6220HW2

Keyan Chen

February 2024

1 problem1

for backward Euler method stability testing,we have:
$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$
$= y_{n+1} = y_n + \Delta t \lambda y_{n+1}$
by rearrange, we have:
$y_{n+1} - \Delta t \lambda y_{n+1} = y_n$
$(1 - \Delta t \lambda) y_{n+1} = y_n$
$y_{n+1} = \frac{y_n}{1 - \Delta t \lambda}$
similar to the forward euler method, the model problem also has a decaying solution for $\lambda < 0$. and we want $\frac{1}{1-\Delta t \lambda} \leq 1$
we find that when $\lambda < 0$, and $\Delta t > 0$, it is stable. because the split point is 0, we can say that it is A-stable because the region of absolute stability includes the entire left half of the complex plane.

For Trapezodial Rule stability testing, we have:
$y_{n+1} = y_n + \frac{\Delta t}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$
$= y_n + \frac{\Delta t}{2}(\lambda y_n + \lambda y_{n+1})$
by rearrange, we have:
$y_{n+1} - \frac{\Delta t}{2}\lambda y_{n+1}) = y_n + \frac{\Delta t}{2}\lambda y_n$
$y_{n+1} = \frac{1+\frac{\Delta t}{2}}{1-\frac{\Delta t}{2}} y_n$
in this method, we want $\frac{1+\frac{\Delta t}{2}}{1-\frac{\Delta t}{2}} \leq 1$
with $\lambda < 0$, when $\Delta t > 0$, it is stable. similar to the above one. 0 is the point. Therefore, we can say that it is A-stable because the region of absolute stability includes the entire left half of the complex plane.

2 problem2

problem(a):
the following code is in file AB2_FE.m

```matlab
function [tvec,yvec] = AB2_FE(t0,y0,f,h,N)
% [tvec,yvec] = AB2_FE(t0,y0,y1,f,h,N)
% Adams-Bashforth 2nd-order method
% Inputs
% t0,y0: initial condition
% y1:    additional start-up value
% f:     names of the right-hand side function f(t,y)
% h:     stepsize
% N:     number of steps
% Outputs
% tvec: vector of t values
% yvec: vector of corresponding y values

yvec = zeros(N+1,1);
tvec = linspace(t0,t0+N*h,N+1)';
yvec(1) = y0;
%Forward Euler for the first step to get y1
yvec(2) = y0 + h * f(t0, y0);
for n=1:N-1
   fvalue1 = f(tvec(n),yvec(n));
   fvalue2 = f(tvec(n+1),yvec(n+1));
   yvec(n+2) = yvec(n+1)+h/2*(3*fvalue2-fvalue1);
end
```

problem(b):

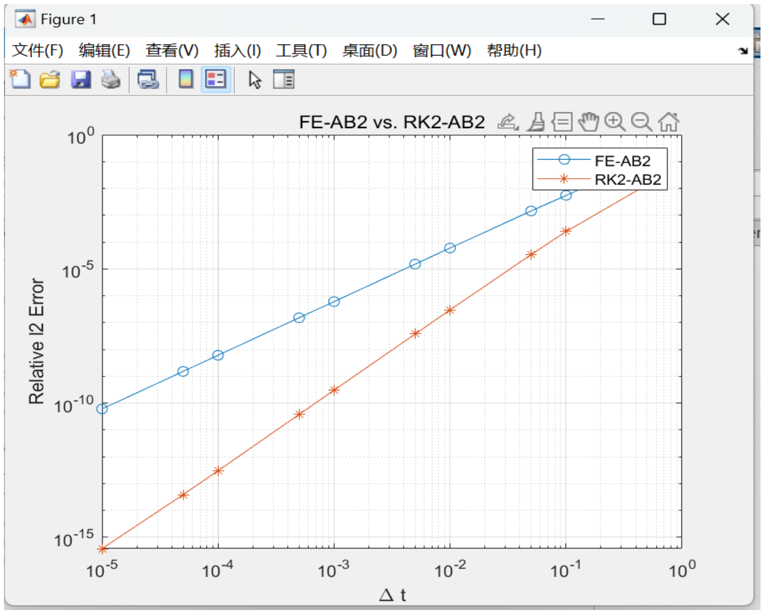
the following code is in file AB2_RK2.m

```matlab
function [tvec,yvec] = AB2_RK2(t0,y0,f,h,N)
% [tvec,yvec] = AB2_RK2(t0,y0,y1,f,h,N)
% Adams-Bashforth 2nd-order method
% Inputs
% t0,y0: initial condition
% y1:    additional start-up value
% f:     names of the right-hand side function f(t,y)
% h:     stepsize
% N:     number of steps
% Outputs
% tvec: vector of t values
% yvec: vector of corresponding y values

yvec = zeros(N+1,1);
tvec = linspace(t0,t0+N*h,N+1)';
yvec(1) = y0;
%RK2 for the first step to get y1
y_star = y0 + h * f(t0, y0);
yvec(2) = y0 + (h/2) * (f(t0, y0) + f(t0 + h, y_star));
for n=1:N-1
   fvalue1 = f(tvec(n),yvec(n));
   fvalue2 = f(tvec(n+1),yvec(n+1));
   yvec(n+2) = yvec(n+1)+h/2*(3*fvalue2-fvalue1);
end
```

problem(c):

I choose the Δt is [0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5]



we can see that at 0.01, the RK2_AB2 have the lower l2 error. and overall, the RK2_AB2 have the better accuracy. when the Δt becomes smaller, we can't see much difference with errors.

3 problem3

problem(a):
the following code is in file AB3.m

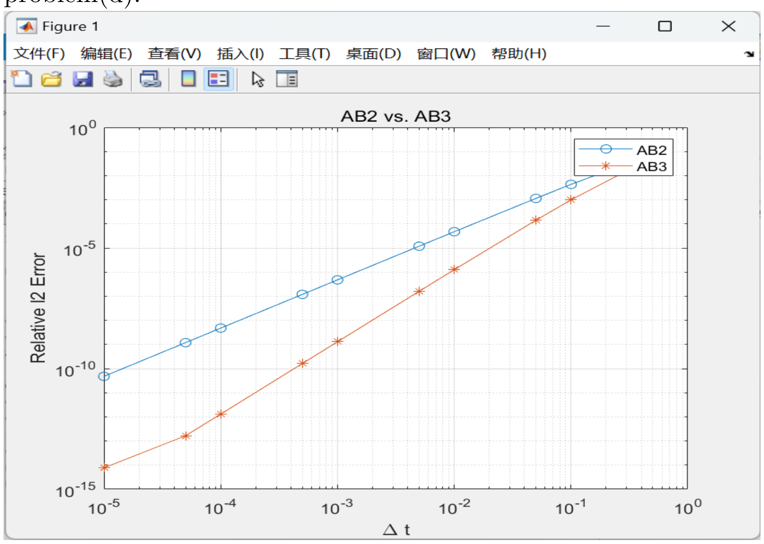
```
function [tvec,yvec] = AB3(t0,y0,y1,f,h,N)
% [tvec,yvec] = AB3(t0,y0,y1,f,h,N)
% Adams-Bashforth 2nd-order method
% Inputs
% t0,y0: initial condition
% y1:    additional start-up value
% f:     names of the right-hand side function f(t,y)
% h:     stepsize
% N:     number of steps
% Outputs
% tvec: vector of t values
% yvec: vector of corresponding y values
yvec = zeros(N+1,1);
tvec = linspace(t0,t0+N*h,N+1)';
yvec(1) = y0;
yvec(2) = y1;

% using RK3 to get third y value that will use in AB3
t1 = t0+h
k1 = f(t1, y1);
k2 = f(t1 + h/2, y1 + (1/2)*h*k1);
k3 = f(t1 + h, y1 - h*k1 + 2*h*k2);
yvec(3) = yvec(2) + (h/6)*(k1 + 4*k2 + k3);

for n=2:N-1
   %using first three value in y vector to calculate three fvalue in AB3
   fvalue1 = f(tvec(n-1), yvec(n-1));
   fvalue2 = f(tvec(n), yvec(n));
   fvalue3 = f(tvec(n+1), yvec(n+1));
   %using three f values to calculate next y value by using AB3 formula
   yvec(n+2) = yvec(n+1) + (h/12)*(23*fvalue3 - 16*fvalue2 + 5*fvalue1);
end
```

problem(b):

A stiff problem is that if all its eigenvalues have negative real parts, and the ratio of the magnitudes of the real parts of largest and smallest eigenvalues is large.
In this case that eigenvalues are negative, we need to focus on the stability. we see that lower order AB method can perform better on stability. Therefore, we should use AB2 in this case.
problem(c):
For a problem with purely imaginary λ, in this case, the $\Delta t\lambda$ values are lie on the vertical axis on the complex plane. Here, we want that the stability region encloses the appropriate part of the imaginary axis. The higher order of AB method will perform better here. Therefore, AB3 is a better choice.
problem(d):



from the diagram, we can see that as the δt decreases, the AB3 l2 error will lower faster, therefore, the AB3 converge faster.

4 problem4

problem(a):
for RK3 method: $y_{n+1} = y_n + \Delta t(b_1k_1 + b_2k_2 + b_3k_3)$
$k_1 = f(t_n, y_n)$
$k_2 = f(t_n + c_2\Delta t, y_n + \Delta t a_{21}k_1)$
$k_3 = f(t_n + c_3\Delta t, y_n + \Delta t a_{31}k_1 + \Delta t a_{32}k_2)$

For k2, using Taylor expansion, we have:
$f(t_n+c_2\Delta t, y_n+\Delta t a_{21}k_1) = f(t_n, y_n)+c_2\Delta t f_t(t_n, y_n)+\Delta t a_{21} f_y(t_n, y_n)f(t_n, y_n)$
for K3, using Taylor expansion, we have:
$f(t_n+c_3\Delta t, y_n+\Delta t a_{31}k_1+\Delta t a_{32}k_2) = f(t_n, y_n)+c_3\Delta t f_t(t_n, y_n)+\Delta t a_{31} f_y(t_n, y_n)k_1+ \Delta t a_{32} f_y(t_n, y_n)k_2$

$= f(t_n,y_n)+c_3\Delta t f_t(t_n,y_n)+\Delta t a_{31} f_y(t_n,y_n)f(t_n,y_n)+\Delta t a_{32} f_y(t_n,y_n)(f(t_n,y_n)+$
$c_2\Delta t f_t(t_n,y_n)+\Delta t a_{21} f_y(t_n,y_n)f(t_n,y_n))$
$= f(t_n,y_n)+c_3\Delta t f_t(t_n,y_n)+\Delta t a_{31} f_y(t_n,y_n)f(t_n,y_n)+\Delta t a_{32} f_y(t_n,y_n)f(t_n,y_n)+$
$\Delta t^2 a_{32}c_2 f_y(t_n,y_n)f_t(t_n,y_n)+\Delta t^2 a_{32}a_{21} f_y(t_n,y_n)^2 f(t_n,y_n)$

for y_{n+1}, using taylor expansion, we have:
$y_{n+1} = y_n + \Delta t f(t_n,y_n) + \frac{(\Delta t)^2}{2}(f_t(t_n,y_n)+f_y(t_n,y_n)f(t_n,y_n)) + O((\Delta t)^3)$
then plug this expression into the above RK3 scheme:
$y_{n+1} = y_n+b_1\Delta t f(t_n,y_n)+b_2\Delta t(f(t_n,y_n)+c_2\Delta t f_t(t_n,y_n)+\Delta t a_{21} f_y(t_n,y_n)f(t_n,y_n))+$
$b_3\Delta t(f(t_n,y_n)+c_3\Delta t f_t(t_n,y_n)+\Delta t a_{31} f_y(t_n,y_n)f(t_n,y_n)+\Delta t a_{32} f_y(t_n,y_n)f(t_n,y_n)+$
$\Delta t^2 a_{32}c_2 f_y(t_n,y_n)f_t(t_n,y_n)+\Delta t^2 a_{32}a_{21} f_y(t_n,y_n)^2 f(t_n,y_n))$
$= y_n+b_1\Delta t f(t_n,y_n)+b_2\Delta t f(t_n,y_n)+b_3\Delta t f(t_n,y_n)+b_2c_2\Delta t^2 f_t(t_n,y_n)+b_3c_3\Delta t^2 f_t(t_n,y_n)+$
$b_2a_{21}\Delta t^2 f_y(t_n,y_n)f(t_n,y_n)+b_3a_{31}\Delta t^2 f_y(t_n,y_n)f(t_n,y_n)+b_3a_{32}\Delta t^2 f_y(t_n,y_n)f(t_n,y_n)+$
$b_3a_{32}c_2\Delta t^3 f_y(t_n,y_n)f_t(t_n,y_n)+b_3a_{32}a_{21}\Delta t^3 f_y(t_n,y_n)^2 f(t_n,y_n)$
$= y_n+(b_1+b_2+b_3)\Delta t f(t_n,y_n)+(b_2c_2+b_3c_3)\Delta t^2 f_t(t_n,y_n)+(b_2a_{21}+b_3a_{31}+$
$b_3a_{32})\Delta t^2 f_y(t_n,y_n)f(t_n,y_n)+b_3a_{32}c_2\Delta t^3 f_y(t_n,y_n)f_t(t_n,y_n)+b_3a_{32}a_{21}\Delta t^3 f_y(t_n,y_n)^2 f(t_n,y_n)$

from above, we can find three nolinear equations:
$b_1+b_2+b_3=1$
$b_2c_2+b_3c_3=\frac{1}{2}$
$b_2a_{21}+b_3a_{31}+b_3a_{32}=\frac{1}{2}$

One possible solution is that:
$b_1=\frac{1}{6}$, $b_2=\frac{4}{6}$,$b_3=\frac{1}{6}$
$c_1=\frac{1}{2}$, $c_2=1$
$a_{21}=\frac{1}{2}$, $a_{31}=-1$,$a_{32}=2$

Then we can derive that one possible RK3 is:
$y_{n+1}=y_n+\Delta t(\frac{1}{6}k_1+\frac{4}{6}k_2+\frac{1}{6}k_3)$
$k_1=f(t_n,y_n)$
$k_2=f(t_n+\frac{1}{2}\Delta t, y_n+\Delta t\frac{1}{2}k_1)$
$k_3=f(t_n+\Delta t, y_n-\Delta t k_1+2\Delta t k_2)$

problem(b):

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
1 & -1 & 2 & \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}$$

problem(c):
the Butcher tableau for the RK4 3/8 method is the following:

$$\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{3} & \frac{1}{3} & & & \\
\frac{2}{3} & -\frac{1}{3} & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
\end{array}$$

The stability is require that:
$y_{n+1} = R(\Delta t\lambda)y_n$
$R(\Delta t\lambda) = 1 + \Delta t\lambda[\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}](I - \Delta t\lambda A)^{-1} * 1$
$= \frac{1}{24}\Delta t^4\lambda^4 + \frac{1}{6}\Delta t^3\lambda^3 + \frac{1}{2}\Delta t^2\lambda^2 + \Delta t\lambda + 1 \leq 1$