Numerical methods in differential equations 1

Problem 1. Numerical methods for approximate solution of the initial value problem

$$y'(t) = f(t, y(t)), \quad t \in [a, b], \quad y(a) = y^0.$$

Taking the step size h we construct a number sequence $\{y^n\}$, whose elements are treated as the approximations to the exact values $y(t_n)$: $y^n \approx y(t_n)$ at mesh points $t_n = a + n \cdot h$, $n = 0, 1, 2, \ldots, N$, $N = [\frac{b-a}{h}]$. Examples of such constructions are given below.

1) the explicit Euler method: $y^{n+1} = y^n + hf(t_n, y^n), \;\; y^0 = y(a)$;
2) the implicit Euler method: $y^{n+1} = y^n + hf(t_{n+1}, y^{n+1}), \;\; y^0 = y(a)$;
3) the trapezoidal method: $y^{n+1} = y^n + \frac{h}{2}(f(t_n, y^n) + f(t_{n+1}, y^{n+1})), \;\; y^0 = y(a)$;
4) the midpoint method: $y^{n+1} = y^{n-1} + 2hf(t_n, y^n)$,
$y^0 = y(a), \; y^1 = y(a) + hf(a, y(a))$.

For each approximate method, we define the error of approximation

$$e(h) = \max_{1 \le n \le N} |y(t_n) - y^n|.$$

One of the main task is an analysis of the approximation error $e(h)$ of the numerical method. We say that the order or convergence rate of the method is $\alpha_0 > 0$, if

$$|e(h)| \le C \cdot h^\alpha,$$

for some constant $C > 0$ and $\alpha_0 \le \alpha$ for any initial value problem. Use methods 1) - 4) to find an approximate solution to the following initial value problem

$$y' = y, \quad t \in [0, 2], \quad y(0) = 1.$$

In this case $y(t) = e^t$ is the exact solution.
1. Do numerical tests with Matlab for various step sizes, for example $h = 0.5, \; 0.2, \; 0.1, \; 0.05, \ldots$ or equivalently $n = 4, \; 10, \; 20, 40, \ldots$
2. Compare graphically the exact and approximate solutions.
3. For each method give an estimate of its convergence rate.
This can be done as follows
a) choose two stepsizes, for example $h_1 = 0.01$ and $h_2 = h_1/2 = 0.005$,
b) compute the errors $e(h_1)$ and $e(h_2)$,

c) the ratio $r = e(h_1)/e(h_2)$ is approximately equal to 2^α, whence we determine α.
4. How does the convergence rate change when the initial values y^0 in methods 1)-3) and y^0 i y^1 in method 4) are perturbed, for example let you consider the following examples

1. $y^0 = y(a) + C \cdot h$ ($C > 0$ is some constant), in methods 1) - 3),

2. $y^0 = y(a)$, $y^1 = y(a) + \frac{1}{2}hf(a, y(a))$, in method 4).

Try to give an explanation of such a behaviour of methods 1)-4).

The Matlab programe should have the following or equivalent structure:

Input data: $a, b, t_0, y_0, f(t, y)$, an exact solution $y_d(t)$
Beginning of the loop: for $n = [10, 20, 40, \ldots]$
t=linspace$(a, b, n + 1)$ h=$\frac{b-a}{n}$
Determining the vector of the exact solution values: $y_{dd}(k) = y_d(t_k)$, $k = 1, 2, \ldots, n$
Determining the vector of approximate solution values: $y_a(k) = \ldots, \; k = 1, 2, \ldots, n$
Collecting errors: er(h)=max $\lvert y_a(k) - y(t_k)\rvert$
Plotting the results: plot$(t, y_{dd}, '.', t, y_a, '.')$
End of the loop

Analysis of the error:
$\mathrm{er}(h)/\mathrm{er}(\frac{h}{2}) = \ldots$

Problem 2. The initial value problem

$$y' = -100y + 100\cos t - \sin t, \quad t \in [0,\pi], \quad y(0) = 1$$

has an exact solution $y(t) = \cos t$. Do tests as in exercise 1 and in each case determine the range of the step size h, for which computations stabilize.

Basic numerical algorithms for the ordinary differential equations :

1. First order methods

1) the explicit Euler method: $y^{n+1} = y^n + hf(t_n, y^n),\ \ y^0 = y(a);$
2) the implicit Euler method: $y^{n+1} = y^n + hf(t_{n+1}, y^{n+1}),\ \ y^0 = y(a);$

2. Second order methods

3) the trapezoidal method: $y^{n+1} = y^n + \frac{h}{2}(f(t_n, y^n) + f(t_{n+1}, y^{n+1})),\ \ y^0 = y(a);$
4) the midpoint method: $y^{n+1} = y^{n-1} + 2hf(t_n, y^n),$
$y^0 = y(a),\ y^1 = y(a) + hf(a, y(a)).$

3. The Runge-Kutta's methods

3a. the second order method

step $y_i \to y_{i+1}$:

$$k_1 = f(t_i, y_i)h$$
$$k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1)h$$
$$y_{i+1} = y_i + k_2$$

3b. the fourth order method

step $y_i \to y_{i+1}$:

$$\begin{aligned}
k_1 &= f(t_i, y_i)h \\
k_2 &= f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1)h \\
k_3 &= f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2)h \\
k_4 &= f(t_i + h, y_i + k_3)h \\
y_{i+1} &= y_i + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4
\end{aligned}$$

4. the second order Runge-Kutta method for the system of equations

$$\begin{cases} x' = f(t, x, y) \\ y' = g(t, x, y) \end{cases}$$

step $\begin{bmatrix} x_i \\ y_i \end{bmatrix} \to \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix}$:

$$\begin{aligned}
k_1 &= \begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} f(t_i, x_i, y_i)h \\ g(t_i, x_i, y_i)h \end{bmatrix} \\
k_2 &= \begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} = \begin{bmatrix} f(t_i + \frac{1}{2}h, x_i + k_{11}, y_i + k_{12})h \\ \\ g(t_i + \frac{1}{2}h, x_i + k_{11}, y_i + k_{12})h \end{bmatrix}
\end{aligned}$$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix}$$