Résolution d'un problème d'optimisation différentiable

Marc BOURQUI, Victor CONSTANTIN, Ian SCHORI, Floriant SIMOND

December 21, 2012

Énoncé du problème

Trouver (une approximation de) la solution du problème suivant en appliquant le théorème de la plus forte pente:

$$\min_{x \in \mathbb{R}^2} (x_1 - 2)^4 + (x_1 - 2)^2 x_2^2 + (x_2 + 1)^2 \tag{1}$$

Réponses aux questions

(a) Implémenter la méthode de plus forte pente (Algorithme 11.3) à l'aide du logiciel MATLAB. Déterminer la taille du pas en appliquant la recherche linéaire, Algorithme 11.2 (les deux conditions de Wolfe).

Listing 1: pfp.m

```
function x = pfp(f, x0, epsilon)
    x = x0;
    alpha = 1;

    beta1 = 0.5;
    beta2 = 0.75;
    lambda = 2;

    iteration = 1;
    [~, gfx] = feval(f, x);
    while abs(gfx) > epsilon
        fprintf('Iteration number %d : x = [%f, %f]\n', ...
            iteration, x(1), x(2));

        d = -gfx;
        %Soit on peut utiliser la fonction dans b) pour ...
            calculer le pas
        alpha = tp(f,x);
        %alpha = rl(f, x, d, alpha, beta1, beta2, lambda);
        x = x + alpha * d;
        [~, gfx] = feval(f, x);

        iteration = iteration + 1;
    end
end
```

Listing 2: rl.m

```
function alpha = rl(f, x, d, alpha0, beta1, beta2, lambda)
    alpha = alpha0;
    alphal = 0;
```

```
    alphar = inf;

    [fx, fgx] = feval(f, x);

    [fxad, fgxad] = feval(f, x + alpha * d);

    while (fxad > fx + alpha * beta1 * fgx' * d) || ...
        (fgxad' * d < beta2 * fgx' * d)
        if fxad > fx + alpha * beta1 * fgx' * d
            alphar = alpha;
            alpha = (alphal + alphar)/2;
        elseif fgxad' * d < beta2 * fgx' * d
            alphal = alpha;
            if alphar < inf
                alpha = (alphal + alphar)/2;
            else
                alpha = lambda * alpha;
            end
        end

        [fxad, fgxad] = feval(f, x + alpha * d);
    end
end
```

(b) Implémenter une fonction qui donne la taille du pas suivant:

$$\alpha_k = \frac{\bigtriangledown f(x_k)^T \bigtriangledown f(x_k)}{\bigtriangledown f(x_k)^T \bigtriangledown^2 f(x_k) \bigtriangledown f(x_k)} \tag{2}$$

Quelle est la nature de ce pas? D'où cette formule vient-elle?

(c) Comparer le comportement de l'algorithme en utilisant les pas (a) et (b).

(d) Comparer la methode de plus forte pente et la methode quasi-Newton (qui est déjà implementée – Série 3).