Assignment 8

Mathematics: Error Analysis for Euler Method
Programming: Vector operations for functions

We want to solve

$$y' = f(x, y)$$
$$y(x_s) = y_s$$

on the interval $[x_s, x_e]$, compare against exact solution and see how our error depends on method, the step size, and the properties of the function.
The simplest method is the so-called Euler Method. We decide on, say, the number of panels to use n, to give us an appropriate step size $h = \frac{x_e - x_s}{n}$. This, in turn, gives us $x_i = x_s + (i-1) * h = x_{i-1} + h$, and $y_i = y_{i-1} + h * f(x_{i-1}, y_{i-1})$ for $1 \leq i \leq n+1$ (be careful about start and end points, remember n panels makes $n+1$ endpoints).
To calculate the error we need the exact answer (or a much more accurate answer than the one we have). A simple way to calculate error is to see the difference between actual and approximation at the end of interval of computation $error = y_{exact} - y_{n+1}$.
To investigate the dependence of error on step-size you may rerun the loop for $k = 1 : 10$ rounds, where in each round you double the number of panels and halve the step-size, $n = 2^k$, resulting in an error $error_k$ that depends on h_k.
Now plot $\ln |error_k|$ vs $ln|h_k|$. Depending on the range of h_k values your graph may have three parts, at high h_k values you will see a curve, at very low h_k values, you will see a zig-zag, and in between you will a linear part. The linear part is where the theoretical approximation holds well and you can use it to find the order of the method.
As usual you are trying to find p and C in $e = Ch^p$. Now p is the slope of the linear part and $\ln C$ is the intercept of the extension of the linear part.
Perform the project for
a) $y' = -2y$, $y(0) = 10$, plot the exact and approximate solutions for $h = 0.1$, find the order of the method.
b) $y' = x - y$, $y(1) = 2$, plot the exact and approximate solutions for $h = 0.1$, find the order of the method.
Matlab is capable of handling a vector of values at the same time, and much more efficiently than doing them one at a time. Rewrite your programs to take advantage of this "parallelization" capability.