MA 584
Homework #1
Due date: 9/8/2020

1. (a) Consider the Euler's method IVP $y'(t) = \lambda y$, $y(0) = 1$. We want to find the solution at $t = 20$. For which λ (positive or negative) that we can apply the Euler's method to get a reasonably accurate numerical result?

   (b) If $\lambda = -5$, can we use arbitrary step size, say $h = 0.75$ in the Euler's method? What's the largest step size h can we use? If we wan to relative error of the computed solution is bounded by 10^{-4}, what is a suitable h (the largest possible)?

   Hint: We have the error estimate $\left|\frac{y(T)-y_N}{y(T)}\right| \le Che^{LT}$, where L is the Lipschitz constant, T is the final time, y_N is the numerical solution of the final step. C is a constant and can be treated as $O(1)$ in your estimate.

2. Consider Lorenz's equation

$$
\begin{aligned}
x' &= -\tau x + \tau y, &(1)\\
y' &= -xz + rx - y, &(2)\\
z' &= xy - bz &(3)
\end{aligned}
$$

where τ, r, b are constant (parameters). One group suggested parameters are $\tau = 10$, $r = 300$, $b = 8/3$, $x(0) = 1$, $y(0) = 0$, and $z(0) = 0$. Set $T = 20$.

   (a) Solve the problem using the forward Euler's method.

   (b) Solve the problem using the Matlab ODE Suite ode45 or ODE23s. (ode45('loren',[y0,tfinal],y0)). Compare with your results. Plot the solution history; the trajectory using plot3(y1,y2,y3), for example.

   (c) (Extra Credit): Solve the problem using the Crank-Nicolson scheme.

3. When we deal with irregular boundaries or use adaptive grids, we need to use non-uniform grids. Derive the finite difference coefficients for the following:

$$
\begin{aligned}
u'(\bar{x}) &\approx \alpha_1 u(\bar{x} - h_1) + \alpha_2 u(\bar{x}) + \alpha_3 u(\bar{x} + h_2), &(4)\\
u''(\bar{x}) &\approx \alpha_1 u(\bar{x} - h_1) + \alpha_2 u(\bar{x}) + \alpha_3 u(\bar{x} + h_2), &(5)\\
u'''(\bar{x}) &\approx \alpha_1 u(\bar{x} - h_1) + \alpha_2 u(\bar{x}) + \alpha_3 u(\bar{x} + h_2) &(6)
\end{aligned}
$$

Are they consistent? In other words, as $h = \max\{h_1,\ h_2\}$ approaches zero, does the error also approach to zero? If so, what are the order of accuracy? Do you see any potential problems with the schemes you have derived?

4. We can use the forward, backward, and central finite difference formulas to approximate $u'(x)$

$$
u'(x) \approx \frac{u(x+h) - u(x)}{h}; \quad u'(x) \approx \frac{u(x) - u(x-h)}{h}; \quad u'(x) \approx \frac{u(x+h) - u(x-h)}{2h}.
$$

Find the local truncation errors including the error constants, for example, $|E_1(h)| \le \frac{h}{2}\|u''\|_\infty$, where $\|u''\|_\infty = \max|u''(x)|$. If we implement the algorithm on a computer, can we take h arbitrarily smaller? What's the best h so that the error can be roughly minimized possible?

Hint: In computing, there are both the local truncation errors that are $O(h)$ or $O(h^2)$, and the round-off errors (from computer number systems) that are $O(1/h)$. The best possible results when they are balanced, see § 2.2.1- § 2.2.2.

5. Modify the Matlab code to solve

$$u''(x) - Ku(x) = f(x), \qquad a < x < b, \qquad u(a) = \alpha, \quad u(b) = \beta.$$

(a) Try you code when $K = 1$, $a = 0$, $b = 1$, $u(x) = \sin(5x)$. What are $f(x)$, α and β? Tabulate the errors for $n = 32, 64, 128, \cdots, 1024$. Can you observe how error decreases? Why? Since we already know the solution, why do we wish to use a finite difference method to solve it?

(b) Repeat with $K = 1$, $u(x) = x^2$. Analyze your result.

(c) Try you code when $K = 1$, $f(x) = x^3$, $a = 0$, $b = 1$, $\alpha = 0$, $\beta = 0$. First find the analytic solution, then repeat the process.

(d) Without computing, consider how the sign of K would affect the numerical method and result. Can we solve the problem for arbitrary K?