

# OCTA

ソフトマテリアルのための統合化シミュレータ

概要

OCTA ユーザーズグループ

DEC. 25 2002

## 執筆者

第1章 土井正男

第2章 佐々木誠

付録 佐々木誠

## 謝辞

本プログラム開発は、経済産業省の出資・補助を受け、新エネルギー・産業技術総合開発機構 (NEDO) が (財) 化学技術戦略推進機構に委託した、大学連携型産業科学技術研究開発プロジェクト「高機能材料設計プラットフォーム」通称「土井プロジェクト」の下で行われたものである。

Copyright ©2000-2002 OCTA Licensing Committee All right s reserved.

# 目次

第 1 章	OCTA システムの目的	1
1.0.1	シームレスズームング	1
1.1	OCTA システムの設計思想	2
1.1.1	シミュレーション・エンジン	2
1.1.2	ユーザーインターフェイス	2
1.1.3	ユーザー定義型データ形式 (User definable format)	3
1.1.4	私たちが目指すもの	4
1.2	なぜ “OCTA” か	4
第 2 章	OCTA システムの構成と利用ガイド	5
2.1	ソフトウェア	5
2.1.1	エンジン	5
2.1.2	UDF 入出力インターフェイスライブラリー	5
2.1.3	ユーザーインターフェイス – GOURMET	6
2.2	ドキュメント	6
付 録 A	用語集	9



# 概要

OCTA は経済産業省の援助のもとに行なわれた「高機能材料設計プラットフォームの開発プロジェクト」の中で名古屋大学で 1998 年から 2002 年にかけて開発されたシステムです。この冊子ではプロジェクトの目的と、その目的を達成するために我々が取った戦略について説明します。また OCTA で何ができるのか、そして、以下のマニュアルの概要についても説明します。



# 第1章 OCTA システムの目的

OCTA は日本の産官学の共同研究プロジェクトとして開発されました。プロジェクトの正式な名称は「高機能材料設計プラットフォームの研究開発」です。このプロジェクトは経済産業省によって 1997 年に提案され、産業技術開発制度に基づいて、NEDO への委託事業として名古屋大学内において 1998 年から 2002 年までの間、11 の企業の参加により行なわれました。

このプロジェクトの目的は、材料のミクロ構造 (または分子構造) とマクロな特性とを結びつけ、材料開発に役に立つような計算機シミュレーションシステムをつくることです。重点を置いた物質はソフトマテリアルであり、それは一般に高分子、コロイド、界面活性剤やゲルなどを指しています。

物質の性質がそのミクロな構造とどのように関連しているのかを理解することは物理、化学、物質科学の中心的課題です。また、ミクロな階層とマクロな階層の関係を理解することは企業における既存材料の改良、新規材料開発などの研究開発においても決定的に重要な問題です。

わたしたちの目標はこのような研究開発に資するシミュレーションシステムをつくることでした。計算機シミュレーションは建築、機械、電子デバイスの設計において、なくてはならないほど有用なものになってきています。わたしたちは材料工学においても同様に有用なシミュレーションツールを作りたいと考えました。この考え方は新しいものではありません。「材料工学における計算機支援技術」(“Computer Aided Material Engineering”) といった言葉は 10 年も前から使われて来ており、いくつかの領域では成功も納めています。しかし、ソフトマテリアルにおいてはこのアイデアは簡単には実現できません。

例えばソフトマテリアルの代表である高分子の性質はそれを構成するモノマーの性質だけで決まるものではありません。分子量、分子量分布、分岐構造、分子鎖の配向の程度、結晶化度、結晶とアモルファスの界面構造などにもよります。この状況は高分子混合材料や微粒子分散材料では、さらに複雑となります。これらの材料では、分散状態や相界面の状態によって物性が大きく変わります。

このような複雑な問題は単独のシミュレータで扱うことはできません。問題となっている現象にはたくさんの長さのスケールが含まれ、またたくさんの物理過程が働いているからです。このようなタイプの問題は複数スケール、複数物理過程の問題と呼ばれており、それを扱うシステムを構築することは、現在の計算科学や計算工学に課せられた挑戦的課題となっています。

## 1.0.1 シームレスズームリング

理想的に言えば、高分子のような複雑な物質を研究するのに必要システムは、どのような長さスケールにでも「ズームイン」して物質に起こっている現象を観察し検証できるシステムです。私たちはこれを「シームレスズームリング」を行なうシステムと呼びます。これは、私たちが考える物質シミュレーションの最終目標をあらわした言葉です。

このようなシミュレータはどうしたら実現できるでしょうか。学問的にはマルチスケールのモデル化がいくつか提案されてきました。たとえば 2 つもしくはそれ以上の物理モデルを同時に使用するようなシミュレータの提案が行われています。これらのアイデアは魅力的なものではありませんが、われわれはプロジェクトの活動においてそのような研究を優先してはきませんでした。それには 2 つの理由があります。第一にはマルチスケールのモデリングはあまりにも野心的かつ実験的な研究テーマであり、われわれのような大規模プロジェクトの中心的課題にするには危険が大きすぎたことです。第二には現在のマルチスケールモデルでは対象を特定のシステムに限定することを余儀なくされることです。2 番目の理由は私たちのプロジェクトにとっては受け入れがたいことでした。私たちに汎用的に使えるツールが必要だったのです。

このプロジェクトではソフトマテリアルに対して現在のレベルのシミュレーション技術に有用なソフトウエ

アを作ることに限定することにしました。ソフトマテリアルの特徴として適切な長さのスケールがメソスコピック領域であるということがあります。それは原子レベル ( $1\text{ nm}$  以下) ではなく、マクロスコピック ( $1\text{ mm}$  以上) でもなく、それらの中間領域です ( $1\text{ nm}$  から  $1\text{ mm}$  の間)。ソフトマテリアルの理論ではメソスコピックなモデリングが行なわれ、また学問的には多数のシミュレーションがそれらをもとにして行なわれてきました。しかし、これらのシミュレータは研究者個人の利用のために作成されたものであり、他の人には利用し難いものです。結果として同じようなシミュレータが世界中のたくさんの研究室で繰り返し繰り返し作られ、そして 2 度と使われずにいました。私たちはこのような状況を変えたかったのです。

## 1.1 OCTA システムの設計思想

### 1.1.1 シミュレーション・エンジン

私たちのプロジェクトでは 2 つの目的を設定しました。ひとつは、ソフトマテリアルの研究において良く使われるであろうシミュレーションプログラムを作り上げることです。私たちはそのようなシミュレーションプログラムを「エンジン」と呼ぶことにしました。それはこれらの役割が利用者の要求にしたがって何度でも繰り返し計算をおこなうことにあるからです。

ソフトマテリアルのシミュレータとしてたくさんの種類のエンジンが必要であるのは明らかですが、私たちは 4 つのエンジンを構築することにしました。それらはそれぞれ分子動力学、レプテーションダイナミクス、Edwards の自己無撞着場理論、連続体力学に基づいています。

**COGNAC** このエンジンは様々な外場 (流れや変形) のもとでの高分子の分子動力学計算を行ないます。フル・アトミスティックモデルからビーズ・スプリングモデルのような粗視化されたものまで様々なモデルを扱うことができます。

**PASTA** からみあった高分子のレオロジー的応答をレプテーションモデルとからみあいのデュアル・スリップリックモデルで計算します。

**SUSHI** Edwards の自己無撞着場方程式を解くことによって、高分子の相分離や表面の効果によって生ずる高分子の構造を計算します。

**MUFFIN** これはソフトマテリアルの様々な問題に現れる偏微分方程式に対する汎用的なソルバーエンジンです。実際には 5 つのシミュレーションプログラムで構成されます。(1) 相分離シミュレータ、(2) 電解質シミュレータ、(3) 電場化での微小領域流体シミュレータ (micro-fluidics)、(4) 固体の弾性変形シミュレータ、そして (5) ゲルの変形と膨潤のシミュレータ。

(これらのエンジンの命名法は、プロジェクトで行われたある夜のピア・パーティで決められたルールに従っています。そのルールとはエンジンの名前は、何がしかのパーティのテーブルの上に見出されるものでなくてはならないというものです。)

各エンジンの詳しい内容は後述のマニュアルに記述されています。

### 1.1.2 ユーザーインターフェイス

上記のエンジン群を統合するために、エンジンが共同して動くためのプラットフォームが必要でした。設計上での重要な問題は異なるエンジンを共同して動かすにはどうするかということでした。この問題は非常に難しいものであることが分かってきました。最初の問題はどうやってエンジン間で情報を共有させるかということでしたが、これからして既にかなり難しいことでした。

私たちは最初物理量を表現する共通の枠組みを作ろうとしました。たとえば高分子であれば「モノマー」「ポリマー」「分子量」などといった概念を表現する枠組みを作ろうとしました。しかしこのアプローチは難しいものであることがすぐにわかりました。各エンジンはそれぞれ異なる物理モデルに基づいていて、異なる観点で「ポリマー」を定義しています。用語を調整するのは困難なことであり、データ構造を確定するのはさらに困難



なことでした。いくたびか真剣な議論を重ねたのち、例え、何らかの共通データ形式をつくり出すことができて、それは誰にとってもハッピーではなく、誰もそれを使わないだろうという結論に達しました。

私たちは共通のデータ形式を定義するというアイデアを放棄しました。私たちが学んだ教訓は統合は強制的になされるべきものではなく、人々の自発的協力でなされるべきものだということでした。私たちはどうしたらそのような協力が可能になるかを考えはじめました。

いろいろ考えた挙句私たちが最終的に到達した結論は、すべてのエンジンで使うことのできる共通のグラフィックなユーザーインターフェイスを作ることでした。これは、平凡な結論ですが、私たちはその社会的意味を考えたのでした。もしも全てのエンジンが同じユーザーインターフェイスをもつならば、ユーザは様々なエンジンを容易に使いこなすことができますはずです。もしそのユーザーインターフェイスに何らかのプログラミング機能があれば、ユーザはひとつのエンジンのデータ形式を別のエンジンのそれに変えることができ、自分の手でズームングを行なうことができます。結果としてこれが「シームレスズームング」に現実味を持たせることになるでしょう。共通のグラフィックなユーザーインターフェイスを作ることが、シームレスズームングという遠大な目標のために私たちが現時点でできるベストな選択であるというのが、私たちの結論でした。私たちはそのような汎用的ユーザーインターフェイスを「プラットフォーム」と呼ぶことにしました。この目的のもとに GOURMET(Graphical Open User interface for Material design Environment) が開発されました。

GOURMET は次節で説明する User Definable Format (UDF) という形式でかかれたテキストファイルを読んで、さまざまなサービスを提供します。テキストファイルが UDF で書かれているならば、ユーザはそのデータを様々な形式で見て編集することができます。また、データをスクリプト言語で処理し、プロットしたり 3 次元のアニメーションで表示することができます。GOURMET はまたエンジンをコントロールする機能も持っています。ユーザはエンジンのあるパラメータをモニターすることができ、エンジンを停止してパラメータの値を変え計算を再開することもできます。

### 1.1.3 ユーザー定義型データ形式 (User definable format)

UDF の基本的アイデアはファイル中のすべてのデータに名前をつけるということです。UDF ファイルはデータ定義部とデータ部の二つの部分から出来ています。データ定義部はファイル中のデータに名前をつけるのに必要な部分です。UDF ファイルが GOURMET で読み込まれると、ファイル内の全ての数値やテキストに名前が与えられます。ユーザはデータを、それらの名前で引用することができます。この機能はデータ解析やデータ処理を行なうのに便利です、というのは、ファイル中の全てのデータはそれらの名前で引用でき、GOURMET に組み込まれたスクリプト言語で処理することができるからです。

エンジンをプログラムする人はプラットフォームライブラリを用いて UDF ファイルの中のデータにそれらの名前アクセスすることができます。いったん UDF ファイルがオープンされればエンジンプログラムはどのような順序でもデータの読み書きを行なうことができます。

データ定義部ではエンジンをプログラムする人は自分のプログラムで使う単位系を定義することができます。またそれを UDF ファイル中の各データに与えることができます。この機能を使うことは、エンジンをプログラムする人に対して強制されているわけではありませんが、我々は、エンジンプログラムに対して単位系を明示すること、各データに単位をつけることをお願いしています。単位は物理量にとって基本的な情報であり、異なるファイルの間でデータ変換を行なうときに絶対に必要とされるものだからです。

UDF ファイルではデータの名前それぞれにコメントを付与することができます。そのコメントにはプログラムにおけるデータの意味や役割を記述することが出来ます。コメントは GOURMET によって UDF ファイルを開いたユーザを手助けするのに使われます。将来においてはこの機能はさらに強化され、例えばそのデータに対してさらに情報を提供する URL にジャンプするといったことに使われるでしょう。

GOURMET のユーザは UDF のデータに関連付けられた手続きをデータに与えることができます。例えば一連の数値データに対して、それらの平均を計算する手続きを与え、それを起動することが出来ます。この機能は入力データを用意したり、出力データを解析したりすることに使えます。

#### 1.1.4 私たちが目指すもの

以上の記述でわかるように、私たちは究極のゴールである「シームレスズーミング」に対して意図的に保守的なアプローチをとってきました。GOURMET の設計には、特定のズーミングのやり方を強制するような機構はありません。ズーミングを行なうことは、OCTA を利用するすべてのユーザに委ねられています。

計算科学や計算工学の立場からは、ズーミングの手続きを保証するシステムを設計することは非常に挑戦的な課題です。そのような研究は、将来、物理学者、化学者そしてソフトウェアエンジニアの協力によって成し遂げられるべきものでしょう。「シームレスズーミング」という究極の目標に向かう上で現時点で必要とされているのは、たくさんのシミュレーションプログラムの協調を容易にするメカニズムであるというのが私たちがたどり着いた結論であります。現時点でのシームレスズーミングは、人々の協調的な活動を意味していると私たちは考えています。私たちの作り上げたものがそのような協調の基になることを私たちは望んでいます。

シームレスズーミングは大きな挑戦です。それは最終的には達成されるものであると私たちは信じています。私たちが到達した地点はまだこの目標からは遠いものです。私たちの開発したエンジンは完璧なものではありません。それらはさらに改良されるべきものですし、新しいエンジンも加えなければなりません。インターフェイスにはブラッシュアップする必要があります。そのためにすべてのソースコードを開示し利用者による改良を可能なものにしました。

### 1.2 なぜ“OCTA”か

OCTA はギリシャ語の“8”であり、数学記号の無限大を 90 度回転したものでもあります。それはマイクロコスモスから宇宙までの長さスケールにわたって様々なシミュレーターの協調でもたらされる無限の可能性を表すものです。漢字では OCTA(8) は“八”であり「末広がり」を表すものです。漢字の“八”は山を表していると見ることもできます。気高くそびえ、登山者の挑戦意欲を永遠にかきたてつづける山です。

私たちはこのプロジェクトへのフィードバックを歓迎します。

## 第2章 OCTA システムの構成と利用ガイド

### 2.1 ソフトウェア

OCTA は以下のような構成要素からなっています。

- (1) シミュレーションプログラム (エンジン)
- (2) 共通データ形式とその取扱いのためのライブラリやツール
- (3) ユーザインタフェース

以下これらに関して簡単に解説します。

#### 2.1.1 エンジン

OCTA システムではシミュレーションプログラムを「シミュレータ」または「エンジン」と呼びます。今回配付するエンジンには以下のものがあります。

- 汎用粗視化分子動力学シミュレータ – COGNAC

原子集合体を単位として行う粗視化分子動力学法による分子レベルのダイナミックスシミュレータ。

- レオロジーシミュレータ – PASTA

土井-Edwards の管模型 (レプテーション理論) に基づいて高分子溶融体のレオロジーを予測する分子レベルのダイナミックスシミュレータ。

- 動的平均場法シミュレータ – SUSHI

相分離構造や界面構造などの高分子系に見られるメソスケール構造を動的平均場法によってシミュレートする界面構造シミュレータ。

- 多相構造シミュレータ – MUFFIN

多成分高分子混合系の分散構造を差分法、有限要素法などによってシミュレートする多相系ダイナミックスシミュレータ。

詳しい内容や使用法については各エンジンの機能マニュアルを参照して下さい。

#### 2.1.2 UDF 入出力インターフェイスライブラリー

各エンジンやツール類が OCTA システムの標準入力形式である UDF 形式データの入出力を行うためのインターフェイスライブラリー。

- libplatform C++ 言語で記述されたプログラムから UDF 形式データの入出力を行うためのライブラリー。OCTA のエンジンの入出力部分はすべてこれを使用して作成されている。
- Python UDF Manager オブジェクト指向スクリプト言語 Python によって記述されたスクリプトプログラムによって UDF 形式データの入出力を行うためのライブラリー。

### 2.1.3 ユーザーインターフェイス – GOURMET

GOURMET はグラフィックを伴う対話型ユーザーインターフェイスプログラムで以下のようなことを行うことができます。

- シミュレーションプログラム (エンジン) 入力用の UDF ファイルの編集
- エンジンの起動
- エンジン計算結果 UDF の表示/解析
- Python スクリプトによる UDF 処理

## 2.2 ドキュメント

OCTA では以下のようなドキュメントが提供されています。これらは配付 CD に含まれる PDF (Portable Document Format) データとしても提供されています。

- 概要 (本冊子)  
高機能材料設計プラットフォーム OCTA の目的、特徴、システム構成、及びパッケージ内容が記述してあります。
- インストールマニュアル  
利用者の計算機への OCTA のインストール方法が説明されています。OCTA を導入されるかたはまずこのドキュメントにしたがってシステムをインストールしてください。
- GOURMET Primer – チュートリアル  
GOURMET への入門のためのチュートリアルです。
- グラフィカルユーザーインターフェイス GOURMET 操作マニュアル  
OCTA のユーザーインターフェイス GOURMET による UDF データの編集、3 次元/2 次元グラフィック表示などの解説。
- 汎用粗視化分子動力学法シミュレータ COGNAC ユーザーズマニュアル  
COGNAC エンジンの理論、使用法などの詳細が記述してあります。
- レオロジーシミュレータ PASTA ユーザーズマニュアル  
PASTA エンジンの理論、使用法などの詳細が記述してあります。
- 動的平均場法シミュレータ SUSHI ユーザーズマニュアル  
SUSHI エンジンの理論、使用法などの詳細が記述してあります。
- 多相構造シミュレータ MUFFIN ユーザーズマニュアル  
MUFFIN エンジンの理論、使用法などの詳細が記述してあります。
- GOURMET PYTHON スクリプト – リファレンスマニュアル  
OCTA の特徴である Python 言語での処理の記述法についての解説です。Python 言語の文法にも簡単に触れられています。
- UDF 文法リファレンスマニュアル  
UDF の文法について記述。UDF データの記述法を詳しく知りたいとき、新しいエンジンのための UDF 定義ファイルを作成する時などに参照するとよいでしょう。

- プラットフォームインターフェースライブラリ libplatform – リファレンスマニュアル

UDF 形式データの入出力を行う C++ ためのインターフェースライブラリーの詳細について記述してあります。新たにエンジンや UDF を扱うツールを作成するときに参照します。「UDF チュートリアル」とあわせて使うとよいでしょう。

- 適用研究事例集

OCTA の適用研究事例の解説。

- 適用研究事例 AMUSE

OCTA を利用したシームレスズームングによる仮想実験プロトタイププロジェクト AMUSE の解説。



## 付 録 A 用語集

OCTA システムの解説で使用する主な用語。

### UDF

OCTA システムで計算入出力データ、異なるシミュレーションプログラム (エンジン) とのやりとりに共通して使用されるデータ形式 (User Definable Format)。データそれ自身の中に、データの意味を記述しておく自己記述型のデータ形式である。

### GOURMET

OCTA システムにおける GUI (Graphical User Interface) のプログラム。シミュレーションプログラム (エンジン) 入力用の UDF ファイルの編集、エンジンの起動、エンジン計算結果 UDF の表示/解析、その他 Python による UDF 処理機能などをもつ。

### エンジン

OCTA システムではシミュレーションプログラムを「シミュレータ」または「エンジン」と呼ぶ。

### COGNAC

OCTA に含まれる粗視化分子動力学エンジン。

### SILK

COGNAC エンジンの入力 UDF のために分子構造データを生成する補助ツール。

### PASTA

OCTA に含まれるレオロジー予測シミュレーションエンジン。

### Fork

ReX エンジンの入力 UDF のための補助ツール。

### SUSHI

OCTA に含まれる界面構造シミュレーションエンジン。

### MUFFIN

OCTA に含まれる分散構造シミュレーションエンジン。固体系シミュレータ MUFFIN MSPD と流体系シミュレータ MUFFIN MFPD を含む。

### プラットフォームインターフェイスライブラリ (libplatform)

シミュレーションプログラム (エンジン) や補助ツールなどが UDF 形式のデータを入力、出力するために必要な手続き (C++ 言語、Python) をまとめたライブラリー。

## Python

OCTA システムで使用する「オブジェクト指向型スクリプト言語」。Guio van Rossum 氏によって開発されたフリーソフトウェアである。Microsoft Windows、各種 UNIX 系システムなどで使用することが出来るようになっている。

## Java

Sun Micro Systems によって開発されたオブジェクト指向プログラミング言語。GOURMET システムは Java で記述されている。

## OpenGL

GOURMET が使用している 3 次元グラフィック処理用プログラミングライブラリー。Linux システムではこのライブラリー仕様のフリーな実装である Mesa ライブラリーが使用されている。

## Cygwin

MS Windows 上で UNIX に相当する環境 (POSIX 標準) を利用できるようにしたソフトウェアである。米 Cygnus Solutions 社 (現在は米 Red Hat Corp. に吸収されている) で開発され、無償で提供されている。OCTA のシミュレーションプログラム (エンジン) のいくつかは Cygwin 環境で動作する。

## OCTA

OCTA のこと。