On increasing the efficiency of gradient optimization routines

Michał J. Pazdanowski

Institute for Computational Civil Engineering, Cracow University of Technology
ul. Warszawska 24, 31-155 Cracow, Poland
e-mail: plpazdan@cyfronet.krakow.pl

Abstract

Solution time of nonlinear constrained optimization problem is highly dependent on number of decision variables, constraints and conditioning of the decision variables space. While the number of decision variables and constraints are external to the solution procedure itself, one may try to affect the conditioning of the decision variables space within the optimization module. This will affect the ratio of convergence of an iterative optimization routine.

An idea how to improve the conditioning of decision variables space using SVD (Singular Value Decomposition) is presented in this paper. The information flow logic in an optimization routine using this idea is presented as well. Results of computer tests performed during minimization of quadratic objective function and quadratic constraints are enclosed and discussed.

Keywords: nonlinear constrained optimization, gradient method, optimization speedup, SVD

1. Introduction

Nonlinear constrained optimization problems are solved numerically using iterative methods, for instance gradient based search methods. Effectiveness of gradient based search algorithm is strongly dependent on number of decision variables, constraints and conditioning of decision variables space.

While number of decision variables and constraints is external to the optimization routine, being defined by the problem at hand, the conditioning of decision variables space may be improved within the optimization procedure, in a manner transparent to the solved problem.

This paper deals with a concept of improving the convergence speed of gradient optimization routine applied to solve the optimization problem defined by quadratic objective function and quadratic nonlinear constraints in a form:

$$\min_{\mathbf{x}} F(\mathbf{x}) = \tfrac{1}{2}\cdot\mathbf{x}^T\cdot\mathbf{A}\cdot\mathbf{x},\quad \mathbf{A}^T=\mathbf{A} \quad , \tag{1}$$

at:

$$G_i(\mathbf{x}) = \tfrac{1}{2}\cdot\mathbf{x}^T\cdot\mathbf{K}_i\cdot\mathbf{x}+\mathbf{L}_i\cdot\mathbf{x}+\mathbf{M}_i \le 0 \quad , \quad i=1,...,n \quad . \tag{2}$$

In such a case conditioning of the decision variables space may be expressed through condition number of the matrix $\mathbf{A}$, which in turn is equal to:

$$\kappa(\mathbf{A}) = \left|\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}\right| \quad . \tag{3}$$

Of course, the lower the condition number κ the better the conditioning of decision variables space.

2. Mathematical formulation

When matrix $\mathbf{A}$ is positive definite, introduction of $\mathbf{L}\cdot\mathbf{L}^T$ (Cholesky) decomposition, and simple change of variables [3] is sufficient to achieve the lowest possible condition factor, i.e. 1 .

Situation is a little more complicated, if matrix $\mathbf{A}$ is not positive definite. In such a case $\mathbf{L}\cdot\mathbf{L}^T$ decomposition is impossible, but one may apply the SVD [4] algorithm to decompose symmetrical non negatively defined matrix $\mathbf{A}$ into:

$$\mathbf{A} = \mathbf{U}^T\cdot\mathbf{D}\cdot\mathbf{U} \quad , \tag{4}$$

where:

$\mathbf{U}$ – orthonormal matrix ($\mathbf{U}^T=\mathbf{U}^{-1}$),

$\mathbf{D}$ – diagonal, non negative matrix containing moduli of eigenvalues of $\mathbf{A}$.

Since $\mathbf{D}$ is diagonal and nonnegative, one may decompose it further into the product of three diagonal matrices:

$$\mathbf{D} = \mathbf{S}\cdot\mathbf{J}\cdot\mathbf{S} \quad , \tag{5}$$

where:

$\mathbf{S}$ – diagonal matrix of square roots of eigenvalues contained in $\mathbf{D}$, with an exception of singular values which are replaced by 1 ,

$\mathbf{J}$ – unit matrix, with an exception of singular value locations in $\mathbf{D}$, at which 1 are replaced by 0 .

Thus, finally matrix $\mathbf{A}$ may be expressed as a product of three matrices, of which one is orthonormal, one is diagonal and nonsingular, and the last one is diagonal and singular:

$$\mathbf{A} = \mathbf{U}^T\cdot\mathbf{S}\cdot\mathbf{J}\cdot\mathbf{S}\cdot\mathbf{U} \quad . \tag{6}$$

In such a manner the singularity of matrix $\mathbf{A}$ is transferred to and contained in matrix $\mathbf{J}$.

Introducing a substitution:

$$\mathbf{y} = \mathbf{S}\cdot\mathbf{U}\cdot\mathbf{x} \quad , \tag{7}$$

and determining a reverse relationship:

$$\mathbf{x} = \mathbf{S}^{-1}\cdot\mathbf{U}^T\cdot\mathbf{y} \quad , \tag{8}$$

one may express formulas (1) and (2) in terms of the new decision variables $\mathbf{y}$:

$$\min_{\mathbf{y}} F(\mathbf{y}) = \tfrac{1}{2}\cdot\mathbf{y}^T\cdot\mathbf{J}\cdot\mathbf{y} \quad , \tag{9}$$

at:

$$G_i(\mathbf{y}) = \left(\tfrac{1}{2}\cdot\mathbf{y}^T\cdot\mathbf{U}\cdot\mathbf{S}^{-1}\cdot\mathbf{K}_i+\mathbf{L}_i\right)\cdot\mathbf{S}^{-1}\cdot\mathbf{U}^T\cdot\mathbf{y}+\mathbf{M}_i \quad . \tag{10}$$

In such situation gradients of the objective function $\nabla F(\mathbf{y})$ and constraints $\nabla G_i(\mathbf{y})$, used to determine new search direction may be expressed in new decision variables as:

$$\begin{aligned}\nabla F(\mathbf{y}) &= \mathbf{J}\cdot\mathbf{y}\\ \nabla G_i(\mathbf{y}) &= \left(\mathbf{y}^T\cdot\mathbf{U}\cdot\mathbf{S}^{-1}\cdot\mathbf{K}_i+\mathbf{L}_i\right)\cdot\mathbf{S}^{-1}\cdot\mathbf{U}^T\end{aligned} \quad . \tag{11}$$

The formula for $\nabla G_i(\mathbf{y})$ may be expressed in old decision variables $\mathbf{x}$ as well:

$$\nabla G_i(\mathbf{y}) = \left(\mathbf{x}^T \cdot \mathbf{K}_i + \mathbf{L}_i\right) \cdot \mathbf{S}^{-1} \cdot \mathbf{U}^T \quad . \tag{12}$$

3. Algorithm

In general two algorithmic approaches are possible in search for the optimum solution. First one may be briefly described as: move to the new decision variables space and search for the solution there, moving back to the original one only after the optimization problem has been solved. The second approach may be summed up as: shift between the old and new sets of decision variables during each iteration – namely calculate as much as possible in old decision variables space (constraint values and gradients) and then transfer results to the new space and there determine search direction and step size.

Both approaches have advantages and disadvantages. If the first approach is to be used, all constraints have to be expressed in terms of new decision variables – a process which, though performed only once, may be very time consuming and clumsy, but then all the remaining operations are executed on the new decision variables. In the second approach one does not need to express constraints in terms of the new decision variables, but at the expense of operating continuously in two separate variable spaces.

It is worth to note, that in both approaches the SVD decomposition of matrix $\mathbf{A}$ (the most time consuming single procedure) is performed only once at the beginning of calculations.

The second approach was tested in current paper.

Information flow logic and the sequence of operations for this approach are presented in Fig. 1.

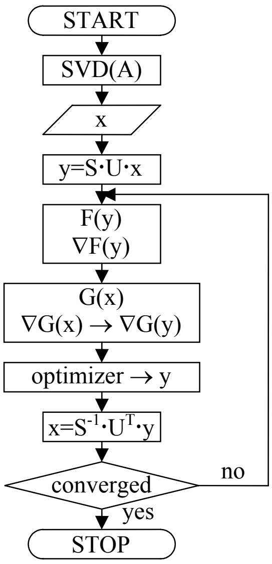

Figure 1: Information flow logic in the code

4. Numerical implementation

The set of procedures in Fortran programming language, performing all the necessary operations outlined above has been implemented as a part of FDM optimization package [2]. The implementation is transparent to the final user, i.e. the user needs not to be concerned with the internal details of the code. He has to supply the matrix $\mathbf{A}$ and procedures to compute values and gradients of constraints only.

5. Tests

Numerical tests were performed computing solution of an elastic-plastic shakedown problem [1]. Several cases differing in material data (yield limit and hardening ratio) and thus in optimization problem size were solved. The results representative of all solved cases are presented in the Table 1. These seem to indicate, that while calculation time for one iteration is on average substantially higher when the SVD decomposition routine is enabled, the number of iterations performed in order to arrive at optimum solution is at least by an order of magnitude lower. In general, when using the same optimization strategy, the SVD enabled optimization routine is about four times as effective as the standard one in terms of total time spent on calculations. The time spent on performing the decomposition itself is negligible in comparison to the total calculation time.

The quality of computed results is precisely the same for standard and SVD enabled optimization in all cases solved.

Table 1. Numerical results (calculation times in seconds);

Case	Decision variables	Cons-traints	Iterations	Total time	Time per iteration
STD	108	17	144	3.48	0.024
SVD	108	17	49	3.46	0.071
STD	256	44	5955	66.27	0.011
SVD	256	44	211	9.28	0.044
STD	396	76	47197	1081.59	0.023
SVD	396	76	1595	157.39	0.099

STD – standard set of optimization routines [2],
SVD – SVD enhanced set of optimization routines [2].

6. Conclusions

An idea of speeding up a solution of nonlinear optimization problem through application of SVD has been proposed. The method is transparent to the user of optimization code. Numerical tests performed so far indicate that effectiveness of the method is on par with the effectiveness of standard approach in case of small optimization problems (up to about one hundred decision variables and twenty nonlinear constraints) and increases rapidly with growing problem size in spite of the fact, that on average one iteration in the new method takes about 4 times as much time as in the old one.

The final solution quality of the optimization problem at hand remained the same for SVD enabled and SVD disabled methods as long as the same termination criteria were applied.

References

[1] Cecot, W., Orkisz, J., Prediction of Actual Residual Stresses Resulting from Cyclic Loading in Kinematic Hardening Material, *Proceedings of the International Conference COMPLAS V*, Barcelona, pp. 1879-1891, 1997.

[2] Orkisz J., Pazdanowski M., On a new Feasible Directions solution approach in constrained optimization, *The Finite Element Method in the 1990's*, Springer Verlag, pp. 621-632, 1991.

[3] Orkisz, J., Pazdanowski, M., On increasing the convergence of the new Feasible Directions Method in nonlinear optimization, *Proceedings of the X PCCM*, Świnoujście, pp. 591-597, 1991.

[4] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in Fortran*, Cambridge University Press, 1992.