



20th June 2011

Optimization Programming 3

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 well known 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 $\|\nabla f(x)\| > \tau_r \tau_0 + \tau_a$.
- You don't need to approximate the Hessian matrix of the Rosenbrock function (e.g. calculate it by hand and implement it).
- 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 $\tau = t_a$ and compute $x_a^{CP} = x - t_a \nabla f(x_a) =: x_V^k$.

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 \mathbf{x} , a current candidate \mathbf{xv} and the current trust region radius r . Here again you should solve the Cauchy point subproblem at the respective positions in the algorithm instead of the minimization problem.

For the parameters in the algorithm use the following configuration:

- $\mu_0 = 0.01, \underline{\mu} = 0.1, \bar{\mu} = 0.9$
- $\underline{\omega} = 1/2, \bar{\omega} = 2$
- $C = 1.0$

Test your program by using the Rosenbrock function with following parameters and call it from a main-file:

`epsilon=1.0e-3, N=1.0e4` and

- a) `x0=[-5;-5], r0=1.0`
- b) `x0=[-5;-5], r0=0.1`
- b) `x0=[-5;-5], r0=0.01`

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.

Note: Please document your programming and the used variables in your code in an adequate way.

Deadline: Monday, 28th May, 8:30 am

Attached you find a corrected version of Algorithmus 5.7 from the lecture.

ALGORITHMUS 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

berechne die Lösung d_V^{l+1} des T.-R.-Hilfsproblems mit 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 $\underline{\mu} \leq \frac{\text{ared}^{(l)}}{\text{pred}^{(l)}} \leq \bar{\mu}$

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

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

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

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

berechne die Lösung d_V^{l+1} des T.-R.-Hilfsproblems mit 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