MATH 309 — PROJECT

Due: Wednesday, December 1, 2004

Design a collection of computer programs for solving the minimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

using Steepest Descent, Newton's method, and the conjugate gradient method. For Newton implement the full step method only, Quasi-Newton (with line search) is optional. Your steepest descent program must include a line search, however. If you want to do extra work, consider implementing the DFP method. Test your program on a quadratic function f, and on at least one other problem whose exact solution you know. Try to get your programs working as soon as possible. It looks harder than it is.

You may work in groups of two or three students (no more than three are allowed - no excetions). It is sufficient to hand in one copy per group. You must also include a brief description on each person's contribution. Freeloaders are not acceptable, group work must be genuine. We may have some brief presentations of the projects in class (December 1, 3, and 6), and each group member must be prepared to make this presentation.

Presentation. Hand in an $8\frac{1}{2}'' \times 11''$ booklet, stapled in the upper left corner. Structure your paper like a report, with title, abstract, overview of the methods used, description of the building blocks. Describe the basic structure of your programs, and include the program listings in an appendix. Present your numerical results in condendsed form, using tables and/or graphs. Include pictures and plots where appropriate. The presentation is an important aspect of the project. It is important to include enough detail to get your message across, while avoiding to just include everything, or describing each line in the code.

Programs. Use double precision (16 digits in Maple). Your programs should be able to handle any n-dimensional minimization problem. Write your programs in modular form (easier to debug, and easier to reuse bits and pieces). Try to include plots, at least for $n = 2$. I strongly recommend you use MATLAB. You can the write your routines as M-files, and run various examples by supplying M-functions to evaluate the function, its gradient and Hessian matrix.

Test problems. Comment on any surprises or interesting phenomena you observe.

Algorithms. Steepest descent and Newton are pretty straightforward. I list here the Conjugate Gradient algorithm, once for the quadratic case, and for the nonlinear case.

Conjugate gradient method:
Quadratic case:
$k = 0$; $r_0 = b - Qx_0$
while $r_k \neq 0$
  $k = k + 1$
  if $(k = 1)$ $p_1 = r_0$
  else
    $\beta_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}$
    $p_k = r_{k-1} + \beta_k p_{k-1}$
  end if
  $\alpha = r_{k-1}^T r_{k-1} / p_k^T Q p_k$
  $x_k = x_{k-1} + \alpha_k p_k$
  $r_k = r_{k-1} - \alpha_k Q p_k$
end while

Conjugate gradient method:
Nonlinear case:
$k = 0$; $g_0 = \nabla f(x_0)$
while ($g_k \neq 0$) **and** $(k < n)$
  $k = k + 1$
  if $(k = 1)$ $d_0 = -g_0$
  else
    $\beta_{k-1} = g_{k-1}^T g_{k-1} / g_{k-2}^T g_{k-2}$
    $d_{k-1} = -g_k + \beta_{k-1} d_{k-1}$
  end if
  $\min f(x_{k-1} - \alpha d_{k-1})$ (Line search)
  $x_k = x_{k-1} - \alpha d_{k-1}$
  $g_k = \nabla f(x_0)$
end while
Set $x_0 = x_n$ and **restart**.

Project Part 1 Find the minimizer of

$$f(x, y) = x^2 + xy + 10y^2 - 22y - 5x$$

numerically by steepest descent, conjugate gradient and Newton. Newton should converge in one step, conjugate gradient in two steps. For steepest descent explore different starting values, such as (1,10), (10,10), (10,1). Does the number of steps depend significantly on the starting guess?

Project Part 2 We are solving a nonlinear system of equations

$$g(\mathbf{x}) = 0, \qquad g : \mathbb{R}^n \to \mathbb{R}^n,$$

by minimizing the sum of squares of the "residuals", i.e., by minimizing

$$f(\mathbf{x}) = \sum_{k=1}^{n} g_k(\mathbf{x})^2.$$

It is clear that any solution of $g(\mathbf{x}) = 0$ is a minimizer of $f(\mathbf{x})$, but not vice versa.

(a) Test your program on a simple linear system, e.g.

$$\begin{aligned} x - y - 1 &= 0 \\ 2x + y - 5 &= 0. \end{aligned}$$

(b) Find a solution of the nonlinear system of equations

$$\begin{aligned} \frac{12x}{y} - x + z^2 - 1 &= 0 \\ \frac{x}{y+12z} - 7y + \frac{1}{2} &= 0 \\ \frac{10y+2z+4}{x-1500} + \frac{5}{y-2} + \frac{1}{4z} - 1 &= 0 \end{aligned}$$

State which method you are using. Explore various initial guesses, and indicate whether you think the solution is unique.

(c) Apply your method also to the system

$$\begin{aligned} ye^{w+x} &= 2xz \\ w+x+y+z &= y^2 - z^2 \\ wxyz &= x-3 \\ \frac{1}{w+y} + \frac{2}{x+z} &= y. \end{aligned}$$

Project Part 3 Apply both of your programs to the *Extended Rosenbrock function* F_n defined below with $n = 2$, and $n = 4$, and initial guess $x^{(0)}$, and $10x^{(0)}$, respectively. Draw plots for the case $n = 2$. How many iterations are needed for each of the methods, and for the different starting values.

$$\begin{aligned} f_{2i-1}(x) &= 10(x_{2i} - x_{2i-1}^2), \qquad f_{2i}(x) = 1 - x_{2i-1}, \qquad i = 1, \ldots, \frac{n}{2}; \\ F_n(x) &= \sum_{j=1}^{n} f_j^2(x). \end{aligned}$$

$$x^{(0)} = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad \text{for } n = 2, \qquad x^{(0)} = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix} \quad \text{for } n = 4.$$

Project Part 4 OPTIONAL This problem is a "large-scale" quadratic problem with a

sparse matrix Q. Assume n is a multiple of 8; take C to be the following $n \times n$ matrix:

$$C = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \\ 0 & \dots & 0 & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{pmatrix},$$

(C is related to taking the second derivative of a periodic function), and define the vector ω by

$$\begin{aligned} \omega_j &= 1 + 16\left(\frac{4j}{n}\right)^2, \qquad j = 1, 2, \dots, \frac{n}{4}, \\ \omega_j &= 17, \qquad j = \frac{n}{4} + 1, \dots, n. \end{aligned}$$

Now set

$$Q = \begin{pmatrix} \omega_1 I_n + C & 0 & 0 & \dots & 0 & -I_n \\ 0 & \omega_2 I_n + C & 0 & 0 & \dots & \\ 0 & \ddots & \ddots & \ddots & \ddots & \\ & 0 & \ddots & \ddots & \ddots & \\ \vdots & 0 & \dots & 0 & \omega_{n-1} I_n + C & 0 \\ -I_n & 0 & \dots & & 0 & \omega_n I_n + C \end{pmatrix},$$

where I_n denotes the $n \times n$ identity matrix; note that Q is an $n^2 \times n^2$ matrix, with at most four nonzero entries in each row. It is basically a block diagonal matrix except for the two nonzero blocks in the first and last block row. For z take the n^2-vector with entries

$$z = [2\ 9\ 1\ 3\ 7\ 9\ 9\ \ 2\ 9\ 1\ 3\ 7\ 9\ 9\ \ \dots]^T,$$

(note the period-7 sequence in the vector), and finally set $b = Qz$.

Apply steepest descent, conjugate gradient, and Newton (which in effect amounts to solving $Qz = b$ directly) to the quadratic function

$$f(x) = \frac{1}{2} x^T Q x - b^T x,$$

with $x^{(0)} = 0$ as the starting vector. When implementing steepest descent and conjugate gradient method, **do not form** the matrix Q; instead, write a routine which for a given vector x returns the product Qx, taking advantage of the sparsity of Q.

Experiment with $n = 16$, $n = 32$, $n = 64$, and unless things get painfully slow, with $n = 128$, or as appropriate. You can use the MATLAB command *flops* to count operations, and compare the performance of the three methods.

Project Part 5 Enjoy! (not mandatory, but recommended)