ΤΕΜΦΕ 5^{o} Εξάμηνο
Αριθμητική Ανάλυση ΙΙ και Εργαστήριο
4^{o} Εργαστήριο και Πρακτική Εξάσκηση

Μία έμμεση πολυβηματική μέθοδος τύπου Adams-Multon λύνει αριθμητικά το πρόβλημα αρχικών τιμών :

$$y' = f\left(x, y\right),\ \ y\left(x_0\right) = y_0$$
$$f : R \times R^n \to R^n$$

με βάση τον ακόλουθο τύπο:

$$y_{n+k+1} = y_{n+k} + h\sum_{j=0}^{k+1} b_j f(x_{n+j}, y_{n+j})$$

Ο αριθμός k+1 αποτελεί τον αριθμό σταδίων της μεθόδου και το βήμα της μεθόδου h είναι σταθερό Για μία τέτοια μέθοδο σε κάθε βήμα της λύσης θα πρέπει να λύνεται ένα μη γραμμικό σύστημα εξισώσεων μιας και το y_{n+k+1} εμφανίζεται και στις δύο πλευρές του παραπάνω τύπου

Για να αποφευχθεί η λύση του μη γραμμικού συστήματος σε κάθε βήμα μπορεί να εφαρμοστεί η τεχνική της πρόβλεψης - διόρθωσης. Σε κάθε βήμα λύσης της έμμεσης μεθόδου αντικαθιστούμε το y_{n+k+1}, που χρειάζεται στο δεξί μέλος, με τη προσέγγιση

$$y^c_{n+k+1} = y_{n+k} + h(\sum_{j=0}^{k} b_j f(x_{n+j}, y_{n+j}) + b_{k+1} y^p_{n+k+1})$$

(πρόβλεψη y^p_{n+k+1}) που μας δίνει μία άμεση μέθοδος κ+2 σταδίων (π.χ. μία Adams-Bashford). Η μέθοδος τότε προωθεί τη λύση με βάση τον τύπο (διόρθωση):

Ένα τέτοιο ζευγάρι πρόβλεψης – διόρθωσης μπορεί να είναι το ακόλουθο:

$$y^c_{n+4} = y_{n+3} + \frac{h}{720}(251 f\left(x_{n+4}, y^p_{n+4}\right) + 646 f\left(x_{n+3}, y_{n+3}\right) - 264 f\left(x_{n+2}, y_{n+2}\right) + 106 f\left(x_{n+1}, y_{n+1}\right) - 19 f\left(x_n, y_n\right))$$
$$y^p_{n+4} = y_{n+3} + \frac{h}{720}(1901 f\left(x_{n+3}, y_{n+3}\right) - 2774 f\left(x_{n+2}, y_{n+2}\right) + 2616 f\left(x_{n+1}, y_{n+1}\right) - 1274 f\left(x_n, y_n\right) + 251 f\left(x_{n-1}, y_{n-1}\right))$$

Σας επισυνάπτεται η συνάρτηση (function) pc5.m που υλοποιεί την παραπάνω μέθοδο πρόβλεψης διόρθωσης και η συνάρτηση ab5semn.m που αποτελεί μία βελτιωμένη έκδοση (ως προς τον αριθμό των υπολογισμών της f) της ab5sem.m.

Να γραφεί οδηγό πρόγραμμα (script) με όνομα με όνομα run41p0.m το οποίο θα λύνει το Πρόβλημα 0 με τις μεθόδους pc5.m και ab5semn.m. Το πρόγραμμα αυτό να εμφανίζει με τη χρήση της subplot δύο γραφήματα. Στο πρώτο να εμφανίζει τη γραφική παράσταση και των δύο αριθμητικών λύσεων στα σημεία της διαμέρισης όπως και το γράφημα της πραγματικής λύσης για μία πυκνή (πλάτους 0.01) διαμέριση της λύσης. Στο δεύτερο να εμφανίζει τον αριθμό των ακριβών δεκαδικών ψηφίων σε κάθε σημείο της διαμέρισης της λύσης για καθεμία από τις δύο προσεγγιστικές λύσεις. **Μην ξεχάσετε ότι θα πρέπει να μετατρέψετε την f0.m ώστε να μπορεί να εφαρμοστεί σε διάνυσμα**. (Δηλαδή να χρησιμοποιεί πράξεις .*,./,.^ όπου χρειάζεται).

```
function [tout, yout] = ab5semn(FunFcn,t0,tfinal,step,y0)
% ab5sem.m Costant Stepsize
% Adams Bashforth 5 step Explicit Method order 4
% calls rk4 for 4 steps
beta =  [251 -1274  2616 -2774  1901  ]/720;
stages=5;
[tout,yout]=rk4(FunFcn,t0,t0+(stages-1)*step,step,y0);
tout=tout(1:stages);
yout=yout(1:stages);
fout=feval(FunFcn,tout(1:stages),yout(1:stages));
t = tout(stages);
y = yout(stages).';
% The main loop
while abs(t- tfinal)> 1e-6
  if t + step > tfinal, step = tfinal - t; end
  y=y+step* beta*[fout(length(fout)-stages+1:length(fout))];
  t = t + step;
  f=feval(FunFcn,t,y);
  tout = [tout; t];
  yout = [yout; y.'];
  fout= [fout; f.'];
end;
```

```
function [tout, yout] = pc5(FunFcn,t0,tfinal,step,y0)
% Costant Stepsize
betap =  [251 -1274  2616 -2774  1901  ]/720;
beta =  [-19  106   -264   646 251 ]/720;
stages=5;
[tout,yout]=rk4(FunFcn,t0,t0+(stages-1)*step,step,y0);
tout=tout(1:stages);
yout=yout(1:stages);
fout=feval(FunFcn,tout(1:stages),yout(1:stages));
t = tout(stages);
y = yout(stages).';
% The main loop
while abs(t- tfinal)> 1e-6
  if t + step > tfinal, step = tfinal - t; end
  yp=y+step* betap*[fout(length(fout)-stages+1:length(fout))];
  t = t + step;
  fp=feval(FunFcn,t,yp);
  y=y+step* beta*[fout(length(fout)-stages+2:length(fout)); fp];
  f=feval(FunFcn,t,y);
  tout = [tout; t];
  yout = [yout; y.'];
  fout= [fout; f.'];
end;
```