Gradient, Newton and conjugate direction methods for unconstrained nonlinear optimization

Consider the gradient method (steepest descent), with exact unidimensional search, the Newton method and the conjugate direction methods with the Fletcher-Reeves (FR) and Polak-Ribière (PR) updates.

Let $\varepsilon > 0$ be the given tolerance value. As a stopping criterion, use the condition $||\nabla f\left(\underline{x}_k\right)||_2 < \varepsilon$, where $\nabla f\left(\underline{x}_k\right)$ is the gradient of the function f computed in $\underline{x}_k$.

a) Implement the gradient method.

b) Implement the Newton method.

c) Implement the conjugate direction method with the Fletcher-Reeves (FR) and Polak-Ribière (PR) updates.

d) Homework: implement the quasi-Newton DFP method and compare it to the other methods on the two functions.

IMPLEMENTATION SKETCH

- FILE `descentmethod.m`: "stub" of a generic descent method. Find a local minimum $\underline{x}^*$ (with value $f(\underline{x})^*$) of the function f starting from an initial point $\underline{x}_0$, given a tolerance $\varepsilon > 0$ and an iteration limit. Return the number of iterations (`counter`) and the norm of the gradient $||\nabla f(\underline{x}^*)||_2$ in the last solution (`error`). The variables `xks` and `fks` contain the list of solutions found at each iteration and their corresponding objective function values (useful to represent graphically the results).

```
% Stub of a descent method
function [xk, fk, counter, error, xks, fks] = ...
    descentmethod(f, x0, epsilon, maxiterations)

  xks = [x0'];
  fks = [feval(f,x0)];

  xk = x0;
  counter = 0;
  error = 1e300;

  while error > epsilon && counter < maxiterations

    counter = counter + 1;
    % d =
    alpha = fminsearch(@(a) feval(f,xk + a*d), 0.0); %exact 1-d opt
    xk = xk + alpha*d;

    error = norm(grad(f,xk));
    fk = feval(f,xk);

    xks = [xks; xk'];
    fks = [fks; fk];

  end

end %of function
```

- FILE `grad.m`: numerical estimation of the gradient $\nabla f(\underline{x})$ of f in the point $\underline{x}$.

```
% Gradient of a function at a point
function gradf = grad(f, x)
  h = 0.0001;
  n = length(x);
  gradf = zeros(n,1);
  for i = 1:n
    delta = zeros(n, 1); delta(i) = h;
    gradf(i) = ( feval(f, x+delta) - feval(f,x) ) / h;
  end
end %of function
```

FILE `hes.m`: numerical estimation of the Hessian $\nabla^2 f(\underline{x})$ of f in $\underline{x}$.

```
% Hessian of a function at a point
function H = hes(f, x)

  h = 0.0001;
  n = length(x);
  H = zeros(n,n);

  for i = 1:n
    delta_i = zeros(n, 1); delta_i(i) = h;
    for j = 1:n
      delta_j = zeros(n, 1); delta_j(j) = h;
      H(i,j) = (feval(f,x+delta_i+delta_j) - feval(f,x+delta_i) ...
          - feval(f,x+delta_j) + feval(f,x)) / (h^2);
    end
  end

end %end of function
```

- FILE `contour_stub.m`: plots the level curves of the bivariate function `myFunction` on the domain $[x_{lb}, x_{ub}] \times [y_{lb}, y_{ub}]$.

```
f = @myFunction;
%plot the contour of the function
[X,Y] = meshgrid(x_lb:x_step:x_ub, y_lb:y_step:y_ub);
Z = zeros(size(X,1),size(X,2));
for i = 1:size(X,1)
    for j = 1:size(X,2)
       Z(i,j) = feval(f,[X(i,j);Y(i,j)]);
    end
end
figure; contour(X,Y,Z,50);
hold on;
```

Exercise 1: quadratic strictly convex function

Consider the problem:

$$\min f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{a}{2}x_2^2$$

with $a \geq 1$.

a) What do you know about the two conjugate direction variants on this kind of problem?

b) Solve the problem with $a = 4$ and $a = 16$, starting from $\underline{x}_0 = (a, 1)$, and compare the sequence of points generated by the four methods.

The code for the function with $a = 4$ is in f_ex_4.m

```
function f = f_ex1_4(x)
  f = 0.5*x(1)^2 + 4/2*x(2)^2;
end
```

and with $a = 16$ in f_ex_16.m

```
function f = f_ex1_16(x)
  f = 0.5*x(1)^2 + 16/2*x(2)^2;
end
```

Exercise 2: Rosenbrock function

Consider the following Rosenbrock function

$$f(\underline{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

It is a quadratic function, nonconvex, often used to test the convergence of nonlinear optimization algorithms.

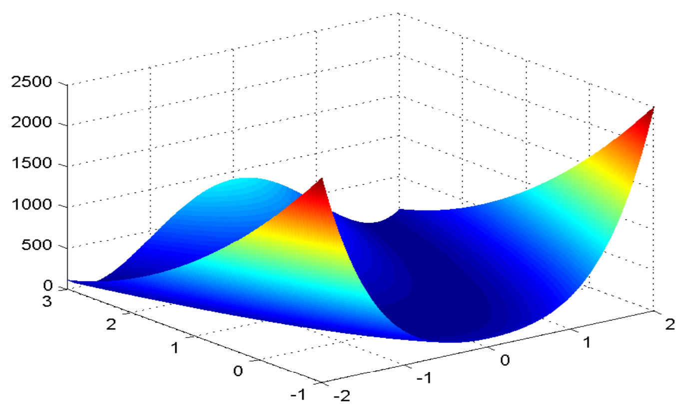

Figure 1: Rosenbrock function

a) Find analytically the only global optimum.

b) Observe the behavior of the gradient method, starting from the initial points: $\underline{x}'_0 = (-2, 2)$, $\underline{x}''_0 = (0, 0)$ and $\underline{x}'''_0 = (-1, 0)$, using a maximum number of 50, 500 and 1000 iterations.

c) Compare the sequence of solutions found by the four methods.

- FILE `f_rosenbrock.m`

```
function f = f_rosenbrock(x)
  f = 100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
end
```