Assignment 9

1. **Linear ODE system, theory.** Consider the linear ODE system $\dot{x} = Ax$. Consider the four stage RK time step algorithm that defines the short time approximation $\widehat{S}(x, \Delta t)$

$$\begin{aligned}
\eta_1 &= \Delta t A x \\
\eta_2 &= \frac{1}{2}\Delta t A\,\eta_1 \\
\eta_3 &= \frac{1}{3}\Delta t A\,\eta_2 \\
\eta_4 &= \frac{1}{4}\Delta t A\,\eta_3 \\
\widehat{S}(x, \Delta t) &= x + \eta_1 + \eta_2 + \eta_3 + \eta_4
\end{aligned}$$

   Show (by calculations, not numerical experiments) that the local truncation error is order Δt^5 and therefore that the global error should be fourth order. *Hint*. This is related to an earlier assignment on computing matrix exponentials. The matrix exponential (as that assignment explained) as a way to express the solution of the differential equation involving A. One of the methods there is this Runge Kutta method, interpreted differently.

2. **Solver validation.**

   Create and validate fixed timestep ODE solvers using

   - Forward Euler (first order).
   - Heun (second order). Pick either the the trapezoid version from the book or the midpoint version from class.
   - The four stage fourth order method from the book.

   The steps to follow are

   - Write a one timestep routine that returns $y = \widehat{S}(x, \Delta t)$. Write one function for each method. The function should have arguments x, Δt, and f. The f is a function that describes the ODE $\dot{x} = f(x)$. The f should be a function that takes x and returns $f(x)$, which should be d component numpy one index arrays. It should "know" whatever parameter values are necessary to specify f. One "pythonic" way to do this is to make f an attribute of a class. There should be three such routines, one for each time step method. Once you create the first one (probably for forward Euler), the next two will be simple.

- Write a function that takes n timesteps to go from $t = 0$ to $t = T$ using one of the methods. This function should take T and n as arguments and then compute Δt as $\Delta t = T/n$. It should also take x_0 and return $\widehat{x}(T)$. It should be possible to change the method just by changing the name of the function that takes a timestep.
- Do a convergence study using the non-linear ODE system

$$\begin{aligned} \dot{x} &= \quad \left(x^2 + y^2\right) y \\ \dot{y} &= -\left(x^2 + y^2\right) x \end{aligned}$$

  Use initial data $x_0 = 1$, $y_0 = 1$ and solve up to time $T = 2$. The exact solution is $x(t) = \cos(t)$, and $y(t) = -\sin(t)$. Verify the order of accuracy of the three methods using the convergence analysis method from earlier in the class and a sequence of n values such as $(10, 20, 40, 80, \cdots)$. You will find that

  - The first order method is so inaccurate that it takes fairly large n before you see first order accuracy.
  - The fourth order method is so accurate that the clear factor of $16 = 2^4$ improvement from n to $2n$ is impossible to see for large n because of roundoff error.
- For various values of T, find the number of f evaluations needed to achieve 1% relative accuracy with each of the three methods (with the fewest timesteps). Comment on which method gets this accuracy "fastest" (fewest evaluations). Is it the same for each T?

3. **A movie.** Put M particles with mass 1 in a quadratic *potential well* $V_0(r) = \frac{g}{2}\|r\|_2^2$. This potential gives a force that pushes a particle toward the origin. Here $r \in \mathbb{R}^2$ is the location of a particle. Take the *interaction potential* to be *repulsive*, which means that each particle "repels" (pushes away) each other particle. The two particle repulsive potential is

$$V_{12}(r_2 - r_1) = \frac{1}{\|r_2 - r_1\|_2} \; .$$

   The *configuration* is determined by the positions of all the particles. We denote it by $R = (r_1, \cdots, r_M)$. The total potential is

$$V(R) = \sum_{j=1}^{M} V_0(r_j) + \sum_{j<k} V_{jk}(r_j - r_k) \; .$$

   The second sum on the right is the sum over all pairs of distinct particles. Taking $j < k$ means, for example, that the force between particle 1 and particle 2 is given by the term V_{12} and is not repeated with the term V_{21}. The force on particle j is determined by the gradient of the total potential with respect to r_j:

$$F_j(R) = -\nabla_{r_j} V(R) \; .$$

For example, the force from the potential well is

$$
\begin{aligned}
& -\nabla_{r_j} \frac{g}{2} \sum_{k=1}^{M} \|r_k\|^2 \\
= & -\frac{g}{2} \nabla_{r_j} \|r_j\|^2 \\
= & -g r_j \ .
\end{aligned}
$$

The force on particle j from the repulsive potentials is a sum over all the other particles. The dynamics are

$$
\ddot{r}_j = F_j(R).
$$

Modify the RK4 (four stage forth order Runge Kutta) ODE solver from exercise 2 to make a movie of the M particles moving in the plane. Represent the particles as dots.