MATH 651: Numerical Analysis II

Instructor: Prof. Wolfgang Bangerth
bangerth@colostate.edu

Homework assignment 5 – due 11/27/2017

Problem 1 (Convergence order of ODE solvers). The following two schemes for the approximation of the ODE $x'(t) = f(t, x(t))$ are both Runge-Kutta methods based on the midpoint rule:

- The explicit Runge-Kutta-2 method:

$$
\begin{aligned}
F_1 &= \Delta t\; f(t_k, x_k) \\
F_2 &= \Delta t\; f\left(t_k + \frac{1}{2}\Delta t, x_k + \frac{1}{2}F_1\right) \\
x_{k+1} &= x_k + F_2.
\end{aligned}
$$

- The implicit midpoint rule, which can be written as an implicit 1-stage Runge-Lutta method:

$$
\begin{aligned}
F_1 &= \Delta t\; f\left(t_k + \frac{1}{2}\Delta t, x_k + \frac{1}{2}F_1\right) \\
x_{k+1} &= x_k + F_1.
\end{aligned}
$$

You can probably guess the convergence order for both of these methods. Prove it rigorously.

(30 points)

Problem 2 (A quantitative comparison of Runge-Kutta methods). For any $0 < \alpha \leq 1$, the Butcher tableau

$$
\begin{array}{c|cc}
0 & & \\
\alpha & \alpha & \\
\hline
 & \left(1 - \frac{1}{2\alpha}\right) & \frac{1}{2\alpha}
\end{array}
$$

defines a family of 2-stage, explicit Runge-Kutta methods that are all of second order. For $\alpha = \frac{1}{2}$, you will recover the explicit Runge-Kutta-2 method of the previous problem. For $\alpha = 1$ you get Heun's method, which is like an explicit variation of the Crank-Nicolson scheme.

All members of this family are explicit and have the same order, so it seems reasonable to ask which α is best. Let us explore this experimentally.

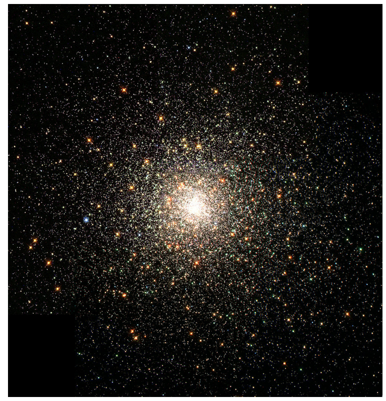
a) Using the equation $x'(t) = x(t), x(0) = 1$, determine experimentally the accuracy of methods defined by different values of α by plotting the error in approximating $x(1) = e$ for $\Delta t = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots \frac{1}{512}$. For which α is the error minimal? (In other words, for which α is the method *best with regards to the error*?)

b) All methods of this family being explicit, they can at best be *conditionally stable*. Using the equation $x'(t) = -x(t), x(0) = 1$, determine experimentally how large you can choose the time step to still obtain a "stable" scheme? For which α is the maximal stable time step maximal? (In other words, for which α is the method *best with regards to the maximal allowed time step*?)

(**Note:** If you want to, you can do the proof for the first part of Problem 1 above for the general method shown here, rather than for the specific case with $\alpha = \frac{1}{2}$. This will yield a remainder term (i.e., the first Taylor term that does not vanish) that contains α, and you can think about what α would minimize this term.)

(30 points)

Problem 3 (N-body simulations). Globular clusters are dense collections of stars that rotate around a common center of mass – a bit like galaxies, just at a smaller scale, and without the large-scale organization where all stars move in roughly the same circular direction. An example of such a cluster is shown on the right.



The Messier M80 globular star cluster. (Source: Wikipedia.)

The motion of these stars can be described by letting each star feel the gravitational force of all other stars. Each star's position $\mathbf{x}_i$, $i = 1, \dots, N_s$, then satisfies the following ODE:

$$m_i \mathbf{x}_i''(t) = \sum_{j=1, j\neq i}^{N} \frac{G m_i m_j}{|\mathbf{x}_i - \mathbf{x}_j|^2} \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|},$$

where the m_i are the masses of the stars, and $G = 6.674 \cdot 10^{11} \frac{\mathrm{m}^3}{\mathrm{kg\,s}^2}$ is the gravity constant.

Let us try to simulate the dynamics of such a cluster. To this end, assume that all of the stars have one solar mass, i.e., $m_i = 1.989 \cdot 10^{30}$kg. Furthermore, assume that the stars' initial positions are random – each component $k = 1, 2, 3$ of each initial position $\mathbf{x}_{i,0}$ is drawn from a Gaussian distribution: $x_{i,0,k} = N(0, \sigma_x)$ where $\sigma_x = 50\,$light years $= 50 \cdot 9.4607 \cdot 10^{15}$m.

We then also need to provide initial conditions. This could be obtained from the virial theorem, but we'll make our lives easier by just assuming $x'_{i,0,k} = N(0, \sigma_v)$ where $\sigma_v = 5\,\frac{\mathrm{m}}{\mathrm{s}}\sqrt{N_s}$ and N_s is the number of stars you consider (because the more stars you have, the larger the gravitational force and consequently the larger the average speed of stars).

Write a code that can simulate a star cluster with N_s stars that is set up this way. Start with $N_s = 2$ and verify that the two stars orbit each other on elliptic orbits. (If they don't, then decrease their velocities until they are slow enough to be gravitationally bound.) Think about how large or small you have to make the time step. Visualize these orbits for at least 10^6 years (i.e., roughly $3.15 \cdot 10^{13}$s).

Then repeat this exercise for larger and larger number of stars. How many stars can you simulate up to at least 10^6 years with your code? Discuss what the limiting factor for your simulations is! **(40 points)**

Problem 4 (A modeling challenge). *This problem is meant as a bonus question. If you feel bored sitting around the table with your family during Thanksgiving, do what mathematicians typically do on such occasions: scribble solution attempts to questions like this on napkins or the back of envelopes. The problem has two parts: a theoretical part that you can do on the napkin (avoid the linen napkins handed down for generations!), and a practical part where you have to implement your model on a computer. In order to comply with rules on animal experiments, we discourage experimental verification of the model. It won't work with a broiled turkey anyway.*

Here's the theoretical problem: Thanksgiving turkeys aren't particularly good at flying. They may try, but they don't really get into the air very gracefully and for extended periods of time. Derive an ODE model for turkey flight that takes into account the following rules (all quantities have units meter, meter per second, etc, as appropriate):

a) the turkey is initially at rest;

b) it then runs horizontally, accelerating at a modest rate of 1.5;

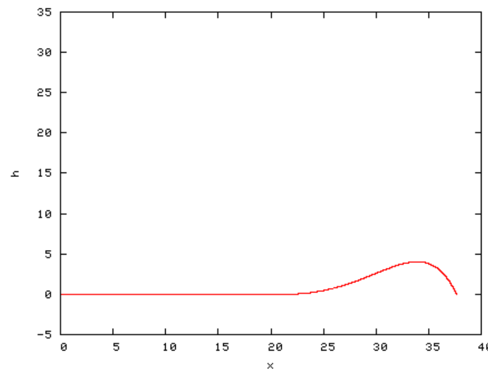
c) when it reaches the lift-off speed of $v = 8$ it gets airborne; from thereon, its vertical (upward) acceleration is $-4 + v$ (in other words, the initial upward acceleration after getting airborne is 4); at the same time, air friction reduces the horizontal velocity by a deceleration of $-v^2/10$;

d) at some point, the turkey's speed will become too slow to sustain flight, its vertical velocity will become negative, and it will eventually fall back to earth.

To write an ODE model for this, you will have to use the following variables: $x(t)$–horizontal distance from the starting point; $v(t)$–horizontal velocity; $h(t)$–height above ground; $u(t)$–vertical velocity.

Practical part: Solve these equations from the turkey's start until where it falls back down to earth. Plot $x(t)$ and $h(t)$ in a single plot to show the turkey's trajectory. If you feel challenged, compute the length of the flight in both seconds and meters.

Hint: A plot of $x(t)$ vs. $h(t)$ (i.e. the turkey's trajectory) would look like the in the figure on the right.



(10 bonus points)

Happy Thanksgiving!