

**付録 C++による非常に簡単な科学技術プログラミング**  
**－実行環境の設定、実行方法、およびプログラムのダウンロードについて－**

**小山敏幸**

名古屋工業大学 大学院工学研究科(しくみ領域)

物質工学専攻(物性分野)

工学部 環境材料工学科(材料機能系プログラム)

〒466-8555 名古屋市昭和区御器所町

TEL:052-735-5124, E-mail:koyama.toshiyuki@nitech.ac.jp

URL: [http://tkoyama.web.nitech.ac.jp/Phase-Field\\_Modeling.htm](http://tkoyama.web.nitech.ac.jp/Phase-Field_Modeling.htm)

## 1. はじめに

本解説は通常の C++ の解説ではない点を、あらかじめお断りしておく。従来、C++ 関連の書籍では、文字列を "Hello World" と表示するところから始まり、クラス概念、インスタンスの生成、継承、・・・等、オブジェクト指向のプログラミングの考え方が展開される場合が多い。C++ を用いてソフトウェア（ごく簡単なものも含めて）を作成することを目的とするならば、このような教科書は有用である。しかし、技術者・研究者で、ソフトウェア作成ではなく、単に数値計算がしたいだけであるならば、上記のテキストの大部分は必要のない知識である。なぜならば、通常、技術者・研究者が計算において必要とする操作の大部分は、以下の 3 種類のみであるからである。

- (1) 手続き的な数値の計算      (2) 結果の可視化（グラフや画像）      (3) 数値データの保存と読み出し

きちんとしたソフトウェア作成自体を目的としないければ、プログラムにおいて、ボタン等のコンポーネントの作成や過度の例外処理などは必要ない。単にソースプログラムが書けて、それをコンパイルし、上記 3 つの操作に対して、プログラムを実行し、結果さえ得られれば良い（計算結果の数値データさえ保存できれば、それを綺麗にグラフ化・可視化するソフトは巷にあふれている）。プログラムに入力値が必要な場合には、極論すれば、ソースコードに毎回直接書き込み、その都度コンパイルして実行しても、ほとんどの場合、日常必要とする数値計算に支障は起きない。さらにプログラムも、実行するたびにその都度強制終了するものであってもかまわないのである。

C++ に限った話ではないが、基本的に通常の Windows ソフトウェアを作成することを目的としてしまうと、イベントドリブン型の部分、すなわち、ボタンやツールバー、テキストボックス、クリック、ドラッグ、・・・など、本来の数値計算部分とは全く無関係な部分を作りこまなくてはならなくなるために、Windows プログラミングは非常に複雑になる（コンポーネントウェアの概念にてプログラミングの手間はかなり削減されているが、根本的に数値計算以外の部分が多く煩雑である点にかわりはない）。数値計算をしたいだけであるならば、ボタンやツールバー、クリック、ドラッグなどのイベントドリブン型の部分の知識は全く必要ないので、この部分の学習は後回しでかまわない（後々、きちんとした Windows ソフトウェアを作成したくなった時にあらためて勉強すれば良い）。

もう 1 つ最近のパソコンの高性能化をあらためて考えてみていただきたい。現在のパソコンは、10 年前の通常的大型計算機並みの能力がある。10 年前、数千万円単位であったマシンが、現在は数万円で個人的で購入できるのである（実験機器でたとえるならば、10 年前の最新鋭の電子顕微鏡が、現在、個人で買えるようになったようなものである）。これを研究・教育に使わないのは、明らかにもったいない。もちろん、既存のソフトウェアは豊富であり、文書作成や研究発表などでパソコンは大いに活用されている。しかし、みずからソースコードを書いて数値計算する用途には、あまり使用されていないのではないだろうか。本当は誰でも簡単にプログラミングして数値計算できるのである。

そこで、本解説では、上記の 3 つの操作に限定して、まず非常に簡単に C++ プログラム（基本的には C プログラム）を作成する手法を説明し、その後、フェーズフィールドシミュレーションに進む。具体的には以下の内容を取り上げる。

- (1) 自由エネルギーを例題とした数値計算、グラフの描画、およびデータの入出力
- (2) 拡散相分離のフェーズフィールドシミュレーションプログラム
- (3) マルテンサイト変態のフェーズフィールドシミュレーションプログラム

相変態・組織形成では、濃度プロファイル等のグラフだけでなく、二次元計算（三次元計算）における組織形態変化も計算対象とするので、画像・動画としての面白さも満喫していただけると嬉しい限りである。なお本稿では、C++ 言語自体についての説明は省略させていただく。C 言語については文献 (1-4)、C++ 言語に関しては文献 (5-7) などを適宜参照していただきたい。

## 2. プログラムの計算環境について

使用する計算機的环境は、

- (1) OS : Ms-Windows 2000, XP, Vista, 7
- (2) コンパイラ : Ms Visual C++ 6.0 以上
- (3) 参考文献(4)のインクルードファイル: wingxa.h

である。なお文献(4)の付録を見ると、MacおよびLinuxでも対応可能となっているので、MacおよびLinux上でも、同じプログラムが実行可能と思われる（著者は確認しておりません）。

Windowsプログラミングは、従来のBASICやFORTRANのように手続型のプログラミングではないので、実際に簡単なプログラムを作成して実行してみようとする場合でもプログラム作成には非常な困難を伴う場合が多い。最近ではVisual Basic等のコンポーネントウェアを中心に、Windowsプログラミングも比較的簡単に行えるようになったが、計算速度を必要とするシミュレーションのプログラム開発には不向きである。なんとかWindows上でこれまでの手続型のプログラムが実行できないかと模索した結果が、上記の文献(4)の活用である。これはプログラミング環境として非常に強力であり、CPUパワー全てを計算部分につぎ込み、かつ従来の手続型としてのプログラミングが可能である。言語はC++であるので、計算速度を確保しつつ、ビットマップ等の画像まで自在に扱うことが出来る。このようなことが可能であるキーポイントは、OSのWindowsとやり取りする部分を全てソースレベルで部品化し、includeファイルとしてリンク出来るようにしている点である。したがって、プログラミングでは、includeするファイル（Windowsとやり取りする部分）をヘッダに一つ追加するだけで、後はBASICやFORTRANの時と同様に、通常のソースプログラムを作成していく手順でWindowsプログラムを作成できる。文献(4)において、このincludeするファイルが提供されている（これは有償ですので、文献(4)を購入する必要があります）。実行ファイルの作成には、当然ながらコンパイラが必要で、著者はVisual C++を使用している。現在、Visual C++のコンパイル環境は、無償にて提供されている（コンパイル環境の具体的な設定方法、およびコンパイルの仕方等は、文献(4)の付録をご参照いただきたい）。

### 3. 各種ソースコードについて

プログラムのソースコードは¥C 内において、以下のように分類されている。

- (1) 自由エネルギーを例題とした数値計算、クラフの描画、およびデータの入出力

¥examples¥Gc.cpp マルテンサイト変態の自由エネルギー曲線の描画、データ入出力

- (2) 拡散相分離シミュレーションのプログラム

¥Spinodal\_1D¥phase\_decomp\_1D.cpp 相分離における 1 次元濃度プロファイルの時間発展

¥Spinodal\_1D¥phase\_decomp\_1D\_save.cpp 相分離における 1 次元濃度プロファイルの時間発展（データ保存）

¥Spinodal\_1D¥plot1D\_c.cpp 計算結果の表示

¥Spinodal\_1D¥select\_1D\_c\_dat.cpp 計算結果のファイルからデータの抽出

¥Spinodal\_2D¥phase\_decomp\_2D.cpp 相分離における 2 次元濃度場の時間発展

¥Spinodal\_2D¥phase\_decomp\_2D\_save.cpp 相分離における 2 次元濃度場の時間発展（データ保存）

¥Spinodal\_2D¥plot\_c.cpp 計算結果の表示

¥Spinodal\_2D¥bplot\_c.cpp 計算結果のイメージデータを保存

¥Spinodal\_2D(Fe-Cr)¥fecr2d\_demo.cpp 相分離における 2 次元濃度場の時間発展

¥Spinodal\_2D(Fe-Cr)¥fecr2d\_demo\_save.cpp 相分離における 2 次元濃度場の時間発展（データ保存）

¥Spinodal\_2D(Fe-Cr)¥fecr3d\_mag.cpp 相分離における 3 次元濃度場の時間発展

¥Spinodal\_2D(Fe-Cr)¥fecr3d\_mag\_save.cpp 相分離における 3 次元濃度場の時間発展（データ保存）

¥Spinodal\_2D(Fe-Cr)¥FeCrtest3D.fld AVS の fld データ表示設定ファイル（データは FeCrtest3D.dat）

- (3) マルテンサイト変態のシミュレーションプログラム（Java 版とは内容が異なる点に注意）

¥Martensite¥martensite.cpp マルテンサイト変態における組織形成の時間発展

¥Martensite¥martensite\_save.cpp マルテンサイト変態における組織形成の時間発展（データ保存）

¥Martensite¥martensite\_input.cpp 外力下でのマルテンサイト変態の時間発展（データ入力）

¥Martensite¥martensite\_input\_save1.cpp 外力下でのマルテンサイト変態の時間発展（データ 1 入力・保存）

¥Martensite¥martensite\_input\_save2.cpp 外力下でのマルテンサイト変態の時間発展（データ 2 入力・保存）

¥Martensite¥plot\_s12.cpp 計算結果の表示

¥Martensite¥bplot\_s12.cpp 計算結果のイメージデータを保存

C++では、ソースプログラムにおいて、ダブルスラッシュ"/"以降の 1 行がコメント文として扱われるので、ソースプログラム内にコメント文として、プログラムの内容を直接書き込んで説明している。なお上記のプログラムを終了するには、表示画面右上のxをクリックして強制終了させる（技術計算に必要な Windows ソフトの余分な操作は一切省略しているので注意）。

### 4. おわりに

ここで説明したプログラムは一例題に過ぎないが、これを修正・改良することによって、様々な技術系の数値計算に対応したプログラムを自由に作成することができる。また本プログラムは各種の実験データの解析な

どにも役立つ。著者の経験から、Windows プログラミングを勉強するのには、少し障壁があるように思う。本稿でその障壁を幾分でも下げることができたならば嬉しいかぎりである。なお本プログラム等に関して不具合・トラブルが発生しても、著者および(株)内田老鶴圃は責任を負いませんのでご了承ください。

## 参考文献

- (1) コンピュータ教育工学研究所 編, 「ガイドブック C」, サイエンス社, (1994).
- (2) 小野塚純夫, 「Cプログラミングガイド」, コロナ社, (1990).
- (3) 柴田望洋, 「新版 明解C言語 入門編」, ソフトバンククリエイティブ, (2004).
- (4) 安居院 猛, 関根証明, 佐野元昭, 「WindowsですぐにできるC言語グラフィックス」, 昭晃堂, (2009)
- (5) 林晴比古, 「新C++言語入門 ビギナー編」, SOFT BANK, (2001).
- (6) 林晴比古, 「新C++言語入門 シニア編 上 基本機能」, SOFT BANK, (2001).
- (7) 林晴比古, 「新C++言語入門 シニア編 下 クラス機能」, SOFT BANK, (2001).

## 参考：変位型変態のシミュレーション

以下は、本書「材料設計計算工学（計算組織学編）」からパラメータセットを以下のように変えた結果である（基本的な計算理論は同じである）。上記の C 言語によるプログラムでは、以下に関するプログラムが収められている。

$$\Delta G_c = 1000[\text{J/mol}], \quad L = 250[\text{nm}], \quad T = 500[\text{K}], \quad a = 10, \quad \kappa_s = 5.0 \times 10^{-15}[\text{J} \cdot \text{m}^2/\text{mol}]$$

$$\mu = 1.235 \times 10^{11}[\text{Pa}], \quad \lambda = 1.500 \times 10^{11}[\text{Pa}], \quad \eta_1 = 0.08, \quad \eta_2 = -0.04, \quad \sigma_{11}^A = 0, -1[\text{GPa}] \quad a_0 = 0.3563[\text{nm}]$$

図 1 は 500K における等温マルテンサイト変態の二次元シミュレーションで、 $t$ が無次元化した時間である。青は立方晶を、赤と緑がそれぞれ正方晶でバリエーションの異なるドメイン  $s_1$  とドメイン  $s_2$  を表している。正方晶の  $c$  軸は赤が横方向で、緑が縦方向である。また正方晶比は 1 よりも大きい（ $c$  軸方向に伸びた単位胞）。図 1 は(001)面で、横方向が[100]および縦が[010]方向である。相分解の初期状態(a)は立方晶で、乱数を用いて図の中央に正方晶の初期核を水平に置いた。初期において、約  $45^\circ$  の方向に傾いた赤と緑のペアが形成され(b)、時間の進行に伴いこの赤緑のペアが槍状に成長していく(c,d)。赤緑ペアの界面は双晶界面である。図 1 では右  $45^\circ$  に傾いた赤緑ペアの分率多く、左  $45^\circ$  のペアが消滅していく(e,f)。後期において右  $45^\circ$  方向に傾いたラメラ組織となり、細い赤ドメインが徐々に消滅していくことがわかる(g,h)。

図 2 と図 3 は、上下方向に 1GPa の圧縮応力をかけた状態での組織変化で、それぞれ初期組織として図 1(h)および図 1(b)を用いた場合に対応している。圧縮応力(1GPa)を上下方向に作用させているので、 $c$  軸が横方向に伸びた単位胞を持つ赤ドメインがエネルギー的に安定となる。図 2 と図 3 の(a)-(h)の時間はそれぞれ等しい。図 2 では確かに赤ドメインが成長する部分が観察されるが、組織全体の形態変化は大きくはない。一方、図 3 では、赤ドメインの方が、緑ドメインのよりも優先的に成長し、組織はいち早く赤の単一ドメインとなる。図 2 はマルテンサイト変態が完了した組織に応力を作用させた時の双晶ドメイン組織の変化過程で、図 3 はマルテンサイト変態の核形成初期の組織に外力を作用させた時のドメイン形成である。図 2 と図 3 の比較から、マルテンサイト変態初期の核形成段階に、外部応力を作用させることが、単一ドメイン形成に効果的に働くことが理解できる。

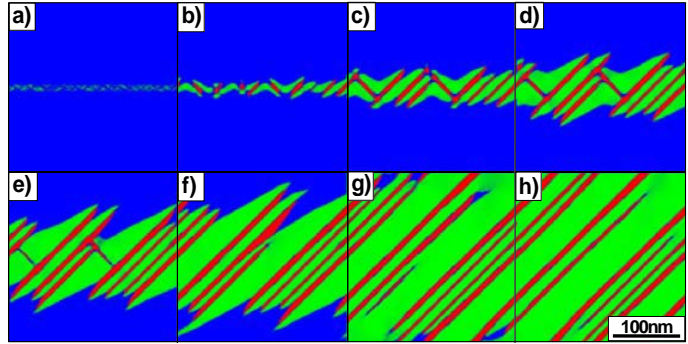


図 1 c→t 構造相転移の二次元シミュレーション  
a)  $t^*=0$ , b)  $t^*=20$ , c)  $t^*=40$ , d)  $t^*=60$ , e)  $t^*=80$ , f)  $t^*=100$ , g)  $t^*=140$ , h)  $t^*=200$ .

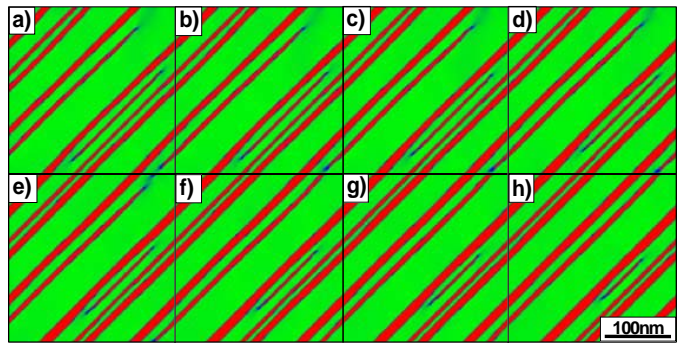


図 2 外力下における組織変化（初期組織：図 1(h)の場合）  
a)  $t^*=0$ , b)  $t^*=10$ , c)  $t^*=20$ , d)  $t^*=30$ , e)  $t^*=40$ , f)  $t^*=50$ , g)  $t^*=60$ , h)  $t^*=70$ .

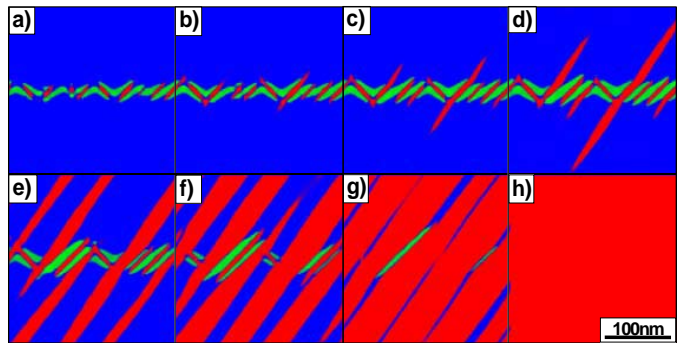


図 3 外力下における組織変化（初期組織：図 1(b)の場合）  
a)  $t^*=0$ , b)  $t^*=10$ , c)  $t^*=20$ , d)  $t^*=30$ , e)  $t^*=40$ , f)  $t^*=50$ , g)  $t^*=60$ , h)  $t^*=70$ .