

利用課題名 フェーズフィールド法による金属材料のデンドライト凝固成長の 大規模 GPU 計算

¹青木尊之、²下川辺隆史、³高木知弘、⁴山中晃徳、¹額田 彰、¹遠藤敏夫、¹丸山直也、¹松岡 聡

- 1 東京工業大学学術国際情報センター
- 2 東京工業大学大学院総合理工学研究科
- 3 京都工芸繊維大学大学院工芸科学研究科
- 4 東京工業大学大学院理工学研究科

邦文抄録

軽量・高強度な新材料の開発は低炭素社会の実現に向けて非常に重要である。材料強度を左右する材料のミクロな組織は凝固過程で決定されるが、機械的強度を判定するには数 mm というマクロなスケールまでの大規模計算を必要とする。フェーズフィールド法というメソスケールのモデルを用い、TSUBAME 2.0 で合金の樹枝状凝固の大規模シミュレーションを行った。CUDA を用いて有限差分法で離散化された時間発展方程式を解き、領域分割により並列化することで TSUBAME 2.0 の 4000 GPU を用いて、2.0 PFLOPS（単精度）という高い実行性能を得た。この成果に対し、2011 年の ACM ゴードンベル賞・特別賞（本賞）が与えられた。

Keywords: GPGPU、フェーズフィールド法、CUDA、ステンシルアプリケーション

1. はじめに

軽量・高強度な新材料を開発することにより、物資を高効率（低燃費）で輸送することができ、低炭素社会の実現に大きく貢献する。金属材料の強度はミクロな材料組織に強く依存し、そのミクロ組織は凝固過程において形作られる（図1）。一方、現実に機械的強度を判定するには、数 mm のマクロなスケールでの解析が必要となる。ミクロな凝固ダイナミクスを解明するために、非平衡統計力学から導出されるフェーズフィールド法¹が近年注目されている。導出される方程式は時間空間の偏微分方程式になっていて、有限差分法や有限要素法などの格子法で解かれることが多い。フェーズフィールド法は複雑な非線形項を多く含み、ステンシル計算で解く場合には1格子点あたりの演算量が多くなる。さらに、非常に狭い界面領域に複数の格子を含む必要があるため、格子サイズを小さくする必要があり、格子数が膨大になり時間ステップを小さく取らなければならない。このため、CPU で計算すると時間がかかり過ぎるため、これまでは主に2次元計算による解析が行われてきた。

本研究では、フェーズフィールド法に基づいて二元合金の過冷却凝固における樹枝状（デンドライト）組織の成長を複数 GPU により大規模計算する。CUDA を用いて有限差分法で離散化された時間発展方程式をプログラミングし、MPI を用い領域分割法で並列化することで TSUBAME 2.0 のほぼ全ての計算資源を使った大規模計算を行う。フェーズフィールド法による超大規模計算を行った例は国内外で報告されておらず、本計算結果が材料科学分野に与えるインパクトは大きい。

2. フェーズフィールド法

フェーズフィールド法は非平衡統計物理学から導出され、

分子スケールとマクロなスケールの中間のメソスケールの現象を記述することができる。秩序変数 ϕ （フェーズフィールド変数）を導入し、固相部分に $\phi = 1$ を、液相部分に

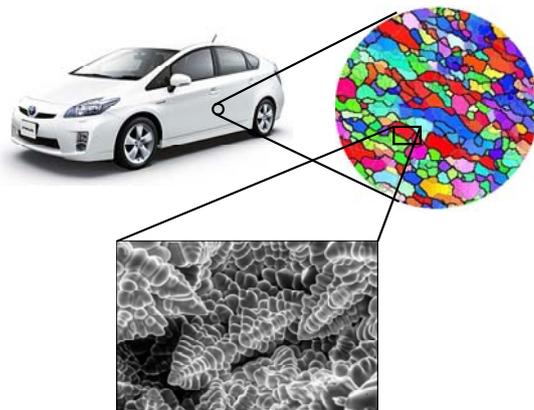


図1 材料のミクロな組成。

$\phi = 0$ を設定する。界面を含む領域は ϕ が 0 から 1 へと急峻かつ滑らかに変化する拡散界面として扱われ、 $\phi = 0.5$ が界面位置となる。フェーズフィールド法では従来から使われている界面追跡法等の手法が不要となり、領域全体で同一の計算を行うことができる。

本研究で対象とする二元合金の樹枝状凝固成長では、フェーズフィールド方程式と溶質の拡散方程式を解く。フェーズフィールドに対しては、界面エネルギーの異方性を考慮した次の方程式

$$\begin{aligned} \frac{\partial \phi}{\partial t} = M_{\phi} & \left[\nabla \cdot (a^2 \nabla \phi) + \frac{\partial}{\partial x} \left(a \frac{\partial a}{\partial \phi_x} |\nabla \phi|^2 \right) \right. \\ & + \frac{\partial}{\partial y} \left(a \frac{\partial a}{\partial \phi_y} |\nabla \phi|^2 \right) + \frac{\partial}{\partial z} \left(a \frac{\partial a}{\partial \phi_z} |\nabla \phi|^2 \right) \\ & \left. - \Delta S \Delta T \frac{d\mu(\phi)}{d\phi} - W \frac{dq(\phi)}{d\phi} \right] \end{aligned} \quad (1)$$

を解く。ここで a は勾配係数、 M_ϕ はモビリティであり、 W はエネルギー障壁、 ΔS は融解エントロピー、 ΔT は過冷却を表している。また、関数 $p(\phi)$ と $q(\phi)$ は、それぞれ $p(\phi) = \phi^3(10 - 15\phi + 6\phi^2)$ と $q(\phi) = \phi^2(1 - \phi)$ を用いている。

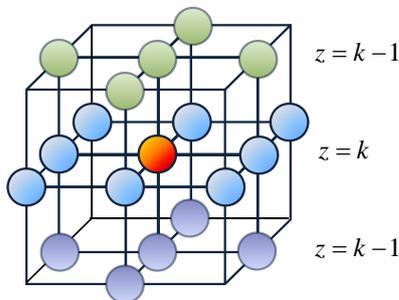
一方、溶質の拡散方程式は、次の方程式を用いる。

$$\frac{\partial c}{\partial t} = \nabla \cdot [D_S \phi \nabla c_S + D_L (1 - \phi) \nabla c_L] \quad (2)$$

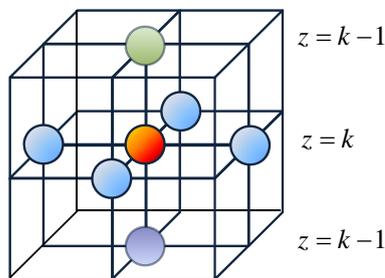
ここで、 D_S と D_L はそれぞれ固相、液相の拡散係数を表す。 c_S 、 c_L は $c = (1 - \phi)c_L + \phi c_S$ を満たしている。

3. 単一 GPU への実装

3次元直交格子上で(1)式と(2)式を2次精度有限差分法で離散化し1次精度の時間積分(オイラー法)を行う。CUDAを用いてGPUコンピューティングのプログラミングを行い、フェーズフィールド変数、濃度変数は全てビデオ・メモリ(CUDAではグローバルメモリと呼ばれる)上に確保し、計算の毎時間ステップでのPCI-Express Busを介したGPUとCPU間のデータ通信を排除している。3次元空間の格子点 (i, j, k) 上のフェーズフィールド変数 ϕ (図2(a))と濃度変数 c (図2(b))を時間発展するために必要な近傍格子点上の変数の空間配置を示す。1タイムステップでは、それぞれの格子点を時間発展するために、フェーズフィールド変数 ϕ の19格子点、濃度変数 c の7格子点を読み込み、メモリへ2回書き込む。



(a) フェーズフィールド変数のステンシル



(b) 濃度変数のステンシル

図2 時間発展に必要な隣接格子点アクセス。

フェーズフィールド・モデルの計算は多くのメモリアクセスを伴うため、効率よく計算するためにはグローバルメモリへのアクセス回数を低減することが重要である。GPUコ

ンピューティングでは、スレッドおよびブロックをどのように実際の計算に割り当てるかが高速化のキーポイントである。1つのGPUが担当する計算領域サイズを $n_x \times n_y \times n_z$ とし、図3のように $64 \times 4 \times 32$ の複数のピースに分割する。GPUで実行されるカーネル関数を1つのスレッドブロックが1つのピースを担当するよう設定し、各ブロックで $64 \times 4 \times 1$ のスレッドを実行させる。ある点 (i, j, k) を割り当てられたスレッドは、 z 方向へマーチング(ループカウンタ k)しながら (i, j, k_0) から (i, j, k_0+31) の32格子点を計算する。ここで k_0 は32の倍数で $0, 32, 64, \dots, n_z - 32$ である。あるスレッドが $k + 1$ 番目の面を計算するのに必要なデータの一部は k 番目の面を計算した際にそのスレッド自身によってすでに使われている。このようなデータはレジスタに一時的に保持し再利用することで、再びグローバルメモリへアクセスすることを回避することができる。本研究で用いる TSUBAME 2.0 に搭載された M2050 GPU は Fermi コアの GPU であり、L1/L2 キャッシュが利用できるため、それまでは複数のスレッド間でデータを共有する目的(ソフトウェア・マネージド・キャッシュ)でシェアードメモリを使用する必要がなくなった。

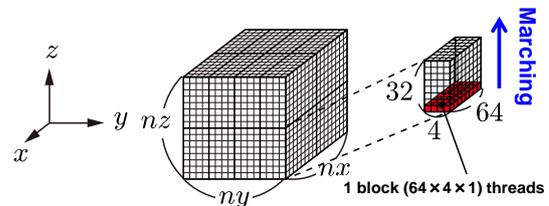


図3 CUDA スレッドとブロックの割り当て

4. 複数ノードに搭載された GPU による計算

大規模問題を高速に計算するには、複数の GPU を用いて計算することが必須となる。GPU 内部での並列計算に加えて、上位での GPU 単位での並列化が必要となる。大規模並列計算では、3次元領域分割を行うことにより領域間のデータ通信量を最小にすることができるが、GPU 計算の場合は x 方向で分割した場合の y - z 断面のメモリアドレスが不連続となり、メモリアクセス性能の低下のオーバーヘッドの方が通信量の低減より大きくなる。ここでは、 y 方向、 z 方向に分割する2次元の領域分割法を用い、それぞれの分割領域の計算を各 GPU へ割り当てる。複数 CPU 計算と同様に隣接する GPU 間での境界領域のデータ交換が必要になるが、現状では GPU は他の GPU のグローバルメモリ上のデータに直接アクセスすることができない。そこで GPU 間のデータ転送はホスト側のメモリを経由し、次の3段階で構成される。(1) CUDA ランタイムライブラリによる GPU から CPU への転送、(2) MPI ライブラリによる CPU 間のデータ転送、(3) CUDA ランタイムライブラリによる CPU から GPU への転送を行う。

従来の複数ノードの CPU 計算の場合と比較すると、GPU の演算性能が高いため GPU 内での計算時間が短い。GPU 間のデータ通信の時間が無視できず、大規模計算では大きなオーバーヘッドになる。GPU 間のデータ通信の時間を GPU 計算と如何にオーバーラップすること

で隠ぺいするかが非常に重要となる。ここでは、(a)GPU-only method、(b) Hybrid-YZ method、(c) Hybrid-Y method の3つの実装を行う。

(a)GPU-only method:比較のために計算と通信のオーバーラップを行わず、上記の通信の3ステップを順に行う。

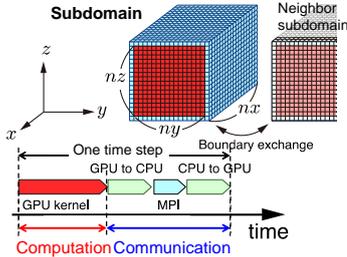


図4 GPU-only method のダイアグラム

(b) Hybrid-YZ method: 1つのGPUが担当するサブ領域を、y方向の境界領域、z方向の境界領域、残りの中心領域に分割する。y、z方向の境界領域をCPUで計算し、中心領域をGPUで計算する。先行研究²⁾では境界も中心領域もGPUで計算したが、ここでは境界領域をCPUで計算しGPU関数を分割することなく通信を隠蔽する。ただし、全計算領域サイズによってCPUによる通信と境界領域の計算にかかる時間が中心領域のGPU計算時間よりも長いことがあり、CPUがボトルネックとなることが考えられる。

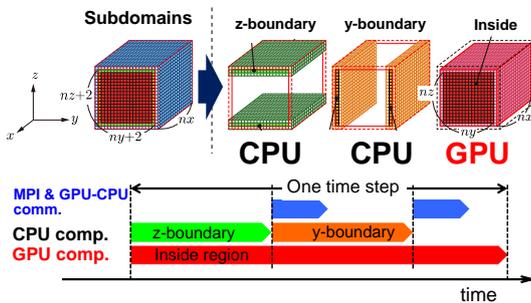


図5 Hybrid-YZ method のダイアグラム

(c) Hybrid-Y method: (b)と同じようにCPU計算も利用するが、z方向の境界領域(x-y断面)はカーネル関数を分割してGPUで計算する。境界領域を担当するGPUカ

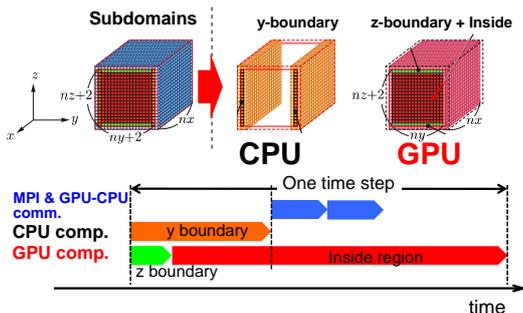


図6 Hybrid-Y method のダイアグラム

ーネル関数は、グローバルメモリの連続アクセスであるた

め計算効率がよく、また通信バッファへデータを詰め替える必要もない。

TSUBAME 2.0の各ノードには3GPUと2CPU sockets (12 CPU cores) が搭載されている。このためCPUによる境界領域の計算に1GPUあたり4CPU coresを割り当て、OpenMPを用いた並列計算をおこなっている。

5. 大規模GPU計算の実行性能

東京工業大学・学術国際情報センターのGPUスパコンであるTSUBAME 2.0の複数GPU(M2050)を用い(a)GPU-only method、(b) Hybrid-YZ method、(c) Hybrid-Y methodの3つの実装の実行性能について述べる。ここでは、Al-Si合金の凝固成長の計算を行った。図7(a)は、大阪大学の安田秀幸教授らが大型放射光施設(SPring-8)を用いて撮影した合金の凝固過程の画像である。図7(b)は、本研究のフェーズフィールド法により、同じような2次元的な形状で、 $4096 \times 128 \times 4096$ 格子を使ってシミュレーションを行った結果である。合金系は異なるが、成長過程が非常によく一致していることが分かる。

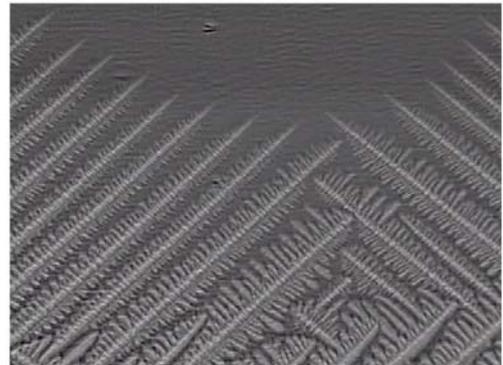


図7(a) SPring-8を用いて撮影された合金の凝固過程の実験画像(安田教授のご好意による)

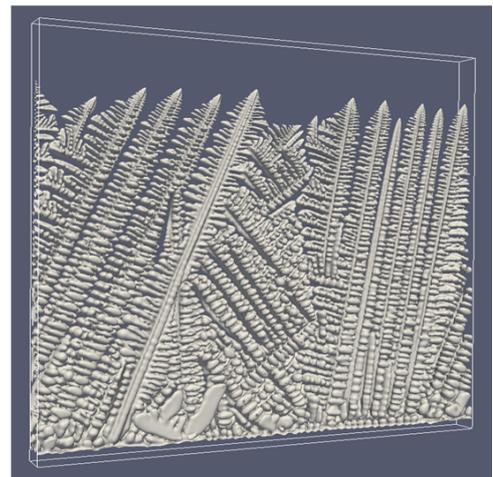


図7(b) フェーズフィールド法を用いてGPUで計算した凝固成長

GPUコードの実行性能の評価において、GPUコードの浮動小数点演算数を直接測定することは困難である。ここでは全く同じ結果を出すCPUコードに対してPAPI

(Performance APD)を用いて実測した浮動小数点演算数を元に GPU の実行時間から実行性能を評価している。図 8 の強スケーリングは、問題を固定して GPU 数を増やして行くときの実行性能の向上を示している。

前述の(a), (b), (c) の 3 つの実装の違いによる特徴が表れている。全体の計算領域を 512^3 , 1024^3 , 2048^3 とした。図 5, 図 6 のように計算と通信のオーバーラップを導入した(b) Hybrid-YZ method, (c) Hybrid-Y method では GPU 数が少ないときには通信を隠蔽できている。GPU 数の増加とともに GPU の計算時間が短くなるため、ある段階で計算時間より通信時間の方が長くなり、もはや通信を隠ぺいすることができなくなる。(c) Hybrid-Y method では、GPU 数を増やした時に CPU の計算時間が隠ぺいの足を引っ張る部分が大幅に改善され、広い範囲の GPU 数で最適化を導入していない(a) GPU-only method と比較して実行性能が大幅に改善されている。

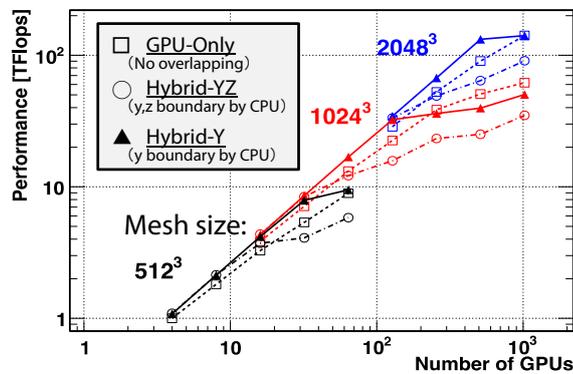


図 8 複数 GPU 計算の実行性能(強スケーリング)。

弱スケーリングは 1GPU あたりで実行する問題規模を一定にし、GPU 数の増加とともに問題サイズも大きくする性能評価である。この測定では、1GPU あたりの計算格子サイズを単精度計算では $4096 \times 160 \times 128$ 、倍精度計算では $4096 \times 160 \times 64$ とした。図 9 のように非常に良い弱スケーリングの結果が得られ、(c) Hybrid-Y method では、4000 GPU を利用して $4096 \times 6480 \times 13000$ 格子に対して行った計算で、2.0 PFLOPS (GPU:1.975 PFLOPS, CPU: 0.025 PFLOPS) という極めて高い実行性能を達成した。これまでにステンシル計算の実アプリケーションとして PFLOPS を超えた計算例は報告されていない。この成果に対し、2011 年の ACM ゴードンベル賞・特別賞(本賞)が与えられた。

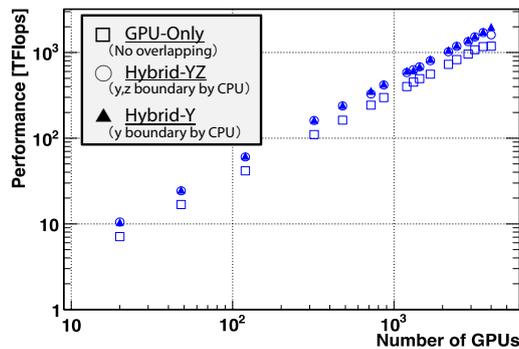


図 9 複数 GPU 計算の弱スケーリング。

している。4000 GPU と 16000 CPU を使って計算し、2.0 PFLOPS (ピーク性能の 44.5%) の実行性能を達成したときの使用した計算ノードとネットワークの消費電力は約 1.36 MW であった(図 10)。電力性能は 1468 MFlops/Watt であり、少ないエネルギー消費で目的とする計算結果が得られたことになる。

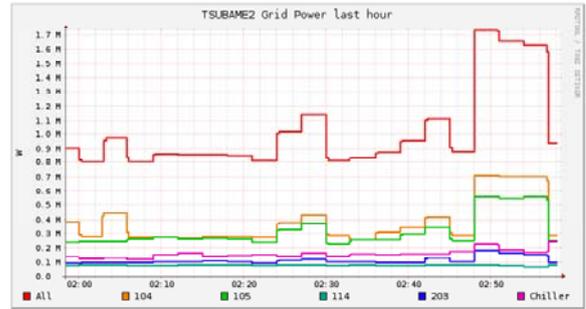


図 10 TSUBAME 2.0 の消費電力モニター。

樹枝状凝固の 3 次元成長を調べるために、 $4096 \times 4096 \times 1024$ 格子での計算を行った。初期に床面に固相のシードを置き、柱状晶が形成される際の相互干渉などによる空間選択のメカニズムを明らかにすることができる。(図 11)

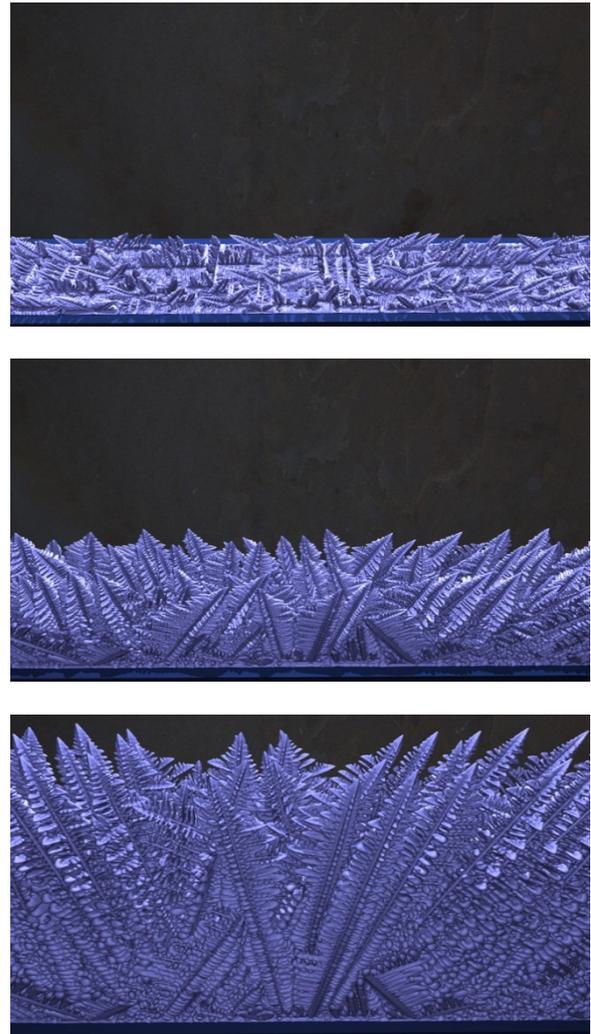


図 11 Al-Si 合金の樹枝状凝固過程の大規模シミュレーション

TSUBAME 2.0 では消費電力を詳細にモニタリング

6. おわりに

フェーズフィールド法により二元合金の樹枝状凝固成長のプロセスを GPU スパコンである TSUBAME 2.0 を用いて大規模計算を行い、格子法に基づいたステンシル計算でありながら単精度で 2.0 PFLOPS という極めて高い実行性能を達成した。これはピーク性能に対して 44.5% であり、GPU スパコンが実用アプリケーションに対して十分有効であることを示すことができた。同じような計算でも GPU スパコンを使うことにより大規模計算が可能になり、さまざまな分野での研究の発展や開発の進展が期待できる。

謝辞

本研究の一部は科学研究費補助金・基盤研究(B) 課題番号 23360046「GPU スパコンによる気液二相流と物体の相互作用の超大規模シミュレーション」、科学技術振興機構 CREST「次世代テクノロジーのモデル化・最適化による低消費電力ハイパフォーマンス」および「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」、日本学術振興会(JSPS) グローバル COE プログラム「計算世界観の深化と展開」(CompView) から支援を頂いた。記して謝意を表す。

参考文献

- [1] R. Kobayashi: Modeling and numerical simulations of dendritic crystal growth. *Physica D, Nonlinear Phenomena*, 63(3-4), 410 - 423 (1993)
- [2] 小川慧, 青木尊之, 山中晃徳: マルチGPUによるフェーズフィールド相転移計算のスケラビリティ — 40GPUで5 TFLOPSの実効性能, *情報処理学会論文誌コンピューティングシステム* Vol. 3 No. 2 67-75 (2010 June)
- [3] A. Yamanaka, T. Aoki, S. Ogawa, and T. Takaki: GPU-accelerated phase-field simulation of dendritic solidification in a binary alloy. *Journal of Crystal Growth*, 318(1):40 - 45 (2011). The 16th International Conference on Crystal Growth (ICCG16)/The 14th International Conference on Vapor Growth and Epitaxy (ICVGE14)
- [4] T. Shimokawabe, T. Aoki, T. Takaki, A. Yamanaka, A. Nukada, T. Endo, N., Maruyama, S. Matsuoka: Peta-scale Phase-Field Simulation for Dendritic Solidification on the TSUBAME 2.0 Supercomputer, in *Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC'11*, IEEE Computer Society, Seattle, WA, USA, Nov. 2011.