Trust region methods for unconstrained optimization

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Institute of Computer Science AS CR, Prague

ALGORITMY 2009
Conference on Scientific Computing
Vysoké Tatry, Podbanské
March 15-20, 2009

Outline

1. Unconstrained optimization
2. Trust region methods
3. Computation of direction vectors
4. Numerical comparison
5. Conclusion

1. Unconstrained optimization

Introduction

Consider the following unconstrained optimization problem

$$x_\star = \arg\min_{x \in \mathcal{R}^n} F(x)$$

where

- $F(x) : \mathcal{R}^n \to \mathcal{R}$
- F is twice continuously differentiable
- F is bounded from below

Notation:

- $g(x) = \nabla F(x) \ldots$ gradient
- $B(x) = \nabla^2 F(x) \ldots$ Hessian (or its approximation)

We focus on finding local minima.

Necessary and sufficient conditions for a minimum

- first-order necessary conditions:

  If $x_\star$ is a local minimum of $F \in \mathcal{C}^1$ then

  $$g(x_\star) = 0$$

- second-order necessary conditions:

  If $x_\star$ is a local minimum of $F \in \mathcal{C}^2$ then

  $$g(x_\star) = 0 \quad \text{and} \quad B(x_\star) \succeq 0$$

- second-order sufficient conditions:

  The point $x_\star$ is a strict local minimum of $F \in \mathcal{C}^2$ if

  $$g(x_\star) = 0 \quad \text{and} \quad B(x_\star) \succ 0$$

Basic optimization methods

Basic optimization methods (line-search and trust-region methods) generate points $x_k \in \mathcal{R},\ k \in \mathcal{N},$ in such a way that x_1 is arbitrary and

$$x_{k+1} = x_k + \alpha_k d_k$$

where

- $d_k \in \mathcal{R}^n$ are direction vectors
  – determination is based on quantities

$$x_j,\ F(x_j),\ g(x_j),\ B(x_j), \quad 1 < j < k$$

- $\alpha_k > 0$ are step-sizes
  – determination is based on a behaviour of F in a neghbourhood of x_k

Basic optimization method is globally convergent if

$$\liminf_{k \to \infty} \|g(x_k)\| = 0$$

for an arbitrary initial vector x_1.

General optimization algorithm

A typical pattern:

1. Choose an arbitrary x_1 and compute $F(x_1)$. Set $k = 1$.
2. Compute a direction vector d_k and a step-length α_k.
3. Set $x_{k+1} = x_k + \alpha_k d_k$ and compute $F(x_{k+1})$.
4. Check termination criterion. If not achieved, set $k := k + 1$ and goto 2.

Stopping tolerances:

- the gradient is small:
$$|g(x_{k+1})| < \varepsilon_g$$
- the reduction in F is insignificant:
$$|F(x_{k+1}) - F(x_k)| < \varepsilon_F$$
- the changes in all variables are insignificant:
$$|x_{k+1} - x_k| < \varepsilon_x$$

Optimization methods I.

1. Steepest descent method:

$$d_k = -g(x_k), \quad \alpha_k = \arg\min_{\alpha \geq 0} F(x_k + \alpha d_k)$$

   + globally convergent
   + uses only vectors from $\mathcal{R}^n$
   – exact choice of a step-length
   – only linearly convergent

2. Newton's method:

$$d_k = -B(x_k)^{-1} g(x_k), \quad \alpha_k = 1$$

   + simple choice of a step-length
   + convergence is quadratic
   – not globally convergent
   – solving a system of linear equations
   – computation of second-order derivatives

Optimization methods II.

1. Line-search methods:
   - developed from the steepest descent method
   - use inexact choice of a step-length
   - faster convergence
   - conjugate gradient method, variable metric method

2. Trust-region methods:
   - developed from the Newton's method
   - ensuring global convergence
   - decreased number of operations
   - Hessian matrix can be indefinite, ill-conditioned, singular
   - excellent convergence properties
   - realization of Newton's method and Gauss-Newton's method for sum a squares minimization

2. Trust region methods

The idea

1. choose an initial approximation x_1 and compute $F(x_1)$
2. construct a quadratic model $Q_k(d)$ of F at x_k
3. choose a trust region radius $\Delta_k > 0$
4. find a minimum d_k of $Q_k(d)$ for which $\|d_k\| \leq \Delta$
5. if the decrease predicted by the model corresponds to the actual decrease in $x_k + d_k$, then set

$$x_{k+1} = x_k + d_k$$

   and possibly increase Δ_{k+1}
6. in the opposite case, the step d_k is too long, set $x_{k+1} = x_k$ and decrease the radius Δ_{k+1}

$$\{d \in \mathcal{R}^n : \|d\| \leq \Delta\}$$

is a region where we trust the model $Q(d)$ is a good representation of F

Notation

We define:

- the quadratic function

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

  which locally approximates the difference $F(x_k + d) - F(x_k)$;

- the vector

$$\omega_k(d) = (B_k d + g_k)/\|g_k\|$$

  for the accuracy of a computed direction;

- the number

$$\rho_k(d) = \frac{F(x_k + d) - F(x_k)}{Q_k(d)}$$

  for the ratio of actual and predicted decrease of the objective function.

Trust-region methods are based on approximate minimizations of $Q_k(d)$ on the balls $\|d\| \leq \Delta_k$ followed by updates of radii $\Delta_k > 0$.

Definition

Direction vectors $d_k \in \mathcal{R}^n$ are chosen to satisfy the conditions

$$\|d_k\| \leq \Delta_k, \tag{1}$$

$$\|d_k\| < \Delta_k \quad \Rightarrow \quad \|\omega_k(d_k)\| \leq \overline{\omega}, \tag{2}$$

$$-Q_k(d_k) \geq \underline{\sigma}\|g_k\| \min(\|d_k\|, \|g_k\|/\|B_k\|), \tag{3}$$

where $0 \leq \overline{\omega} < 1$ and $0 < \underline{\sigma} < 1$.

Step sizes $\alpha_k \geq 0$ are selected so that

$$\rho_k(d_k) \leq 0 \quad \Rightarrow \quad \alpha_k = 0, \tag{4}$$

$$\rho_k(d_k) > 0 \quad \Rightarrow \quad \alpha_k = 1. \tag{5}$$

Trust-region radii $0 < \Delta_k \leq \overline{\Delta}$ are chosen in such a way that

$$\rho_k(d_k) < \underline{\rho} \quad \Rightarrow \quad \underline{\beta}\|d_k\| \leq \Delta_{k+1} \leq \overline{\beta}\|d_k\|, \tag{6}$$

$$\rho_k(d_k) \geq \underline{\rho} \quad \Rightarrow \quad \Delta_i \leq \Delta_{k+1} \leq \overline{\Delta}, \tag{7}$$

($0 < \Delta_1 \leq \overline{\Delta}$ is arbitrary), where $0 < \underline{\beta} \leq \overline{\beta} < 1$ and $0 < \underline{\rho} < 1$.

Maximum step length $\overline{\Delta}$

The use of the maximum step length $\overline{\Delta}$ has no theoretical significance but is very useful for practical computations:

- The problem functions can sometimes be evaluated only in a relatively small region (if they contain exponentials) so that the maximum step-length is necessary.
- The problem can be very ill-conditioned far from the solution point, thus large steps are unsuitable.
- If the problem has more local solutions, a suitably chosen maximum step-length can cause a local solution with a lower value of F to be reached.

Therefore, the maximum step-length $\overline{\Delta}$ is a parameter which is most frequently tuned.

Convergence

Assumptions:

1. F is bounded from below
2. second-order derivatives of F are bounded
3. direction vector d_k satisfies (1)-(7)
4. matrices B_k are uniformly bounded or (weaker assumption)

$$\sum_{k \in \mathcal{N}} M_k^{-1} = \infty$$

where $M_k = \max\{\|B_1\|, \ldots, \|B_k\|\}$

Theorem:

1. global convergence: for an arbitrary chosen initial x_1:

$$\liminf_{k \to \infty} \|g(x_k)\| = 0$$

2. Q-superlinear convergence (additional assumptions)

Two classes of methods

Trust-region subproblem: (to simplify the notation, we omit the major index k)

(8) $$\boxed{\min Q(d) = \tfrac{1}{2} d^T B d + g^T d, \quad \|d\| \leq \Delta}$$

Optimal solution – $d \in \mathcal{R}^n$ solves (8)

- Necessary and sufficient conditions for this solution:

$$\|d\| \leq \Delta, \quad (B + \lambda I)d = -g, \quad B + \lambda I \succeq 0, \quad \lambda \geq 0, \quad \lambda(\Delta - \|d\|) = 0,$$

  where λ is a Lagrange multiplier
- Choleski decomposition methods

Approximate solution – if B is not sufficiently small or sparse or explicitly available, then it is either too expensive or not possible to compute its Choleski factorization.

- d is found on subspaces of $\mathcal{R}^n$, e.g. Krylov subspaces
- methods based on matrix-vector multiplications are convenient

3. Computation of direction vectors

Moré-Sorensen 1983 [MS]

- the most sophisticated method which computes the optimal locally constrained step.
- based on solving the nonlinear equation

$$\phi(\lambda) = \frac{1}{\|d(\lambda)\|} - \frac{1}{\Delta} = 0 \quad \text{with} \quad (B + \lambda I)d(\lambda) + g = 0$$

  by the Newton's method using the Choleski decomposition of $B + \lambda I$.
- function $\phi(\lambda)$ is convex and decreasing in $(-\lambda_1, \infty)$, thus a unique solution exists (λ_1 is the smallest eigenvalue of B)
- this method is very robust but requires 2-3 Choleski decompositions for one direction determination on the average

Powell 1970, Dennis-Mei 1975 [DL]

Simpler methods are based on minimization of $Q(d)$ on the two-dimensional subspace containing the Cauchy and Newton steps

$$d_C = -\frac{g^T g}{g^T B g}\, g, \quad d_N = -B^{-1} g \qquad (\|d_C\| \leq \|d_N\|).$$

The most popular is the dogleg method where

$$d = d_N \quad \text{if} \quad \|d_N\| \leq \Delta$$

and

$$d = (\Delta/\|d_C\|) d_C \quad \text{if} \quad \|d_C\| \geq \Delta.$$

In the remaining case, d is a combination of d_C and d_N such that $\|d\| = \Delta$.

This method requires only one Choleski decomposition for one direction determination.

Steihaug 1983, Toint 1981 [ST]

This method

- is based on the conjugate gradient method applied to the linear system

$$Bd + g = 0,$$

- computes only an approximate solution,
- uses the fact that

$$Q(d_{j+1}) < Q(d_j) \quad \text{and} \quad \|d_{j+1}\| > \|d_j\|$$

  hold in the subsequent CG iterations if the CG coefficients are positive and no preconditioning is used.

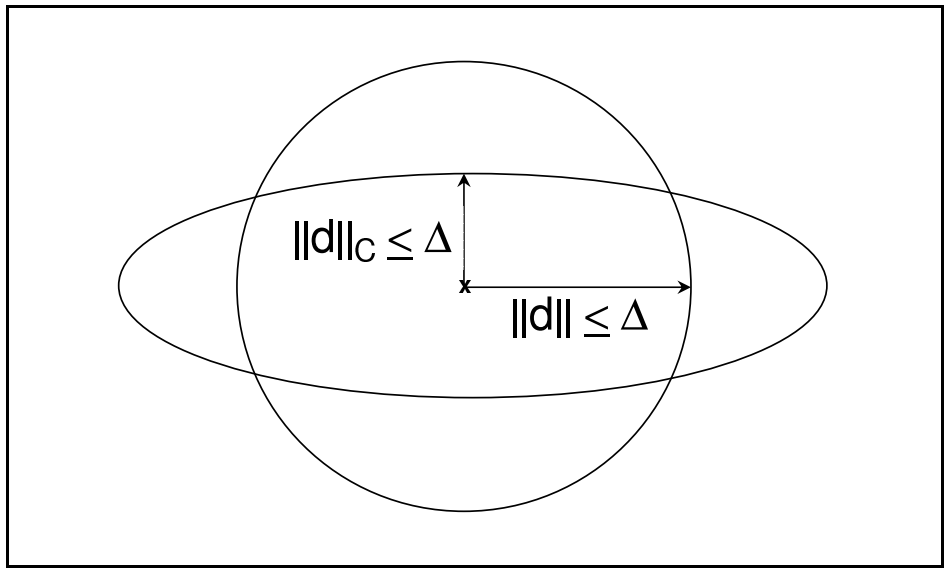
Termination:

- an unconstrained solution with a sufficient precision ($\|r\| \leq \varepsilon$)
- stop on the trust-region boundary
  - a *negative curvature* is encountered
  - the *constraint* is violated

Preconditioned Steihaug-Toint [PST]

For SPD preconditioner C we have

$$\|d_{j+1}\|_C > \|d_j\|_C \quad \text{with} \quad \|d_j\|_C^2 = d_j^T C d_j.$$



- trust-region can be leaved prematurely
- direction vector can be farther from the optimal step obtained without preconditioning
+ it is usually compensated by the rapid convergence of the PCG method

Multiple dogleg [MDL]

The CG steps can be combined with the Newton step $d_N = -B^{-1}g$ in the multiple dogleg method.

1. Let $j \ll n$ and d_j be a vector obtained after j CG steps of the Steihaug-Toint method (usually $j = 5$).
2. If $\|d_j\| < \Delta,$ we use d_k instead of $d_C = d_1$ in the dogleg method.

Gould-Lucidi-Roma-Toint 1997 [GLRT]

This method solves TRS iteratively by using the Lanczos process.
A solution – vector d_j

- is the $j-$th approximation of the optimal step d
- is contained in the Krylov subspace

$$\mathcal{K}_j = \text{span}\{g, Bg, \ldots, B^{j-1}g\}$$

of dimension j defined by the matrix B and the vector g.

In this case, $d_j = Z\tilde{d}_j$, where $\tilde{d}_j$ is obtained by solving the $j-$dimensional subproblem

$$\min 1/2\ \tilde{d}^T T\tilde{d} + \|g\|e_1^T\tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta.$$

Here

- $T = Z^T BZ$ (with $Z^T Z = I$) is the Lanczos tridiagonal matrix
- e_1 is the first column of the unit matrix

Matrix Z has to be stored – we consider only $j \leq n$ cols (usually $j \leq 100$).

Shifted Steihaug-Toint [SST, PSST]

This method applies the ST method to the shifted subproblem

(9) $$\min \tilde{Q}(d) = Q_{\tilde{\lambda}}(d) = 1/2\, d^T(B + \tilde{\lambda} I)d + g^T d \quad \text{s.t.} \quad \|d\| \leq \Delta$$

- the number $\tilde{\lambda} \geq 0$ approximates the optimal λ in MS method
- this method combines good properties of the MS and ST methods
- it can be successfully preconditioned
- the solution is closer to the optimal solution than the ST point

1. Carry out $j \ll n$ steps of the unpreconditioned Lanczos method to obtain the tridiagonal matrix $T = T_j = Z_j^T B Z_j$ (usually $j = 5$).
2. Solve the subproblem

$$\min 1/2\, \tilde{d}^T T \tilde{d} + \|g\| e_1^T \tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta,$$

   using the MS method to obtain the Lagrange multiplier $\tilde{\lambda}$.
3. Apply the PST method to (9) to obtain the direction vector $d = d(\tilde{\lambda})$.

Hager 2001 [PH]

There are several techniques for large scale TR subproblems that are not based on conjugate gradients. This method solves

(10) $$\min\ Q(d) = \frac{1}{2}\, d^T B d + g^T d \quad \text{subject to} \quad \|d\| \leq \Delta$$

with the additional constraint that d is contained in a low-dimensional subspace. They are modified in successive iterations to obtain quadratic convergence to the optimum. We seek vectors $d \in \mathcal{S}$ where $\mathcal{S}$ contains:

- The previous iterate. This causes that the value of the objective function can only decrease in consecutive iterations.
- The vector $Bd + g$. It ensures descent if the current iterate does not satisfy the first-order optimality conditions.
- An estimate for an eigenvector of B ass. with the smallest eigenvalue. It will dislodge the iterates from a nonoptimal stationary point.
- The SQP iterate. The convergence is locally quadratic if $\mathcal{S}$ contains the iterate generated by one step of the SQP algorithm applied to (10).

SQP method

The SQP method is equivalent to the Newton's method applied to the nonlinear system

$$(B + \lambda I)d + g = 0, \quad \frac{1}{2}\, d^T d - \frac{1}{2}\, \Delta^2 = 0.$$

The Newton iterate can be expressed in the following way:

$$d_{SQP} = d + z, \quad \lambda_{SQP} = \lambda + \nu,$$

where z and ν are solutions of the linear system

$$\begin{aligned} (B + \lambda I)z + d\,\nu &= -\big((B + \lambda I)d + g\big), \\ d^T z &= 0, \end{aligned}$$

which can be solved by preconditioned CG method with the incomplete Choleski-type decomposition of matrix $B + \lambda I$.

Rojas-Santos-Sorensen 1997, 2000 [RSS]

Consider the bordered matrix

$$B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & B \end{pmatrix}$$

where $\alpha \in \mathcal{R}$ and observe that

$$\frac{\alpha}{2} + Q(d) = \frac{1}{2}(1, d^T) B_\alpha \begin{pmatrix} 1 \\ d \end{pmatrix}.$$

Thus there exists a value of α such that $\min Q(d)$ is equivalent to

$$\min \frac{1}{2} d_\alpha^T B_\alpha d_\alpha \quad \text{subject to} \quad \|d_\alpha\|^2 \leq 1 + \Delta^2, \quad e_1^T d_\alpha = 1,$$

where $d_\alpha = (1, d^T)^T$ and $e_1 \in \mathcal{R}^{n+1}$ is the first canonical unit vector.

- the desired solution is found in terms of eigenpairs of B_α
- the resulting algorithm is superlinearly convergent

4. Numerical comparison

Test problems

- The methods are implemented in the interactive system UFO
  www.cs.cas.cz/luksan/ufo.html
  as subroutines for solving trust region subproblems.
- They were tested by using two collections of 22 sparse test problems
  www.cs.cas.cz/luksan/test.html
  with 1000 and 5000 variables – subroutines TEST 14 and TEST 15.

The results are given in the following tables or graphs, where

- NIT is the total number of iterations,
- NFV is the total number of function evaluations,
- NFG is the total number of gradient evaluations,
- NDC is the total number of Choleski-type decompositions
  (complete for MS,DL,MDL and incomplete for PH,PST,PSST),
- NMV is the total number of matrix-vector multiplications,
- Time is the total computational time in seconds.

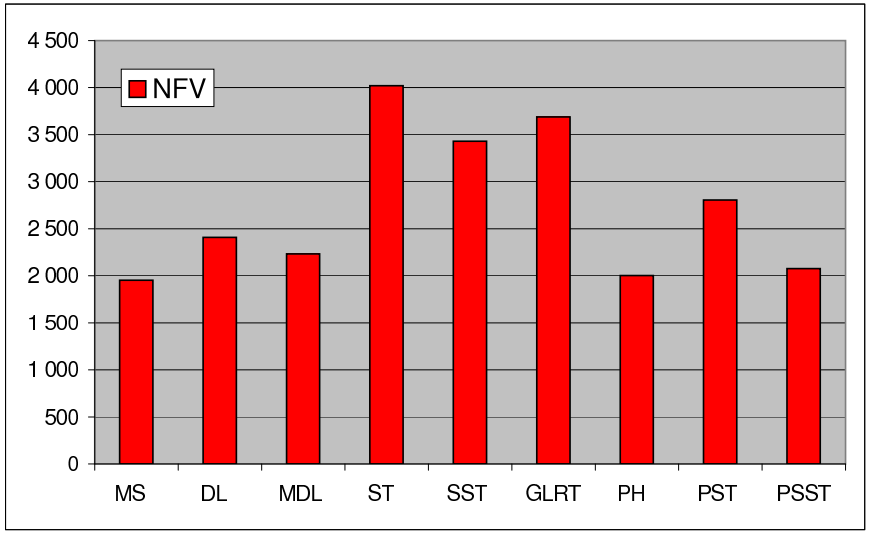
Table 1 – TEST 14 – unconstrained minimization

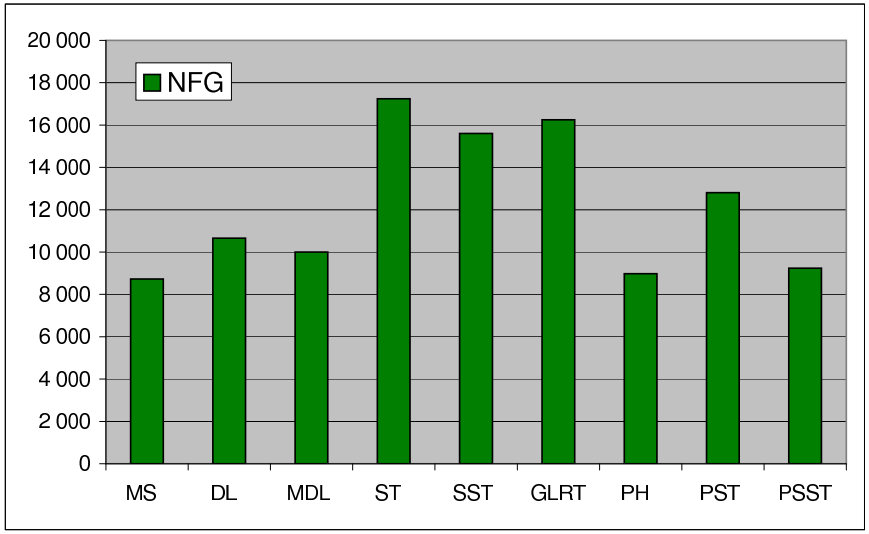
N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1911	1952	8724	3331	1952	3.13
	DL	2272	2409	10653	2195	2347	2.94
	MDL	2132	2232	9998	1721	21670	3.17
	ST	3475	4021	17242	0	63016	5.44
	SST	3149	3430	15607	0	75044	5.97
	GLRT	3283	3688	16250	0	64166	5.40
	PH	1958	2002	8975	3930	57887	5.86
	PST	2608	2806	12802	2609	5608	3.30
	PSST	2007	2077	9239	2055	14440	2.97
	RSS	2183	2337	10075	0	∞	∞
5000	MS	8177	8273	34781	13861	8272	49.02
	DL	9666	10146	42283	9398	9936	43.37
	MDL	8913	9244	38846	7587	91784	48.05
	ST	16933	19138	84434	0	376576	134.52
	SST	14470	15875	70444	0	444142	146.34
	GLRT	14917	16664	72972	0	377588	132.00
	PH	8657	8869	37372	19652	277547	127.25
	PST	11056	11786	53057	11057	23574	65.82
	PSST	8320	8454	35629	8432	59100	45.57

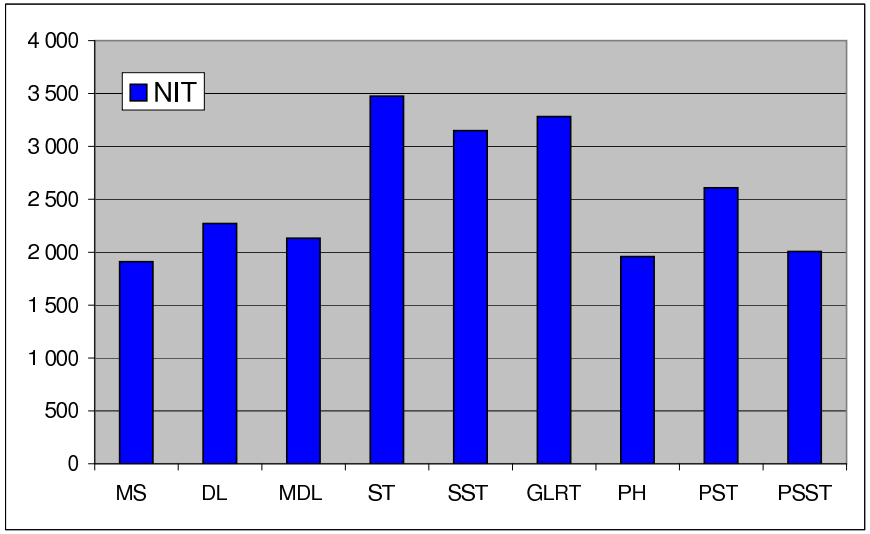
Table 2 – TEST 15 – sums of squares

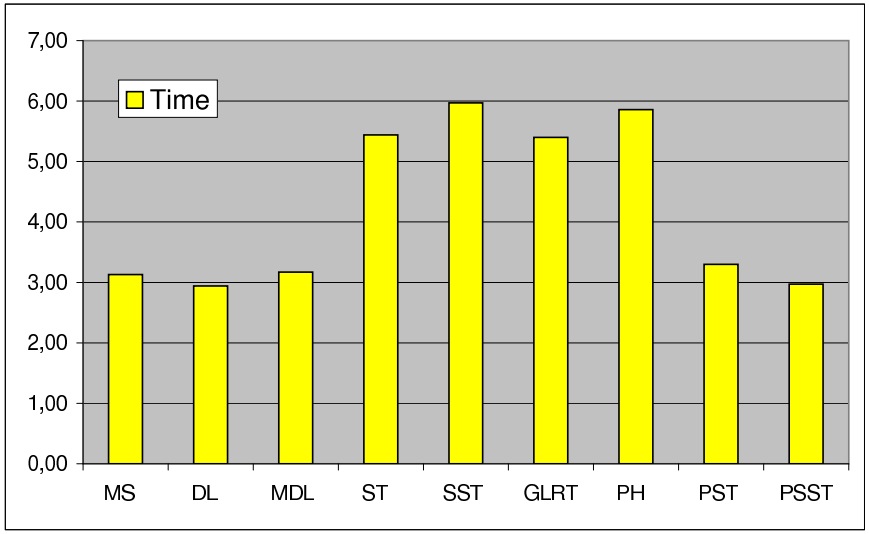
N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1946	9094	9038	3669	2023	5.86
	DL	2420	12291	12106	2274	2573	9.00
	MDL	2204	10586	10420	1844	23139	7.86
	ST	2738	13374	13030	0	53717	11.11
	SST	2676	13024	12755	0	69501	11.39
	GLRT	2645	12831	12547	0	61232	11.30
	PH	1987	9491	9444	6861	84563	11.11
	PST	3277	16484	16118	3278	31234	11.69
	PSST	2269	10791	10613	2446	37528	8.41
5000	MS	7915	33607	33495	14099	8047	89.69
	DL	9607	42498	41958	9299	9963	128.92
	MDL	8660	37668	37308	7689	91054	111.89
	ST	11827	54699	53400	0	307328	232.70
	SST	11228	51497	50333	0	366599	231.94
	GLRT	10897	49463	48508	0	300580	214.74
	PH	8455	36434	36236	20538	281736	182.45
	PST	9360	41524	41130	9361	179166	144.40
	PSST	8634	37163	36881	8915	219801	140.44

TEST 14 – unconstrained minimization – N=1000

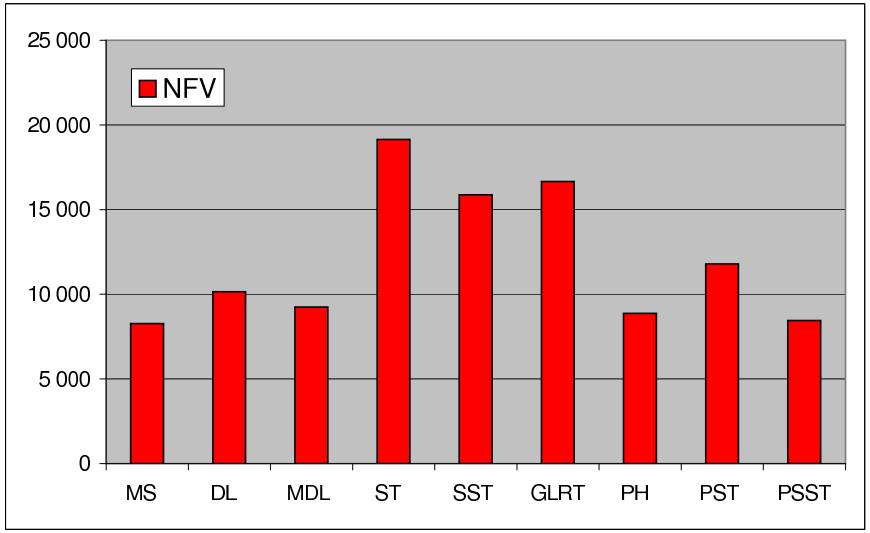


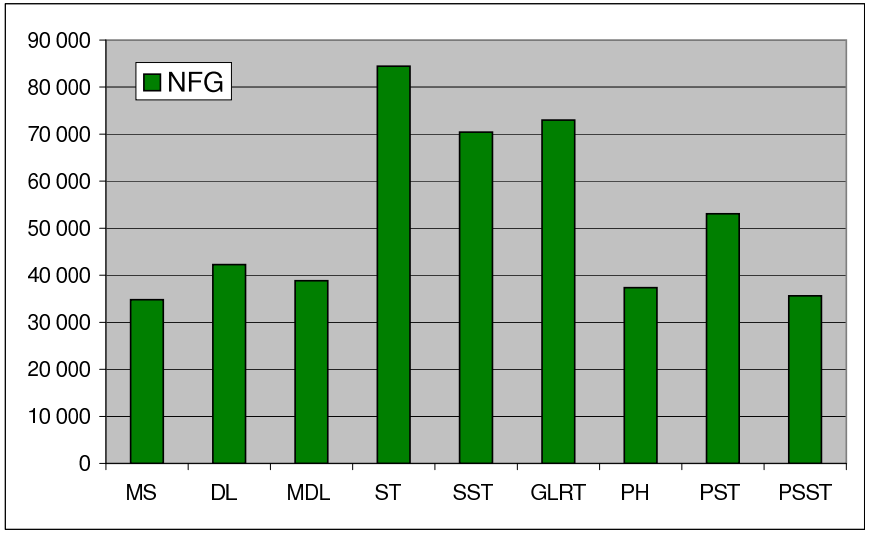


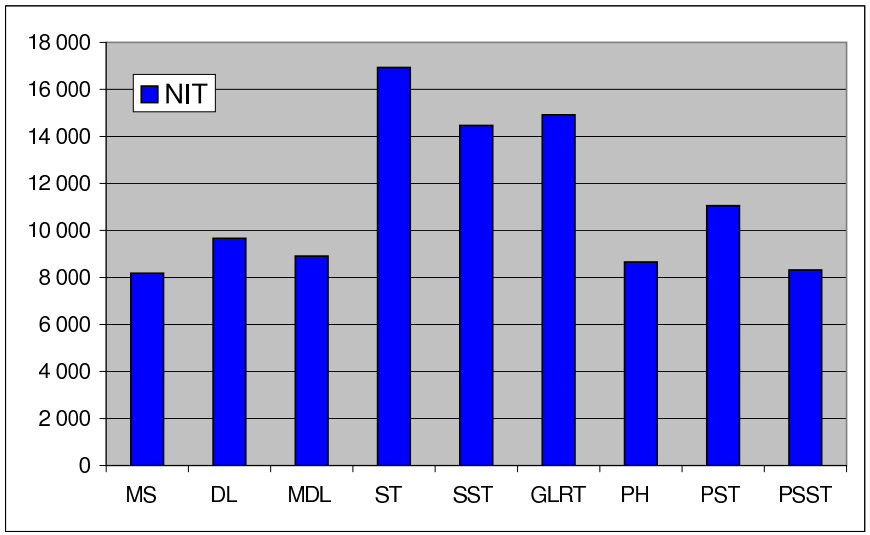


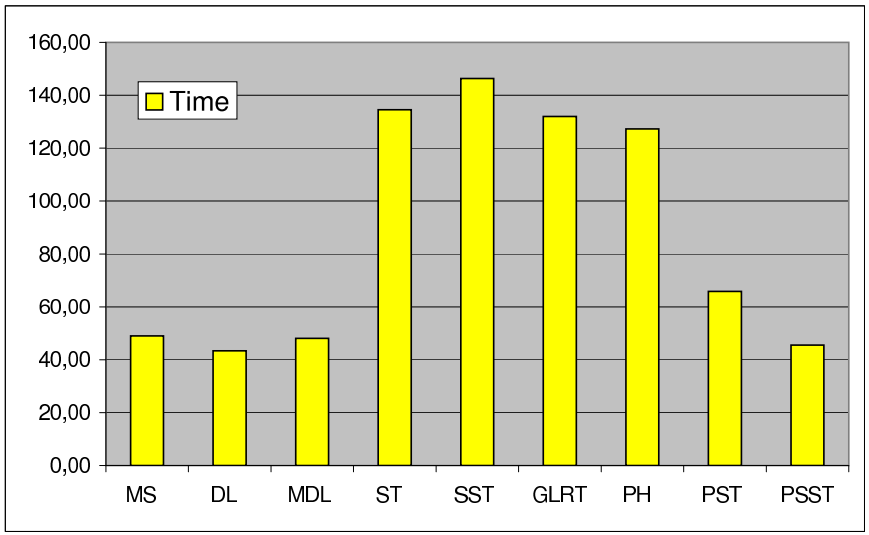


TEST 14 – unconstrained minimization – N=5000

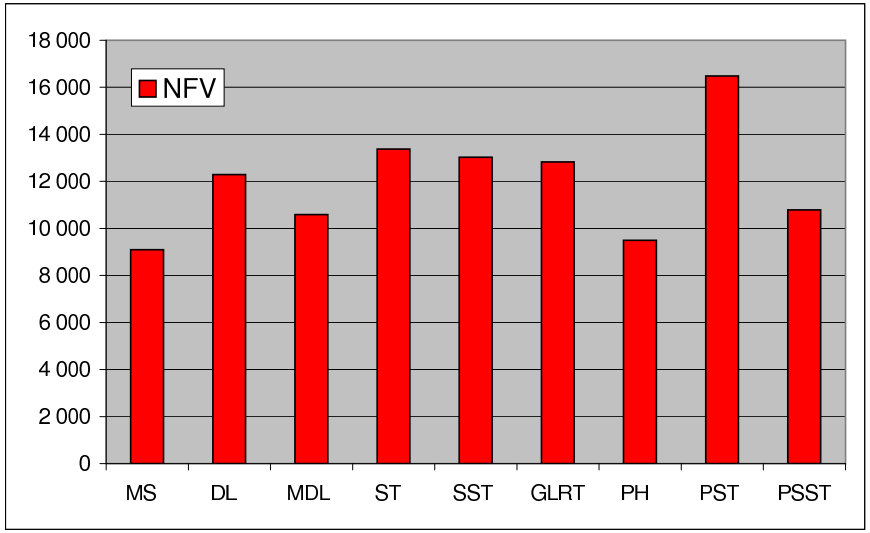


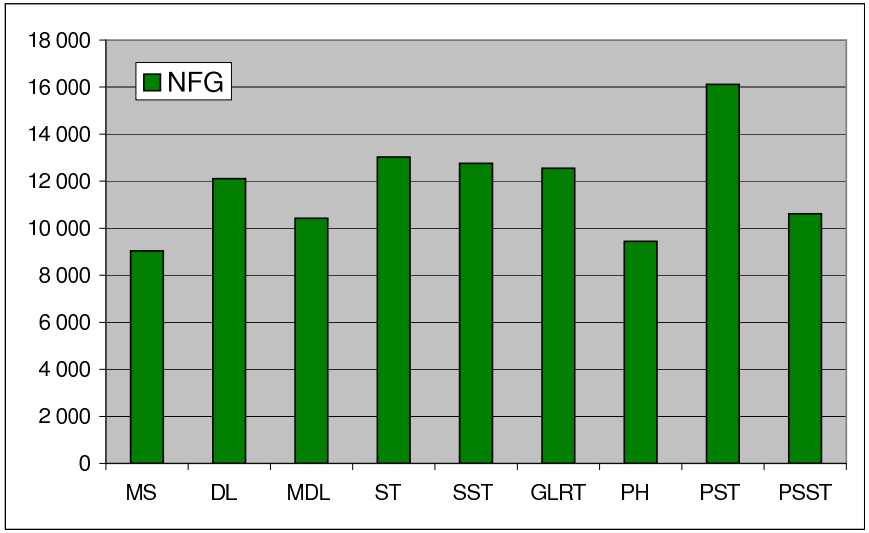


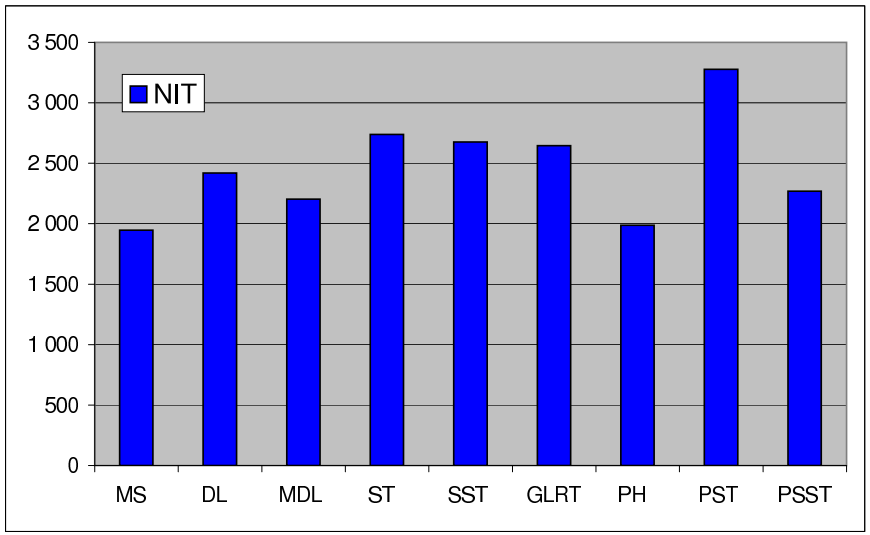


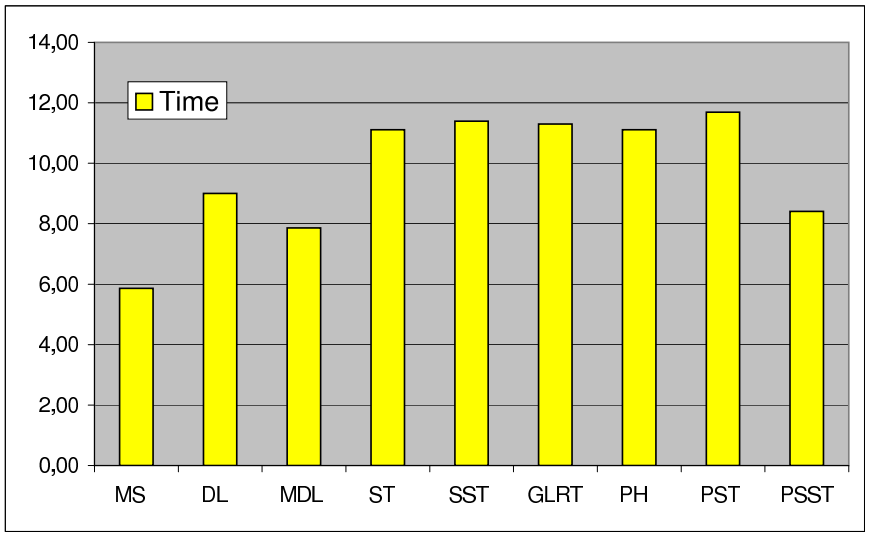


TEST 15 – sums of squares – N=1000

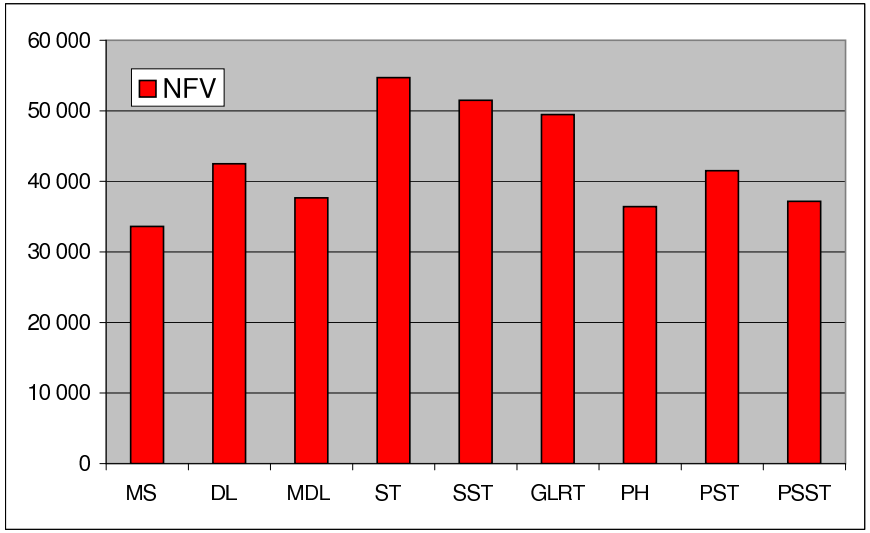


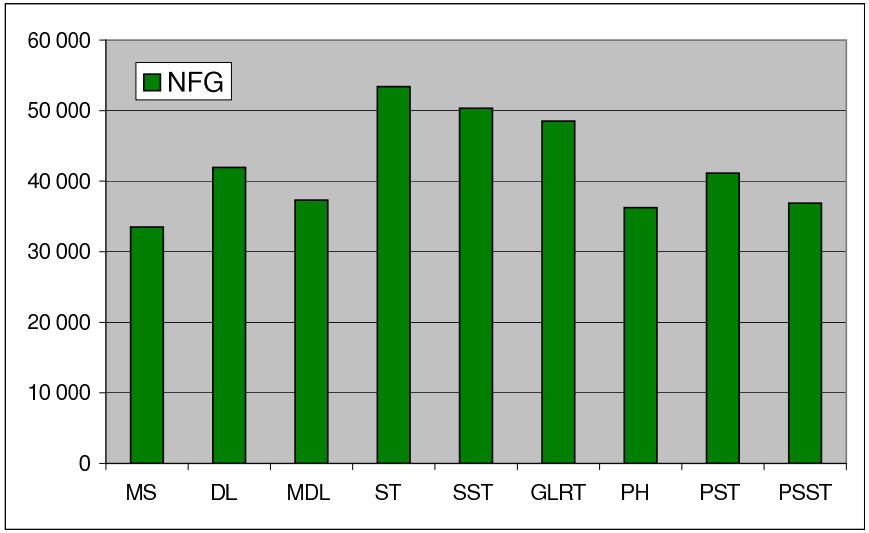


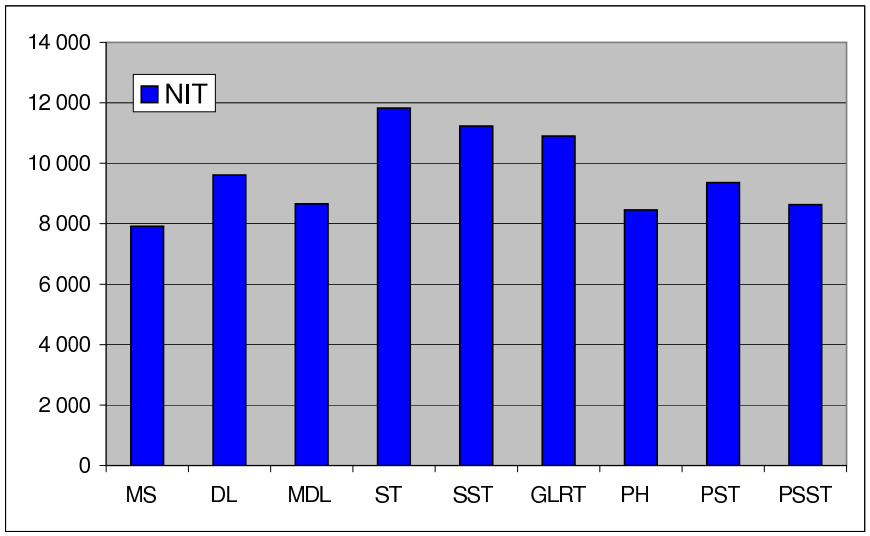


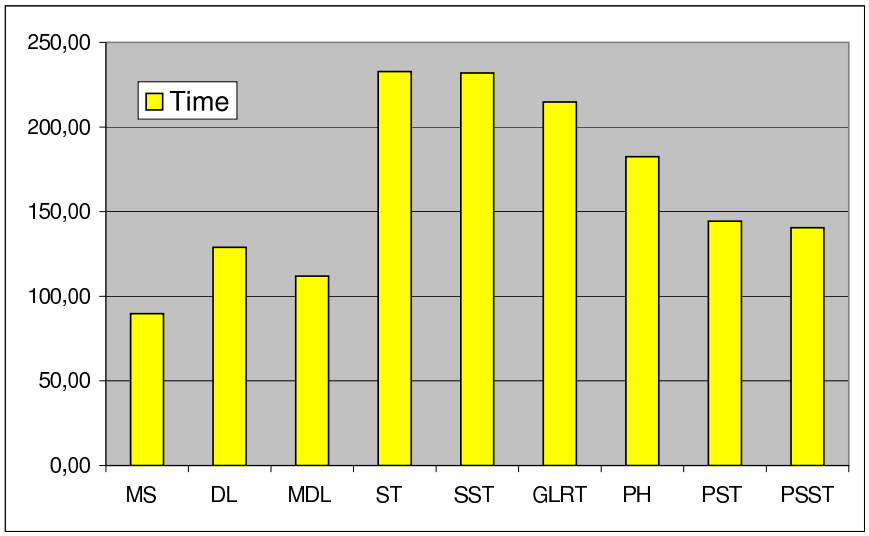


TEST 15 – sums of squares – N=5000









Comments

All problems are sparse $\Rightarrow$ the CD methods (MS,DL,MDL) are very efficient, much better than unpreconditioned MV methods (ST,SST,GLRT). Note that the methods PH,RSS are based on a different principle.

1. Since TEST 14 contains reasonably conditioned problems, the preconditioned MV methods are competitive with the CD methods. Note that NFG is much greater than NFV since the Hessian matrices are computed by using gradient differences.
2. On the contrary, TEST 15 contains several very ill-conditioned problems and thus the CD methods work better than the MV methods. Note that the problems are the sums of squares having the form

$$F(x) = \frac{1}{2} f^T(x) f(x)$$

and NFV denotes the total number of the vector $f(x)$ evaluations. Since $f(x)$ is used in the expression

$$g(x) = J^T(x) f(x),$$

where $J(x)$ is the Jacobian matrix of $f(x)$, NFG is comparable with NFV.

5. Conclusion

Summary

To sum up, our computational experiments indicate the following:

- the CD methods (MS,DL,MDL) are very efficient for ill-conditioned but reasonably sparse problems;
- if the problems do not have sufficiently sparse Hessian matrices, then the CD methods can be much worse than the MV methods (ST,SST,GLRT);
- an efficiency of the MV methods strongly depends on suitable preconditioning (we use an incomplete Choleski decomposition).
- methods based on computation of eigenvectors (PH and especially RSS) are not efficient since computation of eigenvalues is time-consuming (method RSS was developed for regularization of large-scale ill-posed least-squares problems)

Thank you for your attention!