiekin konparatzen badugu. Hala ere, trukerik gabe beteko dugu ezabapena. Hurrengo urratsean, hirugarren ekuazioari $2.5 \cdot 10^3$ bider bigarrena batuko diogu, honela:

$$(5 + (2.5 \cdot 10^3) \cdot 6)x_3 = 2.5 + (2.5 \cdot 10^3) \cdot 6.001.$$

Berdintzaren eskuin aldeko gaian $6.001 \cdot 2.5 \cdot 10^3 = 1.50025 \cdot 10^4$ dugu. Emaitza $1.50025 \cdot 10^4$ da, eta hori ezin da adierazi zehatz-mehatz gure puntu higikorraren zenbaki-sistema

hipotetikoan. Hura $1.5003 \cdot 10^4$ra biribildu beharko dugu. Gero, emaitza hori 2.5 zenbakiari gehituko diogu $1.50055 \cdot 10^4$ lortuz, eta berriro biribilduz $1.5006 \cdot 10^4$ lortzen da. Ondorioz, gure makina hipotetikoan, azken ekuazioa hau bihurtzen da:

$$1.5005 \cdot 10^4 x_3 = 1.5006 \cdot 10^4.$$

Beraz, honekin hasten da atzeranzko ordezkapena:

$$x_3 = \frac{1.5006 \cdot 10^4}{1.5005 \cdot 10^4} = 1.0001 \quad \text{biribilduz.}$$

Emaitza zehatza $x_3 = 1$ denez, erroreak ez dirudi serioegia. Zoritxarrez, x_2 ekuazio honetatik aurkitu behar dugu:

$$-0.001x_2 + 6 \cdot (1.0001) = 6.001,$$

eta horrek hau ematen du:

$$x_2 = \frac{4 \cdot 10^{-4}}{-1.0 \cdot 10^{-3}} = -0.4.$$

Azkenik, x_1 lehenengo ekuazioaz honela zehazten da:

$$10x_1 + (-7) \cdot (-0.4) = 7,$$

hau lortuz:

$$x_1 = 0.42.$$

Alegia, $(0, -1, 1)^T$ lortu beharrean $(0.42, -0.4, 1.0001)^T$ lortu dugu.

Non dago errorea? Ez dago «biribiltze-errorearen pilaketa» sortuta milaka eragiketa aritmetiko eginez. Matrizea ez da ia singularra. Zailtasuna ezabapeneko bigarren urratsean pibot txiki bat hartzetik dator. Ondorioz, biderkatzailea $2.5 \cdot 10^3$ da, eta azken ekuazioak dauzkan koefizienteak ia 10^3 bider jatorrizko problemaren koefizienteak dira.

Ariketa moduan, aztertu zer gertatuko litzatekeen bigarren urratsean trukatuko bagenitu bigarren eta hirugarren ekuazioak.

Biderkatzaile guztiak balio absolutuan 1 edo txikiagoak badira, kalkulatutako soluzioa zuzena dela froga daiteke. Biderkatzaileen balio absolutuak 1 baino handiagoak ez izateko, *pibotatze partziala* izendatutako prozesua erabil dezakegu.

5.4.1. Pibotatze partziala

Ezabatze-prozesuan i-garren lerroan bagaude, i-garren zutabean geratzen diren gaien artean (hots, a_{ji}, non $j = i, i+1, \ldots, n$ baita) balio absolutu handieneko gaia bilatu behar da. Gai hori p-garrena bada, i-garren eta p-garren lerroak trukatuko ditugu. Alegia, hau badugu:

$$\max_{j \geq i} \{|a_{ji}|\} = |a_{pi}|,$$

i-garren eta p-garren lerroak trukatuko ditugu. Truke berdinak egiten dira berdintzaren eskuineko b bektorean. Hots, b_p eta b_i ere trukatzen dira.

5.5. LU faktorizazioa

Oro har, ezabatze gaussiarrak bi etapa ditu: *aurreranzko ebazpena* eta *atzeranzko ebazpena*. Aurreranzko ebazpenak $n-1$ urrats ditu, aurreko atalean ikusi dugun bezala. i-garren urratsean, i-garren ezezaguna ezabatzeko i-garren ekuazioaren multiploak kentzen dizkiegu gainerako ekuazioei. Baldin x_i-ren koefizientea «txikia» bada, gomendagarria da (5.2)-(5.3) prozesua egin baino lehen ekuazioak trukatzea. Ezabapen-urratsak berdintzaren eskuinaldeko gaiei batera aplika diezazkiekegu, edo trukeak eta biderkatzaileak gorde ditzakegu eta geroago aplikatu eskuinaldeko gaiei. Azken hori ikusiko dugu, hain zuzen ere, jarraian. Azkenik, sistemaren atzeranzko ebazpena (5.5)-(5.6) adierazpenak erabiliz lortuko dugu.

Izan bedi $\boldsymbol{P}_i$, $i = 1, \ldots, n-1$, ezabapenaren i-garren urratsean erabilitako permutazio-matrizea. Izan bedi $\boldsymbol{M}_i$ i-garren urratsean biderkatzaileen negatiboak diagonalaren azpian sartuz lortutako matrize behe-triangeluar unitate bat; horrelako matrizeei *ezabapen-matrizeak* deritzegu. Izan bedi $\boldsymbol{U}$ ezabapenaren $n-1$ urratsak egin ondoren lortutako azken matrize goi-triangeluarra. Prozesu osoa ekuazio batez deskriba daiteke:

$$\boldsymbol{M}_{n-1}\boldsymbol{P}_{n-1}\ldots\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}. \tag{5.7}$$

Jarraian ikusiko dugu $\boldsymbol{A}$ matrizea ez-singularra bada, beti $\boldsymbol{P}$ permutazio-matrize bat aurki dezakegula $\boldsymbol{PA} = \boldsymbol{LU}$ idatzi ahal izateko moduan.

Jo dezagun $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$ matrizea eta bi permutazio-matrize, $\boldsymbol{P}_1, \boldsymbol{P}_2$, eta bi ezabapen-matrize, $\boldsymbol{M}_1, \boldsymbol{M}_2$, eraiki ditugula hau gertatzeko moduan:

$$\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}.$$

$\boldsymbol{P}_i$ ortogonala denez, badakigu $\boldsymbol{P}_i^T\boldsymbol{P}_i = \mathbb{1}$. Kasu honetan, gainera, $\boldsymbol{P}_i$ hauek simetrikoak direnez, $\boldsymbol{P}_i^T = \boldsymbol{P}_i$ dugu. Beraz, $\boldsymbol{P}_2\boldsymbol{P}_2 = \mathbb{1}$ dugu. Orduan, $\boldsymbol{P}_2\boldsymbol{P}_2$ adierazpena $\boldsymbol{M}_1$ eta $\boldsymbol{P}_1$ artean sar dezakegu:

$$\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{M}_2\widetilde{\boldsymbol{M}}_1\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U},$$

non $\widetilde{\boldsymbol{M}}_1$ berrordenatutako ezabapen-matrize bat baita; alegia, $\boldsymbol{M}_1$, baina simetrikoki trukatutako lerroekin eta zutabeekin:

$$\widetilde{\boldsymbol{M}}_1 = \boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2.$$

Adibidez, hau badugu:

$$\boldsymbol{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{M}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix},$$

zera lortuko dugu:

$$\widetilde{\boldsymbol{M}}_1 = \boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ -m_{31} & 1 & 0 \\ -m_{21} & 0 & 1 \end{bmatrix}.$$

Demagun orain $\boldsymbol{A} \in \mathbb{R}^{4\times 4}$ matrizea eta bi permutazio-matrize, $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3$, eta bi ezabapen-matrize, $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3$, eraiki ditugula hau gertatzeko moduan:

$$\boldsymbol{M}_3\boldsymbol{P}_3\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}.$$

Orduan, honela eraldatuko dugu, ezkerretik eskuinera:

$$\boldsymbol{M}_3\boldsymbol{P}_3\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{M}_3(\boldsymbol{P}_3\boldsymbol{M}_2\boldsymbol{P}_3)\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{M}_3\widetilde{\boldsymbol{M}}_2\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A},$$

non $\widetilde{\boldsymbol{M}}_2 = \boldsymbol{P}_3\boldsymbol{M}_2\boldsymbol{P}_3$ definitzen baita. Eta gero, hau eginez:

$$\boldsymbol{M}_3\widetilde{\boldsymbol{M}}_2\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{M}_3\widetilde{\boldsymbol{M}}_2(\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2\boldsymbol{P}_3)\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{M}_3\widetilde{\boldsymbol{M}}_2\widetilde{\boldsymbol{M}}_1\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U},$$

non $\widetilde{\boldsymbol{M}}_1 = \boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2\boldsymbol{P}_3$.

Horrela ere aplika diezaiokegu kasu orokorrean (5.7) adierazpenari, hots:

$$\boldsymbol{M}_{n-1}\widetilde{\boldsymbol{M}}_{n-2}\ldots\widetilde{\boldsymbol{M}}_1\boldsymbol{P}_{n-1}\ldots\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}, \tag{5.8}$$

non

$$\widetilde{\boldsymbol{M}}_k = \boldsymbol{P}_{n-1}\ldots\boldsymbol{P}_{k+1}\boldsymbol{M}_k\boldsymbol{P}_{k+1}\ldots\boldsymbol{P}_{n-1}, \quad k = 1,\ldots,n-2. \tag{5.9}$$

Orain, (5.8) adierazpenetik hau ondoriozta dezakegu:

$$\boldsymbol{L}_1\boldsymbol{L}_2\ldots\boldsymbol{L}_{n-1}\boldsymbol{U} = \boldsymbol{P}_{n-1}\ldots\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A}.$$

non $\boldsymbol{L}_k = \widetilde{\boldsymbol{M}}_k^{-1}$ baita, eta hori kalkulatzen da $\widetilde{\boldsymbol{M}}_k$ matrizeko diagonalaren azpiko biderkatzaileen zeinuak aldatuz, eta gero permutazioak aplikatuz (5.9) berdintzak adierazten duen bezala. Beraz, hau badugu:

$$\begin{aligned}\boldsymbol{L} &= \boldsymbol{L}_1\boldsymbol{L}_2\ldots\boldsymbol{L}_{n-1}\\ \boldsymbol{P} &= \boldsymbol{P}_{n-1}\ldots\boldsymbol{P}_2\boldsymbol{P}_1,\end{aligned}$$

orduan

$$\boldsymbol{L}\boldsymbol{U} = \boldsymbol{P}\boldsymbol{A}.$$

Ondorioz, $\boldsymbol{L}$ ezabapenean erabilitako biderkatzaile guztiak gordetzen ditu, eta $\boldsymbol{P}$ permutazio-matrizeak lerro-truke guztiak gordetzen ditu.

Aurreko atalaren adibiderako, hau dugu:

$$\boldsymbol{A} = \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix},$$

Hauek dira ezabapenean definitutako matrizeak:

$$\boldsymbol{P}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{M}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ -0.5 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{M}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.04 & 1 \end{bmatrix}.$$

Beraz,

$$\widetilde{\boldsymbol{M}}_1 = \boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.3 & 0 & 1 \end{bmatrix}.$$

Hauek dira horiei dagozkien $\boldsymbol{L}$ matrizeak:

$$\boldsymbol{L}_1 = \widetilde{\boldsymbol{M}}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{L}_2 = \boldsymbol{M}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.04 & 1 \end{bmatrix}, \quad \boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{bmatrix}.$$

$\boldsymbol{LU} = \boldsymbol{PA}$ erlazioari $\boldsymbol{A}$-ren *LU faktorizazioa* (edo *deskonposizio triangeluarra*) deritzo. Egia esan, *LU* faktorizazioa ezabatze gaussiarra da, notazio matrizialarekin adierazita.

Faktorizazio horrekin, ekuazio-sistema orokor baterako hau dugu:

$$\boldsymbol{Ax} = \boldsymbol{b} \quad \Longleftrightarrow \quad \boldsymbol{PAx} = \boldsymbol{Pb}, \tag{5.10}$$

beraz, $\boldsymbol{LUx} = \boldsymbol{Pb}$ sistema sistema triangeluarren bikote hau bihurtzen da:

$$\begin{cases} \boldsymbol{Ly} &= \boldsymbol{Pb} \\ \boldsymbol{Ux} &= \boldsymbol{y}. \end{cases} \tag{5.11}$$

5.1. teorema. *$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizeak $\boldsymbol{LU}$ faktorizazio bat du, baldin* $\det(\boldsymbol{A}(1:k, 1:k)) \neq 0$ *bada, $k = 1 : n-1$ guztietarako. Baldin $\boldsymbol{LU}$ faktorizazioa existitzen bada eta $\boldsymbol{A}$ ez bada singularra, orduan $\boldsymbol{LU}$ faktorizazioa bakarra da eta* $\det(\boldsymbol{A}) = u_{11} \cdot \ldots \cdot u_{nn}$ *dugu.*

Frogantza. Demagun $k-1$ urratsetan LU metodoa erabili dugula eta $\boldsymbol{A}^{(k-1)}$ lortu dugula. Kontuan izan $a_{kk}^{(k-1)}$ gaia k. pibota dela. Ezabatze gaussiarrengatik

$$\det(A(1:k, 1:k)) = a_{11}^{(k-1)} \cdot \ldots \cdot a_{kk}^{(k-1)}$$

izango dugu. Beraz, $A(1:k, 1:k)$ ez bada singularra, $a_{kk}^{(k-1)}$ ez da zero izango, hots $\boldsymbol{A}$-ren $\boldsymbol{LU}$ faktorizazio bat existitzen da.

Bestalde, demagun bi faktorizazio daudela, $\boldsymbol{A}$ matrize ez-singularrerako, $\boldsymbol{A} = \boldsymbol{L}_1\boldsymbol{U}_1$ eta $\boldsymbol{A} = \boldsymbol{L}_2\boldsymbol{U}_2$, orduan $\boldsymbol{L}_2^{-1}\boldsymbol{L}_1 = \boldsymbol{U}_2\boldsymbol{U}_1^{-1}$ dugu. $\boldsymbol{L}_2^{-1}\boldsymbol{L}_1$ unitate behe-traingeluarra denez eta $\boldsymbol{U}_2\boldsymbol{U}_1^{-1}$ goi-triangeluarra, biak identitatea izan behar dira berdinak izateko. Beraz, $\boldsymbol{L}_1 = \boldsymbol{L}_2$ eta $\boldsymbol{U}_1 = \boldsymbol{U}_2$.

Azkenik, $\boldsymbol{A} = \boldsymbol{LU}$ denez, orduan,

$$\det(\boldsymbol{A}) = \det(\boldsymbol{LU}) = \det(\boldsymbol{L})\det(\boldsymbol{U}) = \det(\boldsymbol{U}) = u_{11} \cdot \ldots \cdot u_{nn}. \quad \square$$

5.2. teorema. *$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizerako* $\det(\boldsymbol{A}(1:k, 1:k)) \neq 0$*, $k = 1 : n$ guztietarako, badugu, orduan $\boldsymbol{L}$ eta $\boldsymbol{M}$ matrize behe-triangeluar bakarrak eta $\boldsymbol{D}$ matrize diagonal bakar bat existitzen dira, non $\boldsymbol{A} = \boldsymbol{LDM}^T$ betetzen baita.*

Frogantza. Aurreko teoremaren emaitzak erabiliz $\boldsymbol{A} = \boldsymbol{LU}$ deskonposizio bakarra existitzen da, eta $d_i = u_{ii}$, $i = 1 : n$, $\boldsymbol{D}$-ren diagonaleko gaiak dira. $\boldsymbol{A}$ singularra ez denez, $\boldsymbol{D}$ ere ez-singularra da eta $\boldsymbol{M}^T = \boldsymbol{D}^{-1}\boldsymbol{U}$ unitate-matrize goi-triangeluarra da. Ondorioz, $\boldsymbol{A} = \boldsymbol{LU} = \boldsymbol{LD}(\boldsymbol{D}^{-1}\boldsymbol{U}) = \boldsymbol{LDM}^T$ eta bakarra da. □

5.3. teorema. *Baldin $\boldsymbol{A}$ matrize simetriko ez-singular baten deskonposizioa $\boldsymbol{A} = \boldsymbol{LDM}^T$ bada, $\boldsymbol{M} = \boldsymbol{L}$ da.*

Frogantza:

$$\boldsymbol{A}(\boldsymbol{M}^{-1})^T = \boldsymbol{LD} \quad \Rightarrow \quad \boldsymbol{M}^{-1}\boldsymbol{A}(\boldsymbol{M}^{-1})^T = \boldsymbol{M}^{-1}\boldsymbol{LD}.$$

Azken matrizea simetrikoa eta behe-triangeluarra da, eta, beraz, diagonala. $\boldsymbol{D}$ ez-singularra denez, $\boldsymbol{M}^{-1}\boldsymbol{L}$ unitate-matrize behe-triangeluarra da eta, simetrikoa izateagatik, $\boldsymbol{M}^{-1}\boldsymbol{L} = \mathbb{1}$. □

5.5.1. Pibotatze baztergarria

Matrize batzuetarako, ez da beharrezkoa pibotatzea.

5.1. Definizioa. $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *matrizea* ***hertsiki diagonal menperatzailea*** *dela esango dugu, hau betetzen bada:*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad i = 1, \ldots, n. \tag{5.12}$$

[15] liburuan honako teorema hau frogatzen da:

5.4. teorema. *$\boldsymbol{A}^T$ diagonal menperatzailea bada, $\boldsymbol{A}$ matrizeak LU faktorizazio bat du eta $|l_{ij}| \leq 1$.*

Alegia, $\boldsymbol{A}^T$ diagonal menperatzailea bada, aurretiko $\boldsymbol{LU} = \boldsymbol{PA}$ faktorizazioa eginez gero, $\boldsymbol{P} = \mathbb{1}$ izango da.

5.6. Matematikako problema baten baldintza

Hitz gutxitan, problema baten *baldintza* neurri bat da problemako datuen aldaketekiko soluzio zehatzaren sentikortasuna adierazteko. Ideia hori kuantifikatzeko, demagun problema hori datuen d multzo batek definitua dela. Izan bedi $s(d)$ problemaren *soluzio zehatza* d

datu horietarako. d datuetan aldaketa txikiek $s(d)$-ko aldaketa txikietara eramaten badute, d datuetarako problema *ondo baldintzatua* dela esango dugu. Aldiz, d-ren aldaketa txikiek $s(d)$-ko aldaketa handietara eramaten badute, d datuetarako problema *txarto baldintzatua* dela esango dugu. Demagun d_1 eta d_2 gerta litezkeen bi datu multzo direla. Problemaren *baldintza* (edo *baldintzazko zenbakia*) honelako ratioen maximoa da, $\|d_1 - d_2\|$ txikia denean:

$$\frac{\|s(d_1) - s(d_2)\|}{\|d_1 - d_2\|}. \tag{5.13}$$

Argi izan behar dugu problema baten baldintza *propietate matematikoa* dela, eta askea dela kalkulu- eta biribiltze-errorearekiko.

Jarraian, problema baten baldintzaren erakusketa erraz bat ikusiko dugu. Izan bedi polinomio honen erroak kalkulatzeko problema:

$$(x-1)^4 = 0, \tag{5.14}$$

haren lau erroak 1 dira, hain zuzen. Demagun aldaketa txiki bat (esate baterako, 10^{-8}) egina dela (5.14) berdintzaren eskuinaldean; orain, hau da ebatzi behar dugun ekuazioa:

$$(x-1)^4 = 10^{-8}, \tag{5.15}$$

Erro zehatza $1 + 10^{-2}$ da. Beraz, zera dugu kasu honetan:

$$\frac{\|s(d_1) - s(d_2)\|}{\|d_1 - d_2\|} = \frac{|1 - (1 + 10^{-2})|}{|0 - 10^{-8}|} = \frac{10^{-2}}{10^{-8}} = 10^6.$$

Ikusi dugun bezala, datuen aldaketa txiki batek, 10^{-8}, soluzioaren aldaketa handi bat sortzen du, 10^{-2}, zeren hori 10^6 bider handiagoa baita datuen aldaketa baino. Alegia, ratio hori askoz handiagoa da 1 baino. Beraz, problema hori *txarto baldintzatua* da. Kontuan izan propietate horrek ez duela zerikusirik erabilitako kalkulu-metodoarekin.

5.7. Matrizeen normak

Matrize-norma bat, bektoreena bezala, $\|\cdot\|$ notazioaz adierazten da, eta, bektore-normak bezala, hiru propietate hauek betetzen ditu:

(i) $\|\boldsymbol{A}\| > 0, \quad \forall \boldsymbol{A} \neq \boldsymbol{0}$;

(ii) $\|c\boldsymbol{A}\| = |c| \cdot \|\boldsymbol{A}\|, \quad c$ eskalar guztietarako;

(iii) $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$.

Matrize-norma baterako, erabilgarria da laugarren propietate hau ere, *trinkotasun* izendatua:

(iv) $\|\boldsymbol{AB}\| \leq \|\boldsymbol{A}\| \cdot \|\boldsymbol{B}\|$.

Jarraian, edozein bektore-normari elkarturiko matrize-norma definituko dugu. Intuitiboki, $\boldsymbol{A}$ matrizeak norma «handia» izan beharko luke, $\boldsymbol{Ax}$ bektorearen norma handia balitz $\boldsymbol{x}$ bektorearen normarekiko.

5.2. Definizioa. *Izan bitez $\boldsymbol{A}$ matrize bat eta $\|\cdot\|$ bektore-norma bat. $\|\boldsymbol{A}\|$* ***eragindako matrize-norma*** *honela definitzen da:*

$$\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|}. \tag{5.16}$$

$\boldsymbol{x}$-ren tamainaren menpe ez izateko, hau da $\|\boldsymbol{A}\|$-ren beste definizio baliokide bat:

$$\|\boldsymbol{A}\| = \max_{\|\boldsymbol{u}\|=1} \|\boldsymbol{Au}\|.$$

$\|\boldsymbol{A}\|$-ren behe-borne bat lor dezakegu $\boldsymbol{x}$ bektore ezagun baterako, $\|\boldsymbol{Ax}\|$-ren zatidura $\|\boldsymbol{x}\|$-rekin kalkulatuz, hots:

$$\|\boldsymbol{A}\| \geq \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|}. \tag{5.17}$$

Hauek dira bektoreen bat-, bi- eta infinitu-normek eragindako matrize-normak:

- $\|\boldsymbol{A}\|_1 = \max_j \|\boldsymbol{A}_{:,j}\|_1$ (zutabe guztien bat-normetako maximoa)
- $\|\boldsymbol{A}\|_2 = \sigma_1(A)$ (balio singular handiena).
- $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ simetrikoa bada, $\|\boldsymbol{A}\|_2 = \max_{1\leq i\leq n} |\lambda_i|$ (λ_i, $1 \leq i \leq n$, $\boldsymbol{A}$-ren autobalioak dira).
- $\|\boldsymbol{A}\|_\infty = \max_i \|\boldsymbol{A}_{i,:}\|_1$ (lerro guztien bat-normetako maximoa)

5.3. Definizioa. *$\|\cdot\|_b$ bektore-norma bat eta $\|\cdot\|_m$ matrize-norma bat* ***bateragarriak*** *direla esaten da, $\boldsymbol{A}$ eta $\boldsymbol{x}$ guztietarako hau betetzen bada:*

$$\|\boldsymbol{Ax}\|_b \leq \|\boldsymbol{A}\|_m \|\boldsymbol{x}\|_b.$$

Bektore-norma bat eta berak eragindako matrize-norma beti dira bateragarriak; ikus (5.17).

Bektore-norma batek ez eragindako matrize-norma garrantzitsu bat *Frobenius*en norma da; hori honela definitzen da $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrize baterako:

$$\|\boldsymbol{A}\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2},$$

eta froga daiteke hau egiaztatzen duela:

$$\|\boldsymbol{A}\|_F^2 = \mathrm{tr}(\boldsymbol{A}^T\boldsymbol{A}),$$

$\mathrm{tr}(\boldsymbol{B})$-k adierazten du $\boldsymbol{B}$-ren traza, diagonaleko gaien batura.

Frobeniusen norma bateragarria da bektore-norma euklidearrarekin; hots, $\boldsymbol{A}$ eta $\boldsymbol{x}$ guztietarako,

$$\|\boldsymbol{A}\boldsymbol{x}\|_2 \leq \|\boldsymbol{A}\|_F\|\boldsymbol{x}\|_2.$$

Bektore edo matrize baterako norma desberdinen balioak oro har desberdinak izan arren, haiek zentzu batean «baliokidetzat» har daitezke. Edozein bi bektore-normetarako, esate baterako $\|\cdot\|$ eta $\|\cdot\|'$, badaude c_b eta c_g konstanteak, bektorearen tamainaren menpekoak bakarrik, non $\boldsymbol{x}$ guztietarako hau betetzen baita:

$$c_b\|\boldsymbol{x}\| \leq \|\boldsymbol{x}\|' \leq c_g\|\boldsymbol{x}\|. \tag{5.18}$$

Antzeko emaitza bat egiaztatzen da bi matrize-normetarako.

Matrize-normen propietate batzuk

Edozein $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrizetarako propietate hauek betetzen dira:

- $\|\boldsymbol{A}\|_2 \leq \|\boldsymbol{A}\|_F \leq \sqrt{n}\|\boldsymbol{A}\|_2$.
- $\dfrac{1}{\sqrt{n}}\|\boldsymbol{A}\|_\infty \leq \|\boldsymbol{A}\|_2 \leq \sqrt{m}\|\boldsymbol{A}\|_\infty$.
- $\dfrac{1}{\sqrt{m}}\|\boldsymbol{A}\|_1 \leq \|\boldsymbol{A}\|_2 \leq \sqrt{n}\|\boldsymbol{A}\|_1$.

Propietate horien arabera, bat-, bi- eta infinitu-normak eta Frobeniusen norma baliokideak dira. Beraz, matrize-norma baten balioa ezagutzen badugu, beste norma batena borna dezakegu propietate horiek erabiliz. Ondorioz, matrize baten norma estimatzeko orduan, bat- edo infinitu-norma aukera ditzakegu, zeren haietarako kalkuluak merkeagoak baitira.

$\|\boldsymbol{X}\|_p$ norma kalkulatzeko MATLABeko `norm(X,p)` funtzioa erabil daiteke, $\boldsymbol{X}$ bektore bat edo matrize bat izanik.

5.7.1. Bi-norma eta espektro-erradioa

Izan bedi $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ matrize karratua. $\boldsymbol{B}$ matrizearen *espektro-erradioa* honela definitzen da:

$$\rho(\boldsymbol{B}) = \max\{|\lambda|, \lambda \textit{ da } \boldsymbol{B} - \textit{ren autobalore bat}\}.$$

Beraz, $\rho(\boldsymbol{B}) = |\overline{\lambda}|$ eta $\overline{\lambda}$ autobaloreari dagokion autobektorea $\overline{\boldsymbol{x}}$ bada, non $\|\overline{\boldsymbol{x}}\| = 1$, hau betetzen da:

$$\rho(\boldsymbol{B}) = |\overline{\lambda}| = |\overline{\lambda}| \cdot \|\overline{\boldsymbol{x}}\| = \|\overline{\lambda}\overline{\boldsymbol{x}}\| = \|\boldsymbol{B}\overline{\boldsymbol{x}}\| \leq \|\boldsymbol{B}\|.$$

Ondorioz, espektro-erradioak beti behe-bornatzen du edozein norma induzitu. Kontuan izan matrize karratu orokor baten norma ez dela espektro-erradioa. Sarritan, espektro-erradioa norma baterako behe-borne on bat izaten da. Are gehiago, edozein $\boldsymbol{A}$ matrizetarako hau froga daiteke:

$$\|\boldsymbol{A}\|_2 = \sqrt{\rho(\boldsymbol{A}^T\boldsymbol{A})} = \sigma_1(\boldsymbol{A}).$$

Eta $\boldsymbol{B}$ matrize simetrikoa bada, $\|\boldsymbol{B}\|_2 = \rho(\boldsymbol{B})$.

5.8. Sistema lineal baten baldintzazko zenbakia

Izan bedi $\boldsymbol{A}$ matrize ez-singular bat; hots, $\boldsymbol{A}^{-1}$ existitzen da eta bakarra da. Demagun sistema lineal hau:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

bere soluzio *zehatza* (eta bakarra) $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ da. Demagun berdintzaren eskuinaldeko gaia $\boldsymbol{b}$ izatetik $\boldsymbol{b} + \delta\boldsymbol{b}$ izatera aldatzen dela (hots, $\boldsymbol{b}$ perturbatzen dugula), $\boldsymbol{A}$ berdina izanik. Izan bedi $\boldsymbol{x} + \delta\boldsymbol{x}_b$ aldatutako problemaren soluzio zehatza, alegia:

$$\boldsymbol{A}(\boldsymbol{x} + \delta\boldsymbol{x}_b) = \boldsymbol{b} + \delta\boldsymbol{b}. \tag{5.19}$$

Atal honetan, bektore edo matrize baten aurrean «δ» izateak beren dimentsio bereko aldaketa txiki (perturbazio) bat esan nahi du; esate baterako, $\delta\boldsymbol{b}$, $\boldsymbol{b}$-ren aldaketa bat da. Bestalde, $\delta\boldsymbol{x}_b$-ren b azpiindizeak esan nahi du $\boldsymbol{x}$-ren aldaketa hau $\boldsymbol{b}$-ren aldaketak eragindakoa dela.

$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ dugunez, (5.19) erlazioak $\boldsymbol{A}\delta\boldsymbol{x}_b = \delta\boldsymbol{b}$ inplikatzen du, eta ondorioz:

$$\delta\boldsymbol{x}_b = \boldsymbol{A}^{-1}\delta\boldsymbol{b}.$$

Edozein eragindako norma erabiliz, $\|\delta\boldsymbol{x}_b\|$-ren borne bat lortuko dugu, zeren (5.17) desberdintzaz hau baitugu:

$$\|\delta\boldsymbol{x}_b\| \leq \|\boldsymbol{A}^{-1}\| \cdot \|\delta\boldsymbol{b}\|. \tag{5.20}$$

Perturbazio erlatiboa bornatzeko, kontuan izan $\boldsymbol{Ax} = \boldsymbol{b}$ berdintzatik eta (5.17) -etik hau ondorioztatzen dela:

$$\|\boldsymbol{b}\| \leq \|\boldsymbol{A}\| \cdot \|\boldsymbol{x}\| \quad \Rightarrow \quad \frac{1}{\|\boldsymbol{x}\|} \leq \|\boldsymbol{A}\| \cdot \frac{1}{\|\boldsymbol{b}\|}$$

eta adierazpen hori atalez atal biderkatuz (5.20) desberdintzarekin, emaitza garrantzitsu hau lortzen dugu:

$$\frac{\|\delta \boldsymbol{x}_b\|}{\|\boldsymbol{x}\|} \leq \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| \cdot \frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|}. \tag{5.21}$$

Kontuan izan desberdintzaren eskuinaldeko kantitatea soluzio zehatzeko perturbazio erlatiboaren balio maximo posiblea dela. Kasu batzuetan, nahiz eta goi-bornea oso handia izan, $\boldsymbol{A}$, $\boldsymbol{b}$ eta $\delta \boldsymbol{b}$ berezi batzuetarako bakarrik betetzen da (5.21) adierazpeneko berdintza.

Jarraian, $\boldsymbol{A}$ matrizea aldatuko dugu pixka bat (hots, perturbatuko dugu) eta $\boldsymbol{b}$ finko eutsiko dugu; orduan, hau dugu:

$$(\boldsymbol{A} + \delta \boldsymbol{A})(\boldsymbol{x} + \delta \boldsymbol{x}_A) = \boldsymbol{b}, \tag{5.22}$$

non $\delta \boldsymbol{x}_A$-ren A azpiindizeak esan nahi baitu $\boldsymbol{x}$-ren aldaketa $\boldsymbol{A}$-ren aldaketak eragindakoa dela. Demagun $\|\delta \boldsymbol{A}\|$ nahiko txikia dela $\boldsymbol{A} + \delta \boldsymbol{A}$ ez-singularra gordetzeko moduan. Orduan, (5.22) -ren eragiketak eginez,

$$\boldsymbol{Ax} + \boldsymbol{A}(\delta \boldsymbol{x}_A) + \delta \boldsymbol{A}(\boldsymbol{x} + \delta \boldsymbol{x}_A) = \boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{A}(\delta \boldsymbol{x}_A) = -\delta \boldsymbol{A}(\boldsymbol{x} + \delta \boldsymbol{x}_A)$$

eta hortik, $\boldsymbol{A}$ ez-singularra denez, zera ondorioztatzen da:

$$\delta \boldsymbol{x}_A = -\boldsymbol{A}^{-1} \delta \boldsymbol{A}(\boldsymbol{x} + \delta \boldsymbol{x}_A).$$

Orain, (5.17) erabiliz, honako hau ateratzen da:

$$\|\delta \boldsymbol{x}_A\| \leq \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}(\boldsymbol{x} + \delta \boldsymbol{x}_A)\| \leq \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\| \cdot \|\boldsymbol{x} + \delta \boldsymbol{x}_A\|$$

eta hortik:

$$\frac{\|\delta \boldsymbol{x}_A\|}{\|\boldsymbol{x} + \delta \boldsymbol{x}_A\|} \leq \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\|,$$

eta, azkenik, aldaketa erlatiboaren bornapen hau lortzen da soluzio zehatzean:

$$\frac{\|\delta \boldsymbol{x}_A\|}{\|\boldsymbol{x} + \delta \boldsymbol{x}_A\|} \leq \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| \cdot \frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}, \tag{5.23}$$

eta berdintza gertatzen da $\delta \boldsymbol{A}$-ren eta $\boldsymbol{b}$-ren balio berezi batzuetarako.

Ikus dezakegunez, $\|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|$ kantitatea agertzen da (5.21) eta (5.23) bornapenetan, eta horrek erakusten du sistema lineal bateko datuen aldaketa batek eragindako aldaketa maximo posiblea soluzio zehatzaren gainean. Gogoan (5.13) izanik, $\boldsymbol{A}$ matrize ez-singular baten *baldintzazko zenbakia* (edo *baldintza* bakarrik) honela definitzen dugu ($\boldsymbol{Ax} = \boldsymbol{b}$ sistemaren ebazpenarekiko):

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|. \tag{5.24}$$

Edozein eragindako normatarako, identitate-matrizearen norma 1 da. Izan ere, $\mathbb{1} = \boldsymbol{A}^{-1}\boldsymbol{A}$ eta $\|\boldsymbol{A}^{-1}\boldsymbol{A}\| \leq \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$, orduan $\kappa(\boldsymbol{A}) \geq 1$. Beraz, matrize *ondo baldintzatu* baten baldintzazko zenbakia unitatearen ordenakoa da, eta matrize *txarto baldintzatu* baten baldintzazko zenbakia unitatea baino askoz handiagoa da. Nahiz eta $\kappa(\boldsymbol{A})$ balioa aldatu bere kalkuluan erabilitako normaren arabera, balio horiek konparagarriak dira normen baliokidetasunagatik.

Bestalde, sistema linealaren datuen perturbazio baterako soluzio zehatzaren aldaketa erlatiboa ezagutzen badugu, baldintzazko zenbakiaren *behe-borne* bat lor dezakegu. Berrordenatuz (5.21) eta (5.23) desberdintzak eta (5.24) kontuan hartuz, hau lortzen da:

$$\kappa(\boldsymbol{A}) \geq \frac{\|\delta\boldsymbol{x}_b\|/\|\boldsymbol{x}\|}{\|\delta\boldsymbol{b}\|/\|\boldsymbol{b}\|} \quad \text{eta} \quad \kappa(\boldsymbol{A}) \geq \frac{\|\delta\boldsymbol{x}_A\|/(\|\boldsymbol{x}+\delta\boldsymbol{x}_A\|)}{\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\|}. \tag{5.25}$$

5.1. adibidea. *Izan bedi sistema lineal hau:*

$$\begin{aligned} 0.550x_1 + 0.423x_2 &= 0.127 \\ 0.484x_1 + 0.372x_2 &= 0.112. \end{aligned}$$

Sistema horretarako, hau dugu:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad non \quad \boldsymbol{A} = \begin{bmatrix} 0.550 & 0.423 \\ 0.484 & 0.372 \end{bmatrix} \quad eta \quad \boldsymbol{b} = \begin{bmatrix} 0.127 \\ 0.112 \end{bmatrix}.$$

Ikusiko dugu sistema horren $\boldsymbol{A}$ matrizea txarto baldintzatua dela.

Ebazpena. Jo dezagun $\boldsymbol{b}$ bektorea honela perturbatzen dugula:

$$\widetilde{\boldsymbol{b}} = \boldsymbol{b} + \delta\boldsymbol{b} = \begin{bmatrix} 0.127 \\ 0.112 \end{bmatrix} + \begin{bmatrix} 0.00007 \\ 0.00028 \end{bmatrix} = \begin{bmatrix} 0.12707 \\ 0.11228 \end{bmatrix}.$$

$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ sistemaren soluzio zehatza $\boldsymbol{x} = (1, -1)^T$ da, baina $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$ sistemaren soluzio zehatza $\boldsymbol{x} = (1.7, -1.91)^T$ da, hots $\delta\boldsymbol{x}_b = (0.7, -0.91)^T$. Infinitu-norma erabiltzen badugu, $\boldsymbol{b}$-ren eta $\boldsymbol{x}$-ren perturbazio erlatiboak kalkulatzeko, zera dugu:

$$\frac{\|\delta\boldsymbol{b}\|_\infty}{\|\boldsymbol{b}\|_\infty} = \frac{0.00028}{0.127} \approx 2.2 \cdot 10^{-3} \quad \text{eta} \quad \frac{\|\delta\boldsymbol{x}_b\|_\infty}{\|\boldsymbol{x}\|_\infty} = 0.91.$$

Beraz, $\delta\boldsymbol{b}$ horretarako, soluzioan egon den aldaketa erlatiboa 413 bider baino gehiago izan da; horrek erakusten du $\boldsymbol{A}$-ren baldintzazko zenbakia gutxienez 413 dela; hots, $\kappa_\infty(\boldsymbol{A}) \geq 413$, ikus (5.25) .

$\boldsymbol{A}$-ren baldintzapen txarra ikus dezakegu a_{21} gaia pixka bat aldatuz ere (0.001 kantitatean), hau izanik:

$$\boldsymbol{A} + \delta\boldsymbol{A} = \begin{bmatrix} 0.550 & 0.423 \\ 0.483 & 0.372 \end{bmatrix}.$$

Orain, $(\boldsymbol{A}+\delta\boldsymbol{A})\widehat{\boldsymbol{x}}=\boldsymbol{b}$ sistemaren soluzio zehatza $\widehat{\boldsymbol{x}}=(-0.4536,0.8900)^T$, zeinak $\|\delta\boldsymbol{x}_A\|_\infty=\|\widehat{\boldsymbol{x}}-\boldsymbol{x}\|_\infty=1.89$ ematen baitu. Ondorioz, zera dugu:

$$\frac{\|\delta\boldsymbol{x}_A\|_\infty/\|\widehat{\boldsymbol{x}}\|_\infty}{\|\delta\boldsymbol{A}\|_\infty/\|\boldsymbol{A}\|_\infty}=\frac{1.89/0.89}{0.001/0.973}\approx 2066.$$

Kasu honetan, soluzio zehatzaren aldaketa erlatiboa 2066 bider $\boldsymbol{A}$-ren aldaketa erlatiboa da; hots, $\kappa_\infty(\boldsymbol{A})\geq 2066$, ikus (5.25) .

Benetan, $\boldsymbol{A}$-ren alderantzizko zehatza (5 digitutara biribildua) hau da:

$$\boldsymbol{A}^{-1}=\begin{bmatrix}-2818.2 & 3204.5\\ 3666.7 & -4166.7\end{bmatrix},$$

eta (infinitu-norma erabiliz) $\|\boldsymbol{A}\|_\infty=0.973$ eta $\|\boldsymbol{A}^{-1}\|_\infty=7833.4$ dugunez, hau da baldintzazko zenbakia:

$$\kappa_\infty(\boldsymbol{A})=7833.4\cdot 0.973\approx 7622.$$

Beraz, benetako baldintzazko zenbakia askoz handiagoa da $\delta\boldsymbol{b}$-ren eta $\delta\boldsymbol{A}$-ren bitartez aurkitutako 413 eta 2066, hurrenez hurren, $\kappa_\infty(\boldsymbol{A})$-ren behe-borneak baino. □

$\boldsymbol{A}$ matrize bat txarto baldintzatua da tamaina bereko bektore desberdinei aplikatuz gero, biderkadura-bektoreen tamainak oso desberdinak direnean. Alegia, $\boldsymbol{x}$ bektore unitario baterako $\|\boldsymbol{A}\boldsymbol{x}\|$ handia bada eta beste bektore unitario baterako txikia bada, orduan, $\boldsymbol{A}$ txarto baldintzatua da. Baina, $\|\boldsymbol{A}\boldsymbol{x}\|/\|\boldsymbol{x}\|$ handia bada $\boldsymbol{x}$ guztietarako (esate baterako, $\boldsymbol{A}=\operatorname{diag}(10^8,10^8)$ kasuan), orduan, $\boldsymbol{A}$ ez da txarto baldintzatua. Aldiz, jo ditzagun matrize eta bektore hauek:

$$\widehat{\boldsymbol{A}}=\begin{bmatrix}10^4 & \\ & 10^{-4}\end{bmatrix},\quad \boldsymbol{x}=\begin{bmatrix}1\\0\end{bmatrix}\quad\text{eta}\quad \widehat{\boldsymbol{x}}=\begin{bmatrix}0\\1\end{bmatrix},$$

non $\|\boldsymbol{x}\|=\|\widehat{\boldsymbol{x}}\|=1$. Orduan, hau dugu:

$$\widehat{\boldsymbol{A}}\boldsymbol{x}=\begin{bmatrix}10^4\\0\end{bmatrix}\quad\text{eta}\quad \widehat{\boldsymbol{A}}\widehat{\boldsymbol{x}}=\begin{bmatrix}0\\10^{-4}\end{bmatrix},$$

zera gertatuz: $\|\widehat{\boldsymbol{A}}\boldsymbol{x}\|=10^4$ (handia) eta $\|\widehat{\boldsymbol{A}}\widehat{\boldsymbol{x}}\|=10^{-4}$ (txikia). Alegia, $\widehat{\boldsymbol{A}}$ matrizeak txarto baldintzatua izan behar du. Egia esan, erraza da ikustea aldaketa handienak $\boldsymbol{x}$ eta $\widehat{\boldsymbol{x}}$ bektore horietarako lortzen dituela, eta $\kappa(\widehat{\boldsymbol{A}})=10^8$ dela.

$\boldsymbol{A}$-ren baldintza eta bere singulartasuna oso kontzeptu erlazionatuak dira. Informalki, txarto baldintzatutako matrize bat «ia singularra» da. Baina, nahiz eta matrize singular baten determinantea zero izan, determinantea ia zero duen matrize baten baldintza ez da nahitaez txarra izan behar. Adibidez, $n\times n$ dimentsioko $\boldsymbol{A}=\operatorname{diag}(10^{-10})$ matrizearen determinantea 10^{-10n} da, baina bere baldintza perfektua da, $\kappa(\boldsymbol{A})=1$ da.

Bestalde, matrize singular batek, gutxienez, autobalio nulu bat dauka. Gainera, $\boldsymbol{A}$ simetrikoa denean, bere baldintzazko zenbakia da autobalio handienak txikienarekiko duen

zatidura; alegia:

$$\kappa_2(\boldsymbol{A}) = \frac{\max\limits_{1\le i\le n} |\lambda_i|}{\min\limits_{1\le i\le n} |\lambda_i|}.$$

Ondorioz, matrize simetriko txarto baldintzatu batek tamaina txikiko autobalio bat izan behar du bere normarekiko (izan ere, $\|A\|_2 = \max_{1\le i\le n} |\lambda_i|$). Oro har, irizpide hori ez da zuzena matrizea simetrikoa ez denean.

$\boldsymbol{P}$ permutazio-matrizea bada, $\boldsymbol{Px}$-ren osagaiak $\boldsymbol{x}$-ren osagaien berrantolaketa direnez, $\|\boldsymbol{Px}\| = \|\boldsymbol{x}\|$ betetzen da $\boldsymbol{x}$ guztietarako, eta beraz:

$$\kappa(\boldsymbol{P}) = 1.$$

$\boldsymbol{A}$ matrizea $c \neq 0$ eskalar batez biderkatzen badugu, $\|c\boldsymbol{A}\| = |c| \cdot \|\boldsymbol{A}\|$ eta $\|(c\boldsymbol{A})^{-1}\| = |1/c| \cdot \|\boldsymbol{A}^{-1}\|$ dira, eta, ondorioz:

$$\kappa(c\boldsymbol{A}) = \|(c\boldsymbol{A})^{-1}\| \cdot \|c\boldsymbol{A}\| = \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| = \kappa(\boldsymbol{A}).$$

$\boldsymbol{D}$ matrize diagonala bada, orduan:

$$\kappa(\boldsymbol{D}) = \frac{\max |d_{ii}|}{\min |d_{ii}|}.$$

MATLABek badauzka funtzio batzuk baldintzazko zenbakia kalkulatzeko:

- $\kappa_2(\boldsymbol{A})$ (bi-norma erabiliz) kalkulatzen du `cond(A)` edo `cond(A,2)` funtzioez. Balio singularren kalkuluan erabiltzen du `svd(A)` funtzioa.
- $\kappa_1(\boldsymbol{A})$ (bat-norma erabiliz) kalkulatzen du `cond(A,1)` funtzioaz. Horrek `inv(A)` funtzioa erabiltzen du. Kalkuluak eragiketa gutxiago behar ditu `cond(A,2)` baino.
- $\kappa_\infty(\boldsymbol{A})$ (infinitu-norma erabiliz) kalkulatzen du `cond(A,inf)` funtzioaz. Horrek `inv(A)` funtzioa erabiltzen du eta `cond(A,1)`-en lan berdina behar du.

5.9. Cholesky-ren faktorizazioa

5.5. teorema. *Izan bedi $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrize simetriko bat. Orduan, $\boldsymbol{A}$-k n autobalio errealak ditu, $\lambda_1, \ldots, \lambda_n$, eta horiei elkartutako $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ autobektore unitarioek $\mathbb{R}^n$-rako oinarri ortonormal bat osatzen dute (hots, $\boldsymbol{v}_i^T\boldsymbol{v}_j = 1$, $i = j$ bada, eta $\boldsymbol{v}_i^T\boldsymbol{v}_j = 0$, $i \neq j$ bada).*

5.6. teorema. *Izan bedi $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrize simetriko bat. Orduan, $\boldsymbol{A}$ definitu positiboa ($\boldsymbol{x}^T\boldsymbol{Ax} > 0 \ \ \forall\, \boldsymbol{x} \neq \boldsymbol{0}$) da, baldin eta soilik baldin autobalio guztiak positiboak badira.*

Frogantza. Izan bitez $\lambda_1, \ldots, \lambda_n$ autobalioak eta horiei dagozkien $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ autobektore ortonormalak. Demagun j baterako $\lambda_j \leq 0$. Orduan:

$$\boldsymbol{v}_j^T \boldsymbol{A} \boldsymbol{v}_j = \boldsymbol{v}_j^T (\lambda_j \boldsymbol{v}_j) = \lambda_j \boldsymbol{v}_j^T \boldsymbol{v}_j = \lambda_j \leq 0,$$

horrek erakusten du $\boldsymbol{A}$ ez dela definitu positiboa. Beraz, $\boldsymbol{A}$ definitu positiboa bada, autobalio guztiak positiboak dira.

Orain, frogatuko dugu $\boldsymbol{A}$-ren autobalio guztiak positiboak badira, $\boldsymbol{A}$ definitu positiboa dela.

Demagun $\lambda_i > 0, \quad i = 1, \ldots, n$. Orduan, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ $\mathbb{R}^n$-rako oinarri bat osatzen dutenez, hau betetzen da edozein $\boldsymbol{v} \in \mathbb{R}^n$ bektore ez-nulu baterako:

$$\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i \quad \Rightarrow \quad \boldsymbol{A}\boldsymbol{v} = \boldsymbol{A} \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i = \sum_{i=1}^{n} \alpha_i \boldsymbol{A}\boldsymbol{v}_i = \sum_{i=1}^{n} \alpha_i \lambda_i \boldsymbol{v}_i$$

eta α_j bat gutxienez ez da zero. Orduan, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ oinarria ortonormala izateagatik, hau dugu:

$$\boldsymbol{v}^T A \boldsymbol{v} = \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i \boldsymbol{v}_i)^T (\lambda_j \alpha_j \boldsymbol{v}_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_j \alpha_i \alpha_j \boldsymbol{v}_i^T \boldsymbol{v}_j = \sum_{j=1}^{n} \lambda_j \alpha_j^2 > 0.$$

Beraz, $\boldsymbol{A}$ definitu positiboa da. □

5.7. teorema (Gerschgorin).

Izan bedi $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ matrize simetriko bat $\lambda_1, \ldots, \lambda_n$ autobalioekin. Orduan, hau dugu:

$$\min_{1 \leq i \leq n} \lambda_i \geq \min_{1 \leq i \leq n} \left\{ a_{ii} - \sum_{j=1, j \neq i}^{n} |a_{ij}| \right\}$$

$$\max_{1 \leq i \leq n} \lambda_i \leq \max_{1 \leq i \leq n} \left\{ a_{ii} + \sum_{j=1, j \neq i}^{n} |a_{ij}| \right\}.$$

5.1. korolarioa. *$\boldsymbol{A}$ matrizea definitu positiboa da, $i = 1, \ldots, n$ guztietarako hau betetzen bada:*

$$a_{ii} - \sum_{j=1, j \neq i}^{n} |a_{ij}| > 0.$$

Frogantza. (Kontuan izan $\boldsymbol{A}$ hertsiki diagonal menperatzailea dela). Gerschgorin-en teoremaren lehenengo desberdintzagatik, autobalio txikiena zero baino handiagoa denez, $\boldsymbol{A}$ definitu positiboa da. □

Hala ere, gerta daiteke $\boldsymbol{A}$ definitu positiboa izatea eta ez hertsiki diagonal menperatzailea.

$\boldsymbol{A}$ simetrikoa bada, bere autobalio txikienaren hurbilpen gisa balio hau hartu ohi da:

$$d = \min_{1\leq i\leq n} \left\{ a_{ii} - \sum_{j=1, j\neq i}^{n} |a_{ij}| \right\}.$$

Ariketa gisa, frogatu $\boldsymbol{A}$ matrizea simetrikoa eta definitu positiboa bada, diagonaleko gai guztiak positiboak direla, eta matrizearen tamaina handieneko gaia diagonalean dagoela.

5.8. teorema. *$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ definitu positiboa bada eta $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ matrizeak k heina badu, $\boldsymbol{B} = \boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} \in \mathbb{R}^{k\times k}$ ere definitu positiboa da.*

Frogantza. Demagun $\boldsymbol{v} \in \mathbb{R}^k$ bektoreak hau betetzen duela:

$$0 \geq \boldsymbol{v}^T\boldsymbol{B}\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{v} = (\boldsymbol{X}\boldsymbol{v})^T\boldsymbol{A}(\boldsymbol{X}\boldsymbol{v}),$$

baina, $\boldsymbol{A}$ definitu positiboa denez, $\boldsymbol{X}\boldsymbol{v} = \boldsymbol{0}$ izan behar da; gainera, $\boldsymbol{X}$-ren k zutabeak linealki askeak direnez, $\boldsymbol{v} = \boldsymbol{0}$ izan behar da. Horrek esan nahi du aurreko desberdintza ezin dela bete eta $\boldsymbol{B}$ definitu positiboa dela. □

$\boldsymbol{A}$ simetrikoa eta definitu positiboa bada, 5.3. teoremagatik, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ da, non $\boldsymbol{L}$ unitate-matrize behe-triangeluarra eta $\boldsymbol{D} = \operatorname{diag}(d_1, \ldots, d_n)$ matrize diagonala baitira. Beraz, $\boldsymbol{L}^{-1}\boldsymbol{A}(\boldsymbol{L}^{-1})^T = \boldsymbol{D}$ dugu, eta aurreko teoremagatik $\boldsymbol{A}$ definitu positiboa bada, $\boldsymbol{D}$ ere bai; eta $\boldsymbol{D}$ diagonala denez, $d_i > 0$, $i = 1, \ldots, n$.

5.9. teorema (Choleskyren faktorizazioa). *$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizea simetrikoa eta definitu positiboa bada, $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ matrize goi-triangeluar bakar bat existitzen da diagonaleko gai positiboekin eta $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ betetzen duena.*

Frogantza. $\boldsymbol{L}$ unitate-matrize behe-triangeluar bakar bat eta $\boldsymbol{D} = \operatorname{diag}(d_1, \ldots, d_n)$ matrize diagonal bakar bat daude $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ betetzen dutenak (ikus 5.3. teorema). d_k gaiak positiboak direnez, $\boldsymbol{R}^T = \boldsymbol{L}\operatorname{diag}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ matrizea erreala eta behe-triangeluarra da gai diagonal positiboekin. Gainera, $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ betetzen du. Bakartasuna $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ faktorizazioaren bakartasunak inplikatzen du. □

$\boldsymbol{A}$ simetrikoa ($\boldsymbol{A} = \boldsymbol{A}^T$) eta definitu positiboa denean, $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ faktorizazioa (edo deskonposizioa) kalkula dezakegu, non $\boldsymbol{R}$ matrize goi-triangeluar bat baita, hots:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{12} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{1n} & a_{2n} & \ldots & a_{nn} \end{bmatrix} = \begin{bmatrix} r_{11} & & & \\ r_{12} & r_{22} & & \\ \vdots & & \ddots & \\ r_{1n} & r_{2n} & \ldots & r_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ & r_{22} & & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}.$$

Hori dela eta, zera dugu:

- $\boldsymbol{A}$-ren a_{11} gaitik 1. lerroari jarraituz, ekuazio hauek ditugu:

$$\begin{aligned} r_{11}^2 &= a_{11} \\ r_{11}r_{12} &= a_{12} \\ &\vdots \\ r_{11}r_{1n} &= a_{1n}. \end{aligned}$$

- $\boldsymbol{A}$-ren a_{22} gaitik 2. lerroari jarraituz, ekuazio hauek ditugu:

$$\begin{aligned} r_{12}^2 + r_{22}^2 &= a_{22} \\ r_{12}r_{13} + r_{22}r_{23} &= a_{23} \\ &\vdots \\ r_{12}r_{1n} + r_{22}r_{2n} &= a_{2n}. \end{aligned}$$

- Horrela jarraituz, $i = 1, \ldots, n$ guztietarako, $\boldsymbol{A}$-ren a_{ii} gaitik i. lerrorako ekuazio hauek ditugu:

$$\begin{aligned} r_{1i}^2 + r_{2i}^2 + \ldots + r_{ii}^2 &= a_{ii} \\ r_{1i}r_{1,i+1} + r_{2i}r_{2,i+1} + \ldots + r_{ii}r_{i,i+1} &= a_{i,i+1} \\ &\vdots \\ r_{1i}r_{1n} + r_{2i}r_{2n} + \ldots\ldots\ldots\ldots + r_{ii}r_{in} &= a_{in}. \end{aligned}$$

- Ondorioz, $i = 1, \ldots, n$ guztietarako, $\boldsymbol{R}$-ren koefizienteak honela kalkula ditzakegu:

$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}, \qquad i = 1, \ldots, n, \tag{5.26}$$

$$r_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj}}{r_{ii}}, \qquad j = i+1, \ldots, n. \tag{5.27}$$

5.2. adibidea. *Kalkula ezazu matrize simetriko eta definitu positibo honen Choleskyren faktorizazioa:*

$$\boldsymbol{A} = \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}.$$

Ebazpena. Lehenengo lerrorako ($i = 1$) (5.26) erabiliz, zera dugu:

$$r_{11} = \sqrt{a_{11}} = \sqrt{6} = 2.44949.$$

Orduan, (5.27) erabiliz koefiziente hauek lortzeko:

$$r_{12} = \frac{a_{12}}{r_{11}} = \frac{15}{2.44949} = 6.123724$$

$$r_{13} = \frac{a_{13}}{r_{11}} = \frac{55}{2.44949} = 22.45366.$$

Hurrengo lerrorako ($i = 2$), zera dugu:

$$\begin{aligned} r_{22} &= \sqrt{a_{22} - r_{12}^2} = \sqrt{55 - (6.123724)^2} = 4.1833 \\ r_{23} &= \frac{a_{23} - r_{12}r_{13}}{r_{22}} = \frac{225 - 6.123724 \cdot 22.45366}{4.1833} = 20.9165. \end{aligned}$$

Hirugarren lerrorako ($i = 3$), hau lortuko dugu:

$$r_{33} = \sqrt{a_{33} - r_{13}^2 - r_{23}^2} = \sqrt{979 - (22.45366)^2 - (20.9165)^2} = 6.110101.$$

Ondorioz, Choleskyren faktorizazioak zera ematen du:

$$\begin{bmatrix} 2.44949 & 6.123724 & 22.45366 \\ & 4.1833 & 20.9165 \\ & & 6.110101 \end{bmatrix}.$$

Faktorizazio horren baliotasuna ontzat emango dugu, $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{A}$ berdintza betetzen dela egiaztatuz. □

$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{A}$ faktorizazio hori lortu ondoren, $\boldsymbol{Ax} = \boldsymbol{b}$ sistema ebatz dezakegu *LU* faktorizazioarekin egiten den bezala; alegia, sistema triangeluar hauek ebatziz:

$$\begin{aligned} \boldsymbol{R}^T\boldsymbol{y} &= \boldsymbol{b} \\ \boldsymbol{Rx} &= \boldsymbol{y}. \end{aligned}$$

$\boldsymbol{A}$ matrize simetriko bat definitu positiboa bada, ez da beharrezkoa lerroen trukaketa; aldiz, *LU* metodoak pibotaze partziala behar du zenbakizko egonkorra izateko. Choleskyren metodoan nahiz eta $\boldsymbol{A}$ txarto baldintzatua izan, ez dugu pibotaziorik egin behar, zeren $i = 1, \ldots, n$ guztietarako hau betetzen baita:

$$r_{1i}^2 + r_{2i}^2 + \ldots + r_{ii}^2 = a_{ii} \tag{5.28}$$

eta, ondorioz, $\boldsymbol{R}$-ko gai guztiak honela bornatuta daude:

$$|r_{ji}| \leq \sqrt{a_{ii}}, \quad j = 1, \ldots, i.$$

Metodo hau oso erabilia da optimizazioan, bere zenbakizko egonkortasunagatik. MATLABen `chol(A)` funtzioak $\boldsymbol{A}$ matrizearen Choleskyren faktorea ematen digu.

5.10. teorema. *Izan bedi $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrize simetrikoa eta definitu positiboa eta $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ bere Choleskyren deskonposizioa. Orduan:*

$$\kappa_2(\boldsymbol{A}) \geq \frac{\max\limits_i r_{ii}^2}{\min\limits_i r_{ii}^2}.$$

Gainera, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ Choleskyren deskonposizioa erabiltzen badugu, $\boldsymbol{L}$ unitate-matrize behetriangeluarra eta $\boldsymbol{D} = \operatorname{diag}(d_i)$ matrize diagonala izanik, hau betetzen da:

$$\kappa_2(\boldsymbol{A}) \geq \frac{\max\limits_i d_i}{\min\limits_i d_i}.$$

Alegia, $\boldsymbol{D}$-ren gai diagonal maximoaren eta minimoaren zatidurak $\boldsymbol{A}$ matrizearen baldintzaren behe-borne bat ematen digu, eta, bide batez, hurbilpen bat.

Frogantza:

$$\|\boldsymbol{A}\|_2 = \max_{\|\boldsymbol{u}\|_2=1} \|\boldsymbol{A}\boldsymbol{u}\|_2 \geq \max_i \|A\boldsymbol{e}_i\|_2 = \max_i \sqrt{a_{1i}^2 + \ldots + a_{ni}^2} \geq \max_i a_{ii} \geq \max_i r_{ii}^2$$

azken desberdintza (5.28) berdintzaren ondorioa da.

Izan bedi $\boldsymbol{B} = \boldsymbol{A}^{-1}$, orduan $\boldsymbol{B} = (\boldsymbol{R}^T\boldsymbol{R})^{-1} = \boldsymbol{R}^{-1}(\boldsymbol{R}^T)^{-1} = \boldsymbol{R}^{-1}(\boldsymbol{R}^{-1})^T$. Beraz,

$$\|\boldsymbol{B}\|_2 \geq \max_i b_{ii} \geq \max_i \frac{1}{r_{ii}^2} = \frac{1}{\min\limits_i r_{ii}^2}.$$

Ondorioz,

$$\kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2\|\boldsymbol{B}\|_2 \geq \frac{\max\limits_i r_{ii}^2}{\min\limits_i r_{ii}^2}.$$

Bestalde, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T = \boldsymbol{R}^T\boldsymbol{R}$ denez, $d_i = r_{ii}^2$ eta horrek frogantza hau bukatzen du. □

5.10. Metodo iteratiboak

LU eta Choleskyren faktorizazioa *metodo zuzenak* dira; izan ere, ez balira egongo biribiltze-erroreak, ikusi dugun bezala, urratsen kopuru finitu batean soluzio zehatzera helduko lirateke. Baina, sistemaren matrizea eskasa bada (hots, gaien ehuneko handi bat zeroak direnean), Gaussen metodoak zeroak hondatzen ditu (hots, ez-zero bihurtzen ditu). Edonola ere, matrizearen egitura hau eskasa denean, badaude estrategia egokiak esplotatzeko; ikus [10, 13]. MATLABeko `nnz(A)` funtzioak $\boldsymbol{A}$ matrizean zero ez direnen gaien kopurua zenbatzen du.

Metodo iteratiboak egokiak dira matrizearen egitura aprobetxatuz eragiketa gutxiago egiteko eta ordenagailuaren memoria gutxiago erabiltzeko, eta hori oso interesgarria da sistema oso handia denean, deribatu partzialetako ekuazioen sistemetan bezala.

5.10.1. Jacobi-ren iterazioa

5.3. adibidea. *Ebatzi sistema hau:*

$$\begin{aligned} 4x - y + z &= 7 \\ 4x - 8y + z &= -21 \\ -2x + y + 5z &= 15. \end{aligned}$$

Ebazpena. Ekuazio horiek honela idatz ditzakegu:

$$\begin{aligned} x &= \frac{7 + y - z}{4} \\ y &= \frac{21 + 4x + z}{8} \\ z &= \frac{15 + 2x - y}{5} \end{aligned}$$

eta, hortik, prozesu iteratibo hau ateratzen dugu:

$$\begin{aligned} x^{(k+1)} &= \frac{7 + y^{(k)} - z^{(k)}}{4} \\ y^{(k+1)} &= \frac{21 + 4x^{(k)} + z^{(k)}}{8} \\ z^{(k+1)} &= \frac{15 + 2x^{(k)} - y^{(k)}}{5}. \end{aligned} \tag{5.29}$$

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1.0	2.0	2.0
1	1.75	3.375	3.0
2	1.84375	3.875	3.025
3	1.9625	3.925	2.9625
4	1.99062500	3.97656250	3.00000000
5	1.99414063	3.99531250	3.00093750
...	...	...	...
15	1.99999993	3.99999985	2.99999993
...	...	...	...
19	2.00000000	4.00000000	3.00000000

5.1. taula. Jacobiren iterazioaren konbergentzia 5.3. adibidean.

Hasierako puntua $(x^{(0)}, y^{(0)}, z^{(0)})^T = (1, 2, 2)^T$ bada, ikusiko dugu iterazio horrek $(2, 4, 3)^T$ soluziora jotzen duela.

$x^{(0)} = 1$, $y^{(0)} = 2$ eta $z^{(0)} = 2$ ordezkatuz (5.29) eskuineko ataletan, hau lortzen dugu:

$$\begin{aligned} x^{(1)} &= \frac{7 + 2 - 2}{4} = 1.75 \\ y^{(1)} &= \frac{21 + 4 \cdot 1 + 2}{8} = 3.375 \\ z^{(1)} &= \frac{15 + 2 \cdot 1 - 2}{5} = 3.00. \end{aligned}$$

Prozesu hori jarraituz, 5.10.1. taula lortzen da. Taula horrek adierazten du (5.29) iterazioak $(2, 4, 3)^T$ soluziora jotzen duela. □

Prozesu horri *Jacobiren iterazioaren metodoa* deritzogu.

Izan bedi $\boldsymbol{Ax} = \boldsymbol{b}$ sistema eta $\boldsymbol{A}$ matrizearen deskonposizio hau:

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U},$$

non

$$\begin{aligned} \boldsymbol{L} &= \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ a_{21} & 0 & \dots & & 0 \\ a_{31} & a_{32} & \ddots & & 0 \\ \vdots & & & 0 & 0 \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{bmatrix} \\ \boldsymbol{D} &= \operatorname{diag}(a_{11}, \dots, a_{nn}) \\ \boldsymbol{U} &= \begin{bmatrix} 0 & a_{12} & \dots & \dots & a_{1n} \\ 0 & 0 & \dots & \dots & \vdots \\ 0 & 0 & \ddots & & a_{n-2,n} \\ \vdots & & & \ddots & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}. \end{aligned} \tag{5.30}$$

Orduan:

$$\boldsymbol{Ax} = (\boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U})\boldsymbol{x} = \boldsymbol{b}$$

eta ondorioz,

$$\boldsymbol{Dx} = \boldsymbol{b} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}.$$

Hortaz, honela idatz dezakegu Jacobiren k-garren iterazioa:

$$\boldsymbol{Dx}^{(k+1)} = \boldsymbol{b} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}^{(k)}. \tag{5.31}$$

Beraz, matrizeak ez dira aldatzen eta, eskasak badira (hots, zero asko badituzte), eragiketa gutxi egin behar ditugu. Orain, (5.31) iterazioa garatuz x_i bakoitzeko, $i = 1, \ldots, n$ guztietarako, zera dugu:

$$a_{ii}x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right).$$

Azkenik, $i = 1, \ldots, n$ guztietarako, hau egingo da Jacobiren iterazioan:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right) \Big/ a_{ii}. \tag{5.32}$$

5.10.2. Gauss-Seidel-en iterazioa

Jacobiren metodoaren konbergentzia azeleratzeko asmoz, beste metodo hau dugu. Aurreko 5.3. adibideko $\{x^{(k)}\}$, $\{y^{(k)}\}$, $\{z^{(k)}\}$ segidek 2ra, 4ra eta 3ra jotzen dute, hurrenez hurren. Izan ere, $x^{(k+1)}$ seguraski $x^{(k)}$ baino bere limitearen hurbilpen hobea denez, arrazoizkoa izango litzateke $y^{(k+1)}$ kalkulatzeko $x^{(k)}$-ren ordez $x^{(k+1)}$ erabiltzea. Hortaz, hobe izango litzateke $z^{(k+1)}$ kalkuluan $y^{(k)}$-ren ordez $y^{(k+1)}$ erabiltzea ere. Hori kontuan hartuz, honela geratzen da iterazioa:

$$\begin{aligned} x^{(k+1)} &= \frac{7 + y^{(k)} - z^{(k)}}{4} \\ y^{(k+1)} &= \frac{21 + 4x^{(k+1)} + z^{(k)}}{8} \\ z^{(k+1)} &= \frac{15 + 2x^{(k+1)} - y^{(k+1)}}{5}. \end{aligned} \tag{5.33}$$

Metodo horri *Gauss-Seidelen iterazioaren metodoa* deritzogu. Eta aldaketa horiek (5.32) ekuazioari eramanez, hau lortzen dugu $i = 1, \ldots, n$ guztietarako:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right) \Big/ a_{ii}. \tag{5.34}$$

Era matrizialean adierazita, honela da:

$$\boldsymbol{D}\boldsymbol{x}^{(k+1)} = \boldsymbol{b} - \boldsymbol{L}\boldsymbol{x}^{(k+1)} - \boldsymbol{U}\boldsymbol{x}^{(k)} \tag{5.35}$$

edo $(\boldsymbol{D} + \boldsymbol{L})\boldsymbol{x}^{(k+1)} = \boldsymbol{b} - \boldsymbol{U}\boldsymbol{x}^{(k)}$.

Orain 5.3. adibidearen sistema ebatziko dugu Gauss-Seidelen metodoa erabiliz (hots, (5.33)); orduan, $y^{(0)} = 2$ eta $z^{(0)} = 2$ ordezkatuz, hau dugu:

$$x^{(1)} = \frac{7 + 2 - 2}{4} = 1.75$$

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1.0	2.0	2.0
1	1.75	3.75	2.95
2	1.95	3.96875	2.98625
3	1.995625	3.999609375	2.99903125
...	...	...	...
8	1.99999983	3.99999988	2.99999996
9	1.99999998	3.99999999	3.00000000
10	2.00000000	4.00000000	3.00000000

5.2. taula. Gauss-Seidelen iterazioaren konbergentzia 5.3. adibidean.

eta $x^{(1)} = 1.75$ eta $z^{(0)} = 2$ ordezkatuz, zera lortzen da:

$$y^{(1)} = \frac{21 + 4 \cdot 1.75 + 2}{8} = 3.75.$$

Azkenik, $x^{(1)} = 1.75$ eta $y^{(1)} = 3.75$ ordezkatuz, hau lortzen dugu:

$$z^{(1)} = \frac{15 + 2 \cdot 1.75 - 3.75}{5} = 2.95.$$

Emaitza hau aurrekoa baino hurbilago dago benetako soluziotik, $(x, y, z)^T = (2, 4, 3)^T$.

Prozesu horri jarraituz, 5.2. taula lortzen da.

5.10.3. Metodo egonkorren konbergentzia

Jacobiren eta Gauss-Seidelen metodoak puntu finkoko metodo bezala ikus ditzakegu (*erlaxazio-metodoak* ere esaten zaie).

Izan bedi $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ gure matrizearen erdibitze bat; hots, $\boldsymbol{N} = \boldsymbol{M} - \boldsymbol{A}$ dugu. Beraz, $\boldsymbol{Ax} = (\boldsymbol{M} - \boldsymbol{N})\boldsymbol{x} = \boldsymbol{b}$, eta $\boldsymbol{Mx} = \boldsymbol{Nx} + \boldsymbol{b}$ idatz dezakegu. Horrek puntu finkoko iterazio honetara eramaten gaitu:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{x}_k + \boldsymbol{M}^{-1}\boldsymbol{b} \\ &= \boldsymbol{M}^{-1}(\boldsymbol{M} - \boldsymbol{A})\boldsymbol{x}_k + \boldsymbol{M}^{-1}\boldsymbol{b} \\ &= \boldsymbol{x}_k - \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{M}^{-1}\boldsymbol{b} \\ &= \boldsymbol{x}_k + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \end{aligned} \tag{5.36}$$

eta hori $\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$ puntu finkoko iterazioa da, non $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$. Metodo horri egonkorra deritzogu; funtzio hori definitzen duten $\boldsymbol{A}$, $\boldsymbol{M}$ eta $\boldsymbol{b}$ bektoreak konstante baitira, ez daude iterazioaren menpe.

Jarraian ikusiko dugu nola $\boldsymbol{e}_k = \boldsymbol{x} - \boldsymbol{x}_k$ erroreak jokatzen duen adierazpen orokor horretan, eta nola errorea erlazionatuta dagoen $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ hondarrarekin.

Errorearen adierazpena $\boldsymbol{A}$-rekin biderkatuz zera dugu:

$$\boldsymbol{A}\boldsymbol{e}_k = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{r}_k \;\Rightarrow\; \underline{\boldsymbol{e}_k = \boldsymbol{A}^{-1}\boldsymbol{r}_k}$$

eta ondorioz

$$\boldsymbol{x} = \boldsymbol{x}_k + \boldsymbol{e}_k = \boldsymbol{x}_k + \boldsymbol{A}^{-1}\boldsymbol{r}_k,$$

baina kalkulu horren zailtasuna eta $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ sistema ebaztekoa berdina da. Hori dela eta, $\boldsymbol{A}\boldsymbol{e}_k = \boldsymbol{r}_k$ sistema ebatzi beharrean, $\boldsymbol{e}_k$ errorea hurbilduko dugu $\boldsymbol{M}\boldsymbol{p}_k = \boldsymbol{r}_k$ sistema ebatziz, non $\boldsymbol{p}_k \approx \boldsymbol{e}_k$. Beraz, gure iterazioa aurkitzeko k bakoitzerako $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ hondarra kalkulatuko dugu, eta gero hau kalkulatuko dugu:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{M}^{-1}\boldsymbol{r}_k,$$

hots, goiko metodo finkoko iterazioa.

Orain, ikus dezagun Jacobiren eta Gauss-Seidelen metodoak mota horretakoak direla.

Baldin $\boldsymbol{M} = \boldsymbol{D}$ hartzen badugu, zera dugu:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{D}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \\
&= \boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{x}_k + \boldsymbol{D}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \\
&= \boldsymbol{D}^{-1}(\boldsymbol{b} - (\boldsymbol{A} - \boldsymbol{D})\boldsymbol{x}_k) \\
&= \boldsymbol{D}^{-1}(\boldsymbol{b} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}_k)
\end{aligned}$$

eta hori Jacobiren iterazioa da; ikus (5.31).

Aldiz, $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{D}$ hartzen badugu, zera dugu:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + (\boldsymbol{L} + \boldsymbol{D})^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \\
&= (\boldsymbol{L} + \boldsymbol{D})^{-1}(\boldsymbol{L} + \boldsymbol{D})\boldsymbol{x}_k + (\boldsymbol{L} + \boldsymbol{D})^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \\
&= (\boldsymbol{L} + \boldsymbol{D})^{-1}(\boldsymbol{b} - (\boldsymbol{A} - (\boldsymbol{L} + \boldsymbol{D}))\boldsymbol{x}_k) \\
&= (\boldsymbol{L} + \boldsymbol{D})^{-1}(\boldsymbol{b} - \boldsymbol{U}\boldsymbol{x}_k)
\end{aligned}$$

eta hori Gauss-Seidelen iterazioa da; ikus (5.35).

Konbergentziaren analisia

Gogora dezagun $\boldsymbol{x}_k$ iterazio bakoitzean $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ hondarra ezaguna dela; aldiz, $\boldsymbol{e}_k = \boldsymbol{x} - \boldsymbol{x}_k = \boldsymbol{A}^{-1}\boldsymbol{r}_k$ errorea ezezaguna da. Jakin nahi dugu noiz gertatzen den $\boldsymbol{e}_k \to \boldsymbol{0}$, $k \to \infty$ denean. (5.36) erabiliz eta adierazpen hauek kenduz:

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} - \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{M}^{-1}\boldsymbol{b} = (\mathbb{1} - \boldsymbol{M}^{-1}\boldsymbol{A})\boldsymbol{x}_{k-1} + \boldsymbol{M}^{-1}\boldsymbol{b} \\
\boldsymbol{x} &= \boldsymbol{x} - \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{M}^{-1}\boldsymbol{b} = (\mathbb{1} - \boldsymbol{M}^{-1}\boldsymbol{A})\boldsymbol{x} + \boldsymbol{M}^{-1}\boldsymbol{b},
\end{aligned}$$

$\boldsymbol{x}-\boldsymbol{x}_k=(\mathbb{1}-\boldsymbol{M}^{-1}\boldsymbol{A})(\boldsymbol{x}-\boldsymbol{x}_{k-1})$ lortu dugu. Orain, $\boldsymbol{T}=\mathbb{1}-\boldsymbol{M}^{-1}\boldsymbol{A}$ izendatuz, zera dugu:

$$\boldsymbol{x}-\boldsymbol{x}_k=\boldsymbol{T}(\boldsymbol{x}-\boldsymbol{x}_{k-1}),$$

eta, ondorioz, $\boldsymbol{e}_k=\boldsymbol{T}\boldsymbol{e}_{k-1}=\boldsymbol{T}(\boldsymbol{T}\boldsymbol{e}_{k-2})=\ldots=\boldsymbol{T}^k\boldsymbol{e}_0$ betetzen da. $\boldsymbol{T}$ matrizeari *iterazio-matrizea* deritzogu. $\boldsymbol{e}_0$ hasierako errorea denez, konbergentzia (hots, $\boldsymbol{e}_k\to\boldsymbol{0}$) izango dugu baldin eta soilik baldin $\boldsymbol{T}^k\to\boldsymbol{0}$. Jakina, hori gertatzen da eragindako matrize-norma baterako hau betetzen bada:

$$\|\boldsymbol{T}\|<1,$$

zeren

$$\|\boldsymbol{e}_k\|=\|\boldsymbol{T}\boldsymbol{T}\ldots\boldsymbol{T}\boldsymbol{e}_0\|\le\|\boldsymbol{T}\|\|\boldsymbol{T}\|\ldots\|\boldsymbol{T}\|\|\boldsymbol{e}_0\|=\|\boldsymbol{T}\|^k\|\boldsymbol{e}_0\|.$$

Puntu finkoaren metodoan dugun konbergentzia-baldintzaren antzekoa da. Aurrekoa kontuan hartuz, teorema hau ondorioztatzen da.

5.11. teorema: metodo egonkorren konbergentzia.

$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ sistema linealerako, izan bedi metodo iteratibo hau:

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\boldsymbol{M}^{-1}\boldsymbol{r}_k,\quad k=0,1,\ldots,$$

non $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ hondar-bektorea eta $\boldsymbol{T}=\mathbb{1}-\boldsymbol{M}^{-1}\boldsymbol{A}$ iterazio-matrizea definitzen baititugu.

Orduan, metodoa konbergentea da baldin eta soilik baldin iterazio-matrizearen espektro-erradioak hau betetzen badu:

$$\rho(\boldsymbol{T})<1.$$

Zenbat eta $\rho(\boldsymbol{T})$ txikiago den, hainbat eta azkarrago konbergitzen da.

Frogantza. Lehendabizi, $\rho(\boldsymbol{T})<1\ \Rightarrow\ \boldsymbol{e}_k\to 0$ betetzen dela frogatuko dugu.

Izan bitez $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n$ bektore linealki independenteak $\boldsymbol{T}$ matrizearen autobektoreak. Orduan, hau idatz dezakegu:

$$\boldsymbol{e}_0=\sum_{i=1}^{n}\alpha_i\boldsymbol{v}_i.$$

Izan bitez $\boldsymbol{T}$-ren $\boldsymbol{v}_i$ autobektoreei dagozkien λ_i autobalioak. Baldin $\rho(\boldsymbol{T})<1$ bada, zera dugu:

$$\boldsymbol{T}\boldsymbol{v}_i=\lambda_i\boldsymbol{v}_i,\quad|\lambda_i|<1,\quad i=1,\ldots,n,$$

ondorioz,

$$\boldsymbol{e}_k=\boldsymbol{T}^k\boldsymbol{e}_0=\sum_{i=1}^{n}\alpha_i\boldsymbol{T}^k\boldsymbol{v}_i=\sum_{i=1}^{n}\alpha_i\lambda_i^k\boldsymbol{v}_i\ \to\ \boldsymbol{0}\qquad(k\to\infty).$$

Orain $\boldsymbol{e}_k\to 0\ \Rightarrow\ \rho(\boldsymbol{T})<1$ frogatuko dugu absurdora eramanez. Baldin $\rho(\boldsymbol{T})\ge 1$ bada, badago i bat non $|\lambda_i|\ge 1$ betetzen baita. Izan bedi $\boldsymbol{v}_i$ dagokion autobektorea. Hortaz,

$\boldsymbol{x}_0 = \boldsymbol{x} - \boldsymbol{v}_i$ hartzen badugu, $\boldsymbol{e}_k = \boldsymbol{T}^k\boldsymbol{e}_0 = \boldsymbol{T}^k(\boldsymbol{x} - \boldsymbol{x}_0) = \boldsymbol{T}^k\boldsymbol{v}_i = \lambda_i^k\boldsymbol{v}_i \not\to \boldsymbol{0}$ da eta ez da konbergentea. □

Jakina, $\|\boldsymbol{T}\| < 1$ bada, $\rho(\boldsymbol{T}) < 1$ da (ikus 5.7.1. atala). Hala ere, espektro-erradioaren gaineko baldintzak gehiago esaten digu $\boldsymbol{T}$-ren normaren baldintzak baino. Zeren eta $\rho(\boldsymbol{T}) < 1$ baldintza beharrezkoa eta nahikoa baita, eta, aldiz, $\|\boldsymbol{T}\| < 1$ baldintza nahikoa bakarrik da.

Zenbat iterazio behar dugu errorea 10 bider txikiagoa egiteko?

Beste hitz batzuetan, $0.1\|\boldsymbol{e}_0\| = \|\boldsymbol{e}_k\| \approx \rho(\boldsymbol{T})^k\|\boldsymbol{e}_0\|$ gertatzeko, zein izan behar du k iterazio kopuruak?

Bi ataletan $\log_{10}$ hartuz eta k bakanduz, hau lortzen da:

$$k \approx -\frac{1}{\log_{10}\rho(\boldsymbol{T})}.$$

Konbergentzia-ratioa honela definitzen da: ratioa$=-\log_{10}\rho(\boldsymbol{T})$. Orduan $k \approx 1/\text{ratioa}$.

Ondorioz, zenbat eta txikiago den espektro-erradioa, hainbat eta handiagoa da ratioa eta, beraz, iterazio gutxiago behar izango dira errore-txikitze maila berdina lortzeko.

Hertsiki diagonal menperatzailea

Izan bedi $\boldsymbol{Ax} = \boldsymbol{b}$ sistema. Aurreko metodo iteratiboak konbergenteak izateko, nahikoa da (5.12) baldintza betetzea, hots, hertsiki diagonal menperatzailea izatea. Jarraian frogatuko dugu.

5.12. teorema.

Izan bedi $\boldsymbol{Ax} = \boldsymbol{b}$ sistema. Jacobiren eta Gauss-Seidelen metodo iteratiboak konbergenteak izateko, nahikoa da $\boldsymbol{A}$ hertsiki diagonal menperatzailea izatea.

Frogantza.

<u>*Jacobirena.*</u> Metodo honetan hau dugu:

$$\boldsymbol{T} = \mathbb{1} - \boldsymbol{D}^{-1}\boldsymbol{A} = \boldsymbol{D}^{-1}\boldsymbol{D} - \boldsymbol{D}^{-1}\boldsymbol{A} = \boldsymbol{D}^{-1}(\boldsymbol{D} - \boldsymbol{A}) = -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U}),$$

eta $\rho(\boldsymbol{T}) \le \|\boldsymbol{T}\|$ denez, zera lortzen da:

$$\rho(\boldsymbol{T}) \le \|\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\|_\infty = \max_{1\le i\le n} \sum_{\substack{j=1\\ j\ne i}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| < 1,$$

Izan ere, $\boldsymbol{A}$ hertsiki diagonal menperatzailea izateagatik azken desberdintza betetzen da. Beraz, $\rho(\boldsymbol{T}) < 1$ denez, metodo hau konbergentea da.

<u>*Gauss-Seidelena.*</u> Metodo honetan hau dugu:

$$\begin{aligned}\boldsymbol{T} &= \mathbb{1} - (\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{A} = (\boldsymbol{L}+\boldsymbol{D})^{-1}(\boldsymbol{L}+\boldsymbol{D}) - (\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{A} \\ &= (\boldsymbol{L}+\boldsymbol{D})^{-1}((\boldsymbol{L}+\boldsymbol{D}) - \boldsymbol{A}) = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}.\end{aligned}$$

Izan bedi

$$\gamma = \max_{1\le i\le n} \frac{\sum_{j>i} |a_{ij}|}{|a_{ii}| - \sum_{j<i} |a_{ij}|}.$$

Orain, $\boldsymbol{A}$ hertsiki diagonal menperatzailea denez, i guztietarako hau dugu:

$$|a_{ii}| - \sum_{j<i} |a_{ij}| > \sum_{j>i} |a_{ij}|,$$

ondorioz $\gamma < 1$. Jarraian, $\|\boldsymbol{T}\|_\infty \le \gamma$ frogatuko dugu.

Demagun $\|\boldsymbol{x}\|_\infty = 1$. Izan bedi $\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x}$, hots, $(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{y} = -\boldsymbol{U}\boldsymbol{x}$. Sistema horren i. ekuazioa, $i = 1, \ldots, n$, hau da:

$$a_{ii}y_i + \sum_{j<i} a_{ij}y_j = -\sum_{j>i} a_{ij}x_j. \tag{5.37}$$

Izan bedi i non $\|\boldsymbol{y}\|_\infty = |y_i|$. Orduan (5.37)-en ezker aldearen balio absolutua adierazpen honek behetik bornatzen du ($|a| - |b| \le |a+b|$ baita):

$$\begin{aligned}|a_{ii}y_i| - |\sum_{j<i} a_{ij}y_j| &\ge |a_{ii}|\ |y_i| - \sum_{j<i} |a_{ij}|\ |y_j| \\ &\ge |a_{ii}|\ |y_i| - \sum_{j<i} |a_{ij}|\ |y_i| \\ &= (|a_{ii}| - \sum_{j<i} |a_{ij}|)\|\boldsymbol{y}\|_\infty.\end{aligned}$$

Era antzeko batean, (5.37)-en eskuin aldearen balio absolutua adierazpen honek goitik bornatzen du:

$$\sum_{j>i} |a_{ij}|\ |x_j| \le \sum_{j>i} |a_{ij}|\ \|\boldsymbol{x}\|_\infty = \sum_{j>i} |a_{ij}|.$$

Aurreko desberdintzak konbinatuz, zera lortzen da:

$$(|a_{ii}| - \sum_{j<i} |a_{ij}|)\|\boldsymbol{y}\|_\infty \le \sum_{j>i} |a_{ij}|,$$

eta orduan:

$$\|\boldsymbol{T}\|_\infty = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{T}\boldsymbol{x}\|_\infty = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{y}\|_\infty \le \max_{1\le i\le n} \frac{\sum_{j>i} |a_{ij}|}{|a_{ii}| - \sum_{j<i} |a_{ij}|} = \gamma < 1. \quad \square$$

Kontuan izan baldintza hori nahikoa dela, ez beharrezkoa. Beraz, hertsiki diagonal menperatzailea ez izan arren, metodo iteratibo horiek konbergenteak izan daitezke $\rho(\boldsymbol{T}) < 1$ betetzen bada.

5.11. Problemak

Eskuz ebazteko problemak:

1. Faktorizatu matrize hauek LU deskonposizioaren bidez eta pibotatze partziala erabiliz:

$$a)\quad \begin{bmatrix} 1 & 3 & 2 \\ 1 & 5 & 3 \\ 2 & 4 & -6 \end{bmatrix}. \qquad b)\quad \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

$$c)\quad \begin{bmatrix} -5 & 2 & -1 \\ 1 & 0 & 3 \\ 3 & 1 & 6 \end{bmatrix}. \qquad d)\quad \begin{bmatrix} 2 & -1 & 1 \\ 3 & 3 & 9 \\ 3 & 3 & 5 \end{bmatrix}.$$

$$e)\quad \begin{bmatrix} 1 & 1 & 0 & 4 \\ 2 & -1 & 5 & 0 \\ 5 & 2 & 1 & 2 \\ -3 & 0 & 2 & 6 \end{bmatrix}. \qquad f)\quad \begin{bmatrix} 4 & 8 & 4 & 0 \\ 1 & 5 & 4 & -3 \\ 1 & 4 & 7 & 2 \\ 1 & 3 & 0 & -2 \end{bmatrix}.$$

   Zein dira matrize horien bat-norma eta infinitu-norma?

   Aurkitu matrize horien baldintzazko zenbakiak bi norma horietarako (hots, κ_1 eta κ_∞).

2. Ebatzi sistema lineal hauek LU deskonposizioaren bidez eta pibotatze partziala erabiliz:

$$a)\quad \begin{aligned} x_1 + 3x_2 + 2x_3 &= 5 \\ x_1 + 5x_2 + 3x_3 &= 10 \\ 2x_1 + 4x_2 - 6x_3 &= -4. \end{aligned}$$

$$b)\quad \begin{aligned} 2x_1 + x_2 + x_3 &= 3 \\ 4x_1 + 3x_2 + 3x_3 + x_4 &= 7 \\ 8x_1 + 7x_2 + 9x_3 + 5x_4 &= 17 \\ 6x_1 + 7x_2 + 9x_3 + 8x_4 &= 15. \end{aligned}$$

$$c)\quad \begin{aligned} -5x_1 + 2x_2 - x_3 &= 2 \\ x_1 + 3x_3 &= 2 \\ 3x_1 + x_2 + 6x_3 &= 2. \end{aligned}$$

$$d)\quad \begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 3x_1 + 3x_2 + 9x_3 &= 0 \\ 3x_1 + 3x_2 + 5x_3 &= 4. \end{aligned}$$

$$e)\quad \begin{aligned} x_1 + x_2 + 4x_4 &= 5 \\ 2x_1 - 1x_2 + 5x_3 &= -6 \\ 5x_1 + 2x_2 + x_3 + 2x_4 &= 3 \\ -3x_1 + 2x_3 + 6x_4 &= 4. \end{aligned}$$

$$f)\quad \begin{aligned} 4x_1 + 8x_2 + 4x_3 &= 8 \\ x_1 + 5x_2 + 4x_3 - 3x_4 &= -4 \\ x_1 + 4x_2 + 7x_3 + 2x_4 &= 10 \\ x_1 + 3x_2 - 2x_4 &= -4. \end{aligned}$$

   Kalkula ezazu sistema guztietarako $\kappa(A)_\infty$-ren behe-borne bat.

3. Izan bedi $\boldsymbol{Ux} = \boldsymbol{b}$ sistema, non $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ matrize goi-triangeluarra eta $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$ baitira. Zenbat eragiketa aritmetiko behar ditugu sistema hori ebazteko?

4. (a) Izan bedi $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizea. Zenbat biderketa/zatiketa behar ditu pibotatzerik gabeko LU deskonposizioak? (*Iradokizuna*: $1^2+2^2+\ldots+m^2 = m(m+1)(2m+1)/6$ berdintza erabili, zeina indukzioz froga baitaiteke).

   (b) Zenbat biderketa/zatiketa behar ditu aurreranzko ordezkapenak? Eta atzeranzko ordezkapenak?

   (c) Sistema lineal bat pibotatzerik gabeko LU metodoaz ebazteko, zenbat biderketa/zatiketa egin behar ditugu?

5. (a) Izan bedi $\boldsymbol{U}$ matrize ez-singular goi-triangeluarra. Frogatu $\boldsymbol{U}^{-1}$-en diagonaleko gaiak $\boldsymbol{U}$-ren diagonaleko gaien erreziprokoak direla.

   (b) Aurreko atalaren emaitza erabiliz, frogatu hau betetzen dela:

$$\|\boldsymbol{U}\|_\infty \geq \max_i |u_{ii}| \quad \text{eta} \quad \|\boldsymbol{U}^{-1}\|_\infty \geq \frac{1}{\min_i |u_{ii}|}.$$

   Infinitu-normako bi desberdintza horiek hau inplikatzen dute:

$$\kappa_\infty(\boldsymbol{U}) \geq \frac{\max_i |u_{ii}|}{\min_i |u_{ii}|}.$$

   Desberdintza horren eskuin aldeko kantitatea sarri erabiltzen dugu matrize goi-triangeluar baten baldintzazko zenbakiaren hurbilpena bezala.

6. Faktorizatu matrize simetriko hauek, $\boldsymbol{R}^T\boldsymbol{R}$ eran, Choleskyren metodoa erabiliz:

$$a) \quad \begin{bmatrix} 4 & -2 & 2 \\ -2 & 2 & -1 \\ 2 & -1 & 10 \end{bmatrix}. \qquad b) \quad \begin{bmatrix} 4 & 2 & -2 \\ 2 & 10 & 2 \\ -2 & 2 & 3 \end{bmatrix}.$$

$$c) \quad \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}. \qquad d) \quad \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}.$$

$$e) \quad \begin{bmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{bmatrix}. \qquad f) \quad \begin{bmatrix} 4 & 1 & -1 & 0 \\ 1 & 3 & -1 & 0 \\ -1 & -1 & 5 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}.$$

   Zein kasutan da matrizea hertsiki diagonal menperatzaile? Hori gertatzen denean, nolakoa da matrizea? Kalkulatu matrize horietarako baldintzazko zenbakiaren beheborne bat.

7. Aurkitu a-ren eta b-ren balio guztiak, matrize hau simetriko definitu positiboa izateko:

$$\begin{bmatrix} a & 1 & 1+b \\ 1 & a & 1 \\ 1-b^2 & 1 & a \end{bmatrix}.$$

8. Sistema hauetarako, aurkitu Jacobiren eta Gauss-Seidelen lehenengo hiru iterazioak, $\boldsymbol{x}^{(0)} = \mathbf{0}$ erabiliz:

$$a)\quad \begin{aligned} -x_1 + 3x_2 &= 1 \\ 6x_1 - 2x_2 &= 2. \end{aligned}$$

$$b)\quad \begin{aligned} 5x_1 - x_2 + x_3 &= 10 \\ 2x_1 + 8x_2 - x_3 &= 11 \\ -x_1 + x_2 + 4x_3 &= 3. \end{aligned}$$

$$c)\quad \begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4. \end{aligned}$$

$$d)\quad \begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 3x_1 + 3x_2 + 9x_3 &= 0 \\ 3x_1 + 3x_2 + 5x_3 &= 4. \end{aligned}$$

$$e)\quad \begin{aligned} 2x_1 - x_2 + 10x_3 &= -11 \\ 3x_2 - x_3 + 8x_4 &= -11 \\ 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25. \end{aligned}$$

$$f)\quad \begin{aligned} 4x_1 - 2x_2 &= 0 \\ -2x_1 + 5x_2 - x_3 &= 2 \\ -x_2 + 4x_3 + 2x_4 &= 3 \\ 2x_3 + 3x_4 &= -2. \end{aligned}$$

Beste galdera batzuk:

i) Zein da metodo bakoitzeko zehaztasun erlatiboa hirugarren iterazioan? Metodo horiek konbergenteak badira, zein baliotara konbergitzen dira? Egiaztatu hori dela sistemaren soluzioa sisteman ordezkatuz.

ii) Aztertu sistema bakoitzean iterazioekin hasi baino lehen jakin dezakegun Jacobiren metodoa konbergentea den ala dibergentea. Zergatik?

iii) Ez bada konbergente, nola bihur dezakegu konbergente (ahal bada)? Kasu horretan, ebatzi berriro sistema hori hasierako puntu berdina erabiliz.

iv) Zein metodok egiten du konbergentzia azkarren (egiten badu)? Justifikatu erantzuna konbergentziari buruzko teoria erabiliz.

v) Zein da problema bakoitzean Jacobiren metodoaren konbergentzia-ratioa?

vi) Zenbat iterazio gehienez behar ditu Jacobiren metodoak hasierako errorea 10^4 bider txikiago egiteko? Erantzuna konbergentziaren teoreman oinarritu behar da.

$$\left(\text{Oharra: } k.\text{ iterazioko zehaztasun erlatiboa} = \frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_\infty}{\|\boldsymbol{x}^{(k)}\|_\infty}\right).$$

9. Izan bedi matrize hau:

$$\boldsymbol{A} = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix}.$$

Aurkitu a-ren balioak $\boldsymbol{A}$ simetriko definitu positiboa izan dadin, baina Jacobiren iterazioa ez konbergitzeko.

10. Froga ezazu $\boldsymbol{A}$ 2×2-matrize simetriko definitu positiboa bada, Jacobiren metodoa konbergentea dela edozein hasierako baliotarako.

MATLABez ebazteko problemak:

11. Izan bedi matrize hau:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & -1 & 3 & 5 \\ 0 & 0 & 2 & 5 \\ -2 & -6 & -3 & 1 \end{bmatrix}$$

eta izan bitez matrize-norma hauek:

- $\|\boldsymbol{A}\|_1 = \max_j \|\boldsymbol{A}_{:,j}\|_1$ (zutabe guztien bat-normetako maximoa)
- $\|\boldsymbol{A}\|_\infty = \max_i \|\boldsymbol{A}_{i,:}\|_1$ (lerro guztien bat-normetako maximoa)

(a) Aztertu **eskuz** $\boldsymbol{A}$ matrize honen singulartasuna, eta kalkulatu bat-norma eta infinitu-norma.

(b) Kalkulatu $\boldsymbol{A}^{-1}$ MATLABeko `inv(A)` funtzioa erabiliz, eta kalkulatu matrize horren bat-norma eta infinitu-norma.

(c) Kalkulatu **eskuz** $\boldsymbol{A}$ matrizearen baldintzazko zenbakia, hau da: $\kappa(\boldsymbol{A}) = \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$ bi norma horietarako.

(d) Egiaztatu lortutako emaitzak MATLABeko `cond(A,1)` eta `cond(A,inf)` funtzioak erabiliz.

(e) Eranskinetik kopiatu kodea MATLABeko `lup.m` funtzio-fitxategia sortzeko, eta kalkulatu $\boldsymbol{A}$ matrizearen $\boldsymbol{L}$, $\boldsymbol{U}$ eta $\boldsymbol{P}$ matrizeak funtzio hori erabiliz.

(f) Kalkulatutako matrize horiek erabiliz (ikus (5.10) - (5.11) adierazpenak), ebatzi **eskuz** sistema hau:

$$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & -1 & 3 & 5 \\ 0 & 0 & 2 & 5 \\ -2 & -6 & -3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

12. Izan bedi $\boldsymbol{Ly} = \boldsymbol{Pb}$ sistema. Sortu MATLABeko funtzio bat `aurrerantz.m` izenekoa, adierazpen hauek kontuan hartuz:

$$\begin{aligned} x_1 &= b_1 \\ x_i &= b_i - \sum_{j=1}^{i-1} l_{ij} x_j, \quad i = 2, 3, \ldots, n-1, n, \end{aligned}$$

ikus (5.5) - (5.6) adierazpenak.

Egiaztatu aurreko problemako $\boldsymbol{Ly} = \boldsymbol{Pb}$ sistema ondo ebatzi duzula funtzio hori erabiliz.

13. Izan bedi $\boldsymbol{Ux} = \boldsymbol{y}$ sistema. Sortu MATLABeko funtzio bat `atzerantz.m` izenekoa, adierazpen hauek kontuan hartuz:

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}} \\ x_i &= \frac{b_i - \displaystyle\sum_{j=i+1}^{n} u_{ij}x_j}{u_{ii}}, \quad i = n-1, n-2, \ldots, 3, 2, 1, \end{aligned}$$

ikus (5.2) - (5.3) adierazpenak.

Egiaztatu aurreko problemako $\boldsymbol{Ux} = \boldsymbol{y}$ sistema ondo ebatzi duzula funtzio hori erabiliz.

14. Sortu M fitxategi bat, $\boldsymbol{Ax} = \boldsymbol{b}$ sistema *LU* faktorizazioaz ebazteko gai izan dadin. Programa horrek sistema hori ebazteko ondoz ondoko lan hauek egin behar ditu:

    (a) Programa horrek goiburu hau izan behar du:
    `function x=ebatzi(A,b).`

    (b) Egiaztatu $\boldsymbol{A}$ ez dela singularra, hots $\det(\boldsymbol{A}) \neq 0$. Kalkulu hori egiteko, MATLABek `det` funtzioa du. Baldin singularra bada, mezu bat emanez bukatuko da.

    (c) $\boldsymbol{LU} = \boldsymbol{PA}$ faktorizazioa egin behar du. Lan hori `lup` funtzioak egin dezake.

    (d) $\boldsymbol{Pb}$ kalkulatu behar du. Lan hori egiteko modu errazena `b(p)` idaztea da (MATLABi esker), non `p` permutazio-bektorea baita (`lup` funtzioak kalkulatzen du).

    (e) $\boldsymbol{LUx} = \boldsymbol{Pb}$ sistema ebatzi behar du, urrats hauei jarraituz:

        (L) Ebatzi $\boldsymbol{Ly} = \boldsymbol{Pb}$. Lan hori `aurrerantz` funtzioak egin dezake.

        (U) Ebatzi $\boldsymbol{Ux} = \boldsymbol{y}$. Lan hori `atzerantz` funtzioak egin dezake.

    Egiaztatu `ebatzi.m` funtzioa ondo dabilela aurreko sistema ebatziz (ez ahaztu M fitxategiak izen hori izan behar duela).

15. Izan bedi sistema hau:

$$\begin{array}{rrrrcr} 4x_1 & -2x_2 & & & = & 0 \\ -2x_1 & +5x_2 & -x_3 & & = & 2 \\ & -x_2 & +4x_3 & +2x_4 & = & 3 \\ & & 2x_3 & +3x_4 & = & -2. \end{array}$$

    (a) Garatu Jacobiren metodorako M fitxategi bat, `jacobi.m` izenekoa. Programa horrek gai izan behar du $\boldsymbol{Ax} = \boldsymbol{b}$ sistema lineala ebazteko, non $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ hertsiki diagonal menperatzailea baita. Programa horrek goiburu hau izan behar du:
    `function x=jacobi(A,b,x0,ze,imax)`,

non `x` soluzioaren hurbilpen bat baita, `x0` hasierako hurbilpen bat, eta `ze` eta `imax` eskatutako zehaztasun erlatiboa eta iterazioen kopuru maximoa baitira, hurrenez hurren. Programaren exekuzioa bukatuko da, k-garren iterazioan lortutako zehaztasun erlatiboak hau betetzen duenean:

$$\frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_\infty}{\|\boldsymbol{x}^{(k)}\|_\infty} \leq ze,$$

edo $k = imax$ denean.

(a) Asmatu zure kabuz sistema bat diagonal menperatzailekoa, eta ebatzi egindako programa erabiliz. Egiaztatu emaitza.

(b) Emandako sistema Jacobiren metodoaz ebatzi gabe, esan konbergituko den ala ez. Eta 11. problemako (f) atalekoa? Arrazoitu erantzunak.

(c) Gutxi gorabehera, zenbat iterazio erabiliko ditu hasierako errorea mila bider txikiagoa egiteko?

(d) Ebatzi emandako sistema kode hori erabiliz, eta egiaztatu aurreko analisia betetzen dela.

(b) Garatu Gauss-Seidelen metodorako MATLABeko M fitxategi bat `gaussseidel.m` izenekoa. Programa horrek gai izan behar du $\boldsymbol{Ax} = \boldsymbol{b}$ sistema lineala ebazteko, non $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ hertsiki diagonal menperatzailea baita. Programa horrek goiburu hau izan behar du:

`function x=gaussseidel(A,b,x0,ze,imax)`,

non `x` soluzioaren hurbilpen bat baita, `x0` hasierako hurbilpen bat, eta `ze` eta `imax` eskatutako zehaztasun erlatiboa eta iterazioen kopuru maximoa baitira, hurrenez hurren. Programaren exekuzioa bukatuko da k-garren iterazioan lortutako zehaztasun erlatiboak hau betetzen duenean:

$$\frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_\infty}{\|\boldsymbol{x}^{(k)}\|_\infty} \leq ze,$$

edo $k = imax$ denean.

(a) Asmatu zure kabuz sistema bat diagonal menperatzailekoa, eta ebatzi egindako programa erabiliz. Egiaztatu emaitza.

(b) Emandako sistema Gauss-Seidelen metodoaz ebatzi gabe, esan konbergituko den ala ez. Eta 11. problemako (f) atalekoa? Arrazoitu erantzunak.

(c) Gutxi gorabehera zenbat iterazio erabiliko ditu hasierako errorea mila bider txikiagoa egiteko?

(d) Ebatzi emandako sistema kode hori erabiliz, eta egiaztatu aurreko analisia betetzen dela.

ERANSKINA

Kode hau $LU = PA$ faktorizazioa kalkulatzeko erabil daiteke:

```
function [L,U,p] = lup(A)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LUP  Triangelu faktorizazioa L*U = PA bete dadin.
%   SARRERAK:
%     A deskonposatu nahi dugun matrizea,
%   EMAITZAK:
%    L matrize behe-triangeluarra,
%    U matrize goi-triangeluarra eta
%    p=permutazio-bektorea.
%
%   (P=permutazio-matrizea eman dezake goiburuan p P-rekin trukatuz.)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n,n] = size(A);
p = (1:n)';

for k = 1:n-1

   % Pibotatze partziala egiten du.
   % Aurkitu diagonaleko (k,k) gaitik behera dagoen tamaina
   % handieneko gaia (pibota) eta posizioa (m) azpibektore horretan.
   [pibota,m] = max(abs(A(k:n,k)));

   % pibotaren lerroa A matrizean:
   m = m+k-1;

   % Jauzi ezabapena zutabea 0 bada.
   if (A(m,k) ~= 0)

      % Trukatu m eta k lerroak, eta eguneratu permutazio-bektorea.
      if (m ~= k)
         A([k,m],:) = A([m,k],:);
         p([k,m]) = p([m,k]);
      end

      % Biderkatzaileak kalkulatzen ditugu, eta A matrizearen diagonal
      % azpian gordetzen ditugu. Gero L matrizeko diagonal azpian jarriko
      % ditugu.
      i = k+1:n;
      A(i,k) = A(i,k)/A(k,k);

      % Eguneratu matrizearen gainerakoa. Gero, U matrize
      % goi-triangeluarrean gordeko dugu.
      j = k+1:n;
      A(i,j) = A(i,j) - A(i,k)*A(k,j);
   end
end

% L, U eta P matrizeak idazten dira;
```

```
% tril(A,-1): A-ren diagonal azpikoa gordetzen du, bestea zero da.
% triu(A): A-ren diagonala eta bere goiko aldea gordetzen ditu, bestea
%          zero da.
L = tril(A,-1) + eye(n,n);
U = triu(A);
P=zeros(n,n);
for i=1:n
    P(i,p(i))=1;
end
```

6. kapitulua

QR faktorizazioa eta minimo karratu linealak

Kapitulu honetan problema hau ebatziko dugu:

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2,$$

non $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$ eta $m \geq n$. Demagun $\boldsymbol{A}$ zutabe hein betekoa dela. Kontuan hartu behar da kasu gaindeterminatuan, $m > n$ denean, ez dela $\boldsymbol{x}$ egoten $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ sistema zehazki bete dezan, nahiz eta biribiltze-errorerik ez egon.

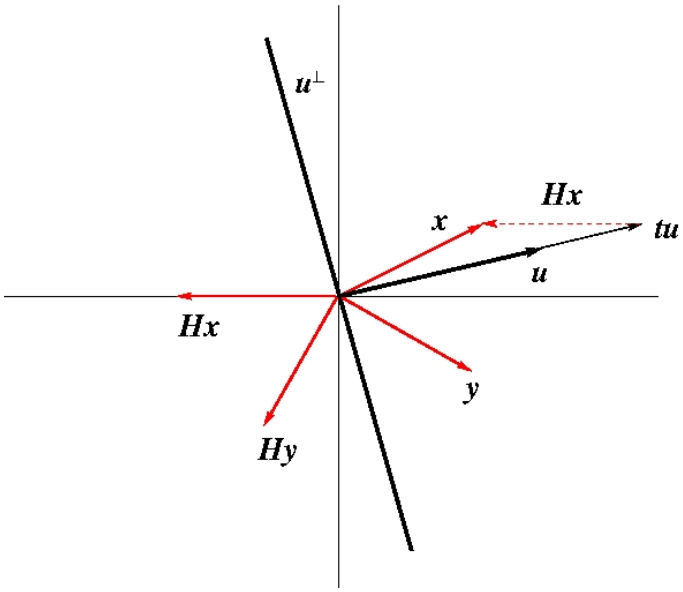
Minimo karratuen problema askotan agertzen da mundu errealeko aplikazioetan, bereziki datu-doitzea behar denean. Problema horri aurre egiteko, oso garrantzitsuak dira transformazio ortogonalak, eta horrelako tresnak azaltzen hasiko gara.

6.1. Householder-en islapenak

Householderren islapenak matrize-transformazioak dira, eta zenbakizko algoritmo moldagarri eta eraginkorrenetariko batzuen oinarria dira. Oso teknika ezaguna da matrize ortogonalen segida bat eraikitzeko, $\boldsymbol{A}$ matrize baten diagonalaren pean dauden gaiak zero bihurtzeko moduan. Beraz, $\boldsymbol{A}$ karratua bada, Householderren transformazioen bidez, matrize triangeluar bat ere lor dezakegu.

6.1. Definizioa. *Edozein* $\boldsymbol{u} \neq \boldsymbol{0}$ *bektore bati dagokion* ***Householderren islapena*** *(edo* ***Householderren transformazioa*** *edo* ***Householderren matrizea****) itxura honetako matrizea da:*

$$\boldsymbol{H} = \mathbb{1} - \rho\boldsymbol{u}\boldsymbol{u}^T, \quad non \quad \rho = \frac{2}{\|\boldsymbol{u}\|_2^2}. \tag{6.1}$$



6.1. irudia. Householderren islapena.

$\boldsymbol{u}$ *bektoreari* ***Householderren bektorea*** *deritzogu.*

$\boldsymbol{H}$ matrizeak simetrikoak ($\boldsymbol{H} = \boldsymbol{H}^T$) eta ortogonalak ($\boldsymbol{H}^{-1} = \boldsymbol{H}^T$) dira (egiaztatu). Kontuan izan $\boldsymbol{H}$ matrizea $\boldsymbol{u}$ bektorearen menpe bakarrik dagoela.

Praktikan, $\boldsymbol{H}$ ez da inoiz eratzen. Izan ere, $\boldsymbol{H}$-ren aplikazioa $\boldsymbol{x}$ bektore baten gainean egiten dugunean, hau dugu:

$$\boldsymbol{H}\boldsymbol{x} = (\mathbb{1} - \rho \boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{x} = \boldsymbol{x} - \rho \boldsymbol{u}(\boldsymbol{u}^T\boldsymbol{x})$$

eta, ondorioz, honela kalkulatzen da $\boldsymbol{H}\boldsymbol{x}$:

$$\begin{aligned} t &= \rho \boldsymbol{u}^T\boldsymbol{x} \\ \boldsymbol{H}\boldsymbol{x} &= \boldsymbol{x} - t\boldsymbol{u}. \end{aligned} \tag{6.2}$$

Geometrikoki, $\boldsymbol{x}$ bektorea $\boldsymbol{u}$ gainean proiektatzen da, eta, gero, $\boldsymbol{x}$-tik bi bider proiekzio hori kentzen da, zeren:

$$t\boldsymbol{u} = \rho(\boldsymbol{u}^T\boldsymbol{x})\boldsymbol{u} = 2\left(\frac{\boldsymbol{u}^T\boldsymbol{x}}{\|\boldsymbol{u}\|_2}\right)\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_2}.$$

6.1. irudiak erakusten ditu $\boldsymbol{u}$ bektore bat eta bere azpiespazio ortogonala (irudian, lerrozuzen bat): $\boldsymbol{u}^\perp$. Hark erakusten ditu $\boldsymbol{x}$ eta $\boldsymbol{y}$ eta beren irudiak, $\boldsymbol{H}\boldsymbol{x}$ eta $\boldsymbol{H}\boldsymbol{y}$. $\boldsymbol{H}$ matrizeak edozein bektore bere islapen bihurtzen du $\boldsymbol{u}^\perp$ lerroarekiko. Edozein $\boldsymbol{x}$ bektoretarako, bektore honek ematen digu $\boldsymbol{x}$-ren eta $\boldsymbol{H}\boldsymbol{x}$-ren arteko erdigunea:

$$\boldsymbol{x} - \frac{t}{2}\boldsymbol{u}$$

eta $\boldsymbol{u}^{\perp}$ lerroan (azpiespazioan) dago (aurrekoaren frogapena ariketa gisa geratzen da). Espazioak bi dimentsio baino gehiago dituenean, azpiespazio hori $\boldsymbol{u}$ bektorearekiko plano perpendikularra (edo hiperplano ortogonala) izango da.

Irudiak erakusten du, baita ere, zer gertatzen den $\boldsymbol{u}$ bektoreak $\boldsymbol{x}$-k eta ardatz batek osatutako angelua erdibitzen duenean. Orduan, $\boldsymbol{Hx}$ ardatz horren gainean dago. Alegia, $\boldsymbol{Hx}$-ren osagai guztiak zero dira, bat izan ezik. Gainera, $\boldsymbol{H}$ ortogonala denez, bektorearen luzera gordetzen du, zeren:

$$\|\boldsymbol{Hx}\|_2^2 = (\boldsymbol{Hx})^T(\boldsymbol{Hx}) = \boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{Hx} = \boldsymbol{x}^T\boldsymbol{x} = \|\boldsymbol{x}\|_2^2 \quad \Rightarrow \quad \|\boldsymbol{Hx}\|_2 = \|\boldsymbol{x}\|_2. \tag{6.3}$$

Ondorioz, $\boldsymbol{Hx}$ bektorearen zero ez den osagai bakarraren balioa $\pm\|\boldsymbol{x}\|_2$ da.

Orduan, $\boldsymbol{x}$ bektore baterako $\boldsymbol{Hx}$-ren k-garren osagaia izan ezik beste guztiak zero bihurtzeko, $\boldsymbol{x}$ bektorearen luzera gordez, honelako $\boldsymbol{H}$ Householderren matrizea eraiki behar dugu:

$$\begin{aligned}
\sigma &= \pm\|\boldsymbol{x}\|_2, && (6.4)\\
\boldsymbol{u} &= \boldsymbol{x} + \sigma\boldsymbol{e}_k, && (6.5)\\
\rho &= 2/\|\boldsymbol{u}\|_2^2 = 1/(\sigma u_k), && (6.6)\\
\boldsymbol{H} &= \mathbb{1} - \rho\boldsymbol{u}\boldsymbol{u}^T, && (6.7)
\end{aligned}$$

non $\boldsymbol{e}_k$ oinarri kanonikoaren k-garren bektorea baita. Biribiltze-errorearen aurrean eta (6.5) berdintza kontuan hartuz, hobe da σ-ren zeinua x_k-renaren berdina hartzea; hau da, $\sigma = \text{zeinu}(\boldsymbol{x}_k)\|\boldsymbol{x}\|_2$.

(6.6) berdintzan, ohartu hau betetzen dela:

$$\begin{aligned}
\|\boldsymbol{u}\|_2^2 &= \boldsymbol{u}^T\boldsymbol{u} = (\boldsymbol{x} + \sigma\boldsymbol{e}_k)^T(\boldsymbol{x} + \sigma\boldsymbol{e}_k)\\
&= \boldsymbol{x}^T\boldsymbol{x} + \sigma^2\boldsymbol{e}_k^T\boldsymbol{e}_k + \sigma\boldsymbol{x}^T\boldsymbol{e}_k + \sigma\boldsymbol{e}_k^T\boldsymbol{x}\\
&= \|\boldsymbol{x}\|_2^2 + \sigma^2 + 2\sigma x_k\\
&= 2\sigma^2 + 2\sigma x_k \qquad \text{dugu (6.4) bidez}\\
&= 2\sigma(x_k + \sigma)\\
&= 2\sigma u_k \qquad \text{dugu (6.5) erabiliz,}
\end{aligned}$$

non u_k zenbakia $\boldsymbol{u}$ bektorearen k-garren osagaia baita.

Azkenik, (6.2) eta (6.5) berdintzak kontuan hartuz, hau lortzen da:

$$\begin{aligned}
\boldsymbol{Hx} &= \boldsymbol{x} - \rho(\boldsymbol{u}^T\boldsymbol{x})\boldsymbol{u}\\
&= \boldsymbol{x} - 2\frac{\boldsymbol{x}^T\boldsymbol{x} + \sigma\boldsymbol{x}^T\boldsymbol{e}_k}{\|\boldsymbol{u}\|_2^2}\boldsymbol{u}\\
&= \boldsymbol{x} - 2\frac{\sigma^2 + \sigma x_k}{2\sigma u_k}\boldsymbol{u}\\
&= \boldsymbol{x} - \boldsymbol{u}\\
&= -\sigma\boldsymbol{e}_k.
\end{aligned} \tag{6.8}$$

Hots, $\boldsymbol{H}$ transformazioak $\boldsymbol{x}$ garraiatzen du k. ardatzera. Zer gertatzen da $\sigma = -\text{zeinu}(\boldsymbol{x}_k)\|\boldsymbol{x}\|_2$ hartzen bada?

6.2. QR faktorizazioa

6.2.1. $\boldsymbol{A}$ matrize karratu ez-singularra

Demagun $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizea ez dela singularra; orduan, aurreko atalaren arabera, $n-1$ Householderren matrizeak eraiki ditzakegu hau bete dadin:

$$\boldsymbol{H}_{n-1}\dots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A} = \boldsymbol{R}, \tag{6.9}$$

non $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ matrizea goi-triangeluarra baita.

Prozesu honen lehenengo urratsa $\boldsymbol{H}_1$ matrizea eraikitzea da, $\boldsymbol{A}$-ren lehenengo zutabea, $\boldsymbol{a}_1$, $\boldsymbol{e}_1$-en multiplo bihurtzeko ($\boldsymbol{e}_1$ oinarri kanonikoaren lehenengo bektorea da), zutabearen norma euklidearra gordez. Alegia, 2. gaitik n. gaira dauden gaiak zero bihurtuz. Orduan, (6.8) kontuan hartuz, zera lortzen da:

$$\boldsymbol{H}_1\boldsymbol{a}_1 = (\mathbb{1} - \rho_1\boldsymbol{u}_1\boldsymbol{u}_1^T)\boldsymbol{a}_1 = -\sigma_1\boldsymbol{e}_1 = \begin{bmatrix} r_{11} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{6.10}$$

non $r_{11} = -\sigma_1$ eta $\sigma_1 = \text{zeinu}(a_{11})\|\boldsymbol{a}_1\|_2$ (biribiltze-erroreak saihesteko). Aurreko atalagatik, badakigu $\boldsymbol{u}_1 = \boldsymbol{a}_1 + \sigma_1\boldsymbol{e}_1$ hartu behar dugula. Beraz, Householderren lehenengo bektorea hau izango da:

$$\boldsymbol{u}_1 = \begin{bmatrix} a_{11} + \sigma_1 \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}. \tag{6.11}$$

Bektore horrekin, $\boldsymbol{A}$ matrizean $\boldsymbol{H}_1$ aplikatuz, zera dugu:

$$\boldsymbol{A}^{(2)} = \boldsymbol{H}_1\boldsymbol{A} = \begin{bmatrix} r_{11} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} \end{bmatrix}, \tag{6.12}$$

non $\boldsymbol{A}$ matrizearen gai guztiak aldatu baitira. Hori ez da gertatzen ezabatze gaussiarrean; metodo horretan, lehenengo lerroa ez da aldatzen. Gainerako matrizea, $\widetilde{\boldsymbol{A}}_2$, lehenengo lerroa eta lehenengo zutabea kenduz geratzen den $(n-1)\times(n-1)$ matrizea da.

6.1. adibidea. *Egin dezagun Householderren ezabatzearen lehenengo urratsa matrize honetarako:*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 3 & -1 & -1 \end{bmatrix}.$$

Ebazpena. (6.4) eta (6.10) aplikatuz, hau dugu:

$$\|\boldsymbol{a}_1\|_2 = \sqrt{14} = 3.742, \quad \sigma_1 = \text{zeinu}(+1)3.742 = +3.742, \quad r_{11} = -3.742$$

eta (6.11) -ren bitartez

$$\boldsymbol{u}_1 = \begin{bmatrix} 1 + 3.742 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4.742 \\ 2 \\ 3 \end{bmatrix}.$$

$\boldsymbol{H}_1$ eta $\boldsymbol{A}^{(2)}$ kalkulatzeko, (6.6) - (6.7) eta (6.12) erabiliz, hurrenez hurren, emaitza hauek ditugu:

$$\rho_1 = 1/(3.742 \cdot 4.742) = 0.05637$$

eta

$$\boldsymbol{H}_1 = \mathbb{1} - \rho_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T = \begin{bmatrix} -0.2673 & -0.5345 & -0.8018 \\ -0.5345 & 0.7745 & -0.3382 \\ -0.8018 & -0.3382 & 0.4927 \end{bmatrix},$$

$$\boldsymbol{A}^{(2)} = \boldsymbol{H}_1 \boldsymbol{A} = \begin{bmatrix} -3.742 & -1.069 & -0.2673 \\ 0 & 2.127 & 0.04368 \\ 0 & -2.309 & -2.434 \end{bmatrix},$$

non zenbaki guztiak lau zifra esanguratsutara biribildu baititugu. □

Bigarren Householderren transformazioaren eraikuntzan, helburua da $\widetilde{\boldsymbol{A}}_2$-ren lehenengo zutabea egokiro eraldatzea, $\boldsymbol{A}^{(2)}$-ren lehenengo lerroa eta lehenengo zutabea aldatu barik. Hori lortzeko, nahikoa da $\boldsymbol{u}_2$-ren lehenengo gaia zero hartzea; ikus (6.2) -ren bigarren berdintza. Aukera horrekin; $\boldsymbol{H}_2$-ren aplikazioak bektore orokor bati ez dio aldatzen lehenengo osagaia, eta $\boldsymbol{H}_2$-ren aplikazioak berdin uzten dio $\boldsymbol{e}_1$-en edozein multiplori.

Aurreko adibidera itzuliz:

$$\widetilde{\boldsymbol{A}}_2 = \begin{bmatrix} 2.127 & 0.04368 \\ -2.309 & -2.434 \end{bmatrix}.$$

Orain, $\widetilde{\boldsymbol{a}}_2$ lehenengo zutabea da eta

$$\|\widetilde{\boldsymbol{a}}_2\|_2 = 3.139, \quad \sigma_2 = \text{zeinu}(+2.127)3.139 = +3.139, \quad r_{22} = -3.139,$$

gainera, (6.5) erabiliz:

$$\boldsymbol{u}_2 = \begin{bmatrix} 0 \\ 2.127 + 3.139 \\ -2.309 \end{bmatrix} = \begin{bmatrix} 0 \\ 5.266 \\ -2.309 \end{bmatrix}.$$

$\boldsymbol{H}_2$ eta $\boldsymbol{A}^{(3)}$ kalkulatzeko, (6.6) - (6.7) eta (6.12) erabiliz, hurrenez hurren, emaitza hauek ditugu:

$$\rho_2 = 1/(3.139 \cdot 5.266) = 0.06050$$

eta

$$\boldsymbol{H}_2 = \mathbb{1} - \rho_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.678 & 0.7357 \\ 0 & 0.7357 & 0.6775 \end{bmatrix}, \quad \boldsymbol{A}^{(3)} = \boldsymbol{H}_2 \boldsymbol{A}^{(2)} = \begin{bmatrix} -3.742 & -1.069 & -0.2673 \\ 0 & -3.139 & -1.820 \\ 0 & 0 & -1.617 \end{bmatrix}.$$

Ondorioz, hau lortu dugu:

$$\boldsymbol{H}_2 \boldsymbol{H}_1 \boldsymbol{A} = \boldsymbol{R},$$

non $\boldsymbol{R} = \boldsymbol{A}^{(3)} \in \mathbb{R}^{3\times 3}$ matrize goi-triangeluarra baita. □

Baldin $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ ez bada singularra, Householderren ezabapenaren $n-1$ urrats egin ditzakegu, eta $\widetilde{\boldsymbol{A}}_i$ gainerako matrizearen $\widetilde{\boldsymbol{a}}_i$ lehenengo zutabea ez da zero izango i-garren urrats bakoitzean, $i = 1, \ldots, n-1$ ($\widetilde{\boldsymbol{A}}_1 = \boldsymbol{A}$ da); eta azken zutabean ez ditugu zeroak egin behar. Orduan, hau dugu:

$$\boldsymbol{H}_{n-1} \ldots \boldsymbol{H}_1 \boldsymbol{A} = \boldsymbol{R},$$

non $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ matrize goi-triangeluarra baita. Izan bedi $\boldsymbol{Q}^T \in \mathbb{R}^{n\times n}$ matrize ortogonal hau:

$$\boldsymbol{Q}^T = \boldsymbol{H}_{n-1} \ldots \boldsymbol{H}_1. \tag{6.13}$$

Beraz,

$$\boldsymbol{Q} = \boldsymbol{H}_1 \ldots \boldsymbol{H}_{n-1}. \tag{6.14}$$

Honako adierazpen bakoitzari $\boldsymbol{A}$-ren *QR faktorizazio* deritzo:

$$\boldsymbol{Q}^T \boldsymbol{A} = \boldsymbol{R} \quad \text{edo} \quad \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}. \tag{6.15}$$

Praktikan, ez dugu $\boldsymbol{Q}$ kalkulatu behar sistema bat ebazteko, zeren $\boldsymbol{Q}^T\boldsymbol{v}$ matrize-bektore biderketetan bakarrik agertzen baita. Bektore hori kalkula daiteke (6.13) adierazpeneko $\boldsymbol{H}_{n-1}, \ldots, \boldsymbol{H}_1$ Householderren banako transformazioak aplikatuz; matrize horiek ere ez dira esplizituki kalkulatu behar.

k-garren Householderren matrizea adierazteko, $\boldsymbol{u}_k$ Householderren bektorearen $n-(k-1)$ osagai gorde behar ditugu, eta ρ_k balioa. Nahiz eta ρ_k kalkula dezakegun $\boldsymbol{u}_k$ erabiliz, bere kalkulurako eragiketak ez errepikatzeko, gordetzen dugu.

Faktorizazioa	Biderketak/Zatiketak	Batuketak/Kenketak
$\boldsymbol{A} = \boldsymbol{PLU}$	$n^3/3$	$n^3/3$
$\boldsymbol{A} = \boldsymbol{QR}$	$2n^3/3$	$2n^3/3$
$\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$	$n^3/6$	$n^3/6$

6.1. taula. Matrize-faktorizazioen kostu aritmetikoa.

6.2.1. taularen arabera, QR faktorizazioak LU faktorizazioaren eragiketen bikoitza erabiltzen du, hau da, lan bikoitza egin behar du matrize bat faktorizatzeko. Hala ere, askotan, QR faktorizazioa aukeratzen da bere zenbakizko egonkortasunagatik. $\boldsymbol{Q}$ ortogonala denez $\kappa_2(\boldsymbol{R}) = \kappa_2(\boldsymbol{A})$; hau da, QR faktorizazioak ez du okerragotzen $\boldsymbol{A}$-ren baldintza. Frogatu hori ariketa gisa.

6.2.2. Sistema lineal karratu determinatu baten ebazpena

Behin $\boldsymbol{A}$-ren QR faktorizazioa ezagutu eta gero, $\boldsymbol{Ax} = \boldsymbol{b}$ kalkula daiteke, hau kontuan hartuz:

$$\boldsymbol{Ax} = \boldsymbol{QRx} = \boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{Rx} = \boldsymbol{Q}^T\boldsymbol{b}. \tag{6.16}$$

Beraz, lehenengo $\boldsymbol{Q}^T\boldsymbol{b}$ kalkulatzen da, eta, gero, (6.16) sistema ebazten dugu.

Aurreko 6.1. adibideari jarraituz, sistema hau dugu:

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 3 & -1 & -1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 3 \\ 2 \\ 6 \end{bmatrix},$$

orduan:

$$\boldsymbol{R} = \begin{bmatrix} -3.742 & -1.069 & -0.2673 \\ 0 & -3.139 & -1.820 \\ 0 & 0 & -1.617 \end{bmatrix}, \quad \boldsymbol{Q}^T\boldsymbol{b} = \begin{bmatrix} -6.682 \\ 1.320 \\ -1.617 \end{bmatrix}$$

(lau zifra esanguratsutara biribildua). Atzeranzko ebazpena eginez, (6.16) sisteman, $\boldsymbol{x} = (2, -1, 1)^T$ emaitza lortzen da ($\boldsymbol{R}$ eta $\boldsymbol{Q}^T\boldsymbol{b}$-ren bertsio zehatzekin!). □

6.3. Givens-en biraketak

Householderren islapenak arras erabilgarriak dira bektore batean zero asko sartzeko. Hala ere, badaude kalkulu batzuk non zeroak era selektiboago batean sartu behar baititugu; orduan *Givensen biraketak* aukeratuko ditugu, eta haiek dira honelako identitaterako biheineko zuzenketak:

$$G(i,k,\theta) = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \begin{matrix} \\ \\ i \\ \\ k \\ \\ \\ \end{matrix}, \qquad \begin{matrix} \\ \\ i \\ \\ k \\ \\ \\ \end{matrix}$$

non θ angelu baterako $c = \cos(\theta)$ eta $s = \sin(\theta)$. Givensen biraketak direnez, matrize ortogonalak dira (egiaztatu). $\boldsymbol{x}$ bektore bat $G(i,k,\theta)^T$ matrizeaz aurrebiderkatzea (i,k) koordenatu-planoan, θ radianeko erlojuaren kontrako noranzko biraketa bat ematea da. Hots, $\boldsymbol{y} = G(i,k,\theta)^T\boldsymbol{x}$ bada, orduan:

$$y_j = \begin{cases} cx_i - sx_k, & j = i \\ sx_i + cx_k, & j = k \\ x_j, & j \neq i,k. \end{cases}$$

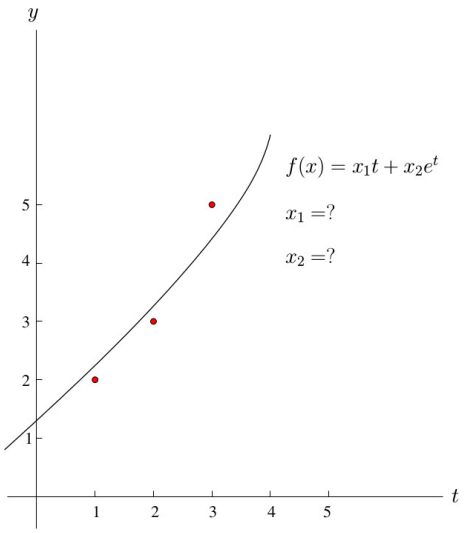
Formula horietatik $y_k = 0$ izatera behartu dezakegu balio hauek hartuz:

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \qquad s = \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}.$$

6.2. adibidea. $\boldsymbol{x} = [1, 2, 3, 4]^T$, $\cos(\theta) = 1/\sqrt{5}$ *eta* $\sin(\theta) = -2/\sqrt{5}$ *badira, orduan* $\boldsymbol{y} = G(2, 4, \theta)\boldsymbol{x} = [1, \sqrt{20}, 3, 0]^T$.

6.4. Minimo karratu linealak: sistema gaindeterminatuak

Demagun m datu-puntu dauzkagula, (t_i, y_i), $f(\boldsymbol{x}, t)$ funtzio batekin, lineala dena $x_1, x_2, \ldots, x_n$ parametro askeekiko. Adibidez, demagun $(1, 2)$, $(2, 3)$ eta $(3, 5)$ hiru bikoteei $f(\boldsymbol{x}, t) = x_1 t + x_2 e^T$ funtzioa egokitu nahi diegula (6.2. irudia).



6.2. irudia. Minimo karratu linealen adibidea.

Doikuntza zehatz bat lortzeko, hots:

$$\begin{aligned} f(\boldsymbol{x}, t_1) &= y_1 \\ f(\boldsymbol{x}, t_2) &= y_2 \\ f(\boldsymbol{x}, t_3) &= y_3, \end{aligned}$$

$\boldsymbol{Ax} = \boldsymbol{b}$ sistema bete behar du, non:

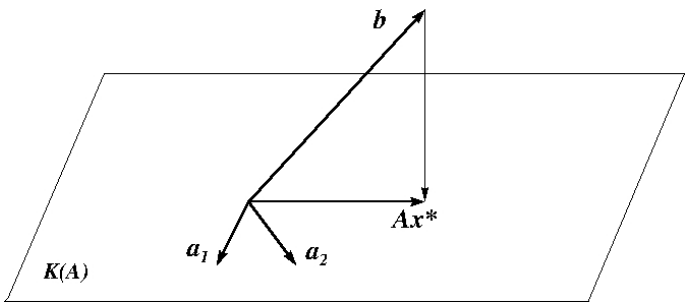
$$\boldsymbol{A} = \begin{bmatrix} 1 & e \\ 2 & e^2 \\ 3 & e^3 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{eta} \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}.$$

Sistema lineal hau *gaindeterminatua* denez ($m > n$ zentzuan), ezin dugu ebatzi ezagutzen ditugun metodoez. Izan ere, guk aukeratuko dugu $\boldsymbol{x}$ bektorea $\boldsymbol{Ax} - \boldsymbol{b}$ hondar-bektorearen neurriren bat minimizatzeko. Izan bedi bi-norma aukeratutako neurria. Beraz, problema hau ebatzi behar dugu:

$$\min_{x \in \mathbb{R}^n} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2, \tag{6.17}$$

non $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m \geq n$, $\boldsymbol{b} \in \mathbb{R}^m$, gure adibidean $m = 3$ eta $n = 2$.

Adibidearen soluzioa kalkulatzea eta «$(2, 3, 5)^T$ bektoretik $(1, 2, 3)^T$ eta $(e, e^2, e^3)^T$ bektoreen konbinazio lineal hurbilena aurkitzea (bi-norman)», problema baliokideak dira. Orokorrean, (6.17) problema ebaztea honela uler daiteke: «Aurkitu $\boldsymbol{b}$-tik $\boldsymbol{A}$ zutabeen konbinazio lineal hurbilena bi-norman». Geometrikoki, horrek esan nahi du $\boldsymbol{b}$ bektoretik $\boldsymbol{A}$-ren zutabeek sortutako $K(\boldsymbol{A})$ azpiespazioko n dimentsioko bektore hurbilena (norma euklidearrean) aurkitzea; ikus 6.3. irudia. Izan bitez $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^n$ bektoreak $\boldsymbol{A}$ matrizeko m zutabeak.



6.3. irudia. Minimo karratu linealen soluzioa.

Badakigu $\boldsymbol{b}$-tik $K(\boldsymbol{A})$ azpiespazioko bektore hurbilena $\boldsymbol{Ax}^* \in K(\boldsymbol{A})$ izango dela; izan ere, $\boldsymbol{Ax}^* - \boldsymbol{b}$ perpendikularra da $K(\boldsymbol{A})$ azpiespazioarekin. Beraz, x^*-k hau bete behar du:

$$\boldsymbol{a}_i^T(\boldsymbol{Ax}^* - \boldsymbol{b}) = 0, \quad i = 1, \ldots, n,$$

edo, baliokideki,

$$\boldsymbol{A}^T(\boldsymbol{Ax}^* - \boldsymbol{b}) = \boldsymbol{0}$$

Bistan dago $\boldsymbol{Ax}^*$ bakarra dela, eta $\boldsymbol{A}$-ren zutabeak linealki askeak badira, $\boldsymbol{x}^*$ bektorea ere bakarra dela. Ondorioz, $\boldsymbol{x}^*$ ekuazio-sistema honen soluzioa da:

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b},$$

bateragarria dena.

Informazio hori hurrengo teoreman finkatzen da.

6.1. teorema. *Izan bitez $m \geq n > 0$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$. Orduan, (6.17) minimo karratuen problemaren soluzioa $\{\boldsymbol{x}^* \mid \boldsymbol{A}^T(\boldsymbol{Ax}^* - \boldsymbol{b}) = \boldsymbol{0}\}$ bektoreen multzoa da.*

Baldin $\boldsymbol{A}$-ren zutabeak linealki askeak badira, $\boldsymbol{x}^*$ soluzio bakarra da, $\boldsymbol{A}^T\boldsymbol{A}$ ez da singularra eta $\boldsymbol{x}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$.

Frogantza. Defini dezagun $f(\boldsymbol{x})$ funtzio hau:

$$\begin{aligned} f(\boldsymbol{x}) &= \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 = (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) \\ &= (\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x}) - (\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{b} - \boldsymbol{b}^T(\boldsymbol{A}\boldsymbol{x}) + \boldsymbol{b}^T\boldsymbol{b} \\ &= \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b}. \end{aligned}$$

Orduan, $\min \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2 = \min f(\boldsymbol{x})$ eta soluzioak $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$ bete behar du; alegia:

$$\nabla f(\boldsymbol{x}^*) = 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}^* - 2\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{0},$$

ondorioz, $\boldsymbol{x}^*$ soluzioak ekuazio sistema hau bete behar du:

$$\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) = \boldsymbol{0}. \tag{6.18}$$

eta $\boldsymbol{A}^T\boldsymbol{A}$, behintzat, erdidefinitu positiboa denez, $f(\boldsymbol{x})$ funtzio konbexua da, eta, ondorioz, $\boldsymbol{x}^*$ minimizatzaile bat da. Beraz, teoremaren lehenengo atala frogatuta dago. Bigarrenean, $\boldsymbol{A}$-ren zutabeak linealki askeak direnez, $\boldsymbol{A}^T\boldsymbol{A}$ ez da singularra, eta orduan $\boldsymbol{x}^*$ minimizatzaile bakarra da. □

Berrordenatuz (6.18), ekuazio hauek lortzen dira:

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}, \tag{6.19}$$

ekuazio normalak deritzenak. Ekuazio normalen matrizea, $\boldsymbol{A}^T\boldsymbol{A}$, edozein $\boldsymbol{A}$-tarako matrize simetrikoa eta erdidefinitu positiboa da, eta definitu positiboa da baldin eta soilik baldin $\boldsymbol{A}$ matrizearen zutabeak linealki askeak badira, hots, $\boldsymbol{A}$ matrizea *zutabe hein betekoa* bada (horren frogapena ariketa gisa geratzen da). Bestalde, ekuazio normalak beti dira bateragarriak, $\boldsymbol{A}^T\boldsymbol{A}$ singularra izan arren.

Ekuazio normalek garrantzi praktiko handia dute, haiek bide zuzen bat ematen baitute minimo karratuen soluzioa kalkulatzeko. $\boldsymbol{A}^T\boldsymbol{A}$ definitu positiboa denean ($\boldsymbol{A}$ hein betekoa denean), ekuazio normalek soluzio bakar bat dute eta $\boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$ da. $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \in \mathbb{R}^{n\times m}$ matrizea ezaguna da $\boldsymbol{A}$ matrizearen *sasialderantzizkoa* izenaz. $\boldsymbol{A}$ hein betekoa denean, minimo karratuen problema ebatz dezakegu algoritmo honekin.

6.1. algoritmoa. Hein beteko minimo karratu linealen problemaren ebazpena, ekuazio normalen bitartez.

0 urratsa. SARRERA. Sartu: $\boldsymbol{A}$ eta $\boldsymbol{b}$.

1 urratsa. Eratu ekuazio normalen matrizea: $\boldsymbol{A}^T\boldsymbol{A}$ eta $\boldsymbol{A}^T\boldsymbol{b}$ bektorea.

2 urratsa. Kalkula ezazu Choleskyren faktorizazioa: $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$, non $\boldsymbol{R}$ goi-triangeluarra baita.

3 urratsa. Ebatzi $\boldsymbol{R}^T\boldsymbol{y} = \boldsymbol{A}^T\boldsymbol{b}$ aurreranzko ordezkapenaz; gero, ebatzi $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{y}$ atzeranzko ordezkapenaz.

4 urratsa. IRTEERA. Emaitzak: $\boldsymbol{x}$.

6.3. adibidea. *Ebatzi* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *sistema gaindeterminatua, hau badugu:*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \textit{eta} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix}.$$

Ebazpena. $\boldsymbol{A}$ matrizearen zutabeak bektore askeak direnez, $\boldsymbol{A}$ hein betekoa da.

$\boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ eta $\boldsymbol{A}^T\boldsymbol{b} = \begin{bmatrix} 1 \\ -4 \end{bmatrix}$ eta $(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x}^* = \boldsymbol{A}^T\boldsymbol{b}$ sistemaren soluzioa $\boldsymbol{x}^* = (2, -3)^T$ da.

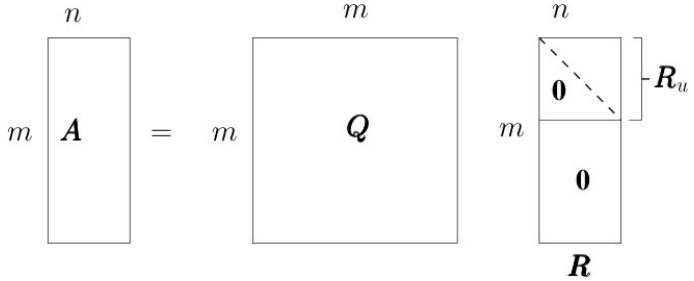
Hau da soluzio horri dagokion *hondar-bektorea*:

$$\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -3 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ 2 \end{bmatrix}.$$

Beraz, hau da sistema horren hondar-bektorearen norma euklidearra: $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2 = \sqrt{12} = 2\sqrt{3} \approx 3.464$ (ikus 6.3. irudia). □

Ariketa moduan, ebatzi ekuazio normalen bidez atal honen hasierako $f(\boldsymbol{x}, t) = x_1 t + x_2 e^T$ funtzioaren parametroen kalkulua, funtzioa $(1, 2)$, $(2, 3)$ eta $(3, 5)$ bikoteei egokitzeko.

Nahiz eta $\boldsymbol{A}$ hein betekoa izan eta, beraz, ekuazio normalek $\boldsymbol{x}^*$ soluzio bakar bat izan, metodo hori erabiltzea ez da beti egokiena minimo karratuen problema ebazteko. Hori da $\boldsymbol{A}^T\boldsymbol{A}$ matrizearen baldintza $\kappa_2(\boldsymbol{A}^T\boldsymbol{A}) = (\kappa_2(\boldsymbol{A}))^2$ betetzeagatik (egiaztatu ariketa gisa). Adibidez, $\kappa_2(\boldsymbol{A}) = 10^3$ baldintza ez da oso txarra, baina $\kappa_2(\boldsymbol{A}^T\boldsymbol{A}) = 10^6$ askoz txarragoa da eta zenbakizko egonkortasunaren arazo larriak sor ditzake. Beraz, metodo horren baldintza $\boldsymbol{A}$ matrizearena baino okerragoa da. Aldiz, guk $\boldsymbol{A}$-ren QR deskonposizioa erabil dezakegu, honela:



(6.20)

($\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ ortogonala, $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ goi-triangeluarra), faktorizazio hori eginez Householderren islapenen bitartez, $\boldsymbol{A}$ matrizea karratua eta ez-singularra den kasuan bezala. Deskonposizio horrek ere ematen du $\boldsymbol{A}$-ren zutabeen ortonormalizazio bat zenbaki egonkorra dela. Honako teorema honek erakusten du nola erabili (6.20) faktorizazioa (6.17) problema ebazteko.

6.2. teorema. *Izan bitez $m \geq n > 0$, $\boldsymbol{b} \in \mathbb{R}^m$ bektorea eta $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ zutabe hein beteko matrizea. Orduan, (6.20) erako $\boldsymbol{A} = \boldsymbol{QR}$ deskonposizio bat existitzen da, non $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ matrize ortogonala baita, $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ goi-triangeluarra eta $\boldsymbol{R}_u$ (hots, $\boldsymbol{R}$-ren lehenengo n lerroak) matrize goi-triangeluarra eta ez-singularra.*

Gainera, hau da (6.17) minimo karratuen problemaren soluzio bakarra:

$$\boldsymbol{x}^* = \boldsymbol{R}_u^{-1}(\boldsymbol{Q}^T\boldsymbol{b})_u, \qquad non \quad (\boldsymbol{Q}^T\boldsymbol{b})_u^T = \left((\boldsymbol{Q}^T\boldsymbol{b})_1, \ldots, (\boldsymbol{Q}^T\boldsymbol{b})_n\right),$$

eta hondarraren luzeraren karratua hau da:

$$\min_{\boldsymbol{x}} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 = \sum_{i=n+1}^{m} (\boldsymbol{Q}^T\boldsymbol{b})_i^2.$$

Frogantza. Householderren transformazioetatik ondorioztatzen da $\boldsymbol{A}$-ren QR deskonposizioaren existentzia, eta $\boldsymbol{A}$-ren zutabe hein betekoa izatetik $\boldsymbol{R}_u$-ren ez-singularitatea ($\boldsymbol{Q}$ ez baita singularra). Orain, $\boldsymbol{Q}^T$ ortogonala denez, bektore batez biderkatzean ez dio aldatzen bere norma euklidearra ($\|\boldsymbol{Q}^T\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$) eta, orduan, zera dugu:

$$\|\boldsymbol{Ax} - \boldsymbol{b}\|_2 = \|\boldsymbol{QRx} - \boldsymbol{b}\|_2 = \|\boldsymbol{Q}^T(\boldsymbol{QRx} - \boldsymbol{b})\|_2 = \|\boldsymbol{Rx} - \boldsymbol{Q}^T\boldsymbol{b}\|_2,$$

hori dela eta, honela berridatz dezakegu (6.17) problema:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{Rx} - \boldsymbol{Q}^T\boldsymbol{b}\|_2.$$

Orduan, $(\boldsymbol{Q}^T\boldsymbol{b})_l^T = ((\boldsymbol{Q}^T\boldsymbol{b})_{n+1}, \ldots, (\boldsymbol{Q}^T\boldsymbol{b})_m)$ bada, hau dugu:

$$\|\boldsymbol{Rx} - \boldsymbol{Q}^T\boldsymbol{b}\|_2^2 = \|\boldsymbol{R}_u\boldsymbol{x} - (\boldsymbol{Q}^T\boldsymbol{b})_u\|_2^2 + \|(\boldsymbol{Q}^T\boldsymbol{b})_l\|_2^2$$

eta hori minimizatzen da $\boldsymbol{x} = \boldsymbol{R}_u^{-1}(\boldsymbol{Q}^T\boldsymbol{b})_u$ denean (lehenengo batugaia zero egiten baita), gainera $\|(\boldsymbol{Q}^T\boldsymbol{b})_l\|_2^2 = \sum_{i=n+1}^m (\boldsymbol{Q}^T\boldsymbol{b})_i^2$ dugu. □

6.2. algoritmoa. Hein beteko minimo karratu linealen problemaren ebazpena, QR faktorizazioaren bitartez.

0 urratsa. SARRERA. Sartu: $\boldsymbol{A}$ eta $\boldsymbol{b}$.

1 urratsa. Kalkula ezazu $\boldsymbol{A}$-ren QR faktorizazioa Householderren transformazioak erabiliz, $\boldsymbol{A} = \boldsymbol{QR}$, non $\boldsymbol{Q}$ ortogonala baita eta $\boldsymbol{R}$ goi-triangeluarra.

2 urratsa. Eratu $\widetilde{\boldsymbol{b}} = \boldsymbol{Q}^T\boldsymbol{b}$ eta $\widetilde{\boldsymbol{b}}_u$ (hots, $\widetilde{\boldsymbol{b}}$-ren lehenengo n osagaiak).

3 urratsa. Ebatzi $\boldsymbol{R}_u\boldsymbol{x} = \widetilde{\boldsymbol{b}}_u$ atzeranzko ordezkapenaz, non $\boldsymbol{R}_u$ $\boldsymbol{R}$-ren lehenengo n lerroek osatutako azpimatrize karratu goi-triangeluarra baita.

4 urratsa. IRTEERA. Emaitza: $\boldsymbol{x}$.

Alegia, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrizea zutabe hein betekoa ($m > n$) eta $\boldsymbol{b} \in \mathbb{R}^m$ badira, eta $\boldsymbol{H}_n \dots \boldsymbol{H}_1\boldsymbol{A} = \boldsymbol{R}$ bada, $\boldsymbol{H}_n \dots \boldsymbol{H}_1\boldsymbol{b}$ kalkulatu behar dugu. Orduan, $\boldsymbol{Q}^T[\boldsymbol{A} \quad \boldsymbol{b}]$ guztia batera kalkulatuko dugu, $\boldsymbol{Q}^T = \boldsymbol{H}_n \dots \boldsymbol{H}_1$ izanik. Gainera, gogoratu $\boldsymbol{Hx}$ kalkulatzeko ez dugula $\boldsymbol{H}$ aurkitu behar.

6.4. adibidea. *Adibide honetan, QR metodoaren bitartez ebatziko da aurreko 6.3. adibidearen $\boldsymbol{Ax} = \boldsymbol{b}$ sistema gaindeterminatua.*

Ebazpena. Gogora dezagun honako hau:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{eta} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix}.$$

Izan bitez $\boldsymbol{a}_1 = [1 \quad 1 \quad 0]^T$ eta $\boldsymbol{a}_2 = [1 \quad 0 \quad 1]^T$ $\boldsymbol{A}$ matrizearen zutabe-bektoreak. Orduan, lau digitu esanguratsutara biribilduz, hau dugu:

$$\sigma_1 = \text{zeinu}(a_{11})\|\boldsymbol{a}_1\|_2 = +\sqrt{2} = 1.414, \qquad \boldsymbol{u}_1 = \begin{bmatrix} 1 + 1.414 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.414 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{eta} \qquad \rho_1 = \frac{1}{\sigma_1 \cdot [\boldsymbol{u}_1]_1} = \frac{1}{1.414 \cdot 2.414} = 0.2929.$$

Beraz, (6.2) adierazpenak erabiliz, hau lortzen dugu:

$$\boldsymbol{H}_1\boldsymbol{a}_1 = \boldsymbol{a}_1 - \rho_1(\boldsymbol{u}_1^T\boldsymbol{a}_1)\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - 0.2929 \left([2.414 \quad 1 \quad 0] \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right) \begin{bmatrix} 2.414 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1.414 \\ 0 \\ 0 \end{bmatrix},$$

espero genuen bezala, ikus (6.10) eta (6.11) adierazpenak (hots, praktikan kalkulu horiek ez ditugu egin behar). Orain $\boldsymbol{H}_1\boldsymbol{a}_2$ kalkulatuko dugu:

$$\boldsymbol{H}_1\boldsymbol{a}_2 = \boldsymbol{a}_2 - \rho_1(\boldsymbol{u}_1^T\boldsymbol{a}_2)\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - 0.2929 \left([2.414 \quad 1 \quad 0] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right) \begin{bmatrix} 2.414 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ -0.7071 \\ 1 \end{bmatrix}.$$

Hori dela eta, hau dugu:

$$\boldsymbol{A}^{(2)} = \begin{bmatrix} -1.414 & -0.7071 \\ 0 & -0.7071 \\ 0 & 1 \end{bmatrix}$$

($\boldsymbol{A}^{(1)} = \boldsymbol{A}$ baita).

Ikusi den bezala, ez dugu $\boldsymbol{H}_1$ kalkulatzeko beharrik. Hala ere, kalkulatu egingo dugu:

$$\boldsymbol{H}_1 = \mathbb{1} - \rho_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T = \begin{bmatrix} -0.7071 & -0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(ariketa gisa, egiaztatu $\boldsymbol{H}_1\boldsymbol{A} = \boldsymbol{A}^{(2)}$ betetzen dela).

Gainera, $\boldsymbol{Q}^T\boldsymbol{b}$ kalkulatzen hasteko, $\boldsymbol{b}^{(2)} = \boldsymbol{H}_1\boldsymbol{b}$ ($\boldsymbol{b}^{(1)} = \boldsymbol{b}$ baita) honela aurkituko dugu:

$$\boldsymbol{H}_1\boldsymbol{b} = \boldsymbol{b} - \rho_1(\boldsymbol{u}_1^T\boldsymbol{b})\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix} - 0.2929 \left(\begin{bmatrix} 2.414 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix} \right) \begin{bmatrix} 2.414 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ -0.7071 \\ -5 \end{bmatrix}.$$

Orain, $\boldsymbol{A}^{(2)}$ matrizeari lehenengo zutabea eta lehenengo lerroa ezabatuz, azpimatrize hau geratzen da:

$$\widetilde{\boldsymbol{a}}_2 = \widetilde{\boldsymbol{A}}_2 = \begin{bmatrix} -0.7071 \\ 1 \end{bmatrix}.$$

Jarraian, $\widetilde{\boldsymbol{a}}_2$ bektoreko lehenengo gaiaren azpiko gaia zero bihurtzeko, hau egingo dugu:

$$\sigma_2 = \text{zeinu}(-0.7071)\|\widetilde{\boldsymbol{a}}_2\|_2 = -1.225,$$

eta, beraz, hau da $\boldsymbol{A}^{(2)}$-ren bigarren zutaberako erabili behar dugun $\boldsymbol{u}_2$ Householderren bektorea:

$$\boldsymbol{u}_2 = \begin{bmatrix} 0 \\ -0.7071 - 1.225 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.932 \\ 1 \end{bmatrix}, \quad \rho_2 = \frac{1}{\sigma_2 \cdot [\boldsymbol{u}_2]_2} = \frac{1}{(-1.225)(-1.932)} = 0.4226;$$

kontuan izan $\boldsymbol{u}_2$-ren lehenengo gaia zero dela, $\boldsymbol{A}^{(2)}$-ren lehenengo zutabea eta lehenengo lerroa ez aldatzeko.

Beraz, (6.2) adierazpenak erabiliz, hau lortzen dugu:

$$\begin{aligned} \boldsymbol{H}_2\boldsymbol{a}_2^{(2)} &= \boldsymbol{a}_2^{(2)} - \rho_2(\boldsymbol{u}_2^T\boldsymbol{a}_2^{(2)})\boldsymbol{u}_2 \\ &= \begin{bmatrix} -0.7071 \\ -0.7071 \\ 1 \end{bmatrix} - 0.4226 \left(\begin{bmatrix} 0 & -1.932 & 1 \end{bmatrix} \begin{bmatrix} -0.7071 \\ -0.7071 \\ 1 \end{bmatrix} \right) \begin{bmatrix} 0 \\ -1.932 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -0.7071 \\ 1.225 \\ 0 \end{bmatrix}, \end{aligned}$$

espero genuen bezala (hots, praktikan kalkulu horiek ez ditugu egin behar).

Hori dela eta, hau dugu:

$$\boldsymbol{A}^{(3)} = \begin{bmatrix} -1.414 & -0.7071 \\ 0 & 1.225 \\ 0 & 0 \end{bmatrix}.$$

Ikusi den bezala, ez dugu $\boldsymbol{H}_2$ kalkulatzeko beharrik. Hala ere, kalkulatu egingo dugu:

$$\boldsymbol{H}_2 = \mathbb{1} - \rho_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.5780 & 0.8165 \\ 0 & 0.8165 & 0.5774 \end{bmatrix},$$

(ariketa gisa, egiaztatu $\boldsymbol{H}_2\boldsymbol{A}^{(2)} = \boldsymbol{A}^{(3)}$ betetzen dela).

Gainera, $\boldsymbol{Q}^T\boldsymbol{b}$-ren kalkulua bukatzeko, $\boldsymbol{b}^{(3)} = \boldsymbol{H}_2\boldsymbol{b}^{(2)}$ (hots, $\boldsymbol{b}^{(3)} = \boldsymbol{Q}^T\boldsymbol{b} = \boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{b}$) honela aurkituko dugu:

$$\begin{aligned} \boldsymbol{H}_2\boldsymbol{b}^{(2)} &= \boldsymbol{b}^{(2)} - \rho_2(\boldsymbol{u}_2^T\boldsymbol{b}^{(2)})\boldsymbol{u}_2 \\ &= \begin{bmatrix} -0.7071 \\ -0.7071 \\ -5 \end{bmatrix} - 0.4226 \left([0 \quad -1.932 \quad 1] \begin{bmatrix} -0.7071 \\ -0.7071 \\ 5 \end{bmatrix} \right) \begin{bmatrix} 0 \\ -1.932 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -0.7071 \\ -3.674 \\ -3.464 \end{bmatrix}. \end{aligned}$$

Azkenik, $\boldsymbol{R} = \boldsymbol{A}^{(3)}$ eta $\boldsymbol{Q}^T\boldsymbol{b} = \boldsymbol{b}^{(3)}$ direnez, $\boldsymbol{R}_u\boldsymbol{x} = (\boldsymbol{Q}^T\boldsymbol{b})_u$ sistema ebatziko dugu; alegia:

$$\boldsymbol{R}_u\boldsymbol{x} = \begin{bmatrix} -1.414 & -0.7071 \\ 0 & 1.225 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ -3.674 \end{bmatrix} = (\boldsymbol{Q}^T\boldsymbol{b})_u.$$

Sistema horren soluzio bakarra $x_1 = 2$ eta $x_2 = -3$ da (biribilduz).

Bestalde, $|(\boldsymbol{Q}^T\boldsymbol{b})_l| = 3.464$ da hondar-bektorearen norma euklidearra (kasu honetan, eskalar bat denez, balio absolutua da). Konparatu emaitza horiek 6.3. adibidean lortutako emaitzekin. □

6.5. QR faktorizazioaren propietateak

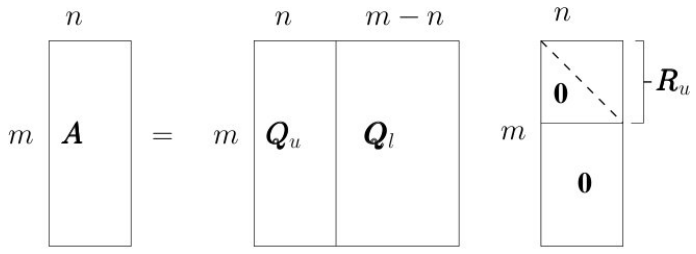
Aurreko algoritmoak frogatzen du QR faktorizazioa existitzen dela. Izan bedi $K(\boldsymbol{A})$ $\boldsymbol{A}$ matrizeko zutabeen konbinazio linealen multzoa; hots, zutabe horiek sortzen duten azpiespazio bektoriala. Jarraian ikusiko dugu zer erlazio dagoen $\boldsymbol{Q}$-ren zutabeen eta $K(\boldsymbol{A})$ eta $K(\boldsymbol{A})^\perp$ azpiespazioen artean.

6.3. teorema. *$\boldsymbol{A} = \boldsymbol{QR}$ deskonposizioa $\boldsymbol{A} \in \mathbb{R}^{m\times n}$-ren zutabe hein beteko QR faktorizazioa bada, eta $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$, $\boldsymbol{Q} = [\boldsymbol{Q}_u, \boldsymbol{Q}_l]$ eta $\boldsymbol{Q}_u = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$, $\boldsymbol{Q}_l = [\boldsymbol{q}_{n+1}, \ldots, \boldsymbol{q}_m]$ zutabekako partiketak badira, orduan, hau betetzen da:*

$$\begin{aligned} K(\boldsymbol{A}) &= K(\boldsymbol{Q}_u), \\ K(\boldsymbol{A})^{\perp} &= K(\boldsymbol{Q}_l) \end{aligned}$$

eta $\boldsymbol{A} = \boldsymbol{Q}_u\boldsymbol{R}_u$, non $\boldsymbol{R}_u \in \mathbb{R}^{n\times n}$ $\boldsymbol{R}$ matrizearen lehenengo n lerroetako azpimatrize goi-triangeluarra baita.

Frogantza. 6.20 adierazpenean $\boldsymbol{Q} = [\boldsymbol{Q}_u, \boldsymbol{Q}_l]$ partiketa sartuta, hau dugu:



. . (6.21)

Beraz, hau betetzen da:

$$\boldsymbol{A} = \boldsymbol{Q}_u\boldsymbol{R}_u + \boldsymbol{Q}_l\boldsymbol{0} = \boldsymbol{Q}_u\boldsymbol{R}_u.$$

Alegia, $k = 1, \ldots, n$ guztietarako, hau betetzen da:

$$\boldsymbol{a}_k = \sum_{i=1}^{k} r_{ik}\boldsymbol{q}_i \in K(\boldsymbol{Q}_u),$$

$(r_{1k}, \ldots, r_{kk}, 0 \ldots, 0)^T$ bektorea $\boldsymbol{R}_u$-ren k-garren zutabea izanik. Ondorioz, $K(\boldsymbol{A}) \subset K(\boldsymbol{Q}_u)$. Azkenik, $\boldsymbol{A}$-ren eta $\boldsymbol{Q}_u$-ren heinak berdinak direnez, $K(\boldsymbol{A}) = K(\boldsymbol{Q}_u)$.

Bestalde,

$$\boldsymbol{A} = \boldsymbol{QR} \quad \Rightarrow \quad \boldsymbol{Q}^T\boldsymbol{A} = \boldsymbol{R} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{Q}_u^T \\ \boldsymbol{Q}_l^T \end{bmatrix} \boldsymbol{A} = \begin{bmatrix} \boldsymbol{R}_u \\ \boldsymbol{0} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{Q}_l^T\boldsymbol{A} = \boldsymbol{0}.$$

Beraz, $K(\boldsymbol{Q}_l) \subset K(\boldsymbol{A})^{\perp}$. Gainera, $K(\boldsymbol{A})^{\perp}$-ren dimentsioa eta $\boldsymbol{Q}_l$-ren heina berdinak dira $(m-n)$. Ondorioz, $K(\boldsymbol{Q}_l) = K(\boldsymbol{A})^{\perp}$. □

Aurreko 6.4. adibidean, $\boldsymbol{H}_1$ eta $\boldsymbol{H}_2$ kalkulatu ditugu. Orduan, hau dugu (lau digitu esanguratsutara biribilduz):

$$\begin{aligned} \boldsymbol{Q} = \boldsymbol{H}_1\boldsymbol{H}_2 &= \begin{bmatrix} -0.7071 & -0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.5780 & 0.8165 \\ 0 & 0.8165 & 0.5774 \end{bmatrix} \\ &= \begin{bmatrix} -0.7071 & 0.4087 & -0.5773 \\ -0.7071 & -0.4087 & 0.5773 \\ 0 & 0.8165 & 0.5774 \end{bmatrix}. \end{aligned}$$

Beraz, zera dugu:

$$\boldsymbol{Q}_u = \begin{bmatrix} -0.7071 & 0.4087 \\ -0.7071 & -0.4087 \\ 0 & 0.8165 \end{bmatrix} \quad \text{eta} \quad \boldsymbol{Q}_l = \begin{bmatrix} -0.5773 \\ 0.5773 \\ 0.5774 \end{bmatrix}.$$

Egiaztatu adibide horretarako $\boldsymbol{Q}_u\boldsymbol{R}_u = \boldsymbol{A}$ eta $\boldsymbol{Q}_l^T\boldsymbol{A} = \boldsymbol{0}$ betetzen direla, 6.3. teoremak dioen bezala.

6.4. teorema. *Demagun $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrizea zutabe hein betekoa dela. QR faktorizazio «garbia» $\boldsymbol{A} = \boldsymbol{Q}_u\boldsymbol{R}_u$ bakarra da, non $\boldsymbol{Q}_u \in \mathbb{R}^{m\times n}$ matrizeak zutabe ortogonalak baititu eta $\boldsymbol{R}_u$ goi-triangeluarra eta diagonaleko gai positiboduna baita. Gainera, $\boldsymbol{R}_u$ matrizea $\boldsymbol{A}^T\boldsymbol{A}$ matrizearen Choleskyren faktorea da.*

Frogantza. $\boldsymbol{A}^T\boldsymbol{A} = (\boldsymbol{Q}_u\boldsymbol{R}_u)^T(\boldsymbol{Q}_u\boldsymbol{R}_u) = \boldsymbol{R}_u^T\boldsymbol{Q}_u^T\boldsymbol{Q}_u\boldsymbol{R}_u = \boldsymbol{R}_u^T\boldsymbol{R}_u$ denez gero, $\boldsymbol{R}_u$ matrizea $\boldsymbol{A}^T\boldsymbol{A}$ matrizearen Choleskyren faktorea da. Faktore hori bakarra da 5.9. teoremagatik eta, $\boldsymbol{Q}_u = \boldsymbol{A}\boldsymbol{R}_u^{-1}$ denez, $\boldsymbol{Q}_u$ ere bakarra da. Horrela da, $\boldsymbol{Q}^T\boldsymbol{A} = \boldsymbol{R}$ eraikitzean Householderren matrizeak aukeratzen direnean $\boldsymbol{R}$-ren diagonaleko gaiak positiboak izan daitezen. □

6.6. Hein urriko QR faktorizazioa

Baldin $\boldsymbol{A}$-ren heina urria bada, QR faktorizazioak ez du ematen oinarri bat $K(\boldsymbol{A})$ azpiespaziorako. Problema hori zuzendu dezakegu $\boldsymbol{A}$-ren bertsio permutatu baten QR faktorizazioa kalkulatuz; hots, $\boldsymbol{AP} = \boldsymbol{QR}$, non $\boldsymbol{P}$ permutazio-matrize bat baita.

Orokorki, $\boldsymbol{A}$-ren heina ezagutzen ez dugunez, zutabeen trukeak egin beharko ditugu. Matrize horren heina urria bada, trukerik gabe bat-batean bukatuko litzateke Householderren ezabapena. Baina, kasu horretan, bi gauza gerta daitezke: bata da beste zutabeak zero izatea, orduan, bukatu da; bestea da beste zutabeen artean baten bat zero ez izatea eta, kasu horregatik, orokorrean egiten dugu zutabeen permutazioa. Adibidez, $\boldsymbol{A}$ matrize honek hiru zutabe ditu, eta hau da bere QR faktorizazioa:

$$\boldsymbol{A} = [\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \boldsymbol{a}_3] = [\boldsymbol{q}_1 \quad \boldsymbol{q}_2 \quad \boldsymbol{q}_3] \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

orduan, $\text{hein}(\boldsymbol{A}) = 2$ da, baina $K(\boldsymbol{A})$ ez da $K([\boldsymbol{q}_1 \quad \boldsymbol{q}_2])$, ezta $K([\boldsymbol{q}_1 \quad \boldsymbol{q}_3])$ edo $K([\boldsymbol{q}_2 \quad \boldsymbol{q}_3])$ ere.

Zorionez, Householderren QR faktorizazioa erraz alda dezakegu $K(\boldsymbol{A})$-rako oinarri ortonormal bat lortzeko. Pibotatzearen estrategia izaten da norma handiena duen zutabea pibot gisa hartzekoa.

Demagun hau dugula k-garren urratsean:

$$(\boldsymbol{H}_{k-1}\dots\boldsymbol{H}_1)\boldsymbol{A}(\boldsymbol{P}_1\dots\boldsymbol{P}_{k-1})=\boldsymbol{R}^{(k-1)}=\begin{bmatrix}\boldsymbol{R}_{11}^{(k-1)} & \boldsymbol{R}_{12}^{(k-1)}\\ \boldsymbol{0} & \boldsymbol{R}_{22}^{(k-1)}\end{bmatrix},$$

non $\boldsymbol{R}_{11}^{(k-1)}$ ez-singularra eta goi-triangeluarra baita. Demagun $\boldsymbol{R}_{22}^{(k-1)}$-ren zutabekako partiketa hau:

$$\boldsymbol{R}_{22}^{(k-1)}=\left[\boldsymbol{z}_k^{(k-1)},\dots,\boldsymbol{z}_n^{(k-1)}\right]$$

eta p dela (non $k\le p\le n$) hau betetzen duen azpiindize txikiena:

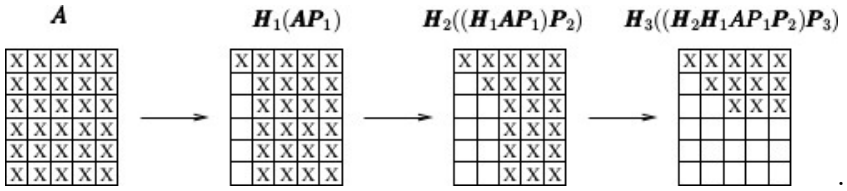
$$\|\boldsymbol{z}_p^{(k-1)}\|_2=\max\left\{\|\boldsymbol{z}_k^{(k-1)}\|_2,\dots,\|\boldsymbol{z}_n^{(k-1)}\|_2\right\}.$$

Kontuan izan, hein($\boldsymbol{A}$) $=k-1$ bada, norma handiena zero izango dela eta faktorizazioa bukatu dugula. Bestela, izan bedi $\boldsymbol{P}_k$ permutazioa, p eta k zutabeak trukatzen dituena, eta aurkitzen dugula $\boldsymbol{H}_k$ Householder transformazio egokia, $\boldsymbol{R}^{(k)}=\boldsymbol{H}_k\boldsymbol{R}^{(k-1)}\boldsymbol{P}_k$, non $\boldsymbol{R}^{(k)}=\boldsymbol{R}^{(k)}(k+1:m,k)=\boldsymbol{0}$ (hots, (k,k) gaiaren azpiko gaiak zero dira).

$\boldsymbol{A}\in\mathbb{R}^{m\times n}$-ren heina r bada, r zutabe-permutazio beharko ditugu. Beraz, r urrats horiek egin ondoren, hau izango dugu:

$$(\boldsymbol{H}_r\dots\boldsymbol{H}_1)\boldsymbol{A}(\boldsymbol{P}_1\dots\boldsymbol{P}_r)=\widetilde{\boldsymbol{R}}=\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},$$

non $\boldsymbol{R}_{11}$ ez-singularra eta goi-triangeluarra baita, $\boldsymbol{R}_{11}$-ren zutabe kopuruak $\boldsymbol{A}$-ren heina ematen digu. Adibide honetan ikus daitezke $n=5$, $m=6$ eta $r=3$ kasuari dagozkion urratsak:



Orain, $\boldsymbol{Q}^T=\boldsymbol{H}_r\dots\boldsymbol{H}_1$ eta $\boldsymbol{P}=\boldsymbol{P}_1\dots\boldsymbol{P}_r$ definitzen baditugu, honela geratuko da aurreko adierazpena:

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P}=\widetilde{\boldsymbol{R}}=\begin{array}{c}\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\\ \begin{matrix}r & n-r\end{matrix}\end{array}\begin{matrix}r\\ m-r\end{matrix}\tag{6.22}$$

edo, baliokideki, $\boldsymbol{A}\boldsymbol{P}=\boldsymbol{Q}\widetilde{\boldsymbol{R}}$. Ondorioz, minimo karratuen problema ebazteko, hau dugu ($\boldsymbol{P}$ eta $\boldsymbol{Q}^T$ ortogonalak direla kontuan hartuz):

$$\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2=\|\boldsymbol{Q}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})\|_2^2=\|(\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P})(\boldsymbol{P}^T\boldsymbol{x})-\boldsymbol{Q}^T\boldsymbol{b}\|_2^2.\tag{6.23}$$

Izan bitez

$$\boldsymbol{P}^T\boldsymbol{x}=\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{z}\end{bmatrix}\begin{matrix}r\\ n-r\end{matrix}\quad\text{eta}\quad\boldsymbol{Q}^T\boldsymbol{b}=\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{d}\end{bmatrix}\begin{matrix}r\\ m-r\end{matrix},$$

orduan, (6.22) eta (6.23) ekuazioen bidez hau lortzen da:

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 &= \left\| \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} \boldsymbol{R}_{11}\boldsymbol{y}+\boldsymbol{R}_{12}\boldsymbol{z}-\boldsymbol{c} \\ -\boldsymbol{d} \end{bmatrix} \right\|_2^2 \\ &= \|\boldsymbol{R}_{11}\boldsymbol{y}-(\boldsymbol{c}-\boldsymbol{R}_{12}\boldsymbol{z})\|_2^2+\|\boldsymbol{d}\|_2^2. \end{aligned} \tag{6.24}$$

Hori dela eta, minimo karratuen soluzioak $\boldsymbol{R}_{11}\boldsymbol{y}=(\boldsymbol{c}-\boldsymbol{R}_{12}\boldsymbol{z})$ bete behar du. Hots, edozein $\boldsymbol{z}$-rako $\boldsymbol{y}=\boldsymbol{R}_{11}^{-1}(\boldsymbol{c}-\boldsymbol{R}_{12}\boldsymbol{z})$ soluzio minimizatzaile bat izango dugu, eta, ondorioz:

$$\boldsymbol{x}=\boldsymbol{P}\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}(\boldsymbol{c}-\boldsymbol{R}_{12}\boldsymbol{z}) \\ \boldsymbol{z} \end{bmatrix}.$$

Baldin $\boldsymbol{z}=\boldsymbol{0}$ hartzen badugu, $\boldsymbol{x}_B$ *oinarri-soluzio* hau lortuko dugu:

$$\boldsymbol{x}_B=\boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix}.$$

Izen hori $\boldsymbol{z}=\boldsymbol{0}$ hartzetik eta, praktikan, $\boldsymbol{A}$-ren hein beteko azpimatrize baterako soluzioa izatetik dator.

6.3. algoritmoa. Hein urriko minimo karratu linealen problemaren ebazpena, QR faktorizazioaren bitartez.

0 urratsa. SARRERA. Sartu: $\boldsymbol{A}$ eta $\boldsymbol{b}$.

1 urratsa. Kalkula ezazu $\boldsymbol{A}$-ren QR faktorizazioa Householderren transformazioak erabiliz, $\boldsymbol{A}\boldsymbol{P}=\boldsymbol{Q}\widetilde{\boldsymbol{R}}$, non $\boldsymbol{Q}$ ortogonala baita eta $\widetilde{\boldsymbol{R}}=\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$, non $\boldsymbol{R}_{11}$ goi-triangeluarra baita eta $r=\text{hein}(\widetilde{\boldsymbol{R}})$.

2 urratsa. Kalkulatu $\begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix}=\boldsymbol{Q}^T\boldsymbol{b}$, non $\boldsymbol{c}$ bektorea $\boldsymbol{Q}^T\boldsymbol{b}$-ren lehenengo r osagaiek osatzen baitute.

3 urratsa. Ebatzi $\boldsymbol{R}_{11}\boldsymbol{y}=(\boldsymbol{c}-\boldsymbol{R}_{12}\boldsymbol{z})$. (Baldin $\boldsymbol{z}=\boldsymbol{0}$ hartzen badugu, $\boldsymbol{R}_{11}\boldsymbol{y}_B=\boldsymbol{c}$ ebazten dugu.)

4 urratsa. Aurkitu $\boldsymbol{x}=\boldsymbol{P}\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}$. ($\boldsymbol{z}=\boldsymbol{0}$ eta $\boldsymbol{y}_B$ baditugu, $\boldsymbol{x}_B=\boldsymbol{P}\begin{bmatrix} \boldsymbol{y}_B \\ \boldsymbol{0} \end{bmatrix}$).

5 urratsa. IRTEERA. Emaitza: $\boldsymbol{x}$. (Baldin $\boldsymbol{z}=\boldsymbol{0}$ hartu badugu, $\boldsymbol{x}_B$ *oinarri-soluzioa* izango dugu).

Normalean, norma euklidear minimoa (luzera minimoa) duen $\boldsymbol{x}_{LM}$ soluzioarekin geratzen gara. Alegia,

$$\|\boldsymbol{x}_{LM}\|_2=\min_{\boldsymbol{z}\in\mathbb{R}^{n-r}}\left\|\boldsymbol{x}_B-\boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12} \\ -\mathbb{1}_{n-r} \end{bmatrix}\boldsymbol{z}\right\|_2. \tag{6.25}$$

Froga daiteke (ikus [11]) hau betetzen dela:

$$1 \leq \frac{\|\boldsymbol{x}_B\|_2}{\|\boldsymbol{x}_{LM}\|_2} \leq \sqrt{1 + \|\boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12}\|_2^2}\ .$$

6.5. adibidea. *Izan bitez*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 2 \\ 7 & 6 & 10 \\ 4 & 4 & 6 \\ 1 & 0 & 1 \end{bmatrix} \quad \textit{eta} \quad \boldsymbol{b} = \begin{bmatrix} 6 \\ 6 \\ 8 \\ 3 \end{bmatrix}.$$

Aurkitu $\min \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$ *problemaren oinarri-soluzioa. Aurkitu* $\boldsymbol{x}_{LM}$ *soluzioa ere (hots, luzera minimoa duena).*

Ebazpena. Hirugarren zutabearen norma euklidearra handiena denez, 1. eta 3. zutabeak trukatuko ditugu permutazio-matrize honen bitartez:

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix};$$

hots,

$$\boldsymbol{AP} = \begin{bmatrix} 2 & 2 & 1 \\ 10 & 6 & 7 \\ 6 & 4 & 4 \\ 1 & 0 & 1 \end{bmatrix}.$$

Householderren transformazioak erabiliz, $\boldsymbol{AP} = \boldsymbol{Q}\widetilde{\boldsymbol{R}}$ deskonposizio hau kalkulatzen da:

$$\boldsymbol{Q} = \begin{bmatrix} -0.1684 & 0.7241 & -0.4453 & 0.4991 \\ -0.8422 & -0.2322 & -0.3881 & -0.2936 \\ -0.5053 & 0.2459 & 0.8049 & 0.1908 \\ -0.0842 & -0.6011 & -0.0571 & 0.7927 \end{bmatrix}, \quad \widetilde{\boldsymbol{R}} = \begin{bmatrix} -11.87 & -7.411 & -8.169 \\ 0 & 1.038 & -0.5191 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

alegia, $\boldsymbol{A}$-ren heina $r = 2$ da, eta

$$\boldsymbol{R}_{11} = \begin{bmatrix} -11.87 & -7.411 \\ 0 & 1.038 \end{bmatrix}, \quad \boldsymbol{R}_{12} = \begin{bmatrix} -8.169 \\ -0.5191 \end{bmatrix}, \quad \boldsymbol{Q}^T\boldsymbol{b} = \begin{bmatrix} -10.36 \\ 3.115 \\ 1.267 \\ 5.138 \end{bmatrix},$$

$\boldsymbol{Q}^T\boldsymbol{b}$ bektorearen lehenengo $r = 2$ osagaiek $\boldsymbol{c} = [-10.36 \ \ 3.115]^T$ bektorea osatzen dute, eta azken $m - r = 4 - 2 = 2$ osagaiek $\boldsymbol{d} = [1.267 \ \ 5.138]^T$ bektorea. Orain $z = 0$ hartuz, $\boldsymbol{R}_{11}\boldsymbol{y}_B = \boldsymbol{c}$ dugu, eta sistema hori ebatziz hau lortzen dugu:

$$\boldsymbol{y}_B = \begin{bmatrix} -1 \\ 3 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{x}_B = \boldsymbol{P}\begin{bmatrix} \boldsymbol{y}_B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -1 \end{bmatrix}.$$

Beraz, $\min \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2 = \|\boldsymbol{A}\boldsymbol{x}_B-\boldsymbol{b}\|_2 = \|\boldsymbol{d}\|_2 = \sqrt{1.267^2+5.138^2} = 5.291$ hondar optimoa (minimoa) da.

Orain, $\boldsymbol{x}_{LM}$ kalkulatuko dugu, (6.25) erabiliz. Lehenik, $\boldsymbol{w} = \boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12}$ aurkituko dugu; hots, $\boldsymbol{R}_{11}\boldsymbol{w} = \boldsymbol{R}_{12}$ ebatzi behar dugu:

$$\begin{bmatrix} -11.87 & -7.411 \\ 0 & 1.038 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -8.169 \\ -0.5191 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}.$$

Gero, bektore hau kalkulatuko dugu:

$$\boldsymbol{v} = \boldsymbol{x}_B - \boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12} \\ -\mathbb{1}_{n-r} \end{bmatrix} \boldsymbol{z} = \begin{bmatrix} 0 \\ 3 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \\ -1 \end{bmatrix} z = \begin{bmatrix} 0 \\ 3 \\ -1 \end{bmatrix} - \begin{bmatrix} -1 \\ -0.5 \\ 1 \end{bmatrix} z = \begin{bmatrix} z \\ 3+0.5z \\ -1-z \end{bmatrix}.$$

Azkenik (6.25) kontuan hartuz, hauxe dugu:

$$\|\boldsymbol{x}_{LM}\|_2 = \min_z \|\boldsymbol{v}\|_2 \;\Rightarrow\; \min\{z^2+(3+0.5z)^2+(-1-z)^2\} = \min\{2.25z^2+5z+10\} \;\Rightarrow\; z = -1.111$$

Beraz,

$$\boldsymbol{x}_{LM} = \begin{bmatrix} -1.111 \\ 3+0.5\cdot(-1.111) \\ -1-(-1.111) \end{bmatrix} = \begin{bmatrix} -1.111 \\ 2.444 \\ 0.111 \end{bmatrix},$$

non $\|\boldsymbol{x}_{LM}\|_2 = 2.687$ baita ($\|\boldsymbol{x}_B\|_2 = 3.162$ da) . □

6.7. Deskonposizio ortogonal osoa

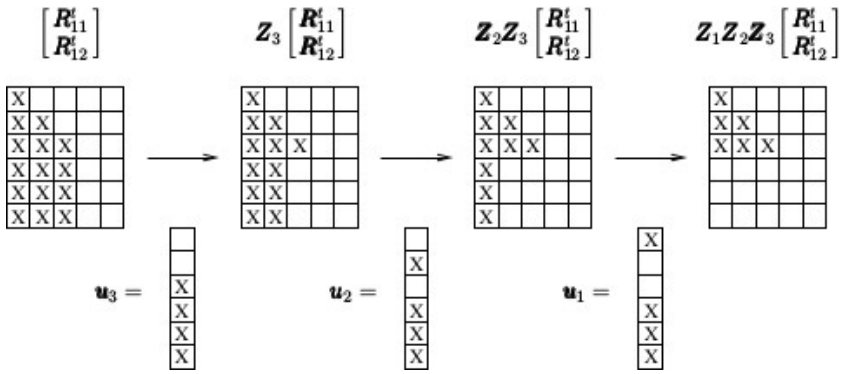
Hein urriko QR faktorizazioan $\widetilde{\boldsymbol{R}}$ matrize hau lortu dugu (ikus (6.22)):

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P} = \widetilde{\boldsymbol{R}} = \begin{matrix} \\ r \\ m-r \end{matrix} \begin{matrix} \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ \begin{matrix} r & n-r \end{matrix} \end{matrix}.$$

Gainera, $\boldsymbol{Z}_i$ Householderren matrize bereziak erabiliz, hau lor dezakegu ($\boldsymbol{T}$ definituz):

$$\boldsymbol{T}^T = \boldsymbol{Z}_1 \ldots \boldsymbol{Z}_r \begin{bmatrix} \boldsymbol{R}_{11}^T & \boldsymbol{0} \\ \boldsymbol{R}_{12}^T & \boldsymbol{0} \end{bmatrix} = \begin{matrix} \begin{bmatrix} \boldsymbol{T}_{11}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ \end{matrix} \begin{matrix} r \\ n-r \end{matrix},$$

$\boldsymbol{T}_{11}^T$ behe-triangeluarra erdietsiz. Hori lortzeko, honela jokatzen da (ikus [12]-ko 193-194 orrialdeak):



Hots, $\boldsymbol{T}^T = \boldsymbol{Z}_1 \dots \boldsymbol{Z}_r (\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{P})^T = \boldsymbol{Z}_1 \dots \boldsymbol{Z}_r \boldsymbol{P}^T \boldsymbol{A}^T \boldsymbol{Q}$. Ondorioz:

$$\boldsymbol{T} = \boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{P} \boldsymbol{Z}_r \dots \boldsymbol{Z}_1 = \boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Z} == \begin{matrix} \\ r \\ m-r \end{matrix} \begin{matrix} \begin{bmatrix} \boldsymbol{T}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ r \quad n-r \end{matrix}, \tag{6.26}$$

non $\boldsymbol{Z} = \boldsymbol{P} \boldsymbol{Z}_r \dots \boldsymbol{Z}_1$ ere ortogonala baita. Deskonposizio horri *deskonposizio ortogonal osoa* deritzogu.

Ondorioz, minimo karratuen probleman hau betetzen da:

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \quad & \|\boldsymbol{Q}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2^2 = \|(\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Z}) \boldsymbol{Z}^T \boldsymbol{x} - \boldsymbol{Q}^T \boldsymbol{b}\|_2^2 = \left\| \boldsymbol{T} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \right\|_2^2 = \\ & \left\| \begin{bmatrix} \boldsymbol{T}_{11} \boldsymbol{w} - \boldsymbol{c} \\ -\boldsymbol{d} \end{bmatrix} \right\|_2^2 = \|\boldsymbol{T}_{11} \boldsymbol{w} - \boldsymbol{c}\|_2^2 + \|\boldsymbol{d}\|_2^2, \end{aligned}$$

non

$$\boldsymbol{Z}^T \boldsymbol{x} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \quad \text{eta} \quad \boldsymbol{Q}^T \boldsymbol{b} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix}.$$

Bistan denez, baldin $\boldsymbol{x}$-k minimo karratuak minimizatu behar baditu, orduan $\boldsymbol{w} = \boldsymbol{T}_{11}^{-1} \boldsymbol{c}$ bete behar du. Gainera, $\boldsymbol{x}$-ren norma euklidearra minimoa izateko $\boldsymbol{y}$ zero izan behar, eta, horrela bada, hau dugu:

$$\boldsymbol{x}_{LM} = \boldsymbol{Z} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Z} \begin{bmatrix} \boldsymbol{T}_{11}^{-1} \boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix}.$$

6.8. Balio singularretako deskonposizioa

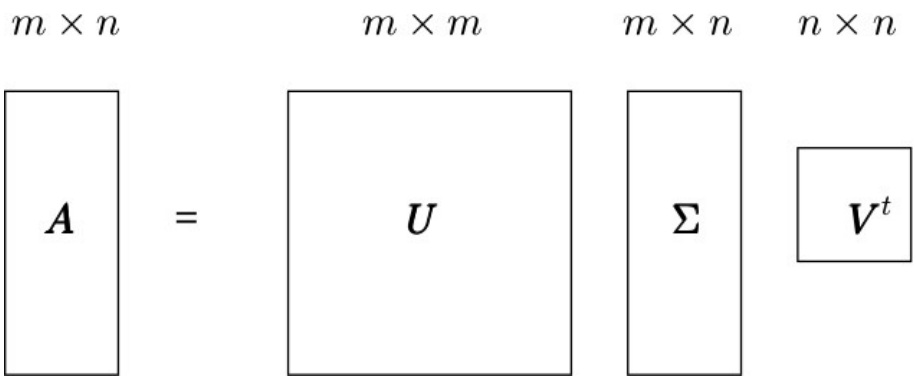
Edozein $m \times n$ $\boldsymbol{A}$ matrizea honela idatz dezakegu:

$$\boldsymbol{A} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T,$$

non $\boldsymbol{U}$ $m \times m$ matrize ortogonal bat baita, $\boldsymbol{V}$ $n \times n$ matrize ortogonal bat, eta $\boldsymbol{\Sigma}$ honelako $m \times n$ matrize diagonal bat:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \\ & & & & & & \end{bmatrix}$$

eta $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ *balio singularrak* dira. $\boldsymbol{\Sigma}$ matrizeko beste gaiak zero dira.



(6.27)

Kontuan izan m eta n-ren artean ez dagoela ordenako erlazio finko bat, eta hein($\boldsymbol{A}$) = $r \leq min\{m, n\}$ betetzen dela. Deskonposizio horretan agertzen diren matrizeen dimentsioak (6.27) adierazpenean ikus daitezke, $m > n$ den kasuan.

σ zenbakia $\boldsymbol{A}$ matrizearen balio singularra bada, $\boldsymbol{U}$ eta $\boldsymbol{V}$ matrizeetan $\boldsymbol{u}$ eta $\boldsymbol{v}$ zutabe bektore bakarrak existitzen dira, hurrenez hurren, non hau betetzen baita:

$$\boldsymbol{A}\boldsymbol{v} = \sigma\boldsymbol{u} \quad \text{eta} \quad \boldsymbol{A}^T\boldsymbol{u} = \sigma\boldsymbol{v}.$$

$\boldsymbol{u}$ eta $\boldsymbol{v}$ bektoreei *ezker* eta *eskuin bektore singularrak* deritzegu, hurrenez hurren.

Matlaben erabiltzen dugun agindua `svd` da (Singular Value Decomposition). Hots, `[U,Sigma,V]=svd(A)` aginduak balio singularretako definizioko hiru matrizeak ematen ditu. Eta `s=svd(A)` idazten badugu, min$\{m, n\}$ tamainako `s` bektore batean balio singularrak lortuko ditugu, handienetik txikienera ordenatuta.

Edozein $\boldsymbol{A}$ matrizetarako balio singularretako deskonposizioari buruzko teorema honek propietate batzuk ematen ditu, aurrerago proposatuko diren problema batzuen bidez frogatzen dira horiek.

6.5. teorema. *Izan bedi* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *eta* $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ *bere balio singularretako deskonposizioa. Izan bitez* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ *balio singular ez-nuluak eta* $\sigma_{r+1} = \ldots = \sigma_p = 0$, *non* $p = \min\{m, n\}$. *Orduan baieztapen hauek betetzen dira:*

1. $\boldsymbol{A}$*-ren heina* r *da.*
2. $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r\}$ *bektoreek* $K(\boldsymbol{A})$*-ren oinarri ortonormal bat osatzen dute.*
3. $\{\boldsymbol{u}_{r+1}, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m\}$ *bektoreek* $K(\boldsymbol{A})$ *azpiespazioaren* $\mathbb{R}^m$*-ko* $K(\boldsymbol{A})^\perp$ *azpiespazio osagarri ortogonaleko oinarri ortonormal bat osatzen dute.*
4. $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r\}$ *bektoreek* $K(\boldsymbol{A}^T)$*-ren oinarri ortonormal bat osatzen dute.*
5. $\{\boldsymbol{v}_{r+1}, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ *bektoreek* $K(\boldsymbol{A}^T)$ *azpiespazioaren* $\mathbb{R}^n$*-ko* $K(\boldsymbol{A}^T)^\perp$ *azpiespazio osagarri ortogonaleko oinarri ortonormal bat osatzen dute.*

6.8.1. Moore-Penrose-ren sasialderantzizko matrizea

$\boldsymbol{A}$ matrize ez-nulu baten alderantzizkoaren orokortze klasiko bat *Moore-Penroseren sasialderantzizkoa* da, $\boldsymbol{A}^+$. $\boldsymbol{A}$ karratua eta ez-singularra denean, $\boldsymbol{A}^+ = \boldsymbol{A}^{-1}$ dugu.

Izan bedi $\boldsymbol{A}$ $m \times n$ matrize bat eta $p = \min\{m, n\}$. Demagun $\boldsymbol{A}$ matrizearen balio singularretako deskonposizioa $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ dela. Orduan, hein$(\boldsymbol{A}) = r$ bada, $r \leq p$ izango da. Beraz, zera izango dugu:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 \quad \text{eta} \quad \sigma_{r+1} = \ldots = \sigma_p = 0.$$

$\boldsymbol{D} \in \mathbb{R}^{m\times n}$ matrize diagonal baten sasialderantzizkoa honela definitzen dugu: $\boldsymbol{D}^+$ $n \times m$ matrize diagonala da, non diagonaleko d_i^+ gaiak honela definitzen baitira:

$$d_i^+ = \begin{cases} 1/d_i, & d_i \neq 0 \text{ bada,} \\ 0, & \text{bestela.} \end{cases}$$

Orduan, $\boldsymbol{A}$ matrize orokor baterako $\boldsymbol{A}^+$ honela definitzen da:

$$\boldsymbol{A}^+ = (\boldsymbol{U\Sigma V}^T)^+ = (\boldsymbol{V}^T)^+ \boldsymbol{\Sigma}^+ \boldsymbol{U}^+.$$

Eta $\boldsymbol{U}$ eta $\boldsymbol{V}^T$ matrize ortogonalak direnez, haien sasialderantzizko matrizeak alderantzizko arruntak dira, beraz: $(\boldsymbol{V}^T)^+ = \boldsymbol{V}$ eta $\boldsymbol{U}^+ = \boldsymbol{U}^T$. Ondorioz, $\boldsymbol{A}$ matrize orokor baten sasialderantzizkoak hau betetzen du:

$$\boldsymbol{A}^+ = \boldsymbol{V\Sigma}^+ \boldsymbol{U}^T.$$

Baina, ohartu $\boldsymbol{\Sigma}^+$-ren gai ez-nuluak ez daudela ordenatuta handienetik txikienera $\boldsymbol{\Sigma}$-n gertatzen den bezala.

Propietate batzuk

1) $(\boldsymbol{A}^+)^+ = \boldsymbol{A}$.

2) $(\boldsymbol{A}^T)^+ = (\boldsymbol{A}^+)^T$.

3) $(\boldsymbol{A}\boldsymbol{A}^+)^T = \boldsymbol{A}\boldsymbol{A}^+$.

4) $(\boldsymbol{A}^+\boldsymbol{A})^T = \boldsymbol{A}^+\boldsymbol{A}$.

5) $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{v} = \boldsymbol{v}$, $\forall \boldsymbol{v} \in K(\boldsymbol{A})$.

6) $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$, $\boldsymbol{A}$-ren zutabeak linealki askeak badira ($\boldsymbol{A}$ ezkerraldetik alderantzikagarria da, $\boldsymbol{A}^+\boldsymbol{A} = \mathbb{1}$).

7) $\boldsymbol{A}^+ = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$, $\boldsymbol{A}$-ren lerroak linealki askeak badira ($\boldsymbol{A}$ eskuinaldetik alderantzikagarria da, $\boldsymbol{A}\boldsymbol{A}^+ = \mathbb{1}$).

8) $\boldsymbol{A}^+\boldsymbol{z} = \boldsymbol{0}$, $\forall \boldsymbol{z} \in K(\boldsymbol{A})^\perp$.

6.8.2. Baldintzazko zenbaki orokortua

$\boldsymbol{A}$ matrizea hein betekoa bada, *baldintzazko zenbakia* honela definitzen da:

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}^+\| \, \|\boldsymbol{A}\|.$$

Definizio hori zabaldu daiteke $\boldsymbol{A}$ matrize orokor baterako, eta, orduan, *baldintzazko zenbaki orokortua* izango dugu.

Jarraian, definizio orokortu hori aplikatuko diogu norma euklidearrerako baldintzazko zenbakiari. Dakigunez, matrize bat matrize ortogonal batez biderkatzen badugu, ez da aldatzen norma euklidearra. Beraz, $\|\boldsymbol{A}\|_2 = \|\boldsymbol{\Sigma}\|_2 = \sigma_1$ dugu, $\boldsymbol{A}$-ren balio singular handiena. Orain $\|\boldsymbol{A}^+\|_2$ kalkulatuko dugu.

$\boldsymbol{A}$-ren heina r bada, σ_r da balio singular ez-nulu txikiena. Gainera, $\boldsymbol{\Sigma}^+$ diagonaleko gaiak $\boldsymbol{A}$-ren balio singular ez-nuluen erreziprokoak direnez, $\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T$ berdintzak erakusten du $\boldsymbol{A}^+$-ren balio singular handiena $1/\sigma_r$ dela. Beraz, $\|\boldsymbol{A}^+\|_2 = 1/\sigma_r$ eta $\boldsymbol{A}^+$-ren bi-normarekiko baldintzazko zenbaki orokortua hau da:

$$\kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}^+\|_2 \, \|\boldsymbol{A}\|_2 = \frac{\sigma_1}{\sigma_r}.$$

Beste kasuetan bezala, baldintzak erakusten du sistema linealen sentsibilitatea datuen aldaketarekiko. Baina, $\boldsymbol{A}$-ren zutabeen menpekotasun linealari buruzko informazioa ere ematen du. Adibidez, matrize hauetarako:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0.0001 \end{bmatrix}$$

$\kappa_2(\boldsymbol{A}) = 1.732$ eta $\kappa_2(\boldsymbol{B}) = 2.8 \cdot 10^4$. Hots, baldintzak argi uzten du $\boldsymbol{A}$-ren zutabeek ez dutela inolako menpekotasunik eta, aldiz, $\boldsymbol{B}$-renek bai.

Hala ere, matrize singularren eta ia singularren arteko diferentzia handia da baldintzazko zenbaki orokortuarekiko. Izan bitez bi matrize hauek:

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 \\ 0 & \varepsilon \end{bmatrix}, \quad \boldsymbol{N} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},$$

$\kappa_2(\boldsymbol{M}) = \|\boldsymbol{M}^{-1}\|_2 \, \|\boldsymbol{M}\|_2 = \frac{1}{\varepsilon} \cdot 1 = \frac{1}{\varepsilon}$ dugu eta, aldiz, $\kappa_2(\boldsymbol{N}) = \|\boldsymbol{N}^{+}\|_2 \, \|\boldsymbol{N}\|_2 = 1$ da. Adibide horrek adierazten du matrize baten heina kalkulatzeko zailtasuna, zero eta ia zero gaiak bereizteko orduan, zenbakizko ikuspuntu batetik.

6.8.3. Minimo karratuen problemaren ebazpen egonkorrena

Izan bedi $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m \geq n$ eta demagun $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ balio singularretako deskonposizioa dugula; orduan, zera betetzen da:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{U}^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} - \boldsymbol{b})\|_2 = \|\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} - \boldsymbol{U}^T\boldsymbol{b}\|_2,$$

non $\boldsymbol{y} = \boldsymbol{V}^T\boldsymbol{x} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix}$ eta $\boldsymbol{z} = \boldsymbol{U}^T\boldsymbol{b} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix}$ definituz, hau lortzen baitugu:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{\Sigma}\boldsymbol{y} - \boldsymbol{z}\|_2 = \left\| \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_r \\ v_1 \\ \cdots \\ v_{n-r} \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ \cdots \\ c_r \\ d_1 \\ \cdots \\ d_{m-r} \end{bmatrix} \right\|_2 .$$

Ondorioz, sistema erraz hau dugu:

$$\sigma_i w_i = c_i, \quad i = 1, \ldots, r.$$

Beraz, $w_i = c_i/\sigma_i$ dugu, $i = 1, \ldots, r$, eta $\|\boldsymbol{y}\|_2$ ahalik eta txikien izateko $\boldsymbol{v} = \boldsymbol{0}$ hartuko dugu. Azkenik, $\boldsymbol{y} = \boldsymbol{V}^T\boldsymbol{x}$ denez, $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{y}$ dugu, eta minimo karratuen problemaren luzera minimoko soluzioa hau da:

$$\boldsymbol{x}_{LM} = \boldsymbol{V} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{0} \end{bmatrix}$$

eta hondarra $\|\boldsymbol{d}\|_2$ da.

6.9. Problemak

Eskuz ebazteko problemak:

1. (i) Izan bitez sistema hauek:

$a)$
$$\begin{aligned} x_1 + 3x_2 + 2x_3 &= 5 \\ x_1 + 5x_2 + 3x_3 &= 10 \\ 2x_1 + 4x_2 - 6x_3 &= -4. \end{aligned}$$

$b)$
$$\begin{aligned} 2x_1 + x_2 + x_3 &= 3 \\ 4x_1 + 3x_2 + 3x_3 + x_4 &= 7 \\ 8x_1 + 7x_2 + 9x_3 + 5x_4 &= 17 \\ 6x_1 + 7x_2 + 9x_3 + 8x_4 &= 15. \end{aligned}$$

$c)$
$$\begin{aligned} -5x_1 + 2x_2 - x_3 &= 2 \\ x_1 + 3x_3 &= 2 \\ 3x_1 + x_2 + 6x_3 &= 2. \end{aligned}$$

$d)$
$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 3x_1 + 3x_2 + 9x_3 &= 0 \\ 3x_1 + 3x_2 + 5x_3 &= 4. \end{aligned}$$

$e)$
$$\begin{aligned} x_1 + x_2 + 4x_4 &= 5 \\ 2x_1 - 1x_2 + 5x_3 &= -6 \\ 5x_1 + 2x_2 + x_3 + 2x_4 &= 3 \\ -3x_1 + 2x_3 + 6x_4 &= 4. \end{aligned}$$

$f)$
$$\begin{aligned} 4x_1 + 8x_2 + 4x_3 &= 8 \\ x_1 + 5x_2 + 4x_3 - 3x_4 &= -4 \\ x_1 + 4x_2 + 7x_3 + 2x_4 &= 10 \\ x_1 + 3x_2 - 2x_4 &= -4. \end{aligned}$$

Aurkitu sistemei elkartutako matrizeen QR faktorizazioa, erabilitako Householderren bektoreak emanez eta Householderren matrizeak kalkulatu barik. Erakutsi, urratsez urrats, egindako eragiketa nagusiak.

(ii) Ebatzi sistemak, aurreko atalean aurkitutako QR faktorizazioak erabiliz.

2. (a) Izan bedi $\boldsymbol{x} = [1, 12, 2, 5, 7]^T$ bektorea. Kalkula ezazu biraketa bat $(4, 2)$ planoan x_2 zero bihurtzeko. Zenbat radianetako biraketa eman behar du $\boldsymbol{x}$ bektoreak plano horretan emaitza hori lortzeko?

(b) Bektorea errenkada bat bada, esate baterako $\boldsymbol{x} = [1, 12, 2, 5, 7]$, nola egingo zenuke $(4, 2)$ planoan x_2 zero bihurtzeko?

3. Aplikatu Givens-en biraketak matrize honi, $(1, 2)$ planoan lehenengo zutabeko bigarren lerroko gaia zero bihurtzeko:

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 3 & 2 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

(a) Horretarako, c eta s balioak honela kalkulatuko ditugu (ikus 6.3. atala):

$$c = \cos(\theta) = \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \quad \text{eta} \quad s = \sin(\theta) = \frac{-x_k}{\sqrt{x_i^2 + x_k^2}},$$

non (i, k) bikoteak esaten baitigu zein planotan egiten den θ angeluko biraketa, bektore baten i. lerroko gaiaz k. lerroko gaia zero bihurtzeko.

(b) Gero, kalkula ezazu dagokion $\boldsymbol{G}(i,k,\theta)$ matrizea.

(c) Kalkula ezazu $\boldsymbol{G}(i,k,\theta)^T\boldsymbol{A}$.

(d) Diseinatu algoritmo bat $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ matrize orokor batekin $\boldsymbol{G}(i,k,\theta)^T\boldsymbol{M}$ kalkulua egiteko. c-ren eta s-ren kalkulua algoritmoaren barnean egin daiteke.

(e) Bigarren zutabeko 2. lerroko gaiaz 3. lerroko gaia zero egiteko, zein izango litzateke $\boldsymbol{G}(i,k,\theta)$ Givensen matrizea?

4. Izan bitez $\boldsymbol{Ax} = \boldsymbol{b}$ sistema gaindeterminatu hauek:

$a)$ $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 2 & -3 \\ 0 & 0 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$. $b)$ $\boldsymbol{A} = \begin{bmatrix} -1 & 1 \\ 2 & 1 \\ 1 & -2 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} 10 \\ 5 \\ 20 \end{bmatrix}$.

$c)$ $\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} 4 \\ 0 \\ 1 \\ 2 \end{bmatrix}$. $d)$ $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ -1 & -2 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$.

$e)$ $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ -1 & 3 \\ 1 & 2 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} -2 \\ 0 \\ 8 \end{bmatrix}$. $f)$ $\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 \\ 3 & -1 & 2 \\ -1 & 5 & 4 \\ 0 & 2 & 2 \end{bmatrix}$ eta $\boldsymbol{b} = \begin{bmatrix} 1 \\ 5 \\ 3 \\ 4 \end{bmatrix}$.

Ebatzi sistema horiek ekuazio normalak erabiliz. Zein da beren hondarraren norma euklidearra?

5. Ebatzi aurreko problemaren sistemak, QR faktorizazioaren bitartez. Kalkula ezazu hondarraren norma euklidearra.

6. Izan bedi $\boldsymbol{Ax} = \boldsymbol{b}$ sistema gaindeterminatua, non

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0.001 \end{bmatrix} \text{ eta } \boldsymbol{b} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}.$$

(a) Zein da ekuazio normalei dagokien matrizearen bi-normarako baldintzazko zenbakia?

(b) Zein da $\boldsymbol{A}$-ren bi-normarako baldintzazko zenbaki orokortua? Zein dira QR faktorizaziorako $\boldsymbol{R}$ matrizea eta $\boldsymbol{R}$-ren bi-normarako baldintzazko zenbaki orokortua? Galdera hauek erantzuteko, kontuan hartu 6.8.2. atala.

(c) Konparatu aurreko bi ataletan lortutako baldintzazko zenbakiak. Zer esan dezakegu?

(d) Ebatzi sistema hori bi metodoekin, lau zifra esanguratsuz, eta konpara itzazu emaitzak.

7. Aurkitu $f(x) = ax+b$ funtzio lineal hoberena, minimo karratuen zentzuan, datu multzo hauetarako:

   a)
x	-1	0	1	2
y	3	2	0	4

   b)
x	-3	-1	1	3
y	15	5	1	5

   Zein dira hondarrak? Marraztu *a)* eta *b)* kasuetan (bakoitza grafiko desberdin batean) funtzio linealak eta taulen puntuak.

8. Aurkitu $f(x) = ax^2 + bx + c$ parabola hoberena, minimo karratuen zentzuan, datu multzo hauetarako:

   a)
x	-1	0	1	2
y	3	2	0	4

   b)
x	-3	-1	1	3
y	15	5	1	5

   Zein dira hondarrak? Marraztu *a)* eta *b)* kasuetan (bakoitza grafiko desberdin batean) funtzio koadratikoa eta taulen puntuak. Konparatu lortutako *a)* kasuko grafikoa aurreko problemari dagokion *a)* kasuko grafikoarekin. Egin berdina *b)* kasuetan. Zer ondorioztatzen da?

9. Izan bedi matrize hau:
$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2.5 \\ 1 & 0 & 2 \end{bmatrix}.$$
   (a) Aurkitu $K(\boldsymbol{A})$ azpiespazioaren oinarri ortonormal bat, QR-faktorizazioa erabiliz.
   (b) Aurkitu $K(\boldsymbol{A})$-ren azpiespazio nuluaren oinarri ortonormal bat, QR-faktorizazioa erabiliz.
   (c) Aurkitu $\boldsymbol{Ax} = \boldsymbol{b}$ sistemaren $\boldsymbol{x}_B$ oinarri-soluzioa, baldin $\boldsymbol{b} = [1 \;\; 3 \;\; 5]^T$ bada.
   (d) Aurkitu $\boldsymbol{Ax} = \boldsymbol{b}$ sistemaren $\boldsymbol{x}_{LM}$ soluzioa; hots, norma euklidear minimoa duen minimo karratuen problemaren soluzioa.

10. Izan bedi matrize hau:
$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 5 & 6 \\ 1 & 8 & 9 \\ 1 & 11 & 12 \end{bmatrix}.$$
    (a) Hein urrikoa da? Arrazoitu erantzuna.
    (b) Kalkula ezazu bere QR-faktorizazioa, zutabeen permutazioak eginez, beharrezkoa bada.
    (c) Aurkitu $K(\boldsymbol{A})$ azpiespazioaren oinarri ortonormal bat, QR-faktorizazioa erabiliz.
    (d) Aurkitu $K(\boldsymbol{A})$-ren azpiespazio nuluaren oinarri ortonormal bat, QR-faktorizazioa erabiliz.
    (e) Aurkitu $\boldsymbol{Ax} = \boldsymbol{b}$ sistemaren $\boldsymbol{x}_B$ oinarri-soluzioa, baldin $\boldsymbol{b} = [2 \;\; 1 \;\; 4 \;\; 3]^T$ bada.
    (f) Aurkitu $\boldsymbol{Ax} = \boldsymbol{b}$ sistemaren $\boldsymbol{x}_{LM}$ soluzioa; hots, norma euklidear minimoa duen minimo karratuen problemaren soluzioa.

11. Izan bedi $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrizea eta $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ bere balio singularretako deskonposizioa, non $\boldsymbol{A}$-ren heina r baita eta $m \geq n \geq r$ ($n \geq m \geq r$ ere izan liteke). Izan bedi $\boldsymbol{A}^+ = \boldsymbol{V\Sigma}^+\boldsymbol{U}^T \in \mathbb{R}^{n\times m}$ non

$$\boldsymbol{\Sigma}^+ = \operatorname{diag}\left\{\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\}.$$

$\boldsymbol{A}^+$ matrizeari $\boldsymbol{A}$-ren *Moore-Penroseren sasialderantzizko matrize* deritzo.

Froga ezazu $\boldsymbol{X} = \boldsymbol{A}^+$-ek Moore-Penroseren lau baldintza hauek betetzen dituela:

(i) $\boldsymbol{AXA} = \boldsymbol{A}$
(ii) $\boldsymbol{XAX} = \boldsymbol{X}$
(iii) $(\boldsymbol{AX})^T = \boldsymbol{AX}$
(iv) $(\boldsymbol{XA})^T = \boldsymbol{XA}$

Ikus 6.8.1. atala.

12. Izan bedi $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ matrizea eta $\boldsymbol{P} \in \mathbb{R}^{m\times m}$ eta $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ matrize ortogonalak. Froga ezazu $\|\boldsymbol{PAQ}\|_2 = \|\boldsymbol{A}\|_2$ betetzen dela.

13. Izan bedi $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ r heineko matrize bat eta $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ balio singularretako deskonposizioa, non $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ balio singular ez-nuluak eta $\sigma_{r+1} = \ldots = \sigma_n = 0$ baitira.

(a) Froga ezazu $\|\boldsymbol{A}\|_2 = \|\boldsymbol{\Sigma}\|_2 = \sigma_1$.
(b) Froga ezazu $r = n$ bada, $\kappa_2(\boldsymbol{A}) = \sigma_1/\sigma_n$.
(c) Baldin balio singular ez-nuluak $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ badira, froga ezazu $\boldsymbol{A} = \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^T + \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^T + \ldots + \sigma_r\boldsymbol{u}_r\boldsymbol{v}_r^T$.
$\boldsymbol{A} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{V}}^T$ adierazpenari *balio singularretako deskonposizio laburtu* deritzogu, non

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}} &= \operatorname{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\}\\ \widehat{\boldsymbol{U}} &= [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r]\\ \widehat{\boldsymbol{V}} &= [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r]\end{aligned}$$

14. (a) Froga ezazu balio singularretako deskonposiziotik $\boldsymbol{AV} = \boldsymbol{U\Sigma}$ atera dezakegula.
(b) Aurreko berdintza erabiliz, froga ezazu $\boldsymbol{A}^T\boldsymbol{A}$ eta $\boldsymbol{AA}^T$ autobalio berdinak dituztela eta balio singularren karratuak direla.
(c) Froga ezazu $\boldsymbol{A}$-ren heina $\boldsymbol{A}^T\boldsymbol{A}$-ren heinaren berdina dela, eta hori dela balio singular ez-nuluen kopuruaren berdina.

15. (a) Froga ezazu espazio bektorial baten oinarri ortonormal baten bektoreak matrize ortogonal batez biderkatzean lortutako bektore berriek espazio horretako beste oinarri ortonormal bat ematen dutela.

(b) Espazio bektorial baten $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$ oinarri kanonikoa $\boldsymbol{Q}$ matrize ortogonal batez biderkatzen badugu, zein da lortzen den oinarri ortonormal berria?

16. Izan bedi $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ r heineko matrize bat eta $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ balio singularretako deskonposizioa. Izan bitez $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ ezker bektore singularrak, $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ eskuin bektore singularrak eta $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ balio singular ez-nuluak eta $\sigma_{r+1} = \ldots = \sigma_p = 0$, non $p = \min\{m, n\}$.

(a) Froga ezazu $\boldsymbol{Av}_1, \ldots, \boldsymbol{Av}_r$ $\mathbb{R}^m$-ko bektoreak ortogonalak direla.

(b) Kalkula ezazu $\boldsymbol{A}^T\boldsymbol{A}$, eta hortik ondorioztatu nola kalkula ditzakezun σ_i balio singularrak eta $\sigma_i > 0$ betetzen duten balioei dagozkien $\boldsymbol{v}_i$ eskuin bektore singularrak.

(c) $\boldsymbol{AV} = \boldsymbol{U\Sigma}$ erabiliz, azaldu nola kalkula dezakezun $\sigma_i > 0$ bakoitzari dagokion $\boldsymbol{u}_i$ ezker bektore singularra.

(d) Froga ezazu $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ $\mathbb{R}^m$-ko bektoreek $K(\boldsymbol{A})$ azpiespazioaren oinarri ortonormal bat osatzen dutela (kontuan hartu $\boldsymbol{AV} = \boldsymbol{U\Sigma}$).

(e) Froga ezazu $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ $\mathbb{R}^n$-ko bektoreek $K(\boldsymbol{A}^T)$ azpiespazioaren oinarri ortonormal bat osatzen dutela (kontuan hartu $\boldsymbol{A}^T$-ren balio singularretako deskonposiziotik $\boldsymbol{A}^T\boldsymbol{U} = \boldsymbol{V\Sigma}^T$ lortzen dela).

(f) Froga ezazu $\boldsymbol{V}$ matrizeko $\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_n$ $\mathbb{R}^n$-ko zutabe-bektoreek $K(\boldsymbol{A}^T)^\perp$ azpiespazioaren oinarri ortonormal bat osatzen dutela (kontuan hartu $\boldsymbol{AV} = \boldsymbol{U\Sigma}$).

(g) Froga ezazu $\boldsymbol{U}$ matrizeko $\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m$ $\mathbb{R}^m$-ko zutabe-bektoreek $K(\boldsymbol{A})^\perp$ azpiespazioaren oinarri ortonormal bat osatzen dutela (kontuan hartu $\boldsymbol{A}^T\boldsymbol{U} = \boldsymbol{V\Sigma}^T$).

(h) $\boldsymbol{AA}^T$ erabiliz, nola kalkula ditzakezu σ_i balio singularrak eta $\sigma_i > 0$ betetzen duten balioei dagozkien $\boldsymbol{u}_i$ ezker bektore singularrak? Gero, nola kalkulatuko zenuke $\sigma_i > 0$ bakoitzari dagokion $\boldsymbol{v}_i$ ezker bektore singularra?

17. (a) Izan bedi matrize hau:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

16. problemaren emaitzak erabiliz, aurkitu $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ balio singularretako deskonposizioa, eta egiaztatu berdintza hori betetzen dela.

(b) $\boldsymbol{b} = [3, 2, 1]^T$ bada, $\boldsymbol{A}$-ren deskonposizio hori aprobetxatuz, ebatzi $\min_{\boldsymbol{x}} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$ minimo karratuen problema $\boldsymbol{x}$ soluzioa norma minimokoa izanik.

18. Izan bedi matrize hau:

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}.$$

16. problemaren emaitzak erabiliz, aurkitu $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ balio singularretako deskonposizioa, eta egiaztatu berdintza hori betetzen dela.

MATLABez programatzeko problemak:

19. Sortu M fitxategi bat $\boldsymbol{Ax} = \boldsymbol{b}$ sistema lineal bat ($\boldsymbol{A}$ karratua eta ez-singularra denean) ebazteko gai izan dadin QR faktorizazioaren laguntzaz, baina $\boldsymbol{Q}$ matrize ortogonala eta Householderren matrizeak kalkulatu barik. Aldiz, gorde itzazu lortutako Householderren bektoreak $\boldsymbol{U}$ matrize batean.

    Programa horrek sistema hori ebazteko ondoz ondoko lan hauek egin behar ditu:

    (a) Programa horrek goiburu hau izan behar du: `function [U,R,x]=qrkes(A,b)`

    (b) Egiaztatu $\boldsymbol{A}$ karratua eta ez-singularra dela.

    (c) $\boldsymbol{R} = \boldsymbol{Q}^T\boldsymbol{A}$ eta $\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{b}$ kalkulatu behar ditu.

    (d) $\boldsymbol{Rx} = \boldsymbol{c}$ sistema ebatzi behar du. Lan hori egiteko, `atzerantz` funtzioa erabili.

20. Izan bedi sistema hau:

$$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & -1 & 3 & 5 \\ 0 & 0 & 2 & 5 \\ -2 & -6 & -3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

    Egiaztatu `qrkes.m` funtzioa ondo dabilela aurreko sistema ebatziz. Egiaztatu emaitza.

21. Sortu M fitxategi bat $\min_x \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$ minimo karratuen problema ebazteko, non $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ eta $m > n$. $\boldsymbol{A}$ hein betekoa dela suposatuko dugu. Bi metodo erabiliko ditugu.

    - *Ekuazio normalen metodoa.* Kasu honetan, $\boldsymbol{A}^T\boldsymbol{A}$ matrizea deskonposatu Choleskyren faktorizazioa erabiliz. Gero, `aurrerantz` eta `atzerantz` funtzioen bidez ebatzi sistema. Sortu funtzio berri hau:

      `function [h,x]=eknorm(A,b)`

    - *QR metodoa.* Kasu honetan, aprobetxa dezakezu lehenengo ariketan egindako `qrkes` funtzioa. Baina, orain $\boldsymbol{A}$ matrizea ez da karratua izan behar. Sortu funtzio berri hau:

      `function [h,x]=qrmkhb(A,b)`

    Bi metodoetan h hondar bektorearen bi-norma da.

22. Aurreko ariketan lortutako bi metodoak erabiliz, ebatzi $\min_x \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$ minimo karratuen problema balio hauetarako:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{eta} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ -5 \end{bmatrix}.$$

    Problema hau 6.3. eta 6.4. adibideetan agertzen da.

7. kapitulua

Ekuazio ez-linealen sistemen ebazpena

Ikasgai honetan aldagai anitzeko problemekin arituko gara, eta gure helburua izango da ekuazio ez-linealen sistemak ebaztea, algoritmo lokalen bidez. Newtonen metodoarekin hasiko gara, eta haren inplementazioa eta ezaugarriak aztertuko ditugu.

7.1. Newtonen metodoa

Izan bedi $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ funtzioa eta $\boldsymbol{f} \in C^1$ (hots, lehenengo ordenako deribatu partzial guztiak funtzio jarraituak dira), non $f_1, \ldots, f_n$ bere osagaiak baitira. Ebatzi behar dugun ekuazioa zera da:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}. \tag{7.1}$$

Aldagai bateko funtzioen kasurako bezala, hemen ere metodo iteratibo bat erabiliko dugu $\boldsymbol{x}_0 \in \mathbb{R}^n$ puntu batetik abiatuz. Demagun k-garren iterazioan dagoela prozesu hori; orduan, puntu horren inguruan, hau da Taylorren lehenengo ordenako garapena:

$$\boldsymbol{M}_k(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{J}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k),$$

non $\boldsymbol{J}(\boldsymbol{x}_k)$ matrizea $\boldsymbol{f}$-ren jacobiarra baita $\boldsymbol{x}_k$ puntuan (hots, $\boldsymbol{f}$-ren lehenengo ordenako deribatua da puntu horretan); honela definitzen da:

$$\boldsymbol{J}(\boldsymbol{x}_k) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\boldsymbol{x}_k) & \cdots & \frac{\partial f_1}{\partial x_n}(\boldsymbol{x}_k) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\boldsymbol{x}_k) & \cdots & \frac{\partial f_n}{\partial x_n}(\boldsymbol{x}_k) \end{bmatrix} = \begin{bmatrix} \nabla f_1(\boldsymbol{x}_k)^T \\ \vdots \\ \nabla f_n(\boldsymbol{x}_k)^T \end{bmatrix}$$

Jarraian, (7.1) ekuazioan $\boldsymbol{f}(x)$-ren ordez $\boldsymbol{x}_k$-ren inguruko $\boldsymbol{M}_k(\boldsymbol{x})$ hurbilpen lineala jarriko dugu; alegia:

$$\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{J}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) = \boldsymbol{0}.$$

Ekuazio horren soluzioa $\boldsymbol{x} = \boldsymbol{x}_k - \boldsymbol{J}(\boldsymbol{x}_k)^{-1}\boldsymbol{f}(\boldsymbol{x}_k)$ denez, hau da Newtonen metodoaren iterazioa:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{J}(\boldsymbol{x}_k)^{-1}\boldsymbol{f}(\boldsymbol{x}_k). \tag{7.2}$$

Newtonen $\boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ urratsak $\boldsymbol{x}^* - \boldsymbol{x}_k$-ren hurbilpen bat ematen digu.

Geometrikoki, aldagai bateko funtzioetarako, Newton-Raphsonen metodoan $f(x)$-ren hurbilpena x_k puntuko zuzen ukitzailea zen. Ekuazio ez-linealen sistemetarako, $f_i(\boldsymbol{x}) = 0$ ekuazio bakoitzeko $f_i(\boldsymbol{x})$ hurbiltzen da $\boldsymbol{x}_k$ puntuko (hiper)plano ukitzaileaz, hots:

$$f_i(\boldsymbol{x}_k) + \nabla f_i(\boldsymbol{x}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) = 0$$

eta ekuazio linealen sistema horren soluzioa, (7.2) adierazpenak definitutakoa, plano ukitzaile horien guztien ebakitze-puntua da.

7.1. adibidea. *Izan bitez funtzio hauek:*

$$\begin{aligned} f_1(x,y) &= x^2 - 2x - y + 0.5 \\ f_2(x,y) &= x^2 + 4y^2 - 4. \end{aligned}$$

Gure helburua ekuazio-sistema hau ebaztea da:

$$\begin{aligned} f_1(x,y) &= 0 \\ f_2(x,y) &= 0. \end{aligned}$$

Ebazpena. $f_1(x,y) = 0$ eta $f_2(x,y) = 0$ ekuazioek XOY planoko bi kurba definitzen dituzte; beraz, sistema horren soluzioa bi kurben $(\overline{x}, \overline{y})$ ebakitze-puntu bat da. Sistemako kurbak oso ezagunak dira:

$$\begin{array}{ll} x^2 - 2x - y + 0.5 = 0 & \text{parabola bat da,} \\ x^2 + 4y^2 - 4 = 0 & \text{elipse bat da.} \end{array}$$

Sistema horrek bi soluzio (bi ebakitze-puntu) ematen ditu, $(-0.2, 1.0)$ eta $(1.9, 0.3)$ puntuetatik hurbil daudenak.

Newtonen metodoan, $(x_0, y_0) = (0, 1)$ puntutik abiatzen bagara, zera dugu:

$$\begin{array}{lll} \nabla f_1(x,y)^T = (2x-2, -1) & \Rightarrow & \nabla f_1(0,1)^T = (-2,-1) \\ \nabla f_2(x,y)^T = (2x, 8y) & \Rightarrow & \nabla f_2(0,1)^T = (0,8). \end{array}$$

Ondorioz, hauek dira $(0,1)$ puntuko hiperplano ukitzaileei dagozkien ekuazioak:

$$\begin{array}{lll} f_1(0,1) + \nabla f_1(x_0,y_0)^T \cdot (x - x_0, y - y_0) = 0 & \Rightarrow & -0.5 + (-2,-1) \cdot (x-0, y-1) = 0 \\ f_2(0,1) + \nabla f_2(x_0,y_0)^T \cdot (x - x_0, y - y_0) = 0 & \Rightarrow & 0 + (0,8) \cdot (x-0, y-1) = 0. \end{array}$$

Horrek sistema hau inplikatzen du:

$$\begin{aligned} -2x - y + 0.5 &= 0 \\ y - 1 &= 0 \end{aligned}$$

eta ebatziz, hau lortzen dugu:

$$x = -0.25 \qquad y = 1$$

eta (7.2) iterazioaz $(x_1, y_1) = (-0.25, 1)$ dugu.

Prozesu hori jarraituz, lau iteraziotan $(x_4, y_4) = (-0.2223147, 0.9938121)$ dugu, eta $f_1(x_4, y_4) = 0.000241126$ eta $f_2(x_4, y_4) = 0.000073786$.

Hau izango da errorean tamaina:

$$\|(f_1(x_4, y_4), f_2(x_4, y_4))\|_\infty = \|(0.000241126, 0.000073786)\|_\infty = 0.000241126.$$

Jarritako errorearen tolerantzia maximoa 0.001 izan bada, soluzio hori onargarria da. □

7.1. algoritmoa. Newtonen metodoa, ekuazio ez-linealen sistemetarako.

0. urratsa. SARRERA. Sartu: $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{x}_0$ (hasierako puntua) $ze > 0$ (zehaztasun erlatiboa), $emax > 0$ (errorearen tolerantzia) eta $imax$ (iterazioen kopuru maximoa). Jarri $k = 0$

1. urratsa. Kalkulatu $\boldsymbol{f}(\boldsymbol{x}_k)$ eta $\boldsymbol{J}(\boldsymbol{x}_k)$.

2. urratsa. Ebatzi sistema lineal hau:

$$\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{p}_k = -\boldsymbol{f}(\boldsymbol{x}_k). \tag{7.3}$$

3. urratsa. Kalkulatu honako puntu hau:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$$

4. urratsa. Baldin $\|\boldsymbol{p}_k\|/\|\boldsymbol{x}_{k+1}\| < ze$ edo $\|\boldsymbol{f}(\boldsymbol{x}_{k+1})\| < emax$ edo $k \geq imax$ bada, gelditu egiten da. Bestela, $k = k + 1$ egiten da, eta 1. urratsera goaz.

5. urratsa. IRTEERA. Emaitza: $\boldsymbol{x}_{k+1}$.

Algoritmo horren 2. urratsean (7.3) Newtonen sistema ebatzi behar dugu. Urrats horretan, LU deskonposizioa edo QR faktorizazioa erabil ditzakegu.

7.2. adibidea. *Izan bedi* $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x + y - 3 \\ x^2 + y^2 - 9 \end{bmatrix}$. *Ebatzi* $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ *sistema.*

Ebazpena. Sistema horrek $(3, 0)^T$ eta $(0, 3)^T$ erroak ditu. Izan bedi $\boldsymbol{x}_0 = (1, 5)^T$. Orduan, hauek dira Newtonen lehenengo bi iterazioak:

$$\boldsymbol{J}(\boldsymbol{x}_0)\boldsymbol{p}_0 = -\boldsymbol{f}(\boldsymbol{x}_0) \quad \Rightarrow \quad \begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix} \boldsymbol{p}_0 = -\begin{bmatrix} 3 \\ 17 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{p}_0 = \begin{bmatrix} -13/8 \\ -11/8 \end{bmatrix},$$

$$\boldsymbol{J}(\boldsymbol{x}_1)\boldsymbol{p}_1 = -\boldsymbol{f}(\boldsymbol{x}_1) \quad \Rightarrow \quad \begin{array}{c} \boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{p}_0 = (-0.625, 3.625)^T, \\ \begin{bmatrix} 1 & 1 \\ -5/4 & 29/4 \end{bmatrix} \boldsymbol{p}_1 = - \begin{bmatrix} 0 \\ 145/32 \end{bmatrix} \end{array} \quad \Rightarrow \quad \boldsymbol{p}_1 = \begin{bmatrix} 145/272 \\ -145/272 \end{bmatrix}$$
$$\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{p}_1 = (-0.092, 3.092)^T.$$

Ikus dezakegunez, $(0,3)^T$ soluziotik nahiko hurbil dago $\boldsymbol{x}_2$. $\square$

7.1.1. Newtonen metodoaren konbergentzia lokala

7.1. lema. *Izan bedi $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ etengabe diferentziagarria $\mathcal{D} \subset \mathbb{R}^n$ multzo konbexu irekian. Orduan, hau betetzen da $\boldsymbol{x}, \boldsymbol{x}+\boldsymbol{p} \in \mathcal{D}$ guztietarako:*

$$\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{p}) - \boldsymbol{f}(\boldsymbol{x}) = \int_0^1 \boldsymbol{J}(\boldsymbol{x}+t\boldsymbol{p})\boldsymbol{p}\ dt \equiv \int_{\boldsymbol{x}}^{\boldsymbol{x}+\boldsymbol{p}} \boldsymbol{f}'(\boldsymbol{z})\ d\boldsymbol{z}. \tag{7.4}$$

7.1. definizioa. *Izan bedi $\boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$. Orduan, $\boldsymbol{G}$ funtzio matriziala $\boldsymbol{x}$ puntuko $\boldsymbol{\gamma}$-**Lipschitz jarraitua** dela esango dugu, $\mathcal{D} \subset \mathbb{R}^n$ multzo irekia, $\boldsymbol{x} \in \mathcal{D}$, eta γ konstante bat existitzen badira, non $\boldsymbol{v} \in \mathcal{D}$ guztietarako hau betetzen baita:*

$$\|\boldsymbol{G}(\boldsymbol{v}) - \boldsymbol{G}(\boldsymbol{x})\| \leq \gamma \|\boldsymbol{v} - \boldsymbol{x}\|. \tag{7.5}$$

Hori gertatzen denean, honela adieraziko dugu: $\boldsymbol{G} \in Lip_\gamma(\mathcal{D})$.

7.2. lema. *Izan bedi $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ etengabe diferentziagarria $\mathcal{D} \subset \mathbb{R}^n$ multzo konbexu irekian, eta izan bedi $\boldsymbol{J}$ γ-Lipschitz jarraitua $\boldsymbol{x} \in \mathcal{D}$ puntuaren $\mathcal{D}$ ingurunean, hots $\boldsymbol{J} \in Lip_\gamma(\mathcal{D})$, bektore-norma bat eta berak eragindako matrize-norma erabiliz. Orduan, hau betetzen da $\boldsymbol{x}+\boldsymbol{p} \in \mathcal{D}$ guztietarako:*

$$\|\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{p}) - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p}\| \leq \frac{\gamma}{2}\|\boldsymbol{p}\|^2. \tag{7.6}$$

Frogantza. Aurreko lemaren arabera,

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}+\boldsymbol{p}) - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p} &= \int_0^1 \boldsymbol{J}(\boldsymbol{x}+t\boldsymbol{p})\boldsymbol{p}\ dt - \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p} \\ &= \int_0^1 (\boldsymbol{J}(\boldsymbol{x}+t\boldsymbol{p}) - \boldsymbol{J}(\boldsymbol{x}))\boldsymbol{p}\ dt. \end{aligned}$$

Orain, eragindako matrize-normaren definizioa eta $\boldsymbol{J}$-ren γ-Lipschitz jarraitutasuna erabiliz $\boldsymbol{x}$-ren $\mathcal{D}$ ingurunean, hau dugu:

$$\begin{aligned} \|\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{p}) - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p}\| &\leq \int_0^1 \|(\boldsymbol{J}(\boldsymbol{x}+t\boldsymbol{p}) - \boldsymbol{J}(\boldsymbol{x}))\| \cdot \|\boldsymbol{p}\|\ dt \\ &\leq \int_0^1 \gamma \|t\boldsymbol{p}\| \cdot \|\boldsymbol{p}\| dt \\ &= \gamma \|\boldsymbol{p}\|^2 \int_0^1 t\ dt \\ &= \frac{\gamma}{2}\|\boldsymbol{p}\|^2. \quad \square \end{aligned}$$

Jarraian, ekuazio ez-linealen sistemetarako, Newtonen metodoaren konbergentzia koadratiko lokala frogatuko dugu. Orain, $\boldsymbol{x}$-ren r erradioko ingurunea $\mathcal{B}_r(\boldsymbol{x}) = \{\overline{\boldsymbol{x}} \in \mathbb{R}^n \mid \|\overline{\boldsymbol{x}} - \boldsymbol{x}\| < r\}$ dugu. Kontuan izan $\boldsymbol{J} \in Lip_\gamma(\mathcal{B}_r(\boldsymbol{x}))$ adierazpenak zera esan nahi duela: $\boldsymbol{J}$ γ-Lipschitz jarraitua dela $\boldsymbol{x} \in \mathcal{D}$ puntuaren $\mathcal{B}_r(\boldsymbol{x})$ ingurunean.

7.1. teorema. *Izan bedi $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ etengabe diferentziagarria $\mathcal{D} \subset \mathbb{R}^n$ multzo konbexu irekian. Demagun $\boldsymbol{x}^* \in \mathbb{R}^n$ eta $r, \beta > 0$ existitzen direla, non $\mathcal{B}_r(\boldsymbol{x}^*) \subset \mathcal{D}$, $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$, $\boldsymbol{J}(\boldsymbol{x}^*)^{-1}$ existitzen baita, $\|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}\| \leq \beta$ izanik, eta $\boldsymbol{J} \in Lip_\gamma(\mathcal{B}_r(\boldsymbol{x}^*))$. Orduan, $\varepsilon > 0$ dago, non $\boldsymbol{x}_0 \in \mathcal{B}_\varepsilon(\boldsymbol{x}^*)$ guztietarako, honako adierazpen honek sortutako $\{\boldsymbol{x}_k\}$ segida ondo definituta baitago eta $\boldsymbol{x}^*$-era jotzen baitu:*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{J}(\boldsymbol{x}_k)^{-1}\boldsymbol{f}(\boldsymbol{x}_k), \qquad k = 0, 1, \ldots$$

eta, gainera,

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \beta\gamma\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2, \qquad k = 0, 1, \ldots. \tag{7.7}$$

Frogantza. ε aukeratuko dugu $\boldsymbol{J}(\boldsymbol{x})$ ez-singularra izateko $\boldsymbol{x} \in \mathcal{B}_\varepsilon(\boldsymbol{x}^*)$ guztietan. Jarraian, eredu lineal honek:

$$\boldsymbol{M}_k(x) = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{J}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k),$$

sortutako errorea $O(\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2)$ denez gero, konbergentzia koadratikoa dela frogatuko dugu.

Izan bedi

$$\varepsilon = \min\left\{r, \frac{1}{2\beta\gamma}\right\}. \tag{7.8}$$

k-ren gaineko indukzioz (7.7) betetzen dela frogatuko dugu, eta hau ere bai:

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|$$

eta, beraz,

$$\boldsymbol{x}_{k+1} \in \mathcal{B}_\varepsilon(\boldsymbol{x}^*). \tag{7.9}$$

Lehendabizi, $\boldsymbol{J}(\boldsymbol{x}_0)$ ez dela singularra ikusiko dugu. Erabiltzen baditugu $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq \varepsilon$, $\boldsymbol{J}$-ren Lipschitz jarraitutasuna $\boldsymbol{x}^*$ puntuan eta (7.8), hau lortzen dugu:

$$\begin{aligned}\|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}[\boldsymbol{J}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}^*)]\| &\leq \|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}\| \cdot \|\boldsymbol{J}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}^*)\| \\ &\leq \beta\gamma\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq \beta \cdot \gamma \cdot \varepsilon \leq 1/2.\end{aligned}$$

Horregatik eta matrize-normaren jarraitutasunagatik, $\boldsymbol{J}(\boldsymbol{x}_0)$ ez dela singularra ateratzen da eta alderantzizkoaren normak hau betetzen duela froga daiteke (ikus [9] 3.1.4. teorema):

$$\begin{aligned}\|\boldsymbol{J}(\boldsymbol{x}_0)^{-1}\| &\leq \frac{\|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}\|}{1 - \|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}[\boldsymbol{J}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}^*)]\|} \\ &\leq 2\|\boldsymbol{J}(\boldsymbol{x}^*)^{-1}\| \\ &\leq 2 \cdot \beta.\end{aligned} \tag{7.10}$$

Beraz, $\boldsymbol{x}_1$ ondo definituta dago eta hau betetzen du:

$$\begin{aligned}\boldsymbol{x}_1 - \boldsymbol{x}^* &= \boldsymbol{x}_0 - \boldsymbol{x}^* - \boldsymbol{J}(\boldsymbol{x}_0)^{-1}\boldsymbol{f}(\boldsymbol{x}_0) \\ &= \boldsymbol{x}_0 - \boldsymbol{x}^* - \boldsymbol{J}(\boldsymbol{x}_0)^{-1}[\boldsymbol{f}(\boldsymbol{x}_0) - \boldsymbol{f}(\boldsymbol{x}^*)] \\ &= \boldsymbol{J}(\boldsymbol{x}_0)^{-1}[\boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{f}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}_0)(\boldsymbol{x}^* - \boldsymbol{x}_0)].\end{aligned}$$

Kontuan izan parentesien arteko gaia $\boldsymbol{f}(\boldsymbol{x}^*)$-ren eta $\boldsymbol{M}_0(\boldsymbol{x}^*)$ ereduaren arteko kendura dela. Beraz, 7.2. lema eta (7.10) erabiliz, hau dugu:

$$\begin{aligned}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\| &\leq \|\boldsymbol{J}(\boldsymbol{x}_0)^{-1}\|\| \cdot \boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{f}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}_0)(\boldsymbol{x}^* - \boldsymbol{x}_0)\| \\ &\leq 2 \cdot \beta \cdot \frac{\gamma}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \\ &= \beta \cdot \gamma\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2.\end{aligned}$$

Horrek (7.7) frogatzen du. Gainera, $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq 1/(2\beta\gamma)$ denez, hau lortzen dugu:

$$\|\boldsymbol{x}_1 - \boldsymbol{x}^*\| \leq \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|,$$

horrek (7.9) frogatzen du eta $k = 0$ kasua osatzen du. Indukzioaren beste urratsak frogatzeko, antzeko era batean jokatzen da. □

7.2. Newtonen metodoaren aldaketak, sistema ez-linealak ebazteko

Newtonen metodorako aldaera batzuk existitzen dira; metodoari elkartutako sistemaren definizioan eta ebazpenean desberdintzen dira. Horien guztien helburua da algoritmoaren fase hori murriztea, eta, horretarako, matrize jacobiarra hurbiltzen da era desberdinetan.

7.2.1. Diferentzia finituzko Newtonen metodoa

Jacobiarraren adierazpen analitikoa kalkulatu ezin denean, Newtonen metodoaren aldaera honek matrize jacobiarra ordezkatzen du bere diferentzia finituzko hurbilpenaz.

7.2. teorema. *Izan bitez* $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ *eta* $\boldsymbol{x}^*$ *7.1. teoremaren hipotesiak betetzen dituztenak. Orduan,* $\varepsilon, h > 0$ *existitzen dira, non baldin* $\{h_k\}$ *segida erreala,* $0 < |h_k| \leq h$, *eta*

$\boldsymbol{x}_0 \in \mathcal{B}_\varepsilon(\boldsymbol{x}^*)$ badira, honako adierazpen honek sortutako $\{\boldsymbol{x}_k\}$ segida ondo definituta dago eta $\boldsymbol{x}^*$-ra jotzen du linealki:

$$\begin{aligned} \boldsymbol{a}_j^{(k)} &= \frac{\boldsymbol{f}(\boldsymbol{x}_k + h_k\boldsymbol{e}_j) - \boldsymbol{f}(\boldsymbol{x}_k)}{h_k}, \quad j = 1, \ldots, n, \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \boldsymbol{A}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k), \qquad\qquad k = 0, 1, \ldots, \end{aligned}$$

($\boldsymbol{a}_j^{(k)}$ bektorea $\boldsymbol{A}_k$-ren j-garren zutabea da).

Baldin hau betetzen bada:

$$\lim_{k\to\infty} h_k = 0,$$

konbergentzia superlineala da.

Baldin c_1 konstante bat badago hau betetzen duena:

$$|h_k| \leq c_1 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_1,$$

edo, baliokideki, c_2 konstante bat badago hau betetzen duena:

$$|h_k| \leq c_2 \|\boldsymbol{f}(\boldsymbol{x}_k)\|_1,$$

orduan, konbergentzia koadratikoa da.

Teorema honen frogantza 7.1. teoremakoaren antzekoa da, eta [9] erreferentzian aurki daiteke.

Edonola ere, iterazio bakoitzean $\boldsymbol{J}(\boldsymbol{x})$-ren hurbilpen hori egiteko, n^2+n funtzio-balioztatzea egin behar dugu: n^2, jacobiarra kalkulatzeko, eta n, funtzioaren balioa kalkulatzeko. Sistema lineala ebazteko, $O(n^3)$ eragiketa.

Honako hau da h parametroa aukeratzeko era arrazonagarri bat. $\boldsymbol{f}(\boldsymbol{x})$ t digitu zuzenekin kalkula daitekeenean, orduan, $\boldsymbol{f}(\boldsymbol{x} + h\boldsymbol{e}_j)$ eta $\boldsymbol{f}(\boldsymbol{x})$ desberdinak izan beharko lirateke azken $t/2$ digitu horietan. Hain zuzen ere, $\boldsymbol{f}(\boldsymbol{x})$-ren kalkuluaren errore erlatiboa η balitz, hau lortu beharko genuke:

$$\frac{\|\boldsymbol{f}(\boldsymbol{x} + h\boldsymbol{e}_j) - \boldsymbol{f}(\boldsymbol{x})\|}{\|\boldsymbol{f}(\boldsymbol{x})\|} \leq \sqrt{\eta}, \qquad j = 1, \ldots, n. \tag{7.11}$$

Ez badugu informazio hoberik, (7.11) lortzeko era arrazonagarri bat x_j osagai bakoitza honela aldatzea da:

$$h_j = \sqrt{\eta} \cdot x_j, \tag{7.12}$$

eta gero, $\boldsymbol{a}_j$ zutabe bakoitza honela kalkulatzea:

$$\boldsymbol{a}_j = \frac{\boldsymbol{f}(\boldsymbol{x} + h_j\boldsymbol{e}_j) - \boldsymbol{f}(\boldsymbol{x})}{h_j}. \tag{7.13}$$

Gainera, $\boldsymbol{f}(\boldsymbol{x})$ formula erraz batez ematen denean, $\eta = \varepsilon_M$ (makinaren errorea) hartzea ere arrazonagarria da. Honako adibide honetan, Newtonen metodoan deribatuak diferentzia finituen bidez (d.d.f.) kalkulatzen dira, hots (7.12)-(7.13) erabiliz $\eta = \varepsilon_M$-rekin, eta baita deribatu analitikoak (d.a.) erabiliz ere; emaitzak hasierako puntu berdinetik abiatuz konparatzen dira. Ikus daitekeenez, emaitzak ia berdinak dira, eta hori da praktikan gertatu ohi dena. Hori dela eta, software pakete batzuek ez dute behar deribatu analitikorik; haiek beti erabiltzen dituzte diferentzia finituak.

7.3. adibidea. *Izan bedi* $\boldsymbol{f}(x) = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ e^{x_1 - 1} + x_2^3 - 2 \end{bmatrix}$. *Ebatzi* $\boldsymbol{f}(\boldsymbol{x}) = 0$, $\boldsymbol{x}_0 = [2,3]^T$ *hartuz. Dakigunez* $\boldsymbol{x}^* = [1,1]^T$.

Ebazpena. Kalkuluak 14 digitu esanguratsuko ordenagailu batean egin dira. Ondorioz, $\eta = \varepsilon_M = 10^{-14}$ dugu; beraz, $h_j = 10^{-7}|x_j|$.

Newtonen metodoa d.a. erabiliz	$\boldsymbol{x}_k$	Newtonen metodoa d.d.f. erabiliz
Âů$[2,3]^T$	$\boldsymbol{x}_0$	$[2,3]^T$
$\begin{bmatrix} 0.57465515807608 \\ 2.1168965612826 \end{bmatrix}$	$\boldsymbol{x}_1$	$\begin{bmatrix} 0.57465515450268 \\ 2.1168966735234 \end{bmatrix}$
$\begin{bmatrix} 0.31178766389307 \\ 1.5241979559460 \end{bmatrix}$	$\boldsymbol{x}_2$	$\begin{bmatrix} 0.31178738552306 \\ 1.5241981016335 \end{bmatrix}$
$\begin{bmatrix} 1.4841388323960 \\ 1.1464779176945 \end{bmatrix}$	$\boldsymbol{x}_3$	$\begin{bmatrix} 1.4841386151178 \\ 1.1464781318492 \end{bmatrix}$
$\begin{bmatrix} 1.0592959013664 \\ 1.0348194625183 \end{bmatrix}$	$\boldsymbol{x}_4$	$\begin{bmatrix} 1.0592958450507 \\ 1.0348195092235 \end{bmatrix}$
$\begin{bmatrix} 1.0008031050945 \\ 1.0014625483617 \end{bmatrix}$	$\boldsymbol{x}_5$	$\begin{bmatrix} 1.0008031056081 \\ 1.0014625533494 \end{bmatrix}$
$\begin{bmatrix} 0.99999872187461 \\ 1.0000026672636 \end{bmatrix}$	$\boldsymbol{x}_6$	$\begin{bmatrix} 0.99999872173640 \\ 1.0000026674316 \end{bmatrix}$
$\begin{bmatrix} 0.99999999999548 \\ 1.0000000000089 \end{bmatrix}$	$\boldsymbol{x}_7$	$\begin{bmatrix} 0.99999999999535 \\ 1.0000000000091 \end{bmatrix}$
4	$[\boldsymbol{J}(\boldsymbol{x}_0)]_{11}$	4.0000003309615
6	$[\boldsymbol{J}(\boldsymbol{x}_0)]_{12}$	6.0000002122252
2.7182818284590	$[\boldsymbol{J}(\boldsymbol{x}_0)]_{21}$	2.7182824169358
27	$[\boldsymbol{J}(\boldsymbol{x}_0)]_{22}$	27.000002470838

Baina, $\boldsymbol{f}(\boldsymbol{x})$ kalkulatzeko adierazpen luzeko kode bat edo prozedimendu iteratibo bat erabili behar dugunean, $\eta \gg \varepsilon_M$ izan daiteke, eta diferentzia finituzko deribatuak ez dira

izango oso zehatzak; orduan hobe da ebakitzailearen metodo bat erabiltzea (geroago ikusiko duguna). Gainera, kasu horretan, $\boldsymbol{A}$ kalkulatzeko iterazio bakoitzean $\boldsymbol{f}(\boldsymbol{x})$-ren n ebaluatze beharko dira, eta horren kostua handia da.

7.2.2. Newtonen metodo aldatua

Lehenengo aldaera matrize jacobiar berdina, $\boldsymbol{J}(\boldsymbol{x}_0)$, prozesu iteratibo osoan izatean datza, edo, iterazio kopuru finko batean zehar, gutxienez.

7.2.3. Jacobiren aldaera

Aldaera honetan, diagonal nagusiko gaien bidez hurbiltzen da matrize jacobiarra. Hots,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{D}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k),$$

non $\boldsymbol{D}_k = \operatorname{diag}\{d_{11}^{(k)}, \ldots, d_{nn}^{(k)}\}$ eta $d_{ii}^{(k)} = [\boldsymbol{J}(\boldsymbol{x}_k)]_{ii},\ i = 1, \ldots, n$. Ikus daitekeenez, sistema linealetarako Jacobiren metodoaren antzekoa da. Diagonalean ez dauden gaiak txikiak badira diagonaleko gaiekiko (hots, diagonal menperatzailea bada), interesgarria izango da metodo hau.

7.2.4. Gauss-Seidelen aldaera

Aldaera honetan, matrize jacobiarra bere diagonalarekin eta diagonalaren azpiko azpimatrizearekin hurbiltzen da (diagonalaren gaineko gaiak ez dira erabiltzen). Hots,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{L}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k),$$

non $[\boldsymbol{L}_k]_{ij} = [\boldsymbol{J}(\boldsymbol{x}_k)]_{ij},\quad i \geq j$. Iterazio bakoitzean $\boldsymbol{L}_k\boldsymbol{p}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$ sistema ebatziko da aurrerantz eta, gero, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$ egingo da.

7.3. Quasi-Newton metodoak

Diferentzia finituzko Newtonen metodoko iterazio bakoitzean, $\boldsymbol{J}(\boldsymbol{x})$-ren hurbilpena egiteko, $n^2 + n$ funtzio-balioztatze egin behar ditugu (n^2, jacobiarra kalkulatzeko, eta n, funtzioaren balioa kalkulatzeko), eta sistema lineala ebazteko $O(n^3)$ eragiketa gehiago. Ahalegin konputazionalaren kopuru hori handiegia da, n txikia izan ezean.

Atal honetan ikusiko dugu sistema ez-linealak ebazteko ebakitzailearen metodoaren hedatze arrakastatsuena. Metodo honek iterazio bakoitzean n funtzio-balioztatze bakarrik behar izango ditu, eta $O(n^2)$ eragiketa aritmetiko.

Gogora dezagun ebakitzailearen metodoan, Newtonen metodoaren $f'(x_1)$-en ordez, adierazpen hau erabiltzen genuela:

$$\frac{f(x_1)-f(x_0)}{x_1-x_0}.$$

Aldagai anitzeko $\boldsymbol{f}(\boldsymbol{x})$ funtzioaren deribatua $\boldsymbol{x}_1$ puntuan, $\boldsymbol{J}(\boldsymbol{x}_1)$, $\boldsymbol{A}_1$ matrizeaz ordezkatzen da propietate honekin:

$$\boldsymbol{A}_1(\boldsymbol{x}_1-\boldsymbol{x}_0)=\boldsymbol{f}(\boldsymbol{x}_1)-\boldsymbol{f}(\boldsymbol{x}_0).$$

Berdintza horri *ebakitzailearen ekuazio* deritzogu. Gainera, $\boldsymbol{s}_k=\boldsymbol{x}_{k+1}-\boldsymbol{x}_k$ notazioa erabiliko dugu k-garren iterazioari dagokion urratsa adierazteko, eta $\boldsymbol{y}_k=\boldsymbol{f}(\boldsymbol{x}_{k+1})-\boldsymbol{f}(\boldsymbol{x}_k)$ notazioa urrats horri dagokion funtzioaren balioaren aldakuntza emateko. Beraz, k-garren iterazioan, $\boldsymbol{A}_{k+1}$ matrizeak ebakitzailearen ekuazio hau bete behar du:

$$\boldsymbol{A}_{k+1}\boldsymbol{s}_k=\boldsymbol{y}_k. \tag{7.14}$$

(7.14) ekuazioa ez da nahikoa espezifikatzeko $\boldsymbol{A}_{k+1}$ bakar bat $n>1$ denean. Egia esan, (7.14) n ekuazioko eta n^2 ezezaguneko sistema bat da. Horregatik, ebakitzaile-hurbilpen arrakastatsu bat eraikitzea da posibilitate horien artean hautatzeko irizpide egokiena aukeratzea.

Quasi-Newton metodoek $\{\boldsymbol{A}_k\}$ segida bat eraikitzen dute, non $\boldsymbol{A}_k$ matrizeak $\boldsymbol{J}(\boldsymbol{x}_k)$ matrize jacobiarra ahalik eta hoberen hurbiltzen baitu. Alegia, k-garren iterazioan $\boldsymbol{f}(\boldsymbol{x})$ ordezkatzen dugu hurbilpen lineal honekin:

$$\boldsymbol{M}_k(\boldsymbol{x})=\boldsymbol{f}(\boldsymbol{x}_k)+\boldsymbol{A}_k(\boldsymbol{x}-\boldsymbol{x}_k),$$

zeina eredu afina baita.

7.3.1. Broyden-en metodoa

Broydenek ideia erraz bat erabili zuen $\boldsymbol{A}_{k+1}$-en hurbilpen egokia lortzeko (ikus [6]): iterazio berrian, $\boldsymbol{A}_k$ matrizeak jacobiarrari buruz ematen digun informazio gehiena gordetzea. Alegia, $\boldsymbol{A}_{k+1}$ aukeratzeko orduan, saiatuko gara minimizatzen eredu afinaren aldakuntza k-garren eredu afinarekiko, hots:

$$\begin{aligned}
\boldsymbol{M}_{k+1}(\boldsymbol{x})-\boldsymbol{M}_k(\boldsymbol{x}) &= [\boldsymbol{f}(\boldsymbol{x}_{k+1})+\boldsymbol{A}_{k+1}(\boldsymbol{x}-\boldsymbol{x}_{k+1})]-[\boldsymbol{f}(\boldsymbol{x}_k)+\boldsymbol{A}_k(\boldsymbol{x}-\boldsymbol{x}_k)]\\
&= f(\boldsymbol{x}_{k+1})+\boldsymbol{A}_{k+1}[(\boldsymbol{x}-\boldsymbol{x}_k)-(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)]-[\boldsymbol{f}(\boldsymbol{x}_k)+\boldsymbol{A}_k(\boldsymbol{x}-\boldsymbol{x}_k)]\\
&= f(\boldsymbol{x}_{k+1})+\boldsymbol{A}_{k+1}(\boldsymbol{x}-\boldsymbol{x}_k)-\boldsymbol{A}_{k+1}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)-\boldsymbol{f}(\boldsymbol{x}_k)-\boldsymbol{A}_k(\boldsymbol{x}-\boldsymbol{x}_k)\\
&= f(\boldsymbol{x}_{k+1})-\boldsymbol{f}(\boldsymbol{x}_k)-\boldsymbol{A}_{k+1}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)+(\boldsymbol{A}_{k+1}-\boldsymbol{A}_k)(\boldsymbol{x}-\boldsymbol{x}_k)\\
&= \boldsymbol{y}_k-\boldsymbol{A}_{k+1}\boldsymbol{s}_k+(\boldsymbol{A}_{k+1}-\boldsymbol{A}_k)(\boldsymbol{x}-\boldsymbol{x}_k) \quad \text{(eta (7.14) kontuan hartuz)}\\
&= (\boldsymbol{A}_{k+1}-\boldsymbol{A}_k)(\boldsymbol{x}-\boldsymbol{x}_k).
\end{aligned}$$

Hortaz, minimizatu behar duguna hau izango da:

$$(\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)(\boldsymbol{x} - \boldsymbol{x}_k).$$

Orain, $\boldsymbol{x} \in \mathbb{R}^n$ guztietarako, beti adieraz dezakegu honela $\boldsymbol{x} - \boldsymbol{x}_k$ kendura:

$$\boldsymbol{x} - \boldsymbol{x}_k = \alpha \boldsymbol{s}_k + \boldsymbol{t},$$

non $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k (\neq \boldsymbol{0})$ eta $\boldsymbol{t}^T \boldsymbol{s}_k = 0$ (hots, $\boldsymbol{t} \in \mathbb{R}^n$ da $\boldsymbol{s}_k \in \mathbb{R}^n$ bektorearekiko $n-1$ dimentsioko azpiespazio ortogonalaren edozein bektore). Ondorioz, $\boldsymbol{M}_{k+1}(\boldsymbol{x}) - \boldsymbol{M}_k(\boldsymbol{x})$ minimizatzeko, hau minimizatu behar dugu:

$$\alpha(\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)\boldsymbol{s}_k + (\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)\boldsymbol{t}. \tag{7.15}$$

Lehenengo gaiaz ezin dugu ezer egin, zeren (7.14), ebakitzailearen ekuazioaren arabera, hau gertatzen baita:

$$(\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)\boldsymbol{s}_k = \boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k, \tag{7.16}$$

hau da, emaitza-bektorea konstantea da (k-garren iterazioan datu horiek guztiak finkatuta daude). Orain, (7.15) adierazpenaren bigarren gaiaz, hau eska dezakegu:

$$(\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)\boldsymbol{t} = \boldsymbol{0} \qquad \forall \boldsymbol{t} \in \mathbb{R}^n \text{ non } \boldsymbol{s}_k^T \boldsymbol{t} = 0.$$

Hori lor dezakegu $\boldsymbol{A}_{k+1}$ egoki bat aukeratuz. Berdintza horrek hau betetzera behartzen du:

$$\boldsymbol{A}_{k+1} - \boldsymbol{A}_k = \boldsymbol{u}\boldsymbol{s}_k^T \qquad \left(= \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} \boldsymbol{s}_k^T = \begin{bmatrix} u_1 \boldsymbol{s}_k^T \\ \vdots \\ u_n \boldsymbol{s}_k^T \end{bmatrix} \right) \tag{7.17}$$

non $\boldsymbol{u} \in \mathbb{R}^n$. Alegia, $\boldsymbol{A}_{k+1} - \boldsymbol{A}_k$ bat heinakoa izan behar du. Orain, (7.14) ebakitzailearen ekuazioa betetzea (7.16) betetzearen baliokidea da, eta, orduan, $\boldsymbol{u}$ bektoreak hau bete behar du:

$$(\boldsymbol{u}\boldsymbol{s}_k^T)\boldsymbol{s}_k = \boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k,$$

ondorioz, ebakitzailearen ekuazioa betetzeko hau izango dugu:

$$\boldsymbol{u} = \frac{\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k}{\boldsymbol{s}_k^T \boldsymbol{s}_k}.$$

Beraz, (7.17) ekuazioko $\boldsymbol{A}_{k+1}$ bakanduz eta $\boldsymbol{u}$ bektore hori ordezkatuz, hau lortzen da:

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \boldsymbol{u}\boldsymbol{s}_k^T = \boldsymbol{A}_k + \frac{\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k}{\boldsymbol{s}_k^T \boldsymbol{s}_k} \boldsymbol{s}_k^T. \tag{7.18}$$

Ondorioz, (7.18) eguneratzeak ondoz ondoko bi eredu afinen diferentzia txikiagotzen du, eta ebakitzailearen ekuazioa betetzen du.

Newtonen sistema hurbilduaren ebazpena

Bat heineko Broydenen eguneratzea erabiltzen badugu, (7.18), n funtzio-balioztatze bakarrik erabili behar dugu iterazio bakoitzean (diferentzia finituen kasuan, n^2+n behar ditugu). Bestalde, $\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{p} = -\boldsymbol{f}(\boldsymbol{x}_k)$ Newtonen sistemaren ordez, sistema hau ebatzi behar dugu:

$$\boldsymbol{A}_k\boldsymbol{p} = -\boldsymbol{f}(\boldsymbol{x}_k),$$

eta sistema hori ebazteko, $O(n^3)$ eragiketak erabili behar ditugu oraindik. Arazo hori gainditzeko, matrizeen propietate bat erabiliz, formula erraz baten bidez lortuko dugu $\boldsymbol{A}_k$-ren alderantzizkoa.

7.3. lema. *(Sherman-Morrison-Woodbury) Izan bitez $\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^n$ bektoreak eta demagun $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ matrizea ez dela singularra. Orduan, $\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T$ ez da singularra, baldin eta soilik baldin hau betetzen bada:*

$$\sigma = 1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0.$$

Gainera, hau betetzen da:

$$(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T)^{-1} = \boldsymbol{A}^{-1} - \frac{1}{\sigma}\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}.$$

Lema hori (7.18) adierazpenari zuzenean aplikatzeak hau ematen digu:

$$\begin{aligned}
\boldsymbol{A}_{k+1}^{-1} &= (\boldsymbol{A}_k + \boldsymbol{u}\boldsymbol{s}_k^T)^{-1} = \boldsymbol{A}_k^{-1} - \frac{\boldsymbol{A}_k^{-1}\boldsymbol{u}\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}}{1+\boldsymbol{s}_k^T\boldsymbol{A}^{-1}\boldsymbol{u}} \\
&= \boldsymbol{A}_k^{-1} - \frac{\boldsymbol{A}_k^{-1}\left(\dfrac{\boldsymbol{y}_k-\boldsymbol{A}_k\boldsymbol{s}_k}{\boldsymbol{s}_k^T\boldsymbol{s}_k}\right)\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}}{1+\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}\left(\dfrac{\boldsymbol{y}_k-\boldsymbol{A}_k\boldsymbol{s}_k}{\boldsymbol{s}_k^T\boldsymbol{s}_k}\right)} \\
&= \boldsymbol{A}_k^{-1} - \frac{(\boldsymbol{A}_k^{-1}\boldsymbol{y}_k-\boldsymbol{s}_k)\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}}{\boldsymbol{s}_k^T\boldsymbol{s}_k+\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}\boldsymbol{y}_k-\boldsymbol{s}_k^T\boldsymbol{s}_k}
\end{aligned}$$

eta, azkenik, *Broydenen alderantzizko formula* hau lortzen da:

$$\boldsymbol{A}_{k+1}^{-1} = \boldsymbol{A}_k^{-1} + \frac{(\boldsymbol{s}_k-\boldsymbol{A}_k^{-1}\boldsymbol{y}_k)\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}}{\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}\boldsymbol{y}_k}. \tag{7.19}$$

Adierazpen horrek $O(n^2)$ eragiketa behar ditu eta, $\boldsymbol{s}_k = -\boldsymbol{A}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k)$ kalkulatzeko, $O(n^2)$ eragiketa egin behar ditugu. Beraz, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{A}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k)$ iterazio berria kalkulatzeko, guztira $O(n^2)$ eragiketa egin behar ditugu, $O(n^3)$ erabili beharrean.

Froga daiteke Broydenen metodoaren konbergentzia lokala superlineala dela; ikus [9].

7.2. algoritmoa. Quasi-Newton metodoa, Broydenen eguneratzeaz.

0. urratsa. SARRERA. Sartu: $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{x}_0$ (hasierako puntua), $\boldsymbol{A}_0^{-1}$, $ze > 0$ (zehaztasun erlatiboa), $emax > 0$ (errorearen tolerantzia) eta $imax$ (iterazioen kopuru maximoa). Kalkulatu $\boldsymbol{f}(\boldsymbol{x}_0)$. Jarri $k = 0$.

1. urratsa. Kalkulatu hau:

$$\boldsymbol{s}_k = -\boldsymbol{A}_k^{-1}\boldsymbol{f}(\boldsymbol{x}_k). \tag{7.20}$$

2. urratsa. Kalkulatu honako puntu hau:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k.$$

3. urratsa. Kalkulatu $\boldsymbol{f}(\boldsymbol{x}_{k+1})$.

4. urratsa. Baldin $\|\boldsymbol{s}_k\|/\|\boldsymbol{x}_{k+1}\| < ze$ edo $\|\boldsymbol{f}(\boldsymbol{x}_{k+1})\| < emax$ edo $k \geq imax$ bada, gelditu egiten da. Bestela, hurrengo urratsera doa.

5. urratsa. Kalkula itzazu $\boldsymbol{y}_k = f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)$ eta, gero, $\boldsymbol{A}_{k+1}^{-1}$ honela:

$$\boldsymbol{A}_{k+1}^{-1} = \boldsymbol{A}_k^{-1} + \frac{(\boldsymbol{s}_k - \boldsymbol{A}_k^{-1}\boldsymbol{y}_k)\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}}{\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}\boldsymbol{y}_k}.$$

6. urratsa. Egin $k = k + 1$ eta joan 1. urratsera.

7. urratsa. IRTEERA. Emaitza: $\boldsymbol{x}_{k+1}$.

Ohartarazi behar da $\boldsymbol{A}_0$ hasierako matrizea identitate-matrizea edo hasierako jacobiarra edo bere diagonala izan daitekeela; edo, algoritmo horretaz esperientzia badugu, matrize egoki bat.

7.4. adibidea. *Izan bedi*

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix},$$

zeinak $[0, 3]^T$ *eta* $[3, 0]^T$ *erroak baititu. Izan bedi* $\boldsymbol{x}_0 = [1, 5]^T$ *eta aplikatu aurreko algoritmoa hau hartuz:*

$$\boldsymbol{A}_0^{-1} = \boldsymbol{J}(\boldsymbol{x}_0)^{-1} = \begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix}^{-1} = \begin{bmatrix} 1.2500 & -0.1250 \\ -0.2500 & 0.1250 \end{bmatrix}.$$

Ebazpena. Algoritmoaren 1. eta 2. urratsak egiten dira:

$$\boldsymbol{s}_0 = -\boldsymbol{A}_0^{-1}\boldsymbol{f}(\boldsymbol{x}_0) = -\begin{bmatrix} 1.2500 & -0.1250 \\ -0.2500 & 0.1250 \end{bmatrix}\begin{bmatrix} 3 \\ 17 \end{bmatrix} = \begin{bmatrix} -1.625 \\ -1.375 \end{bmatrix}.$$

3.ak zera ematen digu:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{s}_0 = \begin{bmatrix} 1 \\ 5 \end{bmatrix} + \begin{bmatrix} -1.625 \\ -1.375 \end{bmatrix} = \begin{bmatrix} -0.625 \\ 3.625 \end{bmatrix}.$$

4.aren bidez, erraz ikus daiteke soluzioa oraindik ez dugula aurkitu hemen:

$$\boldsymbol{f}(\boldsymbol{x}_1) = \begin{bmatrix} 0 \\ 4.53125 \end{bmatrix}.$$

Beraz, algoritmoaren 5. urratsa burutzerakoan, emaitza hauek lortzen dira:

$$\boldsymbol{y}_0 = \boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_0) = \begin{bmatrix} 0 \\ 4.53125 \end{bmatrix} - \begin{bmatrix} 3 \\ 17 \end{bmatrix} = \begin{bmatrix} -3 \\ -12.46875 \end{bmatrix},$$

$$\boldsymbol{z} = -\boldsymbol{A}_0^{-1}\boldsymbol{y}_0 = -\begin{bmatrix} 1.2500 & -0.1250 \\ -0.2500 & 0.1250 \end{bmatrix}\begin{bmatrix} -3 \\ -12.46875 \end{bmatrix} = \begin{bmatrix} 2.19141 \\ 0.80859 \end{bmatrix},$$

$$p = -\boldsymbol{s}_0^T\boldsymbol{z} = -\begin{bmatrix} -1.625 \\ -1.375 \end{bmatrix}\begin{bmatrix} 2.19141 \\ 0.80859 \end{bmatrix} = 4.6729,$$

$$\boldsymbol{C} = \frac{(\boldsymbol{s}_0 + \boldsymbol{z})\boldsymbol{s}_0^T\boldsymbol{A}_0^{-1}}{p} = \frac{\left(\begin{bmatrix} -1.625 \\ -1.375 \end{bmatrix} + \begin{bmatrix} 2.19141 \\ 0.80859 \end{bmatrix}\right)[-1.625 \quad -1.375]\begin{bmatrix} 1.2500 & -0.1250 \\ -0.2500 & 0.1250 \end{bmatrix}}{4.6729}$$

$$= \begin{bmatrix} -0.2045 & 0.0038 \\ 0.2045 & -0.0038 \end{bmatrix},$$

$$\boldsymbol{A}_1^{-1} = \boldsymbol{A}_0^{-1} + \boldsymbol{C} = \begin{bmatrix} 1.2500 & -0.1250 \\ -0.2500 & 0.1250 \end{bmatrix} + \begin{bmatrix} -0.2045 & 0.0038 \\ 0.2045 & -0.0038 \end{bmatrix} = \begin{bmatrix} 1.0455 & -0.1212 \\ -0.0455 & 0.1212 \end{bmatrix}.$$

Berriro 2. urratsa aplikatuz, zera lortzen da:

$$\boldsymbol{s}_1 = -\boldsymbol{A}_1^{-1}\boldsymbol{f}(\boldsymbol{x}_1) = -\begin{bmatrix} 1.0455 & -0.1212 \\ -0.0455 & 0.1212 \end{bmatrix}\begin{bmatrix} 0 \\ 4.53125 \end{bmatrix} = \begin{bmatrix} 0.5492 \\ -0.5492 \end{bmatrix}.$$

Eta 3. urratsaren bitartez, hau:

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{s}_1 = \begin{bmatrix} -0.625 \\ 3.625 \end{bmatrix} + \begin{bmatrix} 0.5492 \\ -0.5492 \end{bmatrix} = \begin{bmatrix} -0.07575 \\ 3.07575 \end{bmatrix}.$$

Ikus dezakegunez, $\boldsymbol{x}_2 \approx [0 \quad 3]^T$. $\square$

Praktikan, inplementazio hau ondo dabil, baina desabantaila bat dauka: nekez detektatzen da $\boldsymbol{A}_{k+1}$-en baldintzapen txarra. $\boldsymbol{A}_k$-ren $\boldsymbol{QR}$ faktorizazioak arazo hori ez duenez eta erabiltzen dituen eragiketen kopurua antzekoa denez ($O(n^2)$), gaur egun, (7.19) formula ordezkatu egin du. Metodo horretan, $\boldsymbol{A}_k = \boldsymbol{QR}$ faktorizazioa dugunez, erraz lortzen da $\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \boldsymbol{u}\boldsymbol{s}_k^T = \boldsymbol{QR} + \boldsymbol{u}\boldsymbol{s}_k^T = \overline{\boldsymbol{Q}}\ \overline{\boldsymbol{R}}$ faktorizazioa. Horretarako, Givensen biraketak (transformazio ortogonalak) erabiltzen dira; ikus 6.3. atala.

7.4. Murrizketarik gabeko optimizazioa

Izan bedi $F(\boldsymbol{x})$ aldagai anitzeko funtzio eskalar bat eta $\boldsymbol{x} \in \mathbb{R}^n$. Orduan, murrizketarik gabeko optimizazioan, honelako problema dugu:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x}).$$

Orokorki, ez dakigu zenbat minimo lokal daukan eta nola aurkitu minimo globala era eraginkorrean, hots, minimo guztietako $F(\boldsymbol{x})$ funtzioaren balio txikiena ematen duena. Bestalde, bistan dago baliokidetasun hau:

$$\max_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x}) \iff \min_{\boldsymbol{x}\in\mathbb{R}^n} -F(\boldsymbol{x}).$$

Aurrez ikusitako sistema linealen/ez-linealen teoria oso ondo datorkigu murrizketarik gabeko optimizazio-problemak ebazteko.

7.3. Aldagai anitzetarako Taylorren teorema. *Izan bedi $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ eta demagun $F(\boldsymbol{x}) \in \mathcal{C}^3$ funtzioa dugula. Orduan, $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)^T$ urrats bektore baterako, Taylorren garapenak hau ematen digu:*

$$F(\boldsymbol{x}+\boldsymbol{d}) = F(\boldsymbol{x}) + \nabla F(\boldsymbol{x})^T\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T\nabla^2 F(\boldsymbol{x})\boldsymbol{d} + O(\|\boldsymbol{d}\|^3). \tag{7.21}$$

Teorema horretan, $\nabla F(\boldsymbol{x})$ *gradiente bektorea* da eta $\nabla^2 F(\boldsymbol{x})$ *matrize hessiarra*, eta honela adieraz ditzakegu:

$$\nabla F(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \cdots \\ \frac{\partial F}{\partial x_n} \end{bmatrix}, \qquad \nabla^2 F(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 x_2} & \cdots & \frac{\partial^2 F}{\partial x_1 x_n} \\ \frac{\partial^2 F}{\partial x_2 x_1} & \frac{\partial^2 F}{\partial x_2^2} & \cdots & \frac{\partial^2 F}{\partial x_2 x_n} \\ & & \ddots & \\ \cdots & \cdots & & \cdots \\ \frac{\partial^2 F}{\partial x_n x_1} & \frac{\partial^2 F}{\partial x_n x_2} & \cdots & \frac{\partial^2 F}{\partial x_n^2} \end{bmatrix}.$$

Bestalde, teoremako baldintzetan F*-ren* $\boldsymbol{d}$ *norabidearekiko deribatua* $\boldsymbol{x}$ *puntuan* honela kalkula daiteke:

$$\nabla F(\boldsymbol{x})^T\boldsymbol{d} = \sum_{i=1}^{n} d_i \frac{\partial F}{\partial x_i}.$$

Gainera, zera dugu:

$$\frac{1}{2}\boldsymbol{d}^T\nabla^2 F(\boldsymbol{x})\boldsymbol{d} = \frac{1}{2}\sum_{i=1,j=1}^{n} \frac{\partial^2 F}{\partial x_i \partial x_j} d_i d_j.$$

Hortaz, $\boldsymbol{x}^*$ minimo lokala bada, hots, puntu horren ingurunean F funtzioak puntu horretan balio txikiena hartzen badu, orduan $\boldsymbol{d}$ guztietarako hau dugu:

$$F(\boldsymbol{x}^*+\boldsymbol{d}) = F(\boldsymbol{x}^*) + \nabla F(\boldsymbol{x}^*)^T\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T\nabla^2 F(\boldsymbol{x}^*)\boldsymbol{d} + O(\|\boldsymbol{d}\|^3) \geq F(\boldsymbol{x}^*).$$

Orduan, $\boldsymbol{x}^*$ minimoa izateko hau da baldintza beharrezkoa:

$$\nabla F(\boldsymbol{x}^*) = \mathbf{0}.$$

Berdintza hori betetzen duten puntuei *puntu kritikoak* deritze.

Gainera, $\boldsymbol{x}^*$ puntu kritikoak minimoa izateko bete behar duen *baldintza nahikoa* $\nabla^2 F(\boldsymbol{x}^*)$ matrize hessiarra definitu positiboa izatea da.

Ikus ditzagun baldintza horiek teorema honetan.

7.4. teorema: murrizketarik gabeko minimizazioaren baldintzak.

Demagun $F \in \mathcal{C}^3$. Orduan zera dugu:

- *$\boldsymbol{x}^*$ puntuan F funtzioak minimo lokal bat izateko, baldintza beharrezkoa da $\boldsymbol{x}^*$ puntu kritikoa izatea eta $\nabla^2 F(\boldsymbol{x}^*)$ matrize hessiarra erdidefinitu positiboa izatea.*
- *$\boldsymbol{x}^*$ puntuan F funtzioak minimo lokal bat izateko, baldintza nahikoa da $\boldsymbol{x}^*$ puntu kritikoa izatea eta $\nabla^2 F(\boldsymbol{x}^*)$ matrize hessiarra definitu positiboa izatea.*

Frogantza. Izan bedi $\|\boldsymbol{d}\|$ oso txikia, $\|\boldsymbol{d}\|^2 \ll \|\boldsymbol{d}\|$ betetzeko. Orduan, $\nabla F(\boldsymbol{x}^*) \neq \mathbf{0}$ bada, beti da posible $\boldsymbol{d}$ norabide bat aurkitzea $\nabla F(\boldsymbol{x}^*)^T \boldsymbol{d} < 0$ gertatzeko, eta, hortaz, $F(\boldsymbol{x}^* + \boldsymbol{d}) < F(\boldsymbol{x}^*)$ eta $\boldsymbol{x}^*$ ez da minimoa izango.

Bestalde, $\boldsymbol{x}^*$ puntu kritikoa minimo lokal hertsia izateko $\boldsymbol{d}$ norabide guztietarako, non $0 < \|\boldsymbol{d}\| \ll 1$, hau bete behar da:

$$F(\boldsymbol{x}^* + \boldsymbol{d}) = F(\boldsymbol{x}^*) + \frac{1}{2}\boldsymbol{d}^T \nabla^2 F(\boldsymbol{x}^*)\boldsymbol{d} + O(\|\boldsymbol{d}\|^3) > F(\boldsymbol{x}^*)$$

eta hori gertatuko da $\nabla^2 F(\boldsymbol{x}^*)$ matrize hessiarra definitu positiboa bada.

Baldintza beharrezkorako, $\boldsymbol{x}^*$ minimo lokala bada, $F(\boldsymbol{x}^* + \boldsymbol{d}) \geq F(\boldsymbol{x}^*)$ gertatu behar da, eta, beraz, matrize hessiarrak ezin du $\boldsymbol{d}^T \nabla^2 F(\boldsymbol{x}^*)\boldsymbol{d} < 0$ bete inolako $\boldsymbol{d}$-tarako, hots, matrize hessiarra erdidefinitu positiboa izan behar da. □

7.4.1. Oinarrizko metodoak

Puntu kritikoa izateko baldintzak ekuazio ez-linealen sistema bat ematen du:

$$\boldsymbol{f}(\boldsymbol{x}) \equiv \nabla F(\boldsymbol{x}) = \mathbf{0}.$$

Hortaz, puntu kritiko bat aurkitzea da ekuazio ez-linealen sistema baten ebazpenaren kasu berezi bat, eta Newtonen metodoa eta bere antzekoak zuzen erabil daitezke. Argi dago

kasu honetan $J(\boldsymbol{x}) = \nabla^2 F(\boldsymbol{x})$, hots, matrize jacobiarra matrize hessiarra dela. Baina, orain, jacobiarra matrize simetrikoa da, eta minimoaren ondoan definitu positiboa izaten da; hori garrantzitsua da.

Newtonen metodoa

7.3. algoritmoa. Murrizketarik gabeko minimizaziorako Newtonen metodoa.

0. urratsa. Jarri $k = 0$.

1. urratsa. $\nabla^2 F(\boldsymbol{x}_k)\boldsymbol{d}_k = -\nabla F(\boldsymbol{x}_k)$ ebatziz $\boldsymbol{d}_k$ lortzen da.

2. urratsa. Jarri $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$.

3. urratsa. Baldin bukaerako baldintzak betetzen badira (esate baterako, $\|\nabla F(\boldsymbol{x}_k)\| < \varepsilon$), emaitza $\boldsymbol{x}_{k+1}$ da. Bestela, $k = k + 1$ jarri eta segi 1. urratsean.

$\boldsymbol{x}$ puntu baterako esango dugu F-rekiko $\boldsymbol{d}$ norabidea *beherakorra* dela, $\nabla F(\boldsymbol{x})^T\boldsymbol{d} < 0$ bada. Orduan (7.21) Taylorren formula kontuan hartuz, $\lambda > 0$ nahiko txiki baterako, $F(\boldsymbol{x} + \lambda\boldsymbol{d}) < F(\boldsymbol{x})$ dugu.

Bestalde, $\boldsymbol{x}^*$ puntu kritikoa zeladura-puntua izan daiteke eta, orduan, puntu horretan existituko dira $\boldsymbol{d}$ urrats beherakorrak ($F(\boldsymbol{x}^* + \boldsymbol{d}) < F(\boldsymbol{x}^*)$) eta $\boldsymbol{d}$ urrats gorakorrak ($F(\boldsymbol{x}^* + \boldsymbol{d}) > F(\boldsymbol{x}^*)$). Hori gertatuko da $\boldsymbol{x}^*$ puntu kritikoan matrize hessiarra indefinitua denean.

Metodo mota bat

Murrizketarik gabeko metodo gehienek honelako iterazioak erabiltzen dituzte:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k\boldsymbol{d}_k, \quad \text{non} \quad \boldsymbol{d}_k = -\boldsymbol{A}_k^{-1}\nabla F(\boldsymbol{x}_k),$$

$\lambda_k > 0$ nahiko txikia hartzen da $F(\boldsymbol{x}_k + \lambda_k\boldsymbol{d}_k) < F(\boldsymbol{x}_k)$ betetzeko.

$\boldsymbol{A}_k$ simetriko definitu positiboa bada, orduan nahitaez $\boldsymbol{d}_k$ norabide beherakorra da, zeren $\nabla F(\boldsymbol{x}_k)^T\boldsymbol{d} = -\nabla F(\boldsymbol{x}_k)^T\boldsymbol{A}_k^{-1}\nabla F(\boldsymbol{x}_k) < 0$. Geroago, $\boldsymbol{A}_k$ hautatzeko aukera desberdinak har ditzakegu kontuan.

Bilaketa metodo bat justifikatzeko, merezi du Newton metodoaren abantailak eta desabantailak aztertzea. Abantailak: konbergentzia lokal koadratikoa. Desabantailak:

- Metodoak matrize hessiarraren existentzia eskatzen du.

- Okerrago, beharrezkoa da hessiarra balioztatzea.
- Iterazio bakoitzean ekuazio linealen sistema bat ebatzi behar da.
- Minimotik kanpoan hessiarra ez-definitu positiboa izan daiteke.
- Konbergentzia ezin da kontrolatu (konbergitzen da minimo batera?).

Horrelako arazoak gainditzeko, honako metodo hauek ikusiko ditugu.

7.4.2. Metodo globalak

Orain arte erabilitako metodoak lokalak izan dira, hots, $\boldsymbol{x}_0$ hasierako puntuak $\boldsymbol{x}^*$ errotik nahiko hurbil egon behar du, metodo horiek konbergenteak izateko. Baina, hori gertatzen ez denean, zein metodo erabil dezakegu?

Gradiente metodoa

Dakigunez, $\nabla F(\boldsymbol{x})^T\boldsymbol{d} = \|\nabla F(\boldsymbol{x})\|_2\|\boldsymbol{d}\|_2\cos(\theta)$ dugu (θ da $\nabla F(\boldsymbol{x})$ eta $\boldsymbol{d}$ bektoreen arteko angelua), eta $\|\boldsymbol{d}\|_2 = 1$ bada, $\nabla F(\boldsymbol{x})^T\boldsymbol{d}$ biderkadurak balio txikiena hartuko du $\theta = -\pi$ denean, hots, $\boldsymbol{d} = -\nabla F(\boldsymbol{x})/\|\nabla F(\boldsymbol{x})\|_2$ denean. Orokorki, $\boldsymbol{d} = -\nabla F(\boldsymbol{x})$ norabidea aukeratzen denean, *gradientearen metodoa* (edo *beherapen azkarreneko metodoa*) dugu. Orduan, $\boldsymbol{d} = -\nabla F(\boldsymbol{x})$ norabidean F funtzioaren balioa txikiagotzen da, $\nabla F(\boldsymbol{x}) \neq \boldsymbol{0}$ izanik, eta honela aurkitzen da norabide horretan eman behar duen urratsaren luzera, alegia, α-rako aukera zehatzena ($\alpha > 0$ izanik):

$$\min_{\alpha} F(\boldsymbol{x} + \alpha\boldsymbol{d}), \tag{7.22}$$

baina, orokorki, kalkulu hori garestiegia da beste metodo ez-zehatz eta eraginkor batzuekin konparatzen badugu. Aurretiko metodo horrek F-ren minimo batera jotzen du konbergentzia linealarekin, eta, batzuetan oso astiro, lineala da.

Gradiente metodoan $\boldsymbol{A}_k = \mathbb{1}$ da.

Funtzio koadratiko baterako konbergentzia-ratioa

Izan bedi $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$ funtzio koadratikoa, non $\boldsymbol{Q}$ simetrikoa eta definitu positiboa baita. Gradientea $\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}$ da, eta $\boldsymbol{x}^*$ minimoa $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{b}$ sistema linealaren soluzio bakarra.

Demagun $\boldsymbol{x}^*$-ren kalkuluan beherapen azkarreneko metodoaz k-garren iterazioan gaudela, hots, $\boldsymbol{x}_k$ dugu. Izan bedi $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$, hortaz, hurrengo iterazioa kalkulatzeko, hau egin

behar dugu:

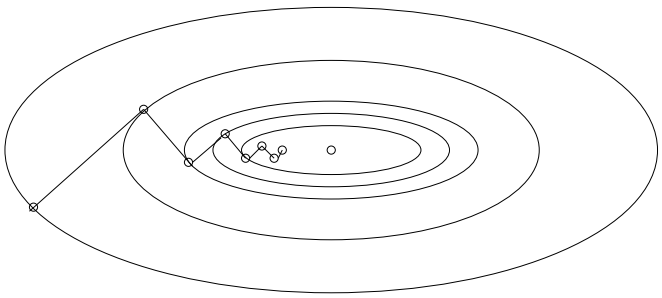
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \boldsymbol{g}_k$$

eta α kalkulatu behar da. Horretarako, bilaketa lineal zehatza (7.22) erabiliko dugu, gure kasuan, $f(\boldsymbol{x}_k - \alpha \boldsymbol{g}_k)$ funtzioa α-rekiko minimizatu behar dugu. Deribatuz eta zerora berdinduz, hau lortzen dugu (frogapena ariketa gisa geratzen da):

$$\alpha_k = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_k^T \boldsymbol{Q} \boldsymbol{g}_k}. \tag{7.23}$$

Ondorioz, funtzio koadratiko definitu positibo baterako beherapen azkarreneko iterazioa hau da:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_k^T \boldsymbol{Q} \boldsymbol{g}_k} \boldsymbol{g}_k. \tag{7.24}$$



7.1. irudia. Gradiente metodoaren urratsak.

Irudian bi dimentsioko funtzio koadratiko baterako metodo honek sortutako iterazio-segida tipiko bat ikus daiteke. f-ren sestra-kurbak elipsoideak dira, eta haien ardatzak $\boldsymbol{Q}$ matrizearen autobektore unitarioen norabideen gainean daude. Kontuan izan iterazioak soluziorantz sigi-sagan doazela.

Konbergentzia-ratioa kalkulatzeko, norma haztatu bat erabiliko dugu, $\boldsymbol{Q}$-norma hau: $\|\boldsymbol{x}\|_{\boldsymbol{Q}}^2 = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$. Orduan, $\boldsymbol{Q}\boldsymbol{x}^* = \boldsymbol{b}$ erlazioa erabiltzen badugu, hau erraz froga dezakegu:

$$\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = f(\boldsymbol{x}) - f(\boldsymbol{x}^*). \tag{7.25}$$

7.5. teorema: konbergentzia-ratioa. *$\boldsymbol{Q}$ matrizea simetrikoa eta definitu positiboa bada eta beherapen azkarreneko (7.24) iterazioa erabiltzen badugu, hau dugu:*

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = \left(1 - \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{(\boldsymbol{g}_k^T \boldsymbol{Q} \boldsymbol{g}_k)(\boldsymbol{g}_k^T \boldsymbol{Q}^{-1} \boldsymbol{g}_k)}\right) \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2. \tag{7.26}$$

Frogantza. (7.25) erabiliz, zera dugu:

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = 2(f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})).$$

Bestalde, lehenengo $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \boldsymbol{g}_k$ erabiliz, hau dugu:

$$\begin{aligned} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) &= (\tfrac{1}{2}\boldsymbol{x}_k^T \boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{x}_k) - (\tfrac{1}{2}\boldsymbol{x}_{k+1}^T \boldsymbol{Q}\boldsymbol{x}_{k+1} - \boldsymbol{b}^T\boldsymbol{x}_{k+1}) \\ &= \alpha \boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{x}_k - \tfrac{1}{2}\alpha^2 \boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k - \alpha \boldsymbol{b}^T \boldsymbol{g}_k = \alpha \boldsymbol{g}_k^T(\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{b}) - \tfrac{1}{2}\alpha^2 \boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k, \end{aligned}$$

eta gero, $\boldsymbol{g}_k = \boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{b}$ adierazpena eta (7.23) bilaketa lineal zehatza kontuan hartuz, hau lortzen da:

$$\begin{aligned} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) &= \alpha \boldsymbol{g}_k^T \boldsymbol{g}_k - \tfrac{1}{2}\alpha^2 \boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k \\ &= \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k}(\boldsymbol{g}_k^T \boldsymbol{g}_k) - \frac{1}{2}\frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{(\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k)^2}\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k = \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k} - \frac{1}{2}\frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{(\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k)} = \frac{1}{2}\frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k}. \end{aligned}$$

Ondorioz,

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k}.$$

Bestalde, $\boldsymbol{g}_k = \boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{b} = \boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{Q}\boldsymbol{x}^* = \boldsymbol{Q}(\boldsymbol{x}_k - \boldsymbol{x}^*)$ denez,

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = (\boldsymbol{x}_k - \boldsymbol{x}^*)^T \boldsymbol{Q}(\boldsymbol{x}_k - \boldsymbol{x}^*) = \boldsymbol{g}_k^T \boldsymbol{Q}^{-1}\boldsymbol{g}_k.$$

Azkenik, aurreko bi berdintzak erabiliz, zera lortzen dugu:

$$\begin{aligned} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 - \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k} &= \left(1 - \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{(\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 \\ &= \left(1 - \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)^2}{(\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{g}_k)(\boldsymbol{g}_k^T \boldsymbol{Q}^{-1}\boldsymbol{g}_k)}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2. \ \square \end{aligned}$$

Adierazpen horrek deskribatzen du f-ren beheratze zehatza iterazio bakoitzean. Erlazionatuko dugu $\boldsymbol{Q}$-ren baldintzazko zenbakiarekin. Horretarako, Kantarovich-en desberdintza behar dugu. Frogapena [18] testuliburuan aurki daiteke.

7.6. lema: Kantarovichen desberdintza. *Izan bedi $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ matrize simetriko eta definitu positibo bat. Orduan, $\forall \boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{y} \neq \boldsymbol{0}$, zera betetzen da:*

$$\frac{(\boldsymbol{y}^T\boldsymbol{y})^2}{(\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y})(\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y})} \geq \frac{4Mm}{(M+m)^2},$$

M eta m, hurrenez hurren, $\boldsymbol{Q}$ matrizearen autobalio handiena eta txikiena izanik.

Ondorioz, hori (7.26) adierazpenean aplikatuz, hau lortzen dugu:

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 \leq \left(1 - \frac{4Mm}{(M+m)^2}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2 = \left(\frac{M-m}{M+m}\right)^2 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{Q}}^2,$$

eta (7.25) erabiliz, azkenik, zera aurkitu dugu:

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^*) \leq \left(\frac{M-m}{M+m}\right)^2 (f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)). \tag{7.27}$$

$\boldsymbol{Q}$-ren zenbakizko baldintza $r = \kappa_2(\boldsymbol{Q}) = M/m$ dugunez, aurreko adierazpenaz hau lortzen da:

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^*) \leq \left(\frac{r-1}{r+1}\right)^2 (f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)). \tag{7.28}$$

Emaitza hori zabal diezaiokegu edozein $f \in \mathcal{C}^2$ helburu funtzio ez-lineali, baina kasu honetan $\boldsymbol{Q} = \nabla^2 f(\boldsymbol{x}^*)$. Aurreko desberdintzak erakusten du gradiente metodoaren konbergentziaratioa oso motela izan daitekeela, nahiz eta matrize hessiarraren baldintzazko zenbakia oso handia ez izan. Adibidez, $\kappa_2(\boldsymbol{Q}) = 800$ eta $f(\boldsymbol{x}^*) = 0$ badira, (7.28) desberdintzak iradokitzen du funtzioaren balioa 0.08 bider bakarrik gutxituko dela, bostehun iterazio egin ondoren gradientearen metodoa erabiltzen badugu.

Newtonen metodo zehaztugabeak

Iterazio batean matrize hessiarra definitu positiboa ez bada, Newtonen metodoak ez du lortuko norabide beherakor bat. Horrelako matrize hessiarra duten problema handietarako $\boldsymbol{d}_k$ norabide beherakorra aurkitzeko, Newtonen metodoaren gisako iterazioa erabiltzen dugu matrize hessiarra aldatuz jarraian azaltzen den moduan.

Lehenengo, Gill and Murray-ren matrize hessiarrari Choleskyren faktorizazio aldatuaren algoritmoa aplikatuz (ikus [11]) $\boldsymbol{A}_k = \nabla^2 F(\boldsymbol{x}_k) + \boldsymbol{D} = \boldsymbol{R}^T\boldsymbol{R}$ lortuko dugu, $\boldsymbol{D}$ matrize diagonala izanik eta bere gaiak ez-negatiboak.

Matrize hessiarra definitu positiboa bada, $\boldsymbol{D} = \boldsymbol{0}$ da, eta ez dugu aldatu behar.

Aldiz, $\boldsymbol{D} \neq \boldsymbol{0}$ bada, Gerschgorin-en teorema kontuan hartuz (ikus 5.7. teorema) μ positibo txikiena aukeratzen dugu $\boldsymbol{A}_k = \nabla^2 F(\boldsymbol{x}_k) + \mu\mathbb{1}$ definitu positiboa izateko (hots, autobalio txikiena zero baino handiagoa izateko). Horretarako, nahikoa da hau hartzea:

$$\overline{\mu} = \min_{1\leq i\leq n} \left\{ a_{ii} - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \right\},$$

eta $\mu = |\overline{\mu}| + \eta$ jarri, non $\eta = \sqrt{\varepsilon_M}$.

Azkenik, hau hartuko dugu:

$$\boldsymbol{A}_k = \nabla^2 F(\boldsymbol{x}_k) + \mu_k\mathbb{1}, \text{ non } \mu_k = \max\{\mu, \|\boldsymbol{D}\|_\infty\}.$$

Horrela kalkulatutako $\boldsymbol{A}_k$ matrizea benetako matrize hessiarretik simetriko definitu positibo hurbila da. Metodo hau Dennis & Schnabel-en [9] testuliburuan aurki daiteke.

Quasi-Newton metodoak

Metodo hauek deribatuen kalkuluaren kostua ekiditeko erabiltzen dira. Ekuazio ez-linealetako sistemak ebazteko Broydenen metodoa erabili dugu. Baina, metodo hori ez datorkigu ondo $\nabla^2 F(\boldsymbol{x})$ matrize hessiarra hurbiltzeko, zeren ez baitu gordetzen simetria (ikus 7.18):

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \boldsymbol{u}_k \boldsymbol{s}_k^T, \quad \text{non} \quad \boldsymbol{u}_k = \frac{\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k}{\boldsymbol{s}_k^T \boldsymbol{s}_k}$$

eta orokorrean $\boldsymbol{u}_k \neq \boldsymbol{s}_k$ gertatzen baita.

Simetria gorde nahi badugu eta ebakitzailearen ekuazioa ($\boldsymbol{A}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k$) betetzea, orduan hasierako $\boldsymbol{A}_0$ simetrikoa izan behar da, eta eguneratzeko formula honelakoa:

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \sigma \boldsymbol{v}_k \boldsymbol{v}_k^T, \quad \text{non} \quad \sigma = \pm 1,$$

$\boldsymbol{v}_k$ bektore baterako. Hots, oraingo matrizea aldatzen dugu bat heineko beste matrize simetriko bat batuz. $\boldsymbol{A}_{k+1}$ behartuz ebakitzailearen ekuazioa betetzera, adierazpen bakar hau lortuko dugu:

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \frac{\boldsymbol{q}_k \boldsymbol{q}_k^T}{\boldsymbol{s}_k^T \boldsymbol{q}_k}, \quad \text{non} \quad \boldsymbol{q}_k = \boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k.$$

Frogapena ariketa bezala geratzen da. Formula horri *symmetric rank-one (SR1)* izenarekin ezagutzen dugu.

Bestalde, nahiz eta $\boldsymbol{A}_k$ definitu positiboa izan $\boldsymbol{A}_{k+1}$ indefinitua izan daiteke. Baina, ebakitzailearen metodoaren bidez ez da simetria bakarrik gorde dezakeguna, $\boldsymbol{A}_k$ definitu positibo izatea ere finka dezakegu, eta, ondorioz, $\boldsymbol{d}_k$ norabide beherakorra izatea. Hori lortzen da *Broyden-Fletcher-Goldfarb-Shanno (BFGS) eguneratzea* erabiltzen badugu. Eguneratze horretan, matrize hessiarraren alderantzikoa ($\nabla^2 F(\boldsymbol{x}_k)^{-1}$) hurbiltzen da $\boldsymbol{H}_k$ matrizeen bidez, adierazpen honen bitartez eguneratuz:

$$\boldsymbol{H}_{k+1} = (\mathbb{1} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T)\boldsymbol{H}_k(\mathbb{1} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^T, \quad \text{non} \quad \rho_k = \frac{1}{\boldsymbol{y}_k^T \boldsymbol{s}_k}.$$

Orain, aurreko adierazpenean Sherman-Morrison-Woodbury formula aplikatuz, $\nabla^2 F(\boldsymbol{x}_k)$ matrize hessiarraren hurbilpena ($\boldsymbol{A}_k$) honela eguneratu dezakegu:

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k - \frac{\boldsymbol{A}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \boldsymbol{A}_k}{\boldsymbol{s}_k^T \boldsymbol{A}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k}.$$

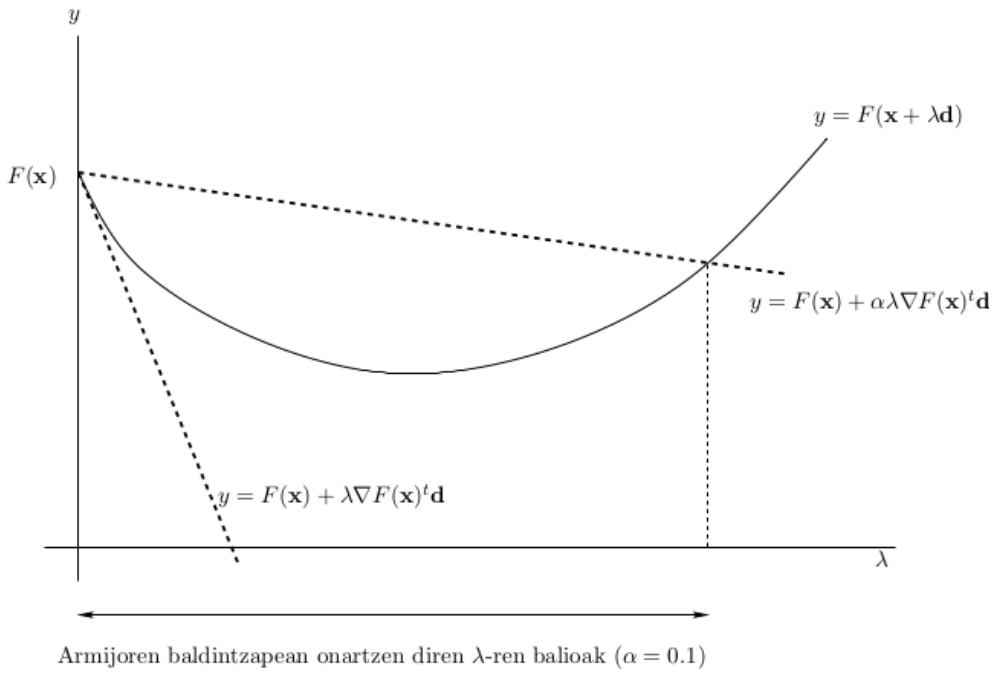
Azken bi formula horietan erraz ikusten denez, aldaketaren heina 2 da.

Konbergentzia globala: Armijo-Goldstein-en baldintzak (edo Wolfe-ren baldintzak)

Armijoren baldintza erabiltzen da funtzioaren balioaren beherapen nahiko bat izateko. Alegia, $\alpha \in (0,1)$ hartuz, eta $\lambda > 0$-ri hau betetzea exijituz:

$$F(\boldsymbol{x} + \lambda \boldsymbol{d}) \leq F(\boldsymbol{x}) + \alpha(\nabla F(\boldsymbol{x})^T \boldsymbol{d})\lambda. \tag{7.29}$$

Ikus 7.2. irudia.

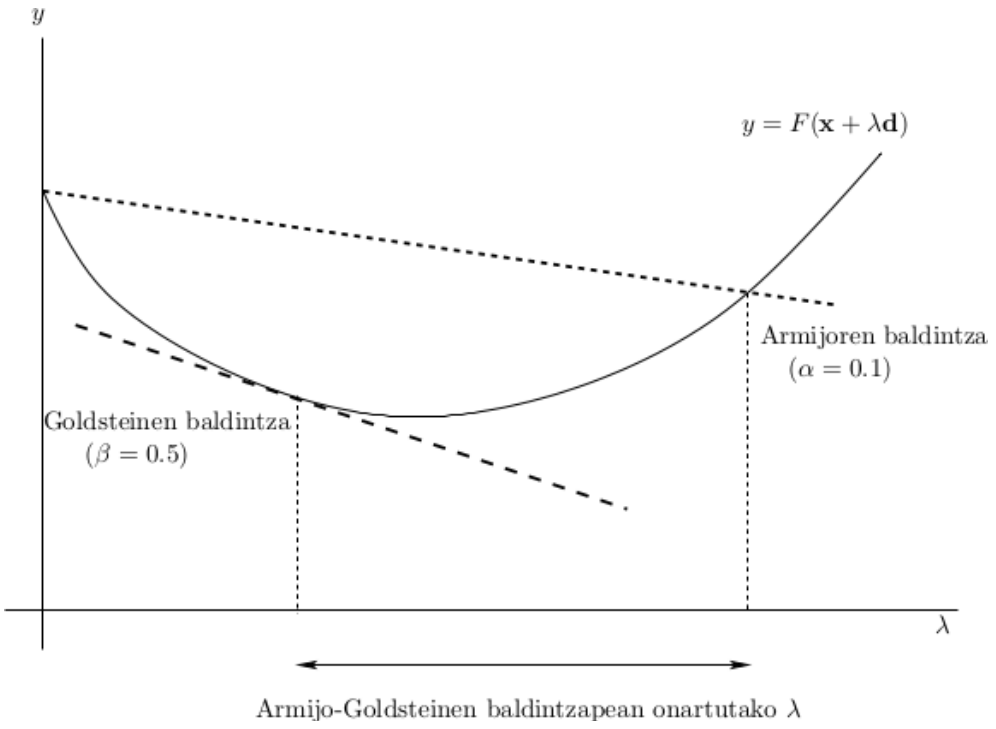


7.2. irudia. Armijoren baldintza.

Jakin badakigu λ nahiko txiki baterako $F(\boldsymbol{x} + \lambda\boldsymbol{d}) < F(\boldsymbol{x})$ betetzen dela, baina ez txikiegia; hori da, hain zuzen ere, Goldsteinen baldintzarekin saihesten dena. Baina, guk nahi dugu $\boldsymbol{d}$ norabidean $\boldsymbol{x} + \lambda\boldsymbol{d}$ puntuko F-ren beherapena aurreko $\boldsymbol{x}$ puntuan dugunaren ratio bat baino handiagoa izatea, hots:

$$\nabla F(\boldsymbol{x} + \lambda\boldsymbol{d})^T\boldsymbol{d} \geq \beta\nabla F(\boldsymbol{x})^T\boldsymbol{d}, \tag{7.30}$$

non $\beta \in (\alpha, 1)$ (ikus 7.3. irudia). Hori da Goldsteinen baldintza. Atzeranzko estrategia erabiltzen badugu, baldintza hori bide batez betetzen da.



7.3. irudia. Armijo-Goldsteinen baldintzak.

Honako teorema honek frogatzen du (ikus [29, 30]) $\boldsymbol{d}$ norabide beherakor bat izanez gero, $\boldsymbol{x} + \lambda \boldsymbol{d}$ puntuak egonik, (7.29) eta (7.30) Armijo-Goldsteinen baldintzak (edo Wolferen baldintzak) betetzen dituztela.

7.7. teorema. *Izan bedi* $F : \mathbb{R}^n \to \mathbb{R}$ *etengabe diferentziagarria (hots,* C^1 *klasekoa). Izan bitez* $\boldsymbol{x}_k, \boldsymbol{d}_k \in \mathbb{R}^n$*, non* $\nabla F(\boldsymbol{x}_k)^T \boldsymbol{d}_k < 0$ *baita (hots,* $\boldsymbol{d}_k$ norabide beherakorra *da), eta demagun* $\{F(\boldsymbol{x}_k + \lambda \boldsymbol{d}_k) \mid \lambda > 0\}$ *behe bornatua dela. Orduan, baldin* $0 \leq \alpha \leq \beta < 1$*, badaude* $\lambda_u > \lambda_l > 0$ *konstanteak, non* $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k$ *puntuak (7.29) eta (7.30) baldintzak betetzen baititu,* $\lambda_k \in (\lambda_l, \lambda_u)$ *bada.*

[9]-ko 6.3.2. teoreman aurki daiteke teorema honen frogantza eta Wolfe-n teoremaren enuntziatua (6.3.3. teorema) eta frogantza [29, 30]. Teorema horrek iterazio guztietan $\boldsymbol{d}_k$ beherakorra bada eta bilaketa linealaren bitartez kalkulatutako λ_k Armijo-Goldsteinen baldintzak betetzen baditu, gradientea zerora konbergitzen da, edo funtzioa ez da behe-bornatua eta ∞-ra jotzen du. Hau da, teorema horrek baldintza horietan konbergentzia globala bermatzen du. $\nabla F(\boldsymbol{x}_k)^T \boldsymbol{d}_k = 0$ ere izan daiteke, baina kasu hori saihestu dezakegu (ikus 123 orrialdea [9] liburuan).

Gainera, testuliburu berean Dennis and Moré-k erakusten dute k_0 batetik aurrera $\boldsymbol{d}_k$ norabide beherakorra Newtonen urratsetik nahiko hurbil badago (hots, $\boldsymbol{d}_k \approx -\nabla^2 F(\boldsymbol{x}_k)^{-1} \nabla F(\boldsymbol{x}_k)$),

orduan k_0-tik aurrera $\lambda_k = 1$ onargarria dela eta, $\nabla F(\boldsymbol{x}^*) = \mathbf{0}$ bada, konbergentzia superlineala izango da. Hori gertatzen da quasi-Newton metodoetan.

7.5. adibidea. *Izan bitez* $F(x_1, x_2) = x_1^4 + x_1^2 + x_2^2$, $\boldsymbol{x} = (x_1, x_2)^T = (1, 1)^T$, $\boldsymbol{d} = (-3, -1)^T$, *eta* $\alpha = 0.1$ *(7.29) adierazpenean eta* $\beta = 0.5$ *(7.30) -ean.*

Ebazpena. Hau dugunez:

$$\nabla F(\boldsymbol{x})^T \boldsymbol{d} = (6, 2)(-3, -1)^T = -20 < 0,$$

$\boldsymbol{x}$ horretan $\boldsymbol{d}$ norabide beherakorra da $F(\boldsymbol{x})$-rekiko. Orain, demagun $\boldsymbol{x}(\lambda) = \boldsymbol{x} + \lambda \boldsymbol{d}$ dela. Baldin $\lambda = 1$ bada, $\overline{\boldsymbol{x}} = \boldsymbol{x}(1) = \boldsymbol{x} + 1 \cdot \boldsymbol{d} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1 \cdot \begin{bmatrix} -3 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$ dugu, eta, orduan:

$$\nabla F(\overline{\boldsymbol{x}})^T \boldsymbol{d} = (-36, 0)(-3, -1)^T = 108 > -10 = \beta \nabla F(\boldsymbol{x})^T \boldsymbol{d}.$$

Beraz, $\overline{\boldsymbol{x}}$ (7.30) betetzen du, baina:

$$F(\overline{\boldsymbol{x}}) = 20 > 1 = F(\boldsymbol{x}) + \alpha \lambda \nabla F(\boldsymbol{x})^T \boldsymbol{d}$$

eta, ondorioz, ez du (7.29) betetzen. Halaber, $\lambda = 0.1$ bada, $\overline{\boldsymbol{x}} = x(0.1) = (0.7, 0.9)^T$ puntuak (7.29) betetzen du, baina (7.30) ez. Aldiz, $\lambda = 0.5$ bada, $\overline{\boldsymbol{x}} = x(0.5) = (-0.5, 0.5)^T$ bi baldintzak betetzen ditu, (7.29) eta (7.30) . Beraz, 7.3. irudian $\lambda = 0.1$ eskualde onargarriaren ezkerraldean dago, $\lambda = 1$ eskualde horren eskuinaldean, eta $\lambda = 0.5$ eskualde onargarrian bertan. □

Atzeranzko bilaketa lineala

Orain zehaztuko dugu nola aukeratu behar dugun λ_k. Gaur egungo estrategia da hasieran $\lambda_k = 1$ hartzea, eta $\boldsymbol{x}_k + \boldsymbol{d}_k$ ez bada onargarria, «atzera egitea» (hots, λ_k txikiagotzea) $\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k$ onargarri bat lortu arte. «Onargarriak» esan nahi du (7.29) eta (7.30) baldintzak betetzea. Bigarren baldintza betetzen da atzeranzko estrategia erabiltzeagatik, zeren horrek urrats txikiegiak saihesten baititu. Hori dela eta, ez da agertzen algoritmoan.

7.4. algoritmoa. Atzeranzko bilaketa lineala.

0. urratsa. Eman $\alpha \in (0, 1/2)$, $0 < l < u < 1$ eta $\lambda_k = 1$;

1. urratsa. $F(\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k) > F(\boldsymbol{x}_k) + \alpha \lambda_k \nabla F(\boldsymbol{x}_k)^T \boldsymbol{d}_k$ gertatzen den bitartean, egin $\lambda_k := \rho \lambda_k$, non $\rho \in [l, u]$;

2. urratsa. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k$.

Praktikan, α nahiko txikia denez, funtzioaren balioari beheratze oso txikia eskatzen baitzaio, gure algoritmoan $\alpha = 10^{-4}$. Bertsekas-ek $\rho = 1/2$ hartzea proposatzen du (ikus [3]). Beraz, horrela bada, λ_k aukeratzeko $\{1, 2^{-1}, 2^{-2}, 2^{-3}, \ldots\}$ segidari jarraituko diogu, eta helduko da 2^{-m} ($m \in \mathbb{N}$) balio batera, non hau betetzen baita (Armijoren baldintza):

$$F(\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k) \leq F(\boldsymbol{x}_k) + 10^{-4}\lambda_k \nabla F(\boldsymbol{x}_k)^T \boldsymbol{d}_k.$$

7.4.3. Minimo karratu ez-linealak

Honelako problemak dira:

$$\min F(\boldsymbol{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(\boldsymbol{x}),$$

non $r_j : \mathbb{R}^n \to \mathbb{R}$ funtzio leuna baita ($r_j \in C^1$) eta $m \geq n$. Gainera, r_j bakoitzari *hondar* deritzogu, eta $\boldsymbol{r}(\boldsymbol{x}) = (r_1(\boldsymbol{x}), r_2(\boldsymbol{x}), \ldots, r_m(\boldsymbol{x}))^T$ bektoreari *hondar-bektore*; $F(\boldsymbol{x})$ honela idatz dezakegu:

$$F(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2.$$

Bestalde, $F(\boldsymbol{x})$-ren deribatuak honela idatz daitezke $\boldsymbol{r}$ funtzioaren matrize jacobiarraren bitartez:

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial r_m}{\partial x_1} & \frac{\partial r_m}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla r_1(\boldsymbol{x})^T \\ \nabla r_2(\boldsymbol{x})^T \\ \vdots \\ \nabla r_m(\boldsymbol{x})^T \end{bmatrix}.$$

$$\nabla F(\boldsymbol{x}) = \sum_{j=1}^{m} r_j(\boldsymbol{x}) \nabla r_j(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}). \tag{7.31}$$

$$\begin{aligned} \nabla^2 F(\boldsymbol{x}) &= \sum_{j=1}^{m} \nabla r_j(\boldsymbol{x}) \nabla r_j(\boldsymbol{x})^T + \sum_{j=1}^{m} r_j(\boldsymbol{x}) \nabla^2 r_j(\boldsymbol{x}) \\ &= \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}) + \sum_{j=1}^{m} r_j(\boldsymbol{x}) \nabla^2 r_j(\boldsymbol{x}). \end{aligned} \tag{7.32}$$

Berdintza horiek frogatzea ariketa gisa geratzen da.

Gauss-Newtonen metodoa

F funtzioa deribagarria bada, bere minimoan $\nabla F(\boldsymbol{x}) = \boldsymbol{0}$ ekuazioa bete behar du. Gainera, baldintza hori $\boldsymbol{x}^*$ puntuan betetzen bada eta $\nabla^2 F(\boldsymbol{x}^*)$ definitu positiboa bada (hori gertatzen da $J(\boldsymbol{x}^*)$ hein betekoa bada), orduan $\boldsymbol{x}^*$ puntuan F funtzioak minimo bat heltzen

du. Beraz, $\nabla F(\boldsymbol{x}) = \mathbf{0}$ sistema ez-lineala ebazteko Newtonen metodoa erabiltzen badugu, iterazio bakoitzean honelako sistema bat ebatzi beharko dugu:

$$\nabla^2 F(\boldsymbol{x})\boldsymbol{d} = -\nabla F(\boldsymbol{x}). \tag{7.33}$$

Gauss-Newtonen metodoa Newtonen metodoaren aldaera bezala ikus daiteke. Kasu honetan, $\boldsymbol{d}_k$ bilaketa-norabidea kalkulatzeko (7.33) Newtonen sisteman, (7.32) deribatuaren bigarren batugaia (hots, $\sum_{j=1}^{m} r_j(\boldsymbol{x})\nabla^2 r_j(\boldsymbol{x})$) kendu eta gero sistema ebatziz lortzen da. Alegia, $\boldsymbol{d}_k^{GN}$ kalkulatzen da sistema hau ebatziz:

$$\boldsymbol{J}_k^T\boldsymbol{J}_k\boldsymbol{d} = -\boldsymbol{J}_k^T\boldsymbol{r}_k, \quad \text{non } \boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{x}_k) \text{ eta } \boldsymbol{r}_k = \boldsymbol{r}(\boldsymbol{x}_k). \tag{7.34}$$

Hauek dira metodo horren abantaila batzuk:

- $\nabla^2 F(\boldsymbol{x}_k) \approx \boldsymbol{J}_k^T\boldsymbol{J}_k$ hurbilpenak $\nabla^2 r_j(\boldsymbol{x})$ matrize hessiarren kalkulua saihesten du.
- Askotan, $\boldsymbol{J}^T\boldsymbol{J}$ gaia kendutako gaia baino askoz esanguratsuagoa da, bai r_j hondarrak txikiak direlako, bai ia linealak direlako; eta, orduan, $\|\nabla^2 r_j\|$ txikia da. Hori dela eta, oso sarri, metodo honek Newtonen metodoaren ia portaera berdina du, eta konbergentzia lokal azkarra (ia koadratikoa). Izan ere, $\boldsymbol{x}^*$ soluzioan hondar-bektorea nulua denean (hots, $\boldsymbol{r}(\boldsymbol{x}^*) = \mathbf{0}$), orduan, baldin $\boldsymbol{x}_0$ hasierako bektorea $\boldsymbol{x}^*$-tik nahiko hurbil badago, konbergentzia koadratikoa dela froga daiteke; ikus [9].
- $\boldsymbol{J}_k$ hein betekoa denean eta $\nabla F(\boldsymbol{x}_k) \neq \mathbf{0}$, $\boldsymbol{d}_k^{GN}$ norabidea norabide beherakorra da, zeren $\boldsymbol{J}_k^T\boldsymbol{J}_k$ definitu positiboa baita. Orduan, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda\boldsymbol{d}_k^{GN}$ kalkulatzeko bilaketa lineala erabil dezakegu λ egoki bat aurkituz (hots, Wolferen baldintzak egiaztatuz), non F-ren beherapena nahikoa baita.
- Bistan denez, $\boldsymbol{d}_k^{GN}$ aurkitzeko (7.34) ekuazio normalak ditugu, eta, dakigunez, baliokideak dira sistema hori ebaztea eta honako minimo karratu lineal hauen problema ebaztea:

  $$\min_{\boldsymbol{d}} \|\boldsymbol{J}_k\boldsymbol{d} + \boldsymbol{r}_k\|^2.$$

  Beraz, $\boldsymbol{d}_k^{GN}$ norabidea kalkula dezakegu ikusitako QR faktorizazioa erabiliz, eta horrela ez dugu izango $\boldsymbol{J}_k^T\boldsymbol{J}_k$ kalkulatzeko beharrik.

Metodo horrek desabantaila hauek ditu:

- Problemak nahiko ez-linealak direnean edo hondar nahiko handiak dituenean, geldiro doa konbergentzia lineal lokala.
- Problemak oso ez-linealak direnean edo hondar oso handiak dituenean, ez da lokalki konbergente.
- Ez dago ondo definituta $\boldsymbol{J}_k$ jacobiarra zutabe hein betekoa ez denean.

Levenberg-Marquardt-en metodoa

$\boldsymbol{J}_k$ hein betekoa ez denean $\boldsymbol{J}_k^T\boldsymbol{J}_k$ singularra izango, da eta, orduan, $\boldsymbol{d}_k$ beherakorra dela finkatzeko, $\boldsymbol{J}_k^T\boldsymbol{J}_k$ matrizea aldatuko dugu $\boldsymbol{D}$ matrize diagonal batez. Horretarako, Choleskyren faktorizazio aldatua erabiliko dugu, eta $\boldsymbol{J}_k^T\boldsymbol{J}_k+\boldsymbol{D}$ definitu positiboa izatea lortuko. Ondorioz, sistema bateragarri zehaztu honetaz lortutako $\boldsymbol{d}_k$ norabidea beherakorra izango da:

$$(\boldsymbol{J}_k^T\boldsymbol{J}_k + \boldsymbol{D})\boldsymbol{d} = -\boldsymbol{J}_k^T\boldsymbol{r}_k.$$

Gero, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda\boldsymbol{d}_k$ kalkulatzeko, bilaketa lineala erabil dezakegu λ egoki bat aurkituz, konbergentzia globala finkatzeko.

Beste aukera bat da μ egoki bat aurkitzea $\boldsymbol{J}_k^T\boldsymbol{J}_k + \mu\mathbb{1}$ definitu positiboa izateko eta $\boldsymbol{d}$ beherakorra; metodo horri Levenberg-Marquardten metodoa deritzogu. Orduan, norabide beherakorra honela kalkulatzen da:

$$(\boldsymbol{J}_k^T\boldsymbol{J}_k + \mu\mathbb{1})\boldsymbol{d} = -\boldsymbol{J}_k^T\boldsymbol{r}_k \tag{7.35}$$

eta sistema hau ebaztea eta minimo karratu linealen problema hau ebaztea baliokideak dira:

$$\min_{\boldsymbol{d}} \frac{1}{2}\left\| \begin{bmatrix} \boldsymbol{J}_k \\ \sqrt{\mu}\mathbb{1} \end{bmatrix} \boldsymbol{d} + \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2. \tag{7.36}$$

μ hori kalkulatzeko, *konfiantza eskualdeko metodo* bat erabil dezakegu, ikus [21].

7.5. Problemak

1. Ebatzi sistema hau Newtonen metodoaren 7.1. algoritmoa erabiliz, $\boldsymbol{x}_0 = (x_0, y_0, z_0)^T = (0.5, 0.5, 0.5)^T$ hartuz:

$$\begin{aligned} f_1(x,y,z) &= x^2 + y^2 + z^2 - 1 = 0 \\ f_2(x,y,z) &= 2x^2 + y^2 - 4z = 0 \\ f_3(x,y,z) &= 3x^2 - 4y + z^2 = 0. \end{aligned}$$

   Bigarren urratsa egiteko, LU metodoa erabili.

   Baldin $\boldsymbol{f}(x,y,z) = (f_1(x,y,z), f_2(x,y,z), f_3(x,y,z))^T$ bada, bukatu iteratzeko prozesua $\|\boldsymbol{f}(x,y,z)\|_\infty < 6 \cdot 10^{-5}$ denean.

2. Ebatzi sistema hau Newtonen algoritmoaz, $\boldsymbol{x}_0 = (x_0, y_0)^T = (1.5, 1.5)^T$ hartuz:

$$\begin{aligned} f_1(x,y) &= x^2 + y^2 - 2 = 0 \\ f_2(x,y) &= e^{x-1} + y^3 - 2 = 0. \end{aligned}$$

   Baldin $\boldsymbol{f}(x,y) = (f_1(x,y), f_2(x,y))^T$ bada, bukatu iteratzeko prozesua $\|\boldsymbol{f}(x,y)\|_\infty < 10^{-4}$ denean.

3. Izan bedi sistema hau:

$$\begin{aligned} f_1(x,y) &= x^2 - y - 0.2 = 0 \\ f_2(x,y) &= y^2 - x - 0.3 = 0. \end{aligned}$$

   Ebatzi sistema hori Newtonen metodoaz, $\boldsymbol{x}_0 = (x_0, y_0)^T = (1.2, 1.2)^T$ hartuz; egin bi iterazio bakarrik. Orain $\boldsymbol{x}_0 = (x_0, y_0)^T = (-0.2, -0.2)^T$ hartu, eta egin Newtonen metodoaren lehenengo bi iterazioak.

4. Ebatzi bigarren problema diferentzia finituzko Newtonen metodoaz.

5. Ebatzi bigarren problema Broydenen metodoaz.

6. Kalkulatu Broydenen metodoaren lehenengo hiru iterazioak sistema honetarako:

$$\begin{aligned} f_1(x,y) &= x_1 + x_2 - 3 &&= 0 \\ f_2(x,y) &= x_1^2 + x_2^2 - 9 &&= 0, \end{aligned}$$

   $\boldsymbol{x}_0 = (x_0, y_0)^T = (2, 7)^T$ eta $\boldsymbol{A}_0 = \boldsymbol{J}(\boldsymbol{x}_0)$ hartuz. Zein da $\|\boldsymbol{x}_3 - \boldsymbol{x}_2\|_\infty$ zehaztasuna? Eta errorea (hots, $\|(f_1(\boldsymbol{x}_3), f_2(\boldsymbol{x}_3))^T\|$)?

7. Izan bedi sistema hau:

$$\begin{aligned} f_1(x,y) &= x^2 + y^2 - 2 = 0 \\ f_2(x,y) &= xy - 1 = 0. \end{aligned}$$

   (a) Egiaztatu $(1,1)$ eta $(-1,-1)$ sistema horren soluzioak direla.

(b) Zelako arazoak ager daitezke Newtonen metodoa aplikatzen saiatzen bagara soluzio horiek kalkulatzeko?

8. Aurkitu sistema ez-lineal hauen soluzio bat, Newtonen metodoa erabiliz:

$$(a)\quad \begin{aligned} &x_1^2 - 10x_1 + x_2^2 + 8 = 0,\\ &x_1x_2^2 + x_1 - 10x_2 + 8 = 0; \end{aligned} \qquad \text{hartu} \quad \boldsymbol{x}_0 = (1.2, 1.2)^T.$$

$$(b)\quad \begin{aligned} &3x_1^2 - x_2^2 = 0,\\ &3x_1x_2^2 - x_1^3 - 1 = 0; \end{aligned} \qquad \text{hartu} \quad \boldsymbol{x}_0 = (0.5, 0.5)^T.$$

Ebatzi sistema horiek Newtonen metodoaz, eta bukatu $\|\boldsymbol{f}(\boldsymbol{x})\|_\infty < 10^{-3}$ denean. Sistema bakoitzean, zenbat iterazio erabili ditugu? Zein da zehaztasun erlatiboa? Zein da konbergentziaren ordena?

9. Ebatzi aurreko problemako (a) eta (b) sistemak Broydenen metodoaz, eta bukatu $\|\boldsymbol{f}(\boldsymbol{x})\|_\infty < 10^{-3}$ denean. Sistema bakoitzean, zenbat iterazio erabili ditugu? Zein da zehaztasun erlatiboa? Zein da konbergentziaren ordena?

10. (a) Froga ezazu $f(\boldsymbol{x}) = x_1^2 - x_2^4 + 1$ funtzioak koordenatu-jatorrian zela-puntu bat duela. Hots, koordenatu-jatorria puntu kritikoa dela, baina ez dela minimo bat ezta maximo bat ere.

(b) Zer gertatzen da Newtonen metodoa aplikatzen badugu zela-puntu hori kalkulatzeko?

11. Izan bedi optimizazioko problema hau:

$$\min\ 5x^2 + 5y^2 - xy - 11x + 11y + 11.$$

(a) Aurkitu minimo izateko lehenengo ordenako baldintza beharrezkoak betetzen dituen puntu bat (hots, puntu kritiko bat).

(b) Froga ezazu puntu hori minimo globala dela.

(c) Zein izango litzateke gradiente metodoaren konbergentzia-ratioa problema honi aplikatzen badiogu?

(d) $\boldsymbol{x}_0 = (0, 0)^T$ puntutik abiatuz, gradiente metodoko zenbat iterazio erabili behar ditugu gehienez, funtzio errorea 10^{-11} bider gutxitzeko?

12. Izan bedi helburu-funtzio hau: $f(\boldsymbol{x}) = 3x_1^2 + 2x_1x_2 + x_2^2, \quad \boldsymbol{x}_0 = (1, 1)^T$.

(a) Zein da f-rako beherapen azkarreneko norabidea $\boldsymbol{x}_0$-tik?

(b) $(1, -1)^T$ norabide beherakorra da?

(c) Lehenengo ataleko norabide beherakorrerako, eta atzeranzko bilaketa lineala erabiliz, aurkitu λ onargarri bat $\lambda_0 = 1$, $\rho = 0.5$ eta $\alpha = 0.1$ hartuz.

13. Izan bedi optimizazioko problema hau: $\min f(x, y) = 3x^2 + y^4$.

(a) Gradiente-metodoaren iterazio bat aplikatu $\boldsymbol{x}_0 = (1, -2)^T$ hartuz, eta, λ aurkitzeko, atzeranzko bilaketa lineala erabili $\lambda = 1$, $\rho = 0.5$ eta $\alpha = 0.1$ hartuz.

(b) Errepikatu (a)-ren kalkulua, baina orain $\lambda = 1$, $\rho = 0.1$ eta $\alpha = 0.1$ hartuz. Konparatu kalkuluetan kostatu dena aurreko atalean kostatu denarekin.

(c) Orain, egin Newtonen metodoaren iterazio bat (a) ataleko baldintza berdinekin λ onargarria kalkulatzeko. Zer gertatu da? Konparatu kalkuluetan kostatu dena aurreko ataletan kostatu denarekin.

14. Izan bedi helburu-funtzio hau: $f(x, y) = x^2 + y^2 + xy - 3x$.

(a) Aurkitu f-ren minimo lokal bat.

(b) Zergatik da (a)-ren soluzioa minimo global bat?

(c) Gradientearen metodoa aplikatzen badugu (a) ataleko minimoa aurkitzeko, zein izango litzateke helburu-funtzioaren konbergentzia-ratioa?

15. Aurkitu gradiente-metodoaren konbergentzia-ratioa aplikatzen diogunean problema honi:

$$\min f(x, y) = x^2 + 1.999xy + y^2.$$

16. Rosenbrock-en funtzioa $\phi(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ da, eta, erraz ikusten den bezala, $(1, 1)^T$-ean minimo bakarra heltzen du. Aplikatu gradientearen metodoa, $\boldsymbol{x}_0 = (0, 0)^T$-tik hasiz, iterazio bat bakarrik eginez, eta atzeranzko bilaketa erabiliz, $\lambda = 1$, $\rho = 0.5$ eta $\alpha = 0.1$ hartuz. Zein da gutxi gorabehera konbergentzia-ratioa? Zenbat iterazio behar ditu, gutxi gorabehera, funtzio-errorea 10^{-6} bider gutxitzeko?

17. Rosenbrock-en $\phi(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ funtzioaren minimo bakarra kalkulatzeko, erabili Newtonen metodoa eta BFGS metodoa bi iteraziotan zehar. Bi kasuetan, atzeranzko bilaketa erabiliz, $\lambda = 1$, $\rho = 0.5$ eta $\alpha = 0.1$ hartuz, eta $\boldsymbol{x}_0 = (0, 0)^T$. Zein metodorekin hurbildu da hobeto soluzioa?

18. Doitu $d = \phi(t, \boldsymbol{x}) = x_1 e^{-x_2 t}$ funtzioa $t_i = -1, 0, 1, 2$ puntuetan $d_i = 2.7, 1, 0.4, 0.1$ balioei. Horretarako, erabili Gauss-Newtonen bi iterazio $\boldsymbol{x} = (1, 1)^t$ puntutik. Bilaketa linealerako, erabili atzeranzko bilaketa, $\lambda = 1$, $\rho = 0.5$ eta $\alpha = 10^{-4}$ hartuz.

19. Froga ezazu 7.4.3. ataleko (7.35) eta (7.36) metodoak baliokideak direla $\boldsymbol{d}_k$ kalkulatzeko.

20. Izan bedi $\boldsymbol{r} : \mathbb{R} \to \mathbb{R}^3$, non $r_i = e^{t_i x} - y_i$, $i = 1, 2, 3$, eta $f(x) = \frac{1}{2}\boldsymbol{r}(x)^T\boldsymbol{r}(x)$, non datu hauek baititugu:

$$(t_1, y_1) = (1, 2) \quad (t_2, y_2) = (2, 4) \quad (t_3, y_3) = (3, \cdot).$$

Jo dezagun y_3-ren balioak eta x_0 hasierako puntuak balio hauek har ditzaketela:

(a) $y_3 = 8$ eta $x_0 = 1$.

(b) $y_3 = -1$ eta $x_0 = 0$.

Ebatzi problema hau (a) eta (b) kasuetan, Newtonen metodoaz eta Gauss-Newtonen metodoaz, urratsaren luzera aldatu barik (hots, λ erabili gabe). Metodo bakoitzeko, bukatu iterazioak $\|\nabla f(\boldsymbol{x}_k)\|_\infty < 10^{-5}$ denean. Zer gertatzen da kasu bakoitzean? Zergatik?

MATLABez programatzeko problemak:

21. Ekuazio ez-linealen sistemetarako, 7.1. Newtonen metodoaren algoritmoa dugu. Eraiki M-fitxategi bat goiko algoritmoa inplementatzeko. Fitxategi horren izena `snewton.m` izango da.

    (a) Gogora ezazue goiburu hau izan behar duela:
    `function [x,nf,err,ze,i]=snewton(x0,emax,zemax,imax)`,
    non `nf` funtzio-balioztatze kopurua baita.

    (b) $\boldsymbol{f}(\boldsymbol{x})$ bektorea kalkulatzeko, funtzio bat definitu behar da M-fitxategi baten bidez, `fun.m` izenekoa (ez ahaztu, bektoreak zutabeak direla idazkera matrizialean).

    (c) Era antzeko batean kalkulatzen da $\boldsymbol{f}(\boldsymbol{x})$-ren deribatua; hots, $\boldsymbol{J}(\boldsymbol{x})$ matrizea kalkulatzeko, M-fitxategi bat eraikitzen da, `Jfun.m` izenekoa.

    (d) Lehenengo urratsean $\boldsymbol{x}_k$ punturako $\boldsymbol{f}$ funtzioaren eta $\boldsymbol{J}$ deribatuaren balioak kalkulatzen dira iterazio horretan.

    (e) Algoritmo horren 2. urratsean, (7.3) Newtonen sistema ebatzi behar dugu. Urrats horretan, erabili *LU* deskonposizioa eta aurreranzko eta atzeranzko ebazpenak.

    (f) Sistema ebatzi eta gero, $\boldsymbol{x}_{k+1}$ berria izango dugu; hor (4. urratsean) erabiliko dugun norma infinitu-norma izango da. $\|\boldsymbol{p}_k\|_\infty/\|\boldsymbol{x}_{k+1}\|_\infty < ze$ edo $\|\boldsymbol{f}(\boldsymbol{x}_{k+1})\|_\infty < emax$ edo $i = imax$ denean, gelditu egiten da; bestela, 1. urratsera joango da.

    (g) Izan bedi $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x^2+y^2-2 \\ e^{x-1}+y^3-2 \end{bmatrix}$. Ebatzi $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ sistema Newtonen metodoaz, zuk eraikitako kodearen bitartez. Hartu $\boldsymbol{x}_0 = (2,2)^T$, $ze = 10^{-6}$, $emax = 10^{-6}$ eta $imax = 20$.

22. (a) Eraiki `J=dfjfun(x)` goiburuko M-fitxategi bat diferentzia finituzko metodoak erabiliz $\boldsymbol{J}(\boldsymbol{x})$ matrize jakobiarra hurbiltzeko; ikus 7.2.1. atala, non $h = \sqrt{\varepsilon_M}$ (MATLABen `eps`=ε_M da).

    (b) Aurreko kodea ondo ibiltzen dela egiaztatu eta gero, eraiki beste kode bat, non $\boldsymbol{J}(\boldsymbol{x})$ matrize jakobiar hurbildu hori baita. Haren goiburua hau izango da:
    `function [x,nf,err,ze,i]=dfnewton(x0,emax,zemax,imax)`,
    non `nf` funtzio-balioztatze kopurua baita.

    Funtzio horrek Newtonen metodoaren funtzioarena egingo du, baina berari matrize jacobiar hurbildua emanez, ez benetakoa. Berak jacobiarraren balio hurbildua lortuko du, `dfjfun.m` erabiliz.

(c) Ebatzi aurreko ariketaren azken atala, parametro berdinekin, `dfnewton` funtzioa erabiliz. Gero, kodean, aldatu h-ren balioa $h = 0.1, 0.01, 0.001, 0.0001$ desberdinetarako, eta, taula batez, konparatu bere eraginkortasuna: iterazio kopurua eta funtzio-balioztatze kopurua.

23. Inplementatu 7.2. algoritmoa: Quasi-Newton metodoa, matrize hessiarra Broydenen metodoaz eguneratuz. Haren goiburua hau izango da:

    `function [x,nf,err,ze,i]=qnewton(x0,emax,zemax,imax)`,

    non `nf` funtzio-balioztatze kopurua baita.

    Ebatzi 21. ariketako (g) atala, parametro berdinekin, `qnewton` funtzioa erabiliz.

24. Konparatu 21.(g) ataleko problemarako 21.-23. ariketetako hiru kodeen eraginkortasuna, taula bat sortuz eta zutabe bakoitzean metodo bakoitzerako funtzio-balioztatze kopurua eta iterazio kopurua idatziz. Zer ondorio atera dezakezu?

Bibliografia

[1] U.M. Ascher eta Ch. Greif, *A First Course in Numerical Methods.* SIAM, 2011.

[2] K.E. Atkinson, *An Introduction to Numerical Analysis.* John Wiley and Sons, 1989.

[3] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, New York, 1982.

[4] D.P. Bertsekas, *Nonlinear Programming. (Second edition)* Athena Scientific, Belmont, USA, 2003.

[5] R.P. Brent, *Algorithms for Minimization without Derivatives.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[6] C.G. Broyden, *A Class of Methods for Solving Nonlinear Simultaneous Equations.* Mathematics of Computation (American Mathematical Society) 19 (92): 577–593.

[7] S.C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists.* McGraw-Hill, AEB, 2008.

[8] T.J. Dekker, *Finding a zero by means of successive linear interpolation.* In B. Dejon; P. Henrici, Constructive Aspects of the Fundamental Theorem of Algebra, Wiley-Interscience, Londres, 1969.

[9] J.E. Dennis eta R.B. Schnable, *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, 1983. SIAMek berrinprimatuta, 1996.

[10] I.S. Duff, A.M. Erisman eta J.K. Reid, *Direct Methods for Sparse Matrices.* Oxford University Press, 1986.

[11] P.E. Gill, W. Murray eta M.H. Wright, *Practical Optimization.* Academic Press, Londres, 1981.

[12] P.E. Gill, W. Murray eta M.H. Wright, *Numerical Linear Algebra and Optimization, Volume 1.* Addison-Wesley, 1991.

[13] J.A. George eta J.W. Liu, *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

[14] A. Gilat eta V. Subramaniam, *Numerical Methods for Engineers and Scientists: An Introduction with Applications Using MATLAB*. John Wiley & Sons, AEB, 2008.

[15] G.H. Golub eta C.F. Van Loan, *Matrix Computations*. The John Hopkins University Press, Baltimore, 1996.

[16] D.J. Higham eta N.J. Higham, *MATLAB Guide*. SIAM, 2005.

[17] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

[18] D.G. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley, AEB, 1989.

[19] J.F. Mathews eta K.D. Fink, *Métodos Numéricos con MATLAB*. Pearson, Prentice Hall, Madrid, 2005.

[20] C.B. Moler, *Numerical Computing with MATLAB*. SIAM, Filadelfia, AEB, 2004.

[21] J. Nocedal eta S.J. Wright, *Numerical Optimization*. Springer Series in Operations Research, New York, 1999.

[22] J.M. Ortega eta W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.

[23] M. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, Filadelfia, 2001.

[24] J.M. Sanz-Serna, *Diez lecciones de Cálculo numérico*. Secretariado de Publicaciones e Intercambio Editorial, Universidad de Valladolid, 2010.

[25] J. Stoer eta R. Bulirsch, *Introduction to Numerical Analysis*. Springer-Verlag, 2002.

[26] G.W. Stewart, *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[27] G.W. Stewart, *Matrix Algorithms: Basic Decompositions*. SIAM, Filadelfia, 1998.

[28] L.N. Trefethen eta D. Bau, *Numerical Linear Algebra*. SIAM, Filadelfia, 1997.

[29] P. Wolfe, *Convergence conditions for ascent methods*. SIAM Review 11, 226–235.

[30] P. Wolfe, *Convergence conditions for ascent methods. II: Some corrections*. SIAM Review 13, 185–188.

Kontzeptuen Aurkibidea

UNIBERTSITATEKO ESKULIBURUAK
MANUALES UNIVERSITARIOS

INFORMAZIOA ETA ESKARIAK • INFORMACIÓN Y PEDIDOS

UPV/EHUko Argitalpen Zerbitzua • Servicio Editorial de la UPV/EHU
argitaletxea@ehu.eus • editorial@ehu.eus
1397 Posta Kutxatila - 48080 Bilbo • Apartado 1397 - 48080 Bilbao
Tfn.: 94 601 2227 • www.ehu.eus/argitalpenak

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea