

Ctirad Matonoha

Nonlinear optimization problem

In: Jan Chleboun and Petr Příkryl and Karel Segeth (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 6-11, 2004. Institute of Mathematics AS CR, Prague, 2004. pp. 162-168.

Persistent URL: <http://dml.cz/dmlcz/702790>

**Terms of use:**

© Institute of Mathematics AS CR, 2004

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*  
<http://dml.cz>

# NONLINEAR OPTIMIZATION PROBLEM

Ctirad Matonoha

## 1. Introduction

In this contribution, we are concerned with a general nonlinear optimization problem [7]: find a local minimum  $x_\star \in \mathbb{R}^n$  of function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $f \in \mathcal{C}^2$  :

$$x_\star = \arg \min_{x \in \mathbb{R}^n} f(x). \quad (1)$$

The main optimization method for solving this problem has an iterative character. After choosing initial point  $x_0$  we construct a sequence of points  $\{x_k\}_{k \in \mathbb{N}_0}$ , so that

$$x_{k+1} = x_k + \alpha_k d_k, \quad (2)$$

where  $d_k$  is a direction vector and  $\alpha_k$  is a steplength. A global convergence is an important assumption for the application of optimization methods. It means satisfying the condition

$$\liminf_{k \rightarrow \infty} g_k = 0, \quad (3)$$

where  $g_k \equiv g(x_k) = \nabla f(x_k)$ .

## 2. Trust region methods

Trust region methods [1], [8] belong to the most effective optimization methods because of their very good convergence properties. They are globally and super-linearly convergent [3], [4] and moreover they use a simple choice of the steplength. Before the definition of trust region methods we'll introduce the following notation:

$$\psi_k(d) = \frac{1}{2} d^T B_k d + g_k^T d$$

for a quadratic function that locally approximates the difference  $f(x_k + d) - f(x_k)$ ,

$$w_k(x) = \frac{B_k d + g_k}{\|g_k\|}$$

for a vector used in the determination of the accuracy of a direction vector, and

$$\varrho_k(x) = \frac{f(x_k + d) - f(x_k)}{\psi_k(d)}$$

for a quotient of an actual and predicted reduction in function  $f$ . Matrix  $B_k$  is either the Hessian matrix  $G_k \equiv G(x_k) = \nabla_{xx}^2 f(x_k)$  or its suitable approximation, and the norm is Euclidean.

**Definition 1** Optimization method (2) is a trust region method if direction vectors  $d_k \in \mathbb{R}^n$ ,  $k \in \mathbb{N}_0$ , are chosen so that

$$\|d_k\| \leq \Delta_k, \quad (4)$$

$$\|d_k\| < \Delta_k \Rightarrow \|w_k(d_k)\| = \omega_k \leq \bar{\omega}, \quad (5)$$

$$\psi_k(d_k) \leq -\underline{\sigma} \|g_k\| \min \left\{ \|d_k\|, \frac{\|g_k\|}{\|B_k\|} \right\}, \quad (6)$$

where  $\bar{\omega} < 1$ ,  $0 < \underline{\sigma} < 1$ . Steplengths  $\alpha_k \geq 0$ ,  $k \in \mathbb{N}_0$ , are chosen so that

$$\varrho_k(d_k) \leq 0 \Rightarrow \alpha_k = 0, \quad (7)$$

$$\varrho_k(d_k) > 0 \Rightarrow \alpha_k = 1. \quad (8)$$

Sequence  $\Delta_k > 0$ ,  $k \in \mathbb{N}_0$ , is constructed so that

$$\varrho_k(d_k) < \underline{\varrho} \Rightarrow \underline{\beta} \|d_k\| \leq \Delta_{k+1} \leq \bar{\beta} \|d_k\|, \quad (9)$$

$$\varrho_k(d_k) \geq \underline{\varrho} \Rightarrow \Delta_k \leq \Delta_{k+1} \leq \bar{\gamma} \Delta_k, \quad (10)$$

where  $0 < \underline{\varrho} < 1$ ,  $0 < \underline{\beta} \leq \bar{\beta} < 1 < \bar{\gamma}$ . Matrices  $B_k$ ,  $k \in \mathbb{N}_0$ , are constructed so that they satisfy the condition

$$\|B_k\| \leq M \quad \forall k \in \mathbb{N}_0, \quad (11)$$

where constant  $M < \infty$  is independent of  $k \in \mathbb{N}_0$ .

The algorithm can be written in the following way.

**Algorithm 1** Trust region method.

Choose  $x_0 \in \mathbb{R}^n$ ,  $0 \neq B_0 \in \mathbb{R}_S^{n \times n}$ ,  $\Delta_0 > 0$ ,  $\varepsilon > 0$ , compute  $f(x_0)$ , set  $k = 0$ .

1. Compute gradient  $g(x_k)$ . If  $\|g(x_k)\| < \varepsilon$ , then STOP.

2. Determine vector  $d_k$  satisfying conditions (4)-(6).

3. Set  $x_k^+ = x_k + d_k$  and compute values  $f(x_k^+)$  and  $\varrho_k(d_k) = \frac{f(x_k^+) - f(x_k)}{\psi_k(d_k)}$ .

4. If  $\varrho_k(d_k) < \underline{\varrho}$ , update  $\Delta_{k+1}$  according to (9).  
If  $\varrho_k(d_k) \geq \underline{\varrho}$ , update  $\Delta_{k+1}$  according to (10).

5. If  $\varrho_k(d_k) \leq 0$ , go to step 2.

If  $\varrho_k(d_k) > 0$ , update matrix  $B_{k+1} \neq 0$  so that it satisfies condition (11), set  $x_{k+1} = x_k^+$ ,  $f(x_{k+1}) = f(x_k^+)$ ,  $k := k + 1$  and return to step 1.

Trust region methods lead to a subproblem of finding minimum  $d_k$  of quadratic function  $\psi_k(d)$  subject to constraint  $\|d\| \leq \Delta_k$ . As this subproblem is solved iteratively for fixed  $k$ , this index can be omitted. If  $d_*$  is a resulting direction vector in step  $k$ , the subproblem is as follows:

$$d_* = \arg \min \psi(d) \equiv \frac{1}{2} d^T B d + g^T d, \quad \|d\| \leq \Delta. \quad (12)$$

Trust region methods are a class of methods that must satisfy conditions of Definition 1. The first class seeks an optimal step on the whole space  $\mathbb{R}^n$ . As computing such a vector is a difficult task, there exists the second class of methods that seeks only an approximation of the optimal step on some subspaces (e.g. Krylov subspaces) of  $\mathbb{R}^n$ . Conditions (4)-(6) are the only ones that computed optimal or approximated solution  $d_*$  must satisfy.

### 3. Direction determination

#### 1. Use of a Cholesky decomposition and Newton's method

We exploit optimality conditions [2] that characterize the optimal step  $d_*$  :

$$\|d_*\| \leq \Delta, \quad B + \xi_* I \succeq 0, \quad \xi_* \geq 0, \quad (B + \xi_* I)d_* = -g, \quad (\|d_*\| - \Delta) \xi_* = 0 \quad (13)$$

and after denoting by  $\lambda_1$  the smallest eigenvalue of  $B$  we solve the equation

$$\phi(\xi) \equiv \frac{1}{\Delta} - \frac{1}{\|d\|} = 0, \quad \xi > \max\{0, -\lambda_1\}$$

for  $\xi_*$  by Newton's method and factorization  $B + \xi I = R^T R$ .

#### 2. Use of a linear combination of eigenvectors

Let  $B = Q D Q^T$  be the eigendecomposition of  $B$ ,  $Q = (q_1, \dots, q_n)$ . Then

$$d_* = \sum_{i=1}^n c_i q_i,$$

where  $c = (c_1, \dots, c_n)^T$  is a certain vector such that  $d_*$  satisfies optimality conditions (13).

#### 3. The dogleg method

$$d_* \in \text{sp}\{g, B^{-1}g\}$$

A simple piecewise linear curve is generated, where Cauchy and Newton points  $d_C = -\frac{g^T g}{g^T B g} g$  and  $d_N = -B^{-1}g$  are considered.

#### 4. Combination of the dogleg method and a Cholesky decomposition

$$d_* = \tau_1 g + \tau_2 B^{-1}g$$

Vector  $\tau_* = (\tau_1, \tau_2)^T$  is the optimal step on subspace  $\mathbb{R}^2$ . As this is a simple problem, a Cholesky decomposition and Newton's method approaches are used.

### 5. Use of the conjugate gradient method

$$d_\star \in \mathcal{K}_{i+1} = \text{sp}\{g, Bg, B^2g, \dots, B^i g\}$$

Vector  $d_\star$  is a linear combination of  $B$ -orthogonal basis of Krylov subspace  $\mathcal{K}_{i+1}$ . A piecewise linear curve, whose final points are conjugate gradient iterations, is generated. Sequence  $\{\|d_j\|\}_{j=0,1,\dots,i}$  is monotonically increasing while sequence  $\{\psi(d_j)\}_{j=0,1,\dots,i}$  is monotonically decreasing.

### 6. The preconditioned conjugate gradient method

For large scale sparse systems there is suitable to use preconditioning of CGM with a symmetric and positive definite matrix  $C$ . We perform an incomplete decomposition  $B \approx R^T R$ , where  $R$  has the same nonzero elements structure as matrix  $B$ , and set  $C = R^T R$ .

### 7. Combination of the conjugate gradient and dogleg methods

As the dogleg method uses just one step of conjugate gradient method, we can generalize this process with using more steps of CGM.

### 8. Use of the Lanczos method

Vector  $d_\star \in \mathcal{K}_{i+1}$  is a linear combination of orthonormal basis,  $d_\star = Q_i h_i$ , where  $Q_i^T B Q_i = T_i$  is a matrix form of the Lanczos method. Such an approach leads to subproblem

$$h_i = \arg \min \tilde{\psi}(h) \equiv \frac{1}{2} h^T T_i h + \tilde{g}^T h, \quad \|h\| \leq \Delta, \quad (14)$$

where  $\tilde{g} = (\|g\|, 0, \dots, 0)^T$ , that is solved by a Cholesky decomposition and Newton's method. It is a simple problem because matrix  $T_i$  is tridiagonal. Thus a sequence of approximations of the optimal step is generated. Unfortunately we cannot use preconditioning because we would lose the orthogonality of the original basis vectors and change a constraint of quadratic subproblem (12).

### 9. Two new modifications of the Lanczos method

#### (a) Combined "Lanczos – conjugate gradient" method

We choose fixed  $m$  (usually small) and compute  $m$  steps of the Lanczos method to obtain tridiagonal matrix  $T_{m-1}$  of order  $m$ . Now we solve tridiagonal quadratic subproblem (14) for  $h_{m-1} \in \mathbb{R}^m$  with a Cholesky decomposition of matrix  $T_{m-1} + \xi I_m$  to obtain parameter  $\xi_\star > 0$ . Such a value is sufficient in computing an approximate solution of original subproblem (12). Finally, equation  $(B + \xi_\star I)d + g = 0$  is solved by the (preconditioned) conjugate gradient method. We'll get a better approximation of the trust region step than in case  $\xi_\star = 0$ .

(b) *Combined “conjugate gradient – Lanczos” method*

We choose a fixed  $m$  (usually small) and compute  $m$  steps of the conjugate gradient method to generate Lanczos vectors. Tridiagonal matrix  $T_{m-1}$  of order  $m$  is obtained and now we proceed as follows. If  $\|d_m\| < \Delta$ , we continue with the conjugate gradient method till end because matrix  $T_{m-1}$  is no longer updated. If  $\|d_m\| \geq \Delta$ , we solve tridiagonal quadratic subproblem (14) for  $h_{m-1} \in \mathbb{R}^m$  with a Cholesky decomposition of matrix  $T_{m-1} + \xi I_m$  and set  $d_\star = Q_{m-1}h_{m-1}$ . However, with this method we cannot use preconditioning for the same reasons as above.

### 10. Parametric eigenvalue problem

We exploit optimality conditions (13) that characterize optimal step  $d_\star$ , define parameter  $\tau \in \mathbb{R}$ , construct matrix

$$A_\tau = \begin{pmatrix} \tau & g^T \\ g & B \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$$

and observe that

$$\frac{\tau}{2} + \psi(d) = \frac{1}{2} (1, d^T) A_\tau (1, d^T)^T.$$

Therefore, we can rewrite subproblem (12) as

$$\min_{y \in \mathbb{R}^{n+1}} \frac{1}{2} y^T A_\tau y, \quad y^T y \leq 1 + \Delta^2, \quad e_1^T y = 1, \quad \text{where } e_1 = (1, 0, \dots, 0)^T.$$

This formulation suggests that we can find the desired solution in terms of an eigenpair of  $A_\tau$ . The main goal is to drive parameter  $\tau$  to an optimal value  $\tau_\star$ , so that the eigenvector associated with the smallest eigenvalue of  $A_{\tau_\star}$  has the form  $y = (1, d_\star)^T$ , where  $d_\star$  is the optimal step for subproblem (12).

## 4. Numerical experiments

Algorithms were implemented in UFO [5] and tested on two collections of large scale structured testing problems with 22 optimization problems without constraints [6] for 1000 and 5000 unknowns. The results are presented in Tables 1-2, where:

- N – number of the method used
- Method – initials of the method used
- P – type of preconditioning: 0, 1, 2 (see below)
- NIT – total number of leading iterations ( $x_k$  in algorithm 1)
- NFV – total number of evaluations of function  $f$
- NFG – total number of evaluations of gradient  $g = \nabla f$
- NCG – total number of CGM iterations (inner iterations for  $d_k$ )
- T – total time

| <b>N</b> | <b>Method</b> | <b>P</b> | <b>NIT</b> | <b>NFV</b> | <b>NFG</b> | <b>NCG</b> | <b>T</b> |
|----------|---------------|----------|------------|------------|------------|------------|----------|
| 1.       | CHDM          | 0        | 4 108      | 4 242      | 4 129      | 0          | 8.84     |
| 3.       | DLM-1         | 0        | 5 731      | 5 898      | 5 751      | 0          | 10.05    |
| 3.       | DLM-2         | 0        | 6 370      | 6 504      | 6 391      | 0          | 10.84    |
| 4.       | DLCHM         | 0        | 5 719      | 5 913      | 5 740      | 0          | 10.24    |
| 5.       | CGM           | 0        | 4 965      | 5 317      | 4 987      | 62 837     | 12.25    |
| 6.       | PCGM          | 1        | 7 639      | 7 851      | 7 659      | 8 445      | 17.03    |
| 6.       | PCGM          | 2        | 7 569      | 7 778      | 7 589      | 8 386      | 16.56    |
| 7.       | CGDLM(5)      | 0        | 3 957      | 4 104      | 3 978      | 23 463     | 8.06     |
| 8.       | LM(100)       | 0        | 5 076      | 5 426      | 5 097      | 71 035     | 12.97    |
| 9.(a)    | LCGM(5)       | 0        | 4 718      | 5 116      | 4 737      | 64 384     | 13.59    |
| 9.(a)    | LCGM(5)       | 1        | 6 065      | 6 183      | 6 087      | 9 188      | 12.66    |
| 9.(a)    | LCGM(5)       | 2        | 5 905      | 6 053      | 5 925      | 7 872      | 12.33    |
| 9.(b)    | CGLM(10)      | 0        | 4 986      | 5 312      | 5 007      | 75 463     | 12.45    |

**Tab. 1:** *Sum of squares minimization,  $n = 1000$ .*

| <b>N</b> | <b>Method</b> | <b>P</b> | <b>NIT</b> | <b>NFV</b> | <b>NFG</b> | <b>NCG</b> | <b>T</b> |
|----------|---------------|----------|------------|------------|------------|------------|----------|
| 1.       | CHDM          | 0        | 8 391      | 8 566      | 35 824     | 0          | 2:02.44  |
| 3.       | DLM-1         | 0        | 9 657      | 10 133     | 42 425     | 0          | 1:55.77  |
| 3.       | DLM-2         | 0        | 9 717      | 10 195     | 42 452     | 0          | 1:52.20  |
| 4.       | DLCHM         | 0        | 9 625      | 10 150     | 42 260     | 0          | 1:56.05  |
| 5.       | CGM           | 0        | 16 894     | 19 163     | 83 933     | 358 111    | 6:04.42  |
| 6.       | PCGM          | 1        | 10 600     | 11 271     | 50 385     | 3 767      | 2:25.42  |
| 6.       | PCGM          | 2        | 10 599     | 11 269     | 50 382     | 83         | 2:26.88  |
| 7.       | CGDLM(5)      | 0        | 8 938      | 9 276      | 39 032     | 47 236     | 2:02.84  |
| 8.       | LM(100)       | 0        | 14 679     | 16 383     | 71 483     | 366 695    | 6:41.45  |
| 9.(a)    | LCGM(5)       | 0        | 14 906     | 16 751     | 72 727     | 355 106    | 6:26.30  |
| 9.(a)    | LCGM(5)       | 1        | 8 347      | 8 454      | 35 939     | 4 329      | 1:48.87  |
| 9.(a)    | LCGM(5)       | 2        | 8 346      | 8 454      | 35 933     | 624        | 1:49.67  |
| 9.(b)    | CGLM(10)      | 0        | 15 655     | 17 723     | 76 696     | 394 060    | 6:30.89  |

**Tab. 2:** *Unconstrained minimization,  $n = 5000$ .*

If we consider preconditioning with matrix  $C$ , then  $P = 1$  or  $P = 2$ . The latter case means that before starting the iteration process we test whether solution  $w_*$  of system  $Cw = -g$  satisfies condition  $\|Bw_* + g\| \leq \varepsilon\|g\|$ . If this is so, we'll set  $d_* = w_*$  and the conjugate gradient method will be omitted.

Methods number 2 and 10, which are based on the knowledge of eigenvalues, are not tested. The former method uses large dense matrix  $Q$ , so it is not suitable for large scale structured problems, and the latter one consumes very much CPU time for eigenpairs computation.

If  $\text{NCG} = 0$ , then the method uses matrix decompositions instead of the conjugate gradient method.

See <http://www.cs.cas.cz/~luksan/test.html> for more details.

## References

- [1] Conn A.R., Gould N.I.M., Toint P.L.: *Trust-region methods*. SIAM, Philadelphia, 2000.
- [2] Fletcher R.: *Practical methods of optimization*. 2nd edition, John Wiley & Sons, Chichester, 1987.
- [3] Lukšan L.: *Metody s proměnnou metrikou*. Academia, Praha, 1990.
- [4] Lukšan L.: *Numerické optimalizační metody pro úlohy bez omezujících podmínek*. Výzkumná zpráva č. V-640, Ústav informatiky AV ČR, Praha, 1995.
- [5] Lukšan L., Tůma M., Šiška M., Vlček J., Ramešová N.: *UFO 2002, Interactive System for Universal Functional Optimization*. Technical report No. 883, Institute of Computer Science AS CR, Prague, 2002.
- [6] Lukšan L., Vlček J.: *Sparse and partially separable test problems for unconstrained and equality constrained optimization*. Technical Report No. 767, Institute of Computer Science AS CR, Prague, 1998.
- [7] Matonoha C.: *Numerická realizace metod s lokálně omezeným krokem*. Disertační práce, Matematicko-fyzikální fakulta UK, Praha, 2004.
- [8] Nocedal J., Wright S.J.: *Numerical optimization*. Springer Series in Operations Research, Springer, Berlin, 1999.