Computational and Simulation Methods - HW1

Jamie Barker

October 2019

1 Write down a finite difference fourth-order accurate Runge-Kutta

Let $\frac{dx}{dt} = f(x, y, t)$ and $\frac{dx}{dt} = g(x, y, t)$.

Then the 4th order Runge-Kutta FD Scheme with time-step h is:

$$a_1 = f(x_n, y_n, t_n)$$

$$b_1 = g(x_n, y_n, t_n)$$

$$a_2 = f\left(x_n + \frac{a_1 h}{2}, y_n + \frac{b_1 h}{2}, t_n + \frac{h}{2}\right)$$

$$b_2 = g\left(x_n + \frac{a_1 h}{2}, y_n + \frac{b_1 h}{2}, t_n + \frac{h}{2}\right)$$

$$a_3 = f\left(x_n + \frac{a_2 h}{2}, y_n + \frac{b_2 h}{2}, t_n + \frac{h}{2}\right)$$

$$b_3 = g\left(x_n + \frac{a_2 h}{2}, y_n + \frac{b_2 h}{2}, t_n + \frac{h}{2}\right)$$

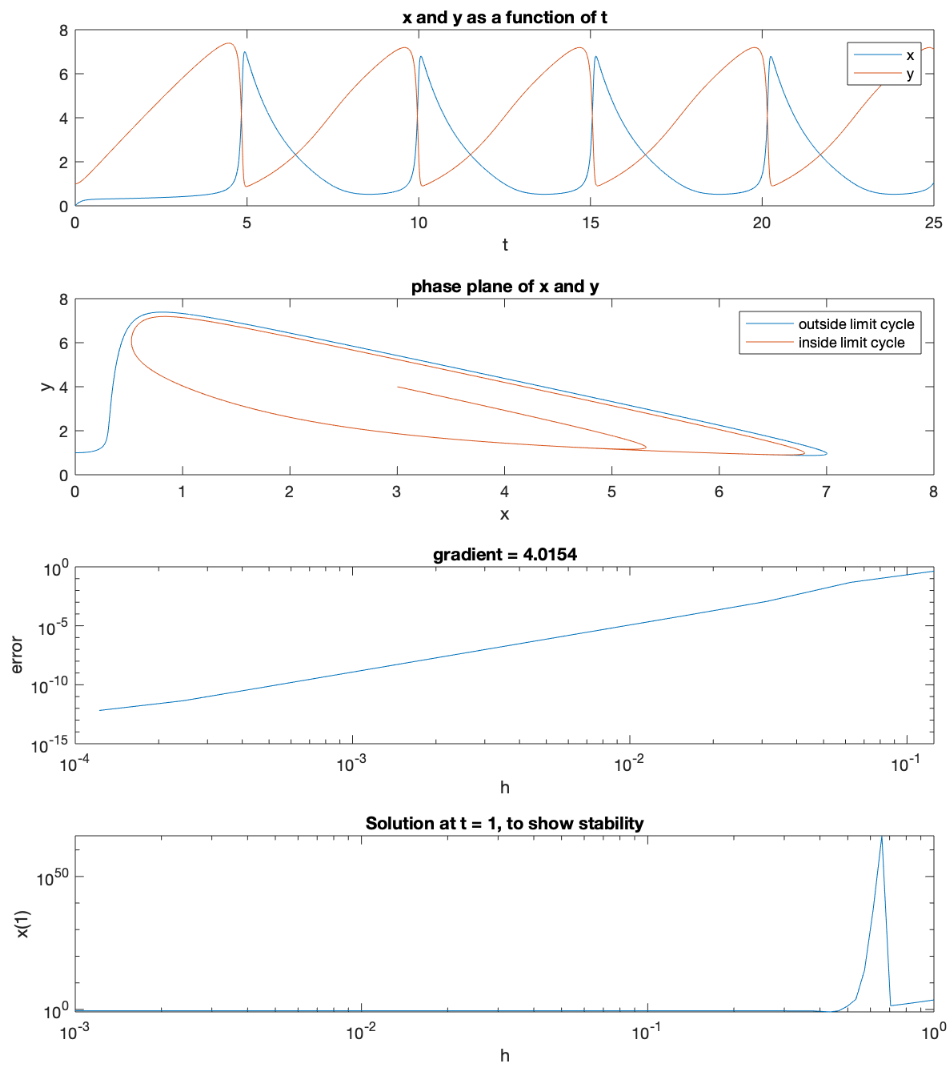
$$a_4 = f\left(x + a_3 h, y + b_3 h, t + h)\right)$$

$$b_4 = g\left(|x + a_3 h, y + b_3 h, t + h)\right)|$$

$$x_{n+1} = x + (a_1 + 2a_2 + 2a_3 + a_4)$$

$$y_{n+1} = x + (b_1 + 2b_2 + 2b_3 + b_4)$$

In the context of a brusselator system, $f\left(x, y, t\right) = A - Bx + yx^2 - x$ and $g\left(x, y, t\right) = Bx - yx^2$



2 Write a Matlab code to solve the finite difference equations

```
A = 2;
```

```
B = 6;
x0 = 0;
y0 = 1;
t0 = 0;
tend = 25;
h = 10^-4;

%Part2
[x, y, t] = brusselator(x0, y0, t0, tend, h, A, B);

%Part3 IC Inside Limit Cycle
[x2, y2, t2] = brusselator(3, 4, t0, tend, h, A, B);

%Part5 Order



 tiledlayout(4,1)

%X,Y against T plot
 nexttile
 plot(t, x, t,y)
 title('x and y as a function of t')
 xlabel('t')
 legend('x', 'y')

%X,Y Phase Plot
 nexttile
 plot(x,y, x2, y2)
 legend('outside limit cycle', 'inside limit cycle')
 title('phase plane of x and y')
 xlabel('x')
 ylabel('y')

%Order Plot
 nexttile

 h_vals = logspace(log10(2^-14), log10(2^-3), 12);

 x_vals = zeros(11,1);
 y_vals = zeros(11,1);
 for i = 1:12
     [x, y, t] = brusselator(x0, y0, t0, tend, h_vals(i), A, B);
     x_vals(i) = x(end);
     y_vals(i) = y(end);
end
```

```
error_vals = (diff(x_vals).^2 + diff(y_vals).^2).^0.5;
loglog(h_vals(2:end), abs(error_vals));
m = polyfit(transpose(log(h_vals(2:end))), log(abs(error_vals)), 1)
title("gradient = "+string(m(1)))
xlabel('h')
ylabel('error')

%Stability Plot

nexttile
h_vals = logspace(-3, 0, 100);
tend = 1;

x_vals = zeros(100,1);
y_vals = zeros(100,1);
for i = 1:100
    [x, y, t] = brusselator(x0, y0, t0, tend, h_vals(i), A, B);
    x_vals(i) = x(end);
    y_vals(i) = y(end);
end
loglog(h_vals, abs(x_vals))
xlabel('h')
ylabel('x(1)')
title("Solution at t = 1, to show stability")



function f = f(x,y,t, A, B)
    f = A - B*x + y*x^2 - x;
end

function g = g(x,y,t,A,B)
    g = B*x - y*x^2;
end

function [x, y, t] = brusselator(x0, y0, t0, t1, h, A, B)

    N = cast((t1-t0)/h, 'uint64')+1


    t = linspace(t0,t1,N);


    x = zeros(N,1);
    y = zeros(N,1);
```

```
    x(1) = x0;
    y(1) = y0;

    for i = 2:N
        [x(i), y(i)] = rungeKutta4(x(i-1), y(i-1), t(i-1), h, A, B);
    end
end

function [x_next, y_next] = rungeKutta4(x, y, t,h, A, B)

    k1 = f(x,y,t,A,B);
    l1 = g(x,y,t,A,B);

    k2 = f(x + k1*h/2, y + l1*h/2, t + h/2, A, B);
    l2 = g(x + k1*h/2, y + l1*h/2, t + h/2, A, B);

    k3 = f(x + k2*h/2, y + l2*h/2, t + h/2, A, B);
    l3 = g(x + k2*h/2, y + l2*h/2, t + h/2, A, B);

    k4 = f(x + k3*h, y+l3*h, t+h, A, B);
    l4 = g(x + k3*h, y+l3*h, t+h, A, B);

    x_next = x + (k1+2*k2+2*k3+k4)*h/6;
    y_next = y + (l1+2*l2+2*l3+l4)*h/6;

end
```

The first two graphs display the limit cycle for a time interval of 0 to 25 seconds. This was chosen to clearly show the oscillations when the solutions reach the limit cycle.

3 What is your conclusion with regard to the global 'attractiveness' of the limit cycle?

By taking initial conditions outside and inside the limit cycle. We see both solutions converge to the limit cycle. It could be shown further by taking even more initial conditions, but for clarity of the graph we conclude the limit cycle is globally attractive.

4 Would you describe the method as stable?

The method is conditionally stable. This is demonstrated by the bottom graph of the figure above. We see as we vary h and calculate x at $t = 1$. We can see that for values of h near 1, the solution explodes but otherwise converges to the solution.

5 Can you design a procedure to verify 4th order?

The procedure is demonstrated in the code but can be summarised as follows:

- Calculate the solution for fixed initial conditions and time interval and doubling time step.
- Take the solution at the final time step, and calculate the absolute difference between that and the next biggest time step (by using the "diff" method in MatLab).
- plot the log of the time step against log of the difference (error) and calculate the gradient.

From the third picture of the graph we can see we get a gradient of approximately 4 which illustrates the scheme is of order 4. MatLab's Polyfit function was used to calculate the gradient

6 Investigate as far as you can the behaviour of the Brusselator system for other choices of the parameters A and B

Initial investigations should start around the fixed point of $x = A$ and $x = \frac{B}{A}$. To study the behaviour of the fixed point, first the Jacobean of the system should be calculated.

$$J(x, y) = \begin{bmatrix} -B - 1 + 2xy & x^2 \\ B - 2xy & -x^2 \end{bmatrix}$$

$$J\left(A, \frac{B}{A}\right) = \begin{bmatrix} B - 1 & A^2 \\ -B & -A2 \end{bmatrix}$$

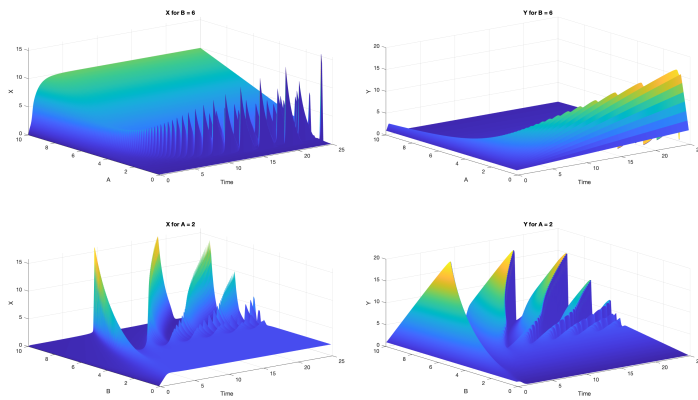
which has $\det\left(J(A, \frac{B}{A}\right) = A^2$ and $Tr\left(J(A, \frac{B}{A}\right) = -A^2 + B - 1$ so has eigenvalue

$$\lambda_{1,2} = \frac{-\left(-A^2 + B - 1\right) \pm \sqrt{\left(-A^2 + B - 1\right)^2 - 4A^2}}{2}$$

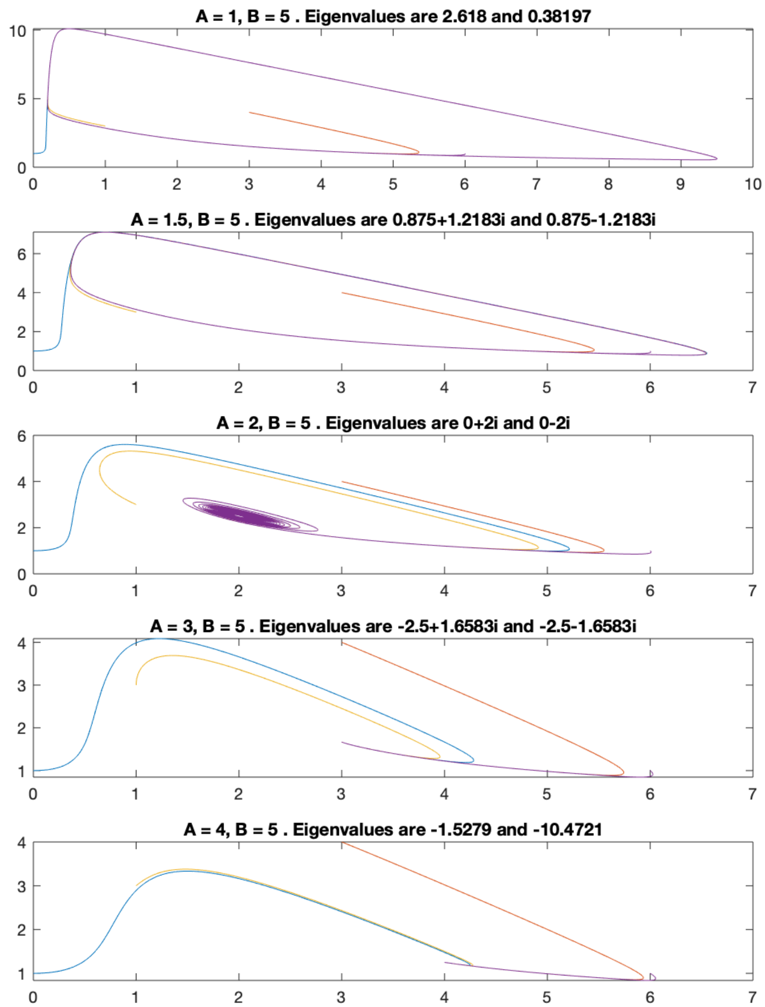
As the determinant is strictly positive, we can assume that the both eigenvalues are of the same sign. ($\det(J) = \lambda_1 \cdot \lambda_2$)

And we can also see for when there is complex eigenvalues, for the fixed point to be stable we require that $B < 1 + A^2$.

Below are graphs of X and Y for either fixed A or B and varying the other. This agrees with the previous result as we can see for $B = 6$ we have oscillations for $A < \sqrt{5}$ and convergence to the fixed point for values above. There is similar behaviour for $A = 2$ and $B > 5$.



It's also worth further demonstrating this with a phase plot, showing the eigenvalues of the Jacobean when they are positive or negative, complex or real or completely imaginary. This further illustrates the behaviour that occurs and the existence of limit cycles and stable points and we can see that this is quite clearly a Hopf Bifurcation at the point where $A = B^2 + 1$.



We can see that we go from an unstable spiral point surrounded by a stable limit cycle, to a centre and then to a stable spiral (focus) as A increases.