Computer Assisted Methods in Engineering and Science, **21**: 141–150, 2014.

SVD as a preconditioner in nonlinear optimization

Michał Pazdanowski
Cracow University of Technology
Warszawska 24, 31-155 Cracow, Poland
e-mail: plpazdan@cyf-kr.edu.pl

Finding a solution of nonlinear constrained optimization problem may be very computer resources consuming, regardless of solution method adopted. A conceptually simple preconditioning procedure, based on singular value decomposition (SVD), is proposed in the current paper in order to speed up the convergence of a gradient based algorithm to solve constrained minimization problem having quadratic objective function. The efficiency of the proposed procedure is tested on a constrained minimization problem with quadratic objective function and quadratic constraints. Accuracy of the results obtained using proposed preconditioning method is checked and verified against the results determined without the preconditioning procedure.

Results obtained so far seem to indicate a significant speedup of the calculations at the expense of, negligible from the engineering point of view, loss of accuracy.

Keywords: numerical method, nonlinear optimization, singular value decomposition.

1. Introduction

Realistic nonlinear optimization problems are solved using iterative solution methods falling into two groups: deterministic and stochastic based ones [10, 19]. In general, stochastic based methods prove themselves when one has to deal with objective function having multiple local extremes, while gradient based procedures are efficient in case of convex objective functions exhibiting continuous differentiability, spanned over convex feasible regions [11].

Effectiveness of gradient based search algorithm is strongly dependent on number of decision variables and constraints, and conditioning of decision variables space. While number of decision variables and constraints is external to the optimization routine, being defined by the solved problem, conditioning of the decision variables space may be improved within the optimization procedure, in a manner transparent to the solved problem.

Singular value decomposition (SVD) [20] plays an important role in linear algebra, having applications in least squares problems, computing the pseudoinverse or Jordan canonical form, digital image processing and optimization [3]. Several efficient SVD algorithms exist [3, 8, 9] which formed the basis for development of computer routines, such as those contained in [18, 21, 22].

Present paper deals with a concept of improving convergence speed of gradient optimization routine [14] applied to solve the minimization problem defined by quadratic objective function:

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

and subjected to quadratic constraints:

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

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

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

where $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum (by modulus) eigenvalues of $\mathbf{A}$, respectively.

Of course the closer this condition factor κ is to unity, the better the conditioning of decision variables space.

2. Formulation of the problem

As long as the matrix $\mathbf{A}$ is positive definite, introduction of $\mathbf{L} \cdot \mathbf{L}^T$ (Cholesky) decomposition, and simple change of variables [15] is sufficient to achieve the lowest possible condition factor, i.e., 1.

Situation gets somewhat more complicated, if matrix $\mathbf{A}$, instead of being positive definite is only non-negatively defined (i.e., it is rank deficient). In such a case, arising for instance when solving a problem of elastic-plastic shake down [2, 13, 17], $\mathbf{L} \cdot \mathbf{L}^T$ decomposition is impossible, but one may apply the SVD algorithm to decompose a symmetrical, non-negatively defined matrix $\mathbf{A}$ into:

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

where $\mathbf{U}$ – orthonormal matrix ($\mathbf{U}^T = \mathbf{U}^{-1}$), $\mathbf{D}$ – diagonal, non-negatively defined matrix containing moduli of eigenvalues of $\mathbf{A}$, ordered by magnitude.

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

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

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 decomposed into a product of three square matrices, of which one is orthonormal, one is diagonal and nonsingular, while the last one is diagonal and singular:

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

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

By introducing a substitution

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

and determining an inverse relationship

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

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

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

subjected to

$$G_i(\mathbf{y}) = \left(\frac{1}{2} \cdot \mathbf{y}^T \cdot \mathbf{U} \cdot \mathbf{S}^{-1} \cdot \mathbf{K}_i + \mathbf{L}_i \right) \cdot \mathbf{S}^{-1} \cdot \mathbf{U}^T \cdot \mathbf{y} + \mathbf{M}_i \le 0, \qquad i = 1, ..., n. \tag{10}$$

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

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

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

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

3. Solution algorithm

In general, two algorithmic approaches are possible in search for the minimum solution. First of those approaches may be briefly described as: move to the $\mathbf{y}$ (new) decision variables space and search for the solution there, returning back to the $\mathbf{x}$ (original) variables space only after the optimization problem has been solved. The second approach may be summed up as: switch between the $\mathbf{x}$ (old) and $\mathbf{y}$ (new) decision variables spaces during each iteration – more precisely, calculate as much as possible in the $\mathbf{x}$ (original) decision variables space (constraint values and gradients), then transfer these results to the $\mathbf{y}$ decision variables space, determine search direction, step size, new approximate solution and check termination criteria here, then return to the $\mathbf{x}$ variables space to begin new iteration (determine constraint values and gradients).

Both approaches have advantages and disadvantages. If the first approach is to be used, all constraints are to be expressed in terms of the $\mathbf{y}$ 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 $\mathbf{y}$ decision variables. In the second approach one needs not express constraints in terms of the $\mathbf{y}$ decision variables, but at the expense of operating continuously in two separate variable spaces. This means, that the transformations (8) and (12) need to be performed once during every iteration.

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

The second approach was tested in the current paper.

Flowchart and the sequence of operations for this approach are presented in Fig. 1.

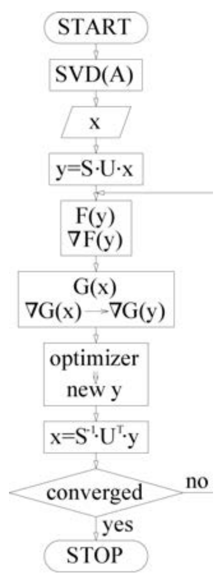

Fig. 1. Program flowchart.

4. Numerical implementation

The set of procedures in Fortran programming language, performing all the necessary operations outlined above, was implemented as a part of the feasible directions method (FDM) optimization package [14] supplemented by externally developed SVD routine [18]. This implementation is transparent to the final user of the optimization package, i.e., the user needs not be concerned with the internal functioning, and the information flow logic of these computer routines. All the user has to do is to supply the matrix **A** at the beginning of calculations and numerical routines to calculate the values and gradients of constraints for a given vector of decision variables **x**.

5. Test problem

A residual stress distribution in an elastic perfectly plastic body subject to cyclic loads exceeding its elastic bearing capacity is sought as the solution of the following two step nonlinear minimization procedure [2, 12, 13, 17], based on [7]:

step I – find a transition matrix **B** relating residual stresses $\boldsymbol{\sigma}^r$ to plastic strains $\boldsymbol{\varepsilon}^p$ as a solution of the following constrained minimization problem:

$$\min_{\boldsymbol{\sigma}^r} \Theta(\boldsymbol{\sigma}^r) = \int_V (\boldsymbol{\sigma}^r)^T \cdot \mathbf{E} \cdot \boldsymbol{\sigma}^r \cdot dV - \int_V \boldsymbol{\varepsilon}^p \cdot \boldsymbol{\sigma}^r \cdot dV, \tag{13}$$

subjected to zero boundary conditions (14) and internal equilibrium conditions (15):

$$\boldsymbol{\sigma}^r \cdot \mathbf{n} = 0 \qquad \text{on} \qquad \partial \mathrm{V}, \tag{14}$$

$$div \boldsymbol{\sigma}^r = 0 \qquad \text{in} \qquad V; \tag{15}$$

step II – find a distribution of plastic strains $\boldsymbol{\varepsilon}^p$ corresponding to statically admissible residual stress field $\boldsymbol{\sigma}^r$ as the solution of the following minimization problem:

$$\min_{\boldsymbol{\varepsilon}^p} \Psi(\boldsymbol{\varepsilon}^p) = \int_V (\boldsymbol{\varepsilon}^p)^T \cdot \mathbf{B}^T \cdot \mathbf{E} \cdot \mathbf{B} \cdot \boldsymbol{\varepsilon}^p \cdot dV, \tag{16}$$

subjected to Huber-Mises-Hencky yield condition:

$$\Phi(\mathbf{B} \cdot \boldsymbol{\varepsilon}^p + \boldsymbol{\sigma}^e) - \sigma_y \leq 0 \qquad \text{in} \qquad V; \tag{17}$$

the remaining denotations in formulas (13)–(17) are as follows: $\boldsymbol{\sigma}^e$ – momentary, time dependent, elastic solution determined as if the considered body deformed purely elastically when subject to the considered loading program, σ_y – yield stress, **n** – vector normal to the boundary, **E** – elastic compliance matrix.

One should note here, that the matrix **B**, being the solution of minimization problem (13)–(15), is square, but singular, with rank substantially smaller than dimension, and thence the matrix $\mathbf{B}^T \cdot \mathbf{D} \cdot \mathbf{B}$ in (16), corresponding to **A** in (1) is singular as well.

In order to check the efficiency of the proposed optimization routine on nonlinear minimization problems varying in the number of decision variables as well as in the number of constraints, the test problem (13)–(17) was solved several times for different levels of external load ($\boldsymbol{\sigma}^e$) applied in different locations and resulting in different plastic zone sizes. Based on the applied load level solved cases may be assigned into three groups, characterized by similar number of decision variables and nonlinear constraints, and further called *low*, *medium* and *high* (Table 1, Figs. 2–5).

Table 1. Summary of cases solved.

Case group	Constraint thickness 0.0001			Constraint thickness 0.0000001		
	Average number of		Average ratio	Average number of		Average ratio
	variables	constraints		variables	constraints	
high	268	48	5.523	351	66	5.337
medium	212	37	5.714	272	50	5.456
low	180	31	5.902	247	44	5.623

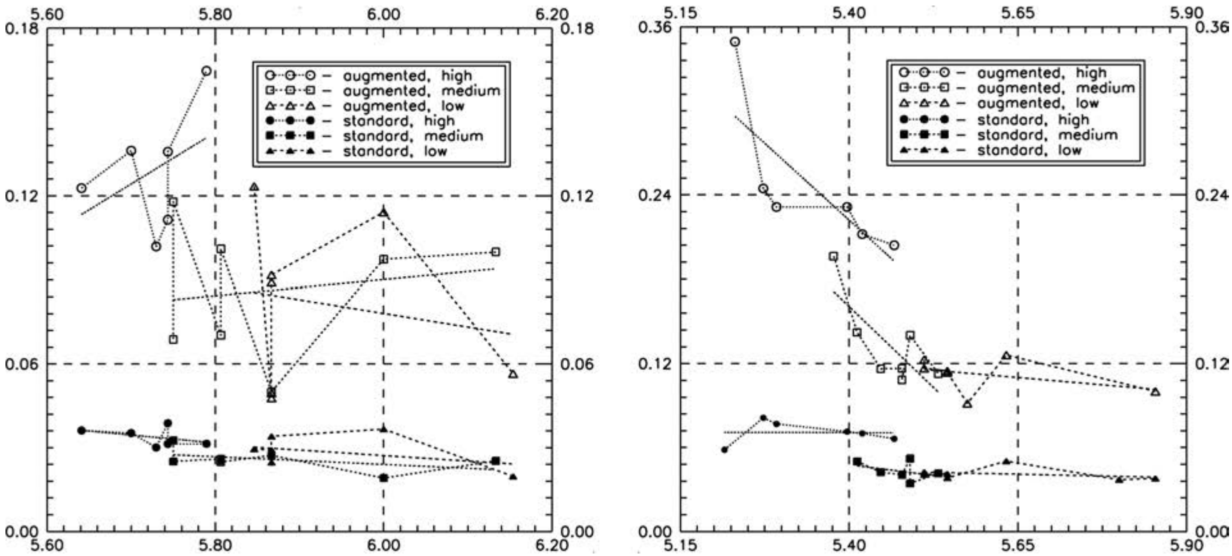

Fig. 2. Averaged single iteration time in seconds (vertical axis) for a rough estimate (left) and high quality results (right) versus the ratio of decision variables to constraints (horizontal axis).

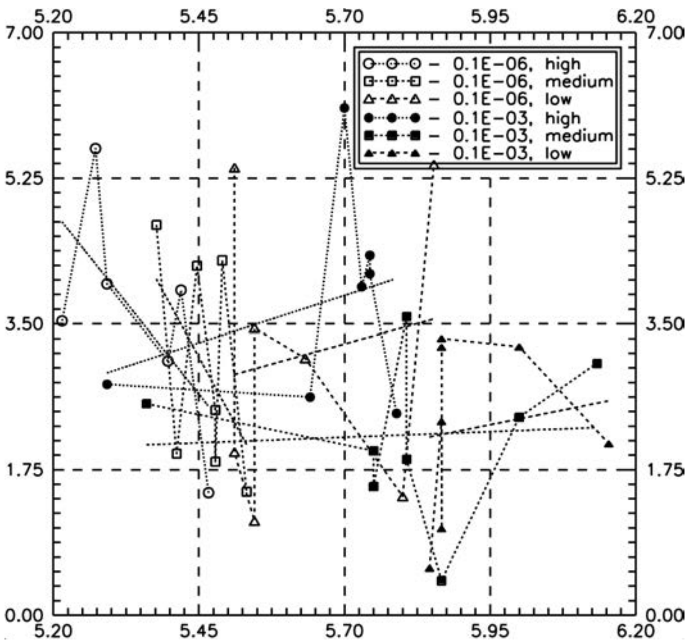

Fig. 3. Speedup ratio of augmented over standard procedure (vertical axis) versus a ratio of decision variables to constraints (horizontal axis).

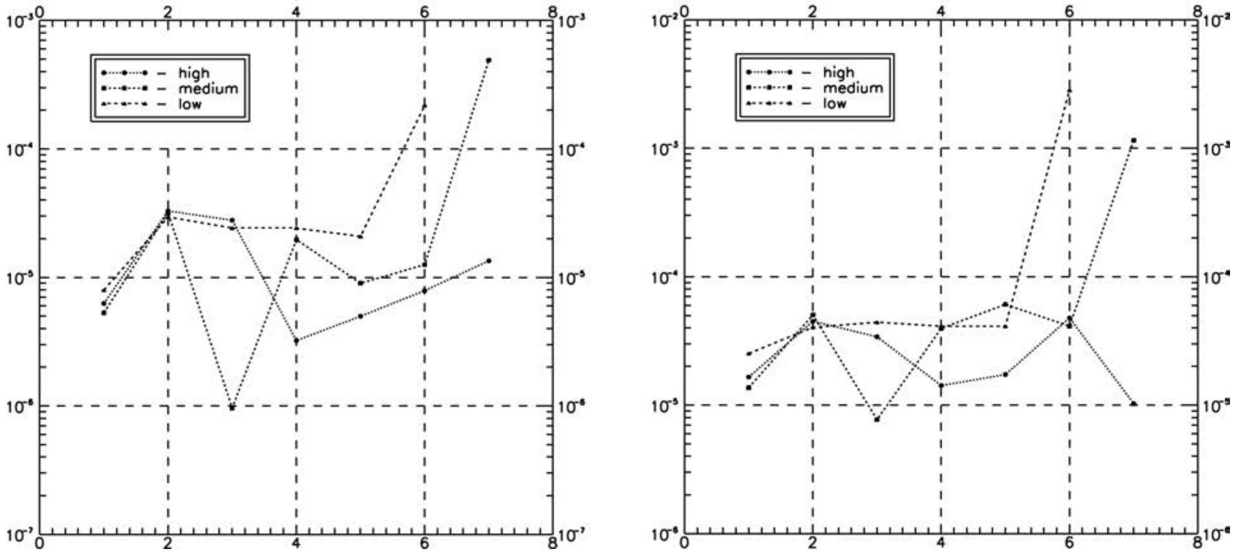

Fig. 4. Global (left) and local (right) hydrostatic pressure based error measure (vertical axis) versus case (load location) number (horizontal axis).

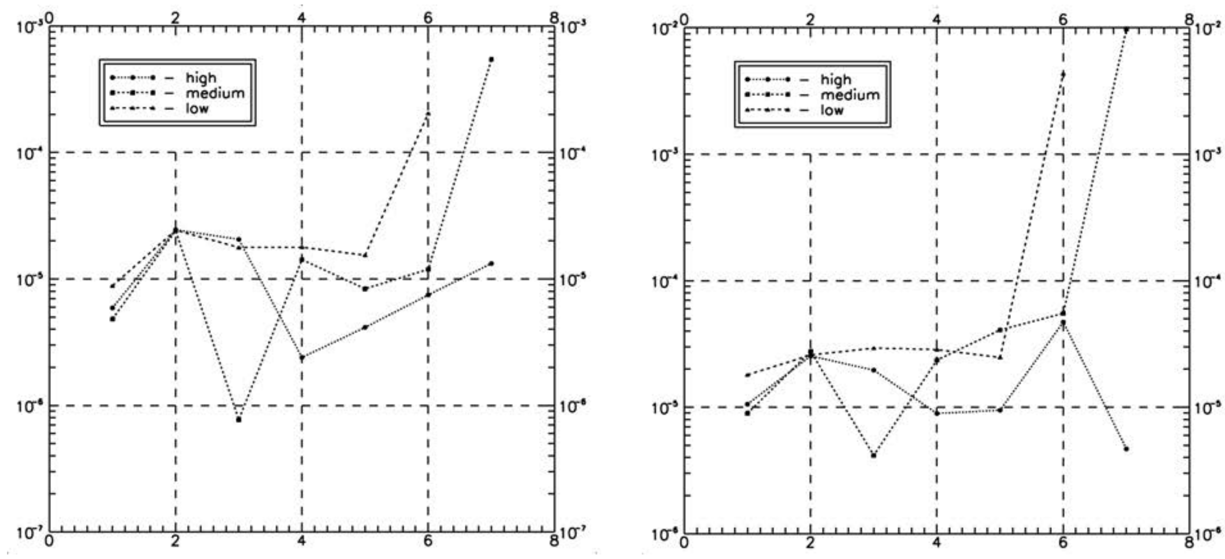

Fig. 5. Global (left) and local (right) Huber-Mises-Hencky equivalent stress based error measure (vertical axis) versus case (load location) number (horizontal axis).

Since the solution quality depends highly on the setting of termination criterion (Kuhn-Tucker optimality criterion [1] is used here), as well as constraint thickness setting, and these in turn affect the calculation time to a large degree, two termination settings were used. In order to get a rough estimate of plastic zone location and size, the threshold values of 0.0001 for constraint thickness and 0.001 for the allowed length of the normalized search direction vector projection onto active constraints (Kuhn-Tucker criterion) were applied based on previous experience [16], while the settings 1000 times smaller were used for both parameters in order to get a final, high quality estimate of both plastic strains $\boldsymbol{\varepsilon}^p$ and residual stresses $\boldsymbol{\sigma}^r$. The average ratio of decision variables to constraints for each group of cases solved is indicated in columns 4 and 7 of Table 1, as it was found out during tests, that this ratio constitutes a good single characteristic of all the optimization problems solved so far.

A subset of solved cases belonging to *high* case group is presented in Tables 2 and 3. In these tables STD denotes standard set of optimization routines, while the SVD denotes the singular value decomposition augmented set of optimization routines applied to solve the test problem. All the calculations were performed on a 2.40 MHz Intel Pentium 4M CPU equipped PC class computer operating under control of Windows XP SP3 operating system. Respective total calculation times in seconds (including initialization and execution of SVD algorithm in the SVD case), time gain factor of augmented procedure over standard one and average times per iteration, computed as total execution time divided by the number of iterations required to arrive at the optimum solution with assigned precision as well as the ratio of one iteration time using the augmented (SVD) approach over standard (STD) approach are indicated in Table 2 (rough estimate of a solution in an underestimated plastic zone) and Table 3 (high precision results in a correctly sized plastic zone). Actual calculation times were determined in milliseconds through appropriate calls to system clock from within the code.

Table 2. Rough estimate solution calculation times (in seconds).

Case	Decision variables	Constraints	Iterations	Total time	Speed up	Time per iteration	Ratio
STD	212	37	3776	113.68	3.94	0.0301	3.38
SVD			283	28.83		0.1019	
STD	220	38	2072	65.16	2.42	0.0315	5.29
SVD			163	26.86		0.1648	
STD	224	39	3735	117.36	4.32	0.0314	4.32
SVD			200	27.16		0.1358	
STD	228	40	4709	175.03	8.39	0.0372	3.66
SVD			153	20.84		0.1362	
STD	344	65	3005	219.92	2.77	0.0732	3.55
SVD			304	79.19		0.2605	
STD	392	74	9738	1002.55	3.26	0.1030	2.54
SVD			1172	307.41		0.2623	

Table 3. High quality solution calculation times (in seconds).

Case	Decision variables	Constraints	Iterations	Total time	Speed up	Time per iteration	Ratio
STD	328	60	4367	289.51	1.47	0.0663	3.08
SVD			964	196.70		0.2041	
STD	336	62	7586	532.05	3.34	0.0701	3.01
SVD			754	159.16		0.2111	
STD	340	63	6466	453.40	2.15	0.0701	3.29
SVD			913	210.85		0.2309	
STD	344	65	9852	758.04	3.98	0.0769	3.02
SVD			819	190.58		0.2327	
STD	348	66	14346	581.17	4.30	0.0405	5.04
SVD			661	135.11		0.2044	
STD	412	79	71019	4151.93	6.49	0.0585	4.61
SVD			2375	639.92		0.2694	

Average single iteration times for all of the cases solved with limited accuracy versus a ratio of decision variables to constraints in a considered case are depicted on the left side of Fig. 2, while the same data for all the cases solved with high precision are depicted on the right side of the same figure. The trend lines depicted in this figure seem to indicate that for the standard procedure single iteration calculation times are fairly independent of the problem size. For the augmented procedure trend lines depicted on left and right parts of this figure seem to indicate, that for the low quality results the calculation times increase with decreasing decision variables to constraints ratio while the opposite is true for high quality results. This may be a disadvantage while trying to solve even bigger optimization problems, as it was found out during previous tests [16, 17] (cf. also Tables 2 and 3) that in absolute terms the most time consuming to solve with high precision were problems having low number of decision variables to constraints ratio.

A ratio of total calculation time needed to solve given case using standard approach over augmented approach versus a ratio of decision variables to constraints in a considered case is depicted in Fig. 3. Trend lines depicted seem to indicate that while for low accuracy results the overall speedup ratio is low and fairly constant, for high accuracy results it seems to increase with decreasing decision variables to constraints ratio. This is encouraging, as it shows, that in spite of the trend observed in Fig. 2 for high quality results (growing single iteration time) a total calculation time gain is still possible, due to the lower global number of iterations performed.

Due to practical considerations [4] two measures are introduced to compare the differences between the residual stress distributions obtained using standard and augmented approaches. First one is based on hydrostatic pressure:

$$\sigma_H = tr(\boldsymbol{\sigma})/3, \tag{18}$$

while the second one is based on Huber-Mises-Hencky equivalent stress:

$$\sigma_{HMH} = \frac{1}{\sqrt{2}} \cdot \sqrt{\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2 + \left(\sigma_{zz} - \sigma_{xx}\right)^2 + 6 \cdot \left(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2\right)}. \tag{19}$$

Both measures are applied in both global (20) and local (21) forms:

$$\sigma_X^G = \frac{\displaystyle\int_A \left|\left(\boldsymbol{\sigma}_{STD}^r - \boldsymbol{\sigma}_{SVD}^r\right)_X\right| \cdot dA}{\displaystyle\int_A \left|\left(\boldsymbol{\sigma}_{STD}^r\right)_X\right| \cdot dA}, \tag{20}$$

$$\sigma_X^L = \frac{\max\left(\boldsymbol{\sigma}_{STD}^r - \boldsymbol{\sigma}_{SVD}^r\right)_X}{\max\left(\boldsymbol{\sigma}_{STD}^r\right)_X}. \tag{21}$$

The following denotations hold in the two above formulas: $\boldsymbol{\sigma}_{STD}^r$ – residual stress tensor in the considered body determined using standard procedure, $\boldsymbol{\sigma}_{SVD}^r$ – residual stress tensor in the considered body determined using SVD augmented procedure, σ_X^G – global hydrostatic pressure based ($X = H$) or Huber-Mises-Hencky equivalent stress based ($X = HMH$) error measure, σ_X^L – local hydrostatic pressure based ($X = H$) or Huber-Mises-Hencky equivalent stress based ($X = HMH$) error measure.

Results presented in Figs. 4 and 5 for global (left) and local (right) measures, grouped in series based on applied load levels, indicate that there is no noticeable loss of accuracy in the final results when the singular value decomposition augmented optimization procedure is used. This may be confirmed as well by the analysis of sample σ_{zz} residual stress distributions presented in Fig. 6. For better readability those distributions are split into positive (tensile) and negative (compressive) parts.

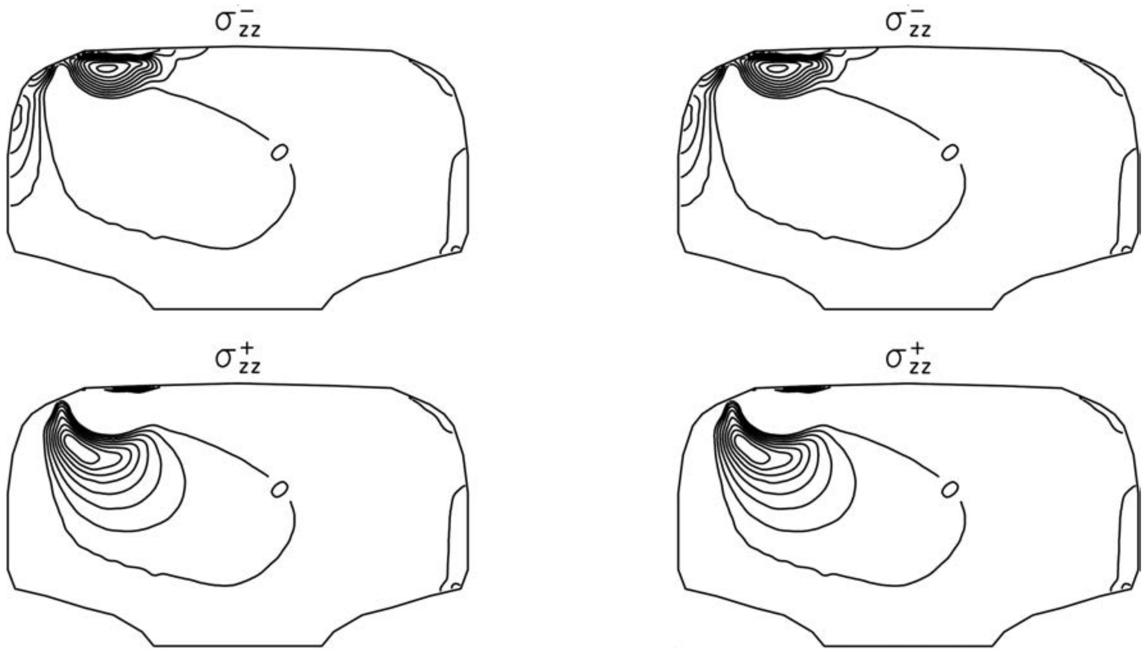

Fig. 6. σ_{zz} residual stress contour plots in a railhead, standard results (left) versus SVD augmented results (right); contour levels 21 MPa for compression and 14 MPa for tension.

6. Concluding remarks

An idea of speeding up the solution of nonlinear optimization problem through application of SVD as a preconditioner was proposed. An application of the computer routines developed according to this idea is transparent to the user of the optimization code. Numerical tests performed indicate, that effectiveness of the method tends to improve with problem size, when high precision solution is sought, on average reaching more than four times speedup factor with maximum of almost 6.5 achieved for the largest problem solved. When only a rough estimate of a solution is sought, though the speedup factor varies between almost 2.5 and almost 8.5, there is no such a general trend in data, and the speedup factors for the smallest and the largest problems solved are relatively close, averaging 3.5.

The final solution quality of the optimization problem considered as the test example remained the same for SVD enabled and SVD disabled solution approaches, as long as the same termination criteria were applied.

Although the current application of proposed procedure is limited to quadratic functions, the basic idea is quite general and the SVD decomposition proposed here may be applied to any objective function, which may be locally approximated by a quadratic one. Unfortunately this may be achieved at the expense of efficiency, as every change of local approximation will necessitate new SVD decomposition, constituting the most time consuming single step of the algorithm. This does require additional testing.

References

[1] S.P. Bod, L. Vandenberghie. *Convex optimization.* Cambridge University Press, 2004.

[2] W. Cecot, J. Orkisz. Prediction of actual residual stresses resulting from cyclic loading in kinematic hardening material. *Proc. of COMPLAS V*, 1879–1891, Barcelona, 1997.

[3] T.F. Chan. An improved algorithm for computing the singular value decomposition. *ACM Transactions on Mathematical Software*, **8**: 72–83, 1982.

[4] K. DangVan, B. Griveau, O. Message. On new multiaxial fatigue criterion: theory and application. *Biaxial and multiaxial fatigue*, 479–496, MEP, London, 1989.

[5] J. Demmel, W. Kahan. Accurate singular values of bidiagonal matrices. *Society for Industrial and Applied Mathematics. Journal on Scientific and Statistical Computing*, **11**: 873–912, 1990.
[6] R. Fletcher. *Practical methods of optimization*. Wiley, 2000.
[7] J.B. Martin. *Plasticity – fundamentals and general results*. MIT Press, 1975.
[8] J.C. Nash. A one-sided transformation method for the singular value decomposition and algebraic eigenproblem. *Computer Journal*, **18**: 74–76, 1975.
[9] J.C. Nash, S. Shlien. Simple algorithms for the partial singular value decomposition. *Computer Journal*, **30**: 268–275, 1987.
[10] G.L. Nemhauser, A.H.G. Rinooy Kan, M.J. Todd. *Optimization*. North Holland, 1989.
[11] J. Nocedal, S.J. Wright. *Numerical optimization*. Springer, 2006.
[12] J. Orkisz, A. Harris. Analysis of residual stresses at shakedown, a hybrid approach. *Theoretical and Applied Fracture Mechanics*, **9**: 109–121, 1988.
[13] J. Orkisz, O. Orringer, M. Hołowiński, M. Pazdanowski, W. Cecot. Discrete analysis of actual residual stresses resulting from cyclic loadings. *Computer and Structures*, **35**: 397–412, 1990.
[14] J. Orkisz, M. Pazdanowski. On a new feasible directions solution approach in constrained optimization. In: E. Onate, J. Periaux, A. Samuelsson, eds. *The Finite Element Method in the 1990's*, 621–632. Springer Verlag, 1991.
[15] J. Orkisz, M. Pazdanowski. On increasing the convergence of the new feasible directions method in nonlinear optimization. *Proc. of the X PCCM*, 591–597, Świnoujście, 1991.
[16] J. Orkisz, M. Pazdanowski. Analysis of residual stresses by the generalized finite difference method. *Proc. of COMPLAS IV*, 189–200, Barcelona, 1995.
[17] M. Pazdanowski. On estimation of residual stresses in rails using shake-down based method. *Archives of Transport*, **XXII**(3): 319–336, 2010.
[18] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical recipes in Fortran*. Cambridge University Press, Cambridge, 1992.
[19] J.C. Spall. *Introduction to stochastic search and optimization: estimation, simulation and control*. Wiley, 2003.
[20] G.W. Stewart. On the early history of the singular value decomposition. *SIAM Review,* **35**: 551–566, 1993.
[21] *LAPACK users' guide*. SIAM, 1999.
[22] *GNU scientific library reference manual – third edition*. Network Theory, 2009.