MATH 923 — PROJECT 2

Report by Wednesday, October 10, 2000

Problem 1 These are some refresher exercises on multivariable calculus.

(i) For $f(x_1, x_2) = x_1^2 x_2^3 + x_1/x_2$ compute the gradient $\nabla f(x)$, and the Hessian matrix $\nabla^2 f(x)$, and find the quadratic approximation to f in a neighbourhood of the point $(1, 2)$.

(ii) Using the first-order necessary conditions, find a minimum point of the function

$$f(x, y, z) = 2x^2 + xy + y^2 + yz + z^2 - 6x - 7y - 8z + 9.$$

Verify that the stationary point you find is a minimum by checking the second derivative. Show that the minimum is a global minimum.

(iii) To approximate a function $h(x)$ over the interval $[0, 1]$ by a polynomial of degree at most n, we minimize the integral

$$f(\boldsymbol{a}) = \int_0^1 (h(x) - p(x))^2 \, dx,$$

where $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$. Find the equations satisfied by the optimal coefficients $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)^T$.

(iv) We define the function f on $\mathbb{R}^2$ by

$$f(x, y) = \begin{cases} \frac{x^3}{x^2+y^2}, & \text{if } (x, y) \neq (0, 0), \\ 0, & \text{if } (x, y) = (0, 0). \end{cases}$$

Show that all directional derivatives of f at the point $(0, 0)$ exist (you can do this, by investigating the functions $g_\theta(s) := f(s\cos\theta, s\sin\theta)$). However, f is **not** differentiable at $(0, 0)$. Why?

Problem 2 Consider the iterative process

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right),$$

where $a > 0$. Assuming the process converges, what is $\lim_{k\to\infty} x_k$? What is the order of convergence?

Write a short program to implement this algorithm (can be done in Matlab, Maple, or even on programmable pocket calculators). What are your observations on global and local convergence?

Problem 3 (i) Apply Newton's method to the cubic polynomial

$$p(x) = (x + 2)(x - 2)(x - 6).$$

Starting with $x_0 = 0$, which iterates are produced?

(ii) Can you generalize your observation from above to Newton's method applied to

$$p(x) = (x - z_1)(x - z_2)(x - z_3),$$

and $x_0 = \frac{1}{2}(z_1 + z_2)$?

Problem 4 Consider the problem of finding the real roots of each of the following three nonlinear functions of one unknown:

$$\begin{aligned} f_1(x) &= x^4 - 12x^3 + 47x^2 - 60x \\ f_2(x) &= x^4 - 12x^3 + 47x^2 - 60x + 24 \\ f_3(x) &= x^4 - 12x^3 + 47x^2 - 60x + 24.1 \end{aligned}$$

Discuss what can and cannot be expected from a computer program which solves this problem.

Apply Newton's method to the equation $f_1(x) = 0$. Start with $x_0 = 2$, and with $x_0 = 1$. Which root do you find? Comments?

Implement the secant method, and use it to solve $f_1(x) = 0$. How does its efficiency compare to Newton's method?

Problem 5 Using MATLAB (Maple, C/C++, Fortran, Pascal, etc. are o.k. as well) write three subroutines for minimizing a function of one variable. The basic description of the golden section algorithm is as follows. Set

$$\tau = (\sqrt{5} - 1)/2.$$

In the first step evaluate f at the points

$$\begin{aligned} \ell &= a + (b - a) * (1 - \tau) \text{ and} \\ r &= a + (b - a) * \tau. \end{aligned}$$

If $f(\ell) \leq f(r)$ then take $[a, r]$ as the new interval, otherwise $[\ell, b]$. Proceed recursively, noting that from now on only one additional function evaluation is required (for example, the previous point ℓ becomes the point r in the new smaller interval $[a, r]$).

- **Golden section search.** Implement the golden section search procedure. Input parameters are the real-valued function f, the search interval $[a, b]$, and a (relative) error tolerance tol.

- **Newton's method.** Implement Newton's method for solving $\min f(x)$, i.e., solve $f'(x) = 0$. Input parameters are now the functions f, f', and f'', an initial guess x_s, and an error tolerance tol.

- **Modified Newton.** Modify Newton's method to force it to find a minimizer. Reject a Newton step if the function value increases. This happens when you choose the "wrong" direction, or when you take too large a step. **Explain** and implement the following modification:

  $d_k = f'(x_k)/f''(x_k)$;
  $y = x_k - d_k$ (Newton step);
  while $f(y) > f(x_k)$ **do**,
      if $d_k * f'(x_k) < 0$ **then**
          $d_k = -d_k$;
      else
          $d_k = d_k/2$;
      end if
      $y = x_k - d_k$;
  end while
  $x_{k+1} = y$;

Test your programs on some simple problems.

Problem 6 Report the results of your programs applied to the functions

(i) $f(x) = x(\frac{6}{5} - x)$ on $[0, 1]$.

(ii) $f(x) = e^x + \frac{1}{2}x^2 - 2x$ on $[0, 1]$.

(iii) $f(x) = 1 - x\cos(\frac{\pi}{2}x)$ on $[0, 1]$.

(iv) $f(x) = x^3 - x^2 + 1$ on $[-1, 1]$.

Use an error tolerance of $5 \star 10^{-5}$. How do your routines compare in terms of number of function evaluations needed to achieve a preset accuracy?