CS 322: Introduction to Scientific Computing
Spring 2003
Problem Set 6

Handed out: Wed., Apr. 23.

Due: Wed., Apr. 30 in lecture.

The policies for this (and other problem sets) are as follows:

- You should hand in your on-time problem set in lecture in the box at the front of the room on the day it is due. Problem sets handed in elsewhere (TA's office, Upson 303, etc.) will be considered late. See the next bullet.
- Late papers may be handed in up to 24 hours late. For instance, this problem set may be handed in up to 11:00 am on May 1. You can hand in a late paper in Upson 303. Late papers get an automatic deduction of 10%. The full late penalty is applied even if you turn in part of the solution on time.
- Problem sets may be done individually or in teams of two. Put your name or names on the front page. Re-read the academic integrity statement on the web for the policy concerning working in larger groups.
- Problem sets count for 20% of the final course grade. The lowest scoring problem set will be dropped.
- You need Matlab for some of the questions. Matlab is available in the following CIT labs: Upson and Carpenter.
- If you need clarification for a homework question, please either ask your question in section, lecture, or office hours or else post it to the newsgroup cornell.class.cs322. The professor reads this newsgroup and will post an answer.
- Write your section number (like this: "Section 2") at the top of the front page of your paper, and circle it. This is the section where your graded paper will be returned. As a reminder, Section 1 is Th 12:20, Section 2 is Th 3:35, Section 3 is F 2:30, Section 4 is F 3:35.

1. Consider computing π by finding a root of the function $f(x) = \sin x$ via Newton's method (for univariate equation solving). Write a Matlab program that implements this, and comment on whether it always converges, and in particular, on whether it converges to the root at π. (Presumably it will converge for a starting point sufficiently close to π). What happens with convergence when the starting point gets close to either $\pi/2$ or $3\pi/2$? Use as a termination test that two successive iterates have a relative difference less than or equal to machine-epsilon, or that an iterate is 0. Hand in listings of any m-files you write as well as answers to the questions raised above.

2. Molecules in a liquid tend to have a preferred distance because of "Vanderwaal's forces." In this problem we will optimize Vanderwaal's energy to find the preferred arrangement

of molecules on line. Given two molecules, one at position $x \in R$ and one at $y \in \mathbf{R}$, a simple model for the Vanderwaal's energy is $d^{-12} - d^{-6}$, where $d = x - y$.

(a) In a two-molecule system, suppose one molecule x is at 0. Find (in closed form) the optimal value(s) for the position y of the other molecule to minimize the Vanderwaal energy.

Suppose there are $n + 1$ molecules interacting under pairwise Vanderwaals energy. Use Newton's method to find the optimal (energy-minimizing) configuration. Let the positions of the molecules be $x_0, \ldots, x_n$, and assume that $x_0 = 0$, so that the Newton optimization problem has n variables. The free energy in this case is

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{n} \sum_{j=i+1}^{n} \left[(x_i - x_j)^{-12} - (x_i - x_j)^{-6} \right].$$

So write a Newton method to minimize this f.

Here are some suggestions. First, write three separate m-files, one to compute f, one to compute ∇f and one to compute $\nabla^2 f$. You can debug these files because you can check the relationship $f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x}) \approx \nabla f(\mathbf{x})\mathbf{h}$ (inner product on the right-hand side) where $\mathbf{h}$ is a small vector. Similarly, $\nabla f(\mathbf{x} + \mathbf{h}) - \nabla f(\mathbf{x}) \approx \nabla^2 f(\mathbf{x})\mathbf{h}$ (matrix-vector product on the right hand side). To assist the graders, please include a printout of $f(\mathbf{x})$, $\nabla f(\mathbf{x})$, and $\nabla^2 f(\mathbf{x})$ in the case that $\mathbf{x} = [.9; 2; 2.8]$ using "format long e". For your reference, these should come out as follows:

```
>> x = [.9;2;2.8];
>> vdw_f(x)
ans =
    1.211214883957694e+001
>> vdw_g(x)
ans =
   -3.433215233214385e+001
    1.893168852421100e+002
   -1.895998004341023e+002
>> vdw_h(x)
ans =
 6.056039641338642e+002 -2.148636570293965e+001  2.277736888042177e-001
-2.148636570293965e+001  3.318021653545106e+003 -3.296689828857792e+003
 2.277736888042177e-001 -3.296689828857792e+003  3.296451023921123e+003
```

From these three m-files for f, ∇f and $\nabla^2 f$, you can write a Newton method. Terminate when two successive iterates have a relative norm distance less than or equal to machine epsilon (i.e., `norm(newx-x)/norm(x)<=eps`).

The Newton method m-file should check if the Hessian is positive definite. Use the `chol` function for this purpose in its form with two return variables (type `help chol` for more info).

Find a starting configuration for which the method converges for $n = 5$, and one for which it doesn't converge or has a non-positive definite Hessian.

Finally, you should discover that at the optimum solution, the points are roughly evenly spaced (not exactly), with distance roughly $2^{1/6}$. Give some intuition why this is expected.

Hand in listings of m-files as well as all the items requested above.

3. Use `ode23` and `ode45` to integrate the IVP describing a frictionless pendulum:

$$\frac{d\theta}{dt} = \omega, \qquad \frac{d\omega}{dt} = -\sin\theta.$$

Here, θ is the angular displacement ($\theta = 0$ means the pendulum is in its rest vertical position) and ω is its angular velocity.

See the text, §9.2.3, for an example of how to use `ode23` and `ode45`. Also, type `help ode23` and `help ode45` for more information. For this problem, you need to specify only the first three arguments to these functions.

Choose initial conditions so that the pendulum starts from a 22.5° angle with no velocity (i.e., $\theta_0 = \pi/8$, $\omega_0 = 0$). Run both solvers from time 0 to time 500. (Until you finish debugging your program, a shorter interval, say $[0, 20]$, is recommended.) Plot θ as a function of t as computed by the two solvers, using two separate axes.

A frictionless pendulum is supposed to swing out the same distance on each swing. Is this what you observe from the solution? [Note: The slow drift in the swing length occurs because `ode23` and `ode45` do not conserve energy when applied to a conservative system, except in the limit as $h \to 0$. For conservative mechanical systems, it is preferable to use a *symplectic* method, which is not covered in CS322.]

Hand in: listings of all m-files, and the two plots of θ as a function of t as computed by `ode23` and `ode45`.