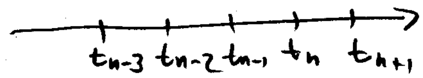
MATH 5620 HW3 Solutions

Problem 1

(a) Derive 4th order Adams-Bashforth:

$$y_{n+1} = y_n + \frac{h}{24}\left(55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}\right)$$

We use undetermined coeff method:



we need numerical integration that is exact for polynomials of degree ≤ 3 (P_3):

$$\int_0^1 p(t)\,dt = A\,p(0) + B\,p(-1) + C\,p(-2) + D\,p(-3) \quad \forall p \in P_3$$

Use basis:

$P_0(t) = 1$

$P_1(t) = t$

$P_2(t) = t(t+1)$

$P_3(t) = t(t+1)(t+2)$

we get system:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & -3 \\ 0 & 0 & 2 & 6 \\ 0 & 0 & 0 & -6 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 5/6 \\ 9/4 \end{bmatrix} \Rightarrow \begin{aligned} A &= 55/24 \\ B &= -59/24 \\ C &= 37/24 \\ D &= -9/24 \end{aligned}$$

(b) Derive 4th order Adams-Moulton method:

$$y_{n+1} = y_n + \frac{h}{24}\left(9 f_{n+1} + 19 f_n - 5 f_{n-1} + f_{n-2}\right).$$

We need an integration formula s.t.

$$\int_0^1 p(t)\,dt = A\,p(1) + B\,p(0) + C\,p(-1) + D\,p(-2)$$

Using same basis as in (a) we get:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & -2 \\ 2 & 0 & 0 & 2 \\ 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 5/6 \\ 9/4 \end{bmatrix} \rightarrow \begin{aligned} A &= 9/24 \\ B &= 19/24 \\ C &= -5/24 \\ D &= 1/24 \end{aligned}$$

Problem 2 The method

$$y_n = y_{n-2} + 2h f_{n-1}$$

has the form

$$a_2 y_n + a_1 y_{n-1} + a_0 y_{n-2} = h\left[b_2 f_n + b_1 f_{n-1} + b_0 f_{n-2}\right]$$

where

$a_2 = 1$ $b_2 = 0$
$a_1 = 0$ $b_1 = 2$
$a_0 = -1$ $b_0 = 0$

thus:

$d_0 = a_2 + a_1 + a_0 = 0$

$d_1 = \sum_{i=0}^{k} (i a_i - b_i) = (0a_0 - b_0) + (1a_1 - b_1) + (2a_2 - b_2)$
$= 0 + (-2) + 2 = 0$,

$d_2 = \sum_{i=0}^{k} \frac{i^2}{2} a_i - i b_i = (0a_0 - 0b_0) + (\frac{1}{2} a_1 - b_1) + (2a_2 - 2b_2)$
$= 0 + (-2) + 2 = 0$

$d_3 = \sum_{i=0}^{k} \frac{i^3}{6} a_i - \frac{i^2}{2} b_i = 0 + \left(\frac{1}{6} a_1 - \frac{1}{2} b_1\right) + \left(\frac{8}{6} a_2 - 2 b_2\right)$
$= 0 + 1(-1) + \frac{8}{6} = \frac{1}{3} \neq 0$

Thus the method is of order 2.

Problem 3

(a) $y_n - y_{n-2} = 2h f_{n-1}$

$p(z) = z^2 - 1$
$q(z) = 2z$

Consistency: yes
$p(1) = 0$
$p'(1) = 2 = q(1)$

Stability: yes
roots of p are ± 1, both have magnitude 1 and simple

→ method is convergent.

(b) $y_n - y_{n-2} = \frac{h}{3}\left[7f_{n-1} - 2f_{n-2} + f_{n-3}\right]$

$p(z) = z^3 - z$
$q(z) = \frac{1}{3}\left[7z^2 - 2z + 1\right]$

Consistency: yes
$p(1) = 0$
$p'(1) = 3 - 1 = 2$
$q(1) = 6/3 = 2$

Stability: yes
roots of p are ± 1 and 0
all roots lie in $|z| \leq 1$.
All roots with $|z| = 1$ are simple.

convergent

(c) $y_n - y_{n-1} = \frac{h}{24}\left[9f_n + 19 f_{n-1} - 5f_{n-2} + f_{n-3}\right]$

$p(z) = z^3 - z^2$

$q(z) = \frac{1}{24}\left[9z^3 + 19z^2 - 5z + 1\right]$

<u>Consistency</u>: yes

$p(1) = 0$

$p'(1) = 3 - 2 = 1$

$q(1) = \frac{1}{24}\left[9 + 19 - 5 + 1\right] = 1$

<u>Stability</u>: yes

roots of p are: 1 (simple)

0 (multiplicity 2)

all roots lie in $|z| \leq 1$, and roots with $|z| = 1$ are simple

Consistent & convergent ⇒ stable.

```
>> drva
```

——————— 5.6.2 a ———————

t	AB2	AB3	AB4	AB5	True
0.1	1.1881188	1.1881188	1.1881188	1.1881188	1.1881188
0.2	1.3498578	1.3461536	1.3461536	1.3461536	1.3461538
0.3	1.4731966	1.4661841	1.4678893	1.4678893	1.4678899
0.4	1.5565908	1.5485054	1.5517428	1.5517232	1.5517241
0.5	1.6029884	1.5957945	1.6003967	1.6003394	1.6000000
0.6	1.6180985	1.6131034	1.6184959	1.6180454	1.6176471
0.7	1.6086462	1.6063884	1.6119612	1.6110982	1.6107383
0.8	1.5811238	1.5815374	1.5867846	1.5855345	1.5853659
0.9	1.5411265	1.5437887	1.5484120	1.5469924	1.5469613
1.0	1.4931330	1.4974823	1.5013657	1.4999071	1.5000000

——————— 5.6.2 b ———————

t	AB2	AB3	AB4	AB5	True
1.1	−1.3478227	−1.3478227	−1.3478227	−1.3478227	−1.3478227
1.2	−1.2700979	−1.2682994	−1.2682994	−1.2682994	−1.2682994
1.3	−1.2033635	−1.2001371	−1.2006111	−1.2006111	−1.2006112
1.4	−1.1455857	−1.1415554	−1.1423959	−1.1422452	−1.1422452
1.5	−1.0950429	−1.0905163	−1.0915669	−1.0913018	−1.0913567
1.6	−1.0504367	−1.0456465	−1.0468223	−1.0464860	−1.0465599
1.7	−1.0107604	−1.0058463	−1.0070697	−1.0066984	−1.0067941
1.8	−0.9752224	−0.9702764	−0.9715163	−0.9711385	−0.9712327
1.9	−0.9431920	−0.9382726	−0.9395038	−0.9391234	−0.9392222
2.0	−0.9141608	−0.9093054	−0.9105164	−0.9101442	−0.9102392

```
>> drvb
```

——————— 5.6.6 a ———————

t	A4PC	True
0.1	1.1881188	1.1881188
0.2	1.3461536	1.3461538
0.3	1.4678893	1.4678899
0.4	1.5516984	1.5517241
0.5	1.5999256	1.6000000
0.6	1.6175240	1.6176471
0.7	1.6105825	1.6107383
0.8	1.5851977	1.5853659
0.9	1.5467972	1.5469613
1.0	1.4998499	1.5000000

——————— 5.6.6 b ———————

t	A4PC	True
1.1	−1.3478227	−1.3478227
1.2	−1.2682994	−1.2682994
1.3	−1.2006111	−1.2006112
1.4	−1.1422321	−1.1422452
1.5	−1.0913369	−1.0913567
1.6	−1.0465369	−1.0465599
1.7	−1.0067696	−1.0067941
1.8	−0.9712077	−0.9712327
1.9	−0.9391974	−0.9392222
2.0	−0.9102149	−0.9102392

```
>>
```

```
% Adams Fourth-Order Predictor-Corrector
function y = a4pc(a,b,n,y0,f)
h=(b-a)/n;
% prime the first steps with RK4
% note: to avoid unnecessary calculations we store all function evaluations.
% it's possible to avoid this memory overhead by using some FIFO structure
y=rk4(a,a+3*h,3,y0,f);
for i=1:4,
 ti=a+(i-1)*h;
 fs(i) = f(ti,y(i));
end;

for i=4:n,
 % predictor: AB4
 y(i+1)  = y(i) + (h/24)*(55*fs(i) - 59*fs(i-1) + 37*fs(i-2) -9*fs(i-3));
 fs(i+1) = f(a+i*h,y(i+1));

 % corrector: AM4
 y(i+1)  = y(i) + (h/24)*(9*fs(i+1) +19*fs(i) - 5*fs(i-1) +   fs(i-2));
 fs(i+1) = f(a+i*h,y(i+1));
end;
```



```
% Adams-Bashforth Two-Step Explicit Method
function y = ab2(a,b,n,y0,f)
h=(b-a)/n;
% prime the first steps with RK4
% note: to avoid unnecessary calculations we store all function evaluations.
% it's possible to avoid this memory overhead by using some FIFO structure
y=rk4(a,a+h,1,y0,f);
for i=1:2,
 ti=a+(i-1)*h;
 fs(i) = f(ti,y(i));
end;

% carry out method
for i=2:n,
 y(i+1)  = y(i) + (h/2)*(3*fs(i) - fs(i-1));
 fs(i+1) = f(a+i*h,y(i+1));
end;
```

```
% Adams-Bashforth Three-Step Explicit Method
function y = ab3(a,b,n,y0,f)
h=(b-a)/n;
% prime the first steps with RK4
% note: to avoid unnecessary calculations we store all function evaluations.
% it's possible to avoid this memory overhead by using some FIFO structure
y=rk4(a,a+2*h,2,y0,f);
for i=1:3,
 ti=a+(i-1)*h;
 fs(i) = f(ti,y(i));
end;

% carry out method
for i=3:n,
 y(i+1)  = y(i) + (h/12)*(23*fs(i) - 16*fs(i-1) + 5*fs(i-2));
 fs(i+1) = f(a+i*h,y(i+1));
end;
```



```
% Adams-Bashforth Four-Step Explicit Method
function y = ab4(a,b,n,y0,f)
h=(b-a)/n;
% prime the first steps with RK4
% note: to avoid unnecessary calculations we store all function evaluations.
% it's possible to avoid this memory overhead by using some FIFO structure
y=rk4(a,a+3*h,3,y0,f);
for i=1:4,
 ti=a+(i-1)*h;
 fs(i) = f(ti,y(i));
end;

% carry out method
for i=4:n,
 y(i+1)  = y(i) + (h/24)*(55*fs(i) - 59*fs(i-1) + 37*fs(i-2) -9*fs(i-3));
 fs(i+1) = f(a+i*h,y(i+1));
end;
```

```
% Adams-Bashforth Five-Step Explicit Method
function y = ab5(a,b,n,y0,f)
h=(b-a)/n;
% prime the first steps with RK4
% note: to avoid unnecessary calculations we store all function evaluations.
% it's possible to avoid this memory overhead by using some FIFO structure
y=rk4(a,a+4*h,4,y0,f);
for i=1:5,
 ti=a+(i-1)*h;
 fs(i) = f(ti,y(i));
end;

% carry out method
for i=5:n,
 y(i+1)  = y(i) + (h/720)*(1901*fs(i) - 2774*fs(i-1) + 2616*fs(i-2) -1274*fs(i-3
) + 251*fs(i-4));
 fs(i+1) = f(a+i*h,y(i+1));
end;
```



```
function drva
% we use for reference 5.6.3 a:
%n=10; a=1; b=2; f = @(t,y) y/t - (y/t)^2; y0=1;
%y = @(t) t./(1+log(t));
fprintf('-------------- 5.6.2 a ------------------\n');
n=10; a=0; b=1; y0=1; f = @(t,y) (2-2*t*y)/(t^2+1);
y = @(t) (2*t+1)./(t.^2+1);
compareab(a,b,n,y0,f,y);

fprintf('-------------- 5.6.2 b ------------------\n');
n=10; a=1;b=2; y0=-1/log(2); f = @(t,y) y^2/(1+t);
y = @(t) -1./log(1+t);
compareab(a,b,n,y0,f,y);

function compareab(a,b,n,y0,f,y)
yab2 = ab2(a,b,n,y0,f);
yab3 = ab3(a,b,n,y0,f);
yab4 = ab4(a,b,n,y0,f);
yab5 = ab5(a,b,n,y0,f);
ytrue=y(a+(0:n)*(b-a)/n);
fprintf('%4s %10s %10s %10s %10s %10s\n','t','AB2','AB3','AB4','AB5','True');
for i=2:n+1,
 fprintf('%4.1f %10.7f %10.7f %10.7f %10.7f %10.7f\n', ...
        a+(i-1)*(b-a)/n,yab2(i),yab3(i),yab4(i),yab5(i),ytrue(i));
end;
```

```
function drvb
%  we use for reference 5.6.3 a:
%n=10; a=1; b=2; f = @(t,y) y/t - (y/t)^2; y0=1;
%y = @(t) t./(1+log(t));
%comparepc(a,b,n,y0,f,y);

fprintf('-------------- 5.6.6 a ------------------\n');
n=10; a=0; b=1; y0=1; f = @(t,y) (2-2*t*y)/(t^2+1);
y = @(t) (2*t+1)./(t.^2+1);
comparepc(a,b,n,y0,f,y);

fprintf('-------------- 5.6.6 b ------------------\n');
n=10; a=1;b=2; y0=-1/log(2); f = @(t,y) y^2/(1+t);
y = @(t) -1./log(1+t);
comparepc(a,b,n,y0,f,y);

function comparepc(a,b,n,y0,f,y)
ypc = a4pc(a,b,n,y0,f);
ytrue=y(a+(0:n)*(b-a)/n);
fprintf('%4s %10s %10s\n','t','A4PC','True');
for i=2:n+1,
 fprintf('%4.1f %10.7f %10.7f\n', ...
         a+(i-1)*(b-a)/n,ypc(i),ytrue(i));
end;
```



```
function y = rk4(a,b,n,y0,f)
 y(1) = y0;
 h=(b-a)/n;
 for i=1:n,
  ti=a+(i-1)*h;
  F1 = h*f(ti,y(i));
  F2 = h*f(ti+h/2,y(i)+F1/2);
  F3 = h*f(ti+h/2,y(i)+F2/2);
  F4 = h*f(ti+h  ,y(i)+F3);
  y(i+1) = y(i) + (F1 + 2*F2 + 2*F3 + F4)/6;
 end;
```