Model Reduction for Cold Rolling Processes

Jens Seidel* and Oliver G. Ernst*

The numerical simulation of rolling processes and its acceleration by model reduction techniques is considered. The physical model is based on minimizing the deformation power by using a flow formulation and the equivalent stress and strain. The reduction uses proper orthogonal decomposition (POD) for linear terms together with the discrete empirical interpolation method (DEIM) for nonlinear terms. In particular, the hardening can be accelerated considerably. The solution can be represented by substantially fewer degrees of freedom while maintaining acceptable accuracy.

1. Introduction

The numerical simulation of cold rolling remains a time consuming computational challenge, especially when many calculations are to be performed as is necessary for process optimization. The two basic options for speeding up the calculation are either simplifying the mathematical model and performing faster but less accurate calculations[1] or applying mathematically inspired methods of model reduction to models representing the full physics. This study follows the second approach, applying proper orthogonal decomposition (POD) in combination with the discrete empirical interpolation method (DEIM) to a finite element model of cold rolling.

2. Mathematical Model

A plane strain assumption and thus a two-dimensional calculation is justified by the typical dimensions of the rolled strip. It is assumed that the whole domain is under plastic deformations. Elastic deformations are negligible as the strain can reach large values and are ignored. The material is assumed to be isotropic and incompressible, the von Mises yield criterion is used. For simulation an Eulerian coordinate system is chosen and the strain rate tensor $\dot{\varepsilon}$ is expressed by a velocity field $\boldsymbol{u} = (u_1, u_2)$, which is incompressible, hence $\nabla \cdot \boldsymbol{u} = 0$ and $\dot{\varepsilon}_{ij} = (u_{i,j} + u_{j,i})/2$. The equivalent strain rate is given by

$$\dot{\varepsilon}_V = \sqrt{\frac{2}{3}\left(\dot{\varepsilon}_{11}^2 + 2\dot{\varepsilon}_{12}^2 + \dot{\varepsilon}_{22}^2\right)}\,.$$

The velocity is determined as the minimizer of the deformation power $P(\boldsymbol{u})$ in the considered geometry[2] (cf. **Figure 1**). It consists of four parts $P = P_p + P_f + P_v + P_b$ representing the plastic deformation described by $P_p = \int_V k_f \dot{\varepsilon}_V \mathrm{d}\boldsymbol{x}$, the friction between workpiece and roll $P_f = \int_{S_C} mk|u_{\mathrm{rel}}|\,\mathrm{d}s$, and boundary terms such as external forces $P_b = -\int_{S_F} \boldsymbol{F} \cdot \boldsymbol{u}\,\mathrm{d}s$. The incompressibility constraint is imposed by a Lagrange multiplier $\lambda(\boldsymbol{x})$ in the term $P_v = \int_V \lambda \dot{\varepsilon}_{\mathrm{vol}} \mathrm{d}\boldsymbol{x}$. λ can also be regarded as mean stress. The notation is summarized in **Table 1**.

Restricting $\boldsymbol{u}$ and λ to a stable pair of finite-dimensional subspaces consisting of piecewise biquadratics for $\boldsymbol{u}$ and piecewise bilinears for λ and performing numerical integration, the power can be written discretized as[3]

$$P_h(\boldsymbol{u}, \boldsymbol{\lambda}) = \sum_i k_f \sqrt{\boldsymbol{u}^T A_i \boldsymbol{u}} + \|L\boldsymbol{u} - \boldsymbol{r}\|_1 + \boldsymbol{\lambda}^T B\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u}. \qquad (1)$$

In (1) and the following, $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ shall refer to the vectors of finite element degrees of freedom of $\boldsymbol{u}$ and λ. In addition, A_i denotes the element stiffness matrix in the ith quadrature node, B, L are matrices representing the incompressibility and friction term, respectively, $\|\cdot\|_1$ is the ℓ_1-norm (sum of absolute values of vector components), $\boldsymbol{r}$ is a vector characterized by the roll speed, and finally $\boldsymbol{f}$ is a vector describing external forces.

2.1. Hardening

In the employed hardening model

$$k_f(\boldsymbol{x}) = Ae^{m_1\theta}\varepsilon_V^{m_2}\dot{\varepsilon}_V^{m_3}\mathrm{e}^{m_4/\varepsilon_V} \qquad (2)$$

the yield stress k_f depends on the strain ε_V, the strain rate $\dot{\varepsilon}_V$, and the temperature θ. Of these, only $\dot{\varepsilon}_V$ is obtained directly from the finite element calculation, whereas ε_V and θ require further computation.

[*] J. Seidel, O. G. Ernst
Department of Mathematics, Chemnitz University of Technology,
D-09107 Chemnitz, Germany
Email: jens.seidel@mathematik.tu-chemnitz.de

DOI: 10.1002/srin.201300220

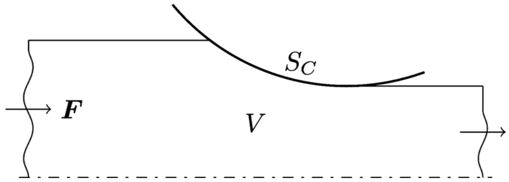

Figure 1. Rolling geometry.

P	Power
$\boldsymbol{u}$	Velocity
u_{rel}	Relative velocity $\boldsymbol{u}$–$\boldsymbol{u}_{\text{roll}}$ on boundary
k_{f}	Yield stress
$\dot{\varepsilon}_V$	Equivalent strain rate
$\boldsymbol{F}$	Boundary force
λ	Lagrange multiplier, mean stress
$\dot{\varepsilon}_{\text{vol}}$	Volumetric strain rate $\nabla \cdot \boldsymbol{u}$
V	Domain
S_{F}	Boundary left/right
S_{C}	Interface boundary
m	Friction coefficient
k	Shear yield stress $k_f/\sqrt{3}$

Table 1. Notations used in the model.

2.1.1. Calculation of Strain

To evaluate $\varepsilon_V(\boldsymbol{x})$ the strain rate $\dot{\varepsilon}_V$ is integrated along the particle trajectory $\boldsymbol{x}(t)$ beginning at the point $\boldsymbol{x}_0$ where the particle enters the domain (cf. **Figure 2**)

$$\varepsilon_V(\boldsymbol{x}(t)) = \varepsilon_V(\boldsymbol{x}_0) + \int_0^t \dot{\varepsilon}_V(\boldsymbol{x}(t'))\mathrm{d}t'. \tag{3}$$

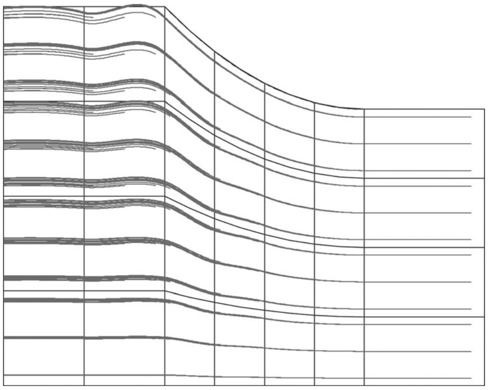

Figure 2. Particle trajectories for strain calculation.

Here $\varepsilon_V(\boldsymbol{x}_0)$ contains initial deformations resulting, e.g., from a previous roll pass and the trajectory is determined by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(\boldsymbol{x}(t)), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0.$$

Since the strain is required at all quadrature nodes of the finite element model this calculation is costly.

2.1.2. Calculation of Temperature

The temperature is determined by the Fourier law

$$\frac{\partial\theta}{\partial t} = -\boldsymbol{u}\cdot\nabla\theta + \frac{1}{\rho c}\nabla\cdot(\kappa\nabla\theta) + \frac{1}{\rho c}k_f\dot{\varepsilon}_V \tag{4}$$

containing a dominant convection term and a source term resulting from plastic deformations. The following boundary conditions were used ($b = \sqrt{\rho\kappa c}$, b_{roll} same with rolling parameters):

$$\begin{aligned}
&\text{Top left/right: } -\kappa\frac{\partial\theta}{\partial\boldsymbol{n}} = h_{\text{air}}(\theta-\theta_{\text{air}}) + \varepsilon_{\text{rad}}\sigma(\theta^4-\theta_{\text{air}}^4),\\
&\text{Roll contact: } -\kappa\frac{\partial\theta}{\partial\boldsymbol{n}} = h_{\text{roll}}(\theta-\theta_{\text{roll}}) - \frac{b}{b+b_{\text{roll}}}mk|u_{\text{rel}}|,\\
&\text{Left: } \theta = \theta_0(\boldsymbol{x}),\\
&\text{Bottom: } \frac{\partial\theta}{\partial\boldsymbol{n}} = 0.
\end{aligned} \tag{5}$$

The occurring material constants are thermal conductivity κ, density ρ, specific heat capacity c, heat transfer coefficient h, Stefan-Boltzmann constant σ, and emissivity ε_{rad}. In addition, θ_0 is the prescribed temperature, θ_{air} and θ_{roll} the temperature of the environment and roll, respectively, and $\boldsymbol{n}$ is the outer unit normal vector.

2.2. Minimization of the Power

To find a stationary point of the power functional $P_h(\boldsymbol{u}, \boldsymbol{\lambda})$ as a function of the finite element degrees of freedom for the velocity $\boldsymbol{u}$ and multiplier λ, Newton's method is used in damped form to solve $\nabla P_h = 0$. This requires a regularization to ensure that the functional is differentiable. The easiest solution is to add a small constant under the square root of the plastic deformation term and to control its size by the amount of admissible error for P_h. The absolute value $|u_{\text{rel}}|$ in the friction term can be regularized using $(2/\pi)u_{\text{rel}}\tan^{-1}(u_{\text{rel}}/u_0)$, where u_0 is a small constant compared to the average relative velocity.[2,4]

3. Model Reduction

In particular due to the nonlinear hardening law (2), solving the full discretized problem (1) is not feasible in real time applications. The most time consuming part is the calculation of the strain ε_V, as many line integrals following the path of the particles have to be determined. The linear systems in each Newton step and the temperature calculation can also be computationally expensive if the number of elements is high. The main idea of the proper orthogonal decomposition[5,6] model reduction approach is to restrict the problem to a lower dimensional subspace, requiring fewer (and problem-adapted) degrees of freedom. To avoid the dependence on the unreduced problem size in nonlinear terms, we also employ the discrete empirical interpolation method.[7] Together, POD and DEIM provide speed-up potential for not only (cold) rolling processes but for further applications such as simulation of temperature or stress, but also the parametrization of material properties such as microstructures. POD begins by solving the full finite element problem (1) for a sufficiently large and well-chosen set of values of the parameters on which the solution depends, among these the quantities $m_1, \ldots, m_4$ from the hardening law (2) as well as roll speed, friction coefficient m and boundary forces $\boldsymbol{F}$. This typically time consuming task is known as the *offline phase* and is used to construct the problem-adapted subspaces for subsequent solves for general parameter values with the reduced model in the so-called *online phase*.

Given a number n_s of solutions to the full finite element problem for a representative set of parameter values, we refer to these as *snapshots* and arrange them as the columns of the *snapshot matrix*

$$Y = \left[\boldsymbol{y}_1 | \cdots | \boldsymbol{y}_{n_s}\right],\ \boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\lambda}_i \end{bmatrix} \in \mathbb{R}^N,\ i = 1, \ldots, n_s. \tag{6}$$

POD now extracts a subspace of dimension $k \ll N$ from the snapshot matrix column space by computing the best approximation of Y by a matrix of rank k. This best approximation is obtained by computing the *singular value decomposition* (SVD)

$$Y = U\Sigma V^T \text{ with } V^T V = I,\ U^T U = I,\ \Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}$$

of the snapshot matrix. The matrices $U \in \mathbb{R}^{N \times N}$ and $V \in \mathbb{R}^{n_s \times n_s}$ are orthogonal and $\sum_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ is a diagonal matrix containing in descending order all positive singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$ of Y. It is well known that the first k columns $X := U_k$ of U contain an orthonormal basis of the range space of the best approximation to the column space of Y. Alternatively it is possible to calculate U and hence X using the matrix YY^T. The matrix U then consists of the normalized eigenvectors of YY^T and the eigenvalues are the squared singular values σ_i^2.

The reduced approximations are now constrained to lie in the space spanned by the k columns of X, i.e., generic solutions of the discrete problem (1) are approximated as

$$\boldsymbol{y} \approx \boldsymbol{x}_1 y_k^1 + \cdots + \boldsymbol{x}_k y_k^k = X\boldsymbol{y}_k, \tag{7}$$

containing only $k \ll N$ degrees of freedom. The error of best approximation (in mean square sense) of any vector in the column space of Y by vectors of the form (7) is known to be bounded by $\sigma_{k+1}^2 + \cdots + \sigma_r^2$. This means that the decay of the singular values of the snapshot matrix determines how large k has to be chosen. **Figure 3** shows a typical plot of the singular value decay.

3.1. Linear Terms

Once the subspace for the reduction is known it is possible to formulate a reduced problem. As an example for the linear case we consider the stationary Fourier equation (4) with vanishing radiation ($\varepsilon_{\mathrm{rad}} = 0$). Discretizing this model using standard finite elements[8] results in a linear system of equations

$$K\boldsymbol{T} = \boldsymbol{f} \tag{8}$$

for the vector $\boldsymbol{T}$ of temperature degrees of freedom. Similar to above we restrict the vector $\boldsymbol{T}$ to a reduced subspace X and replace $\boldsymbol{T}$ by $X\boldsymbol{T}_k$ where $\boldsymbol{T}_k \in \mathbb{R}^k$. Applying the Galerkin method with subspace X to the linear system (8) yields the reduced problem

$$X^T K X \boldsymbol{T}_k = X^T \boldsymbol{f} \quad \text{or} \quad K' \boldsymbol{T}_k = \boldsymbol{f}'. \tag{9}$$

Here, $K' := X^T K X$ and $\boldsymbol{f}' := X^T \boldsymbol{f}$. The reduced problem (9) is of order k, which is typically much smaller than the order N of the full finite element approximation (8). One shortcoming is that the matrix K' is, in contrast to K, in general not sparse. The calculation of K' and X can be performed in the offline phase whereas the solution of the reduced system can be performed in the online phase for different right hand sides at nearly (depending on the size of k) real time.

Other terms of the power functional such as the one for incompressibility $\boldsymbol{\lambda}^T B \boldsymbol{u}$ and equivalent strain rate $\sqrt{\boldsymbol{u}^T A \boldsymbol{u}}$ are handled analogously. For the strain rate,

$$\sqrt{(X\boldsymbol{u}_k)^T A (X\boldsymbol{u}_k)} = \sqrt{\boldsymbol{u}_k^T (X^T A X) \boldsymbol{u}_k} \tag{10}$$

is obtained and $A' := X^T A X$ can also be pre-computed in the offline phase. The structure of the problem is preserved,

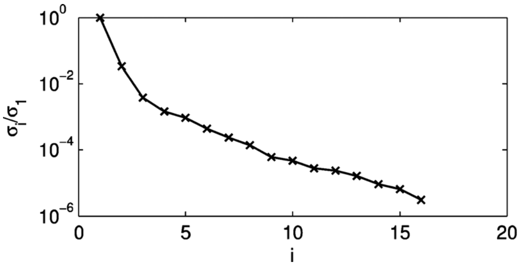

Figure 3. Decay of the scaled singular values σ_i/σ_1 for variation of hardening coefficients $m_1, \ldots, m_4$.

only the individual vectors and matrices change, allowing to reuse most parts of the software for the solution of the unreduced system. As k becomes larger it may be possible that A' contains more entries than A, but in this case it is possible to treat the strain rate term using DEIM, which will be explained in the next section.

3.2. Nonlinear Terms

POD works well for linear problems such as (8) but can also be applied to the nonlinear case. Let us consider again the Fourier equation but with a nonvanishing radiation term. The stiffness matrix K does not change compared to the linear case but the right hand side $\boldsymbol{f}=\boldsymbol{f}(\boldsymbol{T})$ is now state dependent due to the θ^4 term in (5). Applying the POD method and using $\boldsymbol{T} \approx X\boldsymbol{T}_k$ results similar to (9) in

$$K'\boldsymbol{T}_k = X^T \boldsymbol{f}(X\boldsymbol{T}_k). \tag{11}$$

To solve (11) we first consider the unreduced equation and apply Newton's method. The problem is already solved in few iterations as the effect of radiation is weak for cold rolling processes. Starting with an initial guess $\boldsymbol{T}^{(0)}$ (obtained, for example, by solving the linear system without radiation) it reads:

$$\boldsymbol{T}^{(n+1)} = \boldsymbol{T}^{(n)} - (K - \nabla_{\boldsymbol{T}}\boldsymbol{f}(\boldsymbol{T}^{(n)}))^{-1}(K\boldsymbol{T}^{(n)} - \boldsymbol{f}(\boldsymbol{T}^{(n)})).$$

For the reduced system the iteration reads

$$\boldsymbol{T}_k^{(n+1)} = \boldsymbol{T}_k^{(n)} - (K' - X^T\nabla_{\boldsymbol{T}}\boldsymbol{f}(X\boldsymbol{T}_k^{(n)})X)^{-1}(K'\boldsymbol{T}_k^{(n)} - X^T\boldsymbol{f}(X\boldsymbol{T}_k^{(n)})). \tag{12}$$

The calculation of the gradient of $\boldsymbol{f}$ is computationally expensive because it is of size $N \times N$. Also the multiplication with the $k \times N$ matrix X^T from the left and the $N \times k$ matrix X from the right results again in a small $k \times k$ matrix but the matrix products are time consuming. Similar problems arise in the general case, e.g., for the hardening term, and these must be addressed before we can apply model reduction efficiently.

DEIM offers a solution to this problem by finding a different representation for the nonlinearity $\boldsymbol{f}$ in (11). Considering again the general case and denoting the variable by $\boldsymbol{u}$, assume we wish to approximate a function $\boldsymbol{f}(\boldsymbol{u}) : \mathbb{R}^N \to \mathbb{R}^M$ with $M \gg 1$. Similar to (7) the goal is to represent the function values by a low dimensional subspace, i.e.,

$$\boldsymbol{f}(X\boldsymbol{u}_k) \approx U\boldsymbol{c} \tag{13}$$

with a subspace spanned by $U \in \mathbb{R}^{M \times m}$ representing the nonlinearity, a coefficient vector $\boldsymbol{c} \in \mathbb{R}^m$, where $m \ll M$ is the dimension of the subspace. This is an overdetermined linear system of equations which, in general, has no solution. We try to find an approximation by restricting it and proceed as follows:

1. Select m row indices $i_1,\ldots,i_m$ of U.
2. Determine $\boldsymbol{c}$ from the $m \times m$ system:

$$\boldsymbol{f}_{(\boldsymbol{i})}(X\boldsymbol{u}_k) = U_{(\boldsymbol{i})}\boldsymbol{c}. \tag{14}$$

Using the given n_s snapshots $\boldsymbol{u}_j$ a natural requirement is that all $\boldsymbol{f}(\boldsymbol{u}_j)$ are approximated well. Let us denote by F the matrix of function evaluations of the snapshots $F := [\boldsymbol{f}(\boldsymbol{u}_1)|\cdots|\boldsymbol{f}(\boldsymbol{u}_{n_s})]$. As for the treatment of the linear terms the first m left singular vectors $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_m$ of F are the best choice for U. The DEIM algorithm for finding the indices $\boldsymbol{i}$ uses a greedy strategy:

```
INPUT: $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_m \in \mathbb{R}^M$
OUTPUT: $\boldsymbol{i} = [i_1|\ldots|i_m]$

  • $i_1 = \arg\max\{|\boldsymbol{f}_1|\}$, $U = [\boldsymbol{f}_1]$, $\boldsymbol{i} = [i_1]$
  • for $j = 2$ to $m$
      – $\boldsymbol{f} := \boldsymbol{f}_j$
      – Solve $U_{(\boldsymbol{i})}\boldsymbol{c} = \boldsymbol{f}_{(\boldsymbol{i})}$ for $\boldsymbol{c}$
      – $\boldsymbol{r} = \boldsymbol{f} - U\boldsymbol{c}$
      – $i_j = \arg\max\{|\boldsymbol{r}|\}$
      – $U := [U|\boldsymbol{f}]$, $\boldsymbol{i} := [\boldsymbol{i}|i_j]$
```

4. Results and Discussion

With the DEIM algorithm a nonlinear function from a high-dimensional space can be efficiently reduced. The given variational problem (1) is scalar, but can nevertheless be expressed in the required vector form. All except the plastic deformation term are handled by simply substituting $\boldsymbol{u}$ by $X_{\boldsymbol{u}}\boldsymbol{u}_k$, where according to (6)

$$X = \begin{bmatrix} X_{\boldsymbol{u}} \\ X_{\boldsymbol{\lambda}} \end{bmatrix}.$$

The Lagrange multiplier can be omitted if it is not required for postprocessing as the considered subspace ensures incompressibility. Otherwise it is analogously replaced by $\boldsymbol{\lambda} \approx X_\lambda\, \boldsymbol{\lambda}_k$.

The plastic deformation term including hardening in k_f can be written as a sum over all M quadrature points

$$P_{p,h}(\boldsymbol{u}) = \sum_{i=1}^{M} k_f^i \dot{\varepsilon}_V^i = \boldsymbol{k}_{\boldsymbol{f}}^{\mathrm{T}} \dot{\boldsymbol{\varepsilon}}_V.$$

DEIM can be applied to $\boldsymbol{k}_f$ as well as $\dot{\boldsymbol{\varepsilon}}_V$: According to (13) the approximations $\boldsymbol{k}_f \approx U_1\boldsymbol{k}_f'$, $\dot{\boldsymbol{\varepsilon}}_V \approx U_2\dot{\boldsymbol{\varepsilon}}_V'$ with short coefficient vectors $\boldsymbol{k}_f'$, $\dot{\boldsymbol{\varepsilon}}_V'$ are used and hence

$$P_{p,h}(\boldsymbol{u}) \approx \boldsymbol{k}_{\boldsymbol{f}}'^{T} U_1^T U_2 \dot{\boldsymbol{\varepsilon}}_V'.$$

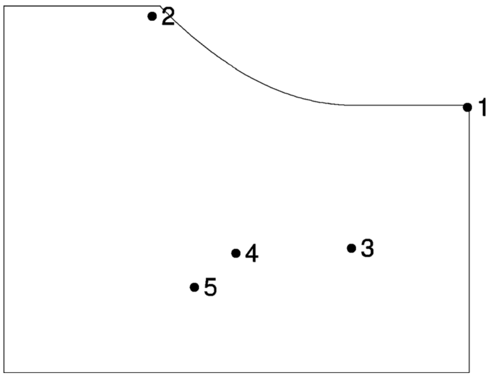

Figure 4. Points selected by the DEIM algorithm for strain calculation for acceleration of hardening.

A [MPa]	m_1 [–]	m_2 [–]	m_3 [–]	m_4 [–]
655.42	−0.00132	0.14496	0.01803	0.00013

Table 2. Used hardening parameters.

The matrix $U_1^T U_2$ can be calculated in the offline phase and then no longer depends on M. It is also possible to apply DEIM only to $\boldsymbol{k_f}$ as this has the strongest effect on computation time and to use (10) for the strain rate.

To illustrate the complexity of the calculation of $\boldsymbol{k'}_f$, note that using (14) the coefficient vector for the nonlinear term is calculated as

$$\boldsymbol{k'}_f = U_{1(\boldsymbol{i})}^{-1} \boldsymbol{k}_{f(\boldsymbol{i})}(X_{\boldsymbol{u}} \boldsymbol{u}_k).$$

This requires only the evaluation of m components of $\boldsymbol{k_f}$. Since each component represents the hardening in a quadrature point it is necessary to calculate k_f in the points characterized by $\boldsymbol{i}$ which is obtained from the DEIM algorithm. Figure 2 shows the determined trajectories for the calculation of ε_V for an unreduced problem, whereas **Figure 4** visualizes the five points selected by DEIM where the trajectories for the approximation of $\boldsymbol{k_f}$ are required.

For the calculation DC04 steel with the material parameters from **Table 2** and **Table 3** was used. The roll radius is 0.1275 m, the roll speed 0.11 m s^{-1}, the initial temperature 25°C, the friction coefficient m is 0.1, and no external forces are applied. For the finite element discretization $N = 4914$ degrees of freedom for the velocity were used. At the various roll passes, the height was reduced successively from 2.00 mm to 1.46 mm, and further 1.13, 0.74, and 0.54 mm. The results for this pass schedule can be found in **Table 4**. **Figure 5** compares the unreduced and reduced vertical stress component for $k = m = 5$ for the first roll pass.

5. Conclusions

We have demonstrated that model reduction using POD combined with DEIM is an effective tool to reduce the calculation time for the numerical simulation of cold rolling. Speedup factors of 30 and more were achieved and will even be greater if the mesh is chosen finer. In particular, the computation of the hardening process was accelerated. A disadvantage of the reduction is the costly setup phase. Moreover, the selection of system snapshots was not

ρ [kg m^{-3}]	c [J kg^{-1} K^{-1}]	κ [W m^{-1} K^{-1}]	h_{roll} [W m^{-2} K^{-1}]	h_{air} [W m^{-2} K^{-1}]
7861.2	466.43	65.08	6000	15

Table 3. Used thermal data.

Pass	Time [s]		$\frac{\lVert \boldsymbol{u} - \boldsymbol{u}_{\text{full}} \rVert}{\lVert \boldsymbol{u}_{\text{full}} \rVert}$	Power [W]	
	Full	Reduced		Full	Reduced
1	243.1	11.0	0.000535	17202.9	17198.7
2	242.6	8.4	0.000229	12151.9	12280.3
3	371.7	10.8	0.000416	30286.7	30165.6
4	259.6	8.7	0.000312	15814.1	15739.9
5	517.0	10.9	0.000671	16908.2	16828.6

Table 4. Calculation for different roll passes.

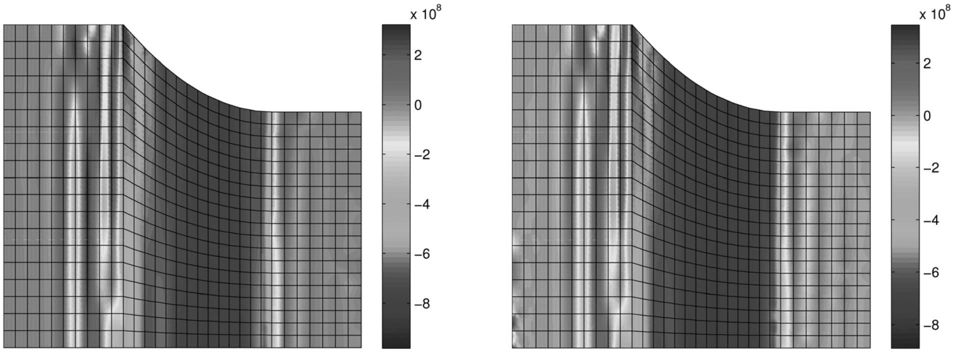

Figure 5. Stress component σ_{yy}, unreduced (left) and reduced (right).

discussed and is not yet satisfactorily automated. Nevertheless, the decay of the singular values provides a good indication of the quality of the approximation and, by adjusting the subspace sizes k and m, a simple control method exists to choose between fast and inaccurate solutions on the one hand and slower but more accurate solutions on the other. This is not easily possible with commonly used approaches based on simplifying the physical model.

Acknowledgments

The authors thank the German Research Foundation (DFG) for financial support through the grant ER 273/6-3 within the Priority Program SPP1204. The work was also funded by the European Social Fund (ESF) under grant number 995.319570.6.
This paper was amended in the Early View because there was a mistake in the publication.

Received: June 3, 2013;
Published online: April 6, 2014

Keywords: cold-rolling; finite element approximation; numerical simulation; model reduction

References

[1] M. Schmidtchen, R. Kawalla, *Steel Res. Int.* **2010**, *81*(*9*), 230.

[2] S. Kobayashi, G. J. Li, *ASME* **1982**, *104*, 55.

[3] J. Seidel, O. Ernst, M. Schmidtchen, B. Senger, R. Kawalla, M. Seefeldt, K. Decroos, Modellierung und Simulation der Werkstoffentwicklung innerhalb der Prozesskette Kaltwalzen von Bändern und Blechen unter dem Aspekt der Prozessoptimierung und online-Steuerung, in *MEFORM 2013*, Institut für Metallformung, TU Bergakademie Freiberg **2013**, p. 278.

[4] S. Kobayashi, S. Oh, T. Altan, *Metal Forming and the Finite-Element Method*, Oxford Science Publications, Oxford University Press, New York **1989**.

[5] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, *C. R. Acad. Sci. Paris Ser. I* **2004**, *339*, 667.

[6] K. Kunisch, S. Volkwein, *SIAM J. Numer. Anal.* **2002**, *40*(*2*), 492.

[7] S. Chaturantabut, D. C. Sorensen, *SIAM J. Sci. Comput.* **2010**, *32*(*5*), 2737.

[8] K. Morton, *Numerical Solution of Convection-Diffusion Problems*, Applied Mathematics and Mathematical Computation, Chapman & Hall, London **1996**.