Kateatutako Runge-Kutta metodoek eskaintzen duten koste baxuko errorea

Elisabete Alberdi Celaya

Matematika Aplikatua Saila. Meatzeen eta Herri Lanen
Ingeniaritza Teknikoko Unibertsitate Eskola.
Euskal Herriko Unibertsitatea (UPV/EHU)

elisabete.alberdi@ehu.es

Jasoa: 2012-12-29 Onartua: 2013-06-30

Laburpena: Hasierako balioko Ekuazio Diferentzial Arrunten sistema bat zenbakizko metodoz askatzerakoan, koste konputazional baxuko eta zehaztasun handiko zenbakizko metodo bat erabiltzeak bezainbesteko garrantzia dauka urratsa onargarritzat emango digun errorearen neurri on eta konputazionalki merkea aukeratzeak. Orokorrean, urrats bakoitzean egiten ari garen errore globala kalkulatzea garestia da eta algoritmo gehienek errore lokalaren estimazio bat baino ez dute kontrolatzen. Algoritmo askok erabiltzen duten errore lokalaren estimazioa ondoz ondoko ordenako bi zenbakizko metodok emandako emaitzen arteko diferentzia da. Helburu modura hartzen da diferentzia hau definitutako tolerantzia bat baino txikiagoa izatea. Kateatutako Runge-Kutta metodoen abantaila da ondoz ondoko $(p, p+1)$ ordenako bi zenbakizko balio eskaintzea ia-ia zenbakizko balio bakarra lortzeko egin beharreko eragiketa kopuru berarekin. Eta bi balio hauen arteko diferentzia, errore lokala neurtzeko tresna baliagarria bezain merkea bilakatzen da.

Hitz gakoak: ekuazio diferentzial arruntak, kateatutako Runge-Kutta metodoak, errorearen estimazioa, MATLAB.

Abstract: When solving an initial value problem for a system of Ordinary Differential Equations by numerical methods, the choice of a computationally cheap measure of the error is as important as choosing a high-order and low-cost numerical method. The great problem presented by any technique for controlling global error is its huge computational cost, therefore most algorithms only use an estimation of the local error. Many algorithms use the difference between the values obtained by two numerical methods of consecutive order as approximation of the local error, so the algorithm tries to maintain this difference below a chosen tolerance. One of the advantages of the embedded Runge-Kutta methods is that a few extra operations provide the numerical values corresponding to methods of orders $(p, p+1)$, and it is low computationally demanding in order to obtain an estimation of the local error.

Keywords: Ordinary Differential Equations, embedded Runge-Kutta methods, error estimation, MATLAB.

1. SARRERA

Izan bedi hasierako balioko Ekuazio Diferentzial Arrunten (EDA) sistema bat:

$$y' = f(t,y(t)), \quad y(a) = y_0 \qquad (1),$$

eta

$y:[a,b] \to \mathbb{R}^m$ eta $f:[a,b]\times\mathbb{R}^m \to \mathbb{R}^m$ funtzioak jarraituak izanik, demagun $T = [a,b]$ tartean bere emaitza kalkulatu nahi dela.

EDA sistema honen soluzio analitikoa, era analitikoan emandako formula bat da. Emaitza hau zehatza da eta normalean, funtzio elementalak erabiliz ematen da (polinomioak, funtzio esponentzialak, logaritmikoak, trigonometrikoak, e.a); batzuetan batura finitu bezala ematen da eta beste batzuetan serie infinitu gisa. EDA baten soluzio analitikoa dugunean, posible da aldagai independenteari balioak emanez, soluzioak $t \in [a,b]$ edozein aldiunetan izango duen balioa kalkulatzea.

Hasierako balioko EDA sistema baten zenbakizko soluzioa aldiz, aukeratutako diskretizazioko $a = t_0 < t_1 < t_2 < \ldots < t_N = b$ puntuetako $y_n \approx y(t_n)$ balio hurbilduen taula da, non $y(t)$ hasierako balioko (1) EDAren soluzio analitikoa den. Balio hurbildu hauek ekuazio diferentziala eta zenbakizko metodo bat erabiliz urratsez urrats lortzen dira, eta menpeko aldagaiak aukeratutako aldagai independentearen balioetan hartzen dituen irudien multzoa da.

Integrazioa, $y(a) = y_0$ balioarekin hasten da eta $t_1 = t_0 + h_0$ puntuan soluzioaren hurbilpen bat kalkulatzen da: $y_1 \approx y(t_1)$. Prozesu hau aurrera eramaten da $y_N \approx y(t_N) = y(b)$ baliora iritsi arte; une bakoitzean $h_i = t_{i+1} - t_i$ tamainako urratsa aurreratzen da. Urrats-luzera konstantea izan daiteke edo gerta daiteke aldiune bakoitzean luzera egokitzea. Guk, testu honetan barrena urrats-luzera konstantea erabiliko dugu, $h_i = h$, hain zuzen.

Zenbakizko metodoaz lortutako emaitza ez da zehatza, hau da, errorea eragingo du. Errore hau emaitza analitiko zehatzaren eta zenbakizko emaitzaren arteko kenketa da. Baina orokorrean, emaitza analitikorik ez dugulako askatzen dugu EDA zenbakizko metodoaren bidez. Garrantzitsua da beraz emaitza analitikorik ezean, zenbakizko metodoaz lortutako emaitza hurbilduari dagokion errorea kalkulatzeko teknika aurkitzea.

2. EKUAZIO DIFERENTZIAL ARRUNTAK ZENBAKIZKO METODOAREN BIDEZ ASKATZEAN EGITEN DEN ERROREA

Hasierako balioko Ekuazio Diferentzial Arrunten sistema bat zenbakizko metodoak erabilita askatzen denean bi errore hauei erreparatzen zaie: mozketa-errore globalari eta mozketa-errore lokalari.

Mozketa-errore globala ekuazio horren soluzioa den balio analitiko zehatzaren, eta metodoa erabiliz kalkulatzen den balio hurbilduaren arteko diferentzia da. Demagun zenbakizko metodo bat erabilita t_n aldiunetik abiatuz t_{n+1} aldiuneko y_{n+1} balioa lortu dugula. Orduan, mozketa-errore globala, *GTE* nomenklatura erabiliz izendatzen dena (*Global Truncation Error* ingelesez), honela emana dago:

$$GTE_{n+1} = y(t_{n+1}) - y_{n+1} \qquad (2),$$

$y(t_{n+1})$ balioa, ekuazio diferentzialaren soluzio zehatzak t_{n+1} aldiunean hartzen duen balioa izanik eta y_{n+1} berriz, zenbakizko metodoa erabilita lortu den aldiune bereko balio hurbildua izanik.

Zenbakizko metodo baten mozketa-errore lokala «*LTE*» (*Local Truncation Error*) kalkulatzeko, aurreko urratsetako balioak zehatzak direnean zenbakizko metodoaren bidez lortutako emaitza y^*_{n+1} hartuko dugu. Balio honen diferentzia soluzio zehatzarekin *LTE* errorea da eta ondorengo formularen bidez emana dator:

$$LTE_{n+1} = y(t_{n+1}) - y^*_{n+1} \qquad (3).$$

Aipatutako bi erroreek metodoaren zehaztasuna neurtzen dute. Honela, metodo bat p ordenakoa da, mozketa-errore globala $O(h^p)$ denean edo gauza bera dena, mozketa-errore lokala $O(h^{p+1})$ denean.

Bi errore hauek daukaten arazoa agerian dago: EDAren emaitza zehatza normalean ez da ezaguna, eta horrek definitu ditugun bi erroreen kalkulua eragozten du.

Hau kontuan izanik, zein neurri erabiltzen ote da orduan zenbakizko metodo batek emandako emaitza onargarritzat jotzeko? Normalean, mozketa-errore lokalaren estimazio bat erabiltzen da eta honek hartzen duen balioaren arabera esaten da zenbakizko metodoaz lortutako emaitza ona den ala urrats tamaina txikiagoa erabilita errepikatu beharrik dagoen [1, 2].

Askotan erabiltzen den teknika bat, Richardsonen estrapolazioa da, [2], non zenbakizko metodo batek h eta $2h$ urrats tamainak erabilita ematen dizkigun balioak lortzen diren, y^h_{n+1} eta y^{2h}_{n+1} hurrenez hurren. Erabilitako zenbakizko metodoa p ordenakoa baldin bada, orduan Richardsonek proposatzen duen errorearen estimazioa honela kalkulatzen da:

$$est_{Richardson} = \frac{y^{2h}_{n+1} - y^h_{n+1}}{2^p - 1} \qquad (4).$$

Estimazio honen kostua da desabantailarik handiena, zenbakizko metodoa erabilita t_{n+1} aldiuneko bi balio kalkulatzea eskatzen baitu: bata h urrats tamaina erabilita eta bestea $2h$ erabilita.

Ondoz ondoko ordenako bi zenbakizko metodok emandako emaitzen arteko diferentzia da sarri erabili ohi den errorearen beste estimazio bat, diferentzia hau aurrez definitutako tolerantzia bat baino txikiagoa izatea helburu duena [3]. Baina, aipatu dugun estimazio honen kostua ere altua da, urrats bakoitzean bi zenbakizko metodori dagozkien emaitzen kalkuluaren beharra baitu. Hurrengo atalean aurkeztuko ditugun Runge-Kutta metodo kateatuek era honetako errorearen estimazio merkea eskaintzen dute.

3. **RUNGE-KUTTA METODOAK**

Runge-Kutta metodoek urrats bakoitzean aurreratutako tartean zenbait puntu aukeratzen dituzte eta soluzioaren deribatua puntu horietan darabilte, aurreko urratsean soluzioaren hurbilketari batu behar zaion gehikuntza zehatzagoa estimatzeko. Horrelakoetan anitz etapa dituela esaten da eta ondorioz, Runge-Kutta metodoak etapa anitzeko metodo edo s-etapako metodo izenaz ezagutzen dira.

Hasierako balioko Ekuazio Diferentzial Arrunten (1) sistema emanik, s-etapako Runge-Kutta metodoak adierazpen hauxe dauka [4, 5, 6]:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \qquad (5),$$

non:

$$k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{s} a_{ij} k_j), \quad i = 1,2,...,s \qquad (6).$$

Runge-Kutta metodoetako a_{ij}, b_i, c_i konstanteak koefizienteen taula batean jasotzen dira eta taula honi, Butcher taula deitzen zaio.

1. taula. Runge-Kutta metodoko Butcher taula.

$$\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}$$

Butcher taula era matrizialean ere idatz daiteke. Horretarako s dimentsioko $\vec{b}$ eta $\vec{c}$ bektoreak eta $s \times s$ dimentsioko A matrizea definitzen dira ondoko eran:

$$\vec{b} = \left[b_1, b_2, \ldots, b_s\right]^T, \vec{c} = \left[c_1, c_2, \ldots, c_s\right]^T, A = \left[a_{ij}\right].$$

Eta era matrizialean emandako Butcher taula hauxe litzateke:

2. taula. Era matrizialean emandako Butcher taula.

$$\begin{array}{c|c} \vec{c} & A \\ \hline & \vec{b}^T \end{array}$$

$\vec{c}$ bektoreko osagai bakoitzak urratsaren barneko etapa adierazten du eta $\vec{b}^T$ bektorea pisu-bektore bat da. A matrizeak etapa bakoitzak aurreko etapekin (metodoa esplizitua denean) edo etapa guztiekin (metodo inplizituaren kasuan) duen menpekotasuna adierazten du. Runge-Kutta metodo bat esplizitua, inplizitua edo erdi-inplizitua izan daiteke A matrizeko elementuen arabera:

— Metodoa esplizitua da A matrizea hertsiki behe-triangeluarra denean, hau da, $a_{ij} = 0$ $j \geq i$ eta $i = 1, 2, \ldots, s$ balioetarako.
— Metodoa erdi-inplizitua da A matrizea behe-triangeluarra denean, hau da, $a_{ij} = 0$ $j > i$ eta $i = 1, 2, \ldots, s$ balioetarako.
— Metodoa inplizitua da A matrizea ez denean behe-triangeluarra, hau da, $a_{ij} \neq 0$ $j > i$ balioren batentzat betetzen denean.

Runge-Kutta metodoa esplizitua denean, (6) adierazpenak forma hau hartzen du:

$$k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{ij} k_j), \quad i = 1,2,\ldots,s \qquad (7).$$

4. RUNGE-KUTTA METODOEN ORDENA-BALDINTZAK

Orokorrean, hau da s-etapa dituen Runge-Kutta metodo baten mozketa-errore lokalaren adierazpena [4]:

$$LTE = \sum_{i=1} h^{i+1}\left(\sum_{r=i+1} \frac{1}{\sigma_{r,q}}\left(\frac{1}{\gamma_{r,q}} - \phi_{r,q}\right) D_{r,q}(y(t_n))\right) \qquad (8).$$

(8) adierazpenean h^{i+1} terminoak $y(t)$ funtzioaren $r = i + 1$ ordenako deribatuak ageri diren batukaria biderkatzen du. r ordenako deribatua, orokorrean, batugai bat baino gehiagoz osatua egongo da eta hau adierazteko q azpiindizea erabili da. $\sigma_{r,q}$ eta $\gamma_{r,q}$ konstanteak dira; $\phi_{r,q}$ terminoa ere konstantea da, eta a_{ij}, b_i eta c_i koefizienteen menpekoa; beraz, s etapa kopuruaren araberakoa da, eta $D_{r,q}$ terminoak oinarrizko diferentzialak dira [4], $y(t)$ funtzioaren r ordenako deribatuan agertzen diren funtzioak, alegia.

3. taulan 1 – 5 ordenei dagozkien $\sigma_{r,q}$, $\gamma_{r,q}$, $\phi_{r,q}$ eta $D_{r,q}$-ren balioak jaso dira. $D_{r,q}$ funtzioak adierazteko Butcherrek erabiltzen duen zuhaitz errotuen notazioa erabili da [4]:

— $r = 1$ denean $y(t)$ funtzioaren lehen ordenako deribatua $y' = f(t,y(t)) \equiv f$ da eta f-ren bidez adierazi da deribatu hau.
— $r = 2$ denean, $y(t)$ funtzioaren bigarren ordenako deribatua, $y'' = f_t + f_y f$ da. Deribatu hau $f'f$ notazioa erabilita adierazi da eta horrek esan nahi du $f_j f^{(j)}$ motako batugaiez osatua dagoela; j azpiindizeak aldagaiarekiko deribatua adierazten du (kasu honetan, t edo y aldagaiekiko deribatua) eta (j) goiindizeak $f^{(j)}$ funtzioa f_j-ri biderkatzen doala adierazten du. Deribatu partziala $j = t$ aldagaiarekikoa denean, $f^{(j)} = 1$ da eta deribatu partziala $j = y$ aldagaiarekikoa denean, $f^{(j)} = f$.
— $r = 3$ denean, $y(t)$ funtzioaren hirugarren ordenako deribatuan bi $D_{r,q}$ funtzio mota ageri dira. $D_{3,1} = f''(f,f)$ eta $D_{3,2} = f'f'f$. $D_{3,1}$ terminoan $f_{jk}f^{(j)}f^{(k)}$ motako batugaiak daude. Kasu honetan, j eta k azpiindizeek t edo y aldagaiekiko deribatuak adierazten dituzte, eta (j) eta (k) goiindizeko $f^{(j)}$ eta $f^{(k)}$ funtzioak deribatu partziala egin den aldagaiarekin lotuta daude. Berriz ere, deribatu partziala $m = t$ aldagaiarekikoa denean, $f^{(m)} = 1$ izango da eta deribatu partziala $m = y$ aldagaiarekikoa denean, $f^{(m)} = f$. Zehazki, $D_{3,1}$ funtzioa hauxe da: $D_{3,1} = f_{tt} + 2f_{ty}f + f_{yy}f^2$. $D_{3,2}$ terminoan $f_j f_k^{(j)} f^{(k)}$ motako batugaiak daude. Kasu honetan ere j eta k azpiindizeek aldagaiekiko deribatuak adierazten dituzte, eta (j) eta (k) goiindizeek $f_k^{(j)}$ funtzioa f_j funtzioari biderkatzen doala eta $f^{(k)}$ funtzioa $f_k^{(j)}$-ri biderkatzen doala adierazten dute. Aurretik esan bezala, $f^{(k)}$ funtzioa deribatu partziala egin den aldagaiaren araberakoa da. Zehazki, $D_{3,2}$ funtzioa hauxe da: $D_{3,2} = f_y f_t + f_y f_y f$.

(8) adierazpenak agerian uzten du, Runge-Kutta metodo baten mozketa-errore lokalaren kalkuluak koste handia duela, berau kalkulatzeko eragiketa asko egin behar direla, alegia.

3. taula. 1 – 5 ordenei dagozkien $\sigma_{r,q}$, $\gamma_{r,q}$, $\phi_{r,q}$ eta $D_{r,q}$-ren balioak.

r	q	$\sigma_{r,q}$	$\gamma_{r,q}$	$\phi_{r,q}$	$D_{r,q}$
1	1	1	1	$\sum_{i=1}^{s} b_i$	f
2	1	1	2	$\sum_{i=1}^{s} b_i c_i$	$f'f$
3	1	2	3	$\sum_{i=1}^{s} b_i c_i^2$	$f''(f,f)$
3	2	1	6	$\sum_{i,j=1}^{s} b_i \left(\sum_{j<i} a_{ij} c_j\right)$	$f'f'f$
4	1	6	4	$\sum_{i=1}^{s} b_i c_i^3$	$f'''(f,f,f)$
4	2	1	8	$\sum_{i,j=1}^{s} b_i c_i \left(\sum_{j<i} a_{ij} c_j\right)$	$f''(f,f'f)$
4	3	2	12	$\sum_{i,j=1}^{s} b_i \left(\sum_{j<i} a_{ij} c_j^2\right)$	$f'f''(f,f)$
4	4	1	24	$\sum_{i,j,k=1}^{s} b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k\right)\right)$	$f'f'f'f$
5	1	24	5	$\sum_{i=1}^{s} b_i c_i^4$	$f^{(4)}(f,f,f,f)$
5	2	2	10	$\sum_{i,j=1}^{s} b_i c_i^2 \left(\sum_{j<i} a_{ij} c_j\right)$	$f^{(3)}(f,f,f'f)$
5	3	2	15	$\sum_{i,j=1}^{s} b_i c_i \left(\sum_{j<i} a_{ij} c_j^2\right)$	$f''(f,f''(f,f))$
5	4	1	30	$\sum_{i,j,k=1}^{s} b_i c_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k\right)\right)$	$f''(f,f'f'f)$
5	5	2	20	$\sum_{i,j=1}^{s} b_i \left(\sum_{j=1}^{s} a_{ij} c_j\right)^2$	$f''(f'f,f'f)$
5	6	6	20	$\sum_{i,j=1}^{s} b_i \left(\sum_{j<i} a_{ij} c_j^3\right)$	$f'f'''(f,f,f)$
5	7	1	40	$\sum_{i,j,k=1}^{s} b_i \left(\sum_{j<i} a_{ij} c_j \left(\sum_{k<j} a_{jk} c_k\right)\right)$	$f'f''(f,f'f)$
5	8	2	60	$\sum_{i,j,k=1}^{s} b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k^2\right)\right)$	$f'f''f'''(f,f)$
5	9	1	120	$\sum_{i,j,k,m=1}^{s} b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} \left(\sum_{m<k} a_{km} c_m\right)\right)\right)$	$f'f'f'f'f$

Runge-Kutta metodo bat *p* ordenakoa da

$$\frac{1}{\sigma_{r,q}}\left(\frac{1}{\gamma_{r,q}}-\phi_{r,q}\right)=0, \qquad \forall q,\ r=1,2,...,p \tag{9}$$

baldintza betetzen denean, eta hau gertatzen denean, metodoaren mozketa-errore lokala (p + 1) ordenakoa da eta honela emana dago:

$$LTE = h^{p+1}\sum_{r=p+1}\frac{1}{\sigma_{r,q}}\left(\frac{1}{\gamma_{r,q}}-\phi_{r,q}\right)D_{r,q}(y(t_n))+O(h^{p+2}) \qquad (10).$$

$\sigma_{r,q}$, $\gamma_{r,q}$ eta $\phi_{r,q}$ konstanteen balioak (9) berdintzan ordezkatuz gero, Runge-Kutta metodoa p ordenakoa izan dadin a_{ij}, b_i eta c_i koefizienteek bete behar dituzten baldintzetara iritsiko gara. Hauxe da hain zuzen, metodoa bosgarren ordenakoa izateko bete behar dituen (11) adierazpeneko baldintzak lortzeko egin duguna.

$$\begin{array}{llll}
1.\phi_{1,1}=1 & 5.\phi_{4,1}=\frac{1}{4} & 9.\phi_{5,1}=\frac{1}{5} & 13.\phi_{5,5}=\frac{1}{20} \\
2.\phi_{2,1}=\frac{1}{2} & 6.\phi_{4,2}=\frac{1}{8} & 10.\phi_{5,2}=\frac{1}{10} & 14.\phi_{5,6}=\frac{1}{20} \\
3.\phi_{3,1}=\frac{1}{3} & 7.\phi_{4,3}=\frac{1}{12} & 11.\phi_{5,3}=\frac{1}{15} & 15.\phi_{5,7}=\frac{1}{40} \\
4.\phi_{3,2}=\frac{1}{6} & 8.\phi_{4,4}=\frac{1}{24} & 12.\phi_{5,4}=\frac{1}{30} & 16.\phi_{5,8}=\frac{1}{60} \\
 & & & 17.\phi_{5,9}=\frac{1}{120}
\end{array} \qquad (11).$$

4. taulan jaso dira *y* funtzioaren *r* ordenako deribatu bakoitzak bete behar duen baldintza kopurua eta bai guztira bete beharreko baldintzak Runge-Kutta metodo bat ordena jakin batekoa izateko. 4. taula era honetan irakurtzen da: metodoa 3. ordenakoa izateko *y* funtzioaren 3. ordenako deribatuak bi baldintza bete behar ditu (3 ordenaren azpian dagoen zenbakia da bi), 1. ordenako deribatuak baldintza bat eta 2. ordenako deribatuak ere baldintza bat. Beraz, guztira 4 baldintza bete behar dira me-

todoa 3. ordenakoa izan dadin. Kopuru osoari baldintza kopuru osoa esaten diogu taulan, deribatu bakoitzean bete beharreko baldintza kopurutik bereizteko.

4. taula. 1-10 ordenako Runge-Kutta metodoek bete beharreko baldintza kopurua [6].

Ordena	1	2	3	4	5	6	7	8	9	10
Deribatu bakoitzeko baldintza kopurua	1	1	2	4	9	20	48	115	286	719
Baldintza kopuru osoa	1	2	4	8	17	37	85	200	486	1205

5. **KATEATUTAKO RUNGE-KUTTA METODOAK**

Esan dugu Runge-Kutta metodoetako mozketa-errore lokaleko termino nagusia kalkulatzeak lan asko ematen duela. Hau dela eta, Mersonek (1957an) urratsean kalkulatutako k_i konstanteen menpe zegoen errorea estimatzeko era bat proposatu zuen. Mersonen ideia p eta $(p + 1)$ ordenako Runge-Kutta metodo bi erabiltzean oinarritzen zen metodo hauek c_i eta a_{ij} konstante berdinak zituzten, metodo bakoitzak bere $\vec{b}^T$ eta $\vec{\hat{b}}^T$ koefizienteak izanik. Era honetan sortutako Runge-Kutta metodoei kateatutako Runge-Kutta metodo deritze eta 5. taulan ikus daiteke hauei dagokien Butcher taula. Taula honetan $\vec{b}^T$ eta $\vec{\hat{b}}^T$ bektoreen diferentzia ere jasota dago: $\vec{E}^T = \vec{b}^T - \vec{\hat{b}}^T$.

5. taula. Kateatutako Runge-Kutta metodo baten Butcher taula.

$$
\begin{array}{c|c}
\vec{c} & A \\
\hline
 & \vec{b}^T \\
\hline
 & \vec{\hat{b}}^T \\
\hline
 & \vec{E}^T
\end{array}
$$

$\vec{c}$ eta $\vec{b}^T$ bektoreek eta A matrizeak definitzen duten Runge-Kutta metodoa p ordenakoa da, eta $\vec{c}$ eta $\vec{\hat{b}}^T$ bektoreek eta A matrizeak definitzen dutena, $(p + 1)$ ordenakoa. Kateatutako Runge-Kutta metodoetan k_i koefizienteak berdinak dira, bai p ordena duen metodoan, bai $(p + 1)$ ordenakoan. Izan ere, k_i koefizienteak c_i eta a_{ij} koefizienteen menpekotasuna daukate bakarrik eta hauek bi metodoetan berdinak dira. p eta $(p + 1)$ ordenako metodoen arteko ezberdintasun bakarra metodoko soluzioa diren $\hat{y}_{n+1}$, $(p + 1)$ ordenakoa, eta y_{n+1}, p ordenakoa kalkulatzeko egin beharreko azken batuketan dago, non $\vec{b}^T$ eta $\hat{\vec{b}}^T$ bektoreetako osagaiak erabiltzen diren:

— $(p + 1)$ ordenako metodoaz lortzen den emaitza:

$$\hat{y}_{n+1} = y_n + h\sum_{i=1}^{s}\hat{b}_i k_i \qquad (12).$$

— p ordenako metodoaz lortutakoa:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \qquad (13).$$

Mersonek mozketa-errore lokala $\hat{y}_{n+1}$ eta y_{n+1} balioen diferentzia bezala estimatzea proposatu zuen:

$$\hat{y}_{n+1} - y_{n+1} = h\sum_{i=1}^{s}(\hat{b}_i - b_i)k_i = h\sum_{i=1}^{s} E_i k_i \qquad (14),$$

non: $E_i = \hat{b}_i - b_i$.

Hurrengo urratseko hasierako balio gisa p ordenaren bidez kalkulatu dugun y_{n+1} balioa erabiltzen dugunean, kateatutako Runge-Kutta metodoa $(p,\ p + 1)$ bezala idazten da. Kasu honetan, p ordenaz kalkulatutako emaitzak kalkulatzen ditugu eta $(p + 1)$ ordenaren bidez kalkulatutakoak bakarrik errorearen estimazioan erabiltzen ditugu. Aldiz, $(p + 1)$ ordenako metodoak ematen digun emaitza, $\hat{y}_{n+1}$, hurrengo urratseko hasierako balio bezala darabilgunean eta p ordenak ematen digun emaitza bakarrik estimaziorako darabilgunean, kateatutako Runge-Kutta metodoa $(p + 1,\ p)$ bezala izendatzen da [7].

Asko dira kateatutako Runge-Kutta metodoak. Besteak beste, hauek:

— Fehlbergek hainbat Runge-Kutta metodo kateatu proposatu ditu [8, 9]. RKF45 bezala ezagutzen dena da Fehlbergen kateatutako Runge-Kutta (4,5) metodorik ezagunenetakoa. Metodo hau 6 etapakoa da ($s = 6$), 4. ordenako metodoa erabilita lortzen den emaitza ematen du eta 5. eta 4. ordenetako soluzioen diferentzia erabiltzen du errorearen estimazioa kalkulatzeko. Metodo honetako koefizienteak mozketa-errore lokal txikia izateko eran aukeratu zituen Fehlbergek. Metodo honen Butcher taula 6. irudian jaso da.

6. taula. RKF45 metodoaren Butcher taula.

c_i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}
0						
$\frac{1}{4}$	$\frac{1}{4}$					
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				
$\frac{12}{13}$	$\frac{1.932}{2.197}$	$-\frac{7.200}{2.197}$	$\frac{7.296}{2.197}$			
1	$\frac{439}{216}$	-8	$\frac{3.680}{513}$	$-\frac{845}{4.104}$		
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3.544}{2.565}$	$\frac{1.859}{4.104}$	$-\frac{11}{40}$	
$\vec{b}^T$	$\frac{25}{216}$	0	$\frac{1.408}{2.565}$	$\frac{2.197}{4.104}$	$-\frac{1}{5}$	0
$\hat{\vec{b}}^T$	$\frac{16}{135}$	0	$\frac{6.656}{12.825}$	$\frac{28.561}{56.430}$	$-\frac{9}{50}$	$\frac{2}{55}$
$\vec{E}^T$	$\frac{1}{360}$	0	$-\frac{128}{4.275}$	$-\frac{2.197}{75.240}$	$\frac{1}{50}$	$\frac{2}{55}$

— Kateatutako beste Runge-Kutta metodo ezagun bat Dormandek eta Princek sortu zutena da, DOPRI (5,4) [10]. Metodoa 5. ordenakoa da $\hat{\vec{b}}^T$ konstanteak darabiltzagunean eta $\vec{b}^T$ konstanteekin aldiz 4. ordenakoa dugu. Teorian 7 etapako metodoa bada ere (s = 7), praktikan 6 etapako metodoa da, $\hat{\vec{b}}^T$ bektoreko azken osagaia zero baita. Baldintza hau betetzen duten metodoei ingelesez FSAL (*First Same As Last*) esaten zaie, lehenengoa azkena bezalakoa esan nahi duena.

Horrek esan nahi du berdinak direla gauden urratseko deribatuaren azken ebaluaketa eta hurrengo urratseko lehen etapako deribatuarena.

7. taula. DOPRI(5,4) metodoaren Butcher taula.

c_i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19.372}{6.561}$	$-\frac{25.360}{2.187}$	$\frac{64.448}{6.561}$	$-\frac{212}{729}$			
1	$\frac{9.017}{3.168}$	$-\frac{355}{33}$	$\frac{46.732}{5.247}$	$\frac{49}{176}$	$-\frac{5.103}{18.656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1.113}$	$\frac{125}{192}$	$-\frac{2.187}{6.784}$	$\frac{11}{84}$	
$\vec{b}^T$	$\frac{5.179}{57.600}$	0	$\frac{7.571}{16.695}$	$\frac{393}{640}$	$-\frac{92.097}{339.200}$	$\frac{187}{2.100}$	$\frac{1}{40}$
$\hat{\vec{b}}^T$	$\frac{35}{384}$	0	$\frac{500}{1.113}$	$\frac{125}{192}$	$-\frac{2.187}{6.784}$	$\frac{11}{84}$	0
$\vec{E}^T$	$\frac{71}{57.600}$	0	$-\frac{71}{16.695}$	$\frac{71}{1.920}$	$-\frac{17.253}{339.200}$	$\frac{22}{525}$	$-\frac{1}{40}$

— Bogackiri eta Shampineri zor diegu kateatutako beste Runge-Kutta metodo ezagun bat [11]. Metodo honetan, $\hat{\vec{b}}^T$ konstanteak darabiltzagunean metodoa 3. ordenakoa da eta $\vec{b}^T$ konstanteekin aldiz 2. ordenakoa. Metodo hau ere 4 etapakoa izan arren, praktikan 3 etapakoa da, FSAL motakoa baita.

8. taula. Bogacki eta Shampinen (3,2) Runge-Kutta metodo kateatua.

c_i	a_{i1}	a_{i2}	a_{i3}	a_{i4}
0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{3}{4}$	0	$\frac{3}{4}$		
1	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	
$\vec{b}^T$	$\frac{7}{24}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{8}$
$\vec{\hat{b}}^T$	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	0
$\vec{E}^T$	$-\frac{5}{72}$	$\frac{1}{12}$	$\frac{1}{9}$	$-\frac{1}{8}$

6. MATLABEN EZARRITA DAUDEN RUNGE-KUTTA METODO KATEATUAK

MATLAB ® software-a kalkulurako eta programaziorako ingurune interaktiboa da. MATLABek tresnak eskaintzen ditu besteak beste, aljebra linealeko hainbat problema ebazteko, ekuazio ezlinealen erroak aurkitzeko, funtzioen integralak kalkulatzeko, polinomioak manipulatzeko, ekuazio diferentzial arruntak zein aljebraikoak askatzeko. Gainera, erraz heda eta pertsonaliza daiteke erabiltzaileak idatzitako funtzioak erabiliz.

MATLABek EDAk askatzeko dituen funtzioen artetik bi, ode23a eta ode45a hain zuzen, kateatutako Runge-Kutta metodoetan oinarritzen dira [12], lehenengoa Bogackik eta Shampinek proposatutako (3,2) Runge-Kutta metodo kateatuan eta bigarrena, aldiz, DOPRI(5,4) kateatutako Runge-Kutta metodoan. Funtzio biek ordenarik altueneko emaitza itzultzen dute, ode23ak 3. ordenakoa eta ode45ak 5. ordenakoa.

Kateatutako Runge-Kutta metodoen abantaila da, p eta $p + 1$ ordenei dagozkien k_i koefizienteak berdinak izatea direla eta bakarrik, (12) eta (13) adierazpenetako azken eragiketak izatea (batuketak eta biderketak) p eta $p + 1$ ordenako zenbakizko balioak lortzeko bereziki egin beharrekoak. Hau da, merkea suertatzen da oso eragiketa gehigarri gutxirekin (p, $p + 1$) ordenako zenbakizko balioak eskura izatean, balio hauen diferentzia errorearen estimazio gisa erabiltzea. Hauxe da hain zuzen, ode23 eta ode45 funtzioek erabiltzen duten errore lokalaren estimazioa, $p + 1$ eta p ordenako zenbakizko balioen diferentzia [13]:

$$est = \hat{y}_{n+1} - y_{n+1} = LTE + O\left(h^{p+2}\right) \tag{15},$$

ode23aren kasuan $p = 2$ izanik eta ode45aren kasuan $p = 4$ izanik. Eta estimazio hau, p ordenako formulari dagokion mozketa-errore lokala (*LTE*) baino ez da [13] eta (14) adierazpena erabilita kalkula daiteke.

Urrats bakoitzaren ostean, lortu den emaitza ona den edo errepikatu beharrik dagoen ikusten da. Urratsa ontzat ematen da errorearen estimazioa definitu dugun tolerantzia baino txikiagoa denean. Tolerantzia hau, orokorrean, emaitzaren osagai bakoitzerako definitzen da. Honela, i osagaiari dagokion tolerantzia ondokoa da:

$$tol^{(i)} = Atol^{(i)} + \left|y^{(i)}\right| Rtol, \quad i = 1,2,...,m \tag{16},$$

m EDA sistemaren dimentsioa izanik, $Rtol$ tolerantzia erlatiboa, $Atol^{(i)}$ tolerantzia absolutuaren i osagaia eta $y^{(i)}$ balioa $\hat{y}_{n+1}$ balioaren i osagaia. Tolerantzia absolutua da $Rtol = 0$ denean; erlatiboa da osagai guztietarako $Atol^{(i)} = 0$ betetzen denean eta bestela, bien arteko konbinazio bat da. Tolerantzia erlatiboa eskalare bat da eta absolutua bektore bat, EDAren emaitza den soluzioaren osagai bakoitzerako zehazten dena. Tolerantzia absolutuaren osagai bakoitza era honetan adieraz daiteke:

$$Atol^{(i)} = Rtol \cdot y_{min}^{(i)} \tag{17},$$

$y_{min}^{(i)}$, $\hat{y}_{n+1}$en osagai bakoitzerako balio minimo esanguratsu bat izanik. Era honetan, (16) adierazpena era honetan idatz daiteke:

$$tol^{(i)} = \left(y_{min}^{(i)} + \left|y^{(i)}\right|\right) Rtol, \quad i = 1,2,...,m \tag{18}.$$

Besterik esan ezean funtzio biek darabilten norma, osagaien diferentzia maximoan oinarritutakoa da, baina norma euklidearra ere erabil dezakete. Tolerantzia osagai bakoitzerako definitzeak, ahalbidetzen du errorearen estimazioaren norma era honetan idaztea:

— Norma bezala osagaien diferentzia maximoa darabilgun kasuan, hauxe dugu:

$$\|est\| = \|\hat{y}_{n+1} - y_{n+1}\| = max_{i=1,2,\ldots,m}\left(\frac{\left|\hat{y}^{(i)} - y^{(i)}\right|}{y^{(i)}_{min} + \left|y^{(i)}\right|}\right).$$

— Norma euklidearraren kasuan, hauxe:

$$\|est\| = \|\hat{y}_{n+1} - y_{n+1}\| = \sqrt{\sum_{i=1}^{m}\left(\frac{\left|\hat{y}^{(i)} - y^{(i)}\right|}{y^{(i)}_{min} + \left|y^{(i)}\right|}\right)^2} .$$

Kasu bietan, $\|est\| \leq Rtol$ baldintza bete behar da, emandako urratsa ona izan dadin. Bestela, eragiketa errepikatzen da aurrez erabilitako urrats-luzera baino laburragoa erabilita.

7. KOSTE KONPUTAZIONALAREN GARRANTZIA PROZESU OSOAN

Ekuazio Diferentzial Arrunten ebazpen hurbilduan, pentsatu beharra dago zenbakizko metodoan ez ezik urratsa ontzat emateko erabiliko dugun errorearen estimazioan ere. Zenbakizko metodoari emaitza ahalik eta onena koste ahalik eta baxuenarekin eskatzen badiogu, ez da bidezkoa aurreratutako denbora guztia galaraziko digun errorearen estimazio bat erabiltzea.

Orokorrean, errorearen estimazioak koste konputazional altua ondorioztatzen du. Aurkeztu dugun bezalaxe, kateatutako Runge-Kutta metodoek errorearen estimazio merkea eskaintzen digute, oso eragiketa gehigarri gutxirekin. Ondoz ondoko ordenako bi zenbakizko metodok ematen dizkiguten soluzio hurbilduen diferentzia errorearen estimaziotzat har daiteke, eta azken hau merke bezain ona suertatzen da.

8. BIBLIOGRAFIA

[1] SHAMPINE L. F. 2005. «Error estimation and control for ODEs». *Journal of Scientific Computing*, **15**, 3-16.

[2] SKEEL R. D. 1986. «Thirteen ways to estimate global error». *Numerische Mathematik*, **48**, 1-20.

[3] DORMAND J. R. 1996. *Numerical Methods for differential equations. A computational approach*. CRC Press, Boca Raton.

[4] BUTCHER J. C. 2008. *Numerical methods for ordinary differential equations*. John Wiley & Sons Ltd., Chichester.

[5] HAIRER E., NøRSETT S. P. eta WANNER G. 1993. *Solving ordinary differential equations, I, Nonstiff problems*. Springer, Berlin.

[6] LAMBERT J. D. 1991. *Numerical Methods for ordinary differential systems: the initial value problem*. John Wiley & Sons Ltd., Chichester.

[7] SHAMPINE L. F. 1977. «Local error control in codes for ordinary differential equations». *Applied Mathematics and Computation*, **3**, 189-210.

[8] FEHLBERG E. 1969. «Low order classical Runge-kutta formulas with stepsize control and their application to some heat transfer problems». *NASA TR R*, **315**.

[9] FEHLBERG E. 1968. «Classical fifth, sixth, seventh and eighth order Runge-Kutta formulas with stepsize control». *NASA TR R*, **287**.

[10] DORMAND J. R. eta PRINCE P. J. 1980. «A family of embedded Runge-Kutta formulae». *Journal of Computational and Applied Mathematics*, **6**, 19-26.

[11] BOGACKI P. eta SHAMPINE L.F. 1989. «A 3(2) Pair of Runge-Kutta Formulas». *Applied Mathematics Letters*, **2**, 1-9.

[12] SHAMPINE L. F. eta REICHELT M. W. 1997. «The MATLAB ODE suite». *SIAM Journal on Scientific Computing*, **18**, 1-22.

[13] SHAMPINE L. F., GLADWELL I. eta THOMPSON S. 2003. *Solving ODEs with Matlab*. Cambridge University Press, New York.