Computational Mechanics (2019) 63:511–520
https://doi.org/10.1007/s00466-018-1606-5

ORIGINAL PAPER

Accelerating coupled finite element-kinetic Monte Carlo models: 200 × speedup of shear transformation zone dynamics simulations

Thomas J. Hardin[1]

Received: 27 January 2018 / Accepted: 3 July 2018 / Published online: 21 July 2018

Abstract
Shear transformation zone dynamics models of metallic glass deformation access experimentally-relevant time scales by using the kinetic Monte Carlo method to simulate small, fast, often discrete events, while the finite element method calculates macroscopic shape change and continuum-level interactions within samples. The most time-consuming portion of these models is the finite element method calculation on each step. However, in cases where the finite element mesh geometry and element elastic properties do not change from step to step, the finite element stiffness matrix (and its Cholesky factors) from previous steps can be reused. This strategy improves the asymptotic complexity of these models and in practice accelerates their execution by nearly 200 ×. This enables simulation of larger samples in more reasonable time. A set of three-dimensional shear transformation zone dynamics simulations, with larger length scales than any currently in the literature, illustrates the utility of this approach.

Keywords Finite element · Kinetics · Plasticity · Shear deformation · Shear band

1 Introduction

Specialized techniques are available for modeling physical phenomena at the extremes of the length and time scales. At the small/fast end of the spectrum, phenomena involving a few atoms and very fast time scales can often be reproduced using first principles techniques [1,2]. Ensembles of many more atoms can routinely be simulated on time scales below a millisecond using molecular dynamics [3–5]. At the opposite end of the spectrum, continuum models treat material as a continuous homogenized medium rather than as a granular assembly of atoms [6]; this assumption creates a lower limit on continuum theories' applicable length and time scales, though that limit shifts relative to acceptable error levels. Mesoscale, multiscale, and coupled multiphysics models have proliferated for studying phenomena between or spanning these length and timescale extremes. Mesoscale examples include dislocation dynamics [7–10], phase field models [11–13], and some kinetic Monte Carlo models [14–17].

This paper concerns a particular class of mesoscale model that uses kinetic Monte Carlo (kMC) to govern discrete, small-scale, relatively fast deformation events, and the Finite Element Method (FEM) to calculate the interactions between the discrete events and their continuum-level cumulative effect (i.e. a sample's macroscopic shape change). These models are cyclical: the FEM computes the sample's stress field and passes it to kMC; kMC uses that stress field to select a localized "transformation" (e.g. a shear event or phase transformation) which is passed back to the FEM; the FEM then applies that transformation as an eigenstrain [18] and calculates an updated stress field. Because they appeal to the raw deformation mechanism kinetics, these methods are able to capture much more granular detail than would a continuum constitutive law, while avoiding the many atomic vibrations that molecular dynamics so exhaustively simulates. Consequently, these methods have in common an exceptional compromise between simulation fidelity and size (spatially and especially temporally). Prototypical of this class of models is Homer's Shear transformation zone dynamics (STZD) model [19] for deformation of bulk metallic glasses (which will be outlined in the next subsection). Other closely

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00466-018-1606-5) contains supplementary material, which is available to authorized users.

✉ Thomas J. Hardin
thomasjhardin@gmail.com

[1] Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, USA

related models (cyclically coupling kinetics and the FEM) include a quantized crystal plasticity model for nanocrystalline materials [20–26] and a kMC model for martensitic phase transformations in shape memory alloys [27]. The Discrete Shear-Transformation-Zone Plasticity model [28,29] also models metallic glass deformation by cycling between kinetics and elasticity, but uses a hybrid of analytical and FEM calculations in its elastic portion.

The computational scaling of coupled kMC–FEM models is generally dominated by the continuum FEM calculation. The memory consumption and computational time required to evaluate an up-to-date stress field in each step has limited most instantiations of this class of models to two-dimensional approximations, and the few three-dimensional examples in literature (for example, [30,31]) invariably model very small samples (at most 60 nm in any direction).

To address this shortcoming of the above-described class of models, this manuscript borrows the well-established concept of stiffness matrix factor caching from closely related modeling techniques. Two particularly relevant examples of reuse of the stiffness matrix decomposition in mesoscale modeling are in discrete dislocation dynamics [32] and coupled atomistic/continuum multiscale models [33]. Despite the historical success of stiffness matrix factor caching in other mesoscale models, this strategy has never before been applied to STZD or it sibling models cited above.

This paper's Methods section describes stiffness matrix factor caching and shows how it accelerates these models. While these methods apply to the entire class of models described above, the data presented in Sect. 3 focus on STZD as a case study. In anticipation of this, the following subsection provides a brief review of the STZD model; the reader is referred to [34] for a more in-depth presentation. This paper concludes with a presentation of the largest-ever three-dimensional STZD simulation, which was executed using stiffness matrix factor caching, and which showed an acceleration of nearly 200 × over the original approach.

1.1 Introduction to the STZD model

The STZD model is based on Argon's theory of metallic glass deformation [35], which postulates shear transformations, groups of atoms collectively shearing, as the fundamental plastic event. This manuscript uses the kinetic model in [30] to calculate the activation rates for various possible shear transformation events; this model is in turn a simplification of Argon's treatment of shear transformations using Eshelby's solutions for ellipsoidal inclusions with stress-free strain [36]. STZD models a sample with a finite element mesh, where the mesh elements coarse-grain the sample's atoms. Clusters of elements (often sharing a common node) constitute potential shear transformation zones (STZ). The physical size of an STZ therefore bounds the maximum element size of the STZD method's FEM mesh; so the physical sample size that can be simulated by STZD is closely connected to the FEM mesh size that can be handled. That is, simply scaling the FEM mesh size is not an option for reaching longer length scales with STZD.

Each step of the STZD model begins with the sample's stress state, which is calculated by FEM, taking into account the sample's loading and preexisting eigenstrain. The activation rate for each STZ is then estimated using transition state theory [37], which predicts an Arrhenius-like relation [38]:

$$\dot{s} = \nu_0 \exp\left(-\frac{\Delta F}{k_B T}\right) \int_{g \in G} \exp\left(\frac{\tau(\sigma, g)\gamma_0 \Omega_0}{2 k_B T}\right) dg \qquad (1)$$

where ν_0 is the transition attempt frequency (on the order of the material's Debye frequency), ΔF is a fixed activation energy barrier, G is the set of combinations of shear plane and direction, τ is the shear stress resolved on $g \in G$, γ_0 is the characteristic STZ shear strain, and Ω_0 is STZ volume. The kMC algorithm then stochastically selects a single STZ shear event as the next transition and computes a time step (a "residence time" before the transition). The probability of choosing any STZ shear event is weighted proportionally to its particular rate. To close the cycle, the FEM applies the appropriate eigenstrain (also called "thermal strain" or "initial strain" in FEM literature) to the FEM mesh elements comprising the selected STZ, increments the sample's loading conditions, and computes the updated sample stress field. The STZD cycle then repeats. Gradually the individual STZ activation events cause eigenstrain to accumulate in the FE mesh, resulting in macroscopic plastic deformation of the sample.

The STZD model was originally implemented in two dimensions [19] and then extended to three dimensions [39]. It has been successfully used to simulate shear samples [19], tensile samples [39,40], and single and cyclic nanoindentation [39,41]. It has also been extended to include free volume as an evolving state variable [42,43] and to study metallic glass matrix composite materials [40]. The key papers reporting results from STZD simulations are shown in Table 1, along with the dimensionality and the length scales of those simulations. These papers have produced valuable insights into shear band nucleation and structure [31] and metallic glass deformation modes [30], among other phenomena, while relying mostly on two-dimensional approximations. The few three-dimensional samples in the literature never exceeded 60 nm in any direction and typically took weeks on multicore architectures to compute. Comparison to selected micromechanical experiments (cited in Table 2; see also [44]) shows a gap between experimentally-relevant length scales and simulation capabilities which prevents side-by-side comparison for calibration, validation, and forward-modeling purposes. This gap is of particular interest in view of the

Table 1 Sizes of STZD simulations in literature, with length scales and brief descriptions of the simulated loading

References	Description	Dimensions (nm)	Volume (nm^3)
Two-dimensional STZ dynamics simulations			
[19]	Shear	27.6×45.8	
[39]	Shear	34.8×57.7	
[41]	Nanoindentation	100×35	
[42]	Shear	60×120	
[43]	Tensile	(Not reported)	
[40]	Tensile	100×300	
[50]	Tensile	50×250	
Three-dimensional STZ dynamics simulations			
[30]	Tensile creep	$10 \times 10 \times 20$	1.57×10^3
	Nanoindentation	$30 \times 30 \times 11$	7.78×10^3
[31]	Tensile	$20 \times 20 \times 60$	1.88×10^4
*	Uniaxial	$60 \times 60 \times 170$	3.46×10^5

The largest simulations reported in this paper are denoted by an asterisk

Table 2 Selected micromechanical experiments on metallic glass from literature, with length scales and brief descriptions of loading

References	Metallic glass micromechanical experiments		
	Description	Dimensions (nm)	Volume (nm^3)
[46]	Pillar compression	$70 \times 70 \times 210$	8.08×10^5
[45]	Pillar compression	$90 \times 90 \times 360$	2.29×10^6
[47]	Pillar bending	$93 \times 93 \times 744$	5.05×10^6
[48]	Tensile	$100 \times 100 \times 650$	5.11×10^6
[49]	Tensile	$70 \times 70 \times 350$	1.35×10^6
[51]	Nanoindentation	Depth 50–100	

experimentally-observed transition in metallic glass plasticity between 80 and 500 nm-diameter uniaxially loaded samples [44–49]; the technique in this paper brings STZD much closer to being able to study this transition *in silico*.

2 Method

The method to follow uses the FEM in its constituent pieces rather than as a "black box." The reader is referred to the first two chapters of [52] for an in-depth introduction to the FEM, but a high-level overview is provided here for context. The FEM takes a discretized mesh of a sample and constructs interpolation functions ("shape functions") on the mesh elements. Then, under the postulate that (in the case of elasticity) the displacement field satisfying stress equilibrium can be approximated by a weighted sum of the shape functions, the FEM constructs a linear system:

$$\mathbf{Kd} = \mathbf{F} \tag{2}$$

where the unknown vector $\mathbf{d}$ consists of the shape function weights best satisfying the underlying differential equation. The symmetric positive definite matrix $\mathbf{K}$ is termed the "stiffness matrix," and is constructed from the elastic constants of the sample and the mesh shape functions. The vector $\mathbf{F}$ is termed the "force vector," and contains (in addition to the stiffness matrix's ingredients) information on both Dirichlet and Neumann boundary values, body forces, eigenstrains, and eigenstresses. Equation (2) is often solved by Cholesky decomposition [53–55] of the stiffness matrix $\mathbf{K} = \mathbf{LL}^T$, followed by solution of $\mathbf{LL}^T\mathbf{d} = \mathbf{F}$ by forward- and back-substitution.

The STZD algorithm can be framed as a cycle with six steps (as shown in Fig. 1a). After a brief setup phase, the stiffness matrix $\mathbf{K}$ is constructed using the sample mesh and elastic stiffness tensors, at a computational cost of $O(n)$ where n is the number of mesh nodes. Second, a sparse Cholesky algorithm factors $\mathbf{K} = \mathbf{LL}^T$; this step is the most computationally expensive, with theoretical $O(n^3)$ complexity, empirically closer to $O(n^2)$ when sparse linear algebra is leveraged. Third, the force vector $\mathbf{F}$ is constructed from the sample loading, body force, and preexisting eigenstrain fields, in $O(n)$ time. Fourth, the system $\mathbf{Kd} = \mathbf{F}$ is solved by forward- and back-substituion, with theoretical $O(n^2)$ complexity, empirically closer to $O(n)$ with sparse computations. Fifth, in $O(n)$ time the displacement field is postprocessed into stress and strain fields for the sample. Sixth and finally, kMC selects the next transition, also in $O(n)$ time. The asymptotic complexities of these steps are listed in Table 3.

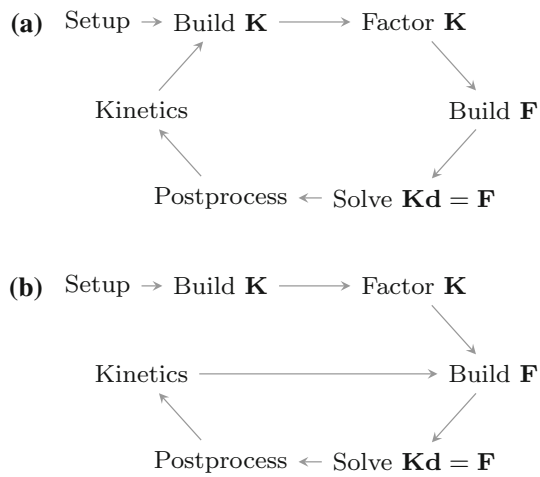

Fig. 1 **a** Kinetic-FEM cycle; **b** kinetic-FEM cycle with shortcut shown

Table 3 Complexity of pieces of the STZD method, assuming dense numerical linear algebra, where n is the number of nodes in the mesh

Step	Dense theoretical	Sparse empirical
Setup	$O(n)$	$O(n)$
Build $\mathbf{K}$	$O(n^2)$	$O(n)$
Factor $\mathbf{K} \rightarrow \mathbf{LL}^T$	$O(n^3)$	$O(n^2)$
Build $\mathbf{F}$	$O(n)$	$O(n)$
Solve $\mathbf{LL}^T\mathbf{d} = \mathbf{F}$	$O(n^2)$	$O(n)$
Postprocess	$O(n)$	$O(n)$
Kinetics	$O(n)$	n/a

The transition selected by kMC takes the form of an eigenstrain which is applied to a cluster of elements in the FEM mesh. Under the original algorithm the cycle then repeats itself, starting with construction of a new stiffness matrix. However, note that the new stiffness matrix will be identical to the previous one; modifying the boundary values and adding eigenstrain to the model changes neither the sample's elastic constants nor the shape functions. Therefore, construction and factorizaton of $\mathbf{K}$ is completely redundant after the first step.

This suggests a simple innovation: to calculate the FEM stiffness matrix and its factors once as a setup step, and then to cache those stiffness matrix factors in memory. This eliminates the necessity of calculating and factoring $\mathbf{K}$ in each simulation cycle; each cycle simply calculates the new force vector, solves the cached stiffness matrix factors against the new force vector, and then postprocesses the newly calculated displacement field to obtain strain and stress data (see Fig. 1b). This strategy is termed "stiffness matrix factor caching." It produces precisely the same results as the original approach; the physics are not altered, nor is the numerical approximation. This is simply an adjustment to the code's logical flow to eliminate redundant calculations.

The potential value of this optimization is apparent from the complexities of each part of the STZD algorithm in Table 3; by eliminating the need to factor a stiffness matrix with each step, the overall asymptotic complexity of each step is reduced from $O(n^3)$ to $O(n^2)$, assuming dense numerical linear algebra, and by a similar margin using sparse linear algebra. Of course this approach, while novel in the context of STZD modeling (and of the other closely related models mentioned in the introduction), is a straightforward application of well-established ideas within mesoscale modeling [32,33]; also, commercial FEM packages routinely reuse stiffness matrix factors in time-series calculations. More broadly, reusing matrix factors or inverses is a standard practice in algorithms for fields as diverse as optimization and image processing.

Since this approach consists only of modifying the logical flow of the STZD algorithm in a way that neither alters the physics nor the order of calculations, the results between the old and improved STZD algorithms are identical (down to and including floating-point error). The author has verified that this is the case for the STZD codes about to be described.

2.1 Implementation details

For this study two STZD codes were constructed, one of which follows the conventional algorithm in Fig. 1a and one of which leverages stiffness matrix factor caching as in Fig. 1b, but both of which are otherwise as similar as possible. Both codes are composed in C++11, and in lieu of a commercial FEM solver both codes use a simple in-house FEM library which takes advantage of the Eigen3 matrix library [56] and the Cholmod sparse linear system solver [57]. Both codes were compiled using the Intel compiler, linked against a single-threaded version of Intel's MKL library, and were run in serial on an Intel Xeon processor clocked at 2.6 GHz in a workstation with 128 Gb of memory. Both codes are instrumented to report the timing breakdown between parts of the STZD cycle to enable more granular comparison between the algorithms. The simulation input is in the form of an .ini file, and the output uses the HDF5 file format [58]. The code that does not use stiffness matrix caching performs similarly to commercial linear FEM implementations in serial execution mode.

3 Results

To examine the scaling of the STZD algorithm with respect to mesh size, wall-clock times were averaged over 10 STZD steps for meshes with between 1606 and 938,407 nodes; the timings are plotted on a log-log axis in Fig. 2. It is evident

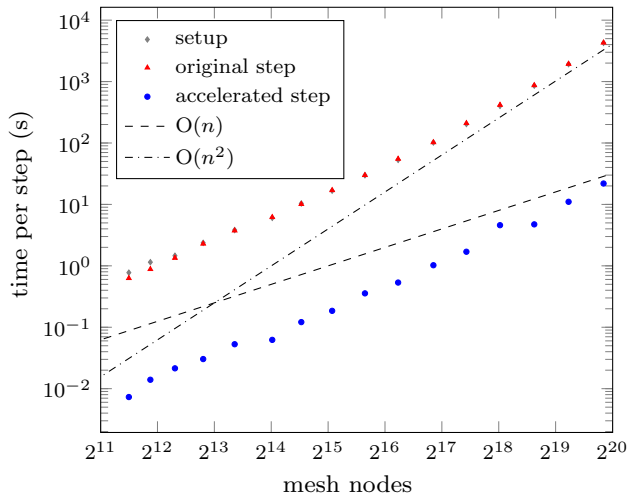

Fig. 2 Time required to execute STZD code described in the text, as a function of FEM mesh size

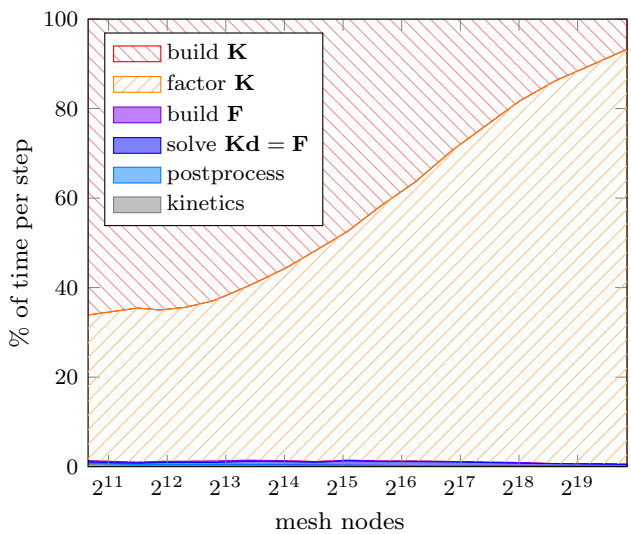

Fig. 3 Fraction of time spent on each part of a step of the STZD algorithm described in the text, as a function of FEM mesh size

that the STZD code using stiffness matrix factor caching is empirically faster than the original approach, with a speedup of 196 x for the largest meshes studied for this paper. The observed deviation from dense matrix asymptotic behavior is due to extensive use of sparse numerical linear algebra.

The effectiveness of caching stiffness matrix factors is further illustrated by fractionally breaking the execution time of an original STZD step into pieces in Fig. 3. The optimization described in this paper eliminates the striped regions of that plot (corresponding to building and factoring the FEM stiffness matrix), cutting 98–99.5% of the computation per STZD step.

To concretely illustrate the utility of this technique the next subsection contains a series of simulated uniaxial tensile and compression tests on the largest-ever STZD samples.

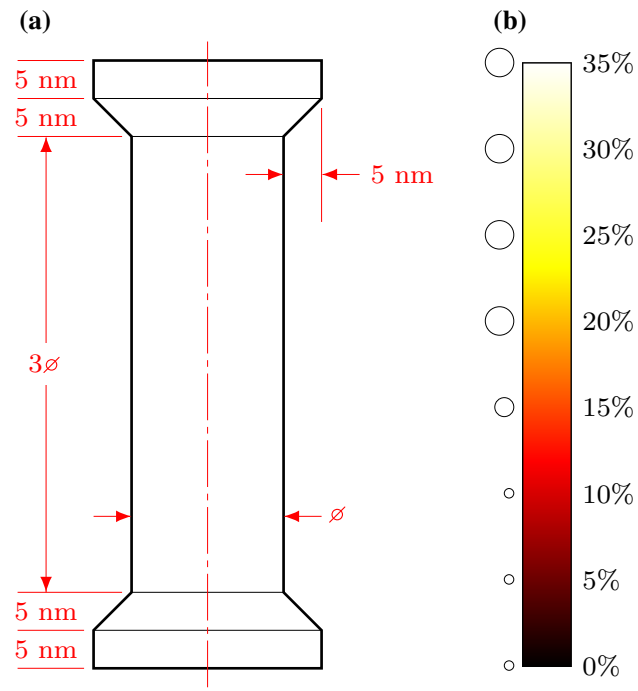

Fig. 4 Legend to the STZD simulation Figs. 6, 7, 8 and 9. Part **a** shows the dimensions of the uniaxial samples in terms of the parameter ⌀, the diameter of the gauge portion of the sample. Part **b** maps the norm of STZ strain to color and dot size. The dot sizes shown here are scaled much larger than those in the figures to follow, but are proportionally correct relative to each other

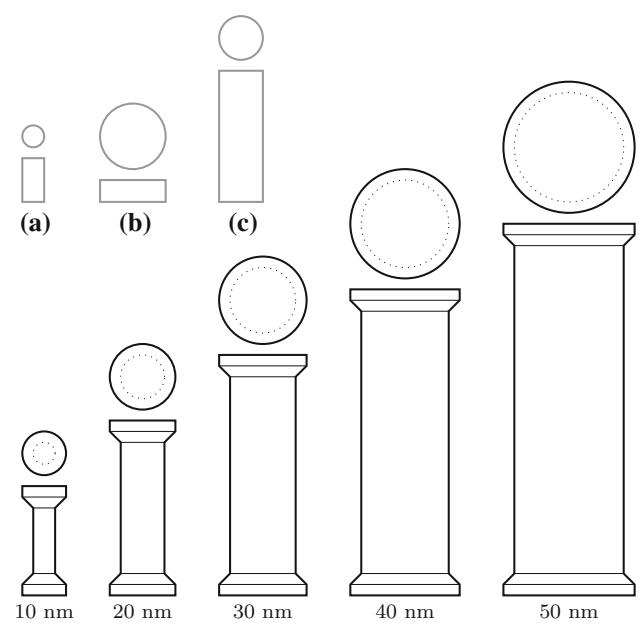

Fig. 5 Relative sizes of three-dimensional STZD simulations in literature and this paper. In the top-left corner are the three largest three-dimensional STZD simulations from literature, with **a** and **b** from [30], and **c** from [31]. Along the bottom are the various samples reported in this paper with their respective gauge section diameters (⌀)

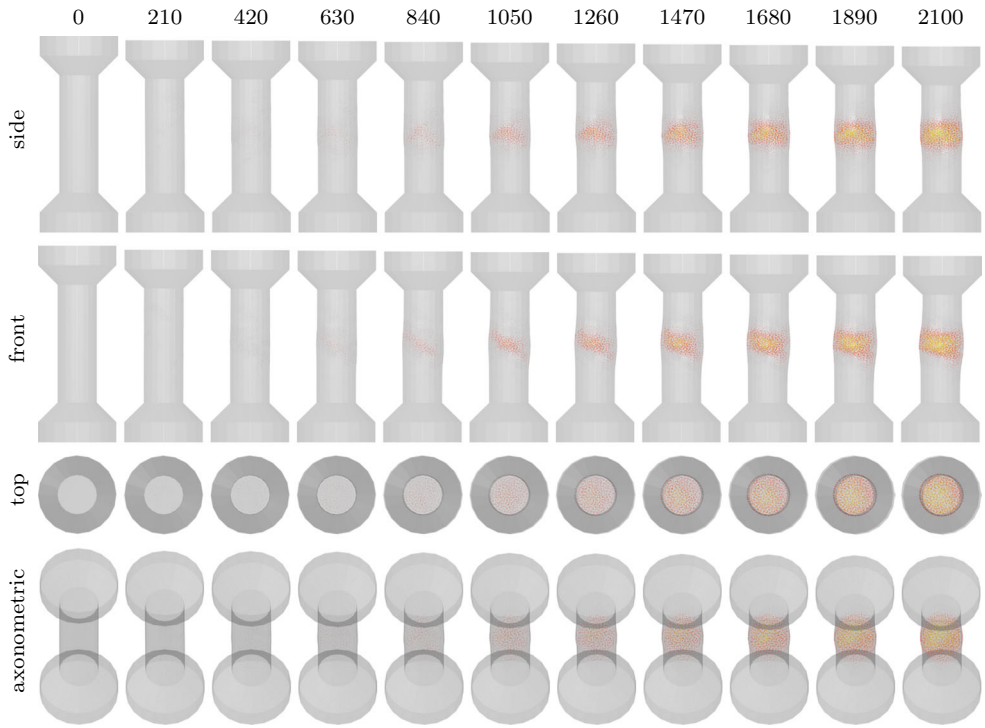

Fig. 6 STZD compression test of $\varnothing = 10$ nm sample. Numbers along the top are KMC steps. This simulation took less than 5 min to run on the machine described in the Sect. 2.1 of the text

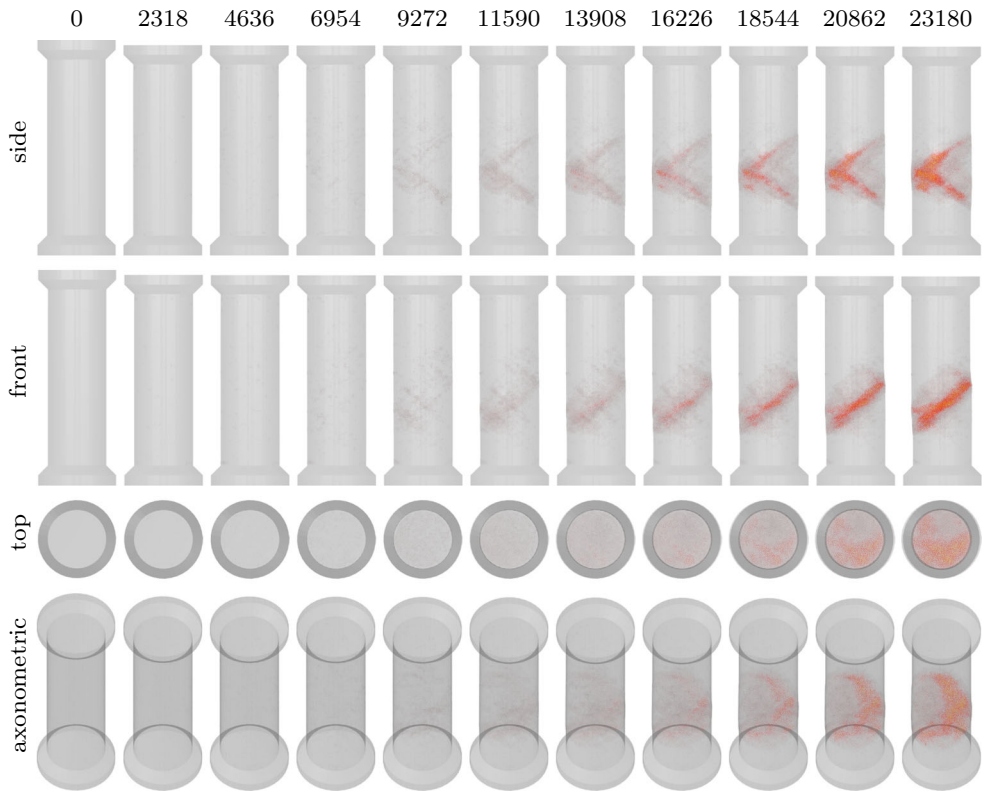

Fig. 7 STZD compression test of $\varnothing = 30$ nm sample. Numbers along the top are KMC steps. Execution time: 13 h

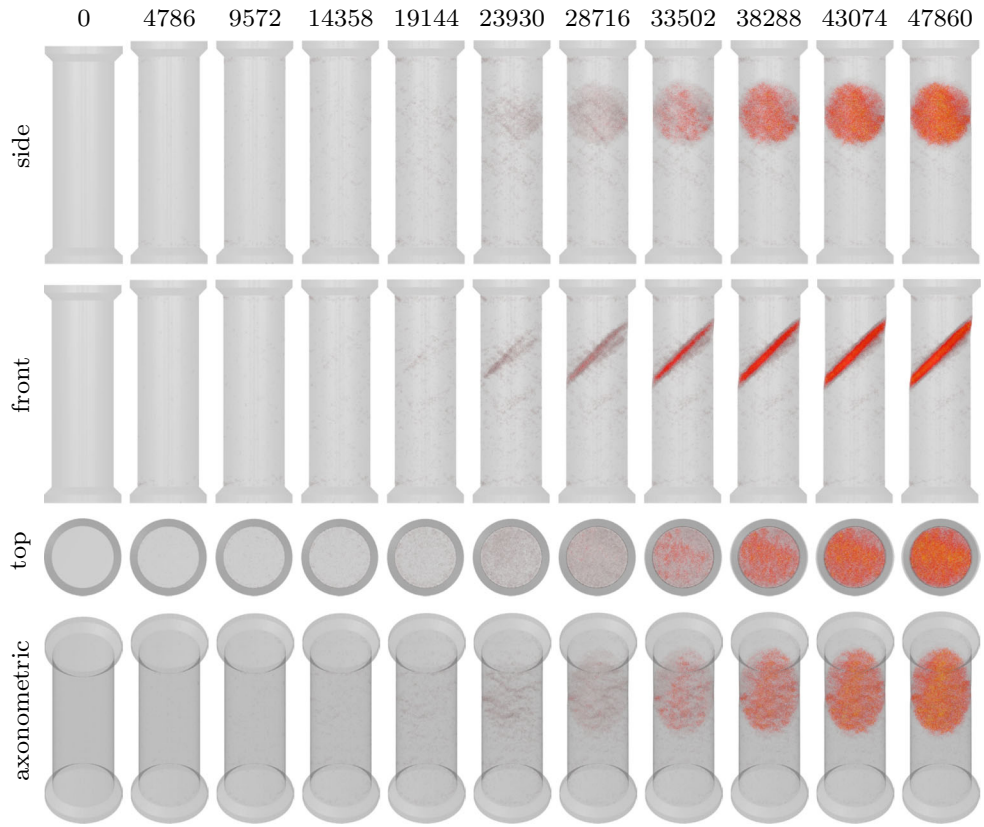

Fig. 8 STZD tension test of $\varnothing = 40$ nm sample. Numbers along the top are KMC steps. Execution time: 3 days. In the "front" view of steps 23,930 and 28,716, subtle wavelike variations in the density of shear transformations are evident near the plane of the nascent shear band

3.1 Uniaxial tensile and compression tests

This section describes uniaxial tests on nanoscale cylindrical samples with gauge diameters from 10 to 50 nm and gauge lengths from 30 to 150 nm; the geometry of these samples is drawn in Fig. 4a. The relative sizes of this paper's samples in comparison to three-dimensional STZD samples in literature are shown in Fig. 5. Each simulation was run for a number of steps proportional to the volume of the sample, to ensure roughly equal amounts of plastic deformation between the simulations. These simulations were run under the conditions described in Sect. 2.1 above; in particular, they were run in serial fashion. A selection of the simulations are plotted in Figs. 6, 7, 8 and 9, and the remainder are included in supplementary material to this article; in these plots, STZs are plotted as small dots, with the size and color of the dot corresponding to the norm of the cumulative STZ strain. The colorbar for STZs is shown in Fig. 4b. The STZD parameters for all the simulations are given in Table 4.

The most notable feature of the $\varnothing = 10$ nm compression sample in Fig. 6 is its runtime of less than 5 min. This is a dramatic improvement on the original approach, which would have taken at least a day to run a comparable simulation on multiple cores. This suggests that stiffness matrix caching will enable simulation of large ensembles of small samples for statistical analysis; this is of particular value because these simulations are stochastic in nature, so analysis of any one simulation might not be representative of the ensemble.

Of course, stiffness matrix factor caching could also enable use of a finer FEM mesh on these small samples. This has been shown to not be a particular issue in STZ Dynamics (assuming that the mesh size is an appropriate fraction of the material's characteristic STZ volume, as is the case in these simulations), but may be useful as kMC–FEM models are extended to new materials systems in the future.

Moving up to the $\varnothing = 30$ nm tensile sample in Fig. 7, which is already larger than any previously published STZD sample, nucleation of orthogonal competing shear bands is observed. The interaction between the shear bands apparently obstructs both of them from crossing the full diameter of the gauge section. This behavior has implications for understanding shear band nucleation and growth, and can only be observed in samples large enough to sustain multiple instances of shear localization. This issue will be thoroughly explored in future articles.

Looking closely at the $\varnothing = 40$ nm tensile sample in Fig. 8, one can observe periodic "waves" in the STZ strain field perpendicular to and along the length of the main shear band.

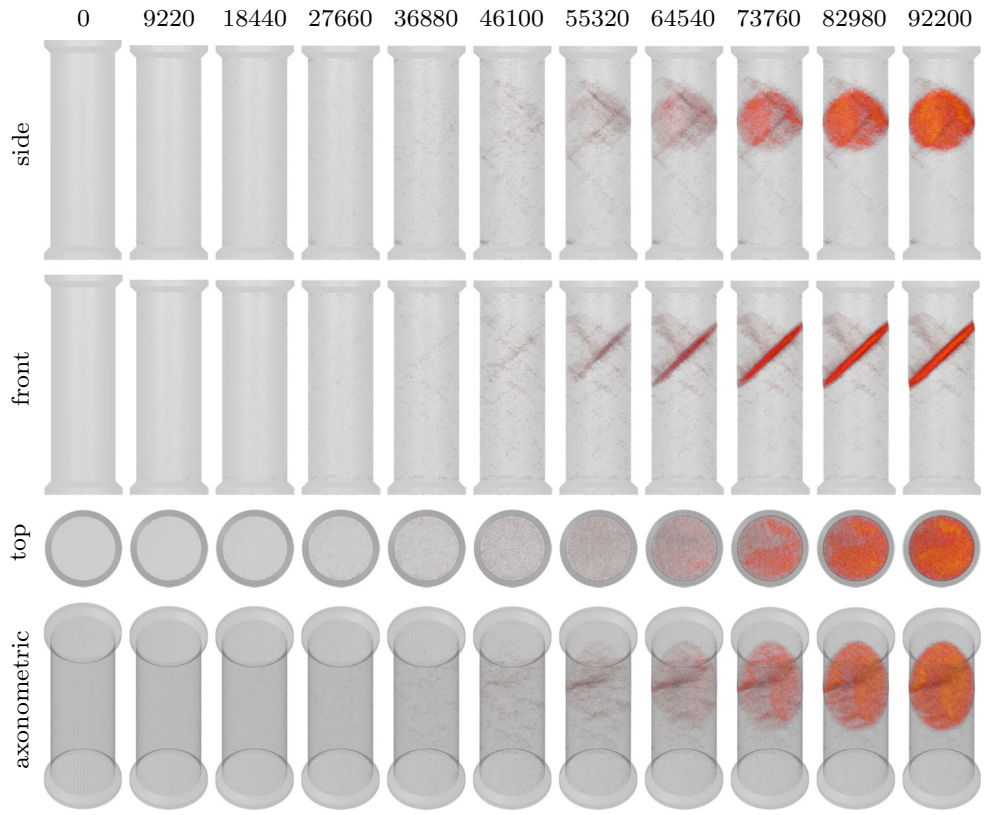

Fig. 9 STZD compression test of $\varnothing = 50$ nm sample. Numbers along the top are KMC steps. Execution time: 19 days

Interestingly, these appear very early in the shear band nucleation process (they are visible as early as step 23930 of the simulation). The wavelength of these oscillations (between 10 and 15 nm) is such that they would be impossible to observe in the smaller STZD samples published to date.

The $\varnothing = 50$ nm compression sample in Fig. 9 shows nucleation of four shear bands along orthogonal planes, but one of the shear bands ultimately dominates the others and propagates across the diameter of the sample (unlike Fig. 7). It is, however, apparent from the axonometric view that the dominant shear band is impeded by its orthogonal competitors. Again, a smaller STZD sample would be unable to sustain these multiple shear localization instances. This sample and the $\varnothing = 50$ nm tension sample in the supplementary material are the largest STZD simulations run to date, exceeding the volume of the previous maximum by a factor of eighteen; these simulations' dimensions approach the scale of physical nanomechanical experiments currently in the literature (as in Table 2).

It is worth noting here that stiffness matrix factor caching does not negate the necessity of remeshing when plastic deformation to the sample invalidates the linear expansion underpinning the FEM. The FEM stiffness matrix will need to be reconstructed and factored after each remeshing. However, in the context of STZD, remeshing events should be spaced many STZD steps apart, so the speed gains described above remain representative of expected performance even with remeshing.

Table 4 Parameters to all of the STZD simulations appearing in this paper

Parameter	Value	Units
ν_0	6.814×10^{12}	s^{-1}
ΔF	2.5×10^{-19}	J
Γ_0	0.1	m/m
T	300	K
G	37	GPa
ν	0.352	
$\dot{\epsilon}_{zz}$	± 1.0	m/(ms)
$\delta t_{\max}$	0.001	s

As mentioned in the implementation details above, the results here are for a serial implementation of STZ Dynamics. Stiffness matrix factor caching is moderately amenable to parallel implementation; there are libraries available with parallel sparse $\mathbf{LDL}^T$ decomposition and solution from factors (e.g. [59]) but increasing inter-process communication would be expected to cause diminishing returns as additional processes are added.

4 Conclusions

Simulation methods that iteratively link kinetics with localized updates in the FEM, typified by the STZ Dynamics method, have suffered from long run times due to superlinear scaling of the FEM with mesh size. However, if these methods do not require modifications to the mesh or the elastic properties of the sample from step to step, as is the case in STZD, then the FEM stiffness matrix is also unchanged from step to step. This enables an acceleration strategy: to calculate the FEM stiffness matrix, factor it, and cache the factorization in an initialization step, and then to use and reuse the factorization in each step. This is termed "stiffness matrix factor caching." While reuse of stiffness matrix factors is a common practice in mesoscale modeling, it has never before been applied to STZD and closely-related methods. Stiffness matrix caching constitutes an asymptotic improvement and empirically has produced a speedup of nearly 200x over the original method. This speedup is useful in two respects: it enables simulation of large numbers of small samples to form an ensemble, and it enables simulation of samples on experimentally-relevant length scales in three dimensions. These simulations of larger samples exhibit multiple (often competing) shear bands, sometimes in apparently periodic arrangements. Future work looks to directly compare real nanomechanical experiments to these large STZD simulations for validation purposes, or to illuminate avenues for improvement of the STZD model's physics. One such improvement might be inclusion of dilatative strain in shear transformations (as in [42]); so long as the elastic stiffness remains static from step to step, the speedup reported here holds. The results in this paper are readily extensible to STZD-like methods in both two and three dimensions; it is hoped that stiffness matrix caching will make three dimensional simulation the norm rather than the exception for STZD and its sibling methods, and that studies comparing these simulations to physical nanomechanical experiments will be forthcoming.

Acknowledgements Thanks to Prof. Chris Schuh of MIT DMSE for his advice on this work.

Funding Funded by a National Defense Science & Engineering Graduate Fellowship.

References

1. Kohn W, Sham LJ (1965) Self-consistent equations including exchange and correlation effects. Phys Rev 140(4A):A1133
2. Koch W, Holthausen MC (2015) A chemist's guide to density functional theory. Wiley, Hoboken
3. Fermi I, Pasta P, Ulam S, Tsingou M (1955) Studies of the nonlinear problems. Los Alamos Scientific Lab., N. Mex., Tech. Rep
4. Rapaport DC, Blumberg RL, McKay SR, Christian W et al (1996) The art of molecular dynamics simulation. Comput Phys 10(5):456–456
5. Plimpton S (1995) Fast parallel algorithms for short-range molecular dynamics. J Comput Phys 117(1):1–19
6. Gurtin ME (1982) An introduction to continuum mechanics, vol 158. Academic Press, Cambridge
7. Amodeo RJ, Ghoniem NM (1990) Dislocation dynamics. i. a proposed methodology for deformation micromechanics. Phys Rev B 41(10):6958
8. Gulluoglu A, Srolovitz DJ, LeSar R, Lomdahl P (1989) Dislocation distributions in two dimensions. Scr Metall 23(8):1347–1352
9. Kubin LP, Canova G, Condat M, Devincre B, Pontikis V, Bréchet Y (1992) Dislocation microstructures and plastic flow: a 3d simulation. In: Solid State Phenomena, vol 23. Trans Tech Publ, pp 455–472
10. Sills RB, Kuykendall WP, Aghaei A, Cai W (2016) Fundamentals of dislocation dynamics simulations. Multiscale materials modeling for nanomechanics. Springer, Berlin, pp 53–87
11. Fix G (1983) Free boundary problems. In: Fasano A, Primicero M (eds) Research notes in mathematics, vol 2. Pitman, New York, pp 580
12. Chen L-Q (2002) Phase-field models for microstructure evolution. Annu Rev Mater Res 32(1):113–140
13. Moelans N, Blanpain B, Wollants P (2008) An introduction to phase-field modeling of microstructure evolution. Calphad 32(2):268–294
14. Young W, Elcock E (1966) Monte carlo studies of vacancy migration in binary ordered alloys: I. Proc Phys Soc 89(3):735
15. Bortz AB, Kalos MH, Lebowitz JL (1975) A new algorithm for monte carlo simulation of ising spin systems. J Comput Phys 17(1):10–18
16. Gillespie DT (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 22(4):403–434
17. Voter AF (2007) Introduction to the kinetic monte carlo method. Radiation effects in solids. Springer, Berlin, pp 1–23
18. Mura T (2013) Micromechanics of defects in solids. Springer, Berlin
19. Homer ER, Schuh CA (2009) Mesoscale modeling of amorphous metals by shear transformation zone dynamics. Acta Mater 57(9):2823–2833
20. Li L, Anderson PM, Lee M-G, Bitzek E, Derlet P, Van Swygenhoven H (2009) The stress-strain response of nanocrystalline metals: a quantized crystal plasticity approach. Acta Mater 57(3):812–822
21. Li L, Lee M-G, Anderson PM, Barlat F, Moon Y, Lee M (2010) A quantized crystal plasticity finite element model for nanocrystalline metals: connecting atomistic simulations and experiments. In: AIP Conference Proceedings, vol 1252, no 1. AIP, pp 841–841
22. Li L, Lee M-G, Anderson PM (2011) Critical strengths for slip events in nanocrystalline metals: predictions of quantized crystal plasticity simulations. Metall Mater Trans A 42(13):3875–3882
23. Li L, Lee M-G, Anderson PM (2012) Probing the relation between dislocation substructure and indentation characteristics using quantized crystal plasticity. J Appl Mech 79(3):031009
24. Van Petegem S, Li L, Anderson P, Van Swygenhoven H (2013) Deformation mechanisms in nanocrystalline metals: insights from in-situ diffraction and crystal plasticity modelling. Thin Solid Films 530:20–24
25. Yuan R, Beyerlein IJ, Zhou C (2015) Emergence of grain-size effects in nanocrystalline metals from statistical activation of discrete dislocation sources. Acta Mater 90:169–181
26. Li L, Anderson PM (2016) Quantized crystal plasticity modeling of nanocrystalline metals. Multiscale materials modeling for nanomechanics. Springer, Berlin, pp 413–440

27. Chen Y, Schuh CA (2015) A coupled kinetic monte carlo-finite element mesoscale model for thermoelastic martensitic phase transformations in shape memory alloys. Acta Mater 83:431–447
28. Kondori B, Benzerga AA, Needleman A (2016) Discrete shear transformation zone plasticity. Extrem Mech Lett 9:21–29
29. Kondori B, Benzerga AA, Needleman A (2018) Discrete shear-transformation-zone plasticity modeling of notched bars. J Mech Phys Solids 111:18–42
30. Homer ER, Schuh CA (2010) Three-dimensional shear transformation zone dynamics model for amorphous metals. Model Simul Mater Sci Eng 18(6):065009
31. Homer ER (2014) Examining the initial stages of shear localization in amorphous metals. Acta Mater 63:44–53
32. O'day M, Curtin WA (2004) A superposition framework for discrete dislocation plasticity. Trans ASME-E-J Appl Mech 71(6):805–815
33. Shilkrot L, Curtin WA, Miller RE (2002) A coupled atomistic/continuum model of defects in solids. J Mech Phys Solids 50(10):2085–2106
34. Homer ER, Li L, Schuh CA (2016) Kinetic monte carlo modeling of nanomechanics in amorphous systems. Multiscale materials modeling for nanomechanics. Springer, Berlin, pp 441–468
35. Argon A (1979) Plastic deformation in metallic glasses. Acta Metall 27(1):47–58
36. Eshelby JD (1957) The determination of the elastic field of an ellipsoidal inclusion, and related problems. Proc R Soc Lond A 241(1226):376–396
37. Eyring H (1935) The activated complex in chemical reactions. J Chem Phys 3(2):107–115
38. Arrhenius S (1889) Über die reaktionsgeschwindigkeit bei der inversion von rohrzucker durch säuren. Zeitschrift für physikalische Chemie 4(1):226–248
39. Homer ER, Rodney D, Schuh CA (2010) Kinetic monte carlo study of activated states and correlated shear-transformation-zone activity during the deformation of an amorphous metal. Phys Rev B 81(6):064204
40. Hardin TJ, Homer ER (2015) Microstructural factors of strain delocalization in model metallic glass matrix composites. Acta Mater 83:203–215
41. Packard C, Homer E, Al-Aqeeli N, Schuh C (2010) Cyclic hardening of metallic glasses under hertzian contacts: experiments and stz dynamics simulations. Philos Mag 90(10):1373–1390
42. Li L, Homer E, Schuh C (2013) Shear transformation zone dynamics model for metallic glasses incorporating free volume as a state variable. Acta Mater 61(9):3347–3359
43. Li L, Wang N, Yan F (2014) Transient response in metallic glass deformation: a study based on shear transformation zone dynamics simulations. Scr Mater 80:25–28
44. Tian L, Wang X-L, Shan Z-W (2016) Mechanical behavior of micronanoscaled metallic glasses. Mater Res Lett 4(2):63–74
45. Kuzmin O, Pei Y, De Hosson JTM (2011) In situ compression study of taper-free metallic glass nanopillars. Appl Phys Lett 98(23):233104
46. Chen CQ, Pei YT, Kuzmin O, Zhang ZF, Ma E, De Hosson JTM (2011) Intrinsic size effects in the mechanical response of taper-free nanopillars of metallic glass. Phys Rev B 83(18):180201
47. Chen C, Pei Y, De Hosson JTM (2010) Effects of size on the mechanical response of metallic glasses investigated through in situ tem bending and compression experiments. Acta Mater 58(1):189–200
48. Chen D, Jang D, Guan K, An Q, Goddard W III, Greer J (2013) Nanometallic glasses: size reduction brings ductility, surface state drives its extent. Nano Lett 13(9):4462–4468
49. Tian L, Shan Z-W, Ma E (2013) Ductile necking behavior of nanoscale metallic glasses under uniaxial tension at room temperature. Acta Mater 61(13):4823–4830
50. Harris MB, Watts LS, Homer ER (2016) Competition between shear band nucleation and propagation across rate-dependent flow transitions in a model metallic glass. Acta Mater 111:273–282
51. Schuh C, Nieh T (2003) A nanoindentation study of serrated flow in bulk metallic glasses. Acta Mater 51(1):87–99
52. Hughes TJ (2012) The finite element method: linear static and dynamic finite element analysis. Courier Corporation, NewYork
53. Benoıt C (1924) Note sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrésa un systeme d'équations linéaires en nombre inférieura celui des inconnues. application de la méthodea la résolution d'un systeme défini d'équations linéaires (procédé du commandant cholesky). Bulletin géodésique 2(1):67–77
54. Fox L, Huskey HD, Wilkinson JH (1948) Notes on the solution of algebraic linear simultaneous equations. Q J Mech Appl Math 1(1):149–173
55. Cholesky A-L (2005) Sur la résolution numérique des systèmes d'équations linéaires. Bulletin de la Sabix. Société des amis de la Bibliothèque et de l'Histoire de l'École polytechnique, 39:81–95
56. Guennebaud G, Jacob B, et al. (2010) Eigen v3. http://eigen.tuxfamily.org. Accessed 13 Aug 2017
57. Chen Y, Davis TA, Hager WW, Rajamanickam S (2008) Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans Math Softw (TOMS) 35(3):22
58. The HDF Group. (2000-2017) Hierarchical data format version 5. [Online]. http://www.hdfgroup.org/HDF5. Accessed 13 Sept 2017
59. Amestoy PR, Duff IS, Koster J, L'Excellent J-Y (2001) A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J Matrix Anal Appl 23(1):15–41