MATH0086 Exercise 1

Nikolay Walters

The Brusselator

1 Introduction

This report is a summary of my attempts to solve the Brusselator model using numerical methods and to investigate its properties. The Brusselator model is described by the following differential equations:

$$\frac{dx}{dt} = A - Bx + x^2y - x \tag{1}$$

$$\frac{dy}{dt} = Bx - x^2y \tag{2}$$

with positive constants A and B.

2 Part A

2.1 Fourth-Order Accurate Runge-Kutta

For a general problem given by $\dot{y} = f(t, y), \quad y(t_0) = y_0$, where y is the unknown function of t, $\dot{y}$ is a time derivative of y and t_0 and y_0 are the initial conditions, the fourth-order Runge-Kutta (RK4) method takes the following steps:

- $k_1 = hf(t_n, y_n)$
- $k_2 = hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2})$
- $k_3 = hf(t_n + \frac{h}{2}, y_n + \frac{k_2}{2})$
- $k_4 = hf(t_n + h, y_n + k_3)$
- $y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4$
- $t_{n+1} = t_n + h$

In the above equations, $t_n = nh$ with a time step h for $n = 1, 2, 3\ldots$ on a uniform grid. The RK4 approximation of $y(t_{n+1})$ is y_{n+1}. Such methodology is widely used, and can be found in Hairer et al. (1989).

To extend this algorithm to a system of two equations, thus making it applicable to the Brusselator we must take great care with the dependencies. One way to generalize such an algorithm for a system of equations can be achieved by rewriting it in the vector form. Let us define RHS of (1) as $f(t, x, y)$ and RHS of (2) as $g(t, x, y)$. The k_n increments now becomes a vector with, in our case, two components (k_n and l_n). Based on our model, $f(t, x, y)$ will be associated with k increments and $g(t, x, y)$ with l. This results in the following algorithm:

- $k_1 = hf(t_n, x_n, y_n)$
- $l_1 = hg(t_n, x_n, y_n)$
- $k_2 = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_1, y_n + \frac{1}{2}l_1)$
- $l_2 = hg(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_1, y_n + \frac{1}{2}l_1)$
- $k_3 = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_2, y_n + \frac{1}{2}l_2)$
- $l_3 = hg(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_2, y_n + \frac{1}{2}l_2)$
- $k_4 = hf(t_n + h, x_n + k_3, y_n + l_3)$
- $l_4 = hg(t_n + h, x_n + k_3, y_n + l_3)$
- $x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
- $y_{n+1} = y_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4)$

2.2 Matlab Code

Code shown below uses the methodology outlined in the previous section to find an approximation for the Brusselator model. It also plots $f(t, x, y)$ and $g(t, x, y,)$ for a specified time range, t. The initial conditions used are: $x_0 = 0$ and $y_0 = 1$. Constant A is 2 and constant B is 6. Time step h is equal to 0.01 and t range is from 0 to 25.

```
clear all
tmax = 25;  % t range
step = 0.01; % h value
tsol = 0:step:tmax; % t array
ysol = zeros(1, length(tsol)); % y array
xsol = zeros(1, length(tsol)); % x array
ysol(1) = 1; % initial y
xsol(1) = 0; % initial x
Aconst = 2; % A constant
Bconst = 6; % B constant
fxyt = @(tsol, xsol, ysol) Aconst - (Bconst * xsol) + xsol^2 * ysol - xsol; % f(t,x,y)
gxyt = @(tsol, xsol, ysol) Bconst * xsol - xsol^2 * ysol; % g(t,x,y)
for i = 1:(length(tsol)-1) % RK4
    k0 = fxyt(tsol(i), xsol(i), ysol(i));
    l0 = gxyt(tsol(i), xsol(i), ysol(i));
    k1 = fxyt(tsol(i)+0.5*step,xsol(i)+0.5*step*k0,ysol(i)+0.5*step*l0);
    l1 = gxyt(tsol(i)+0.5*step,xsol(i)+0.5*step*k0,ysol(i)+0.5*step*l0);
    k2 = fxyt((tsol(i)+0.5*step),(xsol(i)+0.5*step*k1),(ysol(i)+0.5*step*l1));
    l2 = gxyt((tsol(i)+0.5*step),(xsol(i)+0.5*step*k1),(ysol(i)+0.5*step*l1));
    k3 = fxyt((tsol(i)+step),(xsol(i)+k2*step),(ysol(i)+l2*step));
    l3 = gxyt((tsol(i)+step),(xsol(i)+k2*step),(ysol(i)+l2*step));
    xsol(i+1) = xsol(i) + (1/6)*step*(k0 + 2*k1 + 2*k2 + k3); % x_{n+1}
    ysol(i+1) = ysol(i) + (1/6)*step*(l0 + 2*l1 + 2*l2 + l3); % y_{n+1}
end
hold on
title('RK4 approximation of the Brusselator between t = 0 and 25');
xlabel('t');
```

```
ylabel('Approximation');
plot(tsol, xsol,'DisplayName', 'x');
plot(tsol, ysol, 'DisplayName', 'y');
legend('show')
hold off
```

The output of this code is a single graph showing $x(t)$ and $y(t)$: Plotting

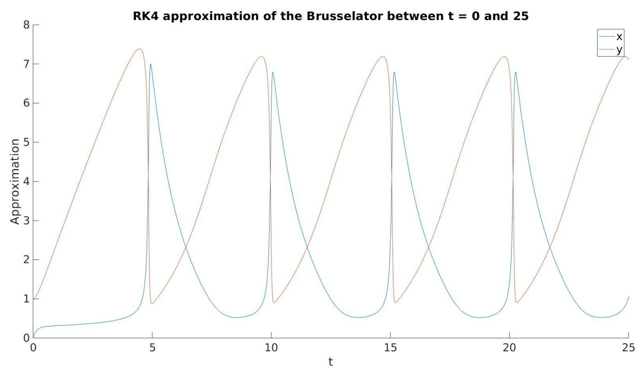

Fig. 1. $x(t)$ and $y(t)$ with $A = 2$, $B = 6$, $h = 0.01$ and $0 \leq t \leq 25$.

$x(t)$ against $y(t)$ for the same t range and parameters as before results in the following plot: The t range employed in these diagrams is deemed acceptable

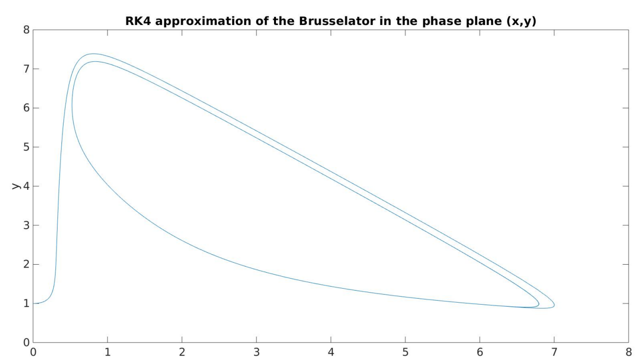

Fig. 2. $x(t)$ against $y(t)$ with $A = 2$, $B = 6$, $h = 0.01$ and $0 \leq t \leq 25$.

for the following reasons. Firstly, Fig. 1 shows at least 3 full periods for the functions. Thus, it can be confirmed that they do indeed show repeating pattern with a time period (p) of approximately $5t$ units. Secondly, in Fig. 2 a clear convergence can be seen very early on. A limit cycle is approached to a good accuracy at around $t = 11$ or 2 periods.

At step of $h = 0.1$ local instability can be seen in both $x(t)$ and $y(t)$ at approximately $t = 5 + np$ for $n = 0, 1, 2, \ldots$ where the rate of change of $x(t)$ and $y(t)$ appears to be the greatest. These can be seen in the figure below.

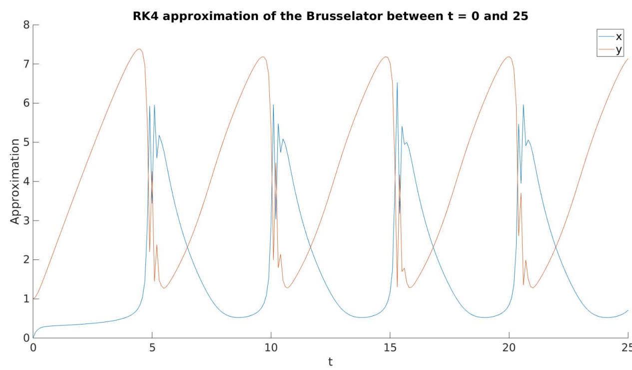

Fig. 3. $x(t)$ and $y(t)$ with $A = 2$, $B = 6$, $h = 0.1$, $0 \leq t \leq 25$ and local ODE dependant instability.

At step size of $h = 0.02$ the local instability becomes impossible to see on the $x(t)$ and $y(t)$ against t plot but can still be distinguished in the phase space diagram. Fig. 4 shows both diagrams. Therefore, $h = 0.001$ is deemed an acceptable choice, since no local instabilities can be seen while the step size remains relatively large for computational efficiency.

These step sizes are of course only applicable to the range of the parameters specified earlier. We could render local instability with $h = 0.001$ by, for example, setting the value of B to 9 while keeping the other parameters as before, as shown in Fig. 6. Finally, larger step sizes, such as $h = 0.25$ are unstable due to finite-difference approximations itself, rather than the nature of the ODEs, as was the case with the local instability. Such approximations reach maximum numerical values that can be stored by a variable extremely quickly and therefore provide little practical application.

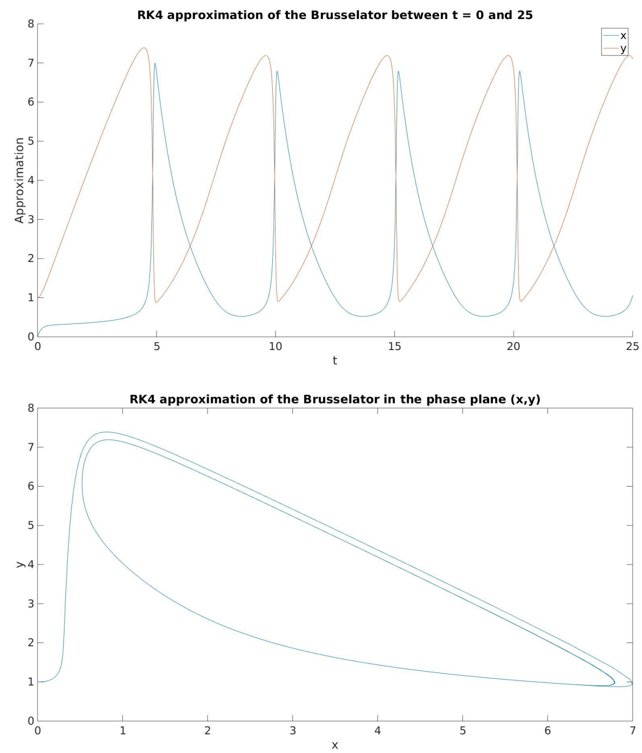

Fig. 4. Top: $x(t)$ and $y(t)$ with $A = 2$, $B = 6$, $h = 0.02$, $0 \leq t \leq 25$ and an unnoticeable instability. Bottom: $x(t)$ against $y(t)$ with $A = 2$, $B = 6$, $h = 0.02$, $0 \leq t \leq 25$ and a noticeable instability in the bottom right quadrant.

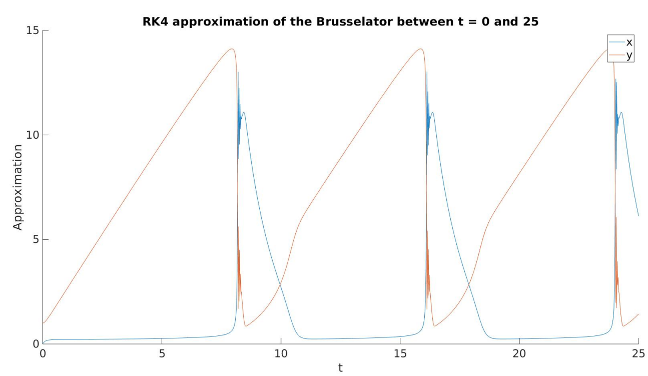

Fig. 5. $x(t)$ and $y(t)$ with $A = 2$, $B = 9$, $h = 0.01$, $0 \leq t \leq 25$ and an local instability.

2.3 Limit Cycle as a Global Attractor

Taking a number of initial x_0 and y_0 conditions in the first quadrant demonstrates that the limit cycle is a global attractor as shown in Fig. 6. There are cases when the initial conditions are so far off the limit cycle that the solution diverges rapidly. An example of the phase space where the solution just manages to converge on the limit cycle is shown in Fig. 7.

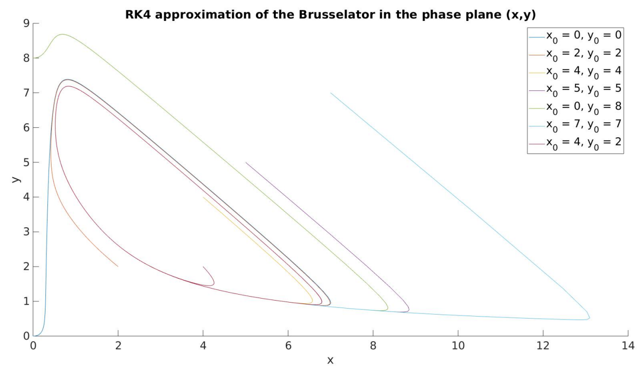

Fig. 6. A number of initial conditions resulting in a convergence at the limit cycle with $A = 2$, $B = 6$, $h = 0.01$ and $0 \leq t \leq 25$.

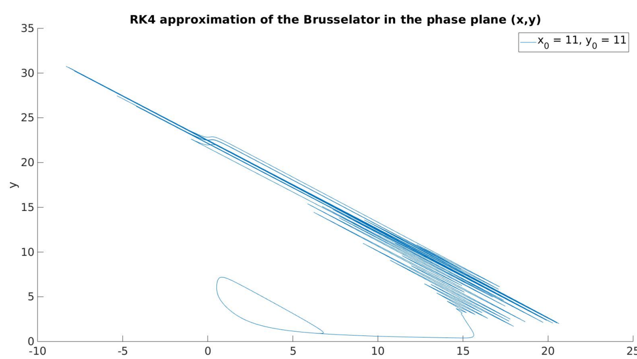

Fig. 7. x_0 and y_0 both set to 11 result in oscillatory behaviour before converging on the limit cycle with $A = 2$, $B = 6$, $h = 0.01$ and $0 \leq t \leq 25$.

2.4 Stability

2.5 Accuracy of the Solution

3 Part B

4 Credits

This section describes the contribution of each authors as well as a general note of gratitude for the overseeing lecturer, **Prof. Raphael C.-W. Phan** whom enlightened the authors with the concepts of computer and network security.

4.1 Author 1 - Chia Jason

1. Literature Review - Parameter Tampering
2. URL analysis
3. Registration analysis
4. Scan Attack
5. Programmer for Sleepin
6. Suggestions to strengthening against (TIM)
7. Rationale of QR attendance system
8. Alternative to QR attendance system
9. Conclusion

4.2 Author 2 - Hor Sui Lyn

1. Abstract
2. Introduction
3. Literature Review - Cross Site Request Forgery
4. Literature Review - URL Manipulation
5. Attack methodology
6. Implementation and Results
7. Suggestions to strengthening against (DEP)
8. Additional Suggestions
9. Proof Reading and Formatting

References