Horacio González Duhart
099058551
MA50174
3 December 2010

Assignment 3

1. *(a)* Attached to the CD the function file *f.m* given as input a vector $u = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ and a matrix $A \in \mathbb{R}^{2\times 2}$ will return as output a vector $f(u) \in \mathbb{R}^2$ defined as:

$$f(u) = \begin{pmatrix} A_{11}x + A_{12}xy \\ A_{21}xy - A_{22}y \end{pmatrix}$$

As an specific case, defining $A = \begin{pmatrix} \alpha & -\beta \\ \delta & -\gamma \end{pmatrix}$ with all parameters $\alpha, \beta, \gamma, \delta > 0$ will lead to the Lotka-Volterra ODE autonomous system, which is a non-linear IVP of the form $\dot{u} = f(u)$.

Also attached to the CD, the *method1.m* script file will ask the user for a timestep value h and find a solution by the forward Euler method to the ODE with boundary condition $u(0) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ in the time interval $[0, 5]$. It will display 2 plots, on the left with red colour the solution for $x(t)$ and with green the solution for $y(t)$. On the right, with blue colour, the plot of x vs y.

(b) The *method2.m* script file does exactly the same thing that the previous one does, but we're not using the forward Euler method any more, the new method is now defined with the following recurrence relationship:

$$U_{n+1} = U_n + hf\left(U_n + \frac{h}{2}f(U_n)\right)$$

(c) Analysing the truncation error for the forward Euler method gives us the following results:

$$\begin{aligned} \tau^E_{n+1} &= u_{n+1} - U_{n+1}(u_n) && \textit{(i)} \\ &= u_{n+1} - (u_n + hf(u_n)) && \textit{(ii)} \\ &= u_n + h\dot{u}_n + \tfrac{h^2}{2}\ddot{u}_n + o(h^3) - u_n - hf(u_n) && \textit{(iii)} \\ &= \tfrac{h^2}{2}\ddot{u}_n + o(h^3) && \textit{(iv)} \end{aligned}$$

Step *(i)* is the definition of the truncation error, just plugging in the actual solution in the iterative step and compare with the numerical approximation.
(ii) follows from the definition of the Euler method.
(iii) follows from the Taylor expansion of $u(t_{n+1})$ near t_n.
(iv) follows from the ODE $\dot{u} = f(u)$. Following the same steps, we are able to calculate the truncation error for what we're calling simply method 2:

$$\begin{aligned}
\tau_{n+1}^{M2} &= u_{n+1} - U_{n+1}(u_n) \\
&= u_{n+1} - \left[u_n + hf\left(u_n + \tfrac{h}{2}f(u_n)\right)\right] && \textit{(i)} \\
&= u_n + h\dot{u}_n + \tfrac{h^2}{2}\ddot{u}_n + \tfrac{h^3}{6}\dddot{u}_n + o(h^4) - u_n - hf\left(u_n + \tfrac{h}{2}f(u_n)\right) && \textit{(ii)} \\
&= h\dot{u}_n + \tfrac{h^2}{2}\ddot{u}_n + \tfrac{h^3}{6}\dddot{u}_n + o(h^4) \\
&\quad -h\left[f(u_n) + \tfrac{h}{2}Df(u_n)f(u_n) + \tfrac{h^2}{8}T\left(f(u_n)\right)\right] && \textit{(iii)} \\
&= \tfrac{h^3}{6}\dddot{u}_n + o(h^4) - \tfrac{h^3}{8}T\left(f(u_n)\right) && \textit{(iv)} \\
&= \frac{h^3}{2}\left[\frac{\dddot{u}_n}{3} - \frac{1}{4}T\left(f(u_n)\right)\right] + o(h^4)
\end{aligned}$$

For these calculations, step *(i)* is just the definition of method 2.
Step *(ii)* follows from the Taylor expansion of $u(t_{n+1})$ near t_n.
The next step, step *(iii)*, is the Taylor expansion for $f\left(u_n + \frac{h}{2}f(u_n)\right)$ near u_n. Notice that $f : \mathbb{R}^2 \to \mathbb{R}^2$, in our case, is a quadratic function, and hence it's equal to the first three terms of its Taylor expansion, and writing $f(u) = \begin{pmatrix} f_1(u) \\ f_2(u) \end{pmatrix}$, it may be expressed as below:

$$f(u+h) = \begin{pmatrix} f_1(u+h) \\ f_2(u+h) \end{pmatrix} = \begin{pmatrix} f_1(u) \\ f_2(u) \end{pmatrix} + \begin{pmatrix} \nabla f_1(u)^T h \\ \nabla f_2(u)^T h \end{pmatrix} + \begin{pmatrix} \frac{1}{2}h^T H f_1(u) h \\ \frac{1}{2}h^T H f_2(u) h \end{pmatrix}$$

Where Hf_k is the Hessian matrix of second derivatives of function f_k. Now define $Df(u) = \begin{pmatrix} \nabla f_1(u)^T \\ \nabla f_2(u)^T \end{pmatrix}$ and $T(u) = \begin{pmatrix} u^T H f_1(0) u \\ u^T H f_2(0) u \end{pmatrix}$, notice that we're evaluating the Hessian matrices in 0 it actually doesn't matter where we evaluate them since they are constant. And so, we get the required result:

$$f(u+hv) = f(u) + hDf(u)v + \frac{h^2}{2}T(v) \quad \forall u, v \in \mathbb{R}^2 \quad \forall h \in \mathbb{R}$$

Finally, step *(iv)* follows from the original ODE problem
$\dot{u} = f(u) \Rightarrow \ddot{u} = Df(u)\dot{u} = Df(u)f(u)$.

Disregarding the $o(h^3)$ terms for the Euler method and the $o(h^4)$ terms for method 2, the script file *q1c.m* makes and interesting comparison describing numerically have we've shown analytically.

In Figure 1 we are able to see 4 plots. On the upper left, the numerical solutions to $x(t)$ using the forward Euler method, 7 time steps are taken, starting with $h = 2^{-4}$ printed in black (actually a "dark" red) and towards $h = 2^{-10}$ printed in red. On the upper right, the approximations to $y(t)$, now from black to green. On the bottom left, we find the plot of the corresponding approximations of (x, y), from the spiralling dark ones to the periodic closed blue one. Finally, on the bottom right, the black stars we plot the value of h vs the calculated truncation errors for each of the approximations evaluated in $t = 5$ (the last value of t). Notice the parabola that approximates this errors.

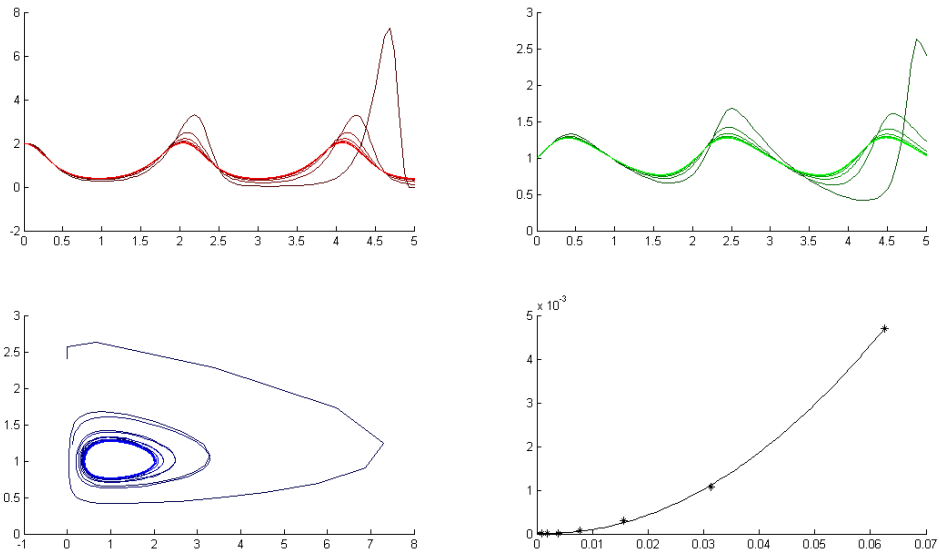

Figure 1: Forward Euler method.

Following the previous set up, in Figure 2 are the corresponding plots for method 2, but the errors are not being approximated by a parabola but rather by a cubic equation on h.

Finally, on Figure 3, there's a comparison between Euler method (red) and method 2

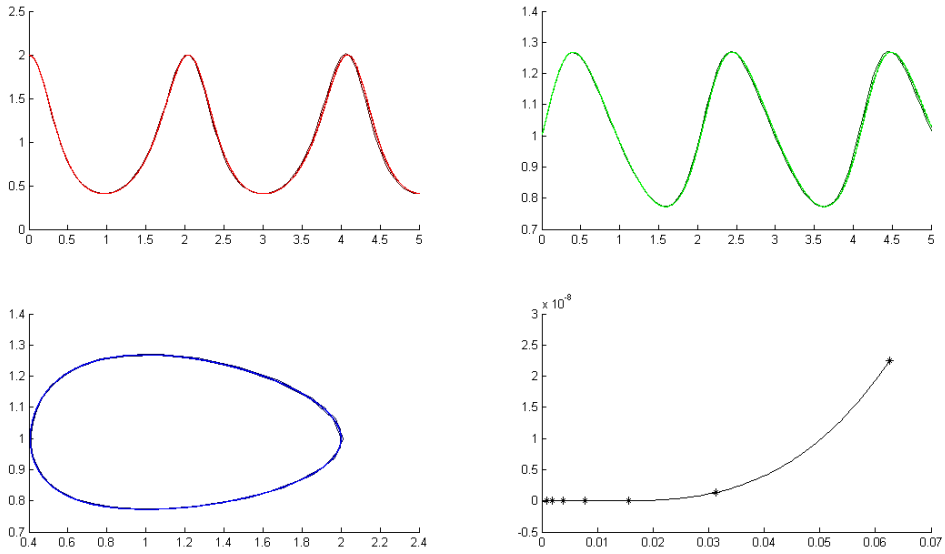

Figure 2: The “method 2” approximations.

(blue) on $h = 2^{-10}$. Both, the calculations and the analytical approach state clear that method 2 is far more superior than the forward Euler method in this scenarios.

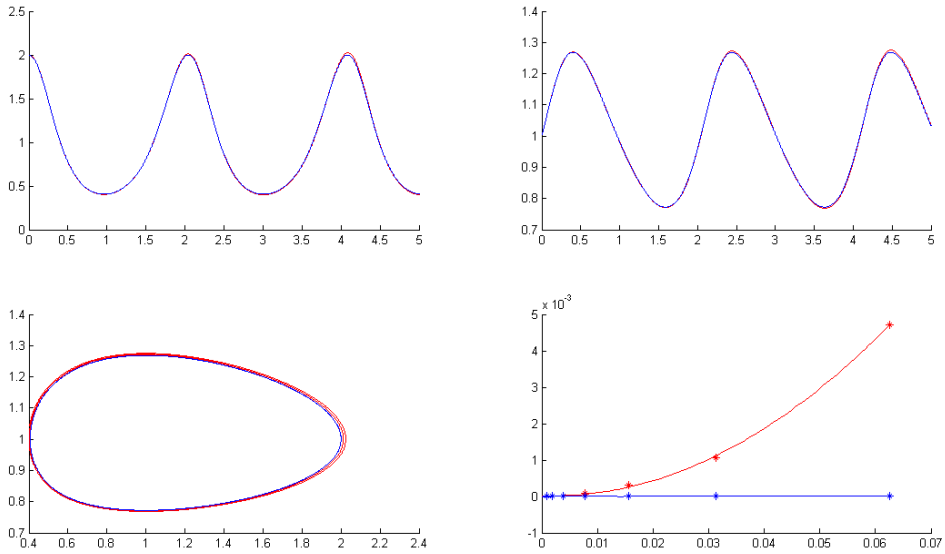

Figure 3: Comparing Euler and method 2.

(d) Changing the boundary conditions to $u(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, it is easy to notice that

$$\dot{u}(0) = f\left(u(0)\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Hence, reaching this value implies no change over time. The numerical calculations are in this case exact, since $U_{n+1} = U_n + hf(U_n) = U_n$ in the Euler method if for some n we have $f(U_n) = 0$, and this happens in the boundary conditions, and for method 2

$$U_{n+1} = U_n + hf\left(U_n + \frac{h}{2}f(U_n)\right) = U_n + hf(U_n) = U_n$$

if for some n we have $f(U_n) = 0$. This case is particularly trivial, and so is the case for $u(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

2. Let us consider the following ODE:

$$\frac{d^2\theta}{dt^2} + \sin\theta = 0, \quad \theta(0) = \alpha < \pi, \quad \frac{d}{dt}\theta(0) = 0$$

(a) Define $E(t) = \frac{1}{2}\left(\frac{d\theta}{dt}\right)^2 - \cos\theta$, then $E(t) = -\cos\alpha \quad \forall t \geq 0$.

Proof

$$\begin{aligned} \tfrac{d}{dt}E(t) &= \frac{d\theta}{dt}\frac{d^2\theta}{dt^2} + \sin\theta\frac{d\theta}{dt} \\ &= \frac{d\theta}{dt}\left(-\sin\theta\right) + \sin\theta\frac{d\theta}{dt} \quad \textit{(i)} \\ &= 0 \end{aligned}$$

Where *(i)* follows from the original ODE $\frac{d^2}{dt^2}\theta = -\sin\theta$, hence E is constant in $\forall t \geq 0$ and from the boundary conditions

$$E(t) = E(0) = \frac{1}{2}\left(\frac{d}{dt}\theta(0)\right)^2 - \cos\left(\theta(0)\right) = -\cos\alpha \quad \forall t \geq 0$$

■

(b) Define $\omega = \dot{\theta} \in \mathbb{R}$, $u = \begin{pmatrix} \theta \\ \omega \end{pmatrix} \in \mathbb{R}^2$, and $f : \mathbb{R}^2 \to \mathbb{R}^2$ as

$f \begin{pmatrix} \theta \\ \omega \end{pmatrix} = \begin{pmatrix} \omega \\ -\sin\theta \end{pmatrix}$. Then, it is clear that the original ODE may be written as

$$\dot{u} = f(u), \quad u(0) = \begin{pmatrix} \alpha \\ 0 \end{pmatrix}, \quad \alpha < \pi$$

The *pendulum.m* script file will solve this equation by either the forward Euler, the symplectic Euler or Störmer-Verlet methods over a time interval $[0, T]$, a given value of $\alpha < \pi$, and a time step $h > 0$. Furthermore, it will plot the approximation for $\theta(t)$ on the left and $E(t)$ on the right. For completeness of this report, the methods are described below.

Forward Euler: $U_{n+1} = U_n + hf(U_n)$

Symplectic Euler: $\Theta_{n+1} = \Theta_n + h\Omega_n, \quad \Omega_{n+1} = \Omega_n - h\sin\Theta_{n+1}$

Störmer-Verlet: $\Theta^\star = \Theta_n + \frac{h}{2}\Omega_n, \quad \Omega_{n+1} = \Omega_n - h\sin\Theta^\star, \quad \Theta_{n+1} = \Theta^\star + \frac{h}{2}\Omega_{n+1}$

Where U_n, Θ_n, and Ω_n are the numerical approximations of u_n, θ_n, and ω_n respectively.

(c) The script file *pendulum2.m* will do the 3 methods with the fixed values of $T = 100$, $h = 0.1$, and $\alpha = 0.25$. We can see the results in Figure 4, in green we appreciate the forward Euler solution (on the upper left) plotted in a completely different scale since it is presenting several problems, the function E (on the upper right) that is supposed to be constant is far from it, and increasing, the solution is not good at all. In red, the symplectic Euler solution, which has less problems, but the function E is still not constant, it is a periodic function around its true value. In blue, the Störmer-Verlet solution, which clearly seems to be the one with the least of the problems, though E is not constant yet.

(d) From the known fact that $\frac{\sin\theta}{\theta} \to 1$ as $\theta \to 0$, we may approximate $\sin\theta \approx \theta$ for small θ. In this case, define the matrix $T = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and we may approximate the original problem as $\dot{u} = f(u) \approx Tu$ with the same boundary conditions (with α small!)

For this linear system, the forward Euler method says

$$U_{n+1} = U_n + hTU_n = (I + hT)U_n$$

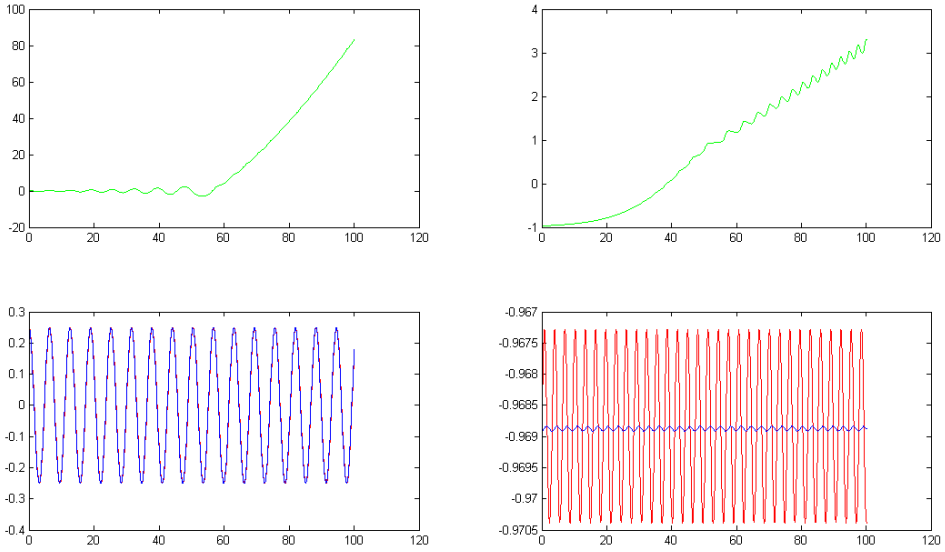

Figure 4: Pendulum solutions with different approaches.

where I is the 2×2 identity matrix. Then it is clear, that by defining $A = I + hT$, we have the relation $U_{n+1} = AU_n$ that implies

$$\begin{aligned}
\|U_{n+1}\|^2 &= \|AU_n\|^2 \\
&= \left\| \begin{pmatrix} 1 & h \\ -h & 1 \end{pmatrix} \begin{pmatrix} \Theta_n \\ \Omega_n \end{pmatrix} \right\|^2 \\
&= (\Theta_n + h\Omega)^2 + (-h\Theta_n + \Omega_n)^2 \\
&= \Theta_n^2 + 2h\Theta_n\Omega_n + h^2\Omega_n^2 + h^2\Theta_n^2 - 2h\Theta_n\Omega_n + \Omega_n^2 \\
&= (1+h^2)\Theta_n^2 + (1+h^2)\Omega_n^2 \\
&= (1+h^2)(\Theta_n^2 + \Omega_n^2) \\
&= (1+h^2)\|U_n\|^2
\end{aligned}$$

Actually, it can be proved that $\|U_{n+1}\|^2 = \|A\|_2^2 \|U_n\|^2$ where $\|A\|_2 = \max\limits_{\|x\| \neq 0} \dfrac{\|Ax\|}{\|x\|}$ for this particular matrix A.

(e) Using the same approximation for $\sin\theta$ and solving by the symplectic Euler method,

we find the following

$$\begin{aligned} & \Theta_{n+1} &=& \Theta_n + h\Omega_n \\ \Rightarrow \quad & \Omega_{n+1} &=& \Omega_n - h\Theta_{n+1} \\ & &=& \Omega_n - h(\Theta_n + h\Omega_n) \\ & &=& -h\Theta_n + (1-h^2)\Omega_n \end{aligned}$$

$$\therefore \quad \begin{pmatrix} \Theta_{n+1} \\ \Omega_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix} \begin{pmatrix} \Theta_n \\ \Omega_n \end{pmatrix}$$

$$\Rightarrow \quad U_{n+1} = AU_n$$

With the, now obvious, definition of A. We will now prove that $\Theta_n^2 + h\Theta_n\Omega_n + \Omega_n^2 = \alpha^2 \quad \forall n \in \mathbb{N}$.

Proof

Define $H = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$ and notice that

$$\begin{aligned} A^T H A &= \begin{pmatrix} 1 & -h \\ h & 1-h^2 \end{pmatrix} \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -h \\ h & 1-h^2 \end{pmatrix} \begin{pmatrix} 1-h^2 & 2h-h^3 \\ -h & 1-h^2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \\ &= H \end{aligned}$$

Then

$$\begin{aligned} \Theta_{n+1}^2 + h\Theta_{n+1}\Omega_{n+1} + \Omega_{n+1}^2 &= U_{n+1}^T H U_{n+1} \\ &= (AU_n)^T H (AU_n) \\ &= U_n^T A^T H A U_n \\ &= U_n^T H U_n \\ &= \Theta_n^2 + h\Theta_n\Omega_n + \Omega_n^2 \end{aligned}$$

Hence, the value $\Theta_n^2 + h\Theta_n\Omega_n + \Omega_n^2$ doesn't depend on n and evaluation in the initial boundary conditions, we find $\Theta_n^2 + h\Theta_n\Omega_n + \Omega_n^2 = \alpha^2 \quad \forall n \in \mathbb{N}$

■

From these results, we may see that for the forward Euler method, $\|U_n\| \to \infty$ as $t \to \infty$ while $\ddot{\theta}$ is supposed to be controlled (because of the E constant function). Whereas the symplectic Euler method manages this control by keeping $U_n^T H U_n$ fixed in time.

3. Let us consider the following linear ODE system:

$$\frac{du}{dt} = f(u) = \Lambda u, \quad \Lambda = \begin{pmatrix} -100 & 1 \\ 0 & -1 \end{pmatrix}, \quad u(0) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

(a) This is a linear system, hence the solution is simply given by $u(t) = e^{t\Lambda}u(0)$. To calculate $e^{t\Lambda}$, we just find the eigenvalues and eigenvectors for Λ

$$\begin{aligned} \Lambda &= \begin{pmatrix} 1 & \frac{1}{\sqrt{1+99^2}} \\ 0 & \frac{99}{\sqrt{1+99^2}} \end{pmatrix} \begin{pmatrix} -100 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{99} \\ 0 & \frac{\sqrt{1+99^2}}{99} \end{pmatrix} \\ \Rightarrow \quad e^{t\Lambda}u(0) &= \begin{pmatrix} 1 & \frac{1}{\sqrt{1+99^2}} \\ 0 & \frac{99}{\sqrt{1+99^2}} \end{pmatrix} \begin{pmatrix} e^{-100t} & 0 \\ 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{99} \\ 0 & \frac{\sqrt{1+99^2}}{99} \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix} \\ \therefore \qquad u(t) &= \begin{pmatrix} 2\left(1 - \frac{1}{99}\right)e^{-100t} + \frac{2}{99}e^{-t} \\ 2e^{-t} \end{pmatrix} \end{aligned}$$

(b) Since $U_1 = (I + h\Lambda)^0 U_1$, by induction over n,

$$U_{n+1} = U_n + h\Lambda U_n = (I + h\Lambda)U_n = (I + h\Lambda)(I + h\Lambda)^{n-1}U_1 = (I + h\Lambda)^n U_1$$

In order for the the method to be stable, we need the eigenvalues of the matrix $I + h\Lambda$ have a modulus less than 1. This means

$$\begin{aligned} & |1 - 100h| < 1 && \text{and} && |1 - h| < 1 \\ \Rightarrow \quad & -1 < 1 - 100h < 1 && \text{and} && 1 < 1 - h < 1 \\ \Rightarrow \quad & -2 < 100h < 0 && \text{and} && -2 < -h < 0 \\ \Rightarrow \quad & 0 < h < \tfrac{2}{100} && \text{and} && 0 < h < 2 \end{aligned}$$

$$\therefore \qquad 0 < h < \tfrac{1}{50} = 0.02$$

(c) Since the truncation error is $\mathcal{E} = \frac{h^2}{2}\|\ddot{u}\|$, if we want the relative error $R = \frac{\mathcal{E}}{\|u\|}$ to be set to a fixed value $r > 0$ then

$$r = \frac{h^2\|\ddot{u}\|}{2\|u\|} \quad \Rightarrow \quad h = \sqrt{\frac{2r\|u\|}{\|\ddot{u}\|}}$$

Now, since we the explicit form of $u(t)$, we can explicitly find

$$\dot{u}(t) = \begin{pmatrix} -200\left(1 - \frac{1}{99}\right)e^{-100t} - \frac{2}{99}e^{-t} \\ -2e^{-t} \end{pmatrix}$$

$$\ddot{u}(t) = \begin{pmatrix} 20000\left(1 - \frac{1}{99}\right)e^{-100t} + \frac{2}{99}e^{-t} \\ 2e^{-t} \end{pmatrix}$$

4.

5.