Third Order Explicit Method for the Stiff Ordinary Differential Equations

Medovikov*. Alexei. A.

Institute Numerical Mathematics.
117334 Russia. Leninski prospect 32-a, Moscow, e-mail: nucrect@inm.ras.ru

Abstract. The time-step in integration process has two restrictions. The first one is the time-step restriction due to accuracy requirement τ_{ac} and the second one is the time-step restriction due to stability requirement τ_{st}. The stability property of the Runge-Kutta method depend on stability region of the method. The stability function of the explicit methods is the polynomial. The stability regions of the polynomials are relatively small. The most of explicit methods have small stability regions and consequently small τ_{st}. It obliges us to solve the ODE with the small step size $\tau_{st} \ll \tau_{ac}$. The goal of our article is to construct the third order explicit methods with enlarged stability region (with the big τ_{st}: $\tau_{st} \geq \tau_{ac}$). To achieve this aim we construct the third order polynomials: $1 - z + z^2/2 - z^3/6 + \sum_{i=4}^{n} d_i z^i$ with the enlarge stability regions. Then we derive the formula for the embedded Runge-Kutta third order accuracy methods with the stability functions equal to above polynomials. The methods can use only three arrays of the storage. It gives us opportunity to solve large systems of differential equations.

1 Introduction.

Let us consider the system of ordinary differential equations:

$$\frac{du}{dt} = f(u,t), \qquad u|_{t=t_0} = u_0, \tag{1}$$

with sufficiently smooth function $f(u,t)$ and Runge-Kutta explicit method:

$$\begin{aligned} Y_i &= y_0 + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, Y_j), \\ y_1 &= y_0 + h \sum_{j=1}^{s} b_j f(t_0 + c_j h, Y_j). \end{aligned} \tag{2}$$

The table of the method (2) is the following:

* This article was written with the kind help of professors Lebedev V.I., Wanner G., Hairer E. and Russian Fund Fundamental Researches.

$$\begin{array}{c|ccccc} 0 & 0 & 0 & \dots & 0 \\ c_2 & a_{21} & 0 & \dots & 0 \\ c_3 & a_{31} & a_{32} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & a_{s,s-1} & 0 \\ \hline 1 & b_1 & b_2 & \dots & b_s \end{array} \quad , or \quad \begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^t \end{array}, \tag{3}$$

where $\mathbf{c}^t = (0, c_2, c_3, \dots, c_s)$, $\mathbf{b}^t = (b_1, \dots, b_s)$, $\mathbf{e}^t = (1, \dots, 1)$. Let us apply method (3) for the simple test problem:

$$u' = -\lambda u, \qquad u\Big|_{t=t_0} = u_0 = y_0. \tag{4}$$

The numerical result is expressed in term of stability function:

$$y_1 = R_s(\lambda h) y_0 = \Big(1 + \sum_{i=0}^{s-1} \mathbf{b}^t A^i \mathbf{e}(-\lambda h)^{i+1}\Big) y_0. \tag{5}$$

Stability function $R_s(z), z = \lambda h$ is the polynomial of degree s.

Definition1.1: *We will call a region* $U = \{z : |R_s(z)| \leq 1\}$ *stability region.*

Definition1.2: *We will call an interval* $I = U \bigcap [0, \infty[$ *real stability interval.*

We will construct a third order explicit methods with enlarged real stability interval. This means that order conditions must satisfy:

$$\begin{aligned} \textstyle\sum_i^s b_i &= 1, \\ \textstyle\sum_{i,j=1}^s b_i a_{ij} = \sum_i^s b_i c_i &= 1/2, \\ \textstyle\sum_{i,j,k=1}^s b_i a_{ij} a_{jk} = \mathbf{b}^t A^2 \mathbf{e} &= 1/6, \\ \textstyle\sum_{i,j,k=1}^s b_i a_{ij} a_{ik} = \sum_i^s b_i c_i^2 &= 1/3. \end{aligned} \tag{6}$$

From the order conditions (6) it follows that:

$$R_s(z) = 1 - z + z^2/2 - z^3/6 + \sum_{i=4}^{s} d_i z^i. \tag{7}$$

Absolute value of stability function $|R_s(z)|$ decreases in the small vicinity of the point $z = +0$. It means that stability region U and real stability interval are not empty set. The aim of our speculations is to construct function (7) with the maximum or 'nearly' real stability interval.

Theorem 1.3([2]): *If* $s > 3$ *and*

$$\max_{0 \leq t \leq M} |R_s(t)| \leq \eta = 1. \tag{8}$$

The polynomial $R_s(t)$ *has a maximal value* M *if there exist* $n + 1 - k$ *(k=3 for the third order methods) points* $t_i : 0 < t_1 < t_2 \dots < t_{n+1-k} \leq M$ *such that:*

$$R_s(t_1) = -\eta = -1, R_s(t_2) = +\eta = 1, \dots, R_s(z_{n+1-k}) = (-\eta)^{n+1-k}, \tag{9}$$

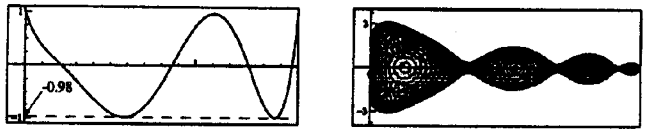

Fig. 1. Third order polynomial and it's stability region.

For practical calculations we will take η not equals to 1 but 'near' optimal value i.e. $\eta \approx 0,98 < 1$. It gives us more stability for many practical problems but it decreases real stability interval. The algorithm of construction of the optimal second and third order polynomials was described in the works of Kovalenko, Medovikov and Lebedev [2, 4]. Here we reproduce only roots of the some third order polynomials. These roots we used for the calculation of the parameters of the third order methods and used it in the program DUMKA3. In the **table 1.1** we represent the roots of the polynomials reduced to the interval [0,1]. To calculate the roots $\{\gamma_i\}_1^s$ of stability function (7) you have to multiply roots from the table and the parameter of stability region from the second column of the table.

TABLE 1.1 The roots of the third order stability polynomials.

Degree	Stability region	Roots
3	2.5005127005	(0.638297752962491,0.0000000000E+0), (0.280728100628313,0.722787568361731), (0.280728100628313,-0.722787568361731)
6	15.96769685542662	(1.316188704042163E-001,0.00000000E+0), (5.217998085157960E-2,-1.472133692919474E-1), (5.217998085157960E-2,1.472133692919474E-1), 5.397127885347366E-1,8.210181090527608E-1, 9.792807844727616E-1
9	38.31795251315424	(5.707036703430203E-2,0.000000000E+0), (2.307842599268251E-2,-6.407179746204085E-2), (2.307842599268251E-2,6.407179746204085E-2), 2.650900447972151E-1,4.564443606877882E-1, 6.434022749551114E-1,8.066819069334241E-1, 9.275476063065802E-1,9.917867224786107E-1
15	109.9635751502718	(2.027487087133956E-2,0.00000000E+0), (8.316021861212946E-3,-2.280465621150311E-2), (8.316021861212946E-3,2.280465621150311E-2), 1.002074585617464E-1,1.825798689818222E-1, 2.765670440070977E-1,3.791595999288834E-1, 4.861890912273565E-1,5.931003215440658E-1 6.952794913650315E-1,7.882921191244471E-1, 8.680911462040328E-1,9.311998388255081E-1, 9.748666481030083E-1,9.971869309844605E-1

Degree	Stability region	Roots
36	644.3020154572322	(3.488129601956453E-3,0.000000000E+0) (1.441852687344269E-3,-3.926697828110118E-3) (1.441852687344269E-3,3.926697828110118E-3) 1.778795197054982E-2,3.322288535643486E-2, 5.192760724673927E-2,7.393156860341134E-2, 9.912096771275233E-2,1.273224815410135E-1, 1.583307101392706E-1,1.919142061529804E-1, 2.278200310551857E-1,2.657765020564892E-1, 3.054957626118224E-1,3.466761985247533E-1, 3.890048660640773E-1,4.321599477783770E-1, 4.758132470553982E-1,5.196327141245989E-1, 5.632849914369575E-1,6.064379628604224E-1, 6.487632895565875E-1,6.899389145951866E-1, 7.296515180870848E-1,7.675989046864292E-1, 8.034923056404166E-1,8.370585780984603E-1, 8.680422850997063E-1,8.962076405169030E-1, 9.213403042299864E-1,9.432490139204415E-1, 9.617670411057193E-1,9.767534603597429E-1, 9.880942220801697E-1,9.957030206519658E-1, 9.995219514104168E-1
48	1145.804705468596	(1.963379226522905E-3,0.000000000E+0), (8.122094719300525E-4,-2.210430853325917E-3), (8.122094719300525E-4,2.210430853325917E-3), 1.006092366834490E-2,1.874663175494967E-2, 2.938902171273918E-2,4.199468711545039E-2, 5.653456570661632E-2,7.295652733345687E-2, 9.119500031334919E-2,1.111743678974924E-1, 1.328104842674921E-1,1.560115518572836E-1, 1.806787615356976E-1,2.067068432197892E-1, 2.339845866405663E-1,2.623953579846579E-1, 2.918176237070712E-1,3.221254861934987E-1, 3.531892327054925E-1,3.848758973520228E-1, 4.170498349018754E-1,4.495733047149258E-1, 4.823070627473591E-1,5.151109593853533E-1, 5.478445407355358E-1,5.803676509224324E-1, 6.125410328983705E-1,6.442269252512830E-1, 6.752896524954869E-1,7.055962063465749E-1, 7.350168155120516E-1,7.634255015728661E-1, 7.907006185865761E-1,8.167253741097289E-1, 8.413883294145599E-1,8.645838767626988E-1, 8.862126916957106E-1,9.061821584084480E-1, 9.244067663858447E-1,9.408084766063493E-1, 9.553170557451642E-1,9.678703769472033E-1, 9.784146858826100E-1,9.869048309461775E-1, 9.933044566154082E-1,9.975861591395987E-1, 9.997316038935454E-1

2 Third order Runge-Kutta method with enlarged stability region

Let us consider method (2) with $s = 3k$. Stability function can be represented in the form:

$$R_s(z) = \prod_{i=1}^{k}(1 - d_1^i z + d_2^i z^2 - d_3^i z^3) = \prod_{i=1}^{k} R_{s_i}^i(z). \tag{10}$$

We will consider composition method [1] where method (2) is comprises k submethods:

$$\begin{aligned}
v_0 &= y_0 \\
Y_2^i &= v_{i-1} + h a_{21}^i f(t_{i-1}, v_{i-1}) \\
Y_3^i &= v_{i-1} + h\left(a_{31}^i f(t_{i-1}, v_{i-1}) + a_{32}^i f(t_{i-1} + hc_2^i, Y_2^i)\right) \\
v_i &= v_{i-1} + h\left(b_1^i f(t_{i-1}, v_{i-1}) + b_2^i f(t_{i-1} + hc_2^i, Y_2^i) + b_3^i f(t_{i-1} + hc_3^i, Y_3^i)\right) \\
t_i &= t_{i-1} + h * (b_1^i + b_2^i + b_3^i) = t_{i-1} + h_i, \\
i &= 1, \ldots, k = s/3, \\
y_1 &= v_k.
\end{aligned}$$

Stability function of each submethods equals to:

$$R_{s_i}^i = 1 - d_1^i z + d_2^i z^2 - d_3^i z^3 \tag{11}$$

To satisfy order conditions (6) we have to set:

$$\begin{cases}
b_1^i + b_2^i + b_3^i = d_1^i \\
b_2^i c_2^i + b_3^i c_3^i = d_2^i \\
b_3^i a_{32}^i c_2^i = d_3^i \\
b_2^i (c_2^i)^2 + b_3^i (c_3^i)^2 = (d_3^i)^3/3 + ((d_1^i)^2 - 2d_2^i)t_{i-1} = B
\end{cases} \tag{12}$$

The fourth equality of (12) follows from the fourth equality of (6). We want every submethod to satisfy cubature formulae:

$$\int_{t_{i-1}}^{t_i} \tau^2 d\tau = b_1^i (t_{i-1})^2 + b_2^i (hc_2^i + t_{i-1})^2 + b_3^i (hc_3^i + t_{i-1})^2 = (t_i^3 - t_{i-1}^3)/3 \tag{13}$$

Another equalities of the system (12) follow from the stability function of the submethod (11). The system (12) has four equations and six variables. There are two free parameters and one can choose it to achieve some additional purpose. We consider only one case here. Let us take:

$$b_1^i = a_{31}^i, b_2^i = a_{32}^i.$$

In this case formulas for parameters of the methods are the following:

$$\begin{aligned} b_3^i &= \rho_1^i, \\ c_3^i &= \rho_2^i + \rho_3^i, \\ c_2^i &= \frac{B - b_3^i (c_3^i)^2}{\rho_2^i \rho_3^i}, \\ a_{32}^i &= \frac{\rho_2^i \rho_3^i d_3^i}{B - b_3^i (c_3^i)^2}, \\ b_2^i &= a_{32}^i, \\ b_1^i &= \rho_2^i + \rho_3^i - b_2^i, \\ a_{31}^i &= b_1^i, \\ i &= 1, \ldots, k = s/3, \end{aligned} \tag{14}$$

where ρ_1^i,ρ_2^i,ρ_3^i are the inverse values for the roots of the stability function $R_{s_i}^i$. Now to construct the method we take the roots of the third order polynomial, separate it on the groups : $1/\gamma_{j_{i_1}} = \rho_1^i, 1/\gamma_{j_{i_2}} = \rho_2^i, 1/\gamma_{j_{i_3}} = \rho_3^i$ (ρ_1^i is chosen as a real value), construct stability function R_3^i and solve systems (12) for $i = 1, \ldots, k = s/3$. This method require only three arrays to store and it is sufficient also for step-size control procedure. Consider the algorithm used in program DUMKA3. Let us h be a step-size of the method, $\{h_i\}_{i=1}^{k=s/3}$ -step sizes of each submethod $\sum_i^{k=s/3} h_i = h$. We remind that value h is chosen so that the spectrum of the problem lie inside the stability region: $h \leq M/\lambda_{max}$.

$$\begin{aligned} v_0 &= y_0 \\ Y_2 &= v_{i-1} + h a_{21}^i f(t_{i-1}, v_{i-1}) \\ Y_3 &= Y_2 + h\left((a_{31}^i - a_{21}^i) f(t_{i-1}, v_{i-1}) + a_{32}^i f(t_{i-1} + h c_2^i, Y_2)\right) \\ v_i &= Y_3 + h b_3^i f(t_{i-1} + h c_3^i, Y_3) \\ i &= 1, \ldots, k = s/3, \\ y_1 &= v_k. \end{aligned} \tag{15}$$

Let us check the order conditions (6). Submethod (15) is Runge-Kutta method with stability function $R_3^i = (1 - d_1^i z + d_2^i z^2 - d_3^i z^3)$ for every i. Stability function of the method is the product $R_s = \prod_{i=1}^{k=s/3} R_3^i$. We use polynomials with the roots from the table 1.1 so we have automatically that $R_s = 1 - z + z^2/2 - z^3/6 + \sum_{i=4}^{s} d_i z^i$. Hence the first three equality of the order conditions (6) are satisfied. The fourth one follows from (13):

$$\int_{t_0}^{t_s} \tau^2 d\tau = \sum_{i=1}^{k=s/3} \left(\int_{t_{i-1}}^{t_i} \tau^2 d\tau \right) = \sum_{i=1}^{k=s/3} \frac{t_i^3 - t_{i-1}^3}{3} = \frac{t_{k=s/3}^3 - t_0^3}{3}.$$

Consequently the last of (6) equalities of the order conditions is satisfied as well:

$$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}.$$

Finally we consider the time-step control procedure. We took this idea from the book by Hairer E., Wanner G.[1]. We use embedded formula to step-size control procedure. For this aim let us derive second order method with solution $\hat{y}_1$. With the help of the same speculation as in [1] we derive the formula:

$$\begin{aligned} err &= \|\hat{y}_1 - y_1\|, \\ h_{new} &= h\min(fac_{max}, \max(fac_{min}, fac(tol/err)^{1/(p+1)})), \end{aligned} \tag{16}$$

where $\|.\|$ is some norm in R^n, h- step-size of the previous step, fac_{min} and fac_{max} - some factors of step-size diminishing and increasing, tol is the tolerance which we require from our calculations and p -accuracy of the embedded method ($p = 2$ in our case). To calculate $\hat{y}_1$ consider the last step in a method (15). Solution $y_1 = v_k$ is the third order solution, so we can calculate second order solution $\hat{v}_{k-1}$ in the point $t_{k-1} = t_k - hc_4^k = t_0 + h$:

$$\hat{v}_{k-1} = v_k - h(c_4^k)\frac{f(t_k, v_k) + f(t_{k-1}, v_{k-1})}{2},$$

and recalculate solution with the second order $\hat{v}_k$ in the point t_k with the help of the formulas:

$$\hat{Y}_2 = \hat{v}_{k-1} + h(c_2^k)\frac{f(t_{k-1}, v_{k-1}) + f(t_{k-1} + hc_2^k, Y_2)}{2},$$

$$\hat{Y}_3 = \hat{Y}_2 + h(c_3^k - c_2^k)\frac{f(t_{k-1} + hc_2^k, Y_2) + f(t_{k-1} + hc_3^k, Y_3)}{2},$$

$$\hat{v}_k = \hat{Y}_3 + h(c_4^k - c_3^k)\frac{f(t_{k-1} + hc_3^k, Y_3) + f(t_k, v_k)}{2}.$$

We can derive:

$$\begin{aligned} \hat{y}_1 - y_1 = \hat{v}_k - v_k = -h(c_4^k)\tfrac{f(t_k,v_k)+f(t_{k-1},v_{k-1})}{2} &+ \\ h(c_2^k)\tfrac{f(t_{k-1},v_{k-1})+f(t_{k-1}+hc_2^k,Y_2)}{2} &+ \\ h(c_3^k - c_2^k)\tfrac{f(t_{k-1}+hc_2^k,Y_2)+f(t_{k-1}+hc_3^k,Y_3)}{2} &+ \\ h(c_4^k - c_3^k)\tfrac{f(t_{k-1}+hc_3^k,Y_3)+f(t_k,v_k)}{2} & \end{aligned} \tag{17}$$

We will use this difference to calculate next step size by formula (16). The algorithm (15) require 3 arrays to store, moreover the last step in (15) use only 2 arrays. This gives us opportunity to calculate the difference (17) without any help of additional array. We calculate the sum:

$$Z2 = h\frac{c_2^k - c_4^k}{2}f(t_{k-1}, v_{k-1}) + h\frac{c_3^k}{2}f(t_{k-1} + hc_2^k, Y_2),$$

after the second step in the formula(15), then calculate $f(t_{k-1} + hc_3^k, Y_3)$ and add to the sum:

$$Z2 = Z2 + h\frac{c_4^k - c_2^k}{2}f(t_{k-1} + hc_3^k, Y_3),$$

after calculation of the solution $y_1 = v_k$ we calculate $f(t_k, v_k)$ which we need for the next step and add this term to the sum:

$$Z2 = Z2 - h\frac{c_3^k}{2}f(t_k, v_k).$$

The sum $Z2$ we use as a difference $\hat{y}_1 - y_1$ and substitute it to the formula (16) calculate new step size. If $||Z2||$ exceeds tolerance: $||\hat{y}_1 - y_1|| > tol$ we can reject this step. We have written the program DUMKA3 to test this algorithm and next chapter is dedicated to numerical results.

3 Numerical results.

To test the program DUMKA3 we use the test problems from the book of Hairer E. and Wanner G.[1]. We take three programs RKC(Sommeijer[3]), DUMKA ([4]) and DUMKA3 and solve these problems. The results are represented in the figure 2 in axes: y- the time of calculations, x - accuracy. To use the program

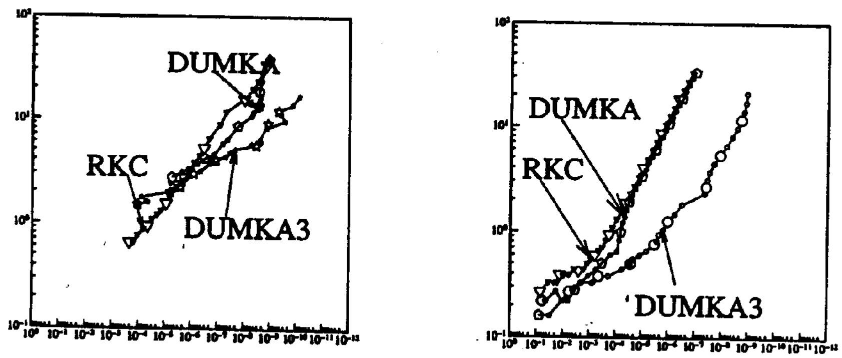

Fig. 2. The result of comparison of the programs for problems: 1-CUSP, 2-BURGERS

DUMKA3 one needs to write two SUBROUTINEs: subroutine of calculation of the right hand of equation and program for evaluation maximal eigen value λ_{max} and the value of $COU = 2/\lambda_{max}$. It is convenient to use and you don't need calculations of Jacoby matrix and linear algebra procedures.

References

1. Hairer E., Wanner G. (1991) Solving ordinary differential equations 2. Stiff problems. Springer-Verlag Berlin Heidelbers New York London Paris Tokyo.
2. Lebedev V.I., Medovikov A.A. (1995) The methods of the second order accuracy with variable time steps. Izvestiya Vuzov. Matematika. N10. Russia.
3. P.J. van der Houwen, B.P. Sommeijer (1980) On the internal stability of explicit m-stage Runge-Kutta methods for large m-values. Z.Angew. Math. Mech. 60.
4. Lebedev V.I. (1987) Explicit difference scheams with time-variable steps for solution of stiff system of equations. Preprint DNM AS USSR N177.