LAB 4

1. Consider the initial value problem

$$y' = -100y + 100t + 101,$$
$$y(0) = y_0.$$

Given y_0, h, and N, where the input y_0 specifies the initial value, h is the size of the uniform time stepping and N is the number of time steps for which the approximate solutions are to be computed, use the the following methods

(a) *method 1:*

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}\left(f_t(t_n, y_n) + f_y(t_n, y_n)f(t_n, y_n)\right)$$

(b) *method 2:*

$$\begin{aligned} y_{n+1} = y_n + hf(t_n, y_n) &+ \frac{h^2}{2}\left(f_t(t_n, y_n) + f_y(t_n, y_n)f(t_n, y_n)\right) \\ &+ \frac{h^3}{6}\Big(f_{tt}(t_n, y_n) + 2f_{ty}(t_n, y_n)f(t_n, y_n) + f_{yy}(t_n, y_n)f^2(t_n, y_n) \\ &\qquad + f_t(t_n, y_n)f_y(t_n, y_n) + f_y^2(t_n, y_n)f(t_n, y_n)\Big) \end{aligned}$$

to solve this IVP to obtain approximations y_n^{M1} and y_n^{M2} respectively for y at uniform time steps $t_n = nh$, $n = 0, 1, \cdots, N$. The first line of your Matlab implementation should read

```
function [y,yM1,yM2] = lab4_ex1(y0,h,N)
```

2. Assume non-rotating spherical earth of radius $r_0 = 6378137\,m$. The *Earth Centered Inertial* (ECI) coordinates of a position vector P are given by

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} r\cos\lambda\cos\Lambda \\ r\cos\lambda\sin\Lambda \\ r\sin\lambda \end{bmatrix}$$

in terms of latitude (λ), longitude (Λ) and radial distance (r). The initial velocity vector V of a projectile launched from the location P, in terms of its magnitude v, flight path angle γ and azimuth angle ψ, in ECI coordinates, is given by

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} -\sin\lambda\cos\Lambda & \sin\Lambda & \cos\lambda\cos\Lambda \\ -\sin\lambda\sin\Lambda & -\cos\Lambda & \cos\lambda\sin\Lambda \\ \cos\lambda & 0 & \sin\lambda \end{bmatrix} \begin{bmatrix} v\cos\gamma\cos\psi \\ v\cos\gamma\sin\psi \\ v\sin\gamma \end{bmatrix}.$$

The time dependent state vector of the projectile, say $U(t)$, consists of its position, $P(t)$ and velocity, $V(t)$, that is

$$U(t) = \begin{bmatrix} P(t) \\ V(t) \end{bmatrix},$$

and is governed by the equations of motion given by

$$\frac{d}{dt}U(t) = F(U(t)),$$

with initial condition

$$U(0) = \begin{bmatrix} P_0 \\ V_0 \end{bmatrix}$$

and the function $F : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3 \times \mathbb{R}^3$, given by

$$F(U) = \begin{bmatrix} V \\ g(P) \end{bmatrix},$$

where $g : \mathbb{R}^3 \to \mathbb{R}^3$ is the acceleration due to gravity. Assume that the acceleration due to gravity in ECI frame at ECI position $P = (P_x, P_y, P_z)$ is given by

$$g(P) = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} -\frac{\mu}{r_0^2}\frac{P_x}{r}\left\{\left(\frac{r_0}{r}\right)^2 + \frac{3}{2}J_2\left(\frac{r_0}{r}\right)^4\left(1 - 5\left(\frac{P_z}{r}\right)^2\right)\right\} \\ -\frac{\mu}{r_0^2}\frac{P_y}{r}\left\{\left(\frac{r_0}{r}\right)^2 + \frac{3}{2}J_2\left(\frac{r_0}{r}\right)^4\left(1 - 5\left(\frac{P_z}{r}\right)^2\right)\right\} \\ -\frac{\mu}{r_0^2}\frac{P_z}{r}\left\{\left(\frac{r_0}{r}\right)^2 + \frac{3}{2}J_2\left(\frac{r_0}{r}\right)^4\left(3 - 5\left(\frac{P_z}{r}\right)^2\right)\right\} \end{bmatrix}$$

where $\mu = 3986004.418 \times 10^8\ \mathrm{m}^3/\mathrm{s}^2$ and $J_2 = 1091.3 \times 10^{-6}$.

Use the fourth order Runge-Kutta method (RK4) to simulate the trajectory of the projectile from the between the start time $t = 0$ and the final time $t = T$. The first line of your Matlab implementation should read

```
function [Px, Py, Pz] = lab4_ex2(lat0, lon0, v0, gam0, phi0, h, T)
```

where the inputs `lat0` and `lon0` are the surface launch latitude λ_0 and longitude Λ_0 respectively while the inputs `v0`, `gam0` and `phi0` correspond to launch speed v_0, flight path angle γ_0 and azimuth angle ψ_0 respectively. Finally, the input `h` is the constant time step used for the simulation and `T` is the final time T up to which the trajectory is to be simulated (within your code, find the closest value h_0 to the input h such that $T/h_0 = N$ is an integer and use h_0 for RK4 time stepping). The outputs `Px`, `Py` and `Pz` are vectors of size $(N + 1)$ containing the x, y and z coordinates respectively of the points on the simulated trajectory.

To experiment with your code, you may take $\lambda_0 = 0°$, $\Lambda_0 = 0°$, $v_0 = 6000$ m/s, $\psi_0 = 0°$, $\gamma_0 = 20°$, $h = 0.02$ s and $T = 890$ s.