II.3. BFGS. Implement algorithm 6.1 (BFGS method) in Matlab with exercises 5.1, 5.8, 7.1 and

From the book, the following exercises: 5.1–5.3, 5.6, 5.8, 5.11, 6.1–6.4, 6.6, 7.1–7.3, 7.5–7.6, 8.1, 8.3, 9.2(a), 9.8–9.9, 10.1–10.2, 10.6, 11.1–11.3, 11.5, 11.8, 11.10. In addition, the exercises below. The Matlab programming exercises are 5.1, 5.8, 7.1 and II.3–II.4.

For exercise 5.8, try 2 different matrices A generated as follows:

% Matrix 1: uniform spectrum
n = 100; l = 1:n; U = gallery('orthog',n); A = U*diag(l)*U';
% Matrix 2: clustered spectrum; try s = 5, 1, 0.001, 0
n = 100; n1 = floor(n/4);
l = [ones(1,n1) n/5*ones(1,n1) n*ones(1,n-2*n1)]; rand('state',314); l = l + s*rand(1,n);
U = gallery('orthog',n); A = U*diag(l)*U';

and compare your results with figs. 5.4–5.5.

Hint for exercise 6.1(a): a function is strongly convex on a convex domain if at each point in the domain all the eigenvalues of the Hessian are positive and bounded away from zero.

II.1. Conjugate directions. Let A be a real, symmetric, positive definite, n × n matrix with eigenvectors U = (u1, . . . , un) and associated eigenvalues Λ = diag(λ1, . . . , λn) (in matrix form). Show that: (i) If R = UA1/2Q, where Q is any orthogonal matrix, then A = RR′. (ii) If {v1, . . . , vn} are nonzero orthogonal vectors then {p1, . . . , pn} where pi = R−Tv is conjugate w.r.t. A (this shows there is an infinite number of conjugate direction sets). (iii) pi = ui are conjugate w.r.t. A (this corresponds to the change of variables x = S−1x with S = U). Hint: use the spectral theorem.

II.2. Quasi-Newton methods. Verify that, in 1D, all 3 quasi-Newton methods BFGS, DFP and SR1 are equivalent to the secant method, where Bk+1 = f′k+1−f′k f′k+1−xk independent of Bk.

II.3. BFGS. Implement algorithm 6.1 (BFGS method) in Matlab with H0 = I and backtracking line search (with initial step length 1). Apply it to the Rosenbrock function (2.23) from two initial points x0 = (1.2, 1.2) and x0 = (−1.2, 1.1) (cf. exercise 3.1). Tabulate ∥xk − x∗∥2, ∥∇f(xk)∥2 and ∥Bk − ∇2f(xk)∥F, where ∥·∥2 is the Euclidean norm (for vectors) and ∥·∥F the Frobenius norm (for matrices). Use your convseq function to estimate the convergence rate.

II.4. Newton-CG. Program a pure Newton iteration without line searches, where the search direction is computed by the CG method. Select stopping criteria such that the rate of convergence is linear, superlinear, and quadratic. Try your program on the following quartic function:

\[ f(x) = \frac{1}{2}x^TAx + \frac{1}{4}\sigma(x^TAx)^2 \]

where A is symmetric pd and σ ≥ 0 is a parameter that allows us to control the deviation from a quadratic. The starting point is x0 = (cos70°, sin70°, cos70°, sin70°)T. Try σ = 1 or larger values and observe the rate of convergence of the iteration. Use the convseq function you wrote in exercise I.4 to estimate the convergence rate.

Also, considering A is pd n × n and x ∈ Rn:

1. Prove that f is a strictly convex function (hint: prove that its Hessian is pd everywhere). Find all the stationary points of f and classify them into minima, maxima and saddle points.

2. Compute the cost in multiplications in O-notation of computing f(x), ∇f(x) and ∇2f(x) at a point x.

3. What would be the cost of computing ∇f(x) and ∇2f(x) approximately with finite differences if only function evaluations are allowed?