Math 4NA3 - Assignment 4

March 15, 2021

Question 1:

a) Here is the code to plot $|g(z)| = 1$ for $g(z) = \sum_{k=1}^{10} \frac{z^k}{k!}$

```
x = linspace(-6,6,100);y = x;
[x,y]=meshgrid(x,y);z = x + 1i*y;

g = @(x,y) z + z.^2/2 + z.^3/6 + z.^4/factorial(4) ...
    + z.^5/factorial(5) + z.^6/factorial(6) + z.^7/factorial(7) ...
    + z.^8/factorial(8) + z.^9/factorial(9) + z.^10/factorial(10);

contourf(x,y,abs(g(x,y)),[1 1]); axis('equal');
grid on;xlabel('Re');ylabel('Im');
```

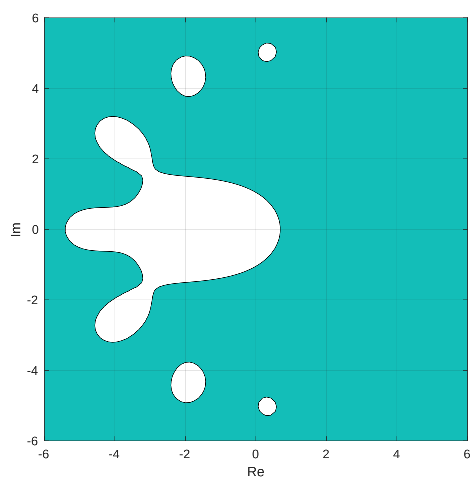

Figure 1: Level Curve $|g(z)| = 1$

b) Under the model equation, we have the following calculation.

$k_1 = y_n \left[(\lambda h)\right]$
$k_2 = (\lambda h)(y_n + (\lambda h)y_n/2) = y_n \left[(\lambda h) + (\lambda h)^2/2\right]$
$k_3 = (\lambda h)(y_n - (\lambda h)y_n + 2(\lambda h)(1 + (\lambda h)/2)) = y_n \left[(\lambda h) + (\lambda h)^2 + (\lambda h)^3\right]$
$\Longrightarrow$
$y_{n+1} = y_n \left[1 + (\lambda h)/6 + 2/3(\lambda h) + (\lambda h)^3/3 + (\lambda h)/6 + (\lambda h)^2/6 + (\lambda h)^3/6\right]$

$= y_n \left[1 + (\lambda h) + (\lambda h)^2/2! + (\lambda h)^3/3!\right]$

$= y_n E[(\lambda h)]$

c) Using the code from part a, we can add the zeroth term and remove the higher order terms to plot $p = 1, \ldots, 5$ resulting in the following figures. We see below that as p increases, so too does the stability region of the Runge-Kutta method.

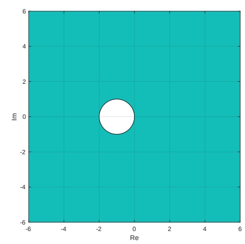

Figure 2: $p = 1$

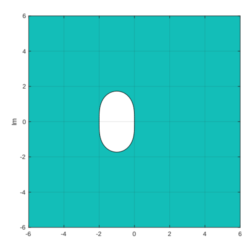

Figure 3: $p = 2$

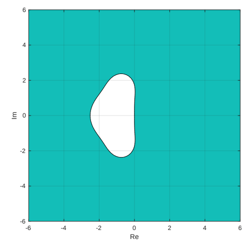

Figure 4: $p = 3$

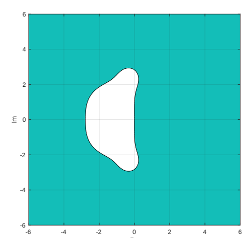

Figure 5: $p = 4$

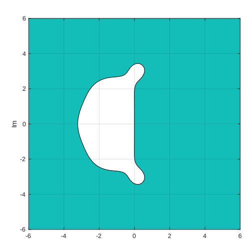

Figure 6: $p = 5$

Question 2:

a) We are given the equation

$$\frac{d^2y}{dx^2} - \frac{dy}{dx} - y = x(x-1),\ 0 < x < 1$$

From the hint, we know the exact solution is

$$y(x) = c_1 \exp\left(a_1 x/2\right) + c_2 \exp\left(a_2 x/2\right),\ \text{where } a_1 = 1+\sqrt{5}, a_2 = 1-\sqrt{5}$$

and where (c_1, c_2) solve

$$a_1 c_1 + a_2 c_2 = -6$$

$$a_1 c_1 \exp\left(a_1/2\right) + a_2 c_2 \exp\left(a_2/2\right) = -2$$

Solving this system in MATLAB gives

```
a1 = 1+sqrt(5);
a2 = 1-sqrt(5);
M = [a1 a2
     a1*exp(a1/2) a2*exp(a2/2)];
B = [-6 -2]';
cs = M\B
```

```
cs =
     0.0847
     5.0758
```

Thus, $c_1 = 0.0847$ and $c_2 = 5.0758$ - note that this also implies that $y(0) = 0.1605$ and $y(1) = 0.1630$. Using the central difference approximations $y'' = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2}$ and $y' = \frac{y_{n+1} - y_{n-1}}{2h}$ in the above equation results in the following

$$y_{n+1}\left[\frac{1}{h^2} - \frac{1}{2h}\right] + y_n\left[\frac{-2}{h^2} - 1\right] + y_{n-1}\left[\frac{1}{h^2} + \frac{1}{2h}\right] = x(x-1)$$

Therefore we can approximate the equation by solving the system as follows

```
h       = 0.1;
x       = 0:h:1;
k       = length(x);
A1      = sparse(1 : k, 1 : k,(-2 / h^2 - 1) * ones(1,k), k, k);
A2   = sparse(2 : k, 1 : k-1,(1  / h^2 + 1 / 2/h) * ones(1, k - 1), k, k);
A3    = sparse(1 : k - 1, 2 : k,(1  / h^2 - 1 / 2/h) * ones(1, k - 1), k, k);
A       = A1 + A2 + A3;
A(1,1) = 1; A(1,2) = 0;
A(end,end) = 1; A(end,end - 1) = 0;
```

```
b = (x.*(x-1))'; b(1) = 0.1605; b(end) = 0.1630;
y = A\b;

hold on
grid on
plot(x, y);
fplot(@(x) 0.0847.*exp(a1.*x./2) + 5.0758.*exp(a2.*x./2) ...
    -x.^2 + 3.*x -5,[0,1]);
```

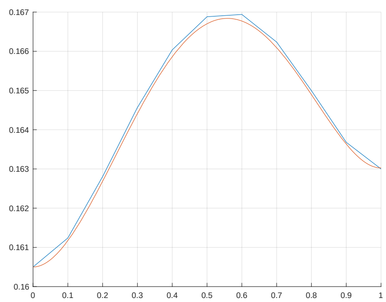

Figure 7: $h = 0.1$

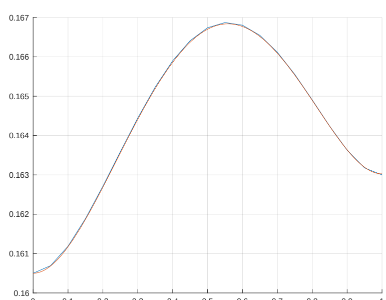

Figure 8: $h = 0.05$

b) Lastly, we plot all of the approximations together including Romberg and the exact curve.

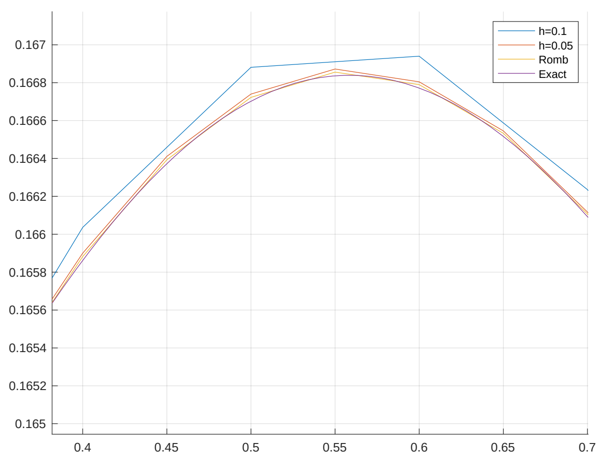

Figure 9: All Plots

Question 3:

We have the system

$$\frac{dx}{dt} = -x + xy,\ x(0) = 3$$
$$\frac{dy}{dt} = x + 2x^2 - 10y,\ y(0) = 5$$

with $t \in [0, 2]$. Semi-Implicit Euler is given by $[I - hDf(t_n, Y_n)][Y_{n+1} - Y_n] = hf(t_n, Y_n)$. In our case, we have

$$f(t, Y) = \begin{bmatrix} -x + xy \\ x + 2x^2 - 1 = y \end{bmatrix},\ Df(t, Y) = \begin{bmatrix} -1 + y & x \\ 1 + 4x & -10 \end{bmatrix}$$

Applying Semi-Implicit Euler results in the following system

$$x_{n+1}[h - hy_n + 1] + y_{n+1}[-hx_n] = x_n - hx_n y_n$$

$$x_{n+1}[-h - 4hx_n] + y_{n+1}[1 + 10h] = y_n - 2hx_n^2$$

The code below solves this and plots each component for $h = 0.01$ and $h = 0.1$.

```
h = 0.01;
t = 0:h:2;
x(1) = 3;
y(1) = 5;
n = length(t);
for k = 1 : n
    t(k+1) = t(k)+h;
    A = [h-h*y(k)+1,-h*x(k);-h-4*h*x(k),1+10*h];
    b = [x(k)-h*x(k)*y(k);y(k)-2*h*x(k)^2];
    Y = A\b;
    x(k+1) = Y(1);
    y(k+1) = Y(2);
end
figure(1); plot(t,x,'.b');
xlabel("t");ylabel("x")
grid on
figure(2); plot(t,y,'.r');
xlabel("t");ylabel("y")
grid on
```

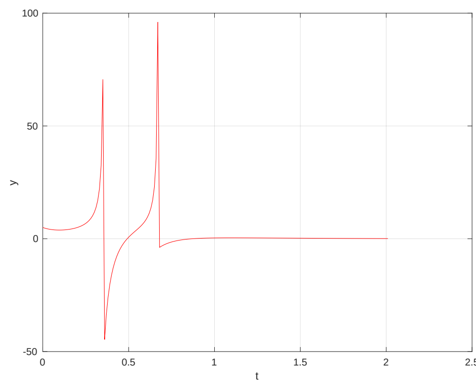

Figure 10: $h = 0.01, y$

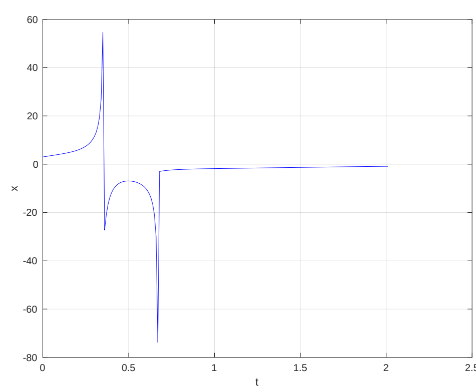

Figure 11: $h = 0.01, x$

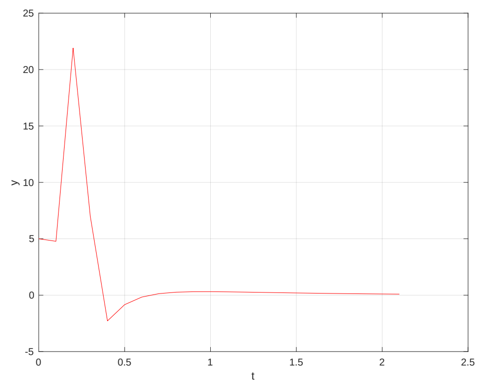

Figure 12: $h = 0.1, y$

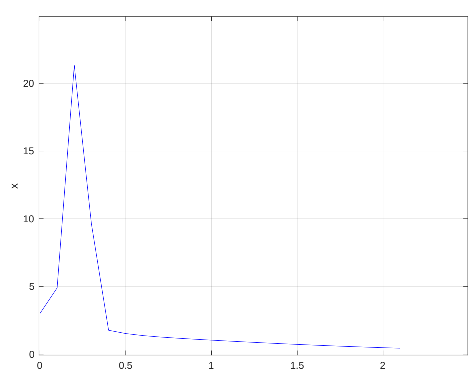

Figure 13: $h = 0.1, x$

Question 4:

a) We are given

$$y_{k+1} = y_{k-1} + \frac{h}{3}(f_{k-1} + 4f_k + f_{k+1})$$

Using the transformation $y_k = q^k$ and the model equation gives us the following quadratic

$$(1 - \frac{\lambda h}{3})q^2 + (\frac{-4\lambda h}{3})q + (-1 - \frac{\lambda h}{3}) = 0$$

This allows us to plot the stability as follows.

```
hSpan = linspace(-10,10,100);
for j = 1 : length(hSpan)
h = hSpan(j);
    q(:,j) = roots([1-h/3,-4*h/3,-1-h/3])';
end

plot(hSpan,max(abs(q)),'b');
hold on
grid on
yline(1,"r")
xlabel('z');ylabel('|q|');
```

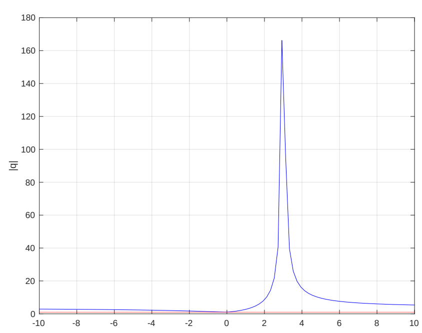

Figure 14: Stability Plot

Notice that $|q|$ is always larger than 1 for at least one root, therefore, the method is unconditionally unstable.

b) Applying the approximation $f_{k+1} \approx f_k + (y_{k+1} - y_k)f'_k$ to Milne's method gives the following semi-implicit method

$$y_{k+1} = y_{k-1} + \frac{h}{3}(f_{k-1} + 5f_k + (y_{k+1} - y_k)f'_k)$$

$\equiv$

$$y_{k+1}(1 - \frac{hf'_k}{2}) = y_{k-1} + \frac{h}{3}(f_{k-1} + 5f_k - y_k f'_k)$$

$\equiv$

$$y_{k+1} = [y_{k-1} + \frac{h}{3}(f_{k-1} + 5f_k - y_k f'_k)]/[1 - \frac{hf'_k}{2}]$$

c) To apply this method to $y' = y(3 - 4y + y^2), y(0) = 2$ we take one Euler step to get a second condition $y_2 = 2 + f(2)h = 2 - 2h$. Here is the code.

```
h = 0.05;
t = 0:h:5;
y(1) = 2;
y(2) = 2-2*h;
n = length(t);
for k = 2 : n
    t(k+1) = t(k)+h;
    c1 = f(y(k-1));
    c2 = f(y(k));
    c3 = fp(y(k));
    top = y(k-1) + (h/3)* (c1 +5*c2 - y(k)*c3);
    bot = 1 - (h/3)*c3;
    y(k+1) = top/bot;
end
figure(1); plot(t,y,'b');
xlabel("t");ylabel("y")
grid on
f(2)

function f1 = f(y)
f1 = y*(3-4*y+y^2);
end

function f2 = fp(y)
f2 = 3-8*y+3*y^2;
end
```

While both plots oscillate, the magnitude of oscillation is smaller and more rapid for $h = 0.05$ when compared to $h = 0.1$.

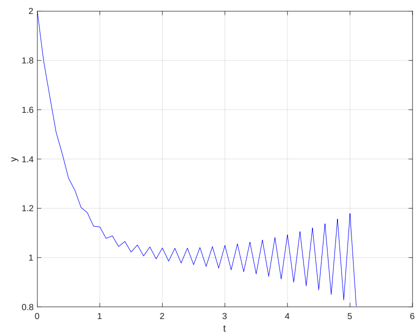

Figure 15: $h = 0.1$

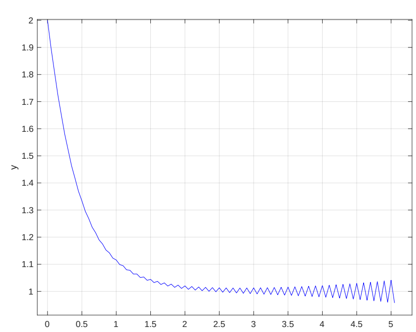

Figure 16: $h = 0.05$