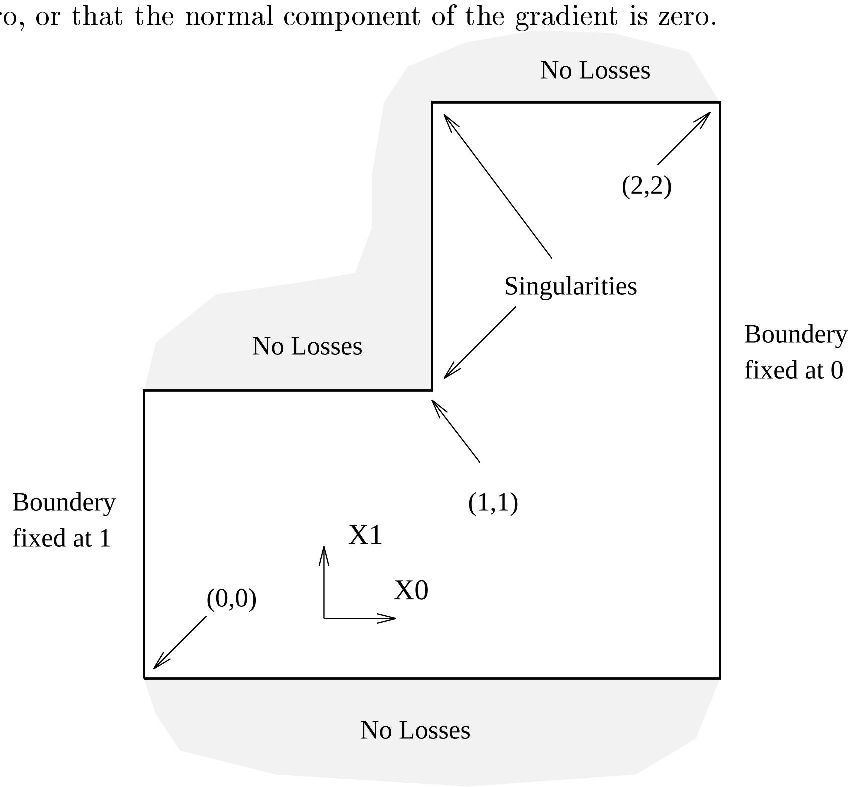
Date: January 20, 1998 Due Date: February 3, 1998

CS 522 Advanced Methods in Scientific Computing

Assignment 2

1 The Physical Problem

Solve for the steady state temperature profile for the conductive heat flow problem in the L-shaped region shown in the figure below. As you will recall, the general heat equation is a time-dependent parabolic problem. However, the steady-state heat equation reduces to solving Laplace's equation. As shown in the figure, there are two types of boundary conditions, the Dirichlet condition, when the value is given along a portion of the boundary, and the Neumann condition, in which there is no loss of heat outside of the boundary. A no loss condition is the same as saying the flux across the boundary is zero, or that the normal component of the gradient is zero.



2 Assignment

1. Write a program, function, or method (depending on rather you use Fortran-77, C, C++, or Matlab) to generate a "Finite Difference Grid" for the problem. This program should accept

a value of h as input. To generate a grid means to break up the solution domain into uniform subregions and specifying the node points.

It is acceptable to require that $1/h$ be an integer in order to avoid problems that arise when the boundary is not on a grid line.

2. Write a program, function, or method to convert this grid into a matrix problem using centered differences.

   You are expected to make this software specific to the Laplacian operator. Comment on what would be required to make equivalent software that would work for an arbitrary governing equation.

3. Solve the resulting linear system using a solver from LinPack, LApack, or some elementary method of your own devising (we will worry about efficient solvers later).

   Rearrange the result in a way that makes the next part possible.

4. Solve the problem (from end-to-end) using several different values of h. Graph these results. Discuss the relationship between the number of elements and the time to solve the linear system.

5. How much software would you have to change to rerun your software on the following geometry (you do not need to actually do the problem, just comment on the reusability of your methods).