

# OCTA

ソフトマテリアルのための統合化シミュレータ



for OCTA

version 1.2

ユーザーズマニュアル

OCTA ユーザーズグループ  
Mar 1 2013

## 執筆者

全章                      増淵 雄一  
OCTA 対応部分    小沢 拓, 西尾 裕三, 吉田 英代

## プログラム開発者

メインプログラム    増淵 雄一  
UDF コンバータ    西尾 裕三, 吉田 英代

## 謝辞

NAPLES の理論はイタリアナポリ大学工学部 Giuseppe Marrucci 教授, Giovanni Ianniruberto 教授, イタリア CNR Francesco Greco 教授との共同研究により生み出されました. 高分子ブレンドに関する理論では東京大学土井正男先生のご指導を受けています.

NAPLES は新エネルギー・産業技術総合開発機構 (NEDO) 平成 13-15 年度産業技術研究助成事業「高分子からみあい系超高速シミュレーターによる熔融構造制御」, 日本学術振興会平成 15-16 年度科学研究費補助金基盤研究 C(2)「からみあい高分子ブレンドにおける高分子ダイナミクス新規高速計算手法の検討」, 科学技術振興機構 (JST) 平成 18 年-22 年度戦略的創造研究推進事業「アナログ&デジタル融合高分子ナノシミュレーション」, 科学技術振興機構 (JST) 平成 16 年-19 年度戦略的創造研究推進事業「ソフトマターの多階層/相互接続シミュレーション」, 日本学術振興会平成 20-22 年度科学研究費補助金基盤研究 (B)「分岐高分子の新規分子モデル」, 日本学術振興会平成 23-25 年度科学研究費補助金基盤研究 (B)「高分子の流動下での緩和」, の成果を利用しています.

解析ソフトウェア gt2et.exe, gt2gw.exe, smooth.exe, gwsmooth.exe は, OCTA - PASTA エンジンに含まれるもので, 名古屋大学滝本淳一先生により開発されたものを転用しています. ここに御礼申し上げます.

また OCTA バージョンのプログラム開発は, 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」とプロジェクト終了後に OCTA ユーザーズグループにて継続して行われたものです.

Copyright ©2000-2012 OCTA Licensing Committee    All rights reserved.

# 目次

<b>第 1 章</b>	<b>NAPLES とは何か</b>	<b>3</b>
<b>第 2 章</b>	<b>インストールと実行方法</b>	<b>5</b>
2.1	インストール	5
2.1.1	OCTA 版 NAPLES について	5
2.1.2	インストール方法	5
2.2	NAPLES によるレオロジー計算の流れ	6
2.2.1	naplesOCTA(.exe) の起動	6
2.2.2	naplesOCTA(.exe) の停止	7
2.2.3	レオロジー計算エンジンの再開	8
2.3	UDF を用いた実行	9
2.4	解析用コマンドリファレンス	9
2.4.1	gt2et(.exe)	10
2.4.2	gt2gw(.exe)	10
2.4.3	smooth(.exe)	11
2.4.4	gwsmooth(.exe)	12
2.4.5	corr(.exe)	12
2.4.6	naplesgwOCTA(.exe)	12
2.4.7	naplesgwOCTA(.exe) の UDF を用いた実行方法	13
2.4.8	getggval(.exe)	14
<b>第 3 章</b>	<b>レオロジー計算と解析の実際</b>	<b>15</b>
3.1	ステップ変形からの応力緩和と緩和弾性率の算出	15
3.1.1	計算の実行と入力ファイル例	15
3.1.2	結果（応力）の表示	16
3.1.3	データのスムージング	17
3.1.4	応力から弾性率への変換	18
3.2	線形緩和弾性率を平衡状態の計算から求める	18
3.2.1	計算方法	19
3.3	G'G'' を計算したい	20
3.3.1	線形緩和弾性率から求める場合	21
3.3.2	線形緩和弾性率からの計算例	21
3.3.3	動的粘弾性試験を行う場合	23
3.3.4	動的粘弾性を連続実行で行う場合	24
3.3.5	スキャン周波数範囲をかえるには	24
3.4	スタートアップ流れによる応力成長	24
3.4.1	結果解析	25
3.4.2	計算例	25
3.5	線形粘度成長曲線	27

3.5.1	結果解析	27
3.6	粘度の求め方	27
3.7	流動曲線（粘度のせん断速度，伸長速度依存性）	28
3.8	実験とのフィッティング	30
3.8.1	NAPLES の単位時間	30
3.8.2	NAPLES の単位応力（単位弾性率）	31
<b>第 4 章</b>	<b>NAPLES 入力ファイル</b>	<b>33</b>
4.1	基本ルール	33
4.2	もっとも単純な入力パラメーターファイル	33
4.3	試料の設定	34
4.3.1	化学種数の指定と関連パラメーターの設定	34
4.3.2	鎖の伸びきり効果の指定	35
4.3.3	摩擦低減効果の指定	35
4.3.4	構成成分数の指定	35
4.3.5	分子形状の指定：概要	37
4.3.6	直鎖高分子	38
4.3.7	星形高分子	39
4.3.8	非対称星形高分子（共重合高分子）	39
4.3.9	ポンポン形高分子	40
4.3.10	ポンポン共重合高分子	41
4.3.11	櫛形高分子	41
4.3.12	櫛形共重合高分子	42
4.3.13	直鎖共重合体	42
4.4	ゴムやゲル等の架橋材料	43
4.4.1	架橋点導入用入力ファイル	43
4.4.2	架橋点導入の応用	44
4.4.3	架橋点導入後のレオロジー計算	45
4.5	分子量 Z 値の決め方	46
4.6	変形および流れ場の設定	46
4.6.1	基本事項	46
4.6.2	時間スケールおよび流動速度について	46
4.6.3	系の平衡化（熱処理）	46
4.6.4	ステップ変形	47
4.6.5	スタートアップ流れ（定常流）	48
4.6.6	動的せん断	49
4.7	計算の詳細条件の設定	49
4.7.1	乱数の種の設定	49
4.7.2	計算セルサイズの設定	50
4.7.3	セグメント数密度	50
4.7.4	計算時間刻みの設定	51
4.7.5	出力ファイルの設定	51
4.8	パラメーターファイルリファレンスマニュアル	51
4.9	UDF ファイルリファレンス	54

<b>第 5 章</b>	<b>NAPLES 出力ファイル</b>	<b>61</b>
5.1	状態報告ファイル <code>movelog.npls</code> . . . . .	62
5.2	実行パラメータ確認ファイル <code>prmcheck.npls</code> . . . . .	63
5.3	応力出力ファイル <code>measure.npls</code> . . . . .	64
5.4	分子形態統計値出力ファイル <code>stat.npls</code> . . . . .	64
5.5	最終形態保存ファイル <code>finalconf.npls</code> . . . . .	65
5.6	途中経過保存ファイル <code>config[xxxx].npls</code> . . . . .	67
5.7	伸長およびせん断応力データファイル群 . . . . .	67
5.8	線形せん断緩和弾性率ファイル <code>gt.npls</code> , <code>gt_smooth.npls</code> . . . . .	68
5.9	線形せん断粘度成長ファイル <code>et.npls</code> , <code>et_smooth.npls</code> . . . . .	69
5.10	動的粘弾性ファイル <code>gw.npls</code> , <code>gw_smooth.npls</code> , <code>gwdirect.npls</code> . . . . .	69
5.11	架橋構造解析ファイル <code>tailNdist.npls</code> , <code>tailZdist.npls</code> . . . . .	69
<b>第 6 章</b>	<b>計算原理</b>	<b>71</b>
6.1	緒言 . . . . .	71
6.2	Primitive Chain Network モデル . . . . .	71
6.3	分岐高分子の扱い . . . . .	75
6.4	異種高分子ブレンドおよび共重合体への拡張 . . . . .	76
6.5	入力パラメーターおよび出力物性単位 . . . . .	78
6.6	計算対象となる物質/ならない物質 . . . . .	79
6.7	計算が可能な物性/不可能な物性 . . . . .	80
<b>第 7 章</b>	<b>NAPLES の TIPS</b>	<b>81</b>
7.1	計算負荷を落として結果を早く出すには? . . . . .	81
7.2	計算精度を上げるには? . . . . .	81
7.3	OCTA に含まれる他のエンジンとの関係 . . . . .	82
7.4	NAPLES 情報ホームページ . . . . .	82
<b>References</b>		<b>83</b>



# 目 次

3.1	ステップせん断における応力緩和の例：データ列をそのまま出力	16
3.2	ステップせん断における応力緩和の例：平衡化している部分をカット	16
3.3	ステップせん断における応力緩和の例：スムージングした結果（両対数で表示）	17
3.4	応力緩和カーブの例	18
3.5	平衡状態におけるせん断応力のゆらぎ	20
3.6	平衡状態から得られた相関関数とステップせん断変形で得られた緩和弾性率の比較	20
3.7	$Z=8$ の直鎖高分子の線形緩和弾性率	22
3.8	$Z=8$ の直鎖高分子の $G'G''$	22
3.9	$Z=8$ の直鎖高分子の $G'G''$ （スムージングとの比較）	23
3.10	動的せん断ひずみを与えた時の応力の時間変化	23
3.11	一軸伸長における応力成長の例：データをそのまま出力	26
3.12	一軸伸長における応力成長の例：平衡化部分を削除	26
3.13	一軸伸長における応力成長の例：スムージング結果とともに表示	27
3.14	スタートアップせん断における粘度成長曲線の例	29
3.15	定常粘度のせん断速度依存性の例	29
4.1	直鎖高分子	38
4.2	星形高分子	39
4.3	非対称星形高分子	40
4.4	ポンポン形高分子	40
4.5	櫛形高分子	41
4.6	トリブロック共重合高分子	43
4.7	Gourmet で UDF ファイルを開いた例	55
6.1	Primitive Chain Network モデルの概念図. (a) からみあい点の運動. (b) チューブ束縛内部の高分子のすべり運動.	72
6.2	primitive chain network モデルの典型的スナップショット. 多数の鎖が系にはあるが, 一本だけを抜き出して表示. トゲのように見える細い線は, からみあっている周囲の分子のセグメント.	74
6.3	$G^*(\omega)$ の実験 [11] と計算の比較.	75
6.4	せん断粘度の実験との比較 [11]	75
6.5	星形高分子の拡散, 実験 [12] との比較	76
6.6	高分子ブレンドの相分離ダイナミクス	77
6.7	共重合高分子のマイクロ相分離	78





## 表 目 次

4.1	分子形状を規定するパラメータ . . . . .	38
4.2	ステップ変形の流動形態指定 . . . . .	47
4.3	スタートアップ流れの流動形態指定 . . . . .	48
4.4	パラメータリファレンス . . . . .	51
6.1	NAPLES の単位量 . . . . .	78



## 第1章 NAPLES とは何か

NAPLES は New Algorithm for Polymeric Liquids Entangled and Strained (からみあい、ひずみを与えられた高分子液体のための新しい計算手法) の略称で、増渕雄一がナポリ大学の Giuseppe Marrucci 教授らと共同で開発を進めている高分子ダイナミクスとレオロジーを予測するためのシミュレーターである [1-5]。高分子個々の運動を計算する分子シミュレーションと呼ばれるシミュレーターに属するが、通常分子シミュレーションでは計算不可能な、高分子の長時間にわたる緩和現象を計算することができる点が大きな特徴である。

分子シミュレーションであるから、ミクロな分子の構造を与え、その構造をもつ分子の運動を計算させ、計算結果からマクロな物性値を予測するのが想定される利用法である。<sup>1</sup> より具体的に想定される利用法として以下が考えられている。

### 想定される用途

- 高分子のからみあった運動を可視化。
- 高分子の分子量、分子量分布、分岐構造を変えたときのレオロジーへの影響を知る。
- セン断、一軸、二軸、平面伸長変形が分子運動に与える影響を知る。
- 高分子ブレンドや共重合体系のレオロジーを調べる。
- からみあった高分子ブレンドや共重合体系の構造形成を観察する。
- ...

理論的背景についての詳細は 6 章をご覧ください。

<sup>1</sup>マクロな物性値から、それを発現するためのミクロな構造を求めることは、いわゆる逆問題であり、直接の機能としては備えていない。



## 第2章 インストールと実行方法

### 2.1 インストール

#### 2.1.1 OCTA 版 NAPLES について

OCTA 版 NAPLES には、Windows 版と X86-Linux 版の実行モジュールと解析ツール群が含まれている。実行モジュールについては、ソースコードは現時点では公開していない。Linux については、RedHat, Fedora Core 系のディストリビューションでの使用を推奨する。もし問題があればご一報いただきたい。

#### 2.1.2 インストール方法

UDF 形式の Input ファイルを用いる場合には、OCTA がインストールされている必要がある。インストーラを実行することで、以下のファイルが OCTA のインストールディレクトリにコピーされる。インストーラを用いない場合、パスが通ったディレクトリに以下のファイルを置く。

なお、解析ツールの一部は OCTA の PASTA エンジンに含まれるフリーウェア (山形大学滝本淳一先生作成) を含む。すでに PASTA がインストールされている場合、これらのモジュールは新たにインストールする必要はない。

1. gt2gw(.exe) : 線形緩和弾性率を  $G'G''$  曲線に変換 (OCTA-PASTA のものと同等)
2. gt2et(.exe) : 線形緩和弾性率を線形せん断応力成長曲線に変換 (OCTA-PASTA のものと同等)
3. smooth(.exe) : 出力結果をログスケールでスムージング (OCTA-PASTA のものと同等)
4. gsmooth(.exe) : gt2gw.exe で得られた  $G'G''$  曲線をスムージング
5. corr(.exe) : 数列の自己相関関数を作成
6. getgval(.exe) : ダイナミックせん断モードの結果ファイルから  $G'G''$  を計算.
7. naplesOCTA(.exe) : NAPLES 実行モジュール.
8. naplesgwOCTA(.exe) : naplesOCTA をダイナミックせん断モードで連続実行. 周波数応答を計算.
9. naplesOCTA.sh(Windows 版では.bat) : Input UDF ファイルを用いて naplesOCTA(.exe) を起動.
10. naplesgwOCTA.sh(Windows 版では.bat) : Input UDF ファイルを用いて naplesgwOCTA(.exe) を起動.
11. naples\_input\_converter(.exe) : Input UDF から, NaplesOCTA が必要とする Input データ形式へ変換する.
12. naples.udf : UDF 形式の Input ファイルを用いるための定義ファイル.
13. 計算結果の解析用アクションファイル (応力解析・実単位への変換・プロット機能)

## 2.2 NAPLES によるレオロジー計算の流れ

naplesOCTA(.exe) がレオロジー計算エンジンである。それを用いてレオロジー計算をする手順の概要をここに説明する。

なお、OCTA 版では *Input* として *UDF* 形式のファイルを用いることも可能である。その場合、*naplesOCTA.sh* (Windows 版では *.bat*) を実行することで *UDF* 形式の *Input* ファイルが *NAPLES* オリジナルの *Input* ファイルに変換されて、*naplesOCTA(.exe)* が自動的に起動される。*UDF* 形式の *Input* ファイルの書式については 4 章参照。

レオロジーを計算する際のあらまし

1. 入力パラメーターファイルを作成する。上述のように、OCTA 版では *UDF* 形式の入力ファイルを用いることも可能である。(以下の流れに従う。ファイル作成の詳細は 4 章参照)
  - (a) 計算する分子の形状 (直鎖, 分岐, それらの混合など) を決める。(4.3.5 参照)
  - (b) 計算したい分子の分子量を決定し, *NAPLES* での分子量 *Z* 値に変換する。(4.5 参照)
  - (c) 計算する分子の最長緩和時間のメドをつけて, 計算時間 (*quit\_dynamics\_count* 値) と平衡化時間 (*pre\_reptation\_step* 値) を決める。(3.8.1 節参照)
  - (d) 変形の様式とひずみ量を決める。(4.6 節参照)
2. *naplesOCTA(.exe)* を起動する。(2.2.1 節参照)
3. 計算の途中経過を *movelog.npls* ファイルおよび *prmcheck.npls* ファイルの内容により確認する。(5 節参照)
4. 入力ファイルに書き込むフラグにより, *measure.npls* ファイル等に各種の結果が出力される。
5. 計算が終了すると, *movelog.npls* ファイルには "ENDS" というメッセージが出力される。
6. 結果を解析する。(3 章参照)

なお架橋された系の場合は, 手順が少々異なるので, 別途 4.4 節も参照のこと。

### 2.2.1 naplesOCTA(.exe) の起動

ここでは *Input* として *UDF* 形式ではなく, *NAPLES* オリジナルのファイル形式を用いる場合について説明する。まず, UNIX であればターミナル, Windows であれば *cygwin* または *DOS* プロンプトを起動する。(cygwin をインストールした場合にはデスクトップにショートカットがある。) コマンドを入力できる状態になったら (MS-DOS のウィンドウまたは *cygwin* のウィンドウが開いたら), *naplesOCTA.exe* を起動する際の条件がかかれた入力パラメーターファイルがあるディレクトリで *naplesOCTA.exe* を起動する。

naplesOCTA(.exe) の起動

```
>naplesOCTA(.exe) [入力パラメーターファイル名]
```

このとき, 入力パラメーターファイルが指定されないと, カレントフォルダにある *input.npls* の条件で起動する。もし *input.npls* ファイルも無いときは, プログラムに組み込まれた初期条件 (*Z*=10, *C*=10, *cellsize*=6, 流動なし) で起動する。

正常に起動しているかどうかを確認するには, *NAPLES* を起動したフォルダに *movelog.npls* および, *prmcheck.npls* という 2 つのファイルが生成されるはずなので, これをノートパッドなどのテキストエディタで見

る。cygwin ではコマンドラインでチェックできる。その例を示す。

—— 起動時の読み込みファイルをチェック (cygwin および UNIX の場合) ——

```
>cat movelog.npls          「movelog.npls の中身を表示」
```

```
-----
```

```
naplesOCTA(.exe) is started (NAPLESrev ) with parameters in input.npls Thu Mar 17 18:32:40 2005
Initialization started. Thu Mar 17 18:32:40 2005
Main calculation started. 0   Thu Mar 17 18:32:46 2005
EQUILIBRIUM_CALC Thu Mar 17 18:32:46 2005
```

この例では、input.npls が入力パラメーターファイルとして選ばれたことが示されている（2行目）。ファイルの中身については5節を参照。

計算の途中経過を見るには、応力が計算されて出力されるファイルである measure.npls の下端部を見るとよい。cygwin や Linux で measure.npls ファイルの下端を見るには、以下のように入力する。

—— 途中経過のチェック例 (cygwin および UNIX の場合) ——

```
>tail measure.npls          「measure.npls の終端部を表示」
```

```
0  0.00e+00    5.519445e-01  1.430961e+00  1.137091e+00  1.193193e+00
1  0.00e+00    1.126641e-01  1.273881e+00  1.186699e+00  1.205338e+00
2  0.00e+00    8.506818e-02  1.227362e+00  1.163581e+00  1.177303e+00
3  0.00e+00    4.590036e-02  1.193680e+00  1.152211e+00  1.175619e+00
4  0.00e+00    3.322798e-02  1.158326e+00  1.172665e+00  1.141329e+00
```

出力されたデータのうち、左端が経過した時間ステップである。なお、ここでの時間ステップは NAPLES 単位時間になっている。実時間との対比については3.8.1節を参照のこと。

## 2.2.2 naplesOCTA(.exe) の停止

計算を途中で停止するには、大きく分けて二つの方法がある。現在の計算状態を残し、後から計算を再開したい場合には、以下で説明する正常終了の手続きを行う。計算を再開する必要がない場合には、Windows のタスクマネージャーを使うか、cygwin や Linux では kill コマンドを使う。

### 正常終了の方法

naplesOCTA(.exe) が実行されているディレクトリ（measure.npls ファイルなどが出力されているディレクトリ）で、ctrl.npls ファイルの内容を書き換える。NAPLES が実行しているとき、ctrl.npls ファイルの中身は以下のようにになっているはずである。

—— naplesOCTA(.exe) 実行時の ctrl.npls の内容 ——

```
RUN
```

これを、以下のように書き換える。

—— naplesOCTA(.exe) を停止させるための ctrl.npls の変更 ——

```
STOP    （または   KILL）
```

ctrl.npls ファイルを書き換えて保存する。NAPLES は一定の時間間隔で ctrl.npls を読んでいたので、しばらくすると形態保存ファイルである finalconf.npls を計算終了時の内容に書き換えて停止する。停止した計算を再開するには、2.2.3節を参照のこと。

### 2.2.3 レオロジー計算エンジンの再開

いったん停止させた（あるいは何らかの理由で停止した）NAPLES を停止時点から再開するには、再開用ファイル `finalconf.npls` を利用する。このファイルは停止直前<sup>1</sup>のシステム全体の情報を記録してあるもので、これがないと再開できない。通常はNAPLESを起動したフォルダに作成されているはずである。また再開時には入力パラメーターファイルの記述を誤るなどすると、`finalconf.npls` ファイルが書き変わってしまい、計算を再開できなくなるので、再開前に必ずオリジナルの `finalconf.npls` ファイルのコピーを保存することをお奨めする。

次に、計算に用いる入力パラメーターファイルに、以下の記述を付け加える。

—— 計算を再開させるためのパラメータ記述 ——

```
calculation_mode=restart
```

この記述がある場合にはNAPLESは必ず `finalconf.npls` を探しに行くので、NAPLES 実行ファイルと同じフォルダに `finalconf.npls` ファイルがないとエラーメッセージを `movelog.npls` ファイルに出力して実行を停止する。また、再開するには、`calculation_mode` 以外の記述は、計算停止前に用いていた入力パラメーターファイルと全く同じである必要がある。（seed 値は違っていてもよい。）`finalconf.npls` と書き換えた入力パラメーターファイルが `naplesOCTA(.exe)` と同じフォルダにあることを確認したら、2.2.1の方法に従ってNAPLESをコマンドラインで起動すればよい。

例えば、DドライブのDataフォルダで、`linear.npls` を入力パラメーターファイルとして計算をおこなったとする。まず `cygwin` でフォルダの中身を確認すると、以下のようになるはずである。

—— 計算をおこなったフォルダの中身をチェック（`cygwin` ないし `UNIX` の場合） ——

```
>cd d:/Data          「データフォルダに移動」
>ls                  「カレントフォルダの中身を表示」
finalconf.npls linear.npls measure.npls movelog.npls prmcheck.npls ctrl.npls
```

`finalconf.npls` があること、および入力パラメーターファイルがあることを確認する。次にノートパッドや `vi` エディタを用いて `linear.npls` ファイル中に `calculation_mode=restart` を書き加えておく。最後に計算を再開するには、以下のように入力する。

—— 計算再開時の例（`cygwin` ないし `UNIX` の場合） ——

```
>cd d:/Data          「データフォルダに移動」
>cp finalconf.npls finalconf.npls.org 「オリジナルファイルを退避」
>cat linear.npls | grep calculation_mode 「計算再開モード確認」
calculation_mode=restart                「もしこの出力がなければエディタで付け加える」
>d:/NAPLES/naplesOCTA(.exe) linear.npls & 「計算を再開」
```

上記の例では、まず `finalconf.npls` ファイルのコピーを `finalconf.npls.org` として保存している。また、`naplesOCTA(.exe)` は、DドライブのNAPLESフォルダにインストールしてあるものとしている。

停止時点から再開しているかどうかを確認するには、ノートパッドなどで `movelog.npls` ファイルを開く。`cygwin` では `cat movelog.npls` と入力すればよい。再開している場合には、以下の記述がある。

—— 正常に計算再開した場合の `movelog.npls` への記述 ——

```
Calculation is restarted from 1000   Thu Oct 17 21:42:47 2002
RESTART with a saved configuration file. 1000   Thu Oct 17 21:42:47 2002
```

これがない場合には再開されていないので上の手順を確認する。なお、再起動時の入力ファイルは計算停止時

<sup>1</sup>強制終了した場合には停止時よりかなり前の状態が保存されていることがある。デフォルトでは100ステップ毎に自動記録されている。



のものと同じであることが前提であるが、流動や変形の様子を変えることは可能である。例えば平衡状態の計算により生成された finalconf.npls ファイルを用いて、せん断流動下での計算を開始することはできる。流動条件、変形条件の記述法は 4.6 節を参照のこと。

—— 応用例：計算再開後にステップ変形を与える ——

calculation\_mode=restart をキーとして与えた場合、ステップ変形を記述した入力ファイルに応答しない。(変形が加えられた後に緩和させている状態を仮定して計算が再開されるため。) もし再開後に変形を加えたい場合には、以下のように記述する。

```
calculation_mode=deform
```

これにより時刻はゼロにリセットされる。なお、定常流の計算ではどちらも本質的な違いはないが、deform を指定すると時刻がゼロにリセットされる。

## 2.3 UDF を用いた実行

例えば、NAPLES が D ドライブの直下にインストールされている場合、以下のようにする。naplesOCTA.bat (Linux 版では.sh) を実行すると、パラメータで指定された UDF 形式の Input ファイルを NAPLES オリジナルの Input ファイルに変換した後に naplesOCTA(.exe) を実行する。naples\_input\_converter(.exe), naplesOCTA(.exe) へのパスが通っている必要がある。また、すでに OCTA がインストールされており、パスなどの諸設定が終了している必要がある。

—— UDF ファイルを Input にした場合の実行方法 ——

```
>cd d:/NAPLES          「NAPLES フォルダに移動」
>naplesOCTA.bat(Linux 版では.sh)  [入力 UDF ファイル名]
```

—— Gourmet から EngineRun パネルを用いて実行する場合の設定例 ——

```
Server: localhost
Engine: C:\OCTA2006Rev1\PF_ENGINE_2006\bin\win32\naplesOCTA.bat
Working Dir: C:\OCTA2006Rev1\PF_ENGINE_2006\Naples_for_OCTA\sample_input_udf\output
Params:

Input UDF: C:\OCTA2006Rev1\PF_ENGINE_2006\Naples_for_OCTA\sample_input_udf
           \sample_input.udf
以下、指定なし
```

※入力 UDF および出力ファイル群は、Working Dir で指定したディレクトリに書き込まれる。

## 2.4 解析用コマンドリファレンス

ここでは計算結果に対する解析を行うためのファイル群について説明を行う。<sup>2</sup> 入力ファイルに UDF を用いる場合には、これらの機能は UDF のアクション機能を用いて実行することが出来る。解析アクションの具体的な内容は、第 3 章を参照していただきたい。

<sup>2</sup>ここで解説する解析ツールの重要ないくつかは、名古屋大学土井プロジェクトで開発されたプログラムパッケージ OCTA [13] から転載している。OCTA および開発者の山形大学滝本淳一先生に感謝いたします。

### 2.4.1 gt2et(.exe)

線形緩和弾性率  $G(t)$  から linear stress growth function  $\eta^+(t)$  を計算する.

$$\eta^+(t) = \int_0^t G(t') dt' \quad (2.1)$$

コマンドラインでの使用法は以下の通り.

— gt2et.exe の使い方 —

`gt2et.exe [-s dt] < [入力ファイル] > [出力ファイル]`

ここで dt には時間刻み幅を指定する. デフォルトは 1.0 である. 入力ファイルの書式は

— gt2et.exe のための入力ファイル書式 —

$G(0)$   
 $G(dt)$   
 $G(2dt)$   
 ...

である. ここで  $G(t)$  は線形緩和弾性率を示す. 一方, 出力ファイルは各行に時刻とその時刻における  $\eta^+(t)$  がタブ区切りで出力される. なおこのコマンドは OCTA-PASTA からの転載である.

### 2.4.2 gt2gw(.exe)

線形緩和弾性率  $G(t)$  から, 動粘度  $\eta^*(\omega)$  または複素弾性率  $G^*(\omega)$  を計算する.

$$\eta^*(\omega) = \int_0^\infty G(t) \exp(i\omega t) dt \quad (2.2)$$

$$G^*(\omega) = i\omega \int_0^\infty G(t) \exp(i\omega t) dt \quad (2.3)$$

コマンドラインでの使用法は以下の通り.

— gt2gw.exe の使い方 —

`gt2gw.exe [-n m] [-s dt] [-e] < [入力ファイル] > [出力ファイル]`

ここで引数 m はフーリエ変換に用いるデータ数  $N = 2^m$  を示している.  $m$  のデフォルトは 16 であり, 実際に入力されるデータ数が  $M$  であるとき  $N = 2^m > M$  であるように選ぶ必要がある. また  $N > 4M$  であることが望ましい. この値のために, 入力されるデータ点数が多いとエラーとなることがある. そのときには -n 22 などとして  $m$  値を増やして計算する必要がある. 引数 dt は入力パラメーターファイルにおける時間刻みであり, デフォルトは 1.0 である. -e オプションが指定されていると, 粘度を算出する. デフォルトでは (このオプションを指定しなければ) 弾性率が算出される. 入力ファイルの書式は gt2et の時と同じで,

— gt2gw.exe のための入力ファイル書式 —

$G(0)$   
 $G(dt)$   
 $G(2dt)$   
 ...

出力は,  $d\omega = 2\pi(Ndt)$  の刻み幅で周波数と粘度や弾性率を出力する.  $G^*(\omega)$  の場合の出力は以下のような

る.

gt2gw.exe の出力ファイル書式 ( $G'G''$  の場合)

0	$G'(0)$	$G''(0)$
$d\omega$	$G'(d\omega)$	$G''(d\omega)$
$2d\omega$	$G'(2d\omega)$	$G''(2d\omega)$
$3d\omega$	$G'(3d\omega)$	$G''(3d\omega)$
...	...	...
$N/2d\omega$	$G'(N/2d\omega)$	$G''(N/2d\omega)$

なおこのコマンドは OCTA-PASTA からの転載である.

### 2.4.3 smooth(.exe)

対数プロットでの  $G(t)$ ,  $\eta^+(t)$ ,  $G^*(\omega)$  等を Savitzky-Golay 法を用いてスムージングする. 使用方法是以下である.

smooth.exe の使い方

`smooth.exe [-n ndiv] [-r r] [-x nmax] [-s skip] -2 [入力ファイル] > [出力ファイル]`

ここで `ndiv` は, ログスケールでの 1 桁に何点出力するかを示している. デフォルトは 20 である. `r` は 1 つ前のデータ出力点までの何倍の範囲を smoothing に用いるかを指定する. デフォルトは 2.0. `nmax` はデータ出力点の左右の何点のデータを用いるかを指定する. デフォルトは 1000. `skip` はスムージングせずにそのまま出力する, 出力当初のデータ点の数である. -2 を引数として指定すると 2 次の SG スムージングを行う.

smooth.exe の入力ファイル書式は以下でなければならない.

smooth.exe の入力ファイル書式

$t_0$	$v(t_0)$
$t_1$	$v(t_1)$
$t_2$	$v(t_3)$
...	...

ここで  $t_0 < t_1 < t_2$  でなければならないが, 等間隔である必要はない. (またここでは便宜的に  $t$  と書いているが,  $\omega$  でもかまわない.) 出力は以下のようになる.

smooth.exe の出力ファイル書式

$T_0$	$\tilde{v}(T_0)$
$T_1$	$\tilde{v}(T_1)$
$T_2$	$\tilde{v}(T_3)$
...	...

データ出力点  $T_0, T_1, T_2, \dots$  は, ログスケールで 1 桁を `ndiv` 等分した点となる.

なお, データ点数が少ない数列に対して smooth を適用するとエラーメッセージを返して起動しない時がある. このような時には, `nmax` の値を小さくするとよい. `nmax` の値が 1000 のままだと, データ点が 2000 から 3000 しかないデータ列に対してはうまく働かない. なおこのコマンドは OCTA-PASTA からの転載である.

### 2.4.4 gsmooth(.exe)

複素弾性率を記録してあるような、3 列（時間列 1 列に対してデータ 2 列）の形式をもつファイルをスムージングする。

smooth.exe の使い方

`gsmooth.exe` [入力ファイル] > [出力ファイル]

スムージングの詳細は 2.4.3 節の `smooth.exe` と同様であるが、スムージングに関する各種パラメーターは `smooth.exe` におけるデフォルト値が用いられる。

入力ファイル、出力ファイルはともに以下の形式である。

gsmooth.exe の入出力ファイル書式

```
t0  v1(t0)  v2(t0)
t1  v1(t1)  v2(t1)
t2  v1(t3)  v2(t3)
...  ...
```

ここで  $v_1$ ,  $v_2$  はデータを示す。

### 2.4.5 corr(.exe)

数列の自己相関関数を計算する。

$$C(\tau) = \frac{\langle \sigma(t)\sigma(t+\tau) \rangle_t}{\langle \sigma(t)^2 \rangle_t} \quad (2.4)$$

ここで  $\langle \dots \rangle$  は時間平均を示す。使用方法は以下の通りである。

corr.exe の使い方

`corr.exe` [-s skip] [入力ファイル] > [出力ファイル]

ここで `skip` には、相関を計算せずにスキップする値を入力する。デフォルトは 100 になっている。入力ファイルは `gt2et.exe` や `gt2gw.exe` と同様に、以下のようにになっている必要がある。

corr.exe のための入力ファイル書式

```
G(0)
G(dt)
G(2dt)
... G( datanum dt)
```

出力は、時間刻み  $dt$  で、ステップ数と式 2.4 の  $C(\tau)$  を出力する。

### 2.4.6 naplesgwOCTA(.exe)

実験と同様の方法で正弦波形の動的せん断ひずみを、周波数を変化させながら加え、得られた結果を `getggval.exe` (2.4.8 節) により解析して、最終的に複素弾性率  $G^*(\omega)$  (ファイル名 `gwdirect.npls:5.10` 節) を得るまでの処理を自動化した実行モジュール。

## — naplesgwOCTA(.exe) の使い方 —

```
Usage: naplesgwOCTA(.exe) ([-options] values) npls file]
    -wmax [the maximum frequency to be scanned: default 1.0]
    -wmin [the minimum frequency to be scanned: default 0.001]
    -div [number of divisions for one decade: default 10]
    -amp [strain amplitude: default 0.3]
    -cycle [strain cycle to be scanned: default 3.0]
    -input [input file (.npls format) : default input.npls]
    -config [configuration file to be read as the initial configuration. ]
    -prerun [pre-thermalization steps if the equilibrated configuration
            does not exist.: default 1000 ]
    -output [output file for G' G'' against w: default gwdirect.npls]
    -engine [engine of naples for calculation: default naplescalc.out]
```

## — naplesgwOCTA(.exe) の使用例 —

```
> cd [複素弾性率を計算するサンプル情報のある input.npls ファイルがある場所へ移動]
> naplesgwOCTA(.exe) -input test.npls -output gwout
```

上記の例では、test.npls にある入力に基づき、周波数域 1.0 から 0.001 までひずみを周波数スキャンし、 $G'$  と  $G''$  を計算して gwout ファイルに出力する。計算には naples.out をエンジンとして用いる。(ただし naplesOCTA(.exe) にパスが通っていない場合にはエラーとなる。)

## 2.4.7 naplesgwOCTA(.exe) の UDF を用いた実行方法

例えば、NAPLES が D ドライブの直下にインストールされている場合、以下のようにする。naplesgwOCTA.bat (Linux 版では.sh) を実行すると、パラメータで指定された UDF 形式の Input ファイルを NAPLES オリジナルの Input ファイルに変換した後に naplesgwOCTA(.exe) を実行する。naples.input\_converter(.exe), naplesgwOCTA(.exe), naplesOCTA(.exe), getggval(.exe) へのパスが通っている必要がある。引数の内容は、2.4.6 節に従う。

## — UDF ファイルを Input にした場合の実行方法 —

```
>cd d:/NAPLES          「NAPLES フォルダに移動」
>naplesgwOCTA.bat(Linux 版では.sh) [入力 UDF ファイル名] [naplesgwOCTA への引数]
```

## — Gourmet から EngineRun パネルを用いて実行する場合の設定例 —

```
Server: localhost
Engine: C:\OCTA2006Rev1\PF_ENGINE_2006\bin\win32\naplesgwOCTA.bat
Working Dir: C:\OCTA2006Rev1\PF_ENGINE_2006\Naples_for_OCTA\sample_input_udf\output_gw
Params:

Input UDF: C:\OCTA2006Rev1\PF_ENGINE_2006\Naples_for_OCTA\sample_input_udf
           \input_dynamic_shear_for_gw.udf
以下、指定なし
```

※入力 UDF および出力ファイル群は、Working Dir で指定したディレクトリに書き込まれる。

### 2.4.8 getggval(.exe)

応力結果ファイル `measure.npls` から、ある周波数での  $G'$ ,  $G''$  を求める。当然ながら、このときの `measure.npls` は動的せん断モード (4.6.6 節) での計算が行われたものでなければならない。利用法は以下である。

— getggval.exe の使い方 —

```
> getggval.exe [周波数の値] [ひずみ振幅の値] measure.npls > [周波数] [G'] [G'']
```

得られる  $G'$ ,  $G''$  は、その周波数における値だけであり、動的粘弾性カーブを得ようとする場合には 3.3.3 節の方法を用いるなどして、周波数を変化させた計算を繰り返し行い、この解析をとおす必要がある。

このコマンドでは以下のような計算原理に基づいている。まず、正弦波形のひずみ  $\gamma(t) = \gamma_0 \sin(\omega t)$  を与えた場合、得られる応答 (応力)  $\sigma$  が線形であると仮定する。すなわち、以下の関数でかけると仮定する。

$$\sigma(t) = \sigma' \sin(\omega t) + \sigma'' \cos(\omega t) \quad (2.5)$$

図 3.10 の例にあるように、実際に得られる応力には揺らぎ等の効果で式 2.5 にぴったりと合ったものにはならない。そこで最小二乗法 (誤差がもっとも小さくなるようなフィッティング法) を用いて、実際の測定応力を式 2.5 にフィッティングし、そこで得られる  $\sigma'$  および  $\sigma''$  を計算し、ひずみ  $\gamma_0$  で割って出力するのが `getggval.exe` の機能である。弾性率の定義式から、 $G = \sigma/\gamma$  であるから、 $G' = \sigma'/\gamma_0$ ,  $G'' = \sigma''/\gamma_0$ , として出力している。

## 第3章 レオロジー計算と解析の実際

NAPLES は分子の運動を与えられた流動や変形条件の下で計算し、その分子の運動から応力を求める。(要するに実験と同じである。) 実験に使用するレオメーターでは、サンプルに流動(変形)を与えた結果出力される応力を自動的に粘度や弾性率、貯蔵弾性率  $G'$  や損失弾性率  $G''$  に変換してくれる機構をもっている。NAPLES も同様に応力を出力し、ユーザーによる解析によって種々の値を得る。

解析を始める前に当然データファイルが必要なので、2.2 節に従ってレオロジー計算を行う。以降この節では、該当する計算が終了し、measure.npls ファイルが得られているものとして解析方法を説明する。本節では、レオロジー計算エンジンを用いたレオロジーの計算法について、解析法まで含めて述べる。

ここでは NAPLES オリジナルの入力ファイルを用いた計算法を説明するが、UDF 形式の入力ファイルを用いた場合も、同じ名前のパラメータを設定することで同様な計算を実施することが可能である。その場合の計算法は 2.3 節を参照のこと。

また、UDF 形式の入力ファイルを用いた場合には各出力ファイルは入力 UDF ファイルからリンクされる。その際、入力 UDF ファイル上のアクション機能によって、以下に紹介する各解析機能呼び出すことが可能である。利用方法は *Naples* 計算実行後、実行ディレクトリに保存される *Naples UDF* ファイルを開き、ボールド表示されている *Output* を右クリックしてメニューを選択する。(アクション機能の詳細は *Gourmet* のマニュアルを参照のこと)

アクション機能は以下の解析に対応している。

アクション機能

- uniaxal\_stress 解析とプロット
- biaxal\_stress 解析とプロット
- planar\_stress 解析とプロット
- shear\_stress 解析とプロット
- Freq\_Scan プロット

また、これらの解析結果を実単位に変換するオプションを備えている。

### 3.1 ステップ変形からの応力緩和と緩和弾性率の算出

#### 3.1.1 計算の実行と入力ファイル例

ステップ変形をかけた計算を行う詳細は 4.6.4 節を参照。NAPLES では、せん断応力と法線応力を measure.npls に書き込んでいるので、望みの応力値を時間に対してグラフ化する。以下のような入力パラメーターファイルで計算したときのデータを例に説明する。

```
cell_size=12
```

```
seed=87911
quit_dynamics_count=4000
component_number=1
molecule_spec1=linear
Z1=10
fraction1=1.0
flow_type=step_shear
initial_strain=1.0
pre_reptation_step=100
```

### 3.1.2 結果（応力）の表示

計算の結果得られる `measure.npls` において、時間を記録している1列目とせん断応力を記録している3列目をグラフ化すると図3.1のようになる。

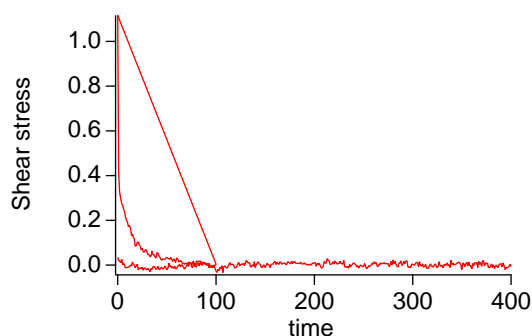


図 3.1: ステップせん断における応力緩和の例：データ列をそのまま出力

ステップ応力緩和の計算では、通常 `pre_reptation_step` で指定された平衡化時間があり、その後変形がかけられた時刻から応力が徐々に減衰する。NAPLES の仕様では `pre_reptation_step` 後、変形がかけられた時刻で計算の時間がリセットされ、再び0となるので、まず表計算ソフトウェアなどで平衡化を行っている期間のデータを取り除く。

この例では、`pre_reptation_step=100` としているので、その部分を切り取ると図3.2となる。

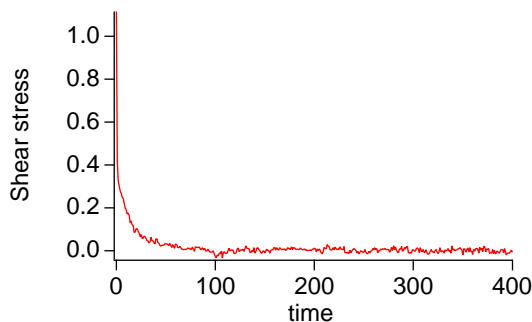


図 3.2: ステップせん断における応力緩和の例：平衡化している部分をカット



### 3.1.3 データのスムージング

レオロジーデータは一般に両対数グラフで解析される．一方 NAPLES では出力データが時間に対して等間隔で出力されている．このようなデータは長時間側でグラフが見つらくなるので，ツールを用いてデータをスムージングする．これを行うには以下の操作が必要である．

まず `measure.npls` ファイルから，スムージングを行いたいデータ列を時間データとともに抜き出して別のファイルを作成する．このようなファイルができれば，スムージングソフトウェアの `smooth.exe` (2.4.3 節) を用いてデータをスムージングする．これを行うには，スムージングを行うデータ列と時間列を書き込んだデータファイルをまず作成する．

Windows ならば EXCEL などの表計算ソフトウェアに `measure.npls` ファイルを読み込んで 1 列目の時間列とスムージングしたいデータ列だけを抽出して作成する．UNIX ならば `awk` を用いておこなうことができる．

— `awk` を使った時間とせん断応力の抽出 —

```
>awk '{printf("%e %e\n",$1,$3)}' <measure.npls> [切り出したデータを保存するファイル名]
```

ここで \$1 が時間列 (1 列目) を示す．この例では，時間列と \$3 (3 列目) のせん断応力を切り出しているが，これを適宜 \$4, \$5 などとすることによって法線応力を切り出すこともできる．

次に，切り出したデータファイルを `smooth.exe` を用いてスムージングする．

— `smooth.exe` を使ったデータのスムージング —

```
smooth.exe [スムージングしたいファイル名] > [結果ファイル名]
```

例えば，データファイルである `measure.npls` が D ドライブの Data フォルダにあり，`smooth.exe` が D ドライブの NAPLES フォルダ内の Tools フォルダにある場合には以下のように入力する．

— `measure.npls` からのせん断応力値の取り出しとスムージングの例 (cygwin) —

```
>cd /D:/Data           「データフォルダへ移動」
>gawk '{printf("%e %e\n",$1,$3)}' <measure.npls> s12      「せん断応力のデータを取り出し」
>D:/NAPLES/Tools/smooth.exe s12 > s12sm                「スムージング」
```

この例では，ファイル `s12` にせん断応力の時間変化の生データ，`s12sm` にそのスムージングされたデータが出力される．

結果をスムージング前のデータとともに図 3.3 に示す．

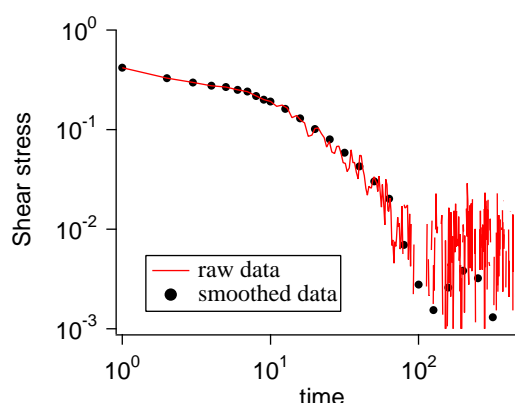


図 3.3: ステップせん断における応力緩和の例：スムージングした結果（両対数で表示）

### 3.1.4 応力から弾性率への変換

弾性率  $G$  は、与えたひずみを  $\gamma$  として応力  $\sigma$  と以下の関係にある。

$$G = \frac{\sigma}{\gamma} \quad (3.1)$$

この関係に従い、様々なひずみを与えて緩和弾性率  $G(\gamma, t)$  を計算した結果を図 3.4 に示す。

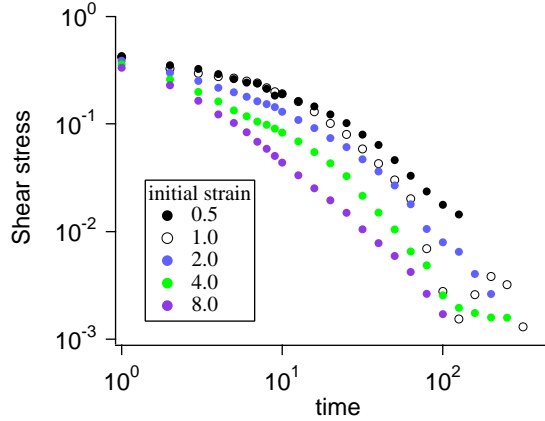


図 3.4: 応力緩和カーブの例

緩和弾性率を計算できればダンピング関数を求めることもできる。ダンピング関数とは、長時間領域において緩和弾性率  $G(\gamma, t)$  が、ひずみのみによる成分  $h(\gamma)$  と時間のみによる成分  $G(t)$  に分離できる ( $G(\gamma, t) = h(\gamma)G(t)$ ) ときのひずみ依存関数  $h(\gamma)$  である。これを求めるには、Hency ひずみで 0.5 以下の小さなひずみをかけた計算により線形緩和弾性率  $G(t)$  を算出し、 $G(\gamma, t)$  を割ればよい。ある特定の時間  $\tau_k$  よりも長時間側では、 $G(\gamma, t)/G(t)$  の値が一定になるので、この値をとる。

なお、線形緩和弾性率の値を求める際にはできるだけ与える変形を小さくする必要があるが、実験と同様に変形が小さいと検出される応力も小さくなるため、相対的に誤差が大きくなる。実用に耐える最小ひずみ量は 0.1-0.2 程度である。それでも 7.2 節の手段で計算精度を向上させる工夫をしなければ緩和弾性率を求めるには厳しい。ひずみ量 0.5 のときの値を線形緩和弾性率としても計算誤差の範囲で正当性がある。それ以上の精度でどうしても求めたいときは、3.2 節で述べる方法で平衡状態の計算を長時間行うことでも得ることができる。

## 3.2 線形緩和弾性率を平衡状態の計算から求める

極小のひずみを与えたときの応力の緩和は、まったくひずみを与えずに（つまり平衡状態で）計算した応力の時間自己相関関数に等しい。この原理を利用すると線形緩和弾性率を平衡状態の計算からもとめることができる。

まず解析したい試料の入力パラメーターファイルを作成し、`flow_type=equilibrium` として平衡状態の計算を長時間行う。このとき”長時間”というのは、最長緩和時間の 10 倍以上、できれば 100 倍程度が望ましい。（最長緩和時間の見積もり方については 3.8.1 節を参照。）最長緩和時間の 100 倍程度の時間を `quit_dynamics_count` パラメーターに指定し、計算を行う。

この方法では、非常に長時間の計算となる。また最長緩和時間に到達しない計算結果に基づいて以下の操作を行うと、意味の無い結果しか返さない。そのため、3.1 節で説明した、せん断変形を与えて緩和弾性率を計算する方法と、複数回の計算内容を平均する方法を組み合わせる方が有効な場合も多いと思われる。しかし、本節で紹介する方法で求められる緩和弾性率が、いわば厳密な定義にのっとり揺らぎから求めた真の緩和弾性率であることには注意されたい。

### 3.2.1 計算方法

マニュアルで求めるには、下記の処理に従う。まず `measure.npls` に出力される応力から、3 列目のせん断応力や、4 列目の法線応力など、望みの応力値だけを取り出したファイルを作成する。この操作は EXCEL などの表計算ソフトウェアに `measure.npls` ファイルを読み込んで応力データを抽出して作成する。あるいは、UNIX または cygwin 環境であれば `awk` を用いてもおこなうことができる。

— `awk` を使ったせん断応力の抽出 —

```
>awk '{printf("%e\n",$3)}' <measure.npls> [切り出したデータを保存するファイル名]
```

この例では、\$3（3 列目）のせん断応力を切り出しているが、これを適宜 \$4, \$5 などとすることによって法線応力を切り出すこともできる。

次に、数列の自己相関関数を計算するプログラム `corr.exe` を使用する。<sup>1</sup> 使用するには、応力値の時間変化だけがかかれたファイルを入力として用いる。

— 相関関数を求める `corr.exe` の使い方 —

```
corr.exe [-s スキップするステップ] [応力だけが出力されたファイル名] > [結果ファイル名]
```

”スキップするステップ”とは、計算開始直後、系が平衡状態になるまでのステップのことである。このステップ数は、分子量や分子の形態により異なる。厳密には系の最長緩和時間までは平衡化される保証がないが、現実にはそこまで待てないので、`measure.npls` に出力される応力値の時間変化を見て、平衡化されたと判断される時間を決める。もし何の手がかりも無い場合は、推奨値は 1000 である。この値は分子量  $Z=30$  程度の分子の緩和時間に相当する。

結果として出力されたファイルは線形緩和弾性率を示すカーブになっている。ここで得られた線形緩和弾性率カーブは、以下の方法でスムージングすることができる。

— 緩和弾性カーブのスムージング —

```
smooth.exe [corr.exe の結果ファイル] > [スムージングされたデータのファイル名]
```

例えば、データファイルである `measure.npls` が D ドライブの Data フォルダにあり、`corr.exe` と `smooth.exe` が D ドライブの NAPLES フォルダ内の Tools フォルダにある場合には以下のように入力する。

— `measure.npls` からのせん断応力値の取り出しと相関関数計算の例 (cygwin) —

```
>cd d:/Data          「データフォルダへ移動」
>gawk '{printf("%e%e\n",$1,$3)}' <measure.npls> s12      「せん断応力のデータを取り出し」
>d:/NAPLES/Tools/corr.exe -s 1000 s12 > s12corr        「相関関数の計算」
>d:/NAPLES/Tools/smooth.exe s12corr > s12tsm          「スムージング」
```

この例では、ファイル `s12` にせん断応力の時間変化の生データ、`s12corr` にその相関関数、`s12tsm` にスムージングされたデータが入る。

実際の計算例を示す。まず  $Z=40$  の直鎖高分子の系で、平衡状態の計算を行ったときの応力の時間変化は図 3.5 のようになっている。

<sup>1</sup>`corr.exe` の詳細については 2.4.5 節を参照

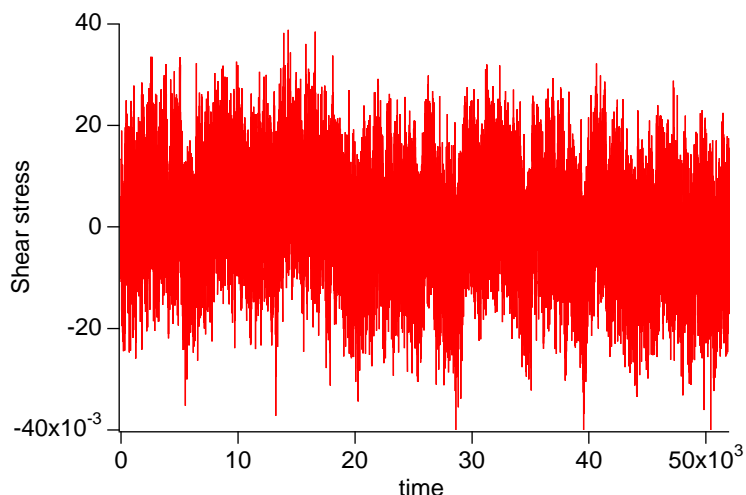


図 3.5: 平衡状態におけるせん断応力のゆらぎ

このデータから corr.exe を用いて相関関数を計算し、その結果をステップせん断変形を加えたときの緩和弾性率と同時に示すと図 3.6 となる。長時間側でまだ精度が不足している面があるが、このようにステップせん断変形の結果と相関関数を求めた結果は等しい。

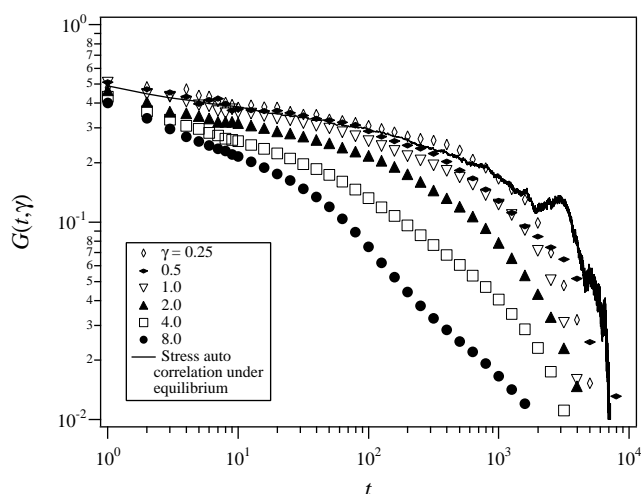


図 3.6: 平衡状態から得られた相関関数とステップせん断変形で得られた緩和弾性率の比較

### 3.3 G'G''を計算したい

線形緩和スペクトルである  $G'$  (貯蔵弾性率) および  $G''$  (損失弾性率) を求めるための方法は2通りある。

一つは、線形緩和弾性率を求めてフーリエ変換する方法である。この方法は短時間側での精度が高く、全プロファイルを一気に求めることができる利点があるが、逆に長時間側の精度は低くなる傾向があり、また最長緩和時間以上の計算をしなければ意味のある結果をえることができない。

もう一つの方法は、実験と同様に正弦波形で時間変化する微小ひずみをあたえ、その結果得られる応答を解析する方法である。この方法は最長緩和時間が不明な系にも容易に適用できる。しかし低周波側での値を求めるための計算は非常に時間を要する場合がある。

本節ではそれぞれの方法を説明していく。

### 3.3.1 線形緩和弾性率から求める場合

まず, 3.1 または 3.2 節により, 線形緩和弾性率を求める。次に, ツール gt2gw.exe を用いて  $G'(\omega)$  および  $G''(\omega)$  を得る。

— gt2gw.exe を用いた G'G'' の算出法 —

```
gt2gw.exe <[線形緩和弾性率のファイル名]> [結果ファイル名]
```

gt2gw.exe は線形緩和弾性率ファイル名を引数として受けるのではなく, リダイレクトを利用するので注意が必要である。これにより出力されるファイルには, 第 1 列目に周波数 (単位は  $2\pi/\tau_e$ ), 2 列目に  $G'$ , 3 列目に  $G''$  が出力されている。なお線形緩和弾性率のデータ点多すぎるとエラーが出る場合があるので, その場合には gt2gw.exe -n 22 というオプションを利用する。(詳しくは 2.4.2 節を参照) さて, 変換を行った後データ点多量に出力されているので, 周波数軸で平均をとり, スムージングする。スムージングには gwsMOOTH.exe を利用する。

— gwsMOOTH(.exe) による G'G'' データのスムージング —

```
gwsMOOTH.exe [gt2gw.exe の結果ファイル] > [結果ファイル名]
```

例えば, 線形緩和弾性率 (スムージングされていないもの) がファイル s12t で D ドライブの Data フォルダにあり, gt2gw.exe と gwsMOOTH.exe が D ドライブの NAPLES フォルダ内の Tools フォルダにある場合には以下のように入力する。(線形緩和弾性率のファイルを得る方法は 2 つあるのでそれぞれ 3.1 節や 3.2 節を参照のこと。)

— 線形緩和弾性率からの G'G'' の計算例 (cygwin) —

```
>cd \d:\Data          「データフォルダへ移動」
>.\NAPLES\Tools\gt2gw.exe <s12t> s12w          「フーリエ変換」
>.\NAPLES\Tools\gwsMOOTH.exe s12w > s12ws      「スムージング」
```

この例では, ファイル s12w の 1 列目が周波数  $\omega$ , 2 列目が貯蔵弾性率  $G'(\omega)$ , 3 列目が損失弾性率  $G''(\omega)$  になる。

なお, gwsMOOTH.exe はスムージングソフトウェアである smooth.exe を利用しているので, smooth.exe と gwsMOOTH.exe は同じディレクトリにあることを確認する。

— gt2gw.exe を用いて G'G'' を算出する際の注意 —

gt2gw.exe を適用する, 線形緩和弾性率が記録されているファイルは, スムージング前のもの (弾性率の時間変化が等時間間隔で記録されているもの) でなければならない。スムージングしてしまった緩和弾性率から G'G'' を求めるプログラムはパッケージに含まれていない。

### 3.3.2 線形緩和弾性率からの計算例

実際に計算例を示す。まず計算には以下の入力インプットを用いた。

```
cell_size=10
C=10
p2=0.01
seed=845
```

```
quit_dynamics_count=50000
component_number=1
flow_type=equilibrium
molecule_spec1=linear
Z1=8
fraction1=1.0
calculation_mode=new
```

この計算から得られた線形緩和弾性率のカーブを図 3.7 に示す.

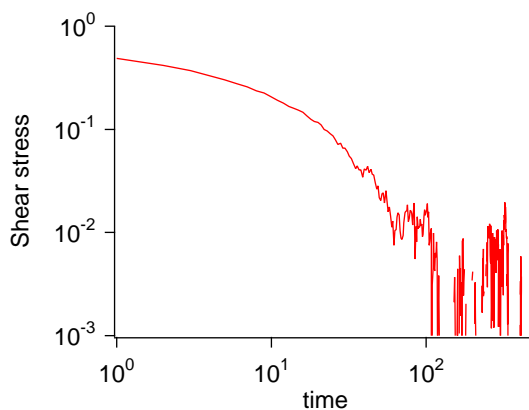


図 3.7:  $Z=8$  の直鎖高分子の線形緩和弾性率

この段階ではスムージングは行っていない. このデータを使って  $G'$  $G''$  を計算した結果を図 3.8 に示す.

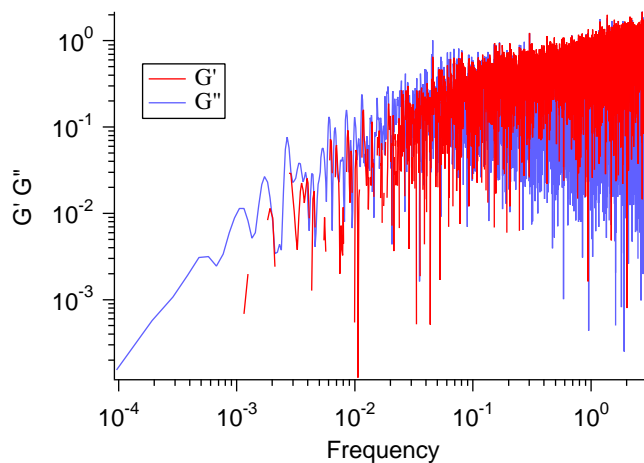


図 3.8:  $Z=8$  の直鎖高分子の  $G'$  $G''$

図 3.8 の結果を gwsMOOTH.exe によりスムージングした例を図 3.9 に示す.

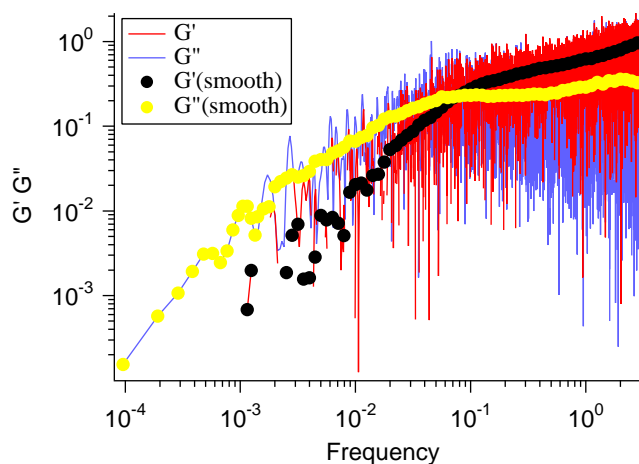


図 3.9: Z=8 の直鎖高分子の G'G'' (スムージングとの比較)

### 3.3.3 動的粘弾性試験を行う場合

input.npls ファイルを 4.6.6 節に従って作成する。この入力ファイルに基づいて NAPLES を実行しただけでは、設定した周波数での動的なひずみに基づく応力の時間変化が得られるだけである。図 3.10 に、動的せん断ひずみを与えたときの応力の時間変化の例を示す。

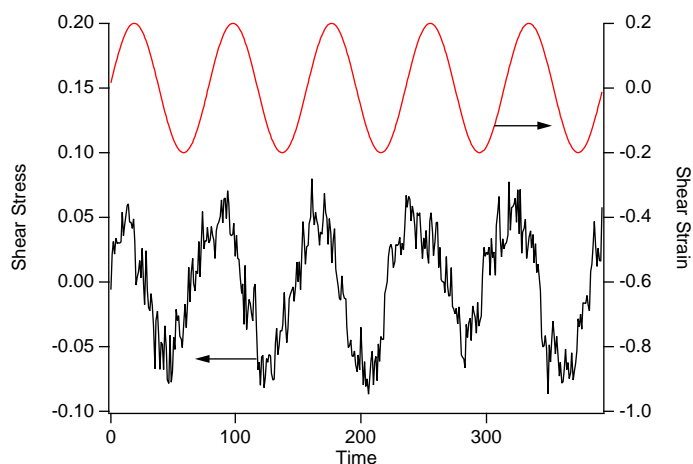


図 3.10: 動的せん断ひずみを与えた時の応力の時間変化

得られた応力の時間変化  $\sigma(t)$  を最小二乗法により

$$\sigma(t) = \sigma'(\omega) \sin \omega t + \sigma''(\omega) \cos \omega t \quad (3.2)$$

の式にフィッティングし、 $G = \sigma/\gamma$  の定義に従って、 $G'(\omega) = \sigma'/\gamma_0$ 、 $G''(\omega) = \sigma''/\gamma_0$ 、として求めたものが  $G'$  および  $G''$  である。図 3.10 のようなデータファイル measure.npls から、 $G'$  および  $G''$  を求めるコマンドは getggval(.exe) であり、以下のように使用する。

— measure.npls からの  $\sigma'$ ,  $\sigma''$  の抽出 —

```
>getgval.exe [周波数] [ひずみ振幅] measure.npls
>[周波数] 0.1537 0.2235
```

ここで [周波数] には、計算に用いた周波数 (input.npls ファイルにおける strain\_omega の値) を入れる。出力結果は、周波数,  $G'$ ,  $G''$  の順である。

このような処理を重ねて、ようやくある周波数における  $G'$ ,  $G''$  の一組がわかるだけである。つまり、一回の NAPLES 起動で、グラフ上の2点が求まるに過ぎない。広範な周波数領域での計算を手動でおこなうのは非常に手間がかかるため、周波数を変化させながら自動計算するためのモジュールが提供されている。次節をご覧ください。

### 3.3.4 動的粘弾性を連続実行で行う場合

input.npls ファイルを4節に従って作成する。input.npls ができたら、naplesgwOCTA(.exe) を実行する。コマンドの詳細は2.4.6節を参照のこと。その結果、gwdirect.npls ファイル (5.10節) が生成される。

### 3.3.5 スキャン周波数範囲をかえるには

naplesgwOCTA(.exe) コマンドが利用できる場合には、起動時のパラメーターとして計算対象となる周波数などを簡単に与えることができる。(2.4.6節を参照。)

## 3.4 スタートアップ流れによる応力成長

### 計算の実行と入力ファイル例

スタートアップ流れによる計算を行うための方法は4.6.5節を参照。ここでは計算終了後の解析方法を述べる。以下の入力パラメーターで一軸伸長流れにおけるポンポン形高分子の応力成長を計算した結果を例にして説明する。

```
cell_size=12
C=10
p2=0.01
seed=4799
quit_dynamics_count=3000
flow_type=uniaxial_startup
strain_rate=0.001
pre_reptation_step=300
component_number=1
molecule_spec1=pompom
armZ1=3
arm_per_junction1=4
bridgeZ1=10
fraction1=1.0
```



### 3.4.1 結果解析

一軸伸長応力は、 $\sigma_{11} - \sigma_{22}$  で定義されるので、measure.npls ファイルにおいて 4 列目の数値から 5 列目の数値を引いた数値を作成する。EXCEL などの表計算ソフトウェアに measure.npls ファイルを読み込んで 1 列目の時間列と、4 列目の数値から 5 列目の数値を抽出し、その後 4 列目から 5 列目を引いた数値を作成する。この操作を awk を用いておこなうには、以下のように入力する。

— awk を使った時間と法線応力差ファイルの作成 —

```
>awk '{printf("%e%e\n", $1, $4-$5)}' <measure.npls> [切り出したデータを保存するファイル名]
```

ここで \$1 が時間列（1 列目）を示す。\$4-\$5 で、4 列目と 5 列目の差を計算している。レオロジーデータは一般に両対数グラフで解析される。このとき、NAPLES のように出力データが均等だと長時間側でグラフが見づらくなるので、ツールを用いてデータをスムージングする。これを行うには、まず measure.npls ファイルから、スムージングを行いたいデータ列を時間データとともに抜き出して別のファイルを作成する。（どのようなファイルを作成すればよいかは 2.4.3 節を参照。）このようなファイルができれば、スムージングソフトウェアの smooth.exe を用いてデータをスムージングする。

— smooth.exe によるスムージング —

```
smooth.exe [スムージングしたいファイル名] > [結果ファイル名]
```

例えば、データファイルである measure.npls が D ドライブの Data フォルダにあり、smooth.exe が D ドライブの NAPLES フォルダ内の Tools フォルダにある場合には以下のように入力する。

— measure.npls からの法線応力値の取り出しとスムージングの例（cygwin） —

```
>cd d:/Data          「データフォルダへ移動」
>gawk '{printf("%e%e\n", $1, $4-$5)}' <measure.npls> >se      「法線応力のデータを取り出し」
>d:/NAPLES/Tools/smooth.exe >se >sem                        「スムージング」
```

この例では、ファイル se にせん断応力の時間変化の生データ、sem にそのスムージングされたデータが出力される。

### 3.4.2 計算例

実際に作成したものをプロットすると図 3.11 のようになる。

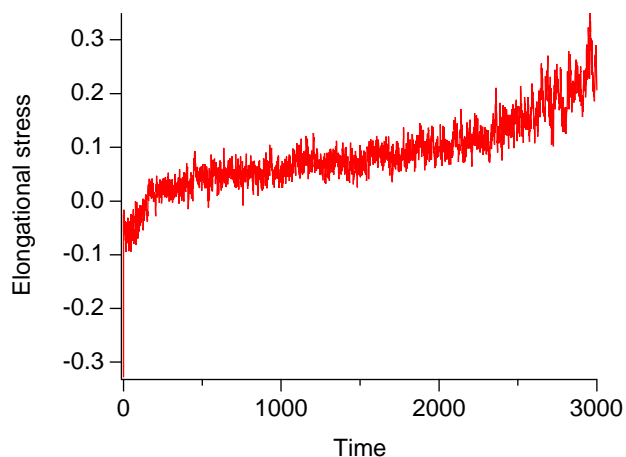


図 3.11: 一軸伸長における応力成長の例：データをそのまま出力

NAPLES では、せん断応力と法線応力を `measure.npls` に書き込んでいるので、望みの応力値を時間に対してグラフ化する。スタートアップ流れの計算では、通常 `pre_reptation_step` で指定された平衡化時間があり、その後流動が始まった時刻から応力が徐々に増加する。NAPLES の仕様では `pre_reptation_step` 後、流動が始まった時刻で計算の時間がリセットされ、再び 0 となるので、まず表計算ソフトウェアなどで平衡化を行っている期間のデータを取り除く。このような処理を行ったデータを図 3.12 に示す。

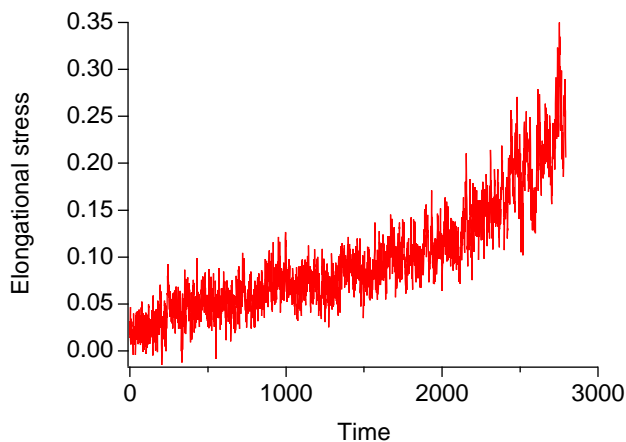


図 3.12: 一軸伸長における応力成長の例：平衡化部分を削除

なお、スタートアップ流れの計算を行うに当たっては、伸長変形ひずみをできるだけ稼ぐために、あらかじめ負のひずみがかかった、立方体でないセルから計算をスタートさせる。そのため、`pre_reptation_step` による平衡化が十分でないと、負の伸長応力が検出されることがある。ここまでの処理で伸長応力がゼロ以下からスタートしているようならば、平衡化時間が不足しているので、入力ファイルにおいて `pre_reptation_step` の値を増やし、計算をやり直した方がよい。

図 3.12 の結果を `smooth.exe` を用いてスムージングし、スムージング前のグラフとともに表すと図 3.13 のようになっている。

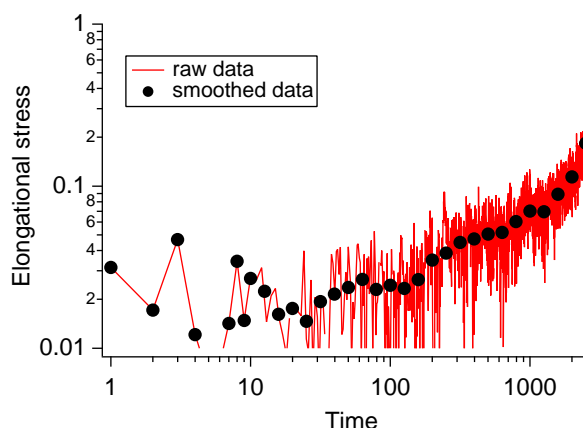


図 3.13: 一軸伸長における応力成長の例：スムージング結果とともに表示

## 3.5 線形粘度成長曲線

非常にゆっくりとした速度で流したときに得られる線形の応力成長曲線を得ることは、実験と同様に得られる応力がきわめて小さくなるので誤差が大きくなり難しい。このときは 3.1 節や 3.2 節で求めた線形緩和弾性率の曲線から計算する。

### 3.5.1 結果解析

長時間の平衡状態での計算結果ファイル `measure.npls` が得られているものとする。

手動で行うには、まず 3.1 節や 3.2 節に従って線形緩和弾性率のファイルを作成し、その後 `gt2et.exe` ツールを用いて得る。

— `gt2et.exe` による線形応力成長カーブの算出 —

`gt2et.exe` [線形緩和弾性率のファイル名] > [線形応力成長を出力するファイル名]

ここで得られたデータもスムージングできる。例えば、線形緩和弾性率（スムージングされていないもの）がファイル `s12t` で D ドライブの Data フォルダにあり、`gt2et.exe` と `smooth.exe` が D ドライブの NAPLES フォルダ内の Tools フォルダにある場合には以下のように入力する。（線形緩和弾性率のファイルを得る方法は 2 つあるのでそれぞれ 3.1 節や 3.2 節を参照のこと。）

— 線形緩和弾性率からの線形応力成長関数の計算例 —

```
>cd d:/Data           「データフォルダへ移動」
>d:/NAPLES/Tools/gt2et.exe <s12t> e12t           「変換」
>d:/NAPLES/Tools/smooth.exe e12t > e12ts         「スムージング」
```

この例では、`e12t` に線形応力成長関数、`e12ts` にそのスムージングされた数列が入っている。

## 3.6 粘度の求め方

まずスタートアップ流れにおける計算（4.6.5 節）を行い、応力成長曲線を得る（3.4 節）。粘度はスタートアップ流れ（定常流れ）場において、応力をひずみ速度（せん断速度あるいは伸長速度）で割ったものとして定義される。例えば定常せん断流れとして以下のように流れ場を設定したとする。

```
flow_type=start_up_shear  
strain_rate=0.001
```

この場合にはせん断速度は 0.001 であるから、measure.npls ファイルの 3 列目に出力されるせん断応力値を 0.001 で割った値が NAPLES における無次元化されたせん断粘度になる（この段階では単位はない）。ここで得られた粘度の値を実験で得られる実際の値に換算するには、NAPLES の応力単位と NAPLES の時間単位を掛け合わせる必要がある。具体的には 3.8 節をごらんいただきたい。ただし、定常せん断粘度を求めるには、実験同様に当然ながらしばらく計算を行って応力が定常値になるまでまって、適当な時間で平均をとって応力値を求め、それをせん断速度で割って粘度とする。（3.7 節を参照。）また線形の粘度成長曲線を得るには、3.4 節を参照のこと。

### 3.7 流動曲線（粘度のせん断速度、伸長速度依存性

3.6 節を参照してスタートアップ流れ場における計算を行い、粘度の成長曲線を得る。例として、以下のように入力パラメーターを用いた場合を説明する。

```
cell_size=8  
C=10  
p2=0.01  
seed=98981  
quit_dynamics_count=2000  
pre_reptation_step=100  
component_number=1  
molecule_spec1=linear  
Z1=17.13  
fraction1=1.0  
flow_type=start_up_shear  
strain_rate=0.0013
```

strain\_rate 値を種々変化させたパラメーターファイルをいくつか作成してそれぞれに計算を行い、粘度成長曲線を得た結果を図 3.14 に示す。

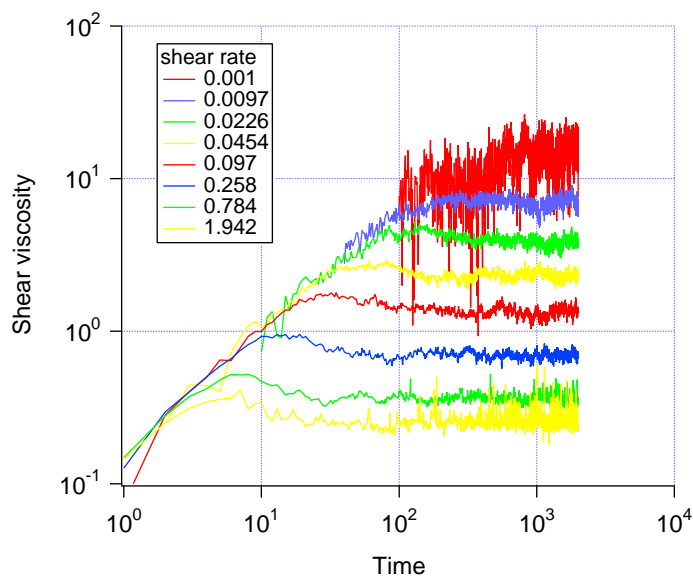


図 3.14: スタートアップせん断における粘度成長曲線の例

長時間領域では、粘度が時間に対して定常値となっている．この時間領域で粘度の時間平均をとり、せん断速度に対する粘度値をもとめる．この粘度値が、定常粘度の値である．図 3.15 に、そのようにして求めた定常粘度のせん断速度依存性を示す．

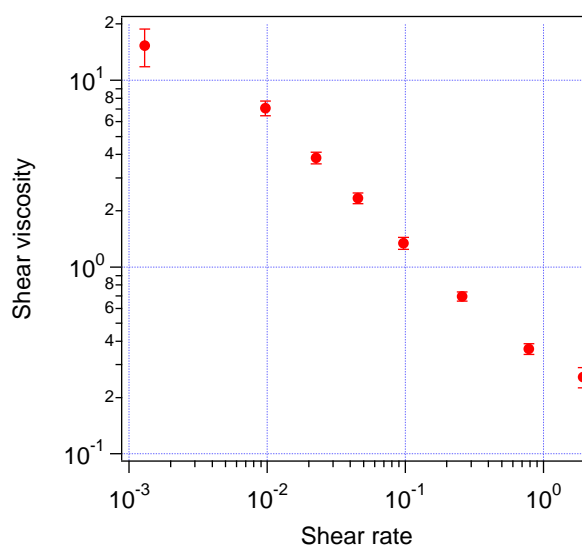


図 3.15: 定常粘度のせん断速度依存性の例

ここで得られた粘度の値を実験で得られる実際の値に換算するには、NAPLES の応力単位と NAPLES の時間単位を掛け合わせる必要がある．また流動速度を実時間のスケールに変換するには、NAPLES の単位時間で割る必要がある．具体的には 3.8 節をごらんいただきたい．

なお伸長流動下では、高分子の有限鎖長効果 (FENE 効果) などの設定を行わないと、定常粘度を得ることができない (各パラメータの設定は 4.8 節を参照のこと)．また計算に用いる周期境界条件のセルの大きさを十分大きく (最低でも `cell.size=16` 以上) とらないと、定常状態に達する前に計算が終了する．

### 3.8 実験とのフィッティング

計算が終わったあとのデータを実験値に対応させるためには、measure.npls などの結果ファイルに出力された無次元化された時間と応力の値を秒やパスカルなどの実単位をもつ値に変換しなければならない。そこで以下の、3.8.1 節および 3.8.2 節では、NAPLES における時間と応力を実単位に変換する方法について説明する。なおこれらの値を変えてフィッティングしてもうまくいかない場合には、 $Z$  の決め方に問題がある可能性がある。必要ならば 4.5 で入力条件を再考する。

Gourmet 上の Naples アクションでは実単位への変換機能をサポートしている。

#### 3.8.1 NAPLES の単位時間

$\tau_e$  は式では以下のようにかける。

$$\tau_e = \frac{\zeta a^2}{6kT} \quad (3.3)$$

ここで  $a$  は、からみあいチューブセグメントの平均長さ、 $k$  はボルツマン乗数、 $T$  は絶対温度である。 $\zeta$  はからみあいチューブセグメントの摩擦係数である。これらの値は、メルトの場合には分子の種類だけ決まれば分岐形状や分子量にはよらない。いくつかの高分子についてはデータベースもある。ただし、 $\zeta$  は温度にもよるためデータベース化することが難しい。

そこで、現実的には以下の方法を用いる。

— NAPLES における単位時間  $\tau_e$  を実時間と対応させる方法 —

1. まず、計算したい高分子の化学種で、直鎖の試料でのレオロジーデータを文献等から見つけるか実際に測定する。分子量や分岐構造、分子量分布は計算しようとするものと同じでなくてよい。ただし溶液の場合には濃度は等しくなければならない。レオロジーデータは何でもよいが、ステップ応力緩和弾性率か動的粘弾性スペクトルデータ（いわゆる  $G'G''$  データ）、あるいは線形領域が示されたスタートアップせん断粘度か伸長粘度がよい。
2. これらのデータから最長緩和時間  $\tau_d$  を抜き出す。ステップ応力緩和の場合には、応力値または緩和弾性率値が、変形直後の値に比べて  $1/e$  になるまでの時間を  $\tau_d$  としてよい。 $G'G''$  の場合には、 $G'(\omega) \propto \omega^2$ ,  $G''(\omega) \propto \omega$  になる  $\omega$  から  $2\pi/\omega$  で決まる時間、あるいは複素粘度  $\eta^*$  が一定になる  $\omega$  から  $2\pi/\omega$  で決まる時間、または大体の目安で  $G'G''$  が交差する  $\omega$  から  $2\pi/\omega$  で決まる時間でよい。スタートアップせん断粘度か伸長粘度の場合には、十分ゆっくりとした速度で流した場合にあわられる線形カーブで、粘度が一定になるニュートン流動領域が現れるまでの時間でよい。
3. レオロジーデータが測定されたサンプルの  $Z$  値を 4.5 節に従って求める。
4. NAPLES における予測最長緩和時間  $\tau_e$  を以下の式により求める。

$$\tau_e = 0.0025Z^{3.6} \quad (3.4)$$

5.  $\tau_e = \tau_d/\tau_e$  となる。

ここまでの手順で概算値を求めることができるが、より正確な値を出したい場合には以下の手順を追加する。

1.  $\tau_d$  を求めるために利用したレオロジーデータを NAPLES を用いて計算してみる。
2. NAPLES の出力応力値に 3.8.2 節にある単位応力、出力時間にここでもとめた  $\tau_e$  をそれぞれ掛け合わせ、実験結果と比較する。両ログのグラフで平行移動することで比較が容易になる。
3. 実験をもっともよく再現するように単位時間と単位応力を調節し、最終的な値を求める。この値（特に  $\tau_e$ ）は基準温度が変わると変化することに注意する。

なお、単位時間と単位応力は、レオロジーデータを両対数グラフで時間や周波数に対してプロットした場合に平行移動させるためのシフトファクターであり、グラフの形状そのものを変化させることはない。グラフの形状を変えるには、4.5 で決めた  $Z$  の値、あるいは分子量分布や分子形状の入力を見直さなければならない。

### 3.8.2 NAPLES の単位応力（単位弾性率）

NAPLES における応力と弾性率の単位はを求めるには以下の方法で算出する。

まず、プラトー弾性率  $G_0$  が必須の物性量である。この値は多くの場合データベースや文献から調べることができる。もしない場合には、動的粘弾性測定やステップ変形による緩和弾性率測定などから求められる。プラトー弾性率を得ることができたとして、NAPLES の応力単位  $G_{NAP}$  は以下の式で決められる。

— 単位応力を求める式 —

$$G_{NAP} = G_0 \times 1.6 \quad (3.5)$$

このように求めた単位応力値を、さらに精度よく決めるためには、以下の手順を行う。

1. 3.8.1 節を参照して、実レオロジーデータとの比較を行う。
2.  $\tau_d$  を求めるために利用したレオロジーデータ-を NAPLES を用いて計算してみる。
3. NAPLES の出力応力値にここで求めた単位応力，出力時間にここ 3.8.1 節で求めた  $\tau_e$  をそれぞれ掛け合わせ，実験結果と比較する．両ログのグラフで平行移動することで比較が容易になる。
4. 実験をもっともよく再現するように単位時間と単位応力を調節し，最終的な値を求める．この値（特に  $\tau_e$ ）は基準温度が変わると変化することに注意する。

なお，単位時間と単位応力は，レオロジーデータを両対数グラフで時間や周波数に対してプロットした場合に平行移動させるためのシフトファクターであり，グラフの形状そのものを変化させることはない．実験と計算を比較してグラフの形状自体がおかしい場合には，4.5 節で決める  $Z$  の値，あるいは分子量分布や分子形状の入力を見直さなければならない。



## 第4章 NAPLES入力ファイル

ここでは、*NAPLES*オリジナルの *Input* ファイルの書式に従って入力パラメータについて説明するが、*OCTA*版で提供されている *UDF* 形式の *Input* ファイルを用いた場合にも、ほとんど同じ名称のパラメータを設定することとなる。*UDF* 形式の *Input* ファイルで用いられる各パラメータ名と、*NAPLES* オリジナルの *Input* ファイルで用いられるパラメータ名との関係は本章の最後にまとめてある。

*NAPLES* は計算するためのパラメーターを入力パラメーターファイルで受け取り、計算した結果を応力出力ファイル等へ書き出していく。本章では、入力ファイルの中身を詳しく説明する。

入力パラメーターファイルはテキストファイルであり、`input.npls` や `Sample` フォルダにあるファイルを Windows 付属のノートパッドなどの適当なテキストエディタで編集することができる。

### 4.1 基本ルール

- ファイル形式：  
入力パラメーターファイルのフォーマットはアスキーテキストファイルでなければならない。Macintosh, Windows, UNIX それぞれのシステムでテキストファイルの形式（改行コード）が異なるため、システム間でファイル転送する場合には注意が必要である。
- 入力文字：  
入力パラメーターファイル中には、半角英数字以外は記入してはならない。全角文字や半角カタカナがあると予期しない結果となる場合がある。
- コメントの記述  
行のはじめに`#`、または`%`があると、その行はコメントとして扱われる。
- パラメーターの記述  
入力パラメーターをファイル中に指定するには、入力するパラメーター名と数値を`=`でつなぐ。具体的には以下のようにする。

————— パラメーターの基本書式 —————

`[パラメータ名]=[入力値][改行]`

- パラメーターの順序：  
どのような順序で記述してもよい。ただし、同じ名称のパラメーターが複数同じファイル中に指定されている場合、もっとも下段にある値が採用される。

### 4.2 もっとも単純な入力パラメーターファイル

`naplesOCTA(.exe)` で計算を行うためのもっとも単純な入力パラメーターファイルは以下のようになる。

—— もっとも単純な入力パラメーターファイルの例 ——

```
# the simplest input file for NAPLES
# 09/10/2002 Y. Masubuchi
component_number=1
molecule_spec1=linear
Z1=10.0
flow_type=equilibrium
```

これは、直鎖単分散で、分子量  $Z$  が 10 の高分子の、平衡状態でのダイナミクスを計算するための入力パラメーターファイルである。入力パラメーターは大きく分けて、材料の特性を記述する部分、流動を記述する部分、その他計算の詳細を設定する部分に分けられる。通常のレオメーターでレオロジーを測定する場合、サンプルと流動条件は当然ユーザーが決定する。同じように NAPLES でも、材料の特性を記述する部分で分子の形状や混合比率を与え、流動を記述する部分でシステムに与える流れを指定する。上記のサンプルファイルでは、この2つをもっとも簡単に与えた例である。

それぞれのパラメーターの意味を説明する。まず `component_number` は計算するシステムに混入する成分の数を表している。単分散の系であれば 1 になる。`molecule_spec1` は、第一成分の分子の形状を示している。ここでは `linear` を指定し、線状高分子としている。 $Z1$  は、第一成分の分子量を、NAPLES の単位で示している。分子量の単位は、からみあい点間分子量であるが、NAPLES の場合通常のからみあい点間分子量よりも小さな値を理論上採用する必要がある。詳細は 6.5 節をごらんいただきたい。なお直鎖以外の高分子の場合は、より詳細な設定が必要になる。最後の `flow_type` は流れのプロファイルを示し、`equilibrium` は平衡状態（流れのない状態）を示している。

この他の計算の詳細に関するパラメーターは入力がない場合には初期値としてプログラムに埋め込まれた値を採用して計算するので、全てのパラメーターを導入する必要はない。ただし計算速度や計算精度など計算のチューニングが必要な場合には、計算の詳細を調節する必要がある。詳しくは 4.7 節を参照のこと。

## 4.3 試料の設定

### 4.3.1 化学種数の指定と関連パラメーターの設定

均一系である場合にはこの節の記述は不要であるが、異なる化学種を混合する場合にはまず化学種の数指定する。現状では 2 種類しか混ぜることはできない。

—— 化学種数を指定するパラメーター ——

```
chemical_component_number=[化学種数, 1 または 2]
```

2 種類の化学種間に働く相互作用は、Flory の  $\chi$  パラメーターで指定する。

—— 化学種間の相互作用パラメーター ——

```
chi12=[相互作用パラメーター]
```

Flory の  $\chi$  パラメーターは分子量には依存しないが、温度等多数の外的要因をとりこんでいる。詳しくは参考文献 [10] をご覧いただきたい。

異種高分子間では、からみあい点間分子量（平坦部弾性率）が異なるのが普通である。セグメント長さはからみあい点間分子量の  $1/2$  乗で与えられることを考慮して、第一成分を基準としたときの第二成分の比を以下で与える。デフォルトでは 1.0 となる。各成分の単位応力の比が分かっている場合、その逆比に従って設定すればよい。

—— 化学種間のからみあいセグメント長さの比 ——

```
bond_ratio_for_chemical[化学種番号]=[からみあいセグメントの長さの比]
```

また異種高分子間ではからみあいセグメントの摩擦係数（単位時間）も通常異なる．第一成分を基準としたときの第二成分の比を以下で与える．デフォルトでは 1.0 となる．各成分の単位時間が分かっている場合，その比に従って設定すればよい．

—— 化学種間のセグメントあたりの摩擦係数の比 ——

```
friction_ratio_for_chemical[化学種番号]=[摩擦係数の比]
```

### 4.3.2 鎖の伸びきり効果の指定

高分子の伸びきり効果を導入するには以下のキーを指定する．（デフォルトでは伸びきりのない Gauss バネが仮定されている．）

—— 伸びきり効果の導入 ——

```
FENE_flag=true
```

伸びきりを指定する場合は最大伸長比も合わせて指定しなければならない．

—— 化学種ごとの最大伸長比の指定 ——

```
maximum_stretch_for_chemical[化学種番号]=[最大伸長比]
```

ブレンドでない場合には，上記の化学種番号は 1 とする．最大伸長比はデフォルトでは 4 に設定されており，これはポリスチレンメルトにおける値が想定されている．溶液の場合は高分子の濃度に応じて適切な値に設定される必要がある．

### 4.3.3 摩擦低減効果の指定

2012 年に提案された，伸長状態における分子摩擦の低下の効果を導入するには以下のキーを指定する．（デフォルトでは摩擦は一定である．）

—— 摩擦低減効果の導入 ——

```
friction_reduction_flag=true
```

摩擦の低減の詳細は原論文 [6] を参照いただきたい．

### 4.3.4 構成成分数の指定

種々の形状，分子量をもつ高分子を混ぜることができる．成分数を指定するには，以下を入力パラメーターファイルに書き込む．

—— 構成成分を指定するパラメーター ——

```
component_number=[構成成分数]
```

デフォルトでは 1（単分散）に設定されている．従って単分散試料での計算を行う場合にはこのパラメーターを記入する必要はない．分子量分布を持たせたい場合には，長さごとに成分比率を指定しなければならない．成分比率は以下のように指定する．

—— 構成成分の比率を指定 ——

```
fraction[成分番号]=[体積分率]
```

化学種が複数ある系では、各成分ごとに化学種を指定する必要がある。

—— 構成成分の化学種を指定 ——

```
chemical_spec_for_component[成分番号]=[化学種番号]
```

また、各成分ごとに分子の形態と分子量を設定しなければならない。具体的には以下のように記述する。

—— 個々の成分の分子形態記述 ——

```
molecular_spec[成分番号]=[分子形態]  
Z[成分番号]=[分子量]  
....
```

以下の例では分子量が  $Z = 3$  の直鎖高分子が 35 %，分子量が  $Z = 20$  の直鎖高分子が 25 %，腕の長さが  $Z = 5$  で，腕の本数が 4 本の星形高分子を 15 %，腕の長さが  $Z = 7$  で腕の本数が 3 本，かつブリッジ部分の分子量が 15 の pompom 高分子が 15 %，腕の長さが  $Z = 2$ ，腕の本数が 5，ブリッジ分子量が 10，1 分子あたりの分岐点の数が 3 の櫛型高分子が 10 %，それぞれ混合されている系を記述する。（サンプルファイル `mixture_uniaxal.npls` の一部）各成分には適宜化学種を割り振る。分子の形態を記述する方法についての詳細は 4.3.5 節を参照いただきたい。

—— レオロジーを調べたい材料に関する記述の例 ——

```
chemical_component_number=2
chi12=2.0
component_number=5
#
chemical_spec_for_component1=1
fraction1=0.35
molecule_spec1=linear
Z1=3.0
#
chemical_spec_for_component2=1
fraction2=0.25
molecule_spec2=linear
Z2=0.25
#
chemical_spec_for_component3=2
fraction3=0.15
molecule_spec3=star
armZ3=5.0
arm_per_junction3=4
#
chemical_spec_for_component4=1
fraction4=0.15
molecule_spec4=pompom
armZ4=7.0
arm_per_junction4=3
bridgeZ4=15
#
chemical_spec_for_component5=2
fraction5=0.10
molecule_spec5=comb
armZ5=2
arm_per_junction5=3
bridgeZ5=10
junction_number5=3
#
```

#### 4.3.5 分子形状の指定：概要

NAPLES ではサンプル内の分子形状と分子量を与えるために、表 4.1 のパラメータ群を利用する。なお各パラメーターの入力例で、パラメーター名の後に続く数値 1 は成分番号をあらわしている。複数成分を混合する場合の詳細は 4.3.4 節を参照のこと。以下で、具体的な分子形状ごとに設定の詳細を説明する。

表 4.1: 分子形状を規定するパラメータ

パラメータ名	意味	例
molecule_spec	分子の形状	molecule_spec1=linear
Z	直鎖分子の分子量	Z1=10.0
armZ	分岐点から伸びる腕の分子量	armZ1=5.0
arm_per_junction	分岐点あたりの腕の本数	arm_per_junction1=5
bridgeZ	pompom および櫛型分子での分岐点間分子量	bridgeZ1=6.0
junction_number	櫛型分子での 1 分子あたり分岐点数	junction_number1=10

#### 4.3.6 直鎖高分子

図 4.1 で示される, 直線状で分岐のない形態の高分子.



図 4.1: 直鎖高分子

この形の分子は, 分子量 (パラメーター名 Z) だけを指定すればよい.

—— 直鎖高分子のパラメータ記述例 ——

```
molecule_spec1=linear
Z1=3.0
```

Z は実分子量を NAPLES における単位分子量で無次元化した値である. 詳細は 6.5 節を参照のこと. また, 分子量分布を与えたい場合には, 分子量の異なる複数の直鎖高分子成分を指定する.

—— 分子量分布のある直鎖高分子系の場合 ——

```
component_number=10
molecule_spec1=linear
Z1=3.0
fraction1=0.05
molecule_spec2=linear
Z2=5.0
fraction2=0.1
molecule_spec3=linear
Z3=10.0
fraction3=0.15
molecule_spec4=linear

Z4=20.0
fraction4=0.2
.....
```

簡単な例として、二様分布の場合を `linear.bdp.npls` としてサンプルフォルダに納めてある。

### 4.3.7 星形高分子

図 4.2 のように、分岐点を 1 点だけもち、そこから放射状に腕が伸びている高分子。

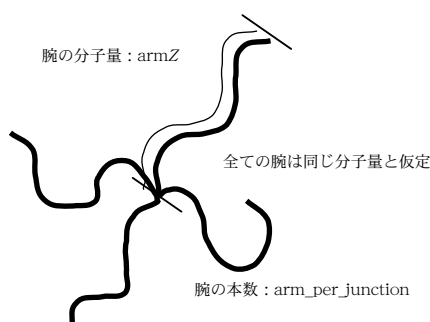


図 4.2: 星形高分子

この形の分子の形状を指定するには、腕の分子量（パラメーター名 `armZ`）に加えて、分岐点から伸びている腕の数（パラメーター名 `arm_per_junction`）の指定が必要である。そこで以下のように記述する。（サンプルファイル `star.npls` の一部）

星形高分子のパラメータ記述例

```
molecule_spec1=star
armZ1=5.0
arm_per_junction1=4
```

この例では、分岐点から伸びている腕の数が 4、腕の長さが  $Z=5.0$  で与えられる星形高分子となる。パラメーター `armZ` は分岐点からのびている腕の分子量を NAPLES における分子量単位  $Z$  で示している。（分子量単位の詳細は 6.5 節を参照）パラメーター `arm_per_junction` は、分岐点から伸びている腕の数を示している。

### 4.3.8 非対称星形高分子（共重合高分子）

分岐点から伸びている腕の長さや化学種の異なる星形高分子。二種類までの異なる腕を混在させることができる。

非対称星形高分子のパラメータ記述例

```
molecule_spec1=astar
arm1Z1=5.0
arm2Z1=10.0
arm1_per_junction1=4
arm2_per_junction1=2
chemical_spec_for_arm1_of_component1=1
chemical_spec_for_arm2_of_component1=2
```

この例では図 4.3 にあるような分子を記述している。なお、異なる化学種間の相互作用等の記述については 4.3.4

を参照のこと.

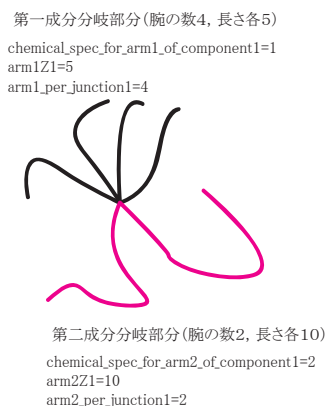


図 4.3: 非対称星形高分子

#### 4.3.9 ポンポン形高分子

図 4.4 のように, 2 つの星形高分子を, 1 本の線状高分子で接続した形をもつ.

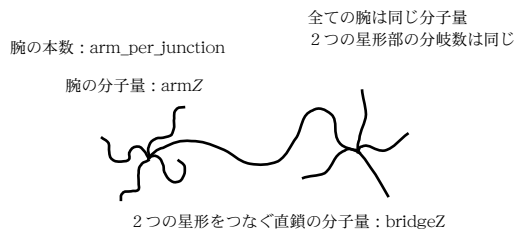


図 4.4: ポンポン形高分子

1998 年に McLeich と Larson [7] により, 分岐高分子のモデルとして星形高分子の次に複雑なクラスであると規定された. この高分子の形状を記述するには, 星形高分子の記述に加えて, 2 つの星形高分子を接続する線状高分子の長さ (パラメーター名 `bridgeZ`) が必要になる. 記述例は以下である. (サンプルファイル `pompom.npls` の一部)

—— ポンポン形高分子のパラメータ記述例 ——

```
molecule_spec1=pompom
armZ1=7.0
arm_per_junction1=3
bridgeZ1=15
```

この例では分岐点から伸びている腕の数が 3, 腕の長さが  $Z=7.0$  で与えられる星形高分子 2 つを, 分子量  $Z=15$  の直鎖高分子で接続した `pompom` 高分子を示している. 星形高分子を接続する直鎖高分子は, `arm_per_junction` のカウントに入っていないので注意されたい.



### 4.3.10 ポンポン共重合高分子

ポンポン高分子の拡張として以下のように記述すると、ポンポン部分とブリッジ部分の化学種を変えた共重合高分子となる。

—— ポンポン形共重合高分子のパラメータ記述例 ——

```
chemical_spec_for_component1=1
molecule_spec1=pompom
armZ1=7.0
arm_per_junction1=3
bridgeZ1=15
chemical_spec_for_arm_of_component1=2
```

chemical\_spec\_for\_component で指定される化学種がブリッジ部分の化学種となる。

### 4.3.11 櫛形高分子

図 4.5 のように、複数の星形高分子を、線状高分子で接続した形をもつ。

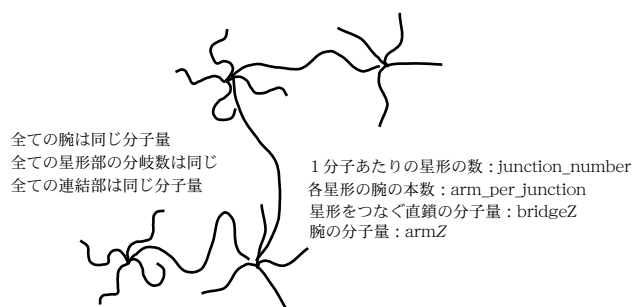


図 4.5: 櫛形高分子

ポンポン高分子の拡張. この高分子の形状を規定するには、ポンポンの場合に加えて、1分子あたりの分岐点の数（連結する星形高分子の数：パラメーター名 `junction_number`）が必要になる。記述例は以下である。（サンプルファイル `comb.npls` の一部）

—— 櫛形高分子のパラメータ記述例 ——

```
molecule_spec1=comb
armZ1=2
arm_per_junction1=3
bridgeZ1=10
junction_number1=3
```

分岐点から伸びている腕の数が3、腕の長さが  $Z=2.0$  で与えられる星形高分子3つを、分子量  $Z=10$  の直鎖高分子で接続した櫛型高分子を示している。また、分岐の数は、分子内で接続されている全ての星形高分子で同じになっている。

### 4.3.12 櫛形共重合高分子

櫛形高分子の拡張として、ポンポン高分子と同様に、分岐部分の化学種を変えることで、共重合体とすることができる。

—— 櫛形共重合高分子のパラメータ記述例 ——

```
molecule_spec1=comb
chemical_spec_for_component1=1
armZ1=5.0
arm_per_junction1=3
bridgeZ1=10
junction_number1=3
chemical_spec_for_arm_of_component1=2
```

### 4.3.13 直鎖共重合体

ジブロック共重合体、トリブロック共重合体を計算するには、直鎖高分子の記述を拡張する。まずジブロック共重合体では以下のようにする。

—— ジブロック共重合高分子のパラメータ記述例 ——

```
molecule_spec1=diblock
Z1=20
block_ratio1_1=0.5
chemical_id_for_block1_1=1
chemical_id_for_block2_1=2
```

まず  $Z1=20$  で全体の分子量を決める。次に  $block\_ratio1\_1=0.5$  で、ブロック比を与える。ブロック比は0以上1以下であり、分子量は全体の分子量にブロック比を掛けたものになる。また、続いて、 $chemical\_id\_for\_block1\_1$  により、それぞれのブロックに対して化学組成を記述する。

次の例でトリブロック共重合体の例を示す。

—— トリブロック共重合高分子のパラメータ記述例 ——

```
molecule_spec1=triblock
Z1=20
block_ratio1_1=0.3
block_ratio2_1=0.3
chemical_id_for_block1_1=1
chemical_id_for_block2_1=2
chemical_id_for_block3_1=1
```

ほぼジブロック共重合体に準じた書式になっている。上記の例で記述される高分子は図4.6に示したものになる。

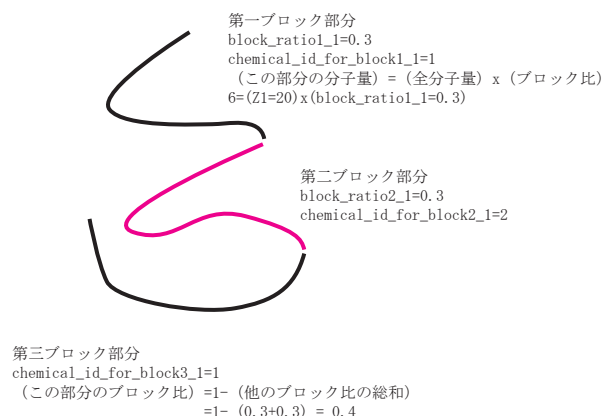


図 4.6: トリブロック共重合高分子

なお分岐共重合高分子の与え方は、星形、ポンポン形、櫛型分岐高分子の記述を参照のこと。

## 4.4 ゴムやゲル等の架橋材料

ゴムやゲル等の架橋材料のシミュレーションを行う場合、いきなり系をつくることができない（初期構造の妥当性を保証できない）ため、まず高分子メルトを想定した計算を行って系の平衡化を行う。その後架橋させて架橋材料系とする。

### 架橋材料の作り方

1. 架橋前の高分子系を 4.3 節に従って作成する。
2. 平衡状態計算を行って系を平衡化する。4.6.3 節。
3. 系に架橋点を導入する。（本節）
4. 架橋された系に変形を与えてレオロジー測定を行う。4.6 節。

高分子系の計算と異なる点は、平衡化後に架橋点の導入を行う点である。

### 4.4.1 架橋点導入用入力ファイル

架橋点を導入するには、架橋させる前の試料の情報を書き、平衡化条件を記述し、さらに架橋させる条件について記述する。以下に入力ファイルの例を示す。

#### 架橋点導入パラメータ記述例

```
molecule_spec1=linear
Z1=20
flow_mode=gelation
pre_reptation_step=100
quit_dynamics_count=100
crosslink_mode=random
crosslink_fraction=0.1
```

系に架橋点を導入するには、`flow_mode` に `gelation` を指定する。まず `pre_reptation_step` で指定される計算ス

トップ（ここでは 100 ステップ）で系を平衡化し、その架橋する。この例では、からみあい点のうち 10 % をランダムに架橋点に変更する。上記の例では、架橋後、さらに 100 ステップの計算を行っている。

#### 4.4.2 架橋点導入の応用

2 種類の異なる化学種の高分子をブレンドしたとき、その片方だけを架橋させることも可能である。その場合には以下のように記述する。

ブレンドでの架橋点導入パラメータ記述例

```
chemical_component_number=2
chi12=2.0
component_number=2
#
molecule_spec1=linear
Z1=20
fraction1=0.5
#
molecule_spec2=linear
Z2=5
fraction2=0.5
#
flow_mode=gelation
pre_reptation_step=100
quit_dynamics_count=100
crosslink_chem=1
crosslink_mode=random
crosslink_fraction=0.1
```

上記の例では、第一成分（つまり  $Z = 20$  の成分）のみで形成されるからみあいだけが架橋される。crosslink\_fraction の値は、指定された化学種どうしのからみあいのなかから、どれだけを架橋させるか、を指定することになる。

なお、化学種についての架橋指定をブロック共重合体で行えば、分子の特定の場所で架橋させることができる。たとえば分子の末端に近い部分だけで架橋させたいときは、まず直鎖のトリブロック共重合体を指定し、メルトの状態では平衡化し、適当に相分離をさせてから架橋させる。

— 共重合体での架橋点導入パラメータ記述例 —

```
chemical_component_number=2
chi12=2.0
component_number=1
#
molecule_spec1=triblock
Z1=20
block_ratio1_1=0.1
chemical_id_for_block1_1=1
block_ratio2_1=0.8
chemical_id_for_block2_1=2
block_ratio3_1=0.1
chemical_id_for_block3_1=1
#
flow_mode=gelation
pre_reptation_step=100
quit_dynamics_count=100
crosslink_chem=1
crosslink_mode=random
crosslink_fraction=0.5
```

この例では、全長が  $Z = 20$  の直鎖高分子の、両端 10 % ずつが架橋点を形成する。

#### 4.4.3 架橋点導入後のレオロジー計算

架橋化した系に変形を与えてレオロジー測定を行いたい場合には、架橋計算の後で得られる finalconf.npls ファイル（系の全状態を保存しているファイル）を用いて計算を再実行する必要がある。

具体例を示して説明する。ここでは架橋系に、ステップせん断変形を与える。まず flow\_mode=gelation として、架橋系を作成する。（4.4.1 節参照。）次に計算に用いた入力パラメーターファイルを、以下のように書き換える。

— 架橋点導入系に変形を与える入力ファイルの記述例 —

```
molecule_spec1=linear
Z1=20
flow_mode=step_shear
calculation_mode=continue
initial_strain=1.0
pre_reptation_step=5
quit_dynamics_count=1000
```

ここでのポイントは、calculation\_mode が continue となっていることである。これにより、初期構造として finalconf.npls に書かれている構造が読み込まれる。また、このとき、molecule\_spec と Z の値については、架橋導入操作に使用した高分子系と同じになっていなければならない。架橋導入に関する記述は、あってもなくても関係ない。

動的粘弾性試験を行い、 $G'G''$  を求める場合は、さらに簡単である。架橋点導入に用いた入力ファイル名を test.npls、計算の結果できた系の状態を保存しているファイルを finalconforg.npls とする。naplesgwOCTA(.exe) を利用できるならば入力ファイルの書き換えは不要で、コマンドラインで以下のように入力すればよい。

— naplesgw.out による架橋系の G'G'' 計算 —

```
> naplesgwOCTA(.exe) -config finalconforg.npls -input test.npls
```

## 4.5 分子量 Z 値の決め方

NAPLES では分子量の単位として、からみあい点間分子量  $M_e$  で分子量  $M$  を割った値  $Z$  を採用している。ただし、一般に実験的に求められる  $M_e$ 、およびレプテーション理論や PASTA で採用されている  $M_e$  をそのまま利用することはできない。以下に述べる補正が必要である。まず実験的には  $M_e$  は以下の式で求められる [9]。

$$M_e = \frac{\rho RT}{G_0} \quad (4.1)$$

ここで  $\rho$  は密度（溶液系の場合には高分子の密度）、 $R$  は気体定数、 $T$  は温度、 $G_0$  はプラトー弾性率である。この値が多くの場合文献やデータベースに登録されている。もし溶液などで  $M_e$  の値がわからない場合には、プラトー弾性率  $G_0$ （と高分子濃度  $\rho$ ）の計測だけは行わないと NAPLES での計算に乗せることはできない。NAPLES で必要な分子量  $Z$  は、対象とする分子の分子量  $M$  と 4.1 式の  $M_e$  を用いて、

— Z 算出のための式 —

$$Z = \frac{M}{M_e} \times 1.6 \quad (4.2)$$

で求めることができる [2]。<sup>1</sup>

## 4.6 変形および流れ場の設定

### 4.6.1 基本事項

流れ場設定は、パラメーター名 `flow_type` で流れ場の基本的な形状を規定し、あわせてそれぞれの流れ場に必要のパラメーターを随時規定する。流れ場は、せん断、1 軸伸長、2 軸伸長、平面伸長のそれぞれの変形態で、ステップ変形とスタートアップ変形（定常変形）、せん断の場合はこれに加えて動的モードを選択できる。以下、それぞれの場合のパラメーター記述方法と記述例を示す。なお、`naplesOCTA(.exe)` では上記の基本変形様式を混合して加えることはできない。

### 4.6.2 時間スケールおよび流動速度について

NAPLES では、6.5 節に示すように  $\tau_e$  を計算の基本的時間スケールとしている。つまり、流動速度も  $1/\tau_e$  の流れを基本としている。従って流動条件を現実に合わせてきめるには  $\tau_e$  の値が必要である。 $\tau_e$  の現実の値を決める方法については、3.8.1 節を参照のこと。

### 4.6.3 系の平衡化（熱処理）

レオロジーの実験をされた経験をお持ちの方ならおわかりのように、サンプルをバージンの状態でいきなり測定を開始しても意味のあるデータをとることができない。この理由は、サンプル内部の状態が十分に平衡化されていないためである。平衡化された状態から開始しなければ、データはサンプルを準備した時の条件に依存して変化してしまうため、再現性も悪い。実験ではこれを避けるために測定に先立って、サンプルをガラス転移温度より十分高い温度でしばらく放置する。これは平衡化あるいは熱処理と呼ばれる。

<sup>1</sup>NAPLES が計算の妥当性を保証する  $Z$  の値の範囲は、2 以上約 100 以下程度である。

NAPLES も同様である。NAPLES における高分子のからみあい構造は、計算開始直後は平衡状態でなく、局所的なストレスがかかった状態にある。これを解消してから計算を開始しないと、データの信頼性が低くなる。そこで、系に流動を加える場合には必ず以下のパラメーターを付け加えることを推奨する。

—— 平衡化時間の設定 ——

```
pre_reptation_step=100
```

ここで与える時間（単位時間は  $\tau_e$ 、詳しくは 6.5 節参照）の間、システムには流れや変形がかからない。変形させずに待つ時間の推奨値は、そのシステムの最長緩和時間（3.8.1 節を参照）である。最長緩和時間  $\tau_c$  は、直鎖分子あるいは星形分子の場合計算する分子の分子量を  $Z$  であらわしたとき  $\tau_c = 0.0025 Z^{3.6} \tau_e$  であらわされる。  $\tau_c$  が計算時間に対して非現実的な場合（10,000 を超える場合）には、最低でも 100、分子量が大きな場合や分岐構造が複雑な場合には 1000 を推奨したい。

なお、以下の場合には平衡化せずにいきなり計算を開始してもよい。その場合には `pre_reptation_step=0` と指定する。

—— 平衡化の必要がない場合 ——

- 流動のない平衡状態（`flow_type=equilibrium`）で計算する場合。このときは後のデータ処理段階で、データを見ながら計算開始直後の意味の無いデータをカットする。
- 定常流動下で定常粘度だけがとりたい場合。粘度が定常に達するのを待てば、系は定常状態にあると判断されるので、定常に達するまでの値を捨てるのであれば平衡化処理は不要。ただし、応力の時間変化を問題にする場合は平衡化してから流れをスタートさせる必要がある。
- 計算を再スタートさせる場合（2.2.3 参照）。再起動するための `calculation_mode=restart` というパラメーターがファイル中にある場合には、`pre_reptation_step` パラメーターは無効化される。

#### 4.6.4 ステップ変形

せん断、一軸、二軸、平面変形をある瞬間にステップ状に与える変形モード。これを行うには、以下のよう

に記述する。

—— ステップ変形を与えるパラメータ記述例 ——

```
flow_type=step_shear
pre_reptation_step=100
initial_strain=0.5
quit_dynamics_count=10000
```

ここで `flow_type` パラメーターが流れ場を設定する。以下に指定できる流動形態の一覧を示す。 `pre_reptation_step`

表 4.2: ステップ変形の流動形態指定

流れ形態	入力する値
せん断	<code>step_shear</code>
一軸	<code>uniaxal_step</code>
二軸	<code>biaxal_step</code>
平面	<code>planar_step</code>

が平衡化のための時間ステップを表し、`initial_strain` が平衡化後に与えるせん断ひずみ量を Hency（ヘンキー）

ひずみであらわしている．quit\_dynamics\_count は計算を終了させるまでの時間を表している．つまりこの例では 100 ステップの平衡化後（流れや変形のない計算を行った後），Hency ひずみで 0.5 のせん断ひずみを与え，その後 10000 ステップの間，再び変形のない状態での計算を行って応力の緩和をみる．せん断ひずみで 0.5 のひずみ量というのはレオメーターなどでよく用いられる%の表示では，この例だと 50%である．また 1 軸伸長ひずみの場合には，Hency ひずみ  $\epsilon$  は，伸長前の系の長さを  $l_0$ ，伸長後の長さを  $l$  とした場合に，

$$\epsilon = \exp \frac{l}{l_0} \quad (4.3)$$

で与えられる．ちなみに引っ張り試験などで用いられる伸長度は通常  $l/l_0$  で与えられるので注意が必要である．

計算を終了させるまでの時間 quit\_dynamics\_count は，3.8.1 節にある最長緩和時間の 2 倍程度とすることがのぞましい．最長緩和時間は，計算する分子の分子量を  $Z$  であらわしたとき  $\tau_c = 0.0025Z^{3.6}$  である．ただしポンポン形や櫛型の場合はもっと長い．いったん終わった計算を再開することもできるので，最長緩和時間が不明な場合には数千から 1 万程度の値を入力しておく．

なお，実験と同様にせん断ひずみ量が小さすぎる場合には応力が検出されない．ひずみ量がおよそ 0.25 以下の場合には数値計算の誤差が大きくなるために意味のある結果を得ることが難しい．線形緩和弾性率の時間依存カーブを精度よく取得したい場合には，平衡状態での計算を長時間行い，応力の自己相関関数を計算する．この方法の詳細は 3.2 節をご覧ください．

#### 4.6.5 スタートアップ流れ（定常流）

せん断，一軸，二軸，平面変形をある瞬間から一定のひずみ速度で与える流動変形モード．これを行うには，以下のように記述する．

—— スタートアップ流れのパラメータ記述例 ——

```
flow_type=start_up_shear
pre_reptation_step=100
strain_rate=0.05
quit_dynamics_count=10000
```

ここで flow\_type パラメーターが流れ場を設定する．流れ場の一覧を以下に示す．pre\_reptation\_step が平衡化

表 4.3: スタートアップ流れの流動形態指定

流れ形態	入力する値
せん断	start_up_shear
一軸	uniaxal
二軸	biaxal
平面	planar

のための時間ステップを表し，quit\_dynamics\_count は計算を終了させるまでの時間を表している．つまりこの例では 100 ステップの平衡化後（流れや変形のない計算を行った後），Hency ひずみでひずみ速度 0.05 の流れを，その後 10000 ステップにわたって与え続ける．ここでひずみ速度は，NAPLES の時間単位  $\tau_e$  で無次元化されていることに注意が必要である．実際のひずみ速度は， $1/\tau_e$  を掛けた値になる． $\tau_e$  の求め方については，3.8.1 節を参照のこと．計算を終了させるまでの時間 quit\_dynamics\_count も同様に  $\tau_e$  の単位になっている．

なお，スタートアップ流れの計算を行うに当たっては，伸長変形ひずみをできるだけ稼ぐために，あらかじめ負のひずみがかかった，立方体でないセルから計算をスタートさせる．そのため，pre\_reptation\_step による平衡化が十分でないと，負の伸長応力が検出されることがある．十分な平衡化時間をとっていただきたい．



また、`flow_type=equilibrium` での計算で得られた `finalconf.npls` ファイルでは、伸長流動の計算をリスタートさせることはできない。せん断のときはこの限りでない。

#### 4.6.6 動的せん断

せん断ひずみが式 4.4 で与えられる時間変化を伴うモード。線形粘弾性プロファイル  $G^*(\omega)$  を求めるために用いられる。

$$\gamma = \gamma_0 \sin(\omega t) \quad (4.4)$$

動的せん断のパラメータ記述例

```
flow_type=dynamic_shear
pre_reptation_step=100
strain_amplitude=0.25
strain_omega=0.1
dynamic_strain_cycle=5.0
```

`pre_reptation_step` が平衡化のための時間ステップを表している。`strain_amplitude` は式 4.4 の  $\gamma_0$ 、`strain_omega` は  $\omega$  をそれぞれ表す。`dynamic_strain_cycle` は、式 4.4 のような振動ひずみを何周期分与えるかを示している。デフォルトは 5 であるが、長時間を計算する場合には長すぎる場合があるので適宜増減する。ここでのひずみ周波数は、NAPLES の時間単位  $\tau_e$  で無次元化されていることに注意が必要である。実際の周波数は、 $\tau_e$  を掛けた値になる。 $\tau_e$  の求め方については、3.8.1 節を参照のこと。

なお、このファイルによる計算だけでは、ある一つの周波数における応力の時間変化が得られるだけである。このような応力ファイルから、その周波数に対する  $G'(\omega)$  および  $G''(\omega)$  の値を求めるには、解析ツールである `getgval.exe` を用いる。さらに連続的に周波数を変化させ、線形粘弾性プロファイル  $G^*(\omega)$  を周波数の関数として得るためには自動処理などを用いる必要がある。詳しくは 3.3 節を参照のこと。

## 4.7 計算の詳細条件の設定

NAPLES では、ユーザーは試料に関する記述と、流動や変形に関する記述を行えば他の計算の詳細にかかわるパラメーター設定は不要である。ただし、チューニングを行うことで、意味のある結果を、より早く、より精度よく得ることは可能であるので、計算時間や精度に疑問のある場合には本節をお読みいただきたい。

NAPLES に限らず、分子運動を計算してそこからマクロな物性値を予測する分子シミュレーションとよばれる手法では、構成方程式を用いた計算とは異なり、多数の分子運動の結果を統計処理して応力を算出する。このため計算の負荷を高くするほど精度が向上し、計算を軽くするほど精度は落ちる。本節では、そのような目的で計算条件の設定を行う場合について説明する。

なお、精度を向上させる方法や計算の負荷を落とす方法は目的別に 7.2 節および 7.1 節にも簡単に説明してあるので参照いただきたい。

### 4.7.1 乱数の種の設定

NAPLES では統計物理学的計算を行うため乱数が重要な意味を持っているが、コンピュータでは疑似乱数列と呼ばれる統計的に乱数と見なすことができる数列を用いる。この乱数列は、初期に与える数値（種という）を変えることで異なる数列となる性質をもっている。NAPLES ではこの性質を利用して、同じ実験条件でもサンプルを変えて統計量をとることができるようになっている。乱数の種は、以下のパラメーターで与える。

乱数の種を設定するパラメータ

`seed=[任意の整数値]`

たとえば、以下のような入力パラメーターファイルを作成し、直鎖単分散の系にステップせん断を与える応力緩和のシミュレーションを行うとする。

```
seed=845
flow_type=step_shear
pre_reptation_step=100
initial_strain=0.5
quit_dynamics_count=1000000
component_number=1
molecule_spec1=linear
Z1=20
fraction1=1.0
```

この入力パラメーターファイルを用いて計算する限りは、常に同じ値が結果ファイルである `measure.npls` に書き込まれる。しかし、一行目の `seed` 値を変更すると、実験を繰り返したときと同様に、ほとんど同じながらも少しだけ異なる結果を返してくる。これは、系に含まれる分子の初期状態が、`seed` 値を変えることで変わるからである。実験でいうなら、サンプルを変えて同じ実験を行う場合に相当する。このように `seed` 値だけを変更した計算を繰り返し、結果の平均をとることで計算の精度を向上させることができる。なお、`seed` 値は指定のない限り NAPLES 内部であらかじめ設定された値が計算に用いられるため、`seed` 値の指定をせずに計算をすると他の計算条件が同じである限り、常に全く同じ結果をかせしてくる。

#### 4.7.2 計算セルサイズの設定

NAPLES は分子シミュレーションに一般によく用いられる周期境界条件を用いている。ユニットセルの一边あたりのサイズは、以下のパラメーターで与えることができる。

計算セルのサイズを設定するパラメータ

`cell_size=[ユニットセルの1辺の長さ. 指定のない場合は8]`

ここでセルサイズの単位は、からみあいセグメントの平均長さ  $a$  である。(詳しくは 6.5 節を参照) 周期境界条件の単位セルのサイズは一般にできるだけ大きくとった方がよいが、大きくとることでシステムサイズが大きくなり、計算しなければならない分子の数が増えてしまう。これにより計算負荷が高まり、計算が重くなる。

一般には、分子のサイズに比べて十分大きくする必要がある。なお、セルサイズの最大推奨値は 16、最小値は 4 である。ただし、伸長流動下で計算をする場合には、最大ひずみがセルサイズによって違う。最大ひずみを大きく取りたい場合には、できるだけ大きなセルサイズを採用する必要がある。

#### 4.7.3 セグメント数密度

NAPLES は多数の分子を同じ計算セル中に導入して計算を行っている。このためセル中にどれくらいの濃度で分子を存在させるかを決めなければならない。実験では高分子の濃度によってからみあい点の数が増え、これによりからみあい点間分子量も変化するため、濃度はきわめて重要なパラメーターであるが、NAPLES の場合には濃度とからみあいセグメントの平均長さ  $a$  は独立なパラメーターとして扱われているため、ユーザーは濃度を任意に設定できる。また出力される結果も極端な場合をのぞけばセグメント数密度にはほとんどよらない。そこでどうしても高速に計算をおわらせたい場合など、結果は保証しないが、このパラメーターを変更することができる。

—— セグメント数密度を設定するパラメータ ——

`C`=[単位体積あたりのからみあいセグメントの数密度. 指定のない場合は 10(推奨値)]

`C` の値を倍の 20 とすると、計算は倍の時間を要するが精度もそれだけよくなる。逆に半分の 5 とすると、精度は犠牲となるが計算は半分の時間で終了する。ただし、分岐高分子の場合の、1 分岐点あたりの分岐数 (`arm_per_junction`) の値よりも `C` の値を小さくすると、セグメントの空間濃度揺らぎが大きくなりすぎて計算がうまくいかなくなるので注意が必要である。

#### 4.7.4 計算時間刻みの設定

まずはじめに、このオプションは設定変更しないことを推奨する。NAPLES では本来連続の時間発展方程式を差分化して数値的に解いているため、計算のための時間刻みが設定されている。一般に数値積分では、時間刻みが小さいほど真の解に近い答えを得ることができるが、時間刻みを小さくするとそれだけ計算時間がかかるため、時間刻みを変えながら計算を行って、結果に影響がでない最大の時間刻み幅を採用する。NAPLES ではこの最適化された時間刻みがあらかじめ入っているが、どうしても高速に計算をおわらせたい場合など、結果は保証しないが、このパラメーターを変更することができる。

—— 数値積分の時間刻みを設定するパラメータ ——

`p2`=[数値積分の時間刻み幅. 指定のない場合は 0.01(推奨最適値)]

`p2` の値を倍の 0.02 にすると、計算は半分の時間で終了できる。ただし結果は保証されない。

#### 4.7.5 出力ファイルの設定

NAPLES では `refoutputfile` に示すように種々のファイルを計算の結果として出力する。ファイルの出力は好みに応じて制御できる。入力パラメーターとして与えることができる制御キーは以下である。

—— 出力ファイルを制御するパラメータ ——

`measure_flag`=[応力ファイル `measure.npls` を書き出すインターバル. デフォルトは 1 (推奨値) ]  
`chain_av_flag`=[分子鎖の形態に関する統計量 `stat.npls` を書き出すインターバル. デフォルトは 0 (書き出さない) ]  
`snap_shot_save_flag`=[スナップショット作成のための全形態 `config[xxxx].npls` を書き出すインターバル. デフォルトは 0 (書き出さない) ]  
`free_end_stat_flag`=[架橋されていない末端に関する統計分布情報 `tailZdist[xxx].npls` および `tailNdist[xxx].npls` を書き出すインターバル. ]

これらのファイルの詳細は、それぞれ 5.3 節 (`measure.npls`)、5.4 節 (`stat.npls`)、5.6 節 (`config.npls`)、5.11 節 (`tailZdist.npls` および `tailNdist.npls`) をご覧いただきたい。

## 4.8 パラメーターファイルリファレンスマニュアル

表 4.4: パラメータリファレンス

パラメーター	意味	例
calculation_mode	計算起動時の設定. new を与えると新規に開始. restart とすると finalconf.npls ファイルの状態から再開. deform とすると finalconf.npl の状態から再開し, かつパラメーターファイルに設定された変形を与える. デフォルトは new.	calculation_mode=new
cell_size	周期境界条件の単位セルのサイズ. デフォルト値は 8. (単位は $a$ )	cell_size=8
C	単位体積 ( $a^3$ ) に含まれるからみあいセグメントの数. デフォルト 10. この値は計算の結果に本質的には影響しない.	C=10
p2	NAPLES の計算式を数値積分する際の刻み幅. デフォルト値は 0.01 (単位は $\tau_e$ ). この値を変更すると結果が保証されない.	p2=0.01
seed	NAPLES の計算式における乱数の種. 同じ計算条件で異なるサンプルを生成するために利用.	seed=845
quit_dynamics_count	計算を終了する時間ステップ. (単位は $\tau_e$ )	quit_dynamics_count=10000
flow_type	流れ場を規定するパラメータ. equilibrium (平衡, 流れなし), step_shear (ステップせん断), uniaxal_step (ステップ一軸新伸長), biaxal_step (ステップ二軸伸長), planar_step (ステップ平面伸長), start_up_shear (スタートアップせん断), uniaxal (一軸伸長), biaxal (二軸伸長), planar (平面伸長), gelation(架橋作成), dynamic_shear (動的せん断) が選択できる. デフォルトは equilibrium	flow_type=uniaxal
pre_reptation_step	流動を与える前に必要な, 流れのない平衡化時間. デフォルトは 100.	pre_reptation_step=100
initial_strain	ステップ変形を与える際のひずみ量. デフォルトは 0.	initial_strain=0.5
strain_rate	スタートアップ流れの際のひずみ速度. 単位は $1/\tau_e$ . デフォルトは 0.	strain_rate=0.01
strain_amplitude	動的せん断モードの際のひずみ振幅. デフォルトは 0.25	sstrain_amplitude=0.25
strain_omega	動的せん断モードの際の周波数. 単位は $1/\tau_e$ . デフォルトは 0.	strain_omega=0.01
dynamic_strain_cycle	動的せん断モードの際のひずみサイクル数を与える. (例えば 2 ならば 2 周期分の正弦波形ひずみをあたえて終了する.) デフォルトは 5.	dynamic_strain_cycle=2.0

パラメーター	意味	例
chemical_component_number	サンプルに含まれる化学種の数. デフォルトは 1	chemical_component_number=2
chi12	化学種間の相互作用パラメーター	chi12=2.0
bond_ratio_for_chemical [化学種]	[化学種] のからみあいセグメント長さ. デフォルトは 1.0	bond_ratio_for_chemical2=1.5
friction_ratio_for_chemical [化学種]	[化学種] のからみあいセグメントあたりの摩擦係数. デフォルトは 1.0	friction_ratio_for_chemical2=1.5
component_number	サンプルに含まれる構成成分の数. デフォルトは 1	component_number=2
chemical_spec_for_component [成分数]	[成分数] 成分の化学種	chemical_spec_for_component2=2
molecule_spec[成分番号]	[成分番号] 成分の分子形状. linear (直鎖), star (星形), pompom (ぼんぼん型), comb (櫛形), astar (非対称星形), diblock (ジブロック共重合体), triblock (トリブロック共重合体) が選択可能.	molecule_spec2=star
fraction[成分番号]	[成分番号] 成分の体積分立.	fraction1=1.0
Z[成分番号]	直鎖成分の分子量. 単位は NAPLES からみあい点間分子量	Z1=10
arm1Z[成分番号]	非対称星形分岐高分子の, 第一分岐部分の分子量. 単位は NAPLES からみあい点間分子量	arm1Z1=3
arm2Z[成分番号]	非対称星形分岐高分子の, 第二分岐部分の分子量. 単位は NAPLES からみあい点間分子量	arm2Z1=5
bridgeZ[成分番号]	ポンポン型および櫛形高分子の分岐間ブリッジの分子量. 単位は NAPLES からみあい点間分子量	bridgeZ2=20
arm_per_junction[成分番号]	分岐高分子の, 分岐あたりの腕の数	arm_per_junction3=3
arm1_per_junction[成分番号]	非対称星形分岐高分子の, 第一分岐部分の腕の数	arm1_per_junction3=3
arm2_per_junction[成分番号]	非対称星形分岐高分子の, 第二分岐部分の腕の数	arm2_per_junction3=3
junction_number[成分番号]	ポンポン型および櫛形高分子の 1 分子あたりの分岐点の数	junction_number4=3
chemical_spec_for_arm_of_component [成分番号]	ポンポン形および櫛形高分子の共重合体の場合, 分岐部分 (腕) の化学組成	chemical_spec_for_arm_of_component2=2
block_ratio1_[成分番号]	直鎖ブロック共重合高分子の第一ブロック部分のブロック比	block_ratio1_1=0.5

パラメーター	意味	例
block_ratio2_[成分番号]	直鎖ブロック共重合高分子の第二ブロック部分のブロック比（トリブロック共重合体の場合のみ有効）	block_ratio2_1=0.5
chemical_id_for_block1_ [成分番号]	直鎖ブロック共重合高分子の第一ブロック部分の化学組成	chemical_id_for_block1_1=1
chemical_id_for_block2_ [成分番号]	直鎖ブロック共重合高分子の第二ブロック部分の化学組成	chemical_id_for_block2_1=2
chemical_id_for_block3_ [成分番号]	直鎖ブロック共重合高分子の第三ブロック部分の化学組成（トリブロック共重合体の場合のみ有効）	chemical_id_for_block3_1=1
crosslink_mode	架橋点生成指示. none または random が指定できる.	crosslink_mode=random
crosslink_fraction	からみあい点における架橋作成の割合.	crosslink_fraction=0.1
crosslink_chem	架橋点生成の化学種.	crosslink_chem=1
chain_av_flag	stat.npls ファイルを出力するためのキー. デフォルトは 0（出力されない.）数字が指定された場合は, 出力する時間のインターバルをしめす.	chain_av_flag=10
measure_flag	measure.npls ファイルを出力するためのキー. デフォルトは 1. 出力する時間のインターバルをしめす.	measure_flag=1
snapshot_save_flag	config[保存時刻].npls ファイルを出力するためのキー. デフォルトは 0（出力されない.）数字が指定された場合は, 出力する時間のインターバルをしめす.	snapshot_save_flag=1000
free_end_stat_flag	tailZdist[保存時刻].npls および tailNdist[保存時刻] ファイルを出力するためのキー. デフォルトは 0（出力されない.）数字が指定された場合は, 出力する時間のインターバルをしめす.	free_end_stat_flag=1000
FENE_flag	tailZdist[保存時刻].npls 伸びきり効果を制御するキー. デフォルトは false で導入されない.	FENE_flag=true
maximum_stretch_for_chemical[化学種]	伸びきり効果が指定された場合に最大伸長比を規定する. デフォルト値は 4.	maximum_stretch_for_chemical1=10
friction_reduction_flag	分子の伸長配向に依存する分子摩擦を制御するキー. デフォルトは false で導入しない.	friction_reduction_flag=true

## 4.9 UDF ファイルリファレンス

UDF 形式のファイルに設定するパラメータ名を記載する. パラメータ名は NAPLES オリジナルのものとはほとんど同じであり, 意味も同様であるので, 詳細は NAPLES オリジナルのパラメータの説明を参照して頂きたい.

Name	Type	Value	
input.udf		-	-
Chemical_Component[]	struct array	-	-
Chemical_Component[0]	struct	-	-
Name	string	chemical_A	
Id	int	1	
G0	double	2600000.0	[Pa]
G_nap	double	4160000.0	
Me_nap	double	647.15	[g/mol]
tau_d	double	0.0	[s]
chi12	double	0.0	
bond_ratio_for_chemical	double	0.0	
friction_ratio_for_chemical	double	0.0	
Component[]	struct array	-	-
Component[0]	struct	-	-
Simulation_Conditions	struct	-	-
Simulation_Parameters	struct	-	-

図 4.7: Gourmet で UDF ファイルを開いた例

## Chemical\_Component[]

UDF 形式の入力ファイルでは化学種は配列として設定する。化学種の数配列の数として自動認識されるので設定しなくて良い。

- Name : 任意の化学種の名前。
- Id : 1 または 2 を設定する。
- G0 : プラトー弾性率 G0[Pa]。特に設定しなくてもよい。
- G\_nap : NAPLES 単位応力 [Pa]。出力結果の有次元化に用いる。必須ではない。
- Me\_nap : NAPLES 絡み合い点間分子量 Me[g/mol]。特に設定しなくてもよい。
- tau\_d : 最長緩和時間 [s]。特に設定しなくてもよい。
- tau\_e : 単位時間 [s]。出力結果の有次元化に用いる。必須ではない。
- chi12 : 化学種 1 と 2 の相互作用パラメータ。化学種 2 が存在する場合のみ登録すればよい。
- bond\_ratio\_for\_chemical : 化学種 1 に対する 2 の絡み合いセグメント長さの比。化学種 2 が存在する場合のみ登録すればよい。
- friction\_ratio\_for\_chemical : 化学種 1 に対する 2 のセグメントあたりの摩擦係数の比。化学種 2 が存在する場合のみ登録すればよい。
- FENE\_flag : 伸びきり効果を制御するキー。デフォルトは false で導入されない。
- maximum\_stretch\_for\_chemical[] : 伸びきり効果が指定された場合に最大伸長比を規定する。デフォルト値は 4。
- friction\_reduction\_flag : 分子の伸長配向に依存する分子摩擦を制御するキー。デフォルトは false で導入しない。

## Component[]

UDF 形式の入力ファイルでは構成成分は配列として設定する。成分の数は配列の数として自動認識されるので設定しなくて良い。

- fraction : 体積分率.
- molecule\_spec : 分子形状 (linear, star, pompom, comb, astar, diblock, triblock )
- linear.chemical\_spec\_for\_component : 成分の化学種番号
- linear.Z : 直鎖成分の分子量
- star.chemical\_spec\_for\_component : 成分の化学種番号
- star.armZ : 分岐点から伸びる腕の分子量
- star.arm\_per\_junction : 分岐点から伸びる腕の数
- astar.chemical\_spec\_for\_arm1\_of\_component : 第 1 分岐部分の化学種番号
- astar.arm1Z : 第 1 分岐部分の分子量
- astar.arm1\_per\_junction : 第 1 分岐部分の腕の数
- astar.chemical\_spec\_for\_arm2\_of\_component : 第 2 分岐部分の化学種番号
- astar.arm2Z : 第 2 分岐部分の分子量
- astar.arm2\_per\_junction : 第 2 分岐部分の腕の数
- pompom.chemical\_spec\_for\_component : 成分の化学種番号
- pompom.armZ : 分岐点から伸びる腕の分子量
- pompom.arm\_per\_junction : 分岐点から伸びる腕の数
- pompom.chemical\_spec\_for\_arm\_of\_component : 腕部分の化学種番号
- pompom.bridgeZ : ブリッジ部分の分子量
- comb.chemical\_spec\_for\_component : 成分の化学種番号
- comb.armZ : 分岐点から伸びる腕の分子量
- comb.arm\_per\_junction : 分岐点から伸びる腕の数
- comb.chemical\_spec\_for\_arm\_of\_component : 腕部分の化学種番号
- comb.bridgeZ : ブリッジ部分の分子量
- comb.junction\_number : 1 分子あたりの分岐点の数
- diblock.chemical\_id\_for\_block1\_ : 第 1 ブロック部分の化学種番号
- diblock.block1Z : 第 1 ブロック部分の分子量
- diblock.chemical\_id\_for\_block2\_ : 第 2 ブロック部分の化学種番号
- diblock.block2Z : 第 2 ブロック部分の分子量
- diblock.block\_ratio1\_ : 第 1 ブロック部分のブロック比. diblock の場合には diblock.block1Z, diblock.block2Z の比と一致するようにする. トリブロック共重合体の場合には tri.block.block1Z, tri.block.block2Z, tri.block.block3Z の比と一致するようにする.



- `tri.block.chemical_id_for_block1_` : 第 1 ブロック部分の化学種番号
- `tri.block.block1Z` : 第 1 ブロック部分の分子量
- `tri.block.chemical_id_for_block1_` : 第 2 ブロック部分の化学種番号
- `tri.block.block1Z` : 第 2 ブロック部分の分子量
- `tri.block.chemical_id_for_block1_` : 第 3 ブロック部分の化学種番号
- `tri.block.block1Z` : 第 3 ブロック部分の分子量
- `tri.block.block_ratio1_` : 第 1 ブロック部分のブロック比. `diblock` の場合と同じ注意が必要.
- `tri.block.block_ratio2_` : 第 2 ブロック部分のブロック比. `diblock` の場合と同じ注意が必要.

## Simulation\_Conditions

- `seed` : 乱数の種
- `cell_size` : セルサイズ
- `C` : セグメント数密度
- `p2` : 時間刻み
- `quit_dynamics_count` : トータルの計算時間
- `pre_reptation_step` : 流れのない平衡化時間
- `flow_type` : 流れ場の選択
- `step_shear.initial_strain` : ステップ変形を与える際のひずみ量
- `uniaxal_step.initial_strain` : ステップ変形を与える際のひずみ量
- `biaxal_step.initial_strain` : ステップ変形を与える際のひずみ量
- `planar_step_step.initial_strain` : ステップ変形を与える際のひずみ量
- `start_up_shear.strain_rate` : スタートアップ流れの際のひずみ速度
- `uniaxal.strain_rate` : スタートアップ流れの際のひずみ速度
- `biaxal.strain_rate` : スタートアップ流れの際のひずみ速度
- `planar.strain_rate` : スタートアップ流れの際のひずみ速度
- `dynamic_shear.strain_amplitude` : 動的せん断モードの際のひずみ振幅
- `dynamic_shear.strain_omega` : 動的せん断モードの際の周波数
- `dynamic_shear.dynamic_strain_cycle` : 動的せん断モードの際のひずみサイクル数
- `gelation.crosslink_mode` : 架橋点生成指示. `none` または `random`
- `gelation.crosslink_fraction` : からみあい点における架橋作成の割合
- `gelation.crosslink_chem` : 架橋点生成の化学種

- chain\_av\_flag : stat.npls ファイルを出力するためのキー
- measure\_flag : measure.npls ファイルを出力するためのキー
- snapshot\_save\_flag : config[保存時刻].npls ファイルを出力するためのキー
- free\_end\_stat\_flag : tailZdist[保存時刻].npls および tailNdist[保存時刻] ファイルを出力するためのキー

## Simulation Parameters

- restart\_flag : リスタートフラグ (0:new, 1:restart, 2:deform)
- simulation\_type : 計算モード. normal または frequency\_scan. 特に設定しなくてもよい.
- frequency\_scan.wmin : 最小周波数. 実際には引数で与えるので, 特に設定しなくてもよい.
- frequency\_scan.wmax : 最大周波数. 実際には引数で与えるので, 特に設定しなくてもよい.
- frequency\_scan.div : 分割数 (one decade). 実際には引数で与えるので, 特に設定しなくてもよい.

5章の出力ファイルを以下の該当項目にフルパスで記載する. これによって, 結果の解析のためのアクションを実行することが可能となる. 解析アクションの具体的な内容は, 第3章を参照していただきたい. 以下の各項目が未入力の場合, 解析アクションを実行した時点で UDF ファイルディレクトリから該当する出力ファイルが検索され, 該当項目に設定される.

## Output

- movelog.out : 状態報告ファイル
- prmcheck.out : 実行パラメータ確認ファイル
- mesure.out : 応力出力ファイル
- stat.out : 分子形態統計値出力ファイル
- finalconf.out : 最終形態保存ファイル
- config.out : 途中経過保存ファイル
- bistress.out : 二軸伸長応力の時間変化
- bistress\_smooth.out : 上記のスムージングされた結果
- plstress.out : 平面伸長応力の時間変化
- plstress\_smooth.out : 上記のスムージングされた結果
- shearstress.out : せん断応力の時間変化
- shearstress\_smooth.out : 上記のスムージングされた結果
- unistress.out : 一軸伸長応力の時間変化
- unistress\_smooth.out : 上記のスムージングされた結果
- gt.out : 線形せん断緩和弾性率ファイル
- gt\_smooth.out : 上記のスムージングされた結果

- et\_out : 線形せん断粘度成長ファイル
- et\_smooth\_out : 上記のスモーディングされた結果
- gw\_out : 動的粘弾性ファイル (線形緩和弾性率をフーリエ変換する方法,)
- gw\_smooth\_out : 上記のスモーディングされた結果
- gwdirect\_out : 動的粘弾性ファイル (実験と同じように周波数の異なる三角関数状のひずみを与える方法)
- tailNdist\_out : 架橋構造解析ファイル (末端部分のモノマー数分布)
- tailZdist\_out : 架橋構造解析ファイル (末端部分が形成するからみあい点個数の分布)



## 第5章 NAPLES出力ファイル

NAPLES と解析ツールにより、様々なファイルが生成される。2.4 節により付属する解析ツールをそのまま用いた場合、それぞれの処理結果ファイルは以下のように定められている。これらは、次の解析処理を行った時には自動的に上書きされてしまうので、必要に応じて退避させる必要がある。なお、すべてのファイルはテキストファイルであり、テキストエディタで閲覧したり、グラフ描画ソフトに読みこませたりすることが可能である。

UDF 形式の入力ファイルを用いた場合には、各出力ファイルは入力 UDF ファイルからリンクされる。入力 UDF ファイル上のアクションによって、応力解析およびプロット機能呼び出すことが可能である。

- `bistress.npls`, `bistress_smooth.npls` : 二軸伸長応力の時間変化ファイルとそのスムージングされた結果ファイル。内容の詳細は 5.7 節を参照。
- `configXXXX.npls` : レオロジー計算モードで、計算の途中経過を保存するファイル。計算途中の状態を可視化するなどの目的に使用できる。詳細は 5.6 節を参照。
- `et.npls`, `et_smooth.npls` : 線形せん断粘度成長曲線とそのスムージングされた結果ファイル。内容の詳細は 5.9 節を参照。
- `finalconf.npls` : レオロジー計算モードで、計算をリスタートするための最終計算状態を保存するファイル。詳細は 5.5 節を参照。
- `gt.npls`, `gt_smooth.npls` : 線形せん断緩和弾性率曲線とそのスムージングされた結果ファイル。内容の詳細は 5.8 節を参照。
- `gw.npls`, `gw_smooth.npls` : 動的粘弾性カーブ（貯蔵弾性率  $G'$ 、損失弾性率  $G''$ ）とそのスムージングされた結果ファイル。内容の詳細は 5.10 節を参照。
- `gwdirect.npls` : 動的粘弾性カーブ（貯蔵弾性率  $G'$ 、損失弾性率  $G''$ ）。内容の詳細は 5.10 節を参照。
- `input.npls` : 起動した際に、引数で Input ファイルを指定しない場合にデフォルトで読み込む入力パラメーターファイル。内容の詳細は 4 節を参照。
- `measure.npls` : NAPLES がレオロジー計算をする際に出力する応力ファイル。内容の詳細は 5.3 節を参照。このファイルを元にして種々のレオロジー量が計算できる。
- `movelog.npls` : NAPLES の計算状況が記録されるログ。内容の詳細は 5.1 節を参照のこと。
- `plstress.npls`, `plstress_smooth.npls` : 平面伸長応力の時間変化ファイルとそのスムージングされた結果ファイル。
- `prmcheck.npls` : NAPLES がレオロジー計算を開始するときに、計算条件を確認のために出力するファイル。内容の詳細は 5.2 節を参照のこと。
- `shearstress.npls`, `shearstress_smooth.npls` : せん断応力の時間変化ファイルとそのスムージングされた結果ファイル。

- stat.npls : NAPLES が出力する分子の形態に関する情報を納めたファイル. 内容の詳細は 5.4 節を参照のこと.
- tmp.npls : 解析の途中に生じる中間作業ファイル. 削除しても差し支えない.
- unistress.npls, unistress\_smooth.npls : 一軸伸長応力の時間変化ファイルとそのスムージングされた結果ファイル.

## 5.1 状態報告ファイル movelog.npls

naplesOCTA(.exe) が正常に実行されると, しばらくして naplesOCTA(.exe) と同じディレクトリに生成される. 中身は以下のようにになっている.

————— 正常実行時の movelog.npls の内容 —————

```
-----
NAPLES is started with parameters in input.npls Thu Oct 17 18:32:40 2002
Initialization started. Thu Oct 17 18:32:40 2002
Main calculation started. 0   Thu Oct 17 18:32:46 2002
EQUILIBRIUM_CALC Thu Oct 17 18:32:46 2002
```

movelog.npls ファイルでは, イベントが起きた日時をエラーメッセージや起動ログと同時に記録する. 上の例では, input.npls という入力パラメーターファイルにて NAPLES が起動しており, まず初期状態を作成した後, 6 秒ほどして流れのない平衡状態での計算を開始したというメッセージが出力されている. 流動変形を与えた場合, 平衡化プロセスも以下のように記録される.

————— ステップせん断変形正常実行時の movelog.npls の内容 —————

```
SINGLE_STEP_CALC Thu Oct 17 18:44:37 2002
PRE_RUN 0   Thu Oct 17 18:44:37 2002
STEP_SHEAR_APPLIED 3   Thu Oct 17 18:44:59 2002
END 10   Thu Oct 17 18:46:08 2002
```

この例では, 平衡化を 3 ステップ行い, その後ステップせん断変形が与えられ, 10 ステップの計算を行って終了している. なお念のためこれらはマニュアル用のテスト計算なのでレオロジー計算ではもっと長い時間の計算が必要である. また, コマンドラインから起動し, コマンドラインで Ctrl-C キーにより naplesOCTA(.exe) を強制終了した場合には ENDS メッセージは記録されない.

入力パラメーターファイルの内容に間違いがあった場合には, 以下のようなメッセージが出力される.

————— パラメーター入力パラメーターファイル中に間違いがあった時の movelog.npls のメッセージ —————

```
Description in the data file, [xxx], is meaningless Thu Oct 17 18:37:36 2002
```

これは, naplesOCTA(.exe) を起動する際に選択した入力パラメーターファイルにおいて, xxx なるパラメーターがあり, 意味がない, と警告している. また, 入力パラメーターファイルが指定されず, input.npls ファイルも読み込むことができなかった場合には, 以下のメッセージを出力してあらかじめプログラムに埋め込まれた条件で起動する.

————— NAPLES に組み込まれたパラメーターで起動したことを示す movelog.npls のメッセージ —————

```
Parameters are set as default Thu Oct 17 18:39:36 2002
```

この場合の起動パラメータについては, 次節で説明する prmcheck.npls ファイルを見ることで把握できる.

movelog.npls ファイルには naplesOCTA(.exe) を起動するたびに以上のような情報が書き足される形で書き込まれていくので, 適宜削除する必要がある.

## 5.2 実行パラメータ確認ファイル prmcheck.npls

naplesOCTA(.exe) を起動すると, prmcheck.npls ファイルが生成される. 中身は以下のようにになっている.

prmcheck.npls の内容

```
parameter_file=input.npls          「入力パラメーターファイルを確認」
calculation_mode=new_calculation    「計算の開始モード」
%-----
cell_size=8.000000                 「計算の周期境界セルの1辺の長さ」
C=10.000000                        「システム内セグメント数密度」
p2=0.010000                        「計算の時間刻み幅」
%-----
flow_type=single_step_shear        「流れの形態」
initial_strain=0.500000             「ステップ状に与えるひずみの量」
pre_run=3                           「変形前の平衡化時間」
%-----
chemical_component_number=2
chi12=2.000000
bond_ratio_for_chemical2=1.500000
friction_ratio_for_chemical2=2.000000
#-----
total_nodes_number=11544            「系に含まれる全セグメント数」
total_molecule_number=598          「系の全分子数」
component_number=3                  「系の構成成分数」
#---component 1 ---
chemical_spec_for_component1=1
N_1=512                             「第一成分分子数」
molecule_spec1=linear              「第一成分形状」
totalZ1=3.000000                    「第一成分総分子量」
fraction1=0.300000                  「第一成分体積分率」
#---component 2 ---
chemical_spec_for_component2=1
N_2=38
molecule_spec2=star
...
```

まず parameter\_file として, 入力パラメーターファイル名を確認表示している. 次に calculation\_mode で, 新規に計算を開始したか, あるいは既存の最終形態保存ファイルを用いて計算を再開したかを確認している. その次のブロックには, 計算条件が書き込まれる. これらの詳細は 4.7 節を参照のこと.

次に, 流動に関する計算条件が書き込まれる. これは入力パラメーターファイルで指定したものがほぼそのまま出力されるはずなので確認できる. 詳しくは 4.6 節を参照のこと.

最後のブロックに, 計算しているシステム (試料) に関する情報が出力される. total\_nodes\_number は, 計算空間中に存在するからみあいセグメントの総数である. この値が 100,000 を超えるようだと, 相当計算負荷が高くなるので計算時間がかかることを覚悟しなければならない. total\_molecule\_number は, 全成分の分子数を合計したシステム全体に含まれる分子数である. component\_number はシステムの構成成分数 (4.3.4 節を参照) である. 以下, 各成分の情報が書き出されている.

prmcheck.npls ファイルは, naplesOCTA(.exe) が起動するたびに上書きされるので, 必要があれば名称を変

えて保存しておく。

### 5.3 応力出力ファイル measure.npls

naplesOCTA(.exe) を起動する目的はレオロジーの計算であるが、そのレオロジー量がこのファイルに書き出されている。他のファイルと同様に measure.npls もテキストファイルなのでノートパッドなどの適当なエディタで開くと、measure.npls の内容は以下になっている。

measure.npls の内容 (例)					
0	0.00e+00	-3.060009e-02	1.820663e+00	1.652494e+00	2.459011e+00
1	0.00e+00	-1.483576e-02	1.167096e+00	1.146231e+00	1.269184e+00
2	0.00e+00	-7.436174e-03	1.107850e+00	1.144368e+00	1.191704e+00
3	0.00e+00	-1.660099e-02	1.163289e+00	1.137091e+00	1.193193e+00
0	0.00e+00	5.519445e-01	1.430961e+00	1.137091e+00	1.193193e+00
1	0.00e+00	1.126641e-01	1.273881e+00	1.186699e+00	1.205338e+00
2	0.00e+00	8.506818e-02	1.227362e+00	1.163581e+00	1.177303e+00
3	0.00e+00	4.590036e-02	1.193680e+00	1.152211e+00	1.175619e+00
4	0.00e+00	3.322798e-02	1.158326e+00	1.172665e+00	1.141329e+00

ここで 1 列めは、計算時間を NAPLES の単位時間である  $\tau_e$  の時間スケールで表している。実時間に変換するには、3.8.1 節に従って  $\tau_e$  を求め、1 列目の数値に掛け合わせればよい。2 列めは系に加えられた総ひずみ量を示す。3 列目は  $\sigma_{12}$  (せん断応力)、4 列目から 6 列目までが  $\sigma_{11}, \sigma_{22}, \sigma_{33}$  (法線応力) である。これらの応力値は、NAPLES における値にスケールされており無次元量である。実単位に変換するには、単位応力値を掛ける。詳しくは 3.8.2 節を参照のこと。

計算したシステムが複数の構成成分をもつ場合、応力データは構成成分ごとに計算され、列が追加されていく。例えば 2 成分からなる系の応力を計算すると、まず 3 列目から 6 列目に系全体の応力、7 列目から 10 列目に第 1 成分が発生している応力、11 列目から 14 列目に第 2 成分が発生している応力がそれぞれ書き込まれる。

measure.npls ファイルは、naplesOCTA(.exe) を実行するフォルダに存在しなければ新規に作成され、ファイルが存在すれば末尾に追加でデータが書き込まれる。

measure.npls ファイルはデフォルトでは毎ステップ (NAPLES の単位時間毎に) 応力を出力する。ファイル容量を減らしたいなどの理由で、このインターバルを調整するには、入力パラメーターファイル中に以下のように記述する。

measure.npls のインターバルを調整するためのパラメータ指定	
measure_flag=[書き出しをおこなう時間間隔 : 例 measure_flag=2]	

ただし、平衡状態の計算から線形粘弾性を求める場合などでは、誤差が大きくなる。またスタートアップ流れやステップ変形の場合にも、短時間領域での誤差が大きくなるので推奨しない。デフォルトのままで毎ステップの応力書き出しを行った後、解析ツール smooth.exe を用いて必要なデータにスムージングを施し、その後 measure.npls ファイルを破棄する方法を推奨する。

### 5.4 分子形態統計値出力ファイル stat.npls

NAPLES ではすべての分子の運動を計算するので、分子の形態に関する情報を入手することができる。stat.npls は、からみあいの個数、からみあい点間の平均距離、分子の末端間距離などの情報を納めているが、標準では出力されない。stat.npls を出力するには、入力パラメーターファイル中に以下のように記述する。



— stat.npls を出力するためのパラメータ指定 —

chain\_av\_flag=[書き出しをおこなう時間間隔：例 chain\_av\_flag=10]

この記述があると、以下のような内容のファイルを出力する。

— stat.npls の例 —

0	4.7709	1.0272	0.759200	0.226836	3.380538	8.335377
10	4.9818	1.1555	0.706237	0.187129	3.094384	6.412024

1 列目は計算時間を NAPLES の単位時間である  $\tau_e$  の時間スケールで表している。実時間に変換するには、3.8.1 節に従って  $\tau_e$  を求め、1 列目の数値に掛け合わせればよい。2 列目は、部分鎖あたりの平均セグメント数、3 列目はその分散、4 列目と 5 列目はからみあい点間の平均距離とその分散、6 列目と 7 列目は部分鎖あたりの平均末端間距離とその分散である。距離はからみあい点間距離  $a$  で無次元化されている。ここで部分鎖とは、分岐高分子の場合の枝部分や分岐点を接続するブリッジ部分をいう。

計算したシステムが複数の構成成分をもつ場合、これらのデータは 1 列目の計算時間を除いて構成成分ごとに計算され、列が追加されていく。例えば 2 成分からなる系を計算すると、まず 2 列目から 7 列目に系全体のデータ、8 列目から 13 列目に第 1 成分、14 列目から 19 列目に第 2 成分のデータがそれぞれ書き込まれる。

stat.npls ファイルは、naplesOCTA(.exe) を実行するフォルダに存在しなければ新規に作成され、ファイルが存在すれば末尾に追加でデータが書き込まれる。

## 5.5 最終形態保存ファイル finalconf.npls

このファイルには、停止された計算を再開させるために必要な naplesOCTA(.exe) のデータが格納されている。計算を再開するには入力パラメーターファイル中で calculation\_mode=restart と記述する必要がある。詳細な方法については、2.2.3 節を参照。finalconf.npls はテキストファイルなので、例えば UNIX から Windows など、異なるシステム間で転送する場合には、転送モードをテキストにする。これをしないで転送すると、改行コードの違いによりファイルが読み込まれず、計算を再開することができない。

このファイルの中身は以下のようにになっている。

— finalconf.npls の内容 —

```
#NAPLES configuration file
# for detail of NAPLES, go to http://masubuchi.jp/NAPLESWEB ->ここまでヘッダ
#Parameters used to generate this file: ->ここから起動パラメータ記述部
#parameter_file=input.npls
calculation_mode=continuous_flow_mode
#-----
cell_size=12.000000
C=10.000000
...                -> 起動パラメーター記述部
#Followings are the data storing the system configuration: ->ここからデータストア部
NAPLESrev11102005    ->バージョン名
3000    300000    14042004    ->単位ステップ数, 計算ループ数, 初期乱数シード
0.000000    0.000000    -0.000000    0 0    ->総ひずみ, 総せん断量, セルスライド量, ひずみカウン
タ, せん断カウンタ
12.000000    12.000000    12.000000    ->ユニットセルサイズ (1,2,3 軸)
1.000000    1.000000    1.000000    ->サブセルサイズ (1,2,3 軸)
1.000000    1.000000    1.000000    ->サブセルサイズの逆数 (1,2,3 軸)
1728  12  12  12    ->サブセル総数, 各軸 (1,2,3 軸)
14042004  798375284  4135353135  2331553198  3683403622  1  (mzran 用保存変数 6 個)

->各質点のデータ
[番号][化学種][種別][所属鎖][位置(xyz)][力(xyz)][ボンドフラグ][ボンドベクトル(xyz)][ボンド
長][モノマー数][すべり力][隣接質点(前・後)][からみあい対象]
..... 質点個数分だけ繰り返し

->部分鎖のデータ
[番号][初期ビーズ数][現ビーズ数][化学種][種別][誘電フラグ][所属分岐点][所属鎖群][絶対位置
(xyz)]
..... 部分鎖個数分だけ繰り返し

->分岐点データ
[番号][所属分子][所有末端個数][所有末端番号(個数分)]
..... 分岐点個数分だけ繰り返し

->鎖群データ
[番号][初期ビーズ数][化学種][鎖種別][Z 数]
..... 鎖群数分だけ繰り返し

->分子データ
[番号][所属成分][分子種別][所有分岐点数][所有分岐点番号(個数分)]
..... 分子数分だけ繰り返し
```

## 5.6 途中経過保存ファイル config[xxxx].npls

このファイルは途中経過を保存するために利用する。デフォルトでは出力されない。このファイルを出力するには、パラメーターファイルに以下のように記述する。

——— 途中経過ファイルを出力するためのパラメータオプション ———

`snapshot_save_flag=[途中経過を保存するステップ刻み：例 snap_shot_save_flag=1000]`

この記述があると、naplesOCTA(.exe) は、指定された時間刻み毎に config[そのときの時刻].npls というファイルを作成する。例えば時間刻みを 1000 と指定すると、config00000.npls, config01000.npls, config02000.npls, ... のように逐次出力されていく。ファイルの中身は finalconf.npls と同じであるが、finalconf.npls は出力される毎に上書きされるため過去の情報を保存しない。従って計算を再開する場所は常にさきの計算が終了した時点となる。一方、config[xxx].npls ファイルでは、好みの時刻から計算を再開できる。ただし、ファイル一つあたりの要領が大きいので、あまり時間刻みを小さくとるとディスクが埋め尽くされるので注意が必要である。config[xxx].npls ファイルを使って計算を再開するには以下の手順をとる。

1. config[xxx].npls ファイルの名前を finalconf.npls と変更する。
2. config[xxx].npls ファイルを出力するのに用いた入力パラメーターファイル中で calculation\_mode=restart と変更する。
3. NAPLESview.exe または naplesOCTA(.exe) を起動する。

naplesOCTA(.exe) を再開させる方法について詳しくは 2.2.3 節を参照のこと。

例として、D ドライブの Data フォルダにある config02000.npls ファイルと、そのファイルを作成するために使用した入力パラメーターファイル test.npls を用いて計算を再開する場合を示す。

——— config[xxx].npls を用いた計算再開時の例（cygwin の場合） ———

```
>cd d:/Data                                「データフォルダに移動」
>mv finalconf.npls finalconf.npls.org      「最終状態保存ファイルを退避」
>cp config02000.npls finalconf.npls        「途中経過ファイルを最終状態保存ファイルとする」
>cat test.npls | grep calculation_mode      「計算再開モード確認」
calculation_mode=restart                    「もしこの出力がなければエディタで付け加える」
>d:/NAPLES/naplesOCTA(.exe) test.npls &    「計算を再開」
```

なお、この例では naplesOCTA(.exe) は D ドライブの NAPLES フォルダにインストールしてあるものとしている。

## 5.7 伸長およびせん断応力データファイル群

5.3 節に説明があるように、NAPLES はその計算途中に、応力テンソルの 11,22,33,12 成分の時間変化を measure.npls に記録している。この measure.npls から、せん断応力や各種の伸長応力だけを抜き出すことが可能である。これを行うには、解析ツールを用いる。3 節をご覧ください。その結果、以下のようなファイルが生成される。

- bistress.npls, bistress\_smooth.npls：二軸伸長応力の時間変化とスムージングされた結果
- plstress.npls, plstress\_smooth.npls：平面伸長応力の時間変化とスムージングされた結果
- shearstress.npls, shearstress\_smooth.npls：せん断応力の時間変化とスムージングされた結果

- `unistress.npls`, `unistress_smooth.npls` : 一軸伸長応力の時間変化とスムージングされた結果

これらのファイルはすべてテキストファイルであり、適当なテキストエディタ（ノートパッドなど）で中身を見ると、内容は以下のようになっている。

— `bistress.npls` の内容（例） —

```
0.000000e+00 -3.933300e-02
1.000000e+00 -5.783300e-02
2.000000e+00 -9.034000e-03
3.000000e+00 -2.207400e-02
4.000000e+00 -4.046700e-02
5.000000e+00 -3.910700e-02
```

ここで 1 列目が時間、2 列目が応力である。時間と応力の単位は NAPLES の計算系で無次元化されている。実際の単位を持たせるには 6.5 節を参照のこと。また、NAPLES が出力する応力を粘度や弾性率に変換するには、実験と同じように以下の関係式を用いる。

$$\eta = \sigma / \dot{\epsilon} \quad (5.1)$$

$$G = \sigma / \epsilon \quad (5.2)$$

ここで  $\eta$  は粘度、 $\sigma$  は応力、 $G$  は弾性率、 $\epsilon$  はひずみ、 $\dot{\epsilon}$  はひずみ速度である。つまり、スタートアップ伸長流れやせん断流れを与えた結果の解析であれば、出力された応力をひずみ速度で割ってやれば伸長粘度やせん断粘度になる。またステップせん断変形やステップ伸長変形を与えた結果の解析であれば、出力された応力をステップで与えたひずみ量で割ってやれば緩和弾性率になる。こういった操作は実験と同じであり、詳しくは参考文献 [18] などをご覧いただきたい。なお、スムージングされた結果ファイルはすべて `smooth(.exe)` コマンドで作られており、時間軸をログで表示した場合に結果が見やすいようになっている。

## 5.8 線形せん断緩和弾性率ファイル `gt.npls`, `gt_smooth.npls`

線形緩和弾性率は、揺動散逸定理により平衡状態における応力の揺らぎの相関関数と関係づけられる。中身は以下のようになっている。

— `gt.npls` の内容（例） —

```
0 1.000000e+00
1 4.879472e-01
2 4.173991e-01
3 3.699413e-01
4 3.302976e-01
5 3.034629e-01
6 2.775755e-01
```

ここで一列目が時間、二列目に緩和弾性率の時間変化が入っている。応力でなく弾性率である点に注意が必要である。時間と弾性率の単位は NAPLES の計算系で無次元化されている。実際の単位を持たせるには 6.5 節を参照のこと。なお、`gt_smooth.npls` は `smooth(.exe)` コマンドで作られており、時間軸をログで表示した場合に結果が見やすいようになっている。

## 5.9 線形せん断粘度成長ファイル et.npls, et\_smooth.npls

5.8 節の線形せん断緩和弾性率が求められていれば、それを積分することにより線形粘度成長曲線を求めることができる。(計算の詳細は 2.4.1 節を参照。) 線形せん断粘度成長曲線とそのスムージングされた結果が入っている。(バッチコマンドの詳細は ?? 節参照) 中身は以下のようになっている。

et.npls の内容 (例)

```
1 0.00140398
2 0.0141274
3 0.0120504
4 0.00352317
5 0.00572446
```

ここで一列目が時間、二列目に線形せん断粘度の時間変化が入っている。応力でなく粘度である点に注意が必要である。トルートン則を用いることで、各種伸長変形の線形粘度曲線を得ることもできる [18]。時間と粘度の単位は NAPLES の計算系で無次元化されている。実際の単位を持たせるには 6.5 節を参照のこと。なお、et\_smooth.npls は smooth(.exe) コマンドで作られており、時間軸をログで表示した場合に結果が見やすいようになっている。

## 5.10 動的粘弾性ファイル gw.npls, gw\_smooth.npls, gwdirect.npls

動的粘弾性プロファイル (貯蔵弾性率  $G'$  および損失弾性率  $G''$ ) を求めるための方法は大きく分けて 2 通りある。一つは、線形緩和弾性率をフーリエ変換する方法、もう一つは、実験と同じように周波数の異なる三角関数状のひずみを与える方法、である。NAPLES では、そのどちらの方法もサポートしている。

まず線形緩和弾性率からフーリエ変換で求めるには、gt2gw(.exe) コマンド (詳細は 2.4.2 節) を用いる。NAPLES が出力する応力ファイル measure.npls からまず線形緩和弾性率を求め、その結果を gt2gw.exe コマンドに与えて動的粘弾性プロファイルを得る。結果が gw.npls に出力される。処理の詳細は 3.3 節をご覧ください。gw.npls を gwsmooth.exe コマンドでスムージングすると gw\_smooth.npls が生成される。gw.npls ファイルの中身は以下のようになっている。

gw.npls の内容 (例)

```
8.000000e-01 1.052871e-01 8.982074e-02
6.000000e-01 1.059345e-01 8.313399e-02
4.000000e-01 9.529423e-02 7.436524e-02
3.000000e-01 8.005790e-02 6.357562e-02
2.000000e-01 6.599811e-02 6.273698e-02
1.000000e-01 4.031614e-02 5.399160e-02
8.000000e-02 3.233802e-02 4.980498e-02
```

ここで一列目が周波数、二列目が貯蔵弾性率  $G'$ 、三列目が損失弾性率  $G''$  である。

次に実験と同様の周波数を変化させる方法では、naplesgwOCTA(.exe) コマンド (2.4.6 節) を用いる。この結果生成される gwdirect.npls ファイルの中身は上記の gw.npls と同様である。これらのファイルでは時間と弾性率の単位は NAPLES の計算系で無次元化されている。実際の単位を持たせるには 6.5 節を参照のこと。

## 5.11 架橋構造解析ファイル tailNdist.npls, tailZdist.npls

架橋化させた構造に残るフリーな末端の分布を解析した結果が記録されるファイル。tailNdist.npls には、末端部分のモノマー数分布が出力され、tailZdist.npls には、末端部分が形成するからみあい点個数の分布が出力

される。このうち tailNdist.npls は一度架橋化させた後は変化しないが、tailZdist.npls は、からみあい個数なので架橋後も変化する。特に大変形を与えた後は時間変化する可能性が高い。

—— tailZdist.npls および tailNdist.npls を出力するためのパラメータ指定 ——

`free_end_stat_flag`=[書き出しをおこなう時間間隔：例 `free_end_stat_flag=100`]

この記述があると、以下のような内容のファイルを出力する。

—— tailZdist.npls の内容 ——

## [記録が行われた時間]

[Z の値] [頻度]

.....

tailNdist.npls も同じフォーマットになっている。

## 第6章 計算原理

NAPLES の計算原理を理解することで NAPLES への入力パラメーターや実験とのフィッティングに関する理解を助けるものと考えられるので、本章では NAPLES の理論モデル、計算に用いている式、その適用範囲と適用限界について述べた。NAPLES を利用する上で必須ではないが、利用者の一助となれば幸いである。

### 6.1 緒言

多数の高分子がからみあった系は、1) 分子量が数万から数百万に及ぶこと、2) からみあい起因する、数百から数千秒以上の時間スケールを持つスローダイナミクス（長時間緩和現象）が重要であること、などの理由により、既存の分子シミュレーションで扱うのが非常に困難な系である。粗視化ポテンシャルを用いたいわゆるバネ-ビーズタイプのシミュレーション [14] でも、分子あたりのからみあい点が数十をこえるような分子の緩和を調べることは事実上不可能にちかい。そこでさらにレベルをあげた粗視化を行うために、土井-Edwards のレプテーション理論 [15] に基づくシミュレーション手法がいくつか提案されている [16]。しかしこれらは本質的に一本鎖問題を解くものであり、多数の分子の間での様々な相互作用や位置関係、複雑な分子形状などをあつかうことはできない。また、分子間相互作用を平均場的に解いてマイクロ相分離構造をもとめる密度汎関数理論に基づくシミュレーターもある [17] が、この理論ではからみあいを扱うことに原理的な困難があるため、からみあった高分子の系におけるダイナミクスの計算は難しく、ましてレオロジーを予測することはできない。

NAPLES は、レプテーション理論に基づく粗視化を行いながらも、既存の分子シミュレーションと同様に分子の形や相互作用を様々に取り入れることができる新しい手法である。NAPLES の基となっている理論モデルの名称は primitive chain network モデルという。この名称はレプテーション理論の管状の束縛が primitive chain とか primitive path とか呼ばれることに起因している。オリジナルのレプテーション理論では primitive chain を 1 本だけ取り出して議論するが、このモデルではそのネットワークを構築してリアルな分子秒像を埋め込んでいるので、primitive chain network モデルと呼んでいる。

### 6.2 Primitive Chain Network モデル

まず高分子を、からみあい点間分子量程度の要素（チューブ要素）に分割する [1]。分割点がからみあい点であり、そこでは他の分子とからみあいを形成する。からみあいは、スリップリングとよばれる束縛に置き換える。各スリップリングには 2 本の鎖が入っていて、それぞれ、末端からしか抜けることができないと仮定する。（高分子はレプテーション運動によってのみからみあいをはずすことができると仮定する。）

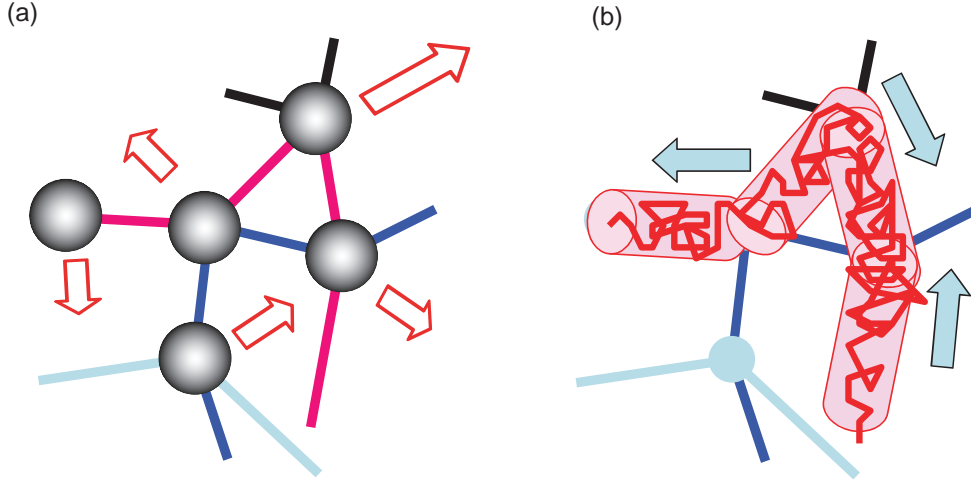


図 6.1: Primitive Chain Network モデルの概念図. (a) からみあい点の運動. (b) チューブ束縛内部の高分子のすべり運動.

高分子の運動は、このスリッリンクの運動 (図 6.1(a)) と、チューブ内部のモノマーの輸送 (図 6.1(b)) により記述されるとする. まずスリッリンクの運動方程式は、以下の Langevin 方程式で書くことができる.<sup>1</sup>

$$\zeta \frac{d\mathbf{R}_i}{dt} = \frac{3kT}{b^2} \sum_j^4 \frac{\mathbf{r}_j}{n_j} - n_0 \nabla \mu + \mathbf{F} \quad (6.1)$$

ここで左辺はスリッリンクが運動することによって媒体から受ける流体抵抗力であり、 $\zeta$  はチューブ要素あたりの抵抗、 $\mathbf{R}$  はスリッリンクの位置を示す. 抵抗係数  $\zeta$  は通常定数と見なされるが、伸長と配向に依存する関数とする場合もある [6].

右辺第 1 項がからみあいにかかる張力のバランス、力学平衡を表している.  $b$  はチューブに含まれる Kuhn の統計セグメントのサイズ、 $\mathbf{r}$  はスリッリンクに接続されているチューブ要素ベクトル、 $n$  は接続されているチューブ要素内に含まれる Kuhn セグメントの数を示している. 図 6.1(a) から分かるように、各スリッリンクは 2 本の鎖からなっていると考えているので、スリッリンクに接続されているからみあいセグメントの数は 4 である. 鎖の伸びきを考慮する場合は、FENE-P と呼ばれるモデルに従ってバネ係数が変化する [6].

右辺第 2 項は力学平衡に起因するネットワークの変形に伴う熱力学的な力 (浸透圧) で  $\mu$  は化学ポテンシャルである. 化学ポテンシャルは自由エネルギーと以下の関係にある.

$$\mu(\mathbf{R}) = \frac{\partial F}{\partial n(\mathbf{R})} \quad (6.2)$$

自由エネルギー  $F$  は、体積を保存するための以下を仮定している.

$$\frac{F}{kT} = \begin{cases} \epsilon \left( \frac{n(\mathbf{R})}{\langle n \rangle} - 1 \right)^2 & \text{for } n(\mathbf{R}) > \langle n \rangle \\ 0 & \text{for } n(\mathbf{R}) \leq \langle n \rangle \end{cases} \quad (6.3)$$

右辺第 3 項はランダム力である. このランダム力は揺動散逸定理により以下で書くことができる.

$$\langle \mathbf{F} \rangle = \mathbf{0} \quad (6.4)$$

$$\langle \mathbf{F}_i(t) \cdot \mathbf{F}_j(t') \rangle = 6kT\zeta\delta_{ij}\delta(t-t') \quad (6.5)$$

<sup>1</sup>慣性項は、高分子液体のように粘性が強い場合には他の項に比べて寄与が小さいので無視する.



チューブ中のモノマーの運動 ( $i$  番目のチューブセグメントに含まれる Kuhn セグメントの数  $n_i$  の変化) は以下の 1 次元の Langevin 方程式で計算する.

$$\frac{\zeta}{2\rho} \frac{dn_i}{dt} = \frac{3kT}{b^2} \left( \frac{r_{i+1}}{n_{i+1}} - \frac{r_i}{n_i} \right) - n_i \text{grad}\mu + f \quad (6.6)$$

$$\rho = \frac{1}{2} \left( \frac{n_{i+1}}{r_{i+1}} + \frac{n_i}{r_i} \right) \quad (6.7)$$

この式はチューブに沿ったモノマーの運動, つまり 1 次元の物質輸送の式を表すが, 式中の各項は式 6.1 と矛盾がないようになっている. よって, 式 6.1 で  $\zeta$  を分子の配向と摩擦の関数とする場合は式 6.7 でも同様に扱う.

左辺は運動に伴う粘性抵抗を表しており,  $\rho$  はモノマーの局所密度を表す.

右辺第 1 項は隣接したチューブセグメント間でのローカルな張力の差である. 張力が大きい方にモノマーは引っ張り込まれる. 鎖の伸びきり効果を考える場合は FENE-P モデルに従ってバネ定数を変化させる.

右辺第 2 項は熱力学的力である. 化学ポテンシャルの定義は式 6.1 と等しい. ここで化学ポテンシャルを計算する際の局所モノマー濃度  $n(\mathbf{R})$  は, 周囲の別の鎖に属するモノマーも考えたものであり, 自分自身のチューブ内部での局所モノマー濃度である  $\rho$  とは別のものである.

第 3 項はランダム力である. ただし 1 次元のランダム力で, しかも今考えている鎖は 1 本なので (式 6.5 はからみあっているので 2 本) ファクター  $1/6$  が必要で以下のようになる.

$$\langle f_i \rangle = 0 \quad (6.8)$$

$$\langle \mathbf{f}_i(t) \mathbf{f}_j(t') \rangle = kT \zeta \delta_{ij} \delta(t - t') \quad (6.9)$$

からみあいトポロジーの組み替えは, 逐次末端チューブセグメントに存在するモノマー数  $n$  を監視し, 式 (3) で示される範囲を越えたらからみあいの生成・消滅を行う.

$$\frac{1}{2} < \frac{n}{n_0} < 1 + \frac{1}{2} \quad (6.10)$$

図 6.2 に, primitive chain network モデルによるシミュレーションのスナップショットを示す.

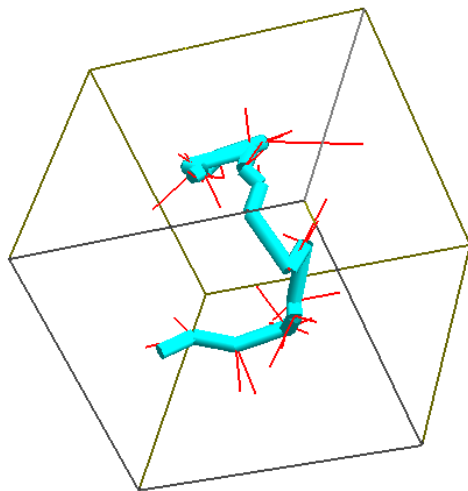


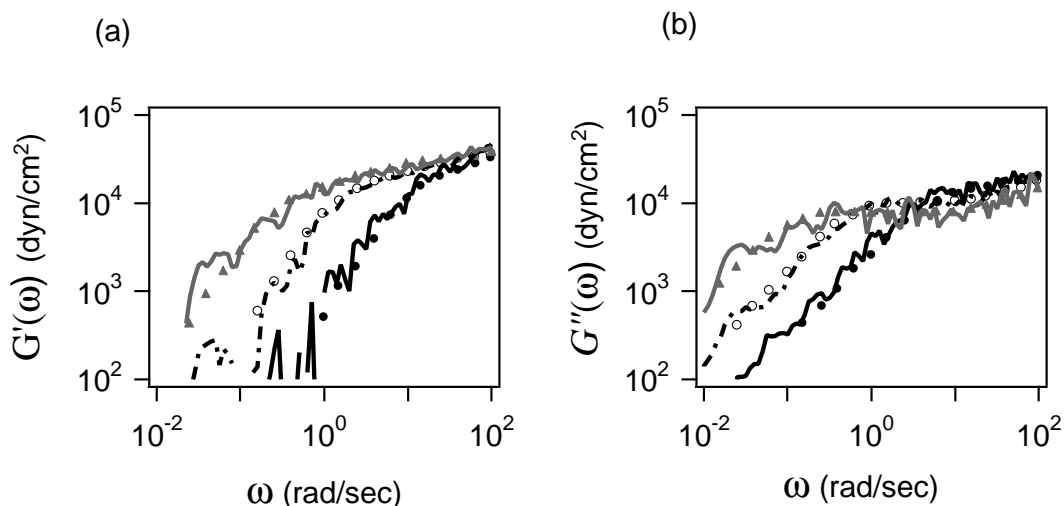
図 6.2: primitive chain network モデルの典型的スナップショット. 多数の鎖が系にはあるが, 一本だけを抜き出して表示. トゲのように見える細い線は, からみあっている周囲の分子のセグメント.

注目すべきは計算で扱う分子の分子量, および計算可能時間である. primitive chain network モデルでは 1 セグメントがからみあい点間分子量のオーダーであるから, 10 セグメントもあれば, 数万の分子量をもつ高分子に相当する. また, 計算の単位時間は

$$\tilde{\tau} = \frac{\zeta a^2}{6kT} \quad (6.11)$$

で与えられるが, なにしる  $a$  がからみあいセグメントの大きさでそもそも分子量 1 万程度の分子のサイズ,  $\zeta$  がその粘性抵抗係数であるから, 実時間にして  $10^{-3} \sim 10^{-1}$  秒程度になるのである.

このモデルで計算すると, 高分子の長時間緩和を定量的にほぼ正確に計算できる. 例として図 6.3 に周波数  $\omega$  の正弦波状のせん断変形を与え, 応答をとった結果を実験結果とともに示す.

図 6.3:  $G^*(\omega)$  の実験 [11] と計算の比較.

数百秒にいたる長時間側（低周波数側）まで再現できている。  
非線形粘弾性も同様に予測できる。

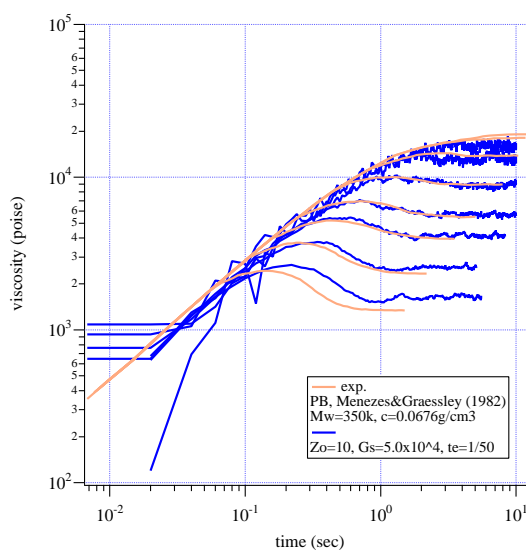


図 6.4: せん断粘度の実験との比較 [11]

図 6.4 にその例を示す。繰り返しになるが、実時間で数十秒に相当する現象が再現できている点に注目していただきたい。

## 6.3 分岐高分子の扱い

分岐高分子の場合には、からみあいネットワーク中にスリップリンクだけでなく、分岐高分子の分岐点にクロスリンクが存在することになるが、そのような場合にも式 6.1 の抵抗係数  $\alpha$  や力学平衡に寄与するチューブセグメント数の変更と、式 6.7 の境界条件の変更だけで容易に対応できる。分岐点においては式 6.1 は以下の

ように変更される．

$$\zeta_b \frac{d\mathbf{R}_i}{dt} = \frac{3kT}{n^2} \sum_j^q \frac{\mathbf{r}_j}{n_j} - kT \nabla q n_0 \mu + \mathbf{f}_i^b \quad (6.12)$$

$$\zeta_b = \frac{q}{4} \quad (6.13)$$

ここで  $q$  は分岐点から分岐している鎖の本数である．分岐点のランダム力  $\mathbf{f}_i^b$  は当然抵抗係数  $\zeta_b$  を反映したものになる．

本モデルで 3 本分岐の星形高分子の拡散を計算し，実験 [12] と比較した結果を図 6.5 に示す．

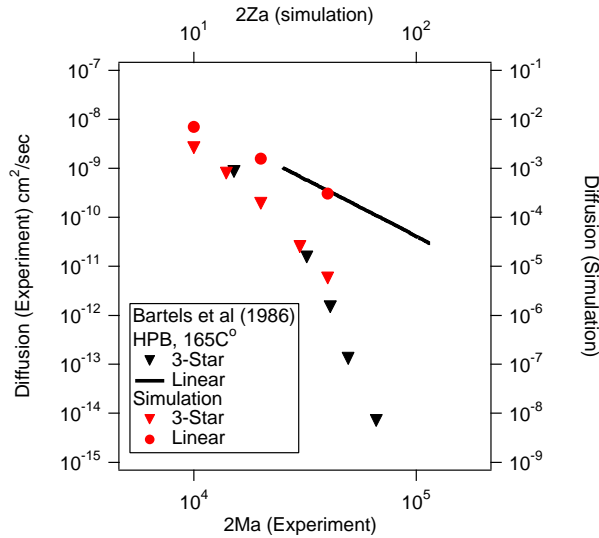


図 6.5: 星形高分子の拡散，実験 [12] との比較

直鎖高分子の拡散に比べて極端に遅くなること，分子量に対する依存性が直鎖に比べて大きくなることなど，実験結果を定量的に再現する事がわかる．

## 6.4 異種高分子ブレンドおよび共重合体への拡張

ここまでの議論は，系を構成する高分子材料を構成するモノマーが化学的に同じである場合についてのものである．前節までの拡張で，成形加工性を改善するために利用される分子量分布や分岐構造の調整に関する知見をえることができる．しかし材料の機能を高めるためには，異種の化学構造をもつモノマーからなる高分子どうしのブレンドが必要になることが多い．

からみあいを持つ高分子ブレンドのレオロジーを予測する手段はこれまで皆無と行って良い．構成方程式を用いる方法で現象論的にパラメーターを求めることはできるが，ブレンド比率や分子量を変えた場合に対する拡張は簡単でない．一方分子シミュレーションでは，Kremer-Grest のビーズスプリングモデルを用いたブレンドの計算がおこなわれてきている．しかし，からみあいの効果を十分に取り入れられる分子量の計算は事実上不可能である．

そこで primitive chain network モデルをブレンドに拡張し，からみあった高分子ブレンド系での分子ダイナミクスとレオロジーを計算できるようにする．式 6.1 および 6.7 は，化学種が増えることだけを考慮すれば良く本質的には何も変わらない．変わるのは自由エネルギー項であり，異なる化学種間の相互作用は平均場的にこのように取り扱う．

$$F = F_{mix} + F_{vol} \quad (6.14)$$

$$\frac{F_{mix}}{kT} = \chi\phi_\alpha\phi_\beta \quad (6.15)$$

$$(6.16)$$

ここで  $F_{mix}$  は Flory-Huggins の高分子ブレンドの混合の自由エネルギーのうち、エンタルピー項のみを考慮したものになっている。エントロピー項を考慮しない理由は、このモデルで考えているセグメントがからみあい点間分子量に相当する大きなものであるために、実質ほとんど無視できるからである。

図 6.6 に、ブレンド系で計算した相分離挙動を示す。からみあった高分子系で、相分離ダイナミクスを計算できる手法は他にない。

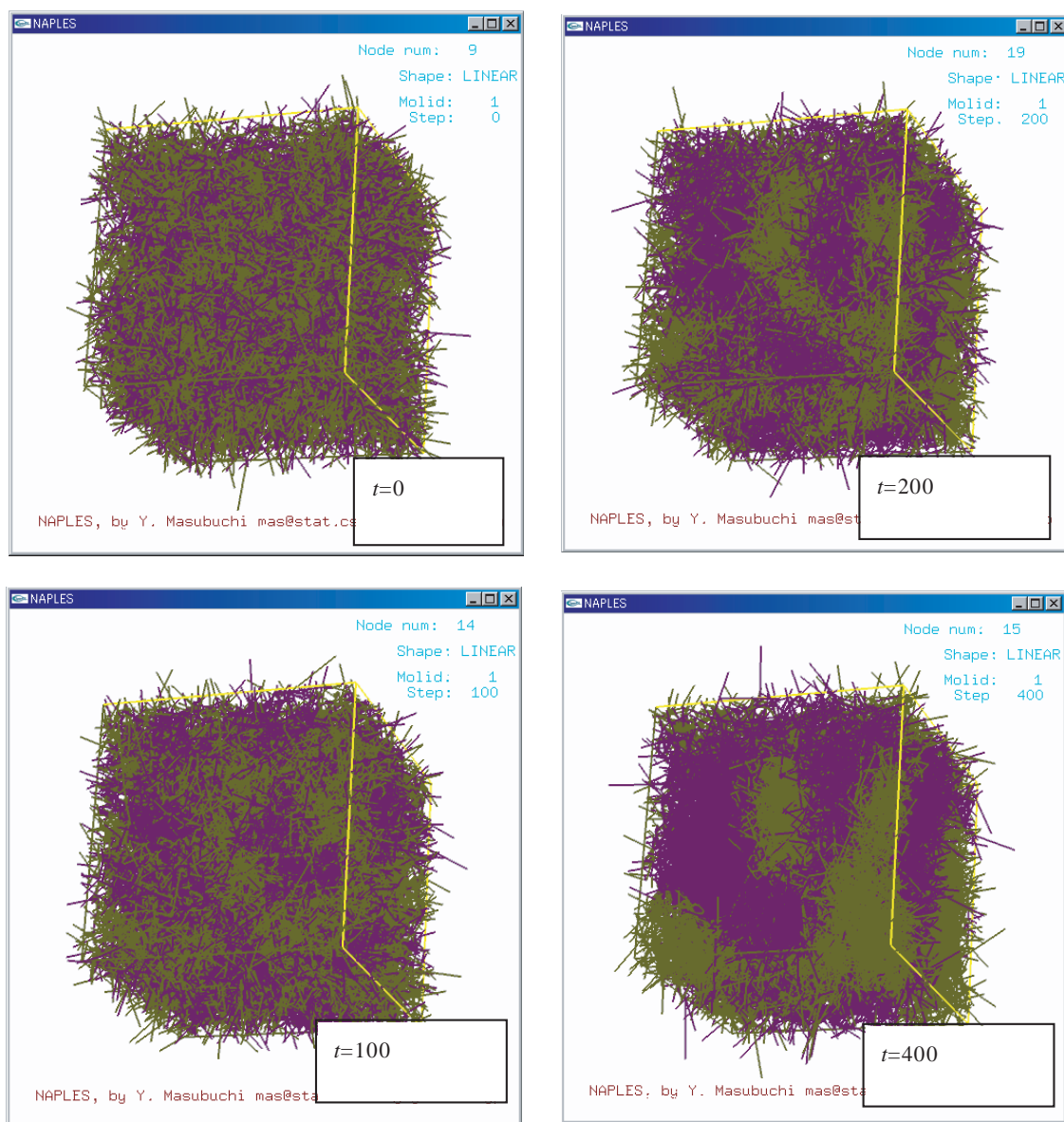


図 6.6: 高分子ブレンドの相分離ダイナミクス

現在のバージョンでは、共重合高分子のミクロ相分離も計算できるようになっている。これも密度汎関数理論による計算 [17] とは異なり、レオロジーや高分子のからみあいダイナミクスが計算できている。

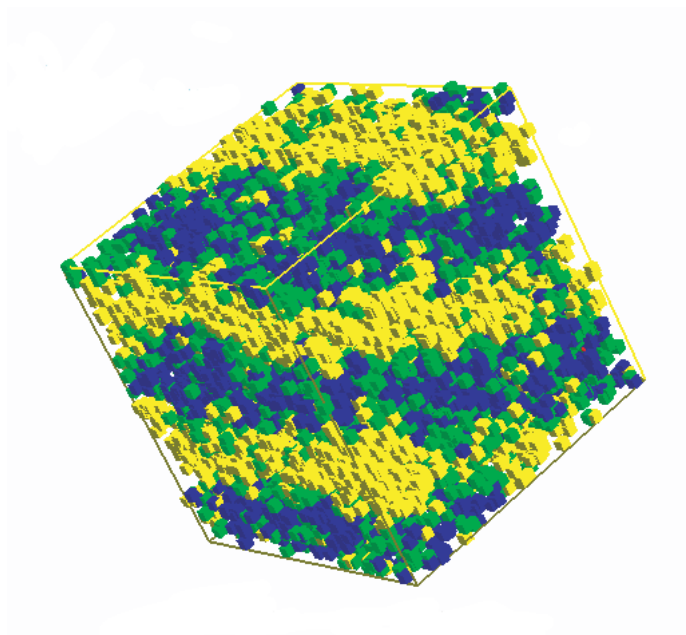


図 6.7: 共重合高分子のマイクロ相分離

## 6.5 入力パラメーターおよび出力物性単位

NAPLES における計算はすべて無次元化された量で計算されている．種々の量の単位を以下にまとめる．以

表 6.1: NAPLES の単位量

量	NAPLES 内の単位	文字と式
長さ	チューブセグメントの平均長さ	$a$
エネルギー	熱エネルギー	$kT$
時間	チューブセグメントあたりのモノマーの Rouse 緩和時間, あるいは一つのからみあい点の拡散時間	$\tau_e = \zeta a^2 / kT$

上の量があれば，応力や粘度も含む，あらゆる物理量を無次元化して表すことができる．例えば応力および弾性率の単位は  $kT/a^3$  であるし，粘度の単位は  $kT\tau_e/a^3$  である．従って物質固有のパラメーターとして必要なのは， $a$  と  $\zeta$  だけである．この 2 つの量は分子量や分子量分布，分岐構造などには依存しないと考えられる．（ただし温度や溶液の場合の濃度には依存する．）従ってある化学構造をもつ材料に対して  $a$  と  $\zeta$  を決めることができれば，その材料に関してはどのような分子量，分子量分布，分岐構造をもつ場合の計算も原理的には可能かつ即座に現実の値に変換することができる．

ただし現実的には， $a$  や  $\zeta$  を直接実験的に求めることは難しい．そこで間接的ながら実験的に決めやすい量によって NAPLES への入力パラメーターを決定し，NAPLES からの出力を現実にあった値に変換する．まず NAPLES には分子の形状を与えなければならないが，その際には分子量は通常値ではなく，平衡状態で長さが  $a$  になる分子量で割った値， $Z$  を入力しなければならない．平衡状態で長さが  $a$  になる分子量は実験的にはからみあい点間分子量  $M_e$  と呼ばれる値に対応している．詳しくは 4.5 を参照のこと．次に重要となるのが時間スケールである．どのくらいの時間計算を行えばよいか，あるいはどのくらいの速度で流動をかけるかを決めるには，単位時間  $\tau_e$  がわからなければならない． $\tau_e$  を決めるには，レオロジーの実験データを利用する．具体的な方法は 3.8.1 節に記載した．最後に，計算結果を現実の値に焼き直すには，応力のスケールが必要で

ある．これはレオロジー測定からデータベースが比較的充実しているプラトー弾性率 ( $G_0 = \rho kT/M_e$ ) を利用する．詳しくは 3.8 節を参照のこと．

## 6.6 計算対象となる物質/ならない物質

具体的に計算可能な物質の例を示す．

### 計算可能な物質の例

- 2 Me 以上の分子量をもつ，単分散の直鎖，星形，ポンポン，櫛型の各高分子メルト
- 上記の混合物（分子量分布を含む）
- 良溶媒に分散し，からみあい臨界濃度を超える高分子の溶液
- ランダム共重合体
- ブレンド
- ブロック共重合体
- ゴム，ゲル
- ...

NAPLES は，からみあった高分子のメルトを対象とする．具体的には，からみあい点間分子量 Me と呼ばれる，物質固有の分子量の 2 倍以上をもつ分子の運動が計算対象である．分岐ポリマーの場合には，分岐部分も Me 以上の分子量をもつ物のほうが原理的に計算精度がよいと考えられる．（検証が必要である）．

計算対象の高分子メルトは，以上のような条件を満たせば，どのような分岐構造，分子量分布，またその混合物でもよい．たとえば線状高分子で分子量分布をもつグレードに，微量の分岐高分子を混合したような場合の計算が可能である．現在 NAPLES で設定されている分岐構造は，線状，星状，ポンポン状，櫛状の 4 種類である．これらの分岐構造の詳細については後に述べる．またランダム分岐構造については次期以降のバージョンでの対応が予定されている．

以下に，計算不可能な系の例も示す．

### 計算不可能な系

- 分子量 Me 以下，または，からみあい臨界濃度以下で，分子がからんでいないメルトや溶液
- 液晶高分子（分子が剛直で Me が定義されない系）
- 高分子固体（結晶，）
- 繊維・粒子充填系
- 結晶性高分子で，結晶・非晶の共存する系
- ...

上にも述べたように，NAPLES はからみあった高分子のメルトを対象とするので，分子量が小さかったり，高分子の濃度が低かったりして，からみあわない状況にある物質の計算はできない．また同じ理由で，分子鎖が剛直でレオロジー的な意味でのからみあいをもたない液晶高分子の計算もできない．また NAPLES は高分子メルトを対象としているため，固体高分子，メルトでも結晶化部分を含む系，ガラス繊維や充填材を含む系には現在対応していない．これらは学問的にも難しい問題であり，検討を進めている．

## 6.7 計算が可能な物性/不可能な物性

現在 NAPLES で予測可能な長時間物性は高分子メルトのレオロジーである．分子レオロジーシミュレーターの名の由来はここにある．NAPLES では，せん断，一軸伸長，二軸伸長，平面伸長の 4 つの基本変形モードを全てあつかうことができる．これらの流動変形下で，せん断応力，法線応力を計算する．従って，平行平板型レオメーター，一軸伸長レオメーター，二軸・平面伸長レオメーター，の基本機能を有すると考えていただくと判りやすい．

具体的に予測可能なレオロジー特性は以下のようになる．

### 計算可能な物性

- 応力緩和（せん断，1 軸，2 軸，平面）
- せん断粘度（定常およびスタートアップ）
- 伸長粘度（1 軸，2 軸，平面）
- 粘弾性スペクトル（ $G'$   $G''$ ）
- 法線応力解析
- ...

ただし，NAPLES では応力しか計算しないため，応力を粘度や弾性率に変換する操作はユーザーが表計算ソフトウェアなどで行う必要がある．これらの方法や式については 3 節を参照のこと．

逆に，以下のようなことには対応していない．

### 計算不可能な物性と現象

- 高分子固体の物性
- 破壊
- 熱物性（粘度や弾性率の温度依存性を含む）
- 結晶化およびガラス転移
- ...

NAPLES は高分子メルトを対象としたシミュレーターであるため，固体の物性は考えていない．また，計算の特性上，メルトにおける破壊現象にも対応していない．計算の境界条件を変更することで対応できる可能性はある（固体はのぞく）．

また NAPLES では熱物性は扱うことができない．NAPLES は高分子メルトで一般的に了解されている，温度時間換算則に基づき計算を行う．温度時間換算則とは，系のもつ緩和時間分布が，温度を変化させても変わらないことを意味する基本原理である．（詳しくは参考文献 [18]）従って，結晶化やガラス転移のように，ある特定の温度をまたぐことによって系の構造が変わり，結果として緩和時間分布が変化するような系には適用できない．

粘度等の温度依存性についても，NAPLES では予測することができない．NAPLES はいわば系の緩和時間分布をもとめるシミュレーターであり，緩和時間の絶対値はフィッティングパラメーターである．したがって，緩和時間の絶対値の温度依存性に関係する粘度や弾性率の温度依存性は，原理的に求めることができない．



## 第7章 NAPLESのTIPS

本章では、NAPLES 利用者から寄せられた質問に対する回答などをまとめている。利用の一助となれば幸いである。

### 7.1 計算負荷を落として結果を早く出すには？

NAPLES は多数の分子の運動を計算する、いわゆる分子シミュレーションといわれる計算手法に属する。分子シミュレーションにおいて計算負荷を軽減させるには、単純に計算する分子数を減らせばよい。ただし、分子数を落とすと得られる結果の精度も落ちることに注意が必要である。以下に分子を減らすための具体的な方法を述べる。

- 計算セルのサイズを小さくする：

計算セルが小さければそこに入る分子の数も同じように小さくなる。入力パラメーターファイルにおいて `cell.size=` と指定することでセルのサイズを指定できる。セルのサイズを  $1/2$  にすると、計算負荷は  $1/2^3 = 1/8$  となる。ただしセルをあまり小さくすると周期境界条件がもつ性質により計算の正当性が保証されなくなる。詳しくは 4.7.2 節を参照。

- 濃度を減らす：

入力パラメーターファイルにおいて `C=` で指定される、からみあいセグメントの数密度を減らすことで分子の数は減る。デフォルトでは `C=10` が指定されているが、これを `C=5` などとすると単純に負荷は  $1/2$  になる。詳しくは 4.7.3 節を参照。

また推奨しないが、時間積分の数値刻みを大きくすることで計算時間を短縮することができる。詳しくは 4.7.4 節を参照のこと。

NAPLES は分子シミュレーションであるので、分子シミュレーションについて習熟している方は分子シミュレーションにおける計算負荷軽減法がそのまま成り立つと考えてよい。ただし、分子間相互作用をダイレクトに計算しないため、系に存在する粒子数  $N$  と計算負荷との関係は、分子シミュレーションで一般的な  $N^2$  とはならず、ほぼオーダー  $N$  である。従って計算負荷を減らす目的で分子数を減らしてもそれほど劇的な効果とはならない。

### 7.2 計算精度を上げるには？

NAPLES のように多数の分子の運動を計算する分子シミュレーションと呼ばれる手法では、一般に計算する分子の数を上げれば統計上精度が向上する。ただし計算にかかる分子の数を増やすと、結果として計算負荷もあがる。計算する分子の数を増やして統計量を稼ぐための具体的方法は以下である。

- 計算セルのサイズを大きくする：

計算セルが大きいと分子の数は当然大きくなる。入力パラメーターファイルで `cell.size=` と指定することでセルサイズを変更できる。セルのサイズを 2 倍にすると分子数は  $2^3 = 8$  倍になる。ただしセルを大きくしすぎると計算負荷が高くなり計算時間が非現実的になる。詳しくは 4.7.2 節を参照。

- 濃度を上げる：  
入力パラメーターファイルにおいて  $C=$  で指定される、からみあいセグメントの数密度を増やすことで分子の数は増える。デフォルトでは  $C=10$  が指定されているが、これを  $C=20$  などとすると分子の数は倍になり精度が向上する。詳しくは 4.7.3 節を参照。
- 異なる初期状態での計算を複数回行う：  
同じ条件の実験を数回行って平均をとるとデータの精度が向上するが、NAPLES でも同様のことができる。入力パラメーターファイルで `seed` で指定される乱数の種を変更すると、異なる初期状態から同じ条件（分子量や分岐、流動形態などが同じ）での計算を行うことになる。もちろんデータの傾向は同じになるが、値は少しずつ異なるため `seed` 値の異なる計算を複数回行ってデータの平均をとることで精度は向上できる。詳しくは 4.7.1 節を参照。
- 計算時間を増やす：  
3.2 節の線形緩和弾性率および 3.3 節の  $G'G''$  の精度を上げるための一つの手段として、平衡状態での計算時間（入力パラメーターファイルにおける `quit_dynamics_count`）を延ばすことがあげられる。線形緩和弾性率を得るために、時間自己相関関数を求めて時間平均をとっているためである。ただし  $G'G''$  はたいていの場合ログスケールでの挙動が問題となるので、精度を上げるには計算時間を 2 倍、4 倍、時には 10 倍としなければならない。これが非現実的である場合には、前項にある異なる初期状態での計算を積み重ねた方がよい。

なお計算時間については、分子数を 2 倍にすることで単純にほぼ 2 倍になると考えてよい。分子シミュレーションで分子間相互作用をあらわに計算する場合には計算の負荷は分子数の 2 乗で増加するが、NAPLES では分子間相互作用を平均場的に計算するため分子数の 1 乗にしかない。

### 7.3 OCTA に含まれる他のエンジンとの関係

OCTA はいくつかのプログラムの集合体であるが、その中で NAPLES と似たものに COGNAC と PASTA がある。

COGNAC は一見 NAPLES と非常に似通って見えるが、基本モデルが全く異なっている。まず COGNAC の計算モデルが 5 モノマー程度を 1 計算質点とするのに対して、NAPLES では百モノマー程度を 1 計算質点とするため、計算できる時間スケールに大きな隔たりがある。このため COGNAC では溶融体のレオロジーを計算することは事実上難しい。また NAPLES では分子間相互作用をあらわに計算しないため相互作用ポテンシャルが不要である。これは入力パラメーターが少ないという点では COGNAC より優れているが、逆に細かい分子構造の特徴を表すような計算には不向きであることを示している。具体的には結晶化や特異的相互作用を示す系の熱物性の計算は NAPLES では不可能であるが、COGNAC では可能である。

次に PASTA との比較について述べる。PASTA と NAPLES は独立に開発されたが基本的に同じ物理モデルから出発している。簡単にいって、PASTA の方が NAPLES より単純化されたモデルを採用している。そのため PASTA が計算できる直鎖高分子の系や星形高分子の系では NAPLES より計算が短時間で終わる。（予測精度比較は現在学術的に進行している段階）ただし単純化されたモデルゆえに制限は大きく、今のところ複雑な分岐形状を持つ系には拡張が予定されていない。またモデルの性格上、多数の高分子がからみあう様子をビジュアルに表現することができない。

### 7.4 NAPLES 情報ホームページ

ホームページアドレス：<http://masubuchi.jp/NAPLESweb/>

# References

- 1) Y. Masubuchi, G. Ianniruberto, F. Greco and G. Marrucci, J. Chem. Phys., **115**, 4387 (2001)
- 2) Y. Masubuchi, G. Ianniruberto, F. Greco and G. Marrucci, J. Chem. Phys., **119**, 6925 (2003)
- 3) Y. Masubuchi, G. Ianniruberto, F. Greco and G. Marrucci, Modelling Simulation Mat. Sci. Eng., **12**, S91-S100, (2004).
- 4) Y. Masubuchi, H. Watanabe, G. Ianniruberto, F. Greco and G. Marrucci, J. Soc. Rheol. Jpn., **32**(4), 197, (2004).
- 5) T. Yaoita, T. Isaki, Y. Masubuchi, H. Watanabe, G. Ianniruberto, F. Greco and G. Marrucci, J. Chem. Phys., **121**, 12650 (2004)
- 6) T. Yaoita, T. Isaki, Y. Masubuchi, H. Watanabe, G. Ianniruberto and G. Marrucci, Macromolecules, **45**, 2773 (2012)
- 7) T. C. B. McLeish and R. G. Larson, J. Rheol. **42** 81 (1998)
- 8) L. J. Fetters, D. J. Lohse, D. Richter, T. A. Witten and A. Zirkel, Macromolecules, **27** 4639 (1994)
- 9) J. D. Ferry, Viscoelastic Properties of Polymers 3rd Ed., Wiley, (1980)
- 10) 井上隆, 市原祥次, ポリマーアロイ, 高分子新素材 OnePoint-12, 高分子学会編集, 共立出版 (1988); L. A. Utracki, Polymer Alloys and Blends, Verlag, Munich (1989); 高性能ポリマーアロイ, 先端高分子材料シリーズ, 高分子学会編, 丸善 (1991); ポリマー ABC ハンドブック, 高分子学会高分子 ABC 研究会編, NTS (2001)
- 11) E. V. Menezes and W. W. Graessley, J. Polym. Sci. Polym. Phys. **20**, 1817 (1982).
- 12) C. R. Bartels, B. Crist and W. W. Graessley, Macromolecules **17**, 2702 (1984)
- 13) <http://octa.jp>
- 14) 青柳ら. <http://www.octa.jp/OCTA/cognac.html>
- 15) M. Doi and S. F. Edwards, The theory of polymer dynamics, Oxford, (1986)
- 16) たとえば J.-I. Takimoto, H. Tasaki and M. Doi, Polym. Preprints Jpn. **49** 2512 (2000), <http://www.octa.jp/OCTA/pasta.html>
- 17) 本田ら. <http://www.octa.jp/OCTA/sushi.html>
- 18) 日本レオロジー学会編, 講座・レオロジー, 高分子刊行会 (1995)