Introduction to Numerical Simulation (Fall 2003)
Problem Set #8 - due December 10 (last day of classes)

1) In this problem you will be using the *moldy* molecular dynamics program demonstrated in class. The software is posted on the course web site.

a) Use the moldy molecular dynamics software to calculate the isothermal compressibility of liquid argon at a temperature $T = 120K$ and a density of $\rho = 1300Kg/m^3$. The isothermal compressibility is defined as

$$k_T = \frac{1}{\rho}\frac{\partial \rho}{\partial P}$$

The experimental value of k_T at these conditions is $k_T \approx 2.6 \times 10^{-9} Pa^{-1}$.

b) How sensitive are your results to the potential cut-off used, and the use of a strict cut-off or a linked-cell list approach.

2) In this problem you will consider the difference between the local and global trunction error for boundary value problems.

a) Consider the equation

$$\frac{\partial^2 \psi}{\partial x^2} = 9\psi \tag{1}$$

on the interval $x \in [0, 1]$, with boundary conditions $\psi(0) = 0$ and $\psi(1) = 1$. Determine, analytically, the exact solution.

b) Construct a finite-difference discretization scheme for computing the solution to the equation in part (a) numerically. Compute the numerical solution to the equation using fixed spatial stepsizes of $h = 0.5, h = 0.1$ and $h = 0.01$.

c) Now consider the equation

$$\frac{\partial^2 \psi}{\partial x^2} = 9\frac{e^{3x} - e^{-3x}}{e^3 - e^{-3}} \tag{2}$$

on the interval $x \in [0, 1]$, with boundary conditions $\psi(0) = 0$ and $\psi(1) = 1$. Show that the exact solution to this problem is the same as the exact solution to the problem in part (a).

d) Construct a finite-difference discretization scheme for computing the solution to the equation in part (c) numerically. Compute the numerical solution to the equation using fixed spatial stepsizes of $h = 0.5, h = 0.1$ and $h = 0.01$.

e) Show that the *local truncation error*, that is the residue when the exact solution is substituted into the discrete equation, is the same for the problem in part (a) as discretized in part (b), and

the problem in part (c) as discretized in part (d). So, these two problems have the same analytic solution and the discretization methods used have the same local trunction error. However, the error in the numerically computed solutions is not the same for the two problems. What is different about the two problems, and how does that difference explain the difference in the numerical errors.

3) For this problem, we developed some extensive matlab code to help you understand boundary-element methods by having you use the method to determine the capacitance of single conductor. The problem is formulated in as followings:

How much charge must we put on a conductor to raise its voltage from zero volts to one volt.

We can solve for the charge density on the conductor surface by solving the integral equation

$$\psi(r) = \int_{surface} \frac{\sigma(r')}{\|r - r'\|} dS'$$

where the potential $\psi(r) = 1$ for all points r on the conductor surface and σ is the unknown charge density. We have developed a set of matlab codes that use the collocation method to compute the charge density. In this problem, you will examine the code and then convert the code to using a Galerkin method. The matlab code includes several files listed below.

calccap.m: This is the main routine which calculates capacitance.
calcp.m: This file contains the routine which analytically computes

$$\int_{panel} \frac{1}{\|r - r'\|} dS'$$

readpanels.m: This routines reads a set of panels describing the discretized geometry.
gencolloc.m: This routine computes panels centroids.
collocation.m: This routine sets up the collocation matrix.

We have also provided two examples, a square plate (files **plate9.qif**, **plate36.qif** and **plate144.qif**) and a sphere (files **sphere48.qif**, **sphere192.qif**, **sphere768.qif**) with three successively refined discretizations, as the file names imply. The plate is a unit square plate uniformly discretized into 9, 36 and 144 panels. The sphere is a unit sphere discretized into a nonuniform collection of 48, 192 and 768 panels. Once you have downloaded the files from the web, you can run an example by typing

[C, matrix] = calccap('plate36.qif')

and then you can also examine the matrix. Be forwarned that sphere768 takes a LONG time to run.

a) Examine the coefficient matrix produced by the collocation method, and explain why the matrix is nearly symmetric for the plate examples, but far from symmetric for the sphere examples.

b) The capacitance of a unit plate is approximately 40.8 picofarads. How rapidly is the collocation

method converging to that answer.

c) Once you have examined the collocation code, you can then probably see how to convert the code to using a Galerkin method. For this problem, we would like you to compare the accuracy of the Galerkin method to the collocation method for a given number of panels, but only for the square plate problem. We only want you to work with the square plate problem because in these problems the panels are all small squares.

To do the Galerkin calculation, you will need to compute the integral of a potential over a square panel. Do not try to come up with an analytic formula (though if you do, please let me know!). There are many ways to approximately calculate the required integral. If you are stuck for an approach, contact the TA for some help.

d) Suppose the collocation or Galerkin matrix equations are being solved by Gaussian elimination. Which will take longer for very large problems, forming the matrix or solving the matrix? Which takes longer for these small problems?

e) If you are using GCR to solve the collocation or Galerkin matrix equations, which approach would you prefer, Galerkin or collocation? In forming your Galerkin matrix, did you use a method that guarantees the matrix will be symmetric?

f) Suppose the collocation or Galerkin matrix equations are being solved using a Krylov-subspace method like GCR. Which will take longer for very large problems, forming the matrix or solving the matrix? What additional information do you need to answer this question?

g) The square plate examples are discretized into $m \times m$ grids of uniformly sized panels (3×3, 6×6, and 12×12 for the three examples above). Examine the coefficient matrix generated by your Galerkin version of calccap for these examples, and notice how many of the matrix entries have exactly the same value. As a function of m, how many unique values are there in the matrices associated with the uniformly discretized square plates? Please give an explanation for your results.

h) Make a special version of your Galerkinized calccap that only works for uniformly discretized square plates, and use the observation you made in (g) to more efficiently compute the coefficient matrix for this special set of problems.

i) Suppose you now try to solve the linear system for the charge vector q by using the GCR algorithm. Can you use the observation in (g) to substantially reduce the memory needed to solve uniformly discretized square plates?