MATH0086 Exercise 1: The Brusselator

Kieran Hobden

November 1, 2019

Abstract

Investigating autocatalytic reactions, we model the Brusselator, a system of 2-dimensional, non-linear differential equations in time. We use the finite-difference, fourth-order accurate Runge-Kutta (RK4) approximation to observe globally attractive limit cycles for given parameters and the subsequent changes to the phase portrait as the parameters are varied. We justify our computations; verifying the stability and accuracy of our numerical solutions, and observing the effects that a poorly implemented method might have on obtaining a physical solution. Further investigation is undertaken to observe new behaviour in the system for different initial parameters.

Keywords— finite-difference, Brusselator, autocatalytic, RK4

Contents

1 Introduction

This paper contains assessed coursework for the Computational and Simulation Methods module in the Mathematical Modelling MSc at UCL. We focus on the computational investigation starting from the governing equations of certain autocatalytic reactions; the Brusselator. As such, no derivation of the governing equations are provided.

The chemical concentration of two reactants is given by the 2-dimensional, non-linear, ordinary differential equations:

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

$$\frac{dy}{dt} = g(x, y) = Bx - x^2 y \tag{2}$$

We begin our model by finding the only equilibrium point $x_{eq} = A$, $y_{eq} = B/A$, representing the time-independent state of the concentrations of the reactants. We do not shift the equilibrium via the transformation $x \to x - x_{eq}$, $y \to y - y_{eq}$ so subsequent plots have a simple and intuitive interpretation. This transformation is included only in our code to produce a video of the phase space.

2 Main Tasks - Part A

2.1 RK4 Method

We define a uniform grid $t_n = nh$ on which we approximate the Brusselator equations using the finite-difference, fourth-order accurate Runge-Kutta (RK4) method. The method is not derived here. The initial value problem is specified with (1), (2) and the initial conditions: $x(0) = x_0$, $y(0) = y_0$.

We then evaluate $x_n = x(t_n)$ at each t_n using the RK4 method below:

$$t_{n+1} = t_n + h$$

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3}$$

Where k_1, k_2, k_3 and k_4 are computed as:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{k_1}{2}, y_n + \frac{k_1}{2})$$

$$k_3 = hf(x_n + \frac{k_2}{2}, y_n + \frac{k_2}{2})$$

$$k_4 = hf(x_n + k_3, y_n + k_3)$$

Similarly, we compute y_{n+1} with:

$$y_{n+1} = y_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \tag{4}$$

Where l_1, l_2, l_3 and l_4 are computed as:

$$l_1 = hg(x_n, y_n)$$

$$l_2 = hg(x_n + \frac{l_1}{2}, y_n + \frac{l_1}{2})$$

$$l_3 = hg(x_n + \frac{l_2}{2}, y_n + \frac{l_2}{2})$$

$$l_4 = hg(x_n + l_3, y_n + l_3)$$

2.2 Finding the Limit Cycle

In computing the RK4 method for $A = 2$, $B = 6$, $x_0 = 0$ and $y_0 = 1$, we observe a limit cycle around our equilibrium, $(x_{eq}, y_{eq}) = (2, 3)$ as shown below.

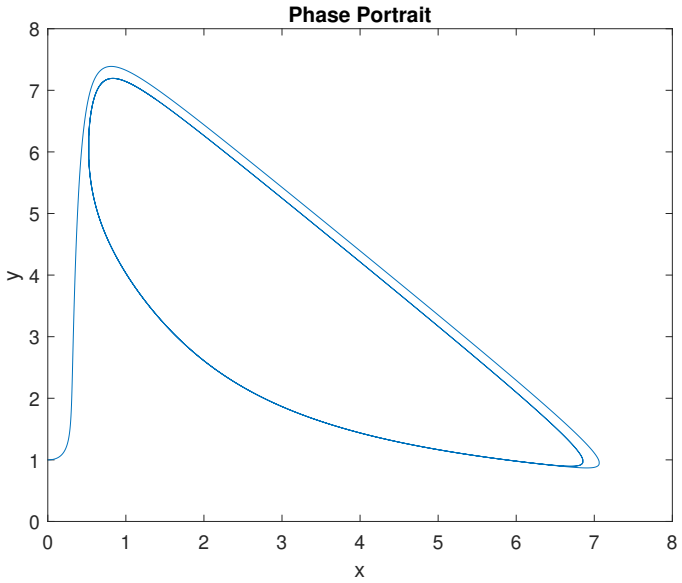
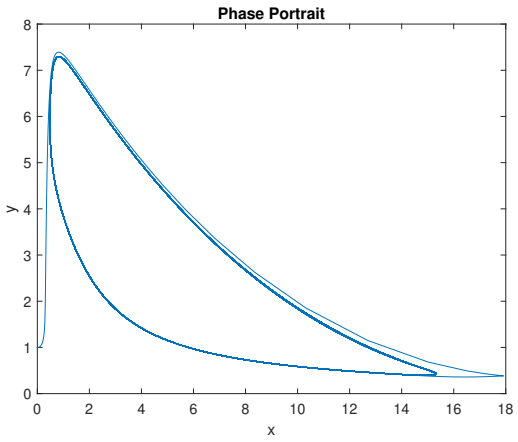


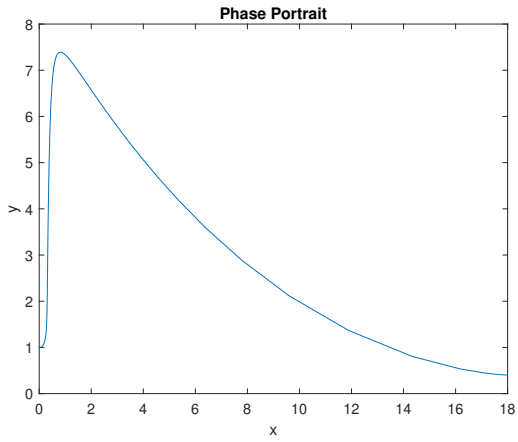
Figure 1: The phase space limit cycle for $A = 2$ and $B = 6$.

A suitable time limit ($t_0 = 0$, $t_f = 100$) and time step ($h = 0.005$) value was required to obtain the limit cycle. We consider a large time step as equivalent to a small time limit for a fixed number of steps as the time step is calculated from the inverse of the time limit. As such, for a very large time step, the limit cycle is not observed as insufficient steps are taken to be sensitive enough to the changing functions f and g. For a very small time step, rounding errors introduced in the computation again prevents the limit cycle from being found and the solution tends to infinity.

The figure below shows an intermediate case where the time step is nearly too large to observe the limit cycle. We can see in the bottom-right corner of the graph that the trajectory tends to a point as the time step is increased. This indicates where the computation fails. In the subsequent graph, we see the trajectory continues beyond the limit cycle towards infinity.



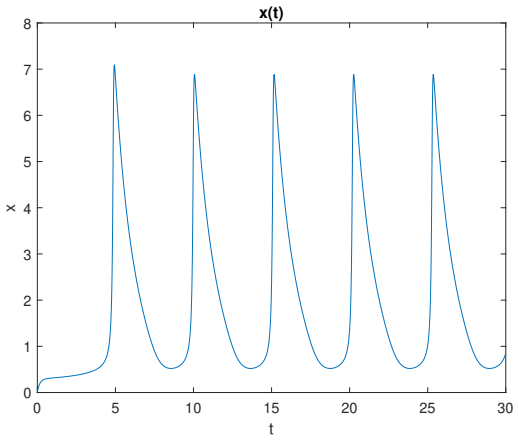
(a) Large time step



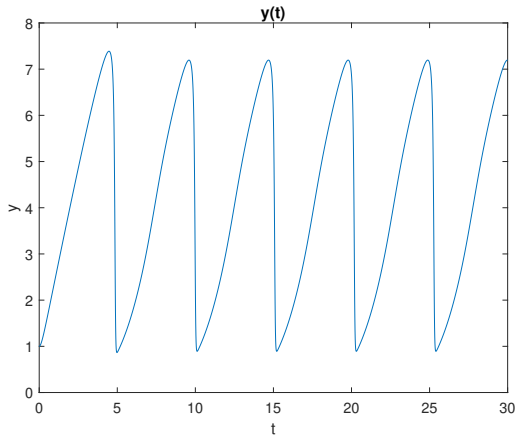
(b) Too large time step

Figure 2: The time-periodic limit cycle in phase space is only for suitable time steps and time limits. Here we see the effect of a time step that is too large or an equivalent time limit that is too small.

In the above limit cycle, we see the concentrations of reactants changes simultaneously within certain limits. To investigate the behaviour of the individual reactants, we plot them separately to observe their minimum and maximum values.



(a) $x(t)$



(b) $y(t)$

Figure 3: The time-periodic reactant concentrations as a function of time as the phase space trajectory tends towards the limit cycle.

2.3 Global 'Attractiveness'

We now wish to investigate if all trajectories originating in the positive quadrant of the phase space ($x_0 > 0$, $y_0 > 0$) tend to the limit cycle as $t \to \infty$. This asymptotic behaviour defines

whether the limit cycle is globally attractive. To investigate the global 'attractiveness' of the limit cycle, we plot the trajectories for multiple initial values.

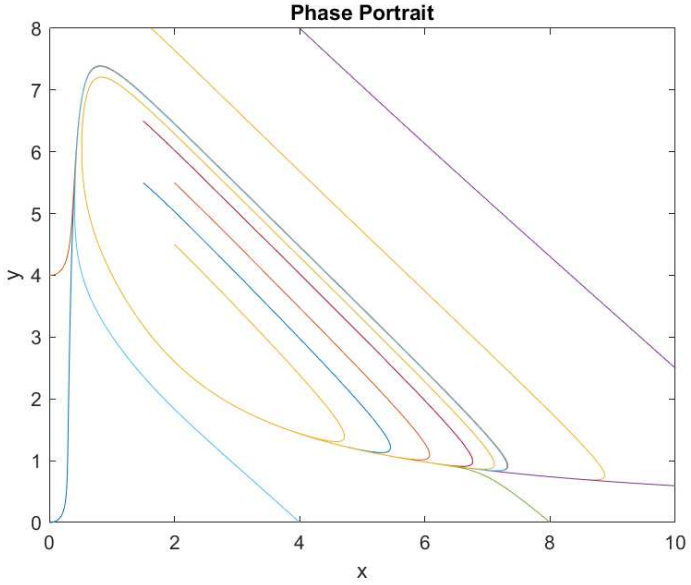


Figure 4: The global 'attractiveness' of the limit cycle.

Clearly, we see that all trajectories starting in the first quadrant tend asymptotically towards the limit cycle. This shows the limit is globally attractive.

2.4 Stability

The method is stable for a specific range of values for the time step, h, where the stable limit cycle is obtained. For too large values of h, the limit cycle is unattainable and similarly, too small values of h prevent the stable cycle from being obtained (Fig. 2) but this is due to rounding effects in the computation, as described in Section 2.2, and is not inherent to the method. The method is thus conditionally stable.

2.5 Accuracy

The accuracy of the RK4 method is known to be fourth-order in h, we now attempt to show our computation has preserved this accuracy. To do this, we assume the relation $y_n - \tilde{y_n} \leq C\mathcal{O}(h^n)$ where y_n and $\tilde{y_n}$ represent our numerical and exact solutions respectively. We notice that by assuming equality and running our approximation for h and $\frac{h}{2}$ and subtracting one equation from the other produces the equation below.

$$y_h - y_{\frac{h}{2}} = C(1 - \frac{1}{2^n})h^n$$

Taking the logarithm allows us to determine order of the method from the gradient from a $log|y_h - y_{\frac{h}{2}}|$ against h plot as shown in the equation below.

$$log|y_h - y_{\frac{h}{2}}| = nlog(h) + const \tag{5}$$

In the plots below, we see the gradient is 3.9997, corresponding to the order of the approximation which was expected to be fourth order.

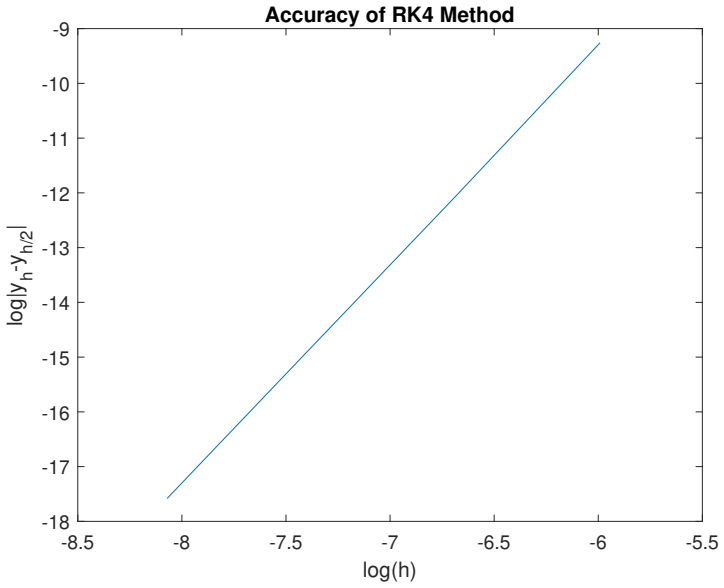


Figure 5: The RK4 method is shown to be fourth order from the gradient of 3.9997 in the log-log plot.

3 Further Investigation - Part B

To investigate the system further, we determine the Jacobian, J, of the system and evaluate it at the equilibrium.

$$J_{eq} = \begin{pmatrix} -B + 2xy - 1 & x^2 \\ B - 2xy & -x^2 \end{pmatrix} \Bigg|_{x=x_{eq}, y=y_{eq}} = \begin{pmatrix} B - 1 & A^2 \\ -B & -A^2 \end{pmatrix} \tag{6}$$

Through computing the trace ($Tr(J_{eq}) = B - 1 - A^2$) and determinant ($Det(J_{eq}) = A^2$) of the matrix, we observe that a stable trajectory, corresponding to $T < 0$ and $D > 0$ is expected for $A^2 > B - 1$.

Clearly, our initial values, $A = 2$ and $B = 6$, do not satisfy this condition and so the equilibrium point is unstable. This may seem counter-intuitive as we have described our limit cycle as stable. However, as the equilibrium point lies inside the limit cycle, all trajectories with initial conditions near the point must diverge as they tend asymptotically towards the limit cycle hence the equilibrium point must be unstable.

We check our bound for instability ($A^2 < B - 1$) by producing the phase portrait for $A = 2$, $B = 10$.

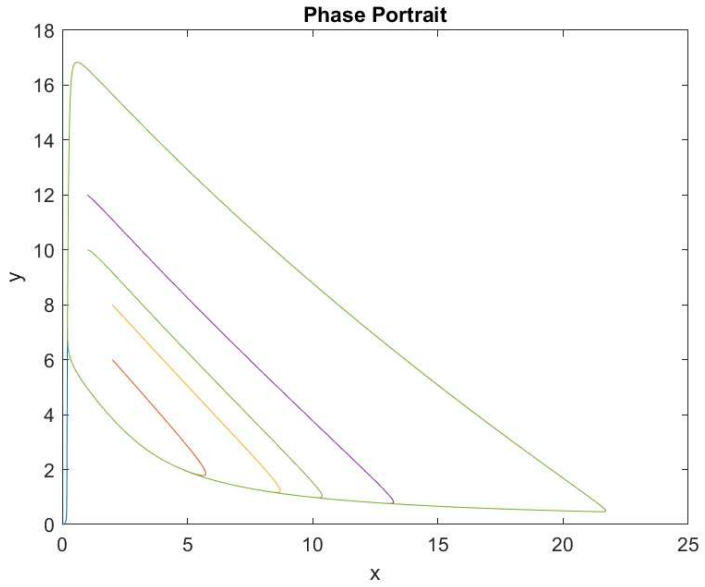
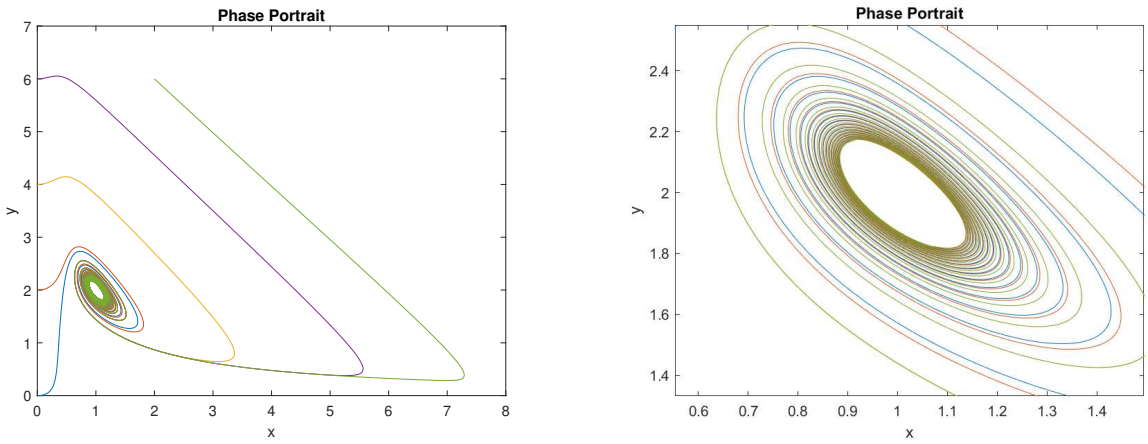


Figure 6: The globally attractive limit cycle for $A = 2$ and $B = 10$.

Again, we see instability inside the limit cycle showing the unstable equilibrium point. From a video, of the phase portrait as the parameters A and B change, we see the limit cycle persists for all $A^2 > B - 1$.

Next we consider the case $A^2 = B - 1$ with values $A = 1$ and $B = 2$. A linearisation suggests we should expect a centre put the Hartman-Grobman theorem shows that this linearisation is not a valid procedure for a non-hyperbolic equilibrium. We must compute the solution to progress.



(a) Focus (b) Smaller section of focus plot

Figure 7: The globally attractive focus for $A = 1$ and $B = 2$.

We observe a globally attractive focus at the equilibrium point. Whilst we do not expect a limit cycle, we produce a new plot below, with initial values near to the equilibrium point. Zooming in to a smaller section of the plot shows the same focus. This scale invariance shows the point is attractive and the phase portrait shows a focus not limit cycle.

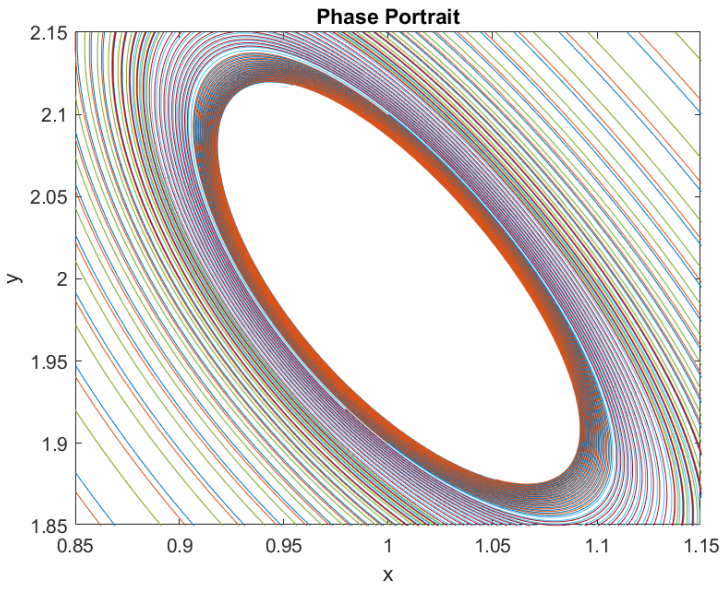


Figure 8: A further smaller section of the focus plot.

Finally, we consider the case $A^2 > B - 1$, where we expect a stable focus, noode or inflected node. This follows from further work in non-linear systems and the application of the Hartman-Grobman theorem, not shown here. We start by observing the inflected node for $A = 2$ and $B = 1$ and note that for larger A we should expect a node yet continue to only observe the inflected node.

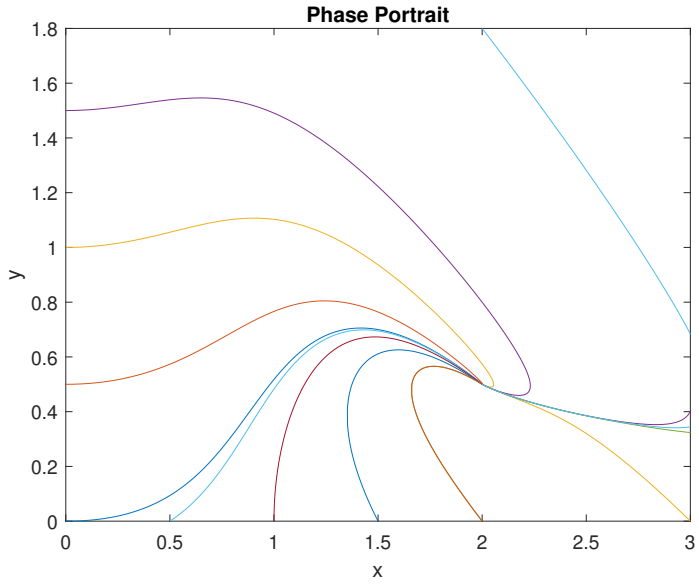


Figure 9: Inflected node

Next, we see the stable focus with $A = 1$ and $B = 1$. The focus persists for all values where $(Tr(J))^2 < 4(Det(J))$ and $A^2 > B - 1$.

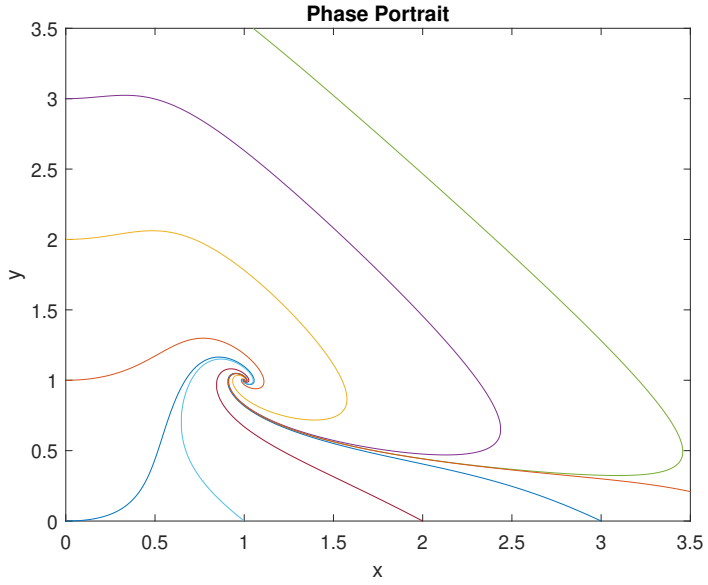


Figure 10: Focus for $A = 1$ and $B = 1$

4 Discussion

We have shown a stable limit cycle can be reached for a range of A and B values, which corresponds to reactant concentrations that oscillate periodically in time. The individual reactant concentrations thus have a well defined minimum and maximum concentration which, if the reaction is studied in the context of pharmacokinetics, should lie below the toxic limit and above the minimum dose.

Let us now consider whether using a higher order method would uncover more detail about the phase portrait. Firstly, we note that the system only has one equilibrium (the limit cycle for a specific range of A and B values) so we do not expect a higher order method to uncover another. But might the limit cycle shape be more precise for a higher order method? No, the limit cycle is globally, asymptotically stable, and so we can make the assumption that finding the limit cycle is sufficient to fully describe the stable system as we can increase the the time limits or decrease the time step to more accurately find the shape of the limit cycle.

Although our investigation is purely computational and the underlying physical intricacies of the autocatalytic reactions has been largely ignored, it is worth, for the sake of physical intuition, considering a simple case where the reaction might occur. The Briggs-Rauscher reaction where a colourless solution of two reactants slowly turns an amber colour and then suddenly changes to a dark blue before fading back to being colourless. This process repeats around ten times and corresponds to a limit cycle such as ours.

5 Conclusion

The Brusselator model shows a globally attractive, stable limit cycle for $B < 1 + A^2$, corresponding to time-periodic oscillations in the reactant concentrations. In this case, the physical system is described as a clock reaction and corresponds to the equilibrium point being unstable.

When the equilibrium point is stable, we observe either a focus or an inflected node. Although the reaction path is different, the final concentrations of the reactants is the same and is as defined at the equilibrium point. For the focus, we observe oscillatory behaviour that may be visible as a decaying clock reaction. In Chemistry, this is the actual definition of a clock reaction as all oscillatory systems eventually come to rest.

No globally unstable phase portrait corresponding to reactant concentrations tending to infinity were observed due to the presence of a limit cycle for all parameters leading to an unstable equilibrium point. This would be an unphysical result so it is reassuring it is not produced by our results.

Our computations uncovered the existence of a minimum and maximum time step corresponding to an accumulation of rounding errors and of an inherent instability in the method respectively. For a suitable time step, the RK4 method can be implemented successfully, determining the phase space behaviour to fourth-order accuracy.

Further investigations might include research into the origins and derivation of the Brusselator model or research into the invariant manifold of the inflected node. Perhaps research should be conducted into producing a faster implementation of the RK4 method which is very computationally demanding.

6 Appendix

Here, we attach the MatLab code included in the project. First, the code used to observe the phase space and plots of $x(t)$ and $y(t)$ against time. Next, we include the code used in determining the order of the method. Finally, we include code which produces a video of a saturated phase space as one of A or B is cycled through a range of values. This code, although impossible to include in this paper, was invaluable in our further investigation. One file, used to determine the global 'attractiveness' of the phase space was omitted as it is a trivial extension of the code used to determine one trajectory by running a for loop over the previous method.

The following code is used to plot the trajectory from one initial condition for a given A and B.

```
%Coursework - Single Trajectory
%Kieran Hobden
%29-Oct-'19

clear all

%Solve the equations dx/dt = f(x,y) and dy/dt = g(x,y)
%Plot a single trajectory in the phase portrait
%Plot x(t) and y(t)

%Initialise variables
A = 2;
B = 10;
t0 = 0; tf = 50;
x(1) = 0; y(1) = 1;
N = 100000;
h = (tf-t0)/N;

%Define f as [dx/dt, dy/dt]
f = @(x,y) [A - B*x + (x^2)*y - x, B*x - (x^2)*y];

for i=1:N
    k1=h*f(x(i),y(i));
    k2=h*f(x(i)+k1(1)/2,y(i)+k1(2)/2);
    k3=h*f(x(i)+k2(1)/2,y(i)+k2(2)/2);
    k4=h*f(x(i)+k3(1),y(i)+k3(2));

    k = (k1+2*k2+2*k3+k4)/6;

    x(i+1) = x(i) + k(1);
    y(i+1) = y(i) + k(2);
```

```
end

%Plot the phase portrait
plot(x,y)
title('Phase Portrait')
xlabel('x'); ylabel('y')
%xlim([0 18])
%ylim([0 8])
pause(2)

%Plot x(t) and y(t)
plot(linspace(t0,tf,N+1),x)
title('x(t)')
xlabel('t'); ylabel('x')
pause(2)

plot(linspace(t0,tf,N+1),y)
title('y(t)')
xlabel('t'); ylabel('y')
```

The following code is used to determine the accuracy of the RK4 method by producing the plots seen in Section 2.5.

```
%Coursework - Accuracy
%Kieran Hobden
%29-Oct-'19

clear all

%Solve the equations f(x,y) = [dx/dt(x,y), dy/dt(x,y)]
%Verify the accuracy of the RK4 method as 4th order by a loglog
    plot

%Initialise variables
A = 2;
B = 10;
t0 = 0; tf = 50;
x(1) = 0; y(1) = 1;
N = 100000;
h = (tf-t0)/N;

%Define f as [dx/dt, dy/dt]
f = @(x,y) [A - B*x + (x^2)*y - x, B*x - (x^2)*y];

%Produce 5 values for y_n for different h values
```

```
for j=1:5
N = (10^4)*(2^(j-1));
h = (tf-t0)/N;

%Run RK4 method for N iterations
for i=1:N
    k1=h*f(x(i),y(i));
    k2=h*f(x(i)+k1(1)/2,y(i)+k1(2)/2);
    k3=h*f(x(i)+k2(1)/2,y(i)+k2(2)/2);
    k4=h*f(x(i)+k3(1),y(i)+k3(2));

    k = (k1+2*k2+2*k3+k4)/6;

    x(i+1) = x(i) + k(1);
    y(i+1) = y(i) + k(2);
end

%h_store and coord_store stores the final h value and coordinates
h_store(j) = h;
coord_store(j,1) = x(end);
coord_store(j,2) = y(end);

end

%Compute v = |y(h)-y(h/2)|
for i=1:(length(h_store)-1)
    v(i) = log(norm(coord_store(i+1,:)-coord_store(i,:)));
end

%Plot result and take first value outputted as the loglog gradient
plot(log(h_store(2:end)),v)
xlabel('log(h)')
ylabel('log|y_{h}-y_{h/2}|')
title('Accuracy of RK4 Method')
polyfit(log(h_store(2:end)),v,1)
```

The following code produces a video of the phase space for changing A and B and uses ODE45 to solve the equation. Although ODE45 was prohibited for the main task, this was only used as part of the further investigation.

```
%Coursework - Video File
%Kieran Hobden
%29-Oct-'19

clear all
```

```
6
7 %Produce a video of the saturated phase space (too many
      trajectories)
8 %This will allow future investigations
9 %Use ODE45 as this is not a core task
10
11 % ''f'' is set of differential equations
12 % ''a'' is array containing x and y variables
13 % ''t'' is time variable
14 % ''x0'' is the initial conditions array
15
16 %Initialise variables
17 A = 2;
18 num_b = 100;
19 N2 = 25
20 B_array = linspace(2, 20, num_b);
21
22 for x = 1:N2
23
24 B = B_array(x)
25
26 %Run RK4 method
27 for i=1:num_b
28     f = @(t,a) [(B-1)*a(1)+(A^2)*a(2)+(B/A)*(a(1)^2)+2*A*a(1)*a(2)
          +(a(1)^2)*a(2);...
29         -B*a(1)-(A^2)*a(2)-(B/A)*(a(1)^2)-2*A*a(1)*a(2)-(a(1)^2)*a
              (2)];
30     x0 = 10*(rand(1,2)-0.5)
31     [t,a] = ode45(f,[0 50],x0);
32     plot(a(:,1),a(:,2)), hold on
33 end
34
35 %Plot saturated phase space
36 title('Coursework - Phase Portrait')
37 xlabel('x'), ylabel('y')
38 xlim([-10 10]), ylim([-10 10])
39 hold off
40
41 %Save 10*no. of frames so the video is slow enough to observe
42 for j=0:9
43     F(10*x-j)=getframe;
44 end
45 end
46
47 %Compile the video as a .avi file
```

```
writerObj = VideoWriter('test.avi')
open(writerObj);
writeVideo(writerObj, F)
close(writerObj);
```