1. (2 points each part) Are the following initial value problems stiff? Explain why or why not. In case a problem is stiff in one part of the time interval and nonstiff in another part, identify the approximate time intervals over which it is stiff or nonstiff, respectively. Assume an error tolerance of $10^{-2}$.

(a) $y' = -10^6y$, $t \in [0, 10^{-6}]$, $y_0 = 1$

(b) $y' = -10^6(y - t^2) + t$, $t \in [0, 1]$, $y_0 = 1$

(c) $y' = -10^6(y - \sin(10^6t)) + \cos(10^6t)$, $t \in [0, 1]$, $y_0 = 0$

2. (2 points each part) Given the differential equation

$$y' = -\lambda(y - g(t)) + g'(t),$$

where $\lambda > 0$.

(a) Write down the ‘smooth solution’ (the solution as $\lambda \to \infty$)

(b) What controls the stability? If you wanted to make the problem ‘stiffer’, how would you do it? If you wanted to make it unstable, how would you do it?

3. Consider the $\theta$-method:

$$y_{n+1} = y_n + \theta h f_n + (1 - \theta) h f_{n+1}$$

where $0 \leq \theta \leq 1$ and $f_n = f(t_n, y_n)$.

(a) (2 points) What is the local truncation error?
(b) (1 point) What is the local error?

(c) (1 point) Given that the method is 0-stable, what is the order of this method? (hint: this depends on $\theta$)

(d) (2 points) A method is called A-stable if its region of absolute stability contains the left half of the complex plane. For which values of $\theta$ is the $\theta$-method A-stable? Justify your answer.

4. (15 points) Write a well-structured MATLAB code to implement the theta method for systems of ODEs.

For $\theta = 0, 0.5, 1$, use your code for solving:

(a) The following problem:

$$
\mathbf{z}' = \frac{1}{2} \begin{pmatrix} -101 & 99 \\ 99 & -101 \end{pmatrix} \mathbf{z} + \begin{pmatrix} 100 \sin t + \cos t \end{pmatrix},
$$

for $0 \leq t \leq 1$, with initial value $z_1 = 3$, $z_2 = -1$. Try this for stepsizes $h = .01$ and $h = .05$.

(b) The predator-prey problem

$$
y_1' = .25y_1 - .01y_1 y_2 \\
y_2' = -y_2 + .01y_1 y_2
$$

for $0 \leq t \leq 100$ with stepsizes $h = 0.1$ and $h = 0.001$, and initial values $y_1 = y_2 = 10$. Plot $y_1$ vs. $t$ and $y_2$ vs. $t$, and $y_1$ vs. $y_2$.

Explain your results for both problems.

Note: You should write your own Newton iteration, but there is no need to write your own linear system solver. Use the function provided in Matlab for that. Provide a function for each problem that computes the Jacobian matrix (i.e. manually use the simple rules of calculus to find the expressions for the elements of the Jacobian, and write a function that populates the matrix with these elements). Do not use finite difference approximation or automatic differentiation to compute the Jacobian. DO NOT USE THE MATRIX INVERSE. IF YOU USE THE MATRIX INVERSE, YOU WILL NOT GET CREDIT FOR THIS PROBLEM!
5. (5 points) Using your theta method code to implement the forward Euler method, backward Euler method, and trapezoid method (i.e. choose $\theta$ to be 1, 0, 0.5, respectively), solve the second-order ODE (convert to first order system first) $y'' = -y$, $y(0) = 0$, $y'(0) = 1$, $t \in [0, 100]$, with stepsizes $h = 0.2$ and $h = 0.02$. Plot the results in the phase plane ($y$ versus $y'$). Explain your results.