ΤΕΜΦΕ 5° Εξάμηνο
Αριθμητική Ανάλυση ΙΙ και Εργαστήριο
2° Εργαστήριο και Πρακτική Εξάσκηση

Μία άμεση μέθοδος Runge Kutta (RK) λύνει αριθμητικά το πρόβλημα αρχικών τιμών με βάση τον ακόλουθο τύπο:

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i k_i, \;\; k_i = f\left(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} k_j \right), \qquad \text{i=1,2,…,s}$$

Ο αριθμός s αποτελεί τον αριθμό σταδίων της μεθόδου και το βήμα της μεθόδου h_n μπορεί να είναι σταθερό ή να αλλάζει σε κάθε βήμα σύμφωνα με κάποιον μηχανισμό ελέγχου του σφάλματος της μεθόδου. Για λόγους οικονομίας οι συντελεστές της μεθόδου μπορούν να παρουσιαστούν με ένα butcher tableau:

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

όπου στην περίπτωση των άμεσων μεθόδων ο πίνακας A είναι αυστηρά κάτω τριγωνικός. Για μία τέτοια μέθοδο ορίζουμε το ολικό σφάλμα αποκοπής σε ένα σημείο της διαμέρισης της λύσης της μεθόδου ως την ποσότητα που αποτυγχάνει η μέθοδος ικανοποιήσει την πραγματική λύση του προβλήματος στο σημείο αυτό:

$$GE_{n+1} = \left\| y_{n+1} - y(x_{n+1}) \right\|.$$

Και το τοπικό σφάλμα αποκοπής σε ένα σημείο της διαμέρισης της λύσης της μεθόδου ως την ποσότητα που αποτυγχάνει ικανοποιήσει την πραγματική λύση του προβλήματος στο σημείο αυτό όταν θεωρηθεί ότι η προσέγγιση στο προηγούμενο βήμα της διαμέρισης ικανοποιεί τη λύση ακριβώς:

$$LTE_{n+1} = \left\| y_{n+1} - \tilde{y}_{n+1} \right\|$$

όπου:

$$\tilde{y}_{n+1} = y(x_n) + h_n \sum_{i=1}^{s} b_i k_i, \;\; k_i = f\left(x_n + c_i h_n, y(x_n) + h_n \sum_{j=1}^{i-1} a_{ij} k_j \right), \qquad \text{i=1,2,…,s}$$

Μία μέθοδος RK λέμε ότι έχει τάξη p όταν για το τοπικό σφάλμα αποκοπής ισχύει $LTE_{n+1} = O(h^{p+1})$ όπου όταν η μέθοδος εφαρμόζεται με μεταβλητό βήμα $h = \max_n \left(h_n \right)$. Αυτό σημαίνει ότι για μία μέθοδο τάξης p ισχύει $\left| LTE_{n+1} \right| \le K \cdot h^{p+1}$, όπου το K είναι σταθερά ανεξάρτητη του s. Όταν η μέθοδος είναι τάξης p τότε για το ολικό σφάλμα αποκοπής ισχύει $GE_{n+1} = O\left(h^p \right)$.

Α. Μέθοδος τεσσάρων σταδίων τετάρτης τάξης σταθερού βήματος (rk4.m)

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & \\ 1/2 & 0 & 1/2 & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

$$y_{n+1} = y_n + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$$

$$k_1 = f\left(x_n, y_n \right)$$

$$k_2 = f\left(x_n + \frac{1}{2}h, y_n + h\frac{1}{2}k_1 \right)$$

$$k_3 = f\left(x_n + \frac{1}{2}h, y_n + h(0k_1 + \frac{1}{2}k_2) \right)$$

$$k_4 = f\left(x_n + 1h, y_n + h(0k_1 + 0k_2 + 1k_3) \right)$$

```
function [tout, yout] = rk4(FunFcn,t0,tfinal,step,y0)
% rk4.m Costant Stepsize 4 step order 4
% Initialization
ceta = [1/2  1/2  1]';
alpha  = [ [ 1/2     0     0    0    ]
           [  0     1/2     0   0    ]
           [  0      0      1   0    ]]';
beta =  [1/6  1/3 1/3 1/6 ]';
stages=4;
t = t0; y = y0(:);f = y*zeros(1,stages);tout = t;yout = y.';
% The main loop
while abs(t- tfinal)> 1e-6
  if t + step > tfinal, step = tfinal - t; end
 % Compute the slopes
   temp = feval(FunFcn,t,y);
   f(:,1) = temp(:);
   for j = 1:stages-1
     temp = feval(FunFcn, t+ceta(j)*step, y+step*f*alpha(:,j));
     f(:,j+1) = temp(:);
   end
   t = t + step;
   y = y + step*f*beta(:,1);
   tout = [tout; t];
   yout = [yout; y.'];
end;
```

Για την εφαρμογή των μεθόδων RK με μεταβλητό βήμα χρειαζόμαστε ένα ζεύγος μεθόδων RK. Η μία έχει τάξη p και συνήθως η δεύτερη τάξη p-1. Οι μέθοδοι αυτοί έχουν τα ίδια a,c αλλά διαφορετικά c. Έτσι σε κάθε βήμα τα k_i υπολογίζονται μία φορά και με ελάχιστο υπολογιστικό κόστος έχουμε δύο προσεγγίσεις της λύσης σε κάθε σημείο της διαμέρισης.

$$y_{n+1}=y_n+h_n\sum_{i=1}^{s}b_i k_i$$

$$\hat{y}_{n+1}=y_n+h_n\sum_{i=1}^{s}\hat{b}_i k_i$$

$$k_1=f\left(x_n,y_n\right)$$

$$k_i=f\left(x_n+c_i h_n, y_n+h_n\sum_{j=1}^{i-1}a_{ij}k_j\right)$$

$$i=2,3,...,s.$$

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

Σε κάθε βήμα υπολογίζουμε μια προσέγγιση του σφάλματος

$$\delta_{n+1}=\left\|y_n-\hat{y}_n\right\|=\left\|h_n\sum_{j=1}^{s}(b_j-\hat{b}_j)k_j\right\|$$

Έχοντας θέσει μια ανοχή TOL, κάνουμε δεκτό το βήμα h_n όταν $\delta_{n+1}<=TOL$.

Τότε το επόμενο βήμα της μεθόδου δίνεται από τον τύπο:

$$h_{n+1}=0.8h_n\cdot\left\{\frac{TOL}{\left\|\delta_{n+1}\right\|}\right\}^{\frac{1}{p}},$$

Στην περίπτωση που $\delta_{n+1}>TOL$, το βήμα δεν γίνεται δεκτό και κάνουμε εκ νέου τους υπολογισμούς με ένα νέο μικρότερο βήμα που δίνεται επίσης από τον παραπάνω τύπο.

Β. Ζεύγος μεθόδων έξι σταδίων τάξεων 5(4) (rkf45.m) μεταβλητού βήματος.

Παρατηρήσεις στους κώδικες:

1. **Το ceta ορίζεται ως διάνυσμα στήλη χωρίς να περιέχει το πρώτο στοιχείο του c.**
2. **Το alpha είναι ο πίνακας που προκύπτει αν δεν λάβουμε υπόψη την πρώτη γραμμή του A και πάρουμε τον ανάστροφο.**
3. **Στον κώδικα σταθερού βήματος το beta είναι ο ανάστροφος του b, ενώ στον κώδικα μεταβλητού βήματος η πρώτη τους στήλη έχει ως στοιχεία τα στοιχεία του b και η δεύτερη τα στοιχεία της διαφοράς $b-\hat{b}$.**
4. **Οι τύποι της μεθόδου RK υλοποιούνται ως γινόμενα πινάκων.**
5. **Για να αλλάξετε μέθοδο στων κώδικα αλλάζετε τα alpha, beta, ceta, stages και pow (για τον κώδικα μεταβλητού βήματος).**

```
function [tout, yout] = frk45(FunFcn, t0, tfinal, y0, tol,trace)
% frk45.m Integrate a system of ordinary differential equations using
% 4th and 5th order Runge-Kutta formulas.  Variable stepsize
% The Fehlberg coefficients:
ceta = [1/4  3/8  12/13  1  1/2]';
alpha = [ [    1      0      0     0      0    0]/4
        [    3      9      0     0      0    0]/32
        [ 1932  -7200   7296     0      0    0]/2197
        [ 8341 -32832  29440  -845      0    0]/4104
        [-6080  41040 -28352  9295  -5643    0]/20520 ]';
beta = [ [902880  0  3953664  3855735  -1371249  277020]/7618050
        [ -2090  0    22528    21970    -15048  -27360]/752400 ]';
pow = 1/5; stages=6;
if nargin < 6, trace = 0; end
if nargin < 5, tol = 1.e-6; end
% Initialization
t = t0;hmax = (tfinal - t)/5;hmin = (tfinal - t)/20000;
h = (tfinal - t)/100; y = y0(:);f = y*zeros(1,stages);
tout = t; yout = y.'; tau = tol * max(norm(y, 'inf'), 1);
if trace,  clc, t, h, y, end
% The main loop
while abs(t- tfinal)> 1e-6  & (h >= hmin)
  if t + h > tfinal, h = tfinal - t; end
  % Compute the slopes
  temp = feval(FunFcn,t,y); f(:,1) = temp(:);
  for j = 1:stages-1
    temp = feval(FunFcn, t+ceta(j)*h, y+h*f*alpha(:,j));
    f(:,j+1) = temp(:);
  end
  % Estimate the error and the acceptable error
  delta = norm(h*f*beta(:,2),'inf');
  tau = tol*max(norm(y,'inf'),1.0);
  % Update the solution only if the error is acceptable
   if delta <= tau
      t = t + h;
      y = y + h*f*beta(:,1);
      tout = [tout; t];
      yout = [yout; y.'];
    end
    if trace, home, t, h, y, end
    % Update the step size
    if delta ~= 0.0
       h = min(hmax, 0.8*h*(tau/delta)^pow);
    end
end;
if (t < tfinal)
  disp('SINGULARITY LIKELY.');  t
end
```

ΤΕΜΦΕ 5^{ο} Εξάμηνο
Αριθμητική Ανάλυση ΙΙ και Εργαστήριο
2^{ο} Εργαστήριο και Πρακτική Εξάσκηση

Τα γνωστά προβλήματα:

Πρόβλημα 0: $\begin{array}{l} y'(t)+y+5e^{-t}\sin(5t)=0 \\ y(0)=1 \end{array}$ με λύση $y(t)=\cos(5t)\ e^{-t}$

Πρόβλημα 1: $\begin{array}{l} y'(t)=\begin{cases} y(t)(-2t+\frac{1}{t}) & t\neq 0 \\ 1 & t=0 \end{cases} \\ y(0)=0 \end{array}$ με λύση $y(t)=t\ e^{-t^2}$

Πρόβλημα 2: $\begin{array}{l} y''(t)+y(t)=0 \\ y(0)=1 \end{array}$ με λύση $y(t)=\cos(t)$ ισοδύναμα $\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix}=\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}$

Τα ερωτήματα:

1. Να γραφεί οδηγό πρόγραμμα (script) με όνομα run21p0.m το οποίο να λύνει το πρόβλημα 0 με τη μέθοδο Euler, με τη μέθοδο Improved Euler και με τη μέθοδο τετάρτης τάξης στο διάστημα [0,3] με βήμα h=0.1 και να εμφανίζει στο ίδιο γράφημα τη γραφική παράσταση και των τριών αριθμητικών λύσεων στα σημεία της διαμέρισης σε ένα γράφημα όπως και το γράφημα της πραγματικής λύσης για μία πυκνή (πλάτους 0.01) διαμέριση της λύσης. Από το γράφημα γίνεται φανερό ποια μέθοδος είναι καλύτερη όσο αφορά το σφάλμα. **Παρατήρηση:** Το ερώτημα είναι παρόμοιο με το ερώτημα 5 του Εργαστηρίου 1 (run03.m). Τώρα έχουμε τρεις προσεγγιστικές λύσεις π.χ. [tou1,yout1], [tout2,yout2], [tou3,yout3] και μία πραγματική [tout4,f0true(tout4)] όπου tout4=t0:0.01:tfinal.
2. Να γραφεί οδηγό πρόγραμμα (script) με όνομα run22p0.m το οποίο να λύνει το πρόβλημα 0 με τη μέθοδο Euler, με τη μέθοδο Improved Euler και με τη μέθοδο τετάρτης τάξης στο διάστημα [0,3] με βήμα h=0.1 και να υπολογίζει για καθεμία από τις τρεις προσεγγιστικές μεθόδους το απόλυτο σφάλμα στα σημεία της διαμέρισης. Στη συνέχεια να παρουσιάζει γραφικά τα πόσα δεκαδικά ψηφία ακρίβεια επιτυγχάνει η κάθε μέθοδος σε κάθε σημείο της διαμέρισης. Παρατηρήστε τη διαφορά στην ακρίβεια που επιτυγχάνει η κάθε μέθοδος. **Παρατήρηση:** Το ερώτημα είναι παρόμοιο με το ερώτημα 6 του Εργαστηρίου 1 (run04.m), θα πρέπει να λάβετε υπόψη και τα όσα κάνατε στο προηγούμενο ερώτημα.
3. Να γραφεί οδηγό πρόγραμμα (script) με όνομα run23p0.m το οποίο να λύνει το πρόβλημα 0 με το ζεύγος μεθόδων 5(4) στο διάστημα [0,8] με παράμετρο ανοχής TOL=10^{-4}. Τρέξτε το ίδιο πρόγραμμα για TOL=10^{-5} και για TOL=10^{-6} και παρατηρήστε ότι όσο μικραίνει η ανοχή TOL τόσο πιο πυκνά είναι τα σημεία της διαμέρισης. Επίσης παρατηρήστε ότι σε ομαλά σημεία της λύσης η μέθοδος παίρνει μεγαλύτερα βήματα από ότι όταν η λύση έχει μεγάλες μεταβολές. **Παρατήρηση:** Στο πρόγραμμα δεν ορίζετε τιμή για το βήμα h αλλά για την παράμετρο TOL.
4. Να γραφεί οδηγό πρόγραμμα (script) με όνομα run24p0.m το οποίο να λύνει το πρόβλημα 0 με το ζεύγος μεθόδων 5(4) στο διάστημα [0,5] για τρεις τιμές της παραμέτρου ανοχής TOL=10^{-4}, TOL=10^{-5} και για TOL=10^{-6} και να υπολογίζει για καθεμία από τις τιμές της παραμέτρου ανοχής TOL μεθόδους το απόλυτο σφάλμα στα σημεία της διαμέρισης. Στη συνέχεια να παρουσιάζει γραφικά, σε ένα γράφημα, τα πόσα δεκαδικά ψηφία ακρίβεια επιτυγχάνει η προσέγγιση σε κάθε σημείο της διαμέρισης για καθεμία από τις τιμές της παραμέτρου ανοχής TOL. Παρατηρήστε ότι όσο μικραίνει η ανοχή TOL τόσο πιο ακριβείς προσεγγίσεις επιτυγχάνουμε. **Παρατήρηση:** Το ερώτημα είναι παρόμοιο με το ερώτημα 6 του Εργαστηρίου 1 ή το Ερώτημα 2 του εργαστηρίου 2.

Αν ο χρόνος το επιτρέψει κάντε τα παραπάνω ερωτήματα για το πρόβλημα 1 και για το πρόβλημα 2 (σύστημα) αλλάζοντας τα ονόματα των scripts. Για το πρόβλημα 2 κάντε ξεχωριστά γράφημα για καθεμία από τις συνιστώσες της λύσης. Χρησιμοποιήστε τη subplot για να εμφανίζονται στο ίδιο παράθυρο και προσέξτε πως θα χρησιμοποιήσετε τα διάφορα διανύσματα.