Applied Mathematics and Computation 235 (2014) 394–407

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Efficient tridiagonal preconditioner for the matrix-free truncated Newton method [☆]

Ladislav Lukšan [a,b,*], Jan Vlček [a]

[a] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
[b] Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

ARTICLE INFO

Keywords:
Unconstrained optimization
Large scale optimization
Matrix-free truncated Newton method
Preconditioned conjugate gradient method
Preconditioners obtained by the directional differentiation
Numerical algorithms

ABSTRACT

In this paper, we study an efficient tridiagonal preconditioner, based on the directional differentiation, applied to the matrix-free truncated Newton method for unconstrained optimization. It is proved that this preconditioner is positive definite for many practical problems. The efficiency of the resulting matrix-free truncated Newton method is demonstrated by results of extensive numerical experiments.

1. Introduction

We consider the unconstrained minimization problem

$$x^* = \arg\min_{x \in R^n} F(x),$$

where function $F : \mathcal{D}(F) \subset R^n \to R$ is twice continuously differentiable and n is large. We use the notation

$$g(x) = \nabla F(x), \quad G(x) = \nabla^2 F(x)$$

and the assumption that $\|G(x)\| \leqslant \overline{G}$, $\forall x \in \mathcal{D}(F)$. Numerical methods for large unconstrained minimization are usually iterative and their iteration step has the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \in N,$$

where d_k is a direction vector and α_k is a step-length. In this paper, we will deal with the Newton method, which uses the quadratic model

$$F(x_k + d) \approx Q(x_k + d) = F(x_k) + g^T(x_k)d + \frac{1}{2} d^T G(x_k) d, \tag{1}$$

for direction determination in such a way that

$$d_k = \arg\min_{d \in \mathcal{M}_k} Q(x_k + d).$$

[☆] This work was supported by the Grant Agency of the Czech Republic, project No. 13-06684S, and the Institute of Computer Science of the AS CR (RVO:67985807).

[*] Corresponding author at: Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic.

E-mail address: luksan@cs.cas.cz (L. Lukšan).

http://dx.doi.org/10.1016/j.amc.2014.03.006
0096-3003/© 2014 Elsevier Inc. All rights reserved.

There are two basic possibilities for direction determination: the line-search method, where $\mathcal{M}_k = R^n$, and the trust-region method, where $\mathcal{M}_k = \{d \in R^n : \|d\| \leqslant \Delta_k\}$ (here $\Delta_k > 0$ is the trust region radius). Properties of line search and trust region methods together with comments concerning their implementation are exhaustively introduced in [3,23], so no more details are given here.

In this paper, we assume that neither matrix $G_k = G(x_k)$ nor its sparsity pattern are explicitly known. In the latter case, direct methods based on Gaussian elimination cannot be used, so it is necessary to compute the direction vector iteratively. There are many various iterative methods making use of the symmetry of the Hessian matrix, see [27]. Some of them, e.g. [7,8,25] allow us to consider indefinite Hessian matrices. Even if these methods are of theoretical interest and lead to non-traditional preconditioners, see [10,11], we confine our attention to modifications of the conjugate gradient method [28–30], which are simple and very efficient (also in the indefinite case). We studied and tested both the line search and the trust region approaches, but the second approach did not give significantly better results than the first one. Therefore, we restrict our attention to the line search implementation of the truncated Newton method.

Since matrix $G(x)$ is not given explicitly, we use numerical differentiation instead of matrix multiplication. Thus the product $G(x)p$ is replaced by the difference

$$G(x)p \approx \frac{g(x+\delta p) - g(x)}{\delta}, \tag{2}$$

where $\delta = \varepsilon/\|p\|$ and $\varepsilon > 0$ is a suitable small number (usually $\varepsilon \approx \sqrt{\varepsilon_M}$ where ε_M is the machine precision). The resulting method is called the truncated Newton method. This method has been theoretically studied in many papers, see [4,5,9,21,24]. The following theorem, which easily follows from the mean value theorem, confirms the choice (2).

Theorem 1. *Let function $F : R^n \to R$ be Lipschitz continuous second order derivatives (with the Lipschitz constant $\bar{L}$). Let $q = G(x)p$ and*

$$\tilde{q} = \frac{g(x+\delta p) - g(x)}{\delta}, \quad \delta = \frac{\varepsilon}{\|p\|},$$

where $\varepsilon > 0$ is an arbitrary (usually small) number. Then it holds

$$\|\tilde{q} - q\| \leqslant \frac{1}{2}\varepsilon \bar{L}\|p\|.$$

To make the subsequent investigations clear, we briefly describe the preconditioned conjugate gradient subalgorithm proposed in [28] where matrix multiplications are replaced by gradient differences (the outer index k is for the sake of simplicity omitted). In this subalgorithm, $0 < \omega < 1$ is a chosen precision (which usually depends on $\|g\|$) and m is the maximum number of the conjugate gradient steps (we have used value $m = n + 3$ in our numerical experiments).

Truncated Newton PCG subalgorithm:

```
$d_1 = 0, \quad g_1 = g, \quad h_1 = C^{-1}g_1, \quad \rho_1 = g_1^T h_1, \quad p_1 = -h_1.$
Do $i = 1$ to $m$
$\delta_i = \varepsilon/\|p_i\|, \quad \tilde{q}_i = (g(x+\delta_i p_i) - g(x))/\delta_i, \quad \sigma_i = p_i^T \tilde{q}_i.$
If $\sigma_i < \varepsilon\|p_i\|^2$ then $d = d_i$, stop.
$\alpha_i = \rho_i/\sigma_i, \quad d_{i+1} = d_i + \alpha_i p_i, \quad g_{i+1} = g_i + \alpha_i \tilde{q}_i,$
$h_{i+1} = C^{-1}g_{i+1}, \quad \rho_{i+1} = g_{i+1}^T h_{i+1}.$
If $\|g_{i+1}\| \leqslant \omega\|g_1\|$ or $i = m$ then $d = d_i$, stop.
$\beta_i = \rho_{i+1}/\rho_i, \quad p_{i+1} = -h_{i+1} + \beta_i p_i.$
End do
```

A disadvantage of the truncated Newton PCG subalgorithm with $C = I$ (unpreconditioned) consists in the fact that it requires a large number of inner iterations (i.e. a large number of gradient evaluations) if matrix $G = G(x)$ is ill-conditioned. Thus a suitable preconditioner should be used. Unfortunately, the sparsity pattern of G is not known, so the standard preconditioning methods requiring the knowledge of the sparsity pattern (e.g. methods based on the incomplete Choleski decomposition) cannot be chosen.

There are various ways for building positive definite preconditioners, which can be utilized in the truncated Newton PCG subalgorithm:

- Preconditioners based on the limited memory BFGS updates. This very straightforward approach is studied in [13,20].
- Preconditioners determined by the Lanczos method equivalent to the conjugate gradient method. This approach is studied in [10,11,21].
- Band preconditioners obtained by the standard BFGS method equivalent to the preconditioned conjugate gradient method. This approach is described in [22], where it is used for building of diagonal preconditioners. More general band preconditioners of this type are studied in [17].