Modelling and Numerics (MA261) Assignment 3 Tom Dalby U1903831

1.1. $H(y) = c \cdot y$ ~~so~~ for some const $c \in \mathbb{R}^m$, as H is a linear invariant, so it follows $\nabla H(y) = c$.

Thus, $\forall y \in \mathbb{R}^m$, $c \cdot f(y) = \nabla H(y) \cdot f(y) = 0$.

Take any $n \in \mathbb{N} \cup \{0\}$.

$$H(y_{n+1}) = c \cdot y_{n+1} = c \cdot \left(y_n + h \sum_{i=1}^{m} \gamma_i k_i(t_n, y_n; h)\right)$$

$$= c \cdot y_n + c \cdot \left(h \sum_{i=1}^{m} \gamma_i k_i(t_n, y_n; h)\right)$$

$$= H(y_n) + h \sum_{i=1}^{m} c \cdot \gamma_i k_i(t_n, y_n; h)$$

$$= H(y_n) + h \sum_{i=1}^{m} \gamma_i \left(c \cdot f\left(t_n + \alpha_i h, y_n + h \sum_{l=1}^{i-1} \beta_{il} k_l(t_n, y_n; h)\right)\right)$$

$$= H(y_n) \quad \text{as } c \cdot f\left(t_n + \alpha_i h, y_n + h \sum_{l=1}^{i-1} \beta_{il} k_l(t_n, y_n; h)\right)^{*} = 0$$

$\forall 1 \le i \le m$, so the sum vanishes.

This is true for any $n \in \mathbb{N}$ so for all n $H(y_n) = H(y_0)$.

* Note f does not depend on the time argument so this dot product is really $c \cdot f(y)$ for some $y \in \mathbb{R}^m$ hence is 0 as per the remarks at the top of the page.

1.2. To determine/approximate $H(x_{n+1}, p_{n+1})$ from $H(x_n, p_n)$ we use the Forward Euler method on x_n and p_n to obtain x_{n+1} and p_{n+1}.

Suppose x and p solve the Hamiltonian system, i.e.

$$x'(t) = \partial_p H(x(t), p(t)) = \partial_p (T(p) + V(x)) = T'(p)$$

and

$$p'(t) = -\partial_x H(x(t), p(t)) = -\partial_x (T(p) + V(x)) = -V'(x),$$

Then

$$x_{n+1} = x_n + h x'(t_n) = x_n + h T'(p_n)$$

and

$$p_{n+1} = p_n + h p'(t_n) = \cancel{p_n + hV} \; p_n - h V'(x_n)$$

We can now use ~~the~~ Taylor's ~~expansion~~ Theorem, noting $H \in C^2$ (as $T, V \in C^2$) yielding,

$$H(x_{n+1}, p_{n+1}) = H(x_n + hT'(p_n),\ p_n - hV'(x_n))$$

$$= H(x_n, p_n) + hT'(p_n)\partial_x H(x_n, p_n) - hV'(x_n)\partial_p H(x_n, p_n)$$

$$+ \frac{1}{2}h^2 T'(p_n)^2 \partial_{xx} H(x_n, p_n) - \frac{2}{2}h^2 T'(p_n)V'(x_n)\partial_{xp} H(x_n, p_n)$$

$$+ \frac{1}{2}h^2 V'(x_n)^2 \partial_{pp} H(x_n, p_n) + R(h),$$

where $R(h)$ is a remainder term of order $O(h^3)$ i.e. $\frac{R(h)}{h^2} \to 0$ as $h \to 0$. We can thus ignore it in our method (to obtain an approximation).

As shown before, $\partial_x H(x_n, p_n) = V'(x_n)$ and $\partial_p H(x_n, p_n) = T'(p_n)$, so $hT'(p_n)\partial_x H(x_n, p_n) - hV'(x_n)\partial_p H(x_n, p_n) = 0$.

$$\partial_{xx} H(x(t), p(t)) = \partial_{xx}(T(p) + V(x)) = V''(x).$$
$$\partial_{pp} H(x(t), p(t)) = \partial_{pp}(T(p) + V(x)) = T''(p)$$
$$\partial_{xp} H(x(t), p(t)) = \partial_x(\partial_p H(x(t), p(t)))$$
$$= \partial_x(T'(p)) = 0.$$

Thus,

$$H(x_{n+1}, p_{n+1}) = H(x_n, p_n) + \frac{1}{2}h^2(T'(p_n)^2 V''(x_n) + V'(x_n)^2 T''(p_n)),$$

in our method (i.e. ignoring $R(h)$).

~~So $H(x_{n+1}, p_{n+1}) - H(x_n, p_n) = \frac{1}{2}h^2(T'(p_n)^2 V''(x_n) + V'(x_n)^2 T''(p_n)) \geq 0$ as $V'', T'' \geq 0$ and $T'(p_n)^2, V'(x_n)^2 \geq 0$, and of course $\frac{1}{2}h^2 \geq 0$. I.e. the value of the Hamiltonian increases in each time~~ step.

$$H(x_{n+1}, p_{n+1}) - H(x_n, p_n) = \frac{1}{2}h^2(T'(p_n)^2 V''(x_n) + V'(x_n)^2 T''(p_n)).$$

$\frac{1}{2}h^2 > 0$. $V''(x_n), T''(p_n) > 0$ as given in the question.

~~Since $V'' > 0$ always, this means V'~~ is

Finally, $(T'(p_n))^2, (V'(x_n))^2 \geq 0$ clearly.

Thus, $H(x_{n+1}, p_{n+1}) - H(x_n, p_n) \geq 0$ so the value of the Hamiltonian increases in each time step.

1.3. Letting $\underline{Y}$ be the exact solution, we define the truncation error as follows,

$$\tau_n := \underline{Y}(t_{n+1}) - \underline{Y}(t_n) - a_n \exp(b_n h) - c_n .$$

As $\underline{Y} \in C^3$, we can use Taylor's theorem to get

$$\underline{Y}(t_{n+1}) = \underline{Y}(t_n+h) = \underline{Y}(t_n) + h\underline{Y}'(t_n) + \frac{h^2}{2}\underline{Y}''(t_n) + O(h^3).$$

$\exp \in \cancel{C^3} C^\infty \subset C^3$ so we get using Taylor's theorem, about 0,

$$\exp(b_n h) = 1 + b_n h + \frac{b_n^2 h^2}{2} + O(h^3).$$

So $\tau_n = \underline{Y}(t_n) + h\underline{Y}'(t_n) + \frac{h^2}{2}\underline{Y}''(t_n) + O(h^3) - \underline{Y}(t_n) - a_n - a_n b_n h$

$$- a_n \frac{b_n^2 h^2}{2} - a_n O(h^3) - c_n$$

$$= \cancel{h\underline{Y}'} - a_n - c_n + h\left(\underline{Y}'(t_n) - a_n b_n\right) + \frac{h^2}{2}\left(\underline{Y}''(t_n) - a_n b_n^2\right)$$

$$+ O(h^3).$$

So to make $\tau_n = O(h^3)$ we require:

$-a_n - c_n = 0$ (i.e. $c_n = -a_n$)

$-a_n b_n + \underline{Y}'(t_n) = 0$

$-a_n b_n^2 + \underline{Y}''(t_n) = 0.$

So $\cancel{a_n = \underline{Y}'}$ $\quad a_n b_n b_n = \underline{Y}''(t_n)$

$\underline{Y}'(t_n) b_n = \underline{Y}''(t_n)$ thus $b_n = \dfrac{\underline{Y}''(t_n)}{\underline{Y}'(t_n)}$, well-defined as $\underline{Y}'(t_n) \neq 0$ always.

Then $a_n = \dfrac{\underline{Y}'(t_n)}{b_n} = \dfrac{(\underline{Y}'(t_n))^2}{\underline{Y}''(t_n)}$, well defined as $\underline{Y}''(t_n) \neq 0$ always

And $c_n = -\dfrac{(\underline{Y}'(t_n))^2}{\underline{Y}''(t_n)}$

With these coefficients $\tau_n = O(h^3) = o(h^2)$ so the method has consistency order 2.

We need to tidy up these coefficients. We assume the ODE system is linear. So $f(t, y(t)) = y'(t) = \lambda y(t)$, $\lambda \in \mathbb{C}$ a constant.

So ~~λ~~ $\underline{Y}'(t) = \lambda \underline{Y}(t)$ and $\underline{Y}''(t) = \lambda \underline{Y}'(t) = \lambda^2 \underline{Y}(t)$.
Thus,

$$\cancel{a_n} = \cancel{\lambda^2 y} \frac{(\underline{Y}'(t_n))^2}{\underline{Y}''(t_n)} = \frac{\lambda^2 \underline{Y}(t_n)^2}{\lambda^2 \underline{Y}(t_n)} = \underline{Y}(t_n)$$

$$\frac{\underline{Y}''(t_n)}{\underline{Y}'(t_n)} = \frac{\lambda^2 \underline{Y}(t_n)}{\lambda \underline{Y}(t_n)} = \lambda.$$

Replacing $\underline{Y}(t_n)$ with y_n, we get

$$a_n = y_n \quad b_n = \lambda \quad c_n = -y_n.$$

So,

$$y_{n+1} = y_n + y_n \exp(\lambda h) - y_n = y_n \exp(\lambda h), \ \lambda \in \mathbb{C}.$$

As shown, this method has consistency order 2. It is also clearly explicit. All that is left to show is that it is L-stable.
$y_{n+1} = R(z) y_n$ where $R(z) = \exp(z)$ and $z = \lambda h$.
We write $z = Re(z) + i Im(z)$.

So ~~$|Re(z)|$~~

$$|R(z)| < 1 \iff |e^{Re(z) + Im(z)i}| < 1$$
$$\iff |e^{Re(z)}||e^{Im(z)i}| < 1$$
$$\iff |e^{Re(z)}| < 1 \quad , \text{as } |e^{Im(z)i}| = 1,$$
$$\iff Re(z) < 0 \text{ based on basic properties of the exponential function.}$$

So $\{z \in \mathbb{C} : Re\, z < 0\} \subset S_R$ (the stability region) as in fact they are equal. This means the method is A-stable.
Furthermore, $|R(z)| = |e^{Re(z)}||e^{Im(z)i}|$
$= |e^{Re(z)}|$, as explained above.
~~So as~~ As $Re(z) \to -\infty$, $|e^{Re(z)}| = e^{Re(z)} \to 0$ thus
$|R(z)| \to 0$ i.e. $R(z) \to 0$.
So, the method is L-stable.

Adi Jayakumar (1905687), Tom Dalby (1903831)

MA261 Modelling and Differential Equations Assignment Sheet 3

2.1)
The code for this section can be found in file "Q1.py".
The results for the Heun method were:

j such that *h_j = T/N_02^j*	EOCs	Errors
0	N/A	0.000411
1	2.085088	0.000097
2	2.049271	0.000023
3	2.025652	0.000006
4	2.013009	0.000001
5	2.006542	0.000000
6	2.003280	0.000000
7	2.001642	0.000000
8	2.000822	0.000000
9	2.000412	0.000000
10	2.000204	0.000000

We see that the errors converge to 0 and the EOCs appear to converge to 2 as expected.

The results for the Crank-Nicholson method were

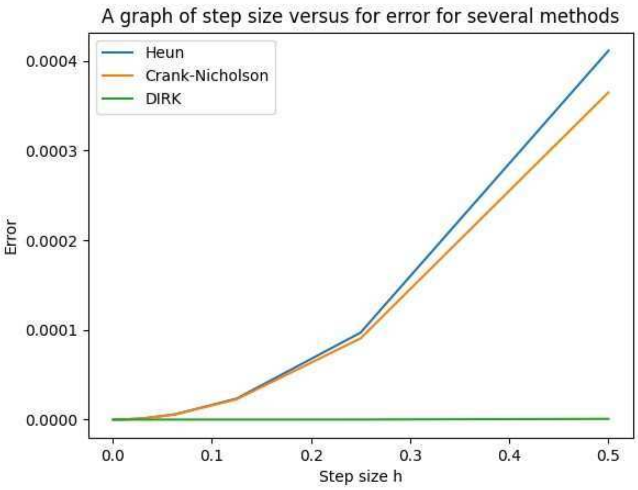
j such that *h_j = T/N_02^j*	EOCs	Errors
0	N/A	0.000365
1	2.009956	0.000091
2	2.002443	0.000023
3	2.000615	0.000006
4	2.000154	0.000001
5	2.000038	0.000000
6	2.000007	0.000000
7	1.999996	0.000000
8	1.999999	0.000000
9	2.000000	0.000000
10	2.000015	0.000000

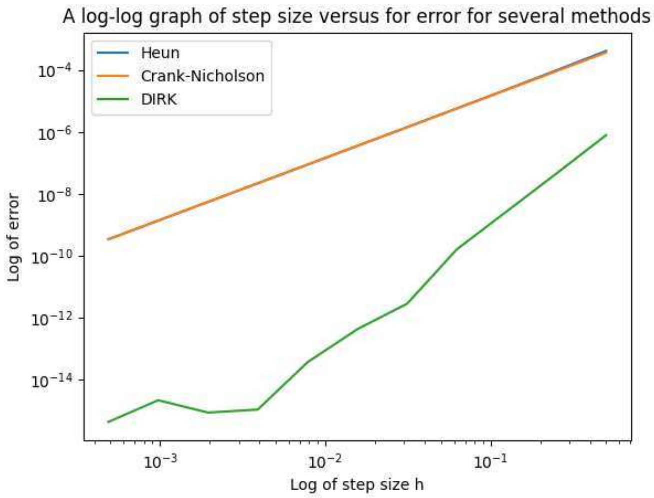
We see that the errors converge to 0 and the EOCs appear to converge to 2 as expected.

j such that *h_j = T/N_02^j*	EOCs	Errors
0	N/A	0.000001
1	4.119707	0.000000
2	4.064285	0.000000
3	4.054269	0.000000
4	5.832484	0.000000
5	2.732821	0.000000
6	3.538420	0.000000
7	5.070389	0.000000
8	0.321928	0.000000
9	-1.321928	0.000000
10	2.321928	0.000000

The errors seem to converge to 0 which is as expected. However, the EOCs seem to be somewhat more erratic. This might be due to the fact that for the smaller step sizes, the error between the approximation and the exact values was on the order of the machine epsilon and therefore we only see noise though we can estimate that the order of convergence of this method on this ODE was 4 from the first 3 values in the table.

The required plots are below:



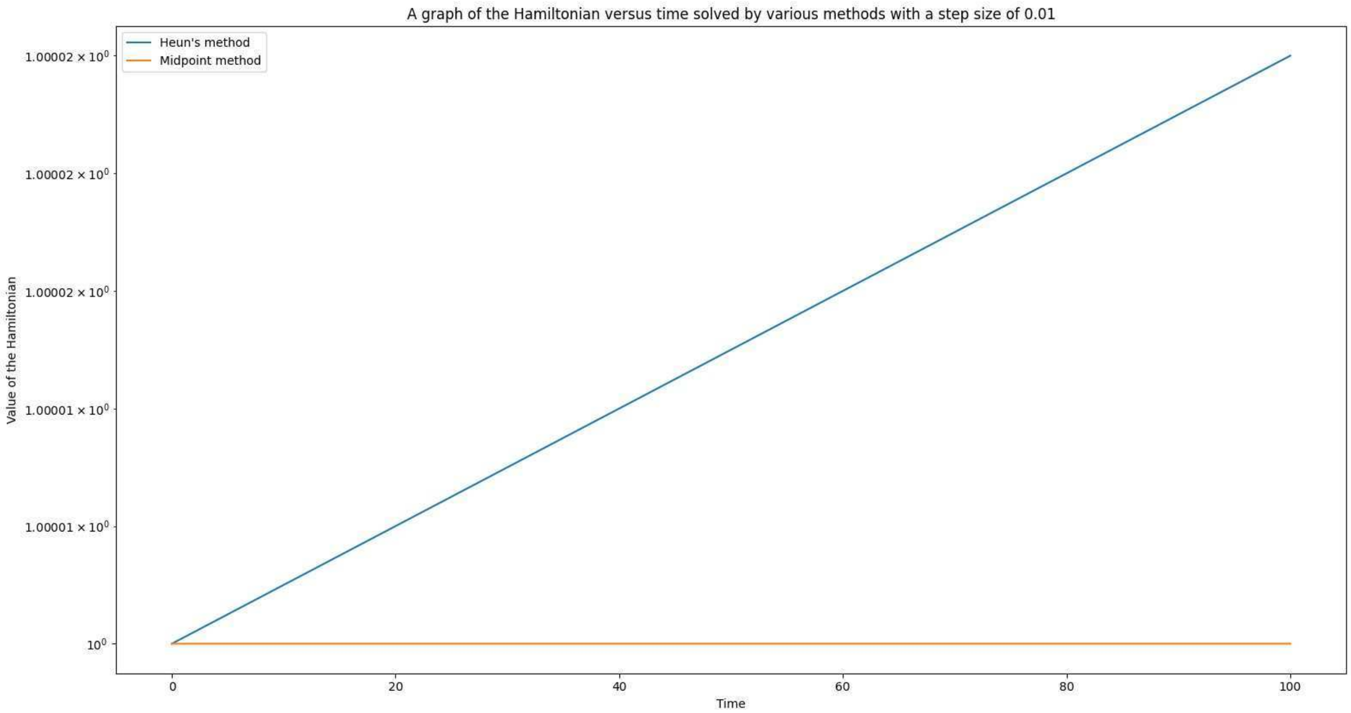


These plots confirm the theory that the errors were on the order of the machine epsilon for the smaller values of *h* and so those results are rather noisy.

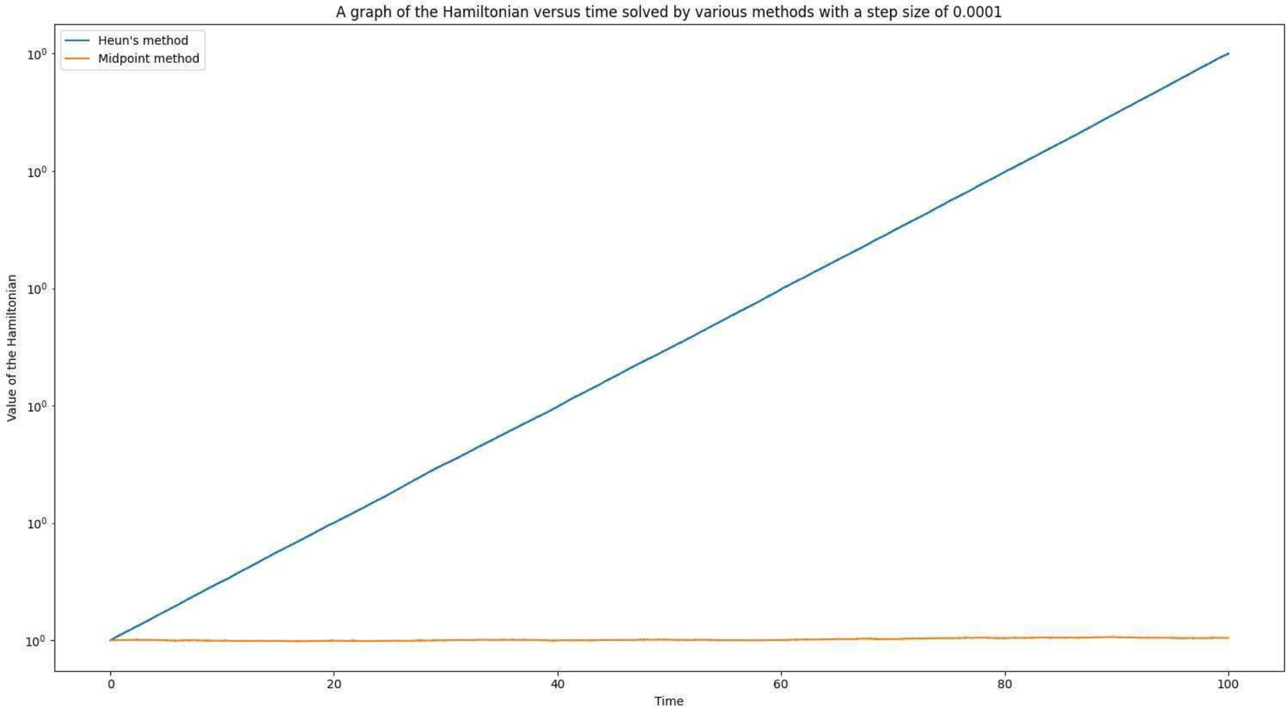
2.2)
The code for this file can be found in file “Q2.py” which consists of essentially the same code from the previous question but with some trivial modifications which are the different differential equation and the requirement to plot the time evolution of the Hamiltonian. However, we felt that each question ought to have its own self contained script hence we duplicated the code.

The system of differential equations arising from this Hamiltonian are:
$x'(t) = p(t),$
$p'(t) = -x'(t)$

The time evolution of the Hamiltonian for both of the required methods with a step size of 0.001 up until a time $t = 100$ was:



Similarly, for a time step of 0.0001 the time evolution of the Hamiltonian was:



For both time step sizes, the Hamiltonian did not deviate greatly for either method. However, with Heun’s method it did increase over time at a seemingly constant rate, whereas with the midpoint method, it stayed constant for both time steps and over the whole time range.

2.3) Consider the Forward-Euler method

From the notes, this is L-Stable whenever

$h \leq \frac{2|Re(\lambda)|}{|\lambda|^2}$

$\lambda = -9+9i$ so $|\lambda|^2 = 2 \cdot 9^2$, $|Re(\lambda)| = 9$

So require $h \leq \frac{1}{9}$.

Consider Heun's method

After simplification in the case $f(t, y(t)) = \lambda y(t)$, Heun's method becomes

$y_{n+1} = \left(1 + \mu + \frac{\mu^2}{2}\right) y_n$ where $\mu = h\lambda$

So we require $\left|1 + \mu + \frac{\mu^2}{2}\right|^2 < 1$

Set $\mu = x + iy \quad x, y \in \mathbb{R}$

So $\mu^2 = x^2 - y^2 + 2ixy$

So $1+\mu+\frac{\mu^2}{2} = 1+x+\frac{(x^2-y^2)}{2}+i(y+xy)$

Recalling $\mu = h\lambda = -qh + qhi$

we get $x=-qh$, $y=qh$

So

$$\left|1+\mu+\frac{\mu^2}{2}\right|^2 = (1-qh)^2 + (qh-q^2h^2)^2$$

Expanding this out and setting <1 and simplif-ying gives

$$\alpha^4 - 2\alpha^3 + 2\alpha^2 - 2\alpha < 0$$

$$\alpha(\alpha^3 - 2\alpha^2 + 2\alpha - 2) < 0$$

Now, using Wolfram Alpha to approximate the roots of the function the LHS and using properties of cubics and quartics, we deduce that LHS <0 if

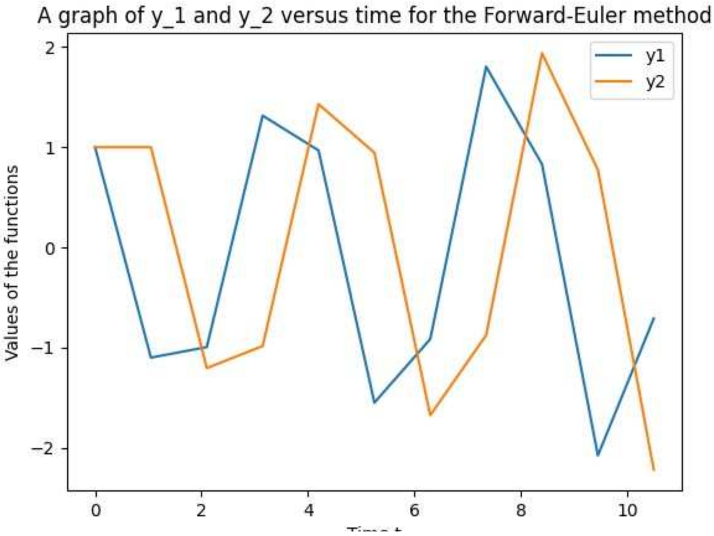
$h < \frac{1.5437}{q}$ $\leftarrow$ approximation of one of the roots of the LHS

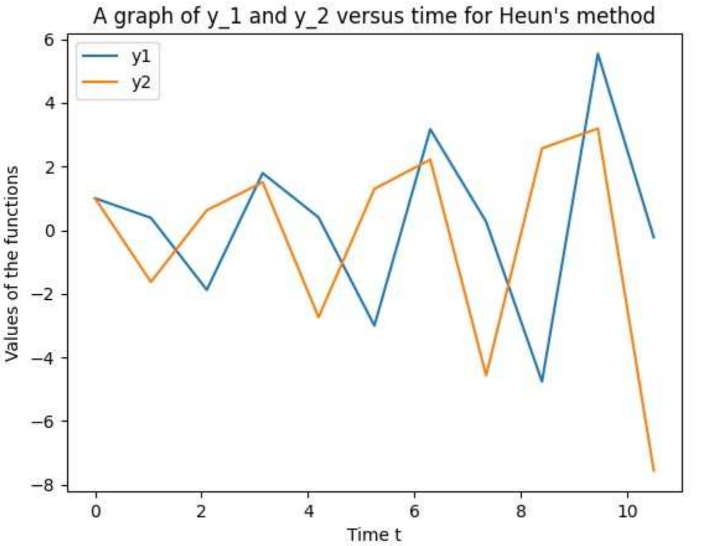
2.3)

The code for this file can be found in file “Q3.py” which consists of essentially the same code from the previous question but with some trivial modifications which are the different differential equations. However, we felt that each question ought to have its own self contained script hence we duplicated the code.

In all cases of values of q and h for this question, $[y\_1(0), y\_2(0)] = [1, 1]$

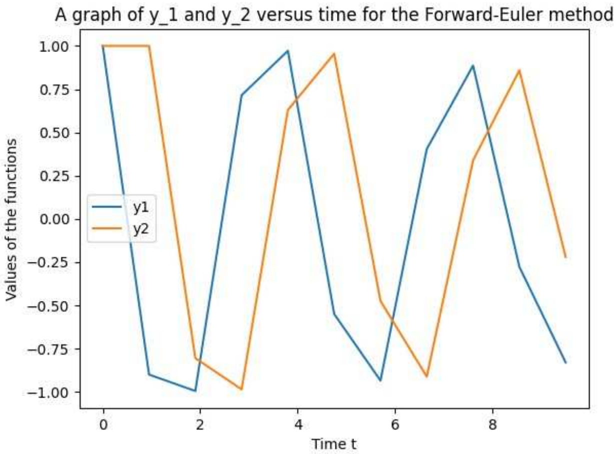
For $q = 1$ and $h = 1.05 * h\_0(q)$, we get

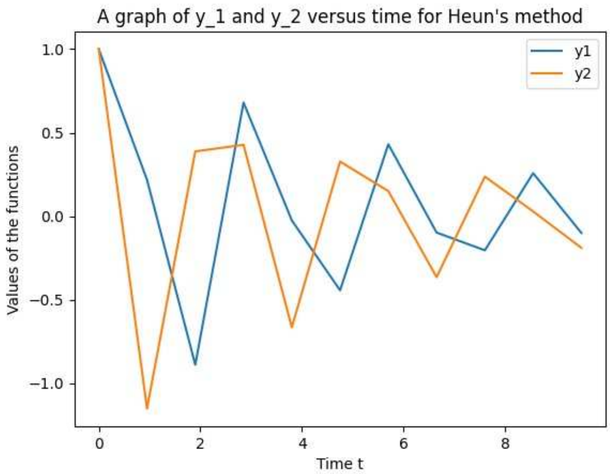




We see that both methods appear to be unstable.

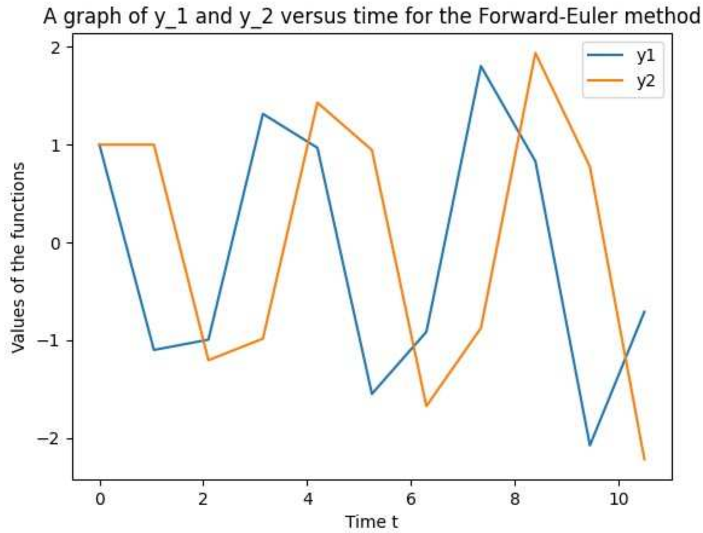
For $q = 1$ and $h = 0.95 * h\_0(q)$, we get:

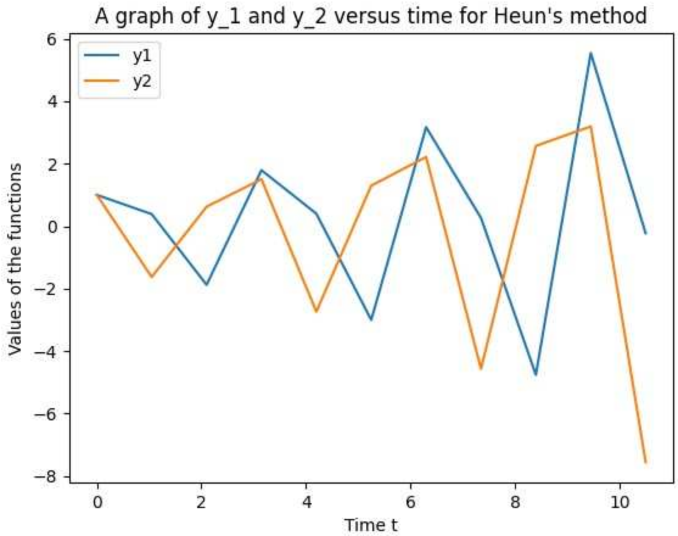




The Forward-Euler method appears to be slightly more stable this time at least staying in between -1 and 1. Heun's method appears to be much more stable this time, converging to 0 for both methods as it should.

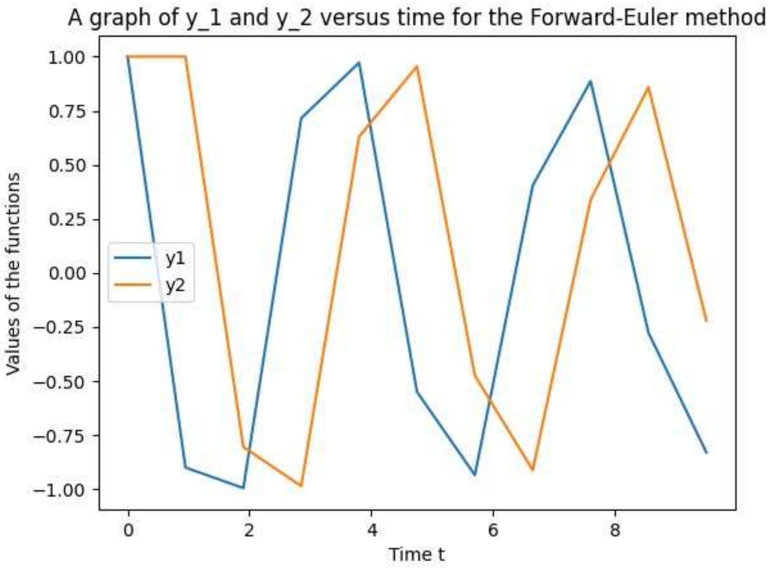
For $q = 10$ and $h = 1.05 * h\_0(q)$, we get

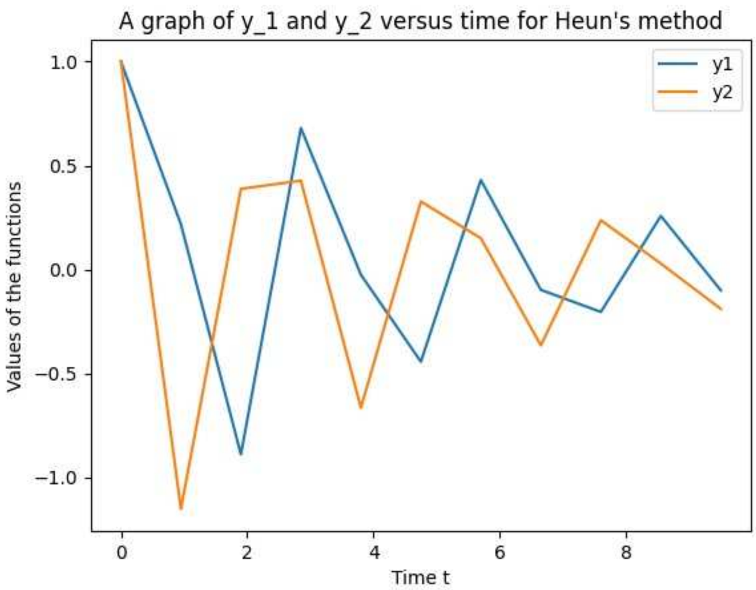




Both methods appear to be unstable in this case as expected.

For $q = 10$ and $h = 0.95 * h\_0(q)$, we get





As expected, both methods appear to be stable in this case and the graphs seem to look like the graphs of the solution functions.