Proceedings of the 2014 ASME Manufacturing Science and Engineering Conference
MSEC
June 9 - 13, 2014, Detroit, Michigan, USA

MSEC2014-3984

TOWARDS COMPUTATIONALLY TRACTABLE SIMULATIONS OF METAL FORMING PROCESSES WITH EVOLVING MICROSTRUCTURES

Marko Knezevic
Department of Mechanical Engineering, University of New Hampshire
Durham, NH, USA

Daniel J. Savage
Department of Mechanical Engineering, University of New Hampshire
Durham, NH, USA

Nicholas W. Landry
Department of Mechanical Engineering, University of New Hampshire
Durham, NH, USA

ABSTRACT

Performing microstructure sensitive metal-forming simulations is widely recognized as a computational challenge because of the need to store large sets of state variables related to microstructure data. We present a rigorous methodology for the compaction of microstructural data associated with a material point and show that the statistical distributions of microstructure of any size can be compacted to several hundred grains. The methodology is based on the spectral representation of microstructure distribution functions through the use of generalizes spherical harmonics. Subsequently, we present a computational framework aimed at dramatically reducing time needed for microstructure sensitive simulations of metal forming processes. The framework is based on a combination of the recently developed numerical implementations of crystal plasticity models in the spectral representation for obtaining the response of single crystals and specialized computer hardware that integrates a graphics-processing unit. We apply these two methodologies on a plane strain compression case study and obtain speedup factors exceeding three orders of magnitude.

INTRODUCTION

Crystal plasticity (CP) constitutive laws are highly desirable for performing accurate simulations of metal forming processes because they are based on crystallographic deformation mechanisms and account for evolution of the microstructure and crystallographic texture during deformation. These constitutive laws are important for understanding evolution of the underlying microstructure and associated effects on the anisotropic stress-strain response [1]. The single-crystal itself is anisotropic and a non-random distribution of single-crystals (i.e. crystallographic texture) leads to anisotropy. Therefore, understanding and modeling deformation should be based on the distribution of grain orientations and micro-scale deformation mechanisms associated with the crystal structure. Several approaches have been developed to link the grain level mechanical response to the response of a polycrystalline aggregate including self-consistent [2], CP finite-element (FE) models [1], and Taylor-type models [1].

A number of CP models have been integrated into FE simulation tools and successfully applied to simple compression and tension tests, bending, cup-drawing, sheet hydroforming and bulk forming [3,4]. However, these models have not been widely adopted by metal forming community because of prohibitive computational efforts and time involved in process simulations. For example, the computational time involved in simulating a simple compression up to a strain of 0.2 with about 1000 elements and 1000 grains at an integration point is about 60h on a regular PC [3]. Clearly significant speedups are required to render metal forming simulations involving crystal plasticity constructive laws practical.

Several strategies have been explored to speed up the crystal plasticity calculations. Database approaches that stores the main characteristics of the crystal plasticity solutions in the form of generalized spherical harmonics coefficients are described in [5]. A process plane concept, based on proper orthogonal decomposition in Rodrigues–Frank space, has been presented in [6]. Other attempts to improve efficiency of the CP simulations rely on the adaptive sampling algorithms and building a database that constantly updates itself [7]. Here, the database of CP responses was populated as a function of process parameters over which the responses can be interpolated. The adaptive sampling framework is not suitable in the case of anisotropic materials that show significant changes in the microstructure during deformation and/or frequent strain path changes.

In past several years we have invested significant efforts in developing efficient numerical schemes and successfully developed a databases containing discrete Fourier transforms of the crystal plasticity solutions [8, 9]. The work showed that CP theories can be formulated using a spectral representation that allows for up to two orders of magnitude acceleration of relevant simulations.

The spectral methods have been used successfully in many other fields of science and engineering with dramatic gains in computational efficiency including representation of texture [10] and material properties [11-13].

In this paper, we utilize the spectral representation to reduce large data sets of microstructural material data in the form of statistical distributions to computationally manageable but representative statistical distributions. We show that a set of microstructural data of any size can be reduced to a few hundred grains and accurately represent the material response. We then present a high-performance implementation of the spectral crystal plasticity formulation on a computational platform integrating one graphic-processing unit (GPU). The new implementation takes the advantage of an efficient GPU8 algorithm for matrix-matrix multiplication [14] and results in major improvements in computational speed exceeding three orders of magnitude over the conventional numerical schemes. The framework presented here obtains the response of a polycrystalline aggregate associated with a single finite element (FE) integration point.

COMPACT REPRESENTATION OF CRYSTALLOGRAPHIC TEXTURE

In our approach a measured distribution functions of the crystallographic orientations, *f(g)*, is represented by Fourier coefficients using generalized spherical harmonics [15].

$$f(g)=F_l^{\mu\nu}\,\dot{\ddot{T}}_l^{\mu\nu}(g) \tag{1}$$

In general, any texture function is a point in the multidimensional Fourier space, defined by the Fourier coefficients [16, 17]. We will call this point as the target point. The target point is obtained by averaging the Fourier coefficients of full set of crystal orientations. The number of dimensions depend on the value l and therefor the accuracy of representation. It has been determined that when l = 16, the representation is indistinguishable from an exact value of the function. Figure 1 shows the accuracy of the representation with $l < 16$ [15]. We defined a texture difference index (TDI) as the normalized distance from the target to a point of reconstructed point. In our approach, we recognized that any target point can be reconstructed in the Fourier space by a significantly smaller set of weighted crystallographic orientations. To this end, we choose several sets of orientations in a way that they cover entire Euler space of orientations and developed a fitting scheme. The sets consisted of 153, 400, 825, and 1476 orientations. The fitting scheme is a linear programing code for fitting a point in a multi-dimensional space. The target point as well as the reconstructed point were calculated for l = 6, 8, 10, 12, and 16 (see Fig. 1). The corresponding number of dimensions in the Fourier space was 7, 12, 22, 36, and 75. To balance accuracy of the reconstruction and the computational efficiency of our fitting code, in this work we chose l = 10 (22 dimensions). Figure 1 shows that the level of accuracy of the reconstructed texture function relative to the target point for l = 10 is about 5%. Fitting a point in the multi-dimensional space corresponding to the l = 16 level of accuracy was computationally very intensive.

We seek for the minimum number of crystallographic orientations necessary for accurate modeling of the mechanical material response. Figure 2 compares the predicted mechanical response in the form of stress-strain curves between original full data set and the reduced data set using 400 and 825 orientations. Figure 2 shows the pole figures indicating that the representation the full set using 153 orientations is not accurate but it is accurate using 400 orientations. Therefore, we conclude that 400 weighted orientations associated with each material point are sufficient to represent any texture.

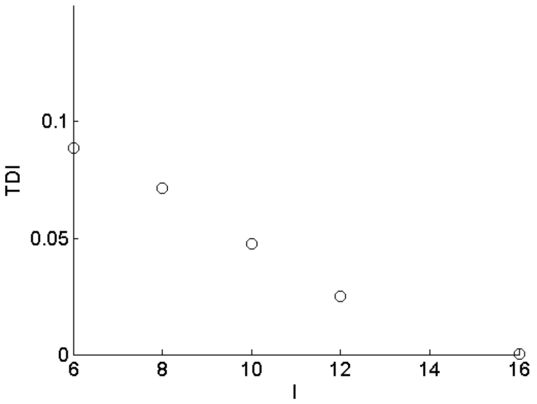

Figure 1 Accuracy of texture representation as a function of number of dimensions.

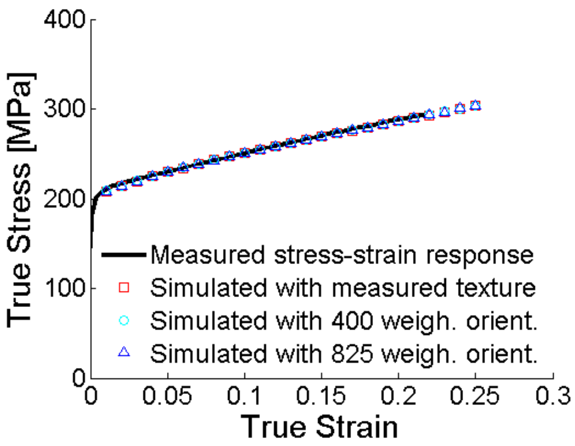

Figure 2 Comparison of the stress-strain response computed using the full data set and two reduced data sets. Also, the measured curve in shown for Cu.

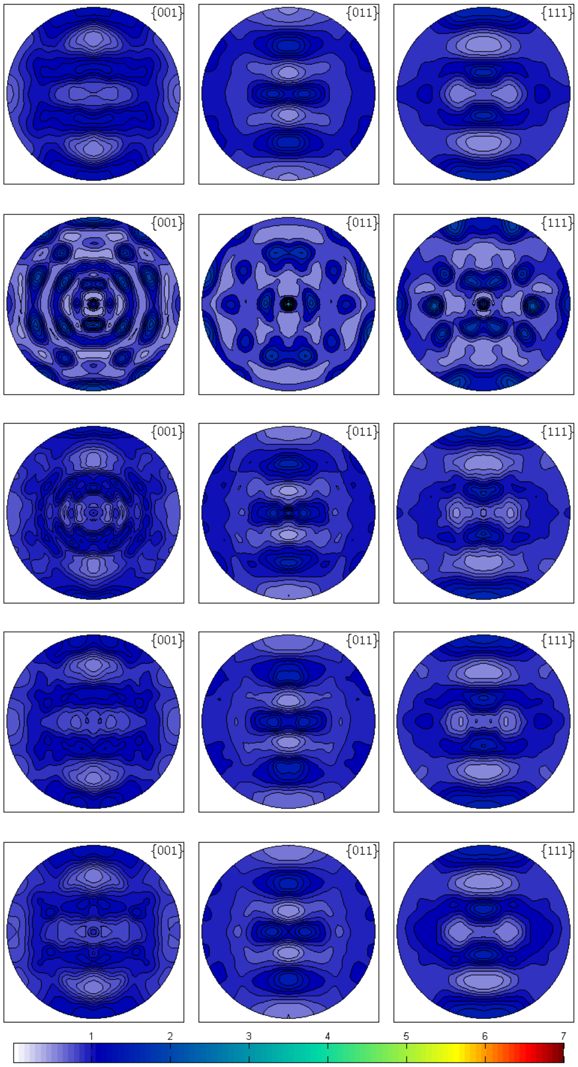

Figure 3 Pole figures for the full set of texture data (top row) and reduced data sets corresponding to the following number of orientations 153, 400, 825, 1476, respectively.

A HYBRID CPU-GPU HIGH-PERFORMANCE COMPUTATIONAL PLATFORM FOR CRYSTAL PLASTICITY

Figure 4 shows a work distribution on a hybrid CPU-GPU computer architecture intended for performing future tractable multi-scale process simulations [18]. The architecture is a computer cluster consisting of Linux nodes that individually integrate multi-core processors (CPUs) and Graphics Processing Units (GPUs). The workload can be distributed such that the grain level calculations are performed on the GPU threads. The motivation for using a GPU card comes from the fact that the card can have thousands of cores and teraflop performance. Next, the meso-scale polycrystalline homogenizations are performed on the CPUs of individual slave Linux nodes. Note that mesh of a finite element (FE) model can be divided into a number of mesh domains. The slave Linux nodes operate on these mesh domains and are controlled by the master Linux node. Finally, the master Linux node obtains the macroscopic component level response. Although parallelization with only CPUs is possible, this would require a significantly more expensive computer cluster as opposed to a simple and cheap computer workstation with GPUs.

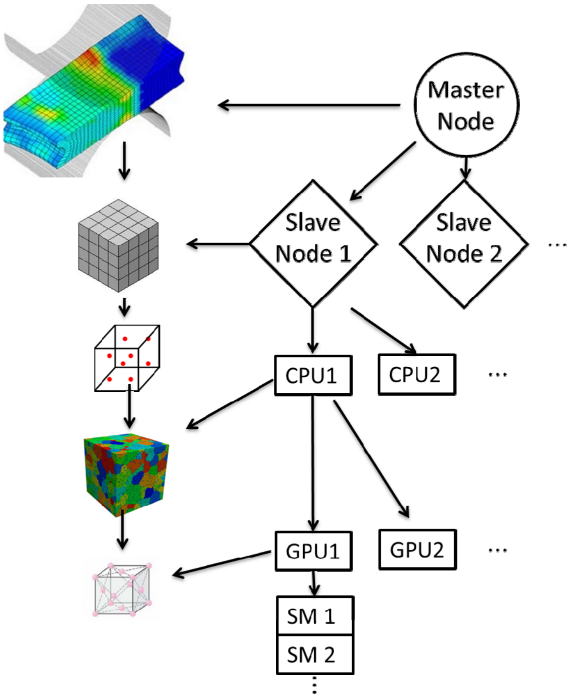

Figure 4 Work distributions on a hybrid CPU-GPU computer architecture aimed at performing tractable multi-scale process simulations. Domains of an FE model are distributed over the number of available nodes in the computer hardware. The response of a polycrystalline aggregates is obtained using multiple processors (CPUs) available per node. The grain-scale response is calculated on the Graphics Processing Unit (GPU) cards consisted of streaming multiprocessors (SMs) containing threads.

COMPUTER IMPLEMENTATIONS OF SPECTRAL CRYSTAL PLASTICITY ON A GPU CARD

Graphical processing units (GPUs) were originally developed to meet the computational needs of algorithms for rendering computer graphics. The GPU technology has rapidly grown resulting in teraflop peak performance and at a cost of a few thousand dollars. A GPU uses a master-slave paradigm in which the GPU operates as a slave under the control of a host CPU processor. In our implementation the host processor is a Linux node that features an Intel Xeon CPU X5660 at 2.67GHz and the slave is an NVIDIA Tesla K20 GPU card with 6GB of memory and 2496 CUDA cores. The Tesla GPU card has a peak performance of 3.52 teraflops in single-precision floating-point operations and 1.17 teraflops in double-precision operations. The board memory of our GPU card can facilitate calculations with a maximum of 131072 crystals if the spectral CP calculations are performed with 1024 transforms. The addition of memory in hardware can enable the GPU based application to run more demanding problems either with a larger number of transforms or a larger number of crystals or both. Alternatively in smaller memory devices, a tandem of GPUs can be used to solve computationally demanding problems. Our computer code is written in the Portland Group, Inc. (PGI) Compute Unified Driver Architecture (CUDA) Fortran programing platform developed by the Portland Group in collaboration with NVIDIA. We used double precision in all calculations.

Reaching the peak performance requires a very careful programing of the application code. In this work, we adopt a divide and conquer approach from the GPU8 algorithm. The method is based on the efficient partitioning of multiplying matrices into sub-matrices where blocks of threads compute resulting sub-matrix and individual threads compute one or several elements of the resulting sub-matrix. In order to exploit the full potential, it is important to establish the dimension of the block threads that yield the best performances. The GPU threads are grouped into blocks and the blocks are organized as a grid. Kernel invocation requires the specification of the block and grid dimensions along with any parameters the kernel needs.

In the core of the spectral CP model is the evaluation of the series i.e. the calculation of the components of the deviatoric part of the Cauchy stress, $\boldsymbol{\sigma}'$, the plastic spin tensor, $\mathbf{W}^p$, and the deformation mode function, $\sum_{\alpha}\left|\dot{\gamma}^{\alpha}\right|$ for individual grains (see Eq. 2). These functions are computed based on the spectral representation of the crystal plasticity solutions described in [8, 19]. B, C, and G are pre-computed constants. Because the evaluation of the series is expensive computationally, the parallelization of that part of the code is essential. Calculation related to the texture evolution, hardening and homogenization are inexpensive and take less the 1% of the total time involved.

Our parallel implementation is executed in two steps. We developed a CUDA kernel for each step. The first kernel builds the matrix of exponentials while the second kernel evaluates the series in the form of a matrix-matrix multiplication. In our GPU implementation of the first kernel, four entries of the matrix of exponentials were evaluated on a single CUDA thread. The matrix elements are independent of each other and can be evaluated separately. The size of this matrix is *N x M*, where, *M*, represents the number of grains (r) in a polycrystalline aggregate and, *N*, is the number of spectral amplitudes used. Therefore a total of *NxM/4* threads were launch on the GPU for the first kernel. In our GPU implementation of the second kernel, we multiply the matrix of spectral amplitudes with the matrix of exponentials to obtain the crystal plasticity field variables:

$$\left\{\begin{array}{c} \mathbf{W}_{rq}^{p} \\ \boldsymbol{\sigma}_{rq}^{'} \\ \sum\limits_{\alpha}\dot{\gamma}_{rq}^{\alpha} \end{array}\right\}_{9\,x\,M} = \left\{\begin{array}{c} \mathbf{B}_{kn} \\ \mathbf{C}_{kn} \\ \mathbf{G}_{kn} \end{array}\right\}_{9\,x\,N} \begin{bmatrix} e^{\frac{2\pi ikr}{N_g}} e^{\frac{2\pi inq}{N_g}} & \dots & e^{\frac{2\pi ikr}{N_g}} e^{\frac{2\pi inq}{N_g}} \\ \vdots & \dots & \vdots \\ e^{\frac{2\pi ikr}{N_g}} e^{\frac{2\pi inq}{N_g}} & \dots & e^{\frac{2\pi ikr}{N_g}} e^{\frac{2\pi inq}{N_g}} \end{bmatrix}_{N\,x\,M} \quad (2)$$

The matrix-matrix multiplication is performed using a divide and conquer approach adapted from the GPU8 algorithm. Both the matrices are divided into sub-matrices and their multiplication results with the calculation of portions of the field variable matrix. We obtained the best performance if the second CUDA kernel is launched using thread blocks of a size of 16x9 where each thread is calculating two elements of the resultant matrix. We apply the GPU implementation of the spectral crystal plasticity to simulate a plane-strain compression (PSC) test and validate the results against the conventional model. In this study, a Taylor-type homogenization scheme is used to obtain the response of a polycrystalline aggregate. A polycrystalline aggregate is represented as a set of grains, each characterized by a crystallographic orientation and experiencing the same strain. The strain is applied in terms of a known velocity gradient tensor, **L**, identically to all grains comprising the polycrystalline aggregate. We emphasize that the responses of single-crystal can be homogenized in a different way e.g. using the self-consistent models. We consider here the case of face-centered cubic (FCC) poly-crystalline oxygen-free high conductivity (OFHC) copper. Hardening parameters are taken from [1]. While more physically based hardening laws exist [20, 21], here we use the simple Voce hardening law from [1].

The Fourier transform amplitudes were obtained in our earlier work [8, 19] and used here. This defines the spectral approximations of the conventional model for this work. The results of the spectral approximations run on the GPU platform are compared against those of the conventional model in Figs. 5 and 6. Figure 5 compares the stress-strain curves obtained from the conventional model and a spectral approximation for plane strain compression test. It can be seen that the spectral approximation compares well with the conventional model. The low strain level portion is less accurate because the spectral representation model is a rigid-plastic model, while the conventional model was an elasto-plastic. The high strain level portion is becoming less accurate because the spectral interpolation error is accumulating. Figure 6 compares texture predictions at a strain of 1.0. It can be seen that our implementation successfully reproduces the strain-stress curves and texture to large plastic strains. Therefore, the spectral representation of the conventional CP implemented programed on a GPU card is an accurate and fast computational framework aiming at next-generation large-scale crystal plasticity simulations.

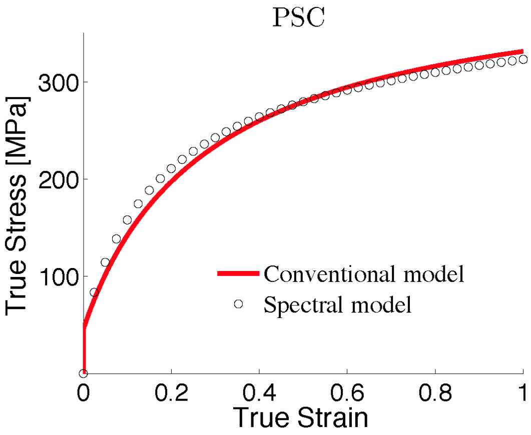

Figure 5 Comparison of stress-strain curves obtained using the conventional model and the spectral model with 1024 spectral coefficients in a plane strain compression test.

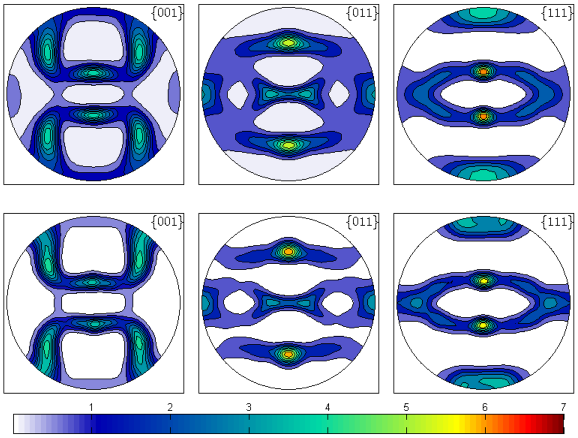

Figure 6 Predicted textures in plane strain compression at a strain of 1.0, as obtained from the conventional model (top row) and spectral approximations of the conventional model (bottom row). The initial texture in these simulations was 400 (see Fig. 3).

In Fig. 7 we show the speedup obtained by GPU parallel implementations of the spectral crystal plasticity theory over the conventional numerical methods for crystal plasticity. The results show an improvement in the efficiency exceeding three orders of magnitude. The efficiency of the algorithm improves with the increase in the size of the problem. A combination of data compaction methodology based on the generalized spherical harmonics and the spectral CP programed on GPU cards have potential to make practical large-scale forming simulations with crystal plasticity constitutive laws.

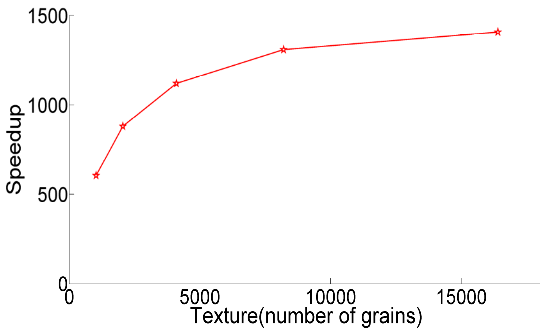

Figure 7 Speedup by the GPU-based parallelization of the CP simulations in the spectral representation over the simulations performed using the conventional model.

CONCLUSIONS

In this paper, we presented a methodology for compacting texture data and a high performance implementation of the spectral crystal plasticity framework aimed at performing computationally tractable metal forming simulations. Data compaction was possible using generalized spherical harmonics. The high performance implementation takes advantage of the advanced GPU8 matrix-matrix multiplication algorithms on a specialized hardware integrating one GPU card. Case studies involving a Taylor-type homogenization of a FCC polycrystalline material show improvements in the computational speed more than three orders of magnitude over the conventional numerical schemes. The speedup factor improves with an increase in number of grains considered in the simulations. Therefore, we conclude that synergy of data compaction and the spectral crystal plasticity programed on a GPU card can render next-generation large-scale forming simulations with crystal plasticity constitutive laws possible.

ACKNOWLEDGMENTS

This research was supported by the faculty startup funds from the University of New Hampshire.

REFERENCES

[1] Kalidindi SR, Bronkhorst CA, Anand L. Crystallographic Texture Evolution in Bulk Deformation Processing of Fcc Metals. Journal of the Mechanics and Physics of Solids. 1992;40:537-69.

[2] Lebensohn RA, Tomé CN. A self-consistent anisotropic approach for the simulation of plastic deformation and texture development of polycrystals: Application to zirconium alloys. Acta Metallurgica et Materialia. 1993;41:2611-24.

[3] Knezevic M, McCabe RJ, Lebensohn RA, Tomé CN, Liu C, Lovato ML, Mihaila B. Integration of self-consistent polycrystal plasticity with dislocation density based hardening laws within an implicit finite element framework: Application to low-symmetry metals. Journal of the Mechanics and Physics of Solids. 2013;61:2034-46.

[4] Knezevic M, Lebensohn RA, Cazacu O, Revil-Baudard B, Proust Gnl, Vogel SC, Nixon ME. Modeling bending of α-titanium with embedded polycrystal plasticity in implicit finite elements. Materials Science and Engineering: A. 2013; 564: 116-26.

[5] Kalidindi SR, Duvvuru HK, Knezevic M. Spectral calibration of crystal plasticity models. Acta Materialia. 2006;54:1795-804.

[6] Sundararaghavan V, Zabaras N. Linear analysis of texture-property relationships using process-based representations of Rodrigues space. Acta Materialia. 2007;55:1573-87.

[7] Barton NR, Knap J, Arsenlis A, Becker R, Hornung RD, Jefferson DR. Embedded polycrystal plasticity and adaptive sampling. International Journal of Plasticity. 2008;24:242-66.

[8] Knezevic M, Al-Harbi HF, Kalidindi SR. Crystal plasticity simulations using discrete Fourier transforms. Acta Materialia. 2009;57:1777-84.

[9] Knezevic M, Kalidindi SR, Fullwood D. Computationally efficient database and spectral interpolation for fully plastic Taylor-type crystal plasticity calculations of face-centered cubic polycrystals. International Journal of Plasticity. 2008;24:1264-76.

[10] Kalidindi SR, Knezevic M, Niezgoda S, Shaffer J. Representation of the orientation distribution function and computation of first-order elastic properties closures using discrete Fourier transforms. Acta Materialia. 2009;57:3916-23.

[11] Fast T, Knezevic M, Kalidindi SR. Application of microstructure sensitive design to structural components produced from hexagonal polycrystalline metals. Computational Materials Science. 2008;43:374-83.

[12] Knezevic M, Kalidindi SR. Fast computation of first-order elastic-plastic closures for polycrystalline cubic-orthorhombic microstructures. Computational Materials Science. 2007;39:643-8.

[13] Knezevic M, Kalidindi SR, Mishra RK. Delineation of first-order closures for plastic properties requiring explicit consideration of strain hardening and crystallographic texture evolution. International Journal of Plasticity. 2008;24:327-42.

[14] Junjie L, Ranka S, Sahni S. Strassen's Matrix Multiplication on GPUs. Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on2011. p. 157-64.

[15] Bunge H-J. Texture analysis in materials science. Mathematical Methods. Göttingen: Cuvillier Verlag; 1993.

[16] Shaffer JB, Knezevic M, Kalidindi SR. Building texture evolution networks for deformation processing of polycrystalline fcc metals using spectral approaches: applications to process design for targeted performance. International Journal of Plasticity. 2010;26:1183–1194.

[17] Wu X, Proust G, Knezevic M, Kalidindi SR. Elastic–plastic property closures for hexagonal close-packed polycrystalline metals using first-order bounding theories. Acta Materialia. 2007;55:2729–2737.

[18] Knezevic M, Savage DJ. A high-performance computational framework for fast crystal plasticity simulations. Computational Materials Science. 2014;83:101–106.

[19] Al-Harbi HF, Knezevic M, Kalidindi SR. Spectral approaches for the fast computation of yield surfaces and first-order plastic property closures for polycrystalline materials with cubic-triclinic textures. Computers Materials & Continua. 2010;15:153–172.

[20] Ardeljan M, Beyerlein IJ, Knezevic M. A dislocation density based crystal plasticity finite element model: Application to a two-phase polycrystalline HCP/BCC composites. Journal of the Mechanics and Physics of Solids. 2014;66:16-31.

[21] Knezevic M. Carpenter JS. Lovato ML. McCabe RJ. Deformation behavior of the cobalt-based superalloy Haynes25: Experimental characterization and crystal plasticity modeling. Acta Materialia. 2014;63:162–168.