

# **OCTA**

**Integrated simulation system for soft materials**

## **OVERVIEW**

**OCTA User 's Group**

**DEC. 25 2002**

## Authors of the Manual

Chapter 1 Masao Doi  
Chapter 2 Makoto Sasaki  
Appendix Makoto Sasaki

## Acknowledgment

This work is supported by the national project, which has been entrusted to the Japan Chemical Innovation Institute (JCII) by the New Energy and Industrial Technology Development Organization (NEDO) under METI's Program for the Scientific Technology Development for Industries that Creates New Industries.

Copyright ©2000-2002 OCTA Licensing Committee All rights reserved.

# Introduction

OCTA is a system developed by the project "Development of the Platform for Designing High Functional Material (OCTA)" conducted at Nagoya University from 1998 to 2002, supported by Ministry of Economy, Trade and Industry, Japan. This booklet explains the objective of the project, and the design policy we took to achieve the objective. It also outlines what you can do by OCTA, and the contents of the subsequent manuals.



# Contents

<b>1</b>	<b>Objective of the OCTA system</b>	<b>1</b>
1.1	The OCTA project . . . . .	1
1.1.1	Outline of the project . . . . .	1
1.1.2	Seamless zooming . . . . .	1
1.2	Design policy of the OCTA system . . . . .	2
1.2.1	Simulation engine . . . . .	2
1.2.2	User interface . . . . .	2
1.2.3	User definable format . . . . .	3
1.3	What we are aiming at . . . . .	3
1.4	Why “OCTA” . . . . .	3
<b>2</b>	<b>Outline of the OCTA system</b>	<b>5</b>
2.1	Software Components . . . . .	5
2.2	Document list . . . . .	5
<b>A</b>	<b>Glossary</b>	<b>7</b>



# Chapter 1

## Objective of the OCTA system

### 1.1 The OCTA project

#### 1.1.1 Outline of the project

OCTA was developed in the joint project of industries, government and academia carried out in Japan. The official name of the project is "Research and Development of the Platform for Designing High Functional Material". The project was proposed by the Ministry of International Trade and Industry in 1997, entrusted to NEDO and conducted at Nagoya University from 1998 to 2002 with 11 industries participating the project.

The purpose of this project is to construct a computer simulation system which bridges the microstructural(or molecular) characteristics of materials with the macroscopic characteristics of the materials. The material we focused on is the soft materials which generically refers to polymers, colloids, surfactants and gels.

Understanding how material properties are related to its microscopic structures has been a central problem in physics, chemistry and material science. It is also a crucial question in the research and development in industries since the understanding of the macro-micro relationship is the base of the technology to control and improve existing materials and/or to create innovative materials.

Our objective was to build a simulation system which aids this activity. As computer is becoming a useful tool in the design of architecture, machines and electrical devices, we wanted to make a similar simulation tool for material engineering. This idea is not new. The word "Computer Aided Material Engineering" has been used for more than 10 years, and it has seen some successes in some areas. However, in soft materials, the idea is not easy to achieve.

The properties of soft materials, say polymers, are not determined by the constituting monomers only: they depend on many other material characteristics, such as molecular weight, molecular weight distribution, branching structure, degree of chain orientation, degree of crystallization, and the states of the crystal-amorphous interfaces. The situation becomes even more complex in the case of polymer blends or composites where the dispersion state, and the interfaces between the component phases change the material property drastically.

Such complex problems cannot be handled by a single simulator. Many length scales are involved and many physics are working in the phenomena. This type of problems is now called multi-scale, multi-physics problem, and how to construct a simulation system to deal with such a problem is now becoming a challenge in computational science and engineering.

#### 1.1.2 Seamless zooming

Ideally, the tool we would like to have to study such materials is a system by which we can "zoom-in" to the materials at any length scale, and see and test the phenomena occurring in the material by computer simulation. We called such system "seamless zooming" system. This represents the image of the ultimate form of the simulator we are aiming at.

How can we realize such simulator? In academia, there have been several proposals for the multi-scale modeling, which typically uses two or more physical models simultaneously in the simulation. Though attractive they are, we did not pursue such study as the top priority of our project activity. There are two reasons. Firstly, the multi-scale modeling is a highly exploratory research theme, and too risky for the

project of our size. Secondly, the current multi-scale modeling requires us to select a specific problem for a specific system. This was not possible in our project: we needed a tool for general use.

In this project, we restricted ourselves to produce softwares which are useful for the current level of modeling soft materials. A characteristic of the soft materials is that the relevant length scale is mesoscopic: it is neither atomistic (less than 1 *nm*) nor macroscopic (more than 1 *mm*), but intermediate (between 1 *nm* to 1 *mm*). Theories of soft materials use mesoscopic modeling, and, in academia, many simulations have been done based on them. However, these simulators are made for the personal use of researchers, and usually unusable for the other people. As a result simulators of similar kinds have been made again and again in many laboratories in the world without being used again. We wanted to change this situation.

## 1.2 Design policy of the OCTA system

### 1.2.1 Simulation engine

We set two objectives for our project. The first objective is to build simulation programs which are typically used in the study of soft materials. We called such simulation programs "engines" since their function is to conduct calculations many many times following the instruction of users.

Clearly, the soft material simulator needs many varieties of engines. We chose to build four engines each based on molecular dynamics, reptation dynamics, Edwards self-consistent field theory, and continuum dynamics. They are:

**COGNAC** This engine conducts molecular dynamics calculation for polymer models under various external field (flow and deformation): it can deal with a large class of molecular models ranging from full atomistic models to coarse grained models such as bead-spring models.

**PASTA** This engine calculates the rheological response of entangled polymers using the reptation dynamics and the dual slip-link model of entanglement.

**SUSHI** This engine solves the self-consistent field equation of Edwards, and calculates the structure of polymers induced by phase separation or surface effect.

**MUFFIN** This is a generic engine solving the partial differential equations appearing in the problem of soft materials. It actually consists of five simulation programs, each dealing with (1)phase separation, (2)electrolyte dynamics, (3)micro-fluidics under electric fields, (4)elastic deformation of solids and (5) deformation and swelling of gels.

( The naming of the above engines follows the project internal rule decided at a beer party that engines must have a name of an object which can be found on any kinds of drinking party.)

Detailed description of the engines will be given in the following manuals.

### 1.2.2 User interface

To integrate the engines described above, we needed some platform on which these engines work together. An important design problem there was how to let different engines work together. This problem turned out to be quite difficult. The first problem we had was how to let engines share information with each other. This problem was already quite difficult.

At first, we wanted to have a framework within which we can express certain common notions like, "monomers", "polymers" or "molecular weight". However this approach turned out to be quite difficult. The engines are based on different physical models, and have different views for the definitions of "polymers". Settling the terminology is difficult, and settling the data structure is even more difficult. After some fierce discussion, we realized that even if we were able to invent some data format, nobody would be happy, and nobody would use it.

We abandoned the idea of defining a common data format. The lesson we learned here is that the integration must be done not by force but by voluntarily collaboration of people. We started to think how we can facilitate such collaboration.

The conclusion we arrived at finally is that we should focus on building a common graphic user interface which can be used by all engines. If all engines have the same user interface, users can use various engines easily. If the user interface have some programming function, user can change the data format of one engine



to the other, and can do the zooming by himself. This will eventually make reality of the "seamless zooming". We called such general user interface "platform". With that aim, we developed the software GOURMET (Graphical Open UseR interface for Material design Environment).

GOURMET is an editor and viewer of the text file which engine uses. For the text file to be understood by GOURMET, it must be written in a certain format called User Definable Format (UDF) which will be described in the next section.

If the text file is written in UDF, user can see and edit the data in various forms, process the data by script language, do plotting and display the result in 3D animation. GOURMET also has functions to control engines: user can monitor certain parameters of the engine, stop, change parameters, and restart again.

### 1.2.3 User definable format

The basic idea of UDF is to give a name to all data in a file. UDF file consists of two parts, the data definition part and the data part. The data definition part is needed to name the data in the file. When a UDF file is read by GOURMET, all numbers and texts in the file are given their names, and can be quoted by their names. This becomes convenient for the data analysis and data process since any data in the file can be quoted by their names, and can be processed by the script language implemented in GOURMET.

The engine programmer can access the data in the UDF file by their names using the platform library: once an UDF file is open, engine program can read and rewrite any data in the file in any sequence.

In the data definition part, engine programmer can define their own unit system using some basic physical quantities, and can state it for each data in the UDF file. Though engine programmer can use UDF file without using this function, we encourage them to state the unit system they use, and to state the unit of each data. Unit is an essential information in physical quantities, and it is essentially needed in converting the data from one UDF file to the other.

In the UDF file, engine programmers can put comments to each data name. The comments can describe the meaning of the data and/or its role in the program. The comments are used by GOURMET and aids the users who opened the UDF file. In future, this function will be enhanced by , for example, jumping to a certain URL, which can give much more information about the meaning of the data.

For a data set in the UDF file, the user of GOURMET can assign a procedure associated with the data set: for example, for a data of random numbers, they can assign a procedure of calculating their averages, and kick this program. This function can be used to prepare the input file and to analyse the output file.

## 1.3 What we are aiming at

As you see in the above description, we intentionally took a conservative approach in the software design toward our ultimate goal "seamless zooming". There is no mechanism in the design of GOURMET which forces zooming in a certain way. The activity of zooming is entirely left to the user who uses OCTA for a particular system.

From the view point of computational science and engineering, to design a system which secures the procedure of zooming is very challenging. Such research should be done in future in the collaboration of physicists, chemists and software engineers. At this stage, we have a view that what is currently needed toward our goal of "seamless zooming" is actually a mechanism which facilitates the collaboration of many simulation programs, which essentially means the collaboration of people. We hope that what we created will become a base for such collaboration.

Seamless zooming is a grand challenge. We believe that it is a goal which will be reached eventually. At this stage, we are far from that goal. The engines we developed here are not complete: they should be enhanced further, and new engines must be added. The interface we made needs to be brushed up. For this purpose, we are making all the source code open and allow modifications of the users.

## 1.4 Why "OCTA"

OCTA is Greek "8", which is a 90 degrees rotation of the mathematical symbol of infinity. It symbolizes the infinite possibility brought by the collaboration of various simulators dealing with various length scales from micro cosmos to universe. In Chinese character, OCTA (eight) is written as "八", which has a meaning

of good luck as it symbolizes the growth as we go down. The Chinese character “人” may be viewed as a mountain, difficult to climb, but eternally inspiring our challenge.

We very much welcome your feed back to this project.

# Chapter 2

## Outline of the OCTA system

### 2.1 Software Components

OCTA consists of the following three components.

- (1) Simulation programs (or engines). OCTA have four simulation programs and several utility programs.

- **Coarse-grained molecular dynamics simulator – COGNAC**

This is the molecular dynamics engine which can handle a variety of molecular models including the fully atomistic molecular model and the coarse grained molecular models for soft materials..

- **Rheology simulator – PASTA**

This engine is dedicated to predict the rheological properties of polymer melts based on the reptation theory for the dual slip-link model of entanglement.

- **Interfacial dynamics simulator – SUSHI**

This engine predicts the mesoscopic scale structure in polymeric systems caused by phase separation, micellization, surface adsorption, grafting etc based on the mean-field theory of Edwards.

- **Continuum dynamics simulator – MUFFIN**

This is a generic simulator which solves continuum equations for flow, diffusion, deformation, electric potential etc in soft materials. It actually consists of five engines each dealing with phase separation, electrolyte dynamics, micro-fluidics under electric fields, elastic deformation of solids and deformation and swelling of gels.

- (2) I/O interface library. OCTA uses UDF (user definable format) as a common expression for the data. The UDF file is a text file, but can be viewed as a random access storage: any data in UDF can be read and written in any order. OCTA supports this function for C/C++ and Python.

- **libplatform**

The library for reading and writing UDF files by C/C++ programs.

- **Python UDF Manager**

The library for reading and writing UDF files by Python.

- (3) Modeling Platform(GOURMET) **GOURMET** is a customizable graphical interface for the engines in OCTA. In running an engine you can use GOURMET to (1) edit the input file, (2) to run the simulation engine in network environment (3) to view the simulation results in tables and 3D animations (4) to do your own analysis for the simulation results. You can create your own interface which suits your own problem without touching the engine program: the customization is done only at the level of GOURMET.

### 2.2 Document list

The following documents are distributed with OCTA. These are offered also as PDF (Portable Document Format) data contained in Distribution CD Volume 2 (under a directory "DOCUMENTS").

- **Overview**(this booklet)

It describes the purpose, the special feature, the system configuration, and the content of the package of OCTA.

- **Installation Guide**

Installation method of OCTA to a user's computer is explained. Those who introduce OCTA need to install a system according to this document first.

- **GOURMET Primer – TUTORIAL**

Tutorial and user's guide for GOURMET.

- **Graphical User Interface GOURMET – Operation's Manual**

Description of edit of the UDF data based on the user interface GOURMET of OCTA, and 3 and 2-dimensional graphical display, etc.

- **COARSE-GRAINED MOLECULAR DYNAMICS PROGRAM – COGNAC – User's Manual**

It includes details of the COGNAC engine, such as theory and usage.

- **RHEOLOGY SIMULATOR – PASTA – User's Manual**

It includes details of the PASTA engine, such as theory and usage.

- **SIMULATION UTILITIES FOR SOFT AND HARD INTERFACES – SUSHI – User's Manual**

It includes details of the SUSHI engine, such as theory and usage.

- **MULTI-PHASE DYNAMICS PROGRAM – MUFFIN – User's Manual**

It includes details of the MUFFIN engine, such as theory and usage.

- **GOURMET Python Script Reference**

It is the description about processing with the Python language which characterize OCTA. It contains brief description of the syntax of a Python language.

- **UDF Syntax Reference**

The syntax of UDF is described. It should be referred when creating the UDF definition file for a new engine or to know the syntax of UDF data in detail.

- **Platform Interface Library – libplatform – Reference**

The detail of the interface library for the C++ program which outputs and inputs UDF format data is described. It should be referred when creating the tool which newly treats an engine and UDF.

- **APPLICATION REPORT**

Report of applications of OCTA. It includes many example of calculations performed by OCTA engines.

- **APPLICATION REPORT – AMUSE**

Description of the virtual experiment prototype project AMUSE by seamless zooming using OCTA.

# Appendix A

## Glossary

The main terms used in description of an OCTA system.

### **UDF**

Data format used by the OCTA system in common for the exchange of data over different simulation programs (engines) or tools (User Definable Format). It is a self-description type of data format which enables description of the meaning of data in the data itself.

### **GOURMET**

GUI (Graphical User Interface) program of the OCTA system. It has UDF processing facilities such as edit of the UDF file for simulation program (engine) input, starting of an engine, a display/analysis of the engine calculation result UDF, and UDF processing using Python scripts etc.

### **Engine**

In OCTA system, a simulation program is called a "simulator" or "engine."

### **COGNAC**

The coarse-grained molecular dynamics engine contained in OCTA.

### **SILK**

An auxiliary tool which generates molecular structure data for the input UDF of the COGNAC engine.

### **PASTA**

A rheology simulation engine contained in OCTA.

### **FORK**

An auxiliary tool for an input UDF generation of the PASTA engine.

### **SUSHI**

An interface structure and dynamics simulator in OCTA. It uses a dynamic mean-field theory.

### **MUFFIN**

A multi component/multi-phase dispersed-system simulation engine contained in OCTA.

### **Platform interface library (libplatform)**

A library which includes procedures (a C++ language, Python) required for a simulation program (engine), an auxiliary tool, etc. to input and output data of UDF format.

## **Python**

An "object-oriented script language" used by the OCTA system. It is a free software developed by Mr. Guio van Rossum. It can be used on many systems, including Microsoft Windows, various UNIX systems, etc.

## **Java**

An object oriented programming language developed by the Sun Micro Systems Inc.. The GOURMET system is described by Java.

## **OpenGL**

A programming library for 3-dimensional graphic operation which GOURMET is using. In the Linux system, the Mesa library which is a free implementation of this library specification is used.

## **Cygwin**

It is a software which enables to use a UNIX-like environment (POSIX standard) on MS Windows system. It is developed by the Cygnus Solutions (currently under Red Hat Corp.), and is provided as charge free. Some of simulation programs (engine) of OCTA operate in the Cygwin environment.

## **OCTA**

That's OCTA !