Homework 1. Due Sept. 9

1. Ref. [1], Chapter 2, Problem 3 (page 25)

2. Ref. [1], Chapter 2, Problem 7 (page 26). The declaration on the top is written in C. It says that `x, y, z, w` are floating point numbers (in single precision) and `oneThird` and `oneHalf` are 1/3 and 1/2 calculated in the floating point arithmetic. Assume that `x, y, z, w` are normal numbers and their exponents are greater than `emin`. Then division by 2 is exact.

3. Ref. [1], Chapter 2, Problem 8 (page 26). Do (a) only in double precision. Do not do plots suggested in (b) and (c). Instead, make a three column print out

$$n, \quad \hat{f}_0 - f_0, \quad \hat{p}_0 - p_0$$

for $3 \le n \le 100$ (note that $\hat{f}_0 - f_0$ and $\hat{p}_0 - p_0$ are functions of n) and answer all the questions in (b) and (c) including the extra credit.

References

[1] Bindel and Goodman, Principles of scientific computing

HW2
Bindel and Goodman
(Links to an external site.)
Principles of Scientific Computing
Chapter 4, pages 98 - 99, exercises 1, 2, 4, 8

HW3
Bindel and Goodman
(Links to an external site.)
Principles of Scientific Computing,
Chapter 4, pages 100 - 102
Exercises: **10** (5 points); **11(a,b,c,d,f)** (10 points); **12** (10 points)

HW4
Bindel and Goodman, "Principles of Scientific Computing"
(Links to an external site.)
Chapter 5. Pages 123 - 124.
Exercises: 1, 2, 3

AMSC660 Maria Cameron

Homework 5. Due Oct. 7

1. Consider the overdetermined system of linear equations $Ax = b$ where A is $m \times n$, $m \ge n$, full rank. Let x^* be the least squares solution of $Ax = b$, i.e.,

$$x^* = \arg \min_{x \in \mathbb{R}^n} ||Ax - b||_2^2.$$

   (a) Let $A = QR$ be the QR decomposition of A. Express x^* in terms of R, Q and b.

   (b) Let $A = U\Sigma V^*$ be the singular value decomposition of A. Express x^* in terms of U, Σ, V and b. *Hint: Use Demmel's definition of the SVD where U is $m \times n$ and Σ and V are $n \times n$.*

2. Consider the underdetermined system of linear equations $Ax = b$ where A is $m \times n$, $m \le n$, full rank. Show that the minimum norm solution

$$x^* = \arg \min_{\{x \,:\, Ax = b\}} ||x||_2^2$$

is given by

$$x^* = A^*(AA^*)^{-1}b.$$

3. Let $f(x)$, $x \in \mathbb{R}$, $f : \mathbb{R} \to \mathbb{R}$ be a smooth function. Assume that the sequence of iterates $\{x_k\}$ produced by the secant method converges to a solution x^* of $f(x) = 0$ and $f'(x^*) \neq 0$. Show that the rate of convergence of the secant method is $p = (1 + \sqrt{5})/2$.

4. The goal of this exercise is to understand how the hybrid method works.

   (a) Use the hybrid method to find the initial displacement θ_0 of the pendulum oscillating according to the equation

$$\ddot{\theta} + 0.98 \sin\theta = 0, \quad \theta(0) = \theta_0, \dot{\theta}(0) = 0,$$

such that the period of oscillations is equal to 10. At each iteration, make your code print the iteration number, the current approximation to the solution, and the deviation of the current period from 10. Submit the printout produced of your routine. *Hint: you might find useful the routines* `bisection.m` *where this problem is solved using the bisection method, and* `hybrid.m` *where the hybrid method is used for solving $-1 + 0.1/x = 0$.*

(b) Read any popular article about the Van der Pol oscillator, e.g. the one in Wikipedia:

$$\dot{x} = y,$$
$$\dot{y} = \mu(1 - x^2)y - x.$$

Use the hybrid method to find the value of parameter μ such that the maximal value of $y = \dot{x}$ in the stable limit cycle is 4. *Hint: y reaches its maximal value when $\dot{y} = \mu(1 - x^2)y - x = 0$ and $\dot{y}$ change its sign from + to −. Therefore, you can set* `value` $= \mu(1 - x^2)y - x$ *and* `direction` $= -1$ *in* `events`. At each iteration, make your code print the iteration number, the current approximation to the solution, and the deviation of the current value of value of μ from 4. Submit a print-out of your code and the printout produced by your program.

References

[1] G.W. Stewart, "Afternotes on numerical analysis". Lecture 5 describes the hybrid method.

[2] J. Demmel, "Applied Numerical Linear Algebra". Contains detailed descriptions of QR and SVD.

AMSC660 Maria Cameron

Homework 6. Due Oct. 14

1. To compute $\sqrt{2}$ we consider the following Eudoxos iterations: starting with $x_0 = y_0 = 1$ we set $x_{n+1} = x_n + y_n$ followed by $y_{n+1} = x_{n+1} + x_n$. Then $y_n/x_n \longrightarrow \sqrt{2}$.

   (a) Explain how does the Eudoxos method work.

   (b) How many iterations are required for an error $|y_n/x_n - \sqrt{2}| \le 10^{-6}$?

   Hint. Write the iterative procedure in the matrix form $\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = A \begin{pmatrix} x_n \\ y_n \end{pmatrix}$. *Find eigenvalues* $\{\lambda_1, \lambda_2\}$ *and eigenvectors* $\{v_1, v_2\}$ *of* A. *Then write the initial data* $(x_0, y_0)^T = av_1 + bv_2$. *Show that* $|y_n/x_n - \sqrt{2}| < Cq^n$. *Find* q. *Estimate the number of iterations that are necessary to achieve the required accuracy.*

2. Download the Matlab program `broyden.m` from the class web site. It solves the equation
$$Tf'' - |f'|^2\nabla V(x,y) + (\nabla V(x,y) \cdot f')f' = 0,$$
where $f(t) = (x(t), y(t))$ is a curve subjected to the boundary conditions $(x(0), y(0)) = (x_a, y_a)$, $(x(1), y(1)) = (x_b, y_b)$, T is a given parameter, and $V(x,y)$ is a given function. Therefore, the dimensionality of the discretized problem is $2(n-2)$, where n is the number of points that represent the curve.

   (a) For $T = 1$, 0.7 and 0.5 run it using a segment of a straight line as the initial guess (type `[x,y]=broyden(T,0,0,0)` in the command window). (The residual might not decay for $T = 0.5$.) Plot the residual versus the iteration number using log scale for the y-axis. Do you observe a superlinear convergence?

   (b) i. resetting the approximate Jacobian $J1$ to identity every m iterations for $m = 5, 10, 20$.

   ii. making steepest descent step (i.e., the step direction is $-r(x)$ where $r(x) = 0$ is the equation to be solved) instead of Broyden's step whenever the line search fails.

   Report your findings. Plot the residual versus the iteration number using log scale for the y-axis for each case.

   (c) You should see that it is easy to obtain the solution for $T = 1$, while it gets harder as T decreases. Use the simple continuation method where you use the found curve at a given $T0$ as an initial guess for $T1 < T0$ (to do it, type `[x,y]=broyden(T1,1,x,y)` in the command window). Obtain a solution for $T = 0.01$ starting from a segment of straight line as the initial guess at $T = 1$ and solving the equation for a sequence of T ending at $T = 0.01$ using the outcome of each run as an initial guess for the next value of T. What sequence of values of T did you use?

3. (Approx. problem 11.12 from Nocedal & Wright) Consider

$$H(x; \lambda) = \lambda(x^2 - 1) + (1 - \lambda)(x - 0.5).$$

   (a) Show that there is a zero path for $H(x, \lambda)$ connecting the points $(x = 0.5, \lambda = 0)$ and $(x = 1, \lambda = 1)$.

   (b) Plot the zero path.

4. Consider the function

$$H(x; \lambda) = x^3 - \lambda x^2 + \lambda^2 x - 1 - \lambda \cos \lambda.$$

If $\lambda = 0$, then the root of $H(x; 0)$ is easy to find. Program the zero-path following method and use it to find the solution $H(x; 10) = 0$. Start your zero path at the point where $\lambda = 0$. Submit the printout of your program. Make your program plot the zero path. Submit the plot. Report the solution of $H(x; 10) = 0$ found by your program.

References

[1] J. Nocedal and S. Wright, "Numerical Optimization", Chapter 11.

AMSC660 Maria Cameron

Homework 7. Due Oct. 21

1. (Approx. problems 2.9 and 2.10 from Nocedal and Wright) The SR1 and BFGS are quasi-Newton methods with the following update rules for the approximate Jacobian:

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k, \quad p_k = -B_k^{-1}\nabla f_k,$$

$$\text{SR1}: \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k},$$

$$\text{BFGS}: \quad B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

The goal of this problem is to show that the Newton method and these quasi-Newton methods are scale-invariant.

(a) Suppose that $\tilde{f}(z) = f(x)$, where $x = Sz + s$ for some $S \in \mathbb{R}^{n\times n}$ and $s \in \mathbb{R}^n$. Show that

$$\nabla \tilde{f}(z) = S^T \nabla f(x), \quad \nabla\nabla \tilde{f}(z) = S^T \nabla\nabla f(x) S.$$

Hint: Use the chain rule to express $\frac{\partial \tilde{f}}{\partial z_j}$ *in terms of* $\frac{\partial f}{\partial x_i}$ *for all* $i, j = 1, \ldots, n$.

(b) Show that the Newton update, the SR1 update, and the BFGS update are scale-invariant (if B_0 is chosen appropriately)i.e show that if these methods are applied to $f(x)$ starting from $x_0 = Sz_0 + s$ (with initial Hessian B_0 for SR1and BFGS), and to $\tilde{f}(z)$ starting from z_0 (with initial Hessian $S^T B_0 S$ for SR1 and BFGS), then all iterates are related by $x_k = Sz_k + s$. Assume unit step lengths for simplicity.

2. (Problem 3.7 from Nocedal and Wright.) Derive the following relationship for the steepest descent method, applied to $f(x) = \frac{1}{2}x^T Q x - b^T x$, where Q is symmetric positive definite:

$$\|x_{k+1} - x^*\|_Q^2 = \left\{1 - \frac{(\nabla f_k^T \nabla f_k)^2}{(\nabla f_k^T Q^{-1} \nabla f_k)(\nabla f_k^T Q \nabla f_k)}\right\} \|x_k - x^*\|_Q^2.$$

3. (Approx. problem 3.1 from Nocedal and Wright) (a) Compute the gradient and the Hessian of the Rosenbrock function

$$f(x,y) = 100(y - x^2)^2 + (1 - x)^2. \tag{1}$$

Show that $(1,1)$ is the only local minimizer, and that the Hessian is positive definite at this point.

(b) Program the steepest descent, Newton, SR1 and BFGS algorithms using the backtracking line search (see Chapter 3 from Nocedal and Wrigth). Use them to minimize the Rosenbrock function (1). First start with the initial guess $(1.2, 1.2)$ and then with the more difficult one $(-1.2, 1)$. Set the initial step length $\alpha_0 = 1$ and plot the step length α_k versus k for each of the methods.

Plot the level sets of the Rosenbrock function using the command `contour` and plot the iterations for each methods over it.

Plot $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis for each method. Do you observe a superlunar convergence? Compare the performance of the methods.

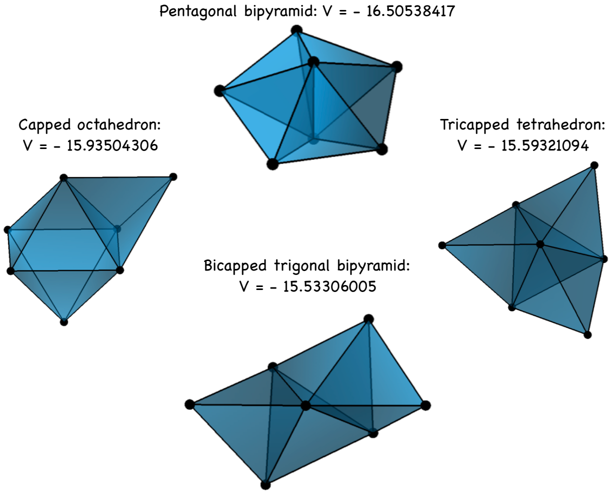
AMSC660 Maria Cameron

Homework 8. Due Oct. 28

1. **5 pts** Read the Wiki article about the Lennard-Jones potential. Consider 7 atoms in 3D interacting according to the Lennard-Jones pair potential. The total energy is given by

$$V = 4\sum_{i=2}^{7}\sum_{j=1}^{i-1}(r_{ij}^{-12} - r_{ij}^{-6}), \quad \text{where} \quad r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}.$$

We will distinguish only those atomic configurations that are not possible to map one into the other by translations, rotations and permutations of atoms. This system has 4 distinct in this sense local minima shown in figure below.



Matlab program `LJ7findmin.m` implements the line search Newton method. The initial approximations for each local minimum are defined so that all edges have the length $2^{1/6}$, the minimizer of the Lennard-Jones pair potential. Rotations and translations are eliminated by fixing atom 1 at the origin, allowing atom 2 to move only along thee x-axis, and allowing atom 3 to move only in the xy-plane. The gradient is calculated using analytic formulas. The Hessian for the Newton method is calculated using finite differences (because the analytic formulas for the Hessian are too much trouble to calculate☺). Whenever Newton's method gives and non-descend direction, the steepest descend direction is used instead. Note that Newton's method converges to two out of those four minima: pentagonal bipyramid and capped octahedron, while fails to converge to the other two local minima: instead, it converges to saddle points with energy values -15.31986444 and -15.28342128. This is an instance of erratic behavior of Newton's method.

(a) Randomly perturb the initial approximations using Gaussian random variables with mean zero and (i) variance $\sigma^2 = 0.01$ and (ii) $\sigma^2 = 0.05$. (Note, you need to perturb only `x` but not `xyz`.) Run the Newton method. Report, whether convergence will happen. If so, does the iterations converge to a minimum or to saddle? If to minimum, name the minimum. Note that the atoms might fly far apart. Then the value of the potential is almost 0. Write a summary of the results.

(b) Try to find the 4 local minima starting from the unperturbed initial approximations using the BFGS and SR1 line search methods. Whenever the proposed direction is not a descend direction, replace it with the steepest descend direction, and replace B with identity. Verify that the algorithms converge to the right local minima. If necessary, play with parameters of the algorithm. Print out your modified code. Write a report summarizing your observations and comparing the performances of Newton's, BFGS, and SR1 methods.

2. **5 pts** (Approx. problem 4.9 from Nocedal and Wright) Derive the solution of the two-dimensional constrained minimization problem

$$\min \ m(p) = f + g^T p + \frac{1}{2} p^T B p, \quad p \in \mathbb{R}^2, \quad \text{subjected to} \ \ p^T p \le \Delta^2$$

in the case where B is positive definite.

3. **10 pts** (Approx. problem 4.2 from Nocedal and Wright) Write a program that implements the trust-region strategy with the dogleg method. Choose B_k to be the(i) BFGS and (ii) SR1 approximations to the Hessian. Apply it to the Rosenbrock function

$$f(x,y) = 100(y - x^2)^2 + (1 - x)^2$$

starting with $(-1.2, 1)$. Repeat with the initial point $(0, 1)$. Print out and submit your code. Plot graphs of $\|\nabla f_k\|$ versus k in the logarithmic scale along the y-axis. Write a report summarizing your observations.

AMSC660 Maria Cameron

Homework 9. Due Nov. 4

1. **5 pts**

   (a) Show that Fletcher-Reeves nonlinear CG method with exact line searches

   $$\beta_{k+1}^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}, \quad p_{k+1} = -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$$

   all generated directions are descend directions.

   (b) Verify that the coefficients β for Polak-Ribiere and for Fletcher-Reeves nonlinear CG methods,

   $$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \quad \text{and} \quad \beta_{k+1}^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k},$$

   are identical in the case where $f(x)$ is a strongly convex quadratic function.

2. **10 pts** Compare the performances of Polak-Ribiere and Fletcher-Reeves nonlinear conjugate gradient methods (PRCG and FRCG) on the problem of finding local minima of the system of 7 atoms in 3D interacting according to the Lennard-Jones pair potential. The total energy of the system is given by

   $$V = 4\sum_{i=2}^{7}\sum_{j=1}^{i-1}(r_{ij}^{-12} - r_{ij}^{-6}), \quad \text{where} \quad r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}.$$

   For brevity, this system is denoted by LJ$_7$. Use PRCG+ ($\beta_{k+1} = \max\{0, \beta_{k+1}^{PR}\}$) and FRCG to search for local minima of LJ$_7$ starting from random initial configurations. Proceed as follows. Generate 10 random initial configurations. For each of them, run both, RPCG+ and FRCG. Check whether each of them converges, i.e., $\|\nabla V_n\| \to 0$. If both converge, check whether they converge to the same stationary point. Record the numbers of iterations and the numbers of V and ∇V evaluations ("fevals") for each of these methods. Make a table with the computed data and write a conclusion where you compare performances of these methods.

   Supplementary materials:

   - The matlab file `LJ7_NonlinCG_random_initial_conf.m` contains a code for generating random initial configurations, a function `[f, df] = LJ(xyz)` evaluating V and ∇V, a function `drawconf(xyz)` for visualizing atomic configurations, and templates for the CG methods.

- Minima of LJ_7: m1 : $V = -16.5053841680$; m2: $V = -15.9350430605$; m3: $V = -15.5932109382$, and m4: $V = -15.5330600546$. If you obtain another value of the potential V at the stationary point, this means that this point is a saddle.

3. **10 pts**

   Strain's Lecture 02, page 7, Exercise 4.

   (a) Check that the solution of the 2D gravity ODE with the initial conditions $x_0 = 1$, $y_0 = 0$, $u_0 = 0$, $v_0 = 1$ lies on the unit circle in the xy-plane, and the time of one revolution is 2π.

   (b) Write a program for the 2D gravity problem and perform tests described in Exercise 4. However, instead of the forward Euler method and the 2nd order Taylor method, implement the following methods:

   - Forward Euler: $u_{n+1} = u_u + hf(t_n, u_n)$, error $O(h)$;
   - Trapezoidal rule with Euler predictor, error $O(h^2)$:

   $$u^* = u_n + hf(t_n, u_n), \quad u_{n+1} = u_n + \frac{h}{2}(f(t_n, u_n) + f(t_{n+1}, u^*));$$

   - Midpoint rule with Euler predictor, error $O(h^2)$

   $$u^* = u_n + \frac{h}{2}f(t_n, u_n), \quad u_{n+1} = u_n + hf(t_n + \tfrac{h}{2}, u^*);$$

   - The 4-stage 4th order Runge-Kutta method (a.k.a. the classic Runge-Kutta, or RK4) (e.g. see Strain's Lecture 06, page 1), error $O(h^4)$;
   - The 3rd order 3-step Adams-Bashforth method, error $O(h^3)$:

   $$u_{n+1} = u_n + h\left(\frac{23}{12}f(t_n, u_n) - \frac{4}{3}f(t_{n-1}, u_{n-1}) + \frac{5}{12}f(t_{n-2}, u_{n-2})\right).$$

   (c) For the last two methods: Pick the time step $h = 0.01\pi$ and estimate the time interval T and the number of revolutions such that the numerical solution by each of these methods deviates from the unit circle by at least 0.1. Or do these methods produce numerical solutions always staying near the unit circle?

AMSC660 Maria Cameron

Homework 10. Due Nov. 11

1. (Approx. Strain's Lecture 3, exercises 4 and 5) The general form of linear explicit 2-step methods is

$$u_{n+1} = a_0 u_n + a_1 u_{n-1} + h(b_0 f_n + b_1 f_{n-1}).$$

   (a) Find the coefficients a_0, a_1, b_0, and b_1 that maximize the consistency order of this method. What is this order?

   (b) Show that the resulting method is unstable. To do so, apply it to the IVP $y' = 0$ with initial condition $y(0) = c$. Set $u_0 = c$ and $u_1 = c + \epsilon$. Derive the formula for u_n. Show that the numerical solution u at a fixed final time $T > 0$ blows up as we refine the time step h, i.e., $|u| \to \infty$ as $h = T/n \to 0$ for any fixed $\epsilon > 0$. *Hint: If you missed the Lecture on Nov. 2, look up how to solve linear discrete recurrence relations anywhere in the internet.*

   (c) Apply this method to the 2D gravity problem with the initial condition $x(0) = v(0) = 1$, $y(0) = u(0) = 0$, and show that the numerical solution blows up.

2. Strain's Lecture 4, page 4, exercise 1.

3. Strain's Lecture 4, page 6, exercise 3.

4. An explicit 3-stage 3$^{\text{rd}}$-order Runge-Kutta method (a. k. a. the Kutta method) is given by

$$\begin{aligned} k_1 &= f(t_n, u_n), \\ k_2 &= f(t_n + \tfrac{1}{2}h, u_n + \tfrac{h}{2}k_1), \\ k_3 &= f(t_n + h, u_n - hk_1 + 2hk_2), \\ u_{n+1} &= u_n + h\left(\tfrac{1}{6}k_1 + \tfrac{2}{3}k_2 + \tfrac{1}{6}k_3\right). \end{aligned}$$

   Consider the application of this method to the linear ODE $y' = \lambda y$, $y \in \mathbb{C}$. Let h be the time step. Then the numerical solution u_n can be written in the form of the recurrence relationship as

$$u_{n+1} = R(h\lambda)u_n.$$

   The function $R(z) \equiv R(h\lambda)$ is called the *stability function.* The *region of absolute stability* or *RAS* is defined as a region on the $z := h\lambda$ complex plane (h is the time step), where $u_n \to 0$ as $n \to \infty$, i.e.,

$$RAS := \{z \in \mathbb{C} \mid |R(z)| < 1\}.$$

   (a) Find the stability function for this method.

(b) Explain the relation between the stability function and the Taylor expansion for the exponent. *Hint: You can start reasoning as follows: the exact solution of* $y' = \lambda y$, $y(t_n) = y_n$, *is* $y(t_n + h) = y_n e^{\lambda h}$. *Then*

$$y_{n+1} \equiv y(t_n + h) = y_n e^{\lambda h} = y_n(1 + \lambda h + \tfrac{1}{2}(\lambda h)^2 + \dots)$$

Now look compare it with the recurrence relation for the numerical solution and think about its order of consistency.

(c) Hypothesize what the stability function for any explicit p-stage p^{th}-order Runge-Kutta method is.

AMSC 660 **Homework Assignment 11. Due Friday, Nov. 20 (sharp!)**

Fall 2015

1. Test the ODE solvers available in MATLAB on the Van Der Pol oscillator

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu((1-y_1^2)y_2) - y_1. \end{aligned}$$

with $\mu = 10^2$, 10^4, and 10^6. Set error tolerances $\epsilon \equiv atol \equiv rtol$, $\epsilon = 10^{-3}$, 10^{-6}, and 10^{-9}. Compute solution for one cycle. Plot the two solution components versus time t and plot y_2 versus y_1. Measure the CPU time required to compute one cycle for each of the solvers for each of the values of μ and plot $\log \epsilon$ versus $\log(CPUtime)$.

Please submit :

- Three figures, one for each μ, with the plots of $\log \epsilon$ versus $\log(CPUtime)$ for each of the solvers. Make sure to sign, which graph is for which solver.
- Report of your observations. Conclude, which solver is the most efficient for each μ and each ϵ. What can you say about the order of the methods implemented in the MATLAB ODE solvers? Are they constant or variable?

2. Construct embedded Runge-Kutta methods: (a) RK1(2) where the second order method is Runge-trapezoidal, (b) RK3(2) where the 3-rd order method is Kutta's method (see Table 1), (c) RK3(2) where the 3-rd order method is Heun's method (see Table 2). Determine which of these methods have the FSAL property (fist same as last).

3. Go to `http://www.unige.ch/∼hairer/software.html` and download `DOPRI5`. You can download the original fortran versions, of c versions, or MATLAB versions, whatever you prefer. *(The Matlab versions are found in "There is a folder with Matlab codes, written by Denis Bichsel "dbichsel (at) infomaniak.ch", for the nonstiff integrators DOPRI5 and DOP853.")* Go through the code and determine

   (a) What is the relationship between the vectors b and $\hat{b}$ of the Butcher array in Table 3 and the coefficients `a` and `er` in the code?

   (b) Is u_{n+1} or $\hat{u}_{n+1}$ is used for the next time step? Is u_{n+1} or $\hat{u}_{n+1}$ is as the output?

   (c) Explain how the error estimation is done in `DOPRI5`?

   (d) How is the step size chosen in `DOPRI5`? What is the formula and what are the values of parameters in it?

   (e) Plot the Region of Absolute Stability for `DOPRI5` (the coefficients regimen in Table 3).

4. The Brusselator is a model for a type of autocatalityc reaction (look it up in the Wikipedia):

$$\begin{aligned} y_1' &= 1 + y_1^2 y_2 - 4y_1, \\ y_2' &= 3y_1 - y_1^2 y_2. \end{aligned}$$

   Find appropriate initial conditions for the periodic solution. Compare CPU times for computing the solution for one period with tolerances $abstol = reltol = 1e-3, 1e-6, 1e-9, 1e-12$ by

   - `ode23` (Bogacki – Shampine method of the type RK3(2), 4 stages, look it up in the Wikipedia),
   - `ode45`,
   - `dop54` (should be the same as `ode45`, but minor details in the code might differ, and
   - `dop853`.

   The last two are the matlab codes for DOPRI5 and DOP853 from `http://www.unige.ch/∼hairer/software.html`. Submit a table with tolerances, CPU times and methods, a figure showing the solution in the phase plane, and a summary of your observations.

5. (a) Derive the interpolation polynomial $v(\theta)$, $\theta \in [0,1]$ for the data $u_0 = y(t_n)$, $f_0 = y'(t_n)$, $u_{1/2} = y(t_n + h/2)$, $u_1 = y(t_n + h)$, and $f_1 = y'(t_n + h)$.

   (b) Compare your result with the formula for the continuous output used in `DOPRI5`. Explain how $u_{1/2}$ is evaluated in the code.

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 1: Kutta's method

0			
$\frac{1}{3}$	$\frac{1}{3}$		
$\frac{2}{3}$	0	$\frac{2}{3}$	
	$\frac{1}{4}$	0	$\frac{3}{4}$

Table 2: Heun's method

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
y_1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
$\hat{y}_1$	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Table 3: Embedded Runge-Kutta Method: Dormand and Prince 5(4).

AMSC 660 **Homework Assignment 12. Due Wed., Dec. 2** Fall 2015

1. Plot the region of absolute stability for each of the methods given in the tables on page 5 of Strain's lecture 12. Compare your results with those on pp. 10-12 of Strain's lecture 13. You can use the matlab script ras.m written by J. Strain. Write a summary of your observations. For each of the three families of methods, Implicit Adams (a.k.a Adams-Moulton), explicit Adams (a.k.a. Adams-Bashforth), and BDF determine the range of orders for which you find their stability properfies satisfactory. Determine which BDF methods are $A(\alpha)$ stable and estimate α for them.

2. (a) Derive a three-step implicit Adams method with variable step size, i.e., assume $h_n = t_{n+1} - t_n$, $h_{n-1} = t_n - t_{n-1}$ and $h_{n-2} = t_{n-1} - t_{n-2}$ are different. In your answer, you can leave coefficients in the form of integrals and divided differences.

   (b) Now set all step sizes to be equal to h. Evaluate the divided differences, perform integrations and compare your result with the three-step Adams-Mouton method.

3. Prove that the 2-step variable step size BDF method (Strain's lecture 11, page 5) is stable iff $\omega_n \equiv h_n/h_{n-1} < 1 + \sqrt{2}$.

4. (a) The $2d \times 2d$ matrix

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

   is called *symplectic*. Find J^{-1} and J^2.

   (b) A linear map $A : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is called *symplectic* if for all $x, y \in \mathbb{R}^{2d}$ the 2-form

$$\omega(x,y) := x^T J y \quad \text{is conserved, i.e.,} \quad \omega(Ax, Ay) = \omega(x,y). \tag{1}$$

   Show that Eq. (1) is equivalent to

$$A^T J A = J. \tag{2}$$

   The geometric interpretation of the 2-form is explained in [1].

   (c) The Stoermer-Verlet method for integration of Hamiltonian systems of the form

$$\frac{dp}{dt} = -\nabla_q H(p,q), \quad \frac{dq}{dt} = \nabla_p H(p,q) \text{ or, equivalently, } \frac{d}{dt}\begin{bmatrix} p \\ q \end{bmatrix} = J^{-1}\nabla H(p,q)$$

   is given by

$$p_{n+1/2} = p_n - \frac{h}{2}\nabla_q H(p_{n+1/2}, q_n), \tag{3}$$

$$q_{n+1} = q_n + \frac{h}{2}\left(\nabla_p H(p_{n+1/2}, q_n) + \nabla_p H(p_{n+1/2}, q_{n+1})\right), \tag{4}$$

$$p_{n+1} = p_{n+1/2} - \frac{h}{2}\nabla_q H(p_{n+1/2}, q_{n+1}). \tag{5}$$

Rewrite this scheme for the case of a separable Hamiltonian, i.e., a Hamiltonian of the form $H(p,q) = T(p) + U(q)$. Show that it is an explicit scheme in this case. This scheme is also known as *velocity Verlet*. Apply the obtained scheme to the simple harmonic oscillator in 1D with the Hamiltonian

$$H(p,q) = \frac{p^2}{2m} + \frac{m\omega^2 q^2}{2}. \tag{6}$$

Rewrite the resulting equations in the form

$$\begin{bmatrix} p_{n+1} \\ q_{n+1} \end{bmatrix} = A \begin{bmatrix} p_n \\ q_n \end{bmatrix},$$

where A is a 2×2 matrix that you need to find.

(d) Show that the linear map given by the found matrix A is symplectic, i.e., $A^T J A = J$.

(e) The velocity Verlet scheme does not conserve the Hamiltonian given by Eq. (6). Prove that it conserves the so called *shadow Hamiltonian* given by

$$H^* = \frac{p^2}{2m} + \frac{1}{2} m\omega^2 q^2 \left(1 - \left(\frac{\omega h}{2}\right)^2\right). \tag{7}$$

This problem was inspired by [3], see slides 16 - 24.

5. Write the Hamiltonian equations for the 1D mathematical pendulum whose Hamiltonian is given by

$$H(p,q) = \frac{1}{2}p^2 - \cos(q).$$

Use the Stoermer-Verlet method to integrate the Hamiltonian equations and demonstrate that the oriented phase area is preserved by the method. In order to do it, evolve some shape in time (for example, you can use the cats face `cat.txt` as it is done in `symplectic_demo.m`, or you can create your own shape). Experiment with placing your shape around different initial points in the plane (p,q). Make figures similar to Fig. 1 in [2] (middle, right).

References

[1] Ernst Hairer, Geometric Numerical Integration. Lecture 1

[2] Ernst Hairer, Geometric Numerical Integration. Lecture 2

[3] M. P. Allen's talk "Molecular Simulation and Statistical Mechanics from a Practical Viewpoint" in the workshop "Computational methods for statistical mechanics - at the interface between mathematical statistics and molecular simulation", June 2 - 6, 2014, Edinburgh, Scotland