Available online at www.sciencedirect.com
ScienceDirect
Computational Materials Science 42 (2008) 368–379
COMPUTATIONAL MATERIALS SCIENCE
www.elsevier.com/locate/commatsci

Readily regenerable reduced microstructure representations ☆

Keita Teranishi [a], Padma Raghavan [a,*], Jingxian Zhang [b], Tao Wang [c], Long-Qing Chen [b], Zi-Kui Liu [b]

[a] Department of Computer Science and Engineering, 111 IST Building, The Pennsylvania State University, University Park, PA 16802, USA
[b] Department of Materials Science and Engineering, 121 Steidle Building, The Pennsylvania State University, University Park, PA 16802, USA
[c] Department of Materials Science and Engineering, 2220 Hoover Hall, Iowa State University, Ames, IA 50011-2300, USA

Received 6 September 2006; received in revised form 18 June 2007; accepted 9 July 2007

Abstract

Many of the physical properties of materials are critically dependent on their microstructure. In recent years, there has been increasing interest in using computer simulations based on phase-field models for the spatial and temporal evolution of microstructures. Although such simulations are computationally expensive, the generated set of microstructures can be stored in a repository and used for further analysis in materials design. However, such an approach requires a substantial amount of storage, for example, approximately 1 Terabyte for a single binary alloy. In this paper, we develop fast data compression and regeneration schemes for two-dimensional microstructures that can reduce storage requirements without compromising the accuracy of computed values, such as stress fields used in analysis. Our main contribution is the development and evaluation of a *sparse skeletal representation* scheme which outperforms traditional compression schemes. Our results indicate that our scheme can reduce microstructure data size by more than two orders of magnitude while achieving better accuracies for the computed stress fields and order parameters.

PACS: 81.40.−z; 87.16.Ac

Keywords: Microstructure; Data compression; Phase-field model; Microstructure representation; Database

1. Introduction

One of the major research directions in computational materials science is to predict and analyze materials microstructures during processing. Microstructures indicate the spatial distributions of phases and other structural features at nano- and/or micro-scales [1]. Microstructures can be measured either experimentally or through computer simulations [1]. Recent advances in computing technology and mathematical schemes have made the latter more tractable and attractive.

Microstructures are either reconstructed and digitized from experiments or directly observed from computer simulations. Such microstructures can be stored in the form of a repository and used with data-mining techniques to explore the design space for optimum microstructures to potentially tailor new materials [2–5]. However, this approach will be feasible only if a large number of microstructures can be stored and manipulated in the repository without prohibitive storage requirements and the corresponding computation nd I/O costs, i.e., moving data to and from CPU and disks for analysis. For example, a computer generated microstructure on a 1024×1024 two-dimensional mesh typically requires 40 MB of space when each mesh point contains five double precision values (corresponding to properties such as orientation, composition, etc.). At a given temperature and composition, if the microstructure evolution is to be retained for 20 different time

☆ This work was supported in part by the National Science Foundation through the grants CCF-0444345, CNS-0202007 and DMR-0205232.
* Corresponding author.
E-mail address: raghavan@psu.edu (P. Raghavan).

0927-0256/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.commatsci.2007.07.015

stamps, the storage costs grow to $40 \times 20 = 800$ MB. Studying such a system for 10 different composition and temperature conditions will lead to a total storage requirement of at least 80 GB. For three-dimensional microstructure representations, the costs will be dramatically higher; calculations indicate that storage requirements can exceed 1.2 Terabytes for a similar study, using a $256 \times 256 \times 256$ microstructure.

In an earlier work towards building functionally useful repositories of limited size, Sundararaghavan and Zabaras [2] developed reduced order models. Another effort involves representing microstructures using the statistics of phase distribution [6,7] for intervals in the physical domain. In this approach, the number of intervals is not predetermined in order to allow an accurate approximation to the original microstructure. However, regenerating accurate microstructures from such representations can be computationally expensive.

In this paper, we develop compact representations of microstructures using 'lossy' methods, i.e., methods in which some of the details are lost. However, we conjecture that such a data loss can be corrected through a fast regeneration scheme in which the errors are corrected using a few iterations of a corresponding phase-field simulation. We view the microstructure as an image and consider multilevel forms of different schemes for image compression. Our main contribution is the development of a new *sparse skeletal representation* which can reduce data size by more than two orders of magnitude; corresponding regenerated and refined microstructures lead to the lowest errors for associated parameters.

This paper is organized as follows. Section 2 contains an introduction to microstructures and phase-field simulations followed by a brief review of existing data compression schemes. Section 3 introduces our microstructure data reduction and regeneration framework including our new compression schemes in Section 3.1. Section 4 provides an empirical evaluation of our schemes using two-phase microstructures from Ni–Al (γ and γ') and Al–Cu (α and θ') binary systems. It includes a comparison of compression rates achieved and the accuracy of regenerated data including the local stress distribution. Section 5 contains some concluding remarks.

2. Background

2.1. Microstructures of materials

A microstructure is a general term that refers to a spatial distribution of structural features which can be phases of (i) different compositions and/or crystal structures, (ii) grains of different orientations, (iii) domains of different structural variants, and (iv) domains of different electrical or magnetic polarizations, as well as structural defects.

The size, shape, and spatial arrangement of the local structural features in a microstructure play a critical role in determining the physical properties of a material such as mechanical, electrical, magnetic and optical properties. Microstructure evolution takes place to reduce the total free energy which may include the bulk chemical free energy, interfacial energy, elastic strain energy, magnetic energy, electrostatic energy, and/or under applied external fields such as applied stress, electrical, temperature, and magnetic fields. Many important properties of a material can be engineered by controlling its microstructure evolution. In computational materials science, a microstructure is digitized on two- or three-dimensional mesh points where each point can be associated with local thermodynamic and other physical properties.

One of the most familiar examples for a local property is the local composition of an atom species. Typically the values associated with composition are uniform within a domain or phase, but they exhibit sharp changes across the interfaces. In practice, graphical images are often used to provide an intuitive understanding on a morphology of phases. For example, Fig. 1 shows 2D graphical images of Ni–Al and Al–Cu microstructures with 512×512 mesh points, where the concentration of Al at each point is represented by a range of colors. According to this figure, there are two different phases representing *precipitates* (in red color) dispersed in a *matrix* (in blue color).

In strain-dominated two-phase microstructure evolution, it is important that the local stress distribution is calculated accurately. The stress-field $\sigma_{ij}(x,y)$ at (x,y) in a 2D microstructure is obtained using the Einstein summation convention as:

$$\sigma_{ij}(x,y) = C_{ijkl}(x,y)\epsilon_{kl}(x,y), \tag{1}$$

where $C_{ijkl}(x,y)$ is the elastic modulus tensor, and $\epsilon_{kl}(x,y)$ is the elastic strain within a microstructure at (x,y). The details regarding the calculation of the elastic strain can be found in [8] for the case of homogeneous modulus approximation.

2.2. Phase-field simulation

The phase-field approach is one of the most powerful methods for modeling many types of microstructure evolution processes [1,9]. Unlike classical approaches, a phase-field model describes a microstructure, both the compositional and structural domains and the interfaces between phases, as a whole by using a set of field variables. The field variables change smoothly from one phase to another across the interfacial regions, and hence the interfaces in a phase-field model are diffuse. Typical examples of field variables are the concentration and long-range order parameters that characterize the compositional and structural heterogeneities, respectively. The temporal evolution of the field variables toward the thermodynamics equilibrium are governed by a pair of continuum equations, namely, the Cahn-Hilliard nonlinear diffusion equation [10] and the Allen-Cahn (time-dependent Ginzburg-Landau) equation [11] shown as follows:

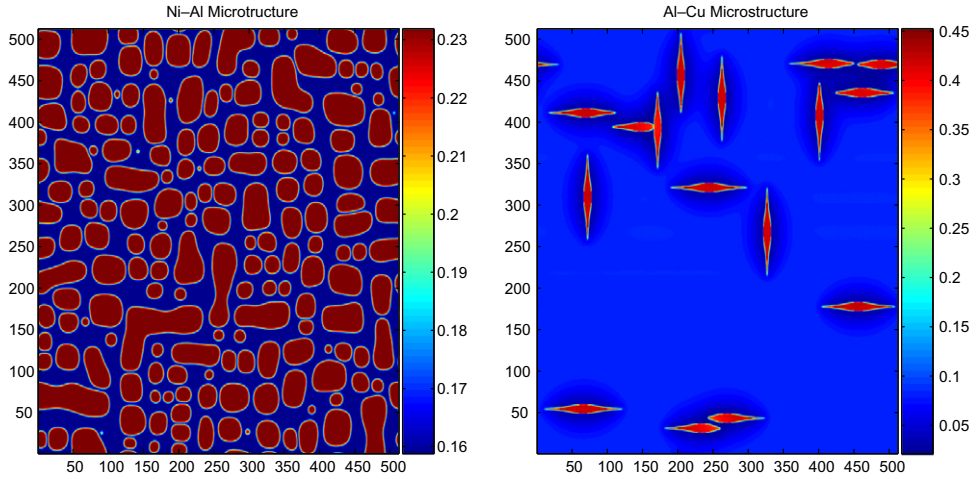

Fig. 1. Graphical images of the 2D Ni–Al (left) and Al–Cu (right) microstructure; the values represent the concentration of Aluminum for Ni–Al, and atomic fraction of Cu for Al–Cu.

$$\frac{\partial \eta_p(\mathbf{r},t)}{\partial t} = -L_{pq}\frac{\delta F}{\delta \eta_q(\mathbf{r},t)}, \quad \text{and} \tag{2}$$

$$\frac{\partial c(\mathbf{r},t)}{\partial t} = M\nabla^2\frac{\delta F}{\delta c(\mathbf{r},t)}. \tag{3}$$

L_{pq} and M are kinetic coefficients related to atom or interface mobility. c represents conserved field variables, η_p is non-conserved field variables, and F is the total free energy of the system. The solutions of these equations provide the temporal evolution of microstructures. To solve these kinetic equations, we employed the semi-implicit Fourier spectral method [12].

2.3. Data compression schemes

The data and image compression techniques can be categorized as performing either lossless or lossy compression. Additionally, both types of schemes can be combined to further increase compression rates [13,14].

Lossless compression schemes include Huffman coding schemes [15], and LZW algorithm [16,17] and their variants. These schemes are also used in the de facto standard compression software such as **zip** and **gzip**. Such methods convert frequent sequences of literals (symbols) to small sequences of bits (0/1 values) and these mappings are stored in a translation table. The associated decoding algorithm uses these translation table values to regenerate the original data from the compressed data with small computational costs.

Lossy schemes allow some data loss or inaccuracy in the regenerated data. These schemes tend to be more domain-specific as opposed to the general-purpose lossless schemes. The accuracy in the regenerated data varies with schemes, and the accuracy of some schemes can be tuned depending on the application. We next describe some lossy schemes that represent starting points in our work to develop compression schemes tailored for developing a microstructure repository.

2.3.1. Low-rank representations

A two-dimensional microstructure (or any image) can be viewed as an $m \times n$ matrix M, for some $m \geqslant n$. Such a matrix could have a rank as large as n thus representing very high dimensional data. A singular value decomposition (SVD) can be used as a compression scheme by constructing the closest p, $p < n$, rank approximation to M as:

$$M = U\Sigma V^{\mathrm{T}}. \tag{4}$$

The $m \times n$ matrix Σ contains positive real entries in its diagonal, and matrices $U\,(m \times m)$ and $V\,(n \times n)$ are unitary. The diagonal elements of Σ are called *singular values* and U and V are called *singular vectors*. Selecting the largest p singular values and corresponding singular vectors (say $\widehat{\Sigma}$, $\widehat{U}$ and $\widehat{V}$) enables a compact representation of M. Upon regeneration, the matrix $\widehat{M}$ is computed as the following matrix product:

$$\widehat{M} = \overline{U}\,\overline{\Sigma}\,\overline{V}^{\mathrm{T}}. \tag{5}$$

Such SVD schemes have been known to capture the main features with relatively small decompositions [18].

2.3.2. JPEG

The standard image compression scheme is JPEG (Joint Photographic Experts Group) [13,14]. It represents a combination of lossy compression based on discrete cosine transforms (DCT) [19] and lossless compression using Huffman coding [15]. For a given image M with $m \times n$ pixels, the algorithm partitions the domain into multiple 8×8 blocks. Then, it applies DCT to obtain the coefficients for 64 frequency components for each block. These coefficients are used when regenerating data through the backward transformation. Compression is achieved using *quantization*, which scales these coefficients to a small range of inte-

ger values. Subsequently, the reduced data after quantization can benefit from further compression through lossless Huffman coding.

2.3.3. Binary image segmentation

The numerical data in microstructures represents the distribution of matrix and precipitates as values associated with mesh points. For the Ni–Al systems of our interest, the data is expressed as the concentration of Aluminum in each mesh point. In this case, the matrix (γ phase) contains approximately 13% of Al, and the precipitates (γ' phase) contain approximately 20% of Al. These numbers, called 'average composition,' can be used to determine the phase at each mesh point. We can now apply this definition to distinguish a microstructure into binary numbers — a '0' represents matrix and a '1' represents precipitate.

This binary image segmentation scheme starts with thresholding to detect precipitate in the microstructure, where the threshold is determined by the method proposed by Otsu [20]. After this step, the precipitates are viewed as a connected components of the image [21], and a unique identification number (ID) is assigned to each precipitate [22]. In addition to the binary representation of the microstructure matrix, the method can store each precipitate's ID number and associated values (such as concentration or order parameter). This method can be successful at data reduction. However, it typically suffers from inaccuracies in representing the data at the interfaces between matrix and precipitates.

3. A framework for readily regenerable reduced microstructure representations

We now develop a framework for storing highly reduced representations of microstructures which can be easily regenerated to full accuracy. We seek schemes that enable compression by several orders of magnitude with negligible computational costs (in terms of execution time) when compared with the costs of the original phase field simulations.

Our framework involves a series of steps as shown in Fig. 2. We first store the input parameters for a phase-field simulation such as the Gibbs free energy, temperature, diffusivity of phases and elastic constants. Next we process microstructures generated by the simulation as follows. We apply lossy compression schemes that are tailored to significantly compress microstructures. Compressed microstructures obtained from this step are then further reduced using lossless 'gzip' like compression. Such a fully compressed microstructure is retrieved and regenerated when needed using the following series of steps. First, an inverse of the lossless gzip compression is applied followed by an inverse of the lossy compression scheme. In addition to directly using this regenerated microstructure, we can apply a refinement step to potentially further reduce errors. The physics of microstructure and phase-field simulations in Section 2.2 indicates that a simulation evolves to minimize the bulk energy of the physical domain. Thus, we can use the regenerated microstructure with the saved phase-field simulation parameters to apply a few steps of the simulation. Since this regenerated microstructure represents a small perturbation of the original, we expect that such a refinement will converge to recover the original structure or its evolved form after a few time steps.

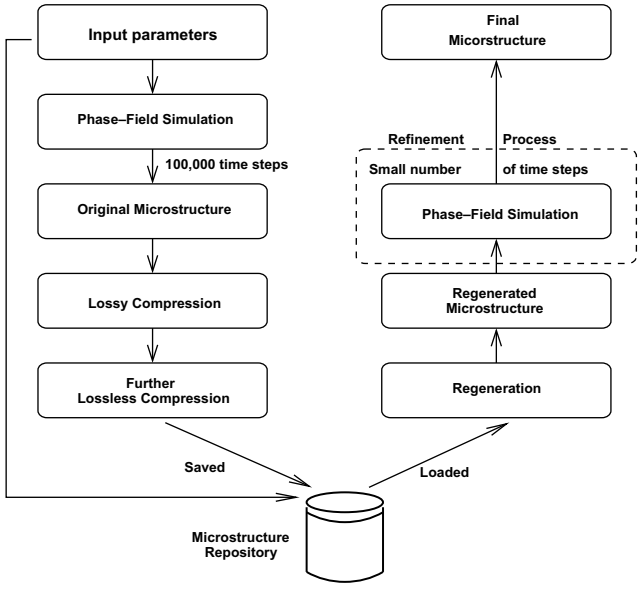

Fig. 2. A framework for our readily regenerable reduced microstructure representations.

We now discuss new lossy compression schemes that are specifically developed for processing microstructures.

3.1. New lossy compression schemes for microstructure compression

In this section, we develop two new lossy microstructure reduction schemes. The first is a *multilevel* scheme based on image resizing through interpolation that can be implemented in conjunction with existing lossy compression scheme. The second is our *sparse skeletal representation* (SSR) scheme which exploits the characteristics of microstructure data to achieve significant compression with high accuracy.

3.1.1. Multilevel scheme

Our multilevel scheme is in part motivated by multilevel methods used for the solution of sparse linear systems arising from the numeric solution of partial differential equations [23,24]. In our scheme, image resizing [25] applied recursively to reduce the *dimensions* of the microstructure by a factor of two at each level. Thus at each level, for a halving of the dimension, the data size is reduced by a factor of four. This process can be applied more than a once and the final reduced structure can then be compressed using standard compression techniques. We call our

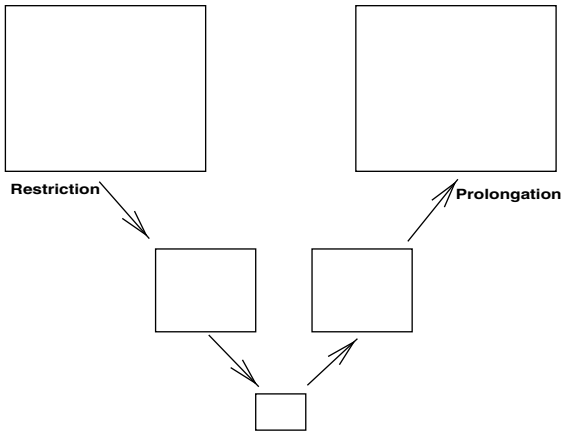

Fig. 3. The multilevel scheme for microstructure reduction.

scheme ML-JPEG when JPEG is used for compression of the reduced structure.

To regenerate the microstructure, an inverse interpolation is applied at each level to double the dimension and quadruple the data until the original dimensions are recovered. The overall multilevel (ML) process is illustrated in Fig. 3. The data reduction step (labeled as restriction) computes the average of four neighbors being eliminated, and this average value is assigned to the mesh points being retained. The recovery step (labeled as prolongation) applies bi-cubic interpolation of the neighbors. During the prolongation steps, we could potentially apply some correction schemes (such as running phase-field simulations or image deblurring filtering [25]) to avoid propagating the error in the lower level to higher levels.

3.1.2. Sparse skeletal representation

In this approach, we view the microstructure as a *sparse* object, i.e., one in which the important features are concentrated at only at $c \times N$ of the total N^2 points in an $N \times N$ image, where c is a small constant. Our *sparse skeletal representation* (SSR) scheme explicitly retains and operates only on data associated with interfaces of different phases. SSR uses thresholding to extract the interface region in a given microstructure. Then, coordinate indices and values (in 8-bit integers) of the interface are explicitly stored using three integer arrays; this is similar to the data structure for a sparse matrix $N \times N$ matrix. Fig. 4 shows the microstructure after the thresholding where the interfaces are clearly demarcated. The regeneration fills out the holes in the precipitate region with average values for precipitates (see Fig. 4) using an algorithm to detect holes in a segmented image [26]. The region for the matrix in the microstructure is filled with average values obtained from the original microstructure. This scheme is more suitable for microstructures where precipitates occupy a smaller fraction of domains.

4. Empirical results

We now evaluate the effectiveness of our readily regenerable reduced microstructure schemes with respect to (i) compression rates, and (ii) the accuracy of the regenerated microstructure. Our new schemes are applied to microstructures of Ni–Al and Al–Cu binary systems [27–29] and compared with well-known compression schemes reviewed in Section 2.3.

We describe our experiments and methodology in Section 4.1. In Section 4.2, we investigate how our sparse skeletal representation (SSR) differs from the standard JPEG scheme with respect to the accuracy of values associated with interface. Sections 4.3, 4.4 and 4.5 focus on a performance analysis of our schemes using metrics such as computational costs, compression factors, and quality of the regenerated data.

4.1. Methodology and experiments

All methods are tested by applying the compression and regeneration schemes to each component of the microstruc-

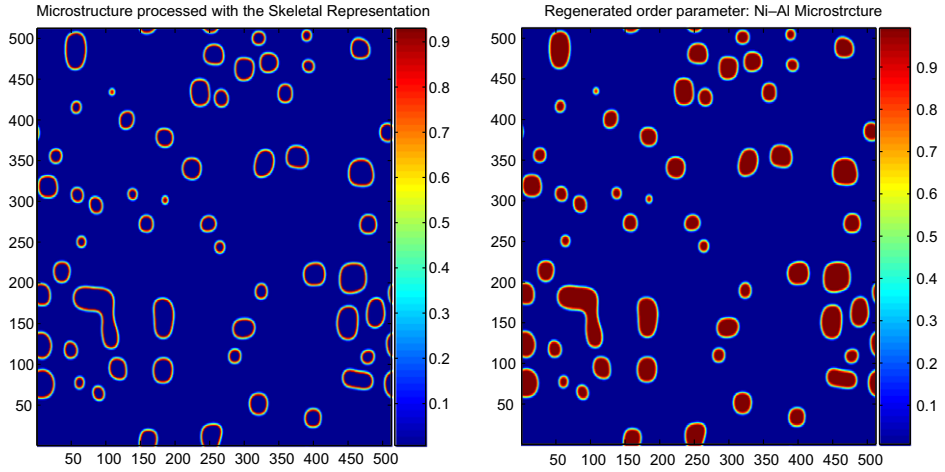

Fig. 4. The sparse skeletal representation of an order parameter of the 2D Ni–Al microstructure; data after thresholding (left) and data after regeneration (right).

tures of Ni–Al and Al–Cu binary systems represented using 512×512 regular mesh. For each system, we obtain microstructures at every 10,000 time steps from the 100,000 time step phase-field simulation for each system. Unless indicated otherwise, results in figures and tables indicate the average value for these 10 microstructures for each system.

The Ni–Al binary system contains γ and γ' phases evolved for approximately 40 s of real time through its phase-field simulation [29]. The initial microstructure for the simulation contains many small particles of precipitates in random distribution. The microstructure is represented by the composition field and four order parameter fields. The total total size of single microstructure data is approximately 10 MB (as binary-format file). The simulation involves time stepped ordinary differential equations, using the spectral method with a two-dimensional FFT [30]. For the simulation, we set 1173 K for the temperature, 2.0×10^{-9} m for the mesh size and 2.0×10^{-4} s for a single time step. The concentration of Al is set to 19.25% so that precipitates occupy approximately 50% of the physical domain. The input parameters for thermodynamic properties are determined from [31] and kinetic properties are obtained from modeling by [32,33]. Note that the preparation of microstructure with 100,000 time steps takes approximately 8 h.

The Al–Cu binary system has α and θ' phases evolved for approximately 881 s of the real time through the simulation [28]. The mesh size is 10^{-9} m and the microstructure is represented by the composition field and two order parameters for a total data size of approximately 6 MB. The temperature is set to 500 K and the concentration of Cu is set to 8.82% with 12 precipitate particles randomly deployed in the domain. Once again, kinetic and thermodynamic properties are obtained from appropriate models [28]. Such a phase-field simulation with 100,000 time steps takes approximately 32 h.

The parameters used for the compression schemes are as follows. For the JPEG scheme, we use its default values for the quantization parameters for standard image compression. For SVD, we use a rank-50 approximation, i.e., the largest 50 singular values and corresponding vectors; this dimension was selected in order to obtain accuracies comparable to other methods. For our ML-JPEG scheme, we use one level of restriction and prolongation to retain comparable accuracy in the regenerated data; more levels can be sought to meet the application needs. For our SSR scheme, we compute the thresholding value based on the largest and smallest values in a given microstructure ($v_{\max}$ and $v_{\min}$). We then treat the values less than $0.05(v_{\max} - v_{\min}) + v_{\min}$ as the matrix. Values greater than $0.95(v_{\max} - v_{\min}) + v_{\min}$ are treated as precipitate. All the methods are followed by gzip for further lossless compression. Our tests were performed on a 2.66 GHz Intel Pentium 4 Linux workstation with 1 GB memory.

We evaluate schemes using measures of the computational cost, the data compression rates and the accuracy of the regenerated data. The computational cost is defined by the execution time for the data compression and data regeneration. The accuracy is measured using the (i) matrix norm of the error for every parameter of the microstructure, and (ii) the maximum absolute error in the stress-field values. Let the original microstructure data and regenerated data M and $\widehat{M}$ respectively; both M and $\widehat{M}$ are 512×512 matrices. Now the matrix norm error is computed through the matrix infinity norm as:

$$\text{Error} = \frac{\|\widehat{M} - M\|_\infty}{\|M\|_\infty}. \tag{6}$$

For the values on the stress-field, the maximum error is computed as follows:

$$\text{MaxError} = \max_{i,j} |\widehat{M}_{ij} - M_{ij}|. \tag{7}$$

To measure the degree of compression, we use the compression factor F defined as the ratio of data sizes of the original and compressed microstructures,

$$F = \frac{\text{size}(M)}{\text{size}(\widehat{M})}. \tag{8}$$

4.2. Performance of sparse skeletal representation

We use an example to evaluate how our SSR method performs relative to the traditional JPEG scheme.

We first observe that JPEG can preserve the morphology of precipitates very accurately as shown in Fig. 5. However, as shown in Fig. 6, the stress-field computed from the order parameters exhibits a large difference for the regenerated data when compared to the original. Furthermore, as shown in Fig. 7, the differences (errors) in the stress-field are magnified at the interface regions between matrix and precipitate because JPEG fails to retain the accuracy of order parameters. Our SSR method seeks to address that problems by retaining exact values at the interface. As indicated in Fig. 8, this results in a more accurate stress-field representation.

4.3. Computational costs

We now investigate the efficiency of the data reduction schemes using execution time as a measure of computational costs. Table 1 demonstrates that all the schemes are computationally inexpensive at under a few hundred seconds. This is quite negligible in relation to the time required for the original phase-field simulation in the order of 8–32 h. The compression using SVD is time consuming because of the complexity of the SVD algorithm $O(N^3)$ for an $N \times N$ microstructure. The other schemes have lower complexities: $O(N^2 \log N)$ for JPEG and $O(N^2)$ for segmentation. The skeletal representation requires relatively large amount of time to compress the data for the Ni–Al system because of the large number of particles in the system. However, these costs are still very small compared to the time for the original phase-field simulation and they grow

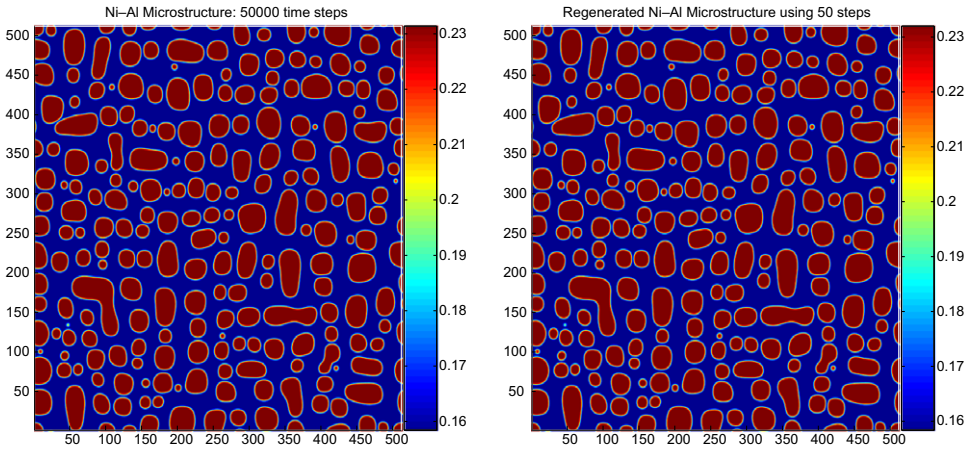

Fig. 5. Microstructures for the 2D Ni–Al system: concentration in original (left) and concentration using regenerated data after JPEG compression (right).

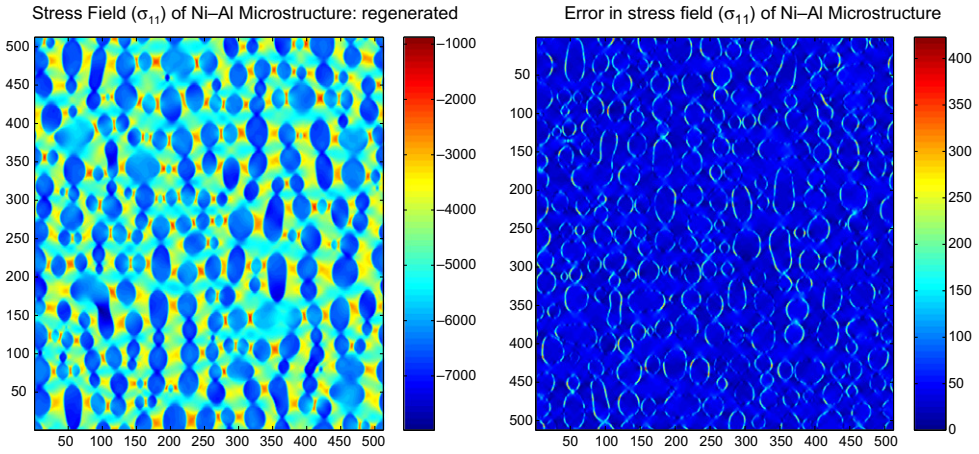

Fig. 6. The stress-field for the Ni–Al system regenerated after using JPEG (left) and the error (right) compared to the original data.

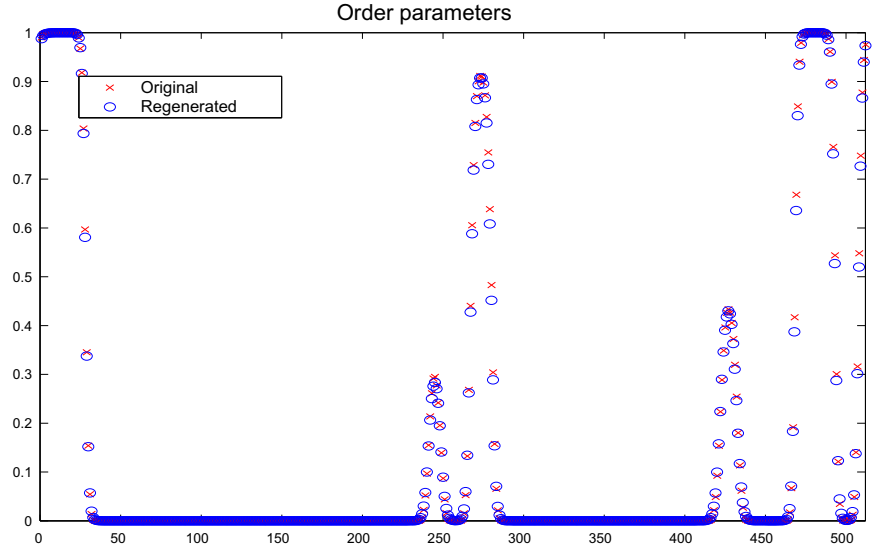

Fig. 7. The values of order parameters of a Ni–Al microstructure where the data represents the values on the cross-section at the middle of Y-coordinate. The X-axis represents the X-coordinate of 2D microstructure and Y-axis represents the value.

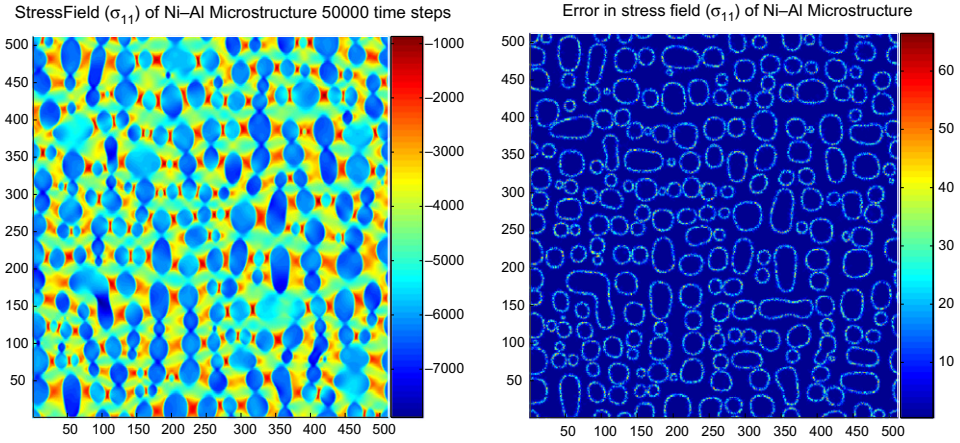

Fig. 8. The stress-field for the Ni–Al system regenerated after using SSR (left), and the error (right) compared to the original data.

Table 1
Execution time in seconds for reduced representation and regeneration

Method	System					
	Ni–Al			Al–Cu		
	Compress	Regenerate	Total	Compress	Regenerate	Total
SVD	58.2	0.05	58.3	57.4	0.05	57.5
JPEG	0.4	0.3	0.7	0.3	0.2	0.5
Segment	0.7	0.3	1.0	0.2	0.1	0.3
ML-JPEG	1.2	0.7	1.9	0.5	0.4	0.9
SSR	115.2	3.7	121.9	0.5	1.7	2.3

Execution time of a 100,000 step phase field simulation for Ni–Al is 8 h; for Al–Cu it is 32 h.

linearly with the microstructure size. Regeneration is almost equally efficient across schemes.

4.4. Compression factors

We next compute the average compression factor of 10 different microstructures for each system (obtained after every 10,000 time steps in an evolution of 100,000 time steps). An important metric is the factor by which the data size is compressed. We represent this by the compression factor F which indicates that the final microstructure is $\frac{1}{F}$ of the original size.

The average compression factor is shown in Fig. 9 for the Ni–Al and Al–Cu systems. For both systems, SVD is less effective than the other schemes. Segmentation appears to be the best. However, as we show in the next section, the regenerated data is highly inaccurate with respect to the order parameters and composition. Observe that in ML-JPEG scheme, the ML stage (1 level) provides a compression factor of 4. The overall compression factor of ML-

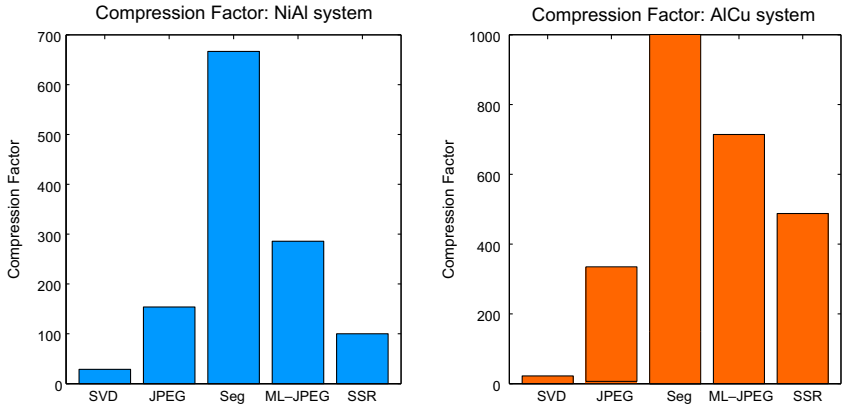

Fig. 9. The compression factors (F) of reduced representation schemes for Ni–Al microstructure (left) and Al–Cu microstructure (right); reduced microstructure is $\frac{1}{F}$ of the original size.

JPEG is the product of 4 (the compression factor of ML level 1) and the compression factor from applying JPEG on the ML output. The latter is typically lower than for JPEG on the original data; JPEG's compression factors are known to vary depending on data [13,14]. This is reflected in Fig. 9, where compression factors for Ni–Al and Al–Cu using ML-JPEG are better than JPEG, but only by a factor of 2. SSR shows fairly good compression factors for the Ni–Al system, at approximately 100. SSR compression factors are higher at approximately 400 for simpler systems such as Al–Cu.

4.5. Quality of regeneration

The quality of the regenerated microstructure data can be measured by differences in composition, order parameters and analyzed stress from those in the original ones. We first study the errors for all the schemes and the error

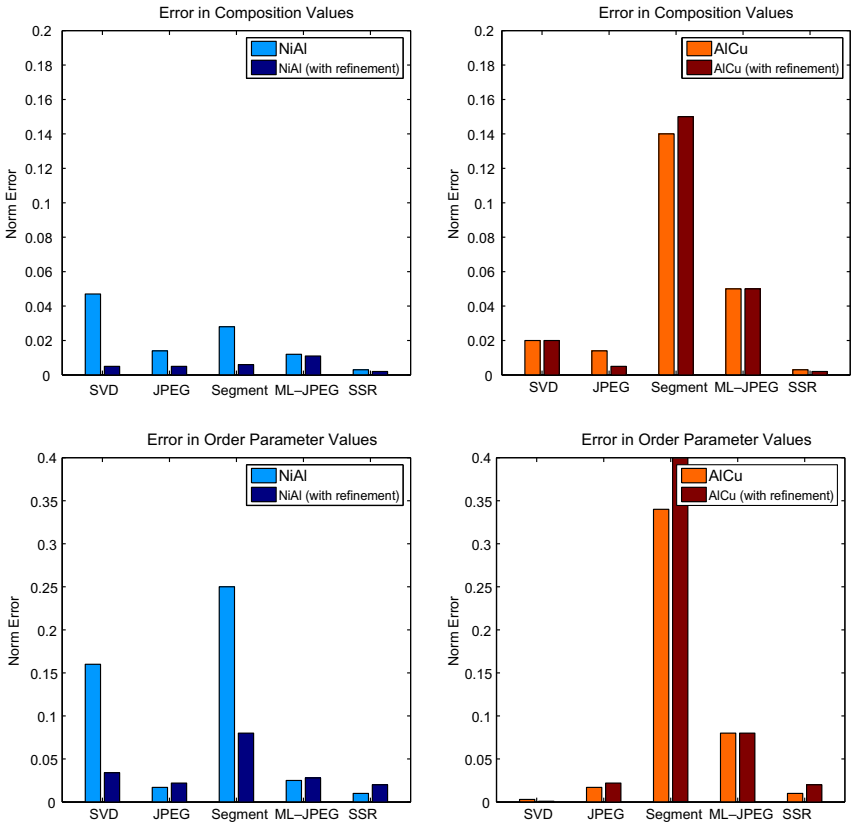

Fig. 10. Relative norm error for composition (top) and order parameters (bottom) of the regenerated microstructures for Ni–Al microstructure (left) and Al–Cu microstructure (right). The errors are shown before refinement ($\widehat{M}_t$ with respect to M_t), and after refinement with 50 time steps of the phase field simulation ($\widehat{M}_{t+50}$ with respect to M_{t+50}).

Table 2
Relative norm error in composition with different time steps of refinement (x) for Ni–Al and Al–Cu systems for $t = 50{,}000$

Method	$x = 0$	$x = 10$		$x = 30$		$x = 50$	
		M_t	M_{t+x}	M_t	M_{t+x}	M_t	M_{t+x}
Ni–Al system							
SVD	0.0279	0.0104	0.0104	0.0066	0.0066	0.0056	0.0056
JPEG	0.0128	0.0054	0.0054	0.0048	**0.0047**	0.0047	**0.0045**
ML-JPEG	0.0190	0.0103	0.0103	0.0100	0.0100	0.0101	0.0101
SSR	0.0035	0.0020	0.0020	0.0015	0.0015	0.0014	**0.0013**
Al–Cu system							
SVD	0.0326	0.0298	0.0298	**0.0273**	0.0281	**0.0256**	0.0259
JPEG	0.0143	0.0129	0.0129	0.0117	**0.0111**	0.0101	**0.0099**
ML-JPEG	0.0624	0.0608	0.0608	0.0581	0.0581	0.5578	**0.0555**
SSR	0.0067	0.0062	0.0062	0.0061	**0.0059**	0.0065	**0.0056**

The column labeled M_t denotes the error in $\widehat{M}_{t+x}$ with respect to M_{t+x}. The column labeled M_{t+x} denotes the error in $\widehat{M}_{t+x}$ with respect to M_{t+x}. For each method, the smaller value of the error is indicated in bold.

against the original microstructure after refinement using 50 time steps of the phase-field simulations. The comparison is made using the original microstructure at tth time step, M_t for t in the range 10,000 to 100,000, the regenerated microstructure $\widehat{M}_t$ and a refined regenerated microstructure $\widehat{M}_{t+50}$. Fig. 10 indicates that segmentation is highly inaccurate for both systems and in particular, and fails to regenerate for the Al–Cu system which has very thin particles. The SVD scheme achieves good accuracy for Al–Cu because it successfully captures the smaller set of features in this system. However, SVD fails to maintain a good accuracy for the Ni–Al systems. This inaccuracy can potentially be reduced by increasing the rank and thus retaining more singular vectors. However, this will come at the expense of smaller compression factors, which are already very small as shown in Fig. 9. For both Ni–Al and Al–Cu systems, SSR achieves very accurate data regeneration compared to other schemes.

Fig. 10 indicates the impact of the refinement after evolution using 50 time steps of the phase-field simulations. The first bar in each graph shows the error between $\widehat{M}_t$ and M_t while the second bar of each graph shows the error

Table 3
Relative norm error in order parameters with different time steps of refinement (x) for Ni–Al and Al–Cu systems for $t = 50{,}000$

Method	$x = 0$	$x = 10$		$x = 30$		$x = 50$	
		M_t	M_{t+x}	M_t	M_{t+x}	M_t	M_{t+x}
Ni–Al system							
SVD	0.0726	0.0609	**0.0600**	0.0485	**0.0453**	0.0423	**0.0368**
JPEG	0.0174	0.0159	**0.0149**	0.0169	**0.0136**	0.0196	**0.0142**
ML-JPEG	0.0243	0.0274	**0.0264**	0.0312	**0.0277**	0.0351	**0.0290**
SSR	0.0177	0.0150	**0.0141**	0.0140	**0.0107**	0.0146	**0.0087**
Al–Cu system							
SVD	0.0066	0.0092	**0.0059**	0.0189	**0.0072**	0.0273	**0.0068**
JPEG	0.0240	0.0201	**0.0189**	0.0229	**0.0132**	0.0286	**0.0100**
ML-JPEG	0.0799	0.0780	**0.0756**	0.0766	**0.0689**	0.0782	**0.0636**
SSR	0.0245	0.0237	**0.0230**	0.0275	**0.0206**	0.0324	**0.0195**

The column labeled M_t denotes the error in $\widehat{M}_{t+x}$ with respect to M_{t+x}. The column labeled M_{t+x} denotes the error in $\widehat{M}_{t+x}$ with respect to M_{t+x}. For each method, the smaller value of the error is indicated in bold.

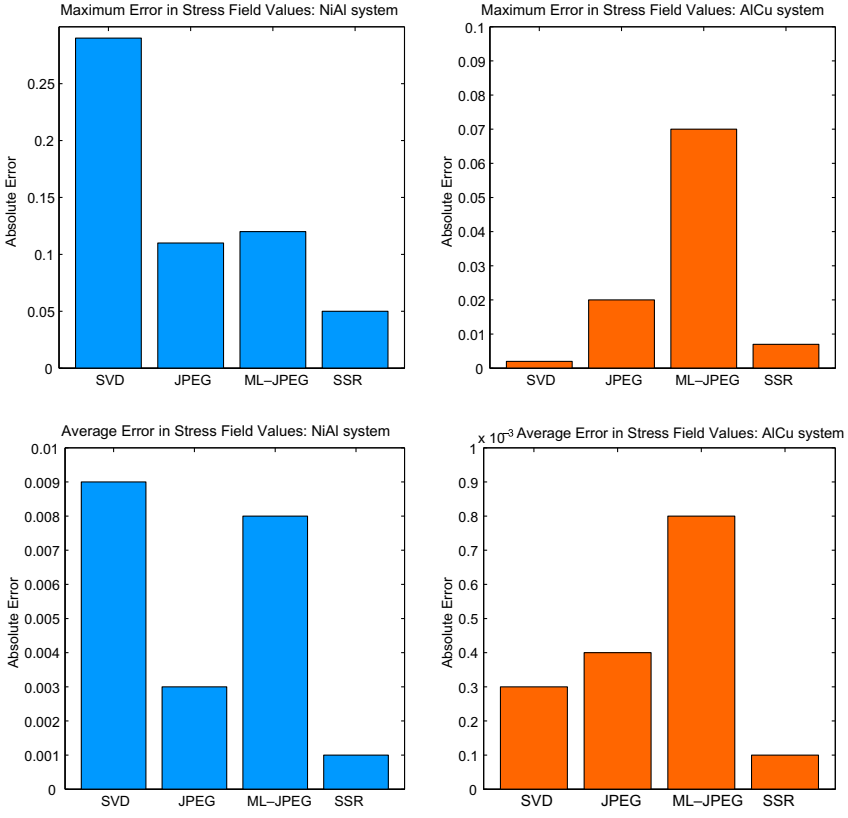

Fig. 11. Maximum (top) and average (bottom) error for internal stress of the regenerated microstructures for 2D Ni–Al microstructure (left) and Al–Cu microstructure (right).

between $\widehat{M}_{t+50}$ and M_t. Observe that refinement reduces the error in the composition for all methods for the Ni–Al system. However, the errors in the composition for Al–Cu and the order parameters for both systems do not necessarily improve after refinement. This can potentially be attributed to the fact that the error is computed as the difference between the original microstructure, M_t, and the regenerated microstructure which has further evolved after 50 time steps, i.e., $\widehat{M}_{t+50}$. Thus, the error shown in Fig. 10 includes the effect of further evolution of the regenerated microstructure. This raises interesting issues on how the error should be defined and evaluated. Although thus far we have considered the error in $\widehat{M}_{t+50}$ with respect to M_t, it may be quite appropriate to instead consider the error between $\widehat{M}_{t+50}$ and M_{t+50}, i.e., the original microstructure after 50 time steps of evolution.

We now evaluate the error (after compression, regeneration and refinement), that can be attributed to the difference between M_t and $\widehat{M}_{t+x}$ due to the evolution of M_t by x time steps to M_{t+x}. For this evaluation, we choose $t = 50{,}000$ and $x = 0{,}10{,}30$ and 50. Tables 2 and 3 indicate the error in composition and order parameters with respect to x in both forms, i.e., $\widehat{M}_{t+x}$ with respect to M_t and $\widehat{M}_{t+x}$ with respect to M_{t+x}. Table 2 indicates that the error in composition does not change in both forms. However, the error in order parameters shown in Table 3 is substantially smaller when the error is evaluated with respect to M_{t+x}. This indicates that comparing $\widehat{M}_{t+x}$ with M_t does not account for the further evaluation through x time steps of refinement. Thus, our overall scheme is indeed effective. Furthermore, it may be more appropriate to consider the error with respect to M_{t+x} because of the impact of further evolution.

We now consider the impact of regeneration on the computed internal stress and we report errors in Fig. 11. The errors correspond to the best instance of results (with or without refinement) shown in Fig. 10. Note that we do not report the performance of segmentation because the inaccuracy of order parameters makes stress analysis impractical. For all the schemes, the maximum error is observed at the interface between two different phases and the average error is approximately two orders of magnitude smaller than the maximum error. The SVD scheme has low maximum error for Al–Cu, but the corresponding value is much higher for the more complex Ni–Al system. More significantly, average errors for SVD are substantially higher than for SSR. The bar graphs in Fig. 11 clearly indicate that our SSR scheme has the least average and worst case errors for both Ni–Al and Al–Cu systems. This is because it successfully preserves values associated with the interface of different phases while achieving high orders of compression.

5. Conclusions

In this paper, we have developed a framework for efficient construction of reduced representations of two-dimensional microstructures, to decrease storage requirements for building repositories for computational materials design. Our framework allows for fast regeneration to near original accuracy and the use of a few time steps of phase-field simulation for further refinement to potentially reduce errors. Our new 'sparse skeletal representation' (SSR) scheme is particularly effective at achieving high factor of compressions and higher accuracy of computed order parameters and stress fields. This is primarily because SSR can compress data while allowing its accurate regeneration, particularly at interface regions where other schemes like JPEG suffer from relatively large errors.

Future extensions of our work concern the development of compression and regeneration for three-dimensional microstructure data. We also plan to extend our study for improving effectiveness of the data refinement through phase-field simulations. As one reviewer indicated, there is also a need to quantify and model the tradeoffs between the compression factor, accuracy of reduced representations, and computational costs for compression and regeneration.

Acknowledgement

The authors wish to thank all the reviewers for their helpful comments and suggestions.

References

[1] L.Q. Chen, Annu. Rev. Mater. Res. 32 (2002) 113–140.
[2] V. Sundararaghavan, N. Zabaras, Acta Mater. 53 (2004) 4111–4119.
[3] Z.K. Liu, L.Q. Chen, P. Raghavan, Q. Du, J. Sofo, S.A. Langer, C. Wolverton, J. Comput.–Aid. Mater. 11 (2004) 183–199.
[4] C.C. Seepersad, M.G. Fernandez, J.H. Panchal, H.-J. Choi, J.K. Allen, D.L. McDowell, F. Mistree, Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, 2004, Paper Number: AIAA-2004-4300.
[5] K. Teranishi, P. Raghavan, Z.K. Liu, Proceedings of CCGrid04: IEEE International Symposium on Cluster Computing and the Grid, Chicago, IL, 2004, pp. 664–669.
[6] B.L. Hansen, B.L. Adams, M.E. Lyon, A.J. Henrie, J. Comput.–Aid. Mater. 10 (2003) 163–173.
[7] S. Torquato, Annu. Rev. Mater. Res. 32 (2002) 77–111.
[8] A.G. Khachaturyan, Theory of structural transformations in solids, Wiley, New York, 1983.
[9] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Annu. Rev. Mater. Res. 32 (2002) 163–194.
[10] J.W. Cahn, Acta Metall. Mater. 9 (1961) 795–801.
[11] S.M. Allen, J.W. Cahn, J. Phys. 9 (1961) 795–801.
[12] L.Q. Chen, J. Shen, Computer Physics Communications 108 (1998) 147–158.
[13] W.B. Pennbaker, J.L. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1992.
[14] G.K. Wallace, Communications of the ACM 34 (4) (1991) 30–44.
[15] D.A. Huffman, Proc. IRE 40 (10) (1952) 1098–1101.
[16] T. Welch, IEEE Comput. Magaz. 17 (6) (1984) 8–19.
[17] J. Ziv, A. Lempel, IEEE Trans. Inform. Theor. 23 (3) (1977) 337–343.
[18] J.W. Demmel, Applied Numerical Linear Algebra, SIAM, 1997.
[19] G. Strang, SIAM Rev. 41 (1) (1999) 135–147.
[20] N. Otsu, IEEE Trans. Syst., Man, Cybernet. 9 (1) (1979) 62–66.
[21] R. Sedgewick, Algorithms in C, third ed., Addison-Wesley, 1998, pp. 11–20.
[22] R.M. Haralick, L.G. Shapiro, Computer and Robot Vision vol. I, Addison-Wesley, 1992, pp. 28–48.

[23] W.L. Brigg, V.E. Henson, S.F. McCormick, Multigrid Tutorial, SIAM, second ed., 2000.
[24] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition, Cambridge University Press, Cambridge, England, 1996.
[25] R.C. Gonzalez, R.E. Woods, Digital Image Processing, second ed., Prentice Hall, New Jersey, USA, 2002.
[26] P. Soille, Morphological Image Analysis: Principles and Applications, Springer-Verlag, 1999.
[27] J.Z. Zhu, T. Wang, A.J. Ardell, S. H Zhou, Z.K. Liu, L.Q. Chen, Acta Mater. 52 (2004) 2837–2845.
[28] V. Vaithyanathan, C. Wolverton, L.Q. Chen, Acta Mater. 52 (10) (2004) 2973–2987.
[29] J. Zhu, L.Q. Chen, J. Shen, V. Tikare, Phys. Rev. E 60 (4) (1999) 3564–3572.
[30] M. Frigo, A fast Fourier transform compiler, in: Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation, ACM, May 1999, pp. 169–180.
[31] I. Ansara, N. Dupin, B. Sundman, CALPHAD 25 (2001) 279–298.
[32] A. Engstrom, J. Agren, Z. Metallkd. 87 (1996) 92–97.
[33] T. Wang, J.Z. Zhu, R.A. Mackay, L.Q. Chen, Z.K. Liu, Metall. Mater. Trans. A 35A (2004) 2313–2321.