Universität Konstanz Sommersemester 2013
Fachbereich Mathematik und Statistik
Prof. Dr. Stefan Volkwein
Roberta Mancini, Stefan Trenz, Carmen Gräßle, Marco Menner, Kevin Sieg

Optimierung

http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/

Program 3

Submission by E-mail: 2013/07/10, 10:00am

Implement the **Trust-Region method** according to Algorithmus 5.8 (Trust-Region-Framework) and Algorithmus 5.7 in the lecture notes and test it using the Rosenbrock function. Write therefore a function

```
function X = trustregion(fhandle,x0,r0,epsilon,N)
```

with the function handle `fhandle`, an initial point `x0`, an initial trust region radius `r0`, a termination tolerance `epsilon` and a maximum number of iteration steps `N`. The function should return a matrix `X = [x0; x1; x2; ...]` containing the whole iterations.

To ease the implementation modify the Algorithmus 5.8 in the following way:

- change the termination condition to $\|\nabla f(x)\| > \epsilon$ instead of $\|\nabla f(x)\| > \tau_r \tau_0 + \tau_a$,
- use the exact Hessian matrix of the Rosenbrock function, which you should provide using `fhandle`, together with the function and gradient values,
- instead of solving the minimization problem to obtain d_V^k you should compute the Cauchy point x_a^{CP} to get a candidate for the next iteration $x_V^k = x_a^{CP}$.

To compute the Cauchy point x_a^{CP} solve the subproblem described on pages 24 and 25 in the lecture notes. Write therefore a function

```
function t = trsubproblem(fhandle,x,r)
```

with the function handle `fhandle`, a current point `x` and the current trust region radius `r` that returns a step size `t`=t_a and compute $x_a^{CP} = x_a - t_a \nabla f(x_a) =: x_V^k$.

To compute (x^{k+1}, Δ_{k+1}) implement Algorithmus 5.7 according to the lecture notes by writing a function

```
function [x,delta]= trupdate(fhandle,x,xv,r)
```

with the function handle `fhandle`, a current point `x`, a current candidate `xv` and the current trust region radius `r`. The parameters used in this function should be `C=1.0`, `mu_upper=0.9`, `mu_lower=0.1`, `mu_zero=0.01`, `omega_upper=2` and `omega_lower=1/2`.

Test your program by using the Rosenbrock function in the following cases:

1. x0=[-5;-5], r0=1.0
2. x0=[-5;-5], r0=0.1
3. x0=[-5;-5], r0=0.01

using epsilon=1.0e-3 and N=10000. Describe your observations in the written report and generate also a contour plot of the Rosenbrock function in which you display the computed iteration points.

Please, refer to the following Algorithms for developing the your Program.

Algorithm 5.7

`Input:` $x_a \in \mathbb{R}^n,\ x_V \in \mathbb{R}^n,\ \Delta \in \mathbb{R}^+$

`Begin`

$z^0 := x_a;\ z_V^0 := x_V;\ \hat{\Delta}^{(0)} := \Delta;\ l := 0$

`While` $z^l = x_a$

$\text{ared}^{(l)} := f(x_a) - f(z_V^l);\ d_V^l := z_V^l - x_a;$
$\text{pred}^{(l)} := -\nabla f(x_a)^T d_V^l - \frac{1}{2}(d_V^l)^T H_a d_V^l$

`If` $\frac{\text{ared}^{(l)}}{\text{pred}^{(l)}} < \mu_0$

$z^{l+1} := x_a,\ \hat{\Delta}^{(l+1)} := \underline{\omega}\hat{\Delta}^{(l)}$

`If` $l \geq 1\ \&\ \hat{\Delta}^{(l)} > \hat{\Delta}^{(l-1)}$

$z^{l+1} := z_V^{l-1},\ \hat{\Delta}^{(l+1)} := \hat{\Delta}^{(l-1)}$

`Else`

compute the solution d_V^{l+1} of the T-R subproblem with radius $\hat{\Delta}^{(l+1)}$
$z_V^{l+1} := x_a + d_V^{l+1}$

`End(If)`

`Elseif` $\mu_0 \leq \frac{\text{ared}^{(l)}}{\text{pred}^{(l)}} \leq \underline{\mu}$

$z^{l+1} := z_V^l;\ \hat{\Delta}^{(l+1)} := \underline{\omega}\hat{\Delta}^{(l)}$

`Elseif` $\underline{\mu} \leq \frac{\text{ared}^{(l)}}{\text{pred}^{(l)}} \leq \overline{\mu}$

$z^{l+1} := z_V^l,\ \hat{\Delta}^{(l+1)} := \hat{\Delta}^{(l)}$

`Elseif` $\overline{\mu} \leq \frac{\text{ared}^{(l)}}{\text{pred}^{(l)}}$

`If` $\|d_V^l\| = \hat{\Delta}^{(l)} \leq C\|\nabla f(x_a)\|$

$z^{l+1} := x_a;\ \hat{\Delta}^{(l+1)} := \overline{\omega}\hat{\Delta}^{(l)}$
compute the solution d_V^{l+1} of the T-R subproblem with radius $\hat{\Delta}^{(l+1)}$
$z_V^{l+1} := x_a + d_V^{l+1}$

`Else`

$z^{l+1} := z_V^l,\ \hat{\Delta}^{(l+1)} := \hat{\Delta}^{(l)}$

`End(If)`

`End(If)`

$l := l + 1$

`End(While)`

$x_+ := z^l;\ \Delta_+ := \hat{\Delta}^{(l)}$

`End`

Algorithm 5.8 (Trust-Region Framework)

`Input:` $x^0 \in \mathbb{R}^n;\ \Delta_0 \in \mathbb{R}_+$

`Begin`

$k := 0;\ \tau_0 := \|\nabla f(x^0)\|$

`While` $\|\nabla f(x^k)\| > \tau_r \tau_0 + \tau_a$

compute an approximation H^k of the Hessian $\nabla^2 f(x^k)$

compute d_V^k as solution of

$$\min f(x^k) + \nabla f(x^k) d + \frac{1}{2} d^T H^k d \text{ u.d.N. } \|d\| \le \Delta_k$$

compute (x^{k+1}, Δ_{k+1}) with Algorithm 5.7 with `Input` $x^k;\ x_V^k := x^k + d_V^k;\ \Delta_k$

$k := k + 1$

`End(While)`

`End`