

論文 / 著書情報  
Article / Book Information

題目(和文)	大規模数値シミュレーションの高速化に関する研究
Title(English)	
著者(和文)	大西有希
Author(English)	YUKI ONISHI
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第6175号, 授与年月日:2005年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第6175号, Conferred date:2005/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

2004年度(平成16年度) 博士論文  
大規模数値シミュレーションの高速化に関する研究

指導教官 天谷 賢治 助教授  
情報理工学研究科 情報環境学(機械系)専攻  
大西 有希

2004年12月

# 目次

第1章 緒論	1
1.1 研究背景	1
1.2 従来研究の問題点	2
1.3 研究目的	3
参考文献	4
第2章 CPPLapack/CPPScaLapack : 大規模数値解析のための行列計算ライブラリ	5
2.1 緒言	5
2.2 CPPLapack/CPPScaLapack の特徴とその実装	5
2.2.1 高速行列計算	5
2.2.2 直感的なユーザーインターフェイス	6
2.2.3 並列プログラムへの簡単な移行	6
2.2.4 一時オブジェクトへの不必要なデータコピーの抑制	7
2.2.4.1 一時オブジェクト生成過程とその問題点	7
2.2.4.2 Smart Temporary システムの概要	7
2.2.4.3 Smart Temporary システムの諸注意点	10
2.3 コード例	10
2.3.1 連立一次方程式の直接解法	10
2.3.2 連立一次方程式の反復解法	12
2.4 計算速度についての検討	12
2.5 結言	13
参考文献	14

第3章	一様断面弾性体に対する動的有限要素解析の効率化	15
3.1	緒言	15
3.2	効率的な動的有限要素解析手法	16
3.2.1	要素分割	16
3.2.2	1スライス方程式	16
3.2.3	弾性体全体の方程式	17
3.2.4	方程式の解法	18
3.2.4.1	陰解法による定式化	18
3.2.4.2	漸化的解法	19
3.2.4.3	$[A']$ の収束特性	20
3.2.4.4	Dirichlet 境界を取扱う際の修正方法	20
3.2.4.5	$\{u\}$ の更新	21
3.3	解析例	21
3.3.1	片持ち梁の衝撃曲げ	21
3.3.2	側面を拘束した角棒の横衝撃	23
3.3.3	Hopkinson-bar の全体解析	24
3.4	実用例：一点曲げ衝撃試験の数値シミュレーションへの適用	27
3.4.1	緒言	27
3.4.2	静止き裂を持つ Timoshenko 梁モデル	27
3.4.2.1	基礎方程式	28
3.4.2.2	固有関数	29
3.4.2.3	外力に対する応答	30
3.4.2.4	動的応力拡大係数の簡便式	31
3.4.3	高速進展き裂を持つ Timoshenko 梁モデル	32
3.4.3.1	外力に対する応答	32
3.4.3.2	有限要素法および Euler-Bernoulli 梁モデルとの比較	33
3.4.4	一点曲げ衝撃試験のシミュレーション	35

3.4.4.1	衝撃力	35
3.4.4.2	一点曲げ衝撃試験の改良・拡張に対する検討	37
3.5	結言	39
	参考文献	40
<b>第4章</b>	<b>ポテンシャル問題に対する注目領域解の境界要素高速解法</b>	<b>42</b>
4.1	緒言	42
4.2	手法概要	42
4.3	高速解法	46
4.3.1	遠方境界の Fourier 級数展開	46
4.3.2	境界積分方程式の導出	47
4.3.2.1	境界積分方程式 1	47
4.3.2.2	境界積分方程式 2	50
4.3.3	離散化とマトリックス方程式	52
4.3.3.1	離散化された Fourier 級数展開	53
4.3.3.2	離散化された境界積分方程式 1	53
4.3.3.3	離散化された境界積分方程式 2	54
4.3.3.4	マトリックス方程式	56
4.3.4	注目領域解の計算	57
4.4	解析例	58
4.4.1	計算精度比較	58
4.4.1.1	ドーナツ領域での例	58
4.4.1.2	円形領域での例	59
4.4.1.3	花形領域での例	62
4.4.2	計算時間比較	67
4.5	結言	67
	参考文献	69
<b>第5章</b>	<b>結論</b>	<b>70</b>

# 第1章 緒論

## §1.1 研究背景

数値シミュレーションを用いた各種産業技術および社会技術の研究開発は、今日では日常的に行われるものとなっている。適用される分野も広く、機械、電気電子、化学、生物など多岐に渡っている。取り扱う解析対象が複雑になるにつれ、数値シミュレーションを用いなければ事実上解析不可能な事例も多く見受けられる。また、旧来は単に実験が困難な問題に対する模擬実験としての位置付けが色濃かった数値シミュレーションであるが、近年では予測、最適化、逆解析等の本質的に数値シミュレーションが必要な問題に対する適用も広く行われるようになってきている。この様な産業界および学術研究界を取り巻く環境を受け、数値シミュレーションはその重要度を増して来ている。

近年の計算機性能の飛躍的向上に伴って、一昔前には全く手に負えなかった程に大規模な解析も盛んに行われるようになってきている。パーソナルコンピュータの性能向上も目覚しく、一般の研究者にも手が届く価格のコンピュータで数万自由度の静解析ならば十分に実行可能な状況になっている。しかし、現状はこれで満足とは行かず、より大規模な解析に対する要望はむしろ天井知らずに加速しているように見受けられる。例えば、従来はシステムの一部のみの解析であったものをシステム全体の解析に、静解析であったものを動解析に、順解析で終わっていたものを最適化や逆解析に、単純なモデル化で済ませていたものを複雑なモデルに等々、より負荷の高い計算へと解析を発展させたい要望は後を断たない。

特筆すべきは、計算機性能の向上は数値シミュレーションの高速化に関する諸研究を無用にするものとはなっていないということである。上記の例の中には、計算機と数値シミュレーションの両方の技術向上があって初めて実現する事例が数多くある。大規模な計算を実現可能にするためには計算機と計算プログラム、言わばハードウェアとソフトウェアの両方の技術向上が不可欠な要素であると考えられる。故にソフトウェアを扱う数値シミュレーションの研究者は、高速な計算機による恩恵を積極的に享受しつつ、それだけでは十分に解決されない問題に対する高速解法の研究を進めて行くことが求められる。

また、数値シミュレーションの適用範囲の多様化により、個々の事例に特化した解析プログラム、言わば特注品を製作する必要がある場合が多くなっている。汎用解析ソフトで解析が不可能な問題だけでなく、汎用解析ソフトでも原理的には解析可能であるが計算時間や記憶容量がかかり過ぎる為に事実上解析が困難な問題に対しても、その問題に特化した解析プログラムを製作する必要がある。このように大規模な問題に対する高速なプログラムを自作しなければならない状況は今後も頻繁に現れることが予想され、大規模数値シミュレーションの高速化に関する諸研究は引き続き重要であると考えられる。

## §1.2 従来研究の問題点

本論文では、大規模数値シミュレーションに関する従来研究において未着手あるいは不完全であると思われる次の3点に着目した。

第1点目は、数値シミュレーションの高速化の研究では往々にして「計算時間」の短縮についてのみに主眼が置かれており、「開発時間」の短縮があまり重要視されていない点である。

先に述べた通り、数値解析のプログラム開発者は、解析事例毎に特化した個別のプログラムの製作を要求されることが少なくない。何らかの汎用解析パッケージを利用する場合と異なり、プログラムを自ら製作する場合の「解析時間」は「開発時間」と「計算時間」の和であり、真の意味での高速化を実現するためには「開発時間」の短縮は不可欠な要素である。特に、開発に際してモデル化やアルゴリズムの試行錯誤を繰り返す必要がある場合、「開発時間」は「計算時間」を遥かに上回る事もしばしば起こる。

ところで、数値計算による物理現象のシミュレーションの多くは最終的に数学的な行列計算に帰着される。従って、数値計算のプログラムを製作する際に行列を簡便に扱える事は「開発時間」を短縮する上で重要である。行列計算の「計算時間」の短縮方法としては、単一計算機用のライブラリである BLAS<sup>[1.1]</sup>と LAPACK<sup>[1.2]</sup>、ならびにその並列計算機用である PBLAS<sup>[1.4]</sup>と ScaLAPACK<sup>[1.3]</sup>が開発されており、広く用いられている。しかし、そのインターフェイスは複雑であり、行列を簡便に扱えるとは言い難い状況である。

第2点目は、物体の幾何学的特徴を利用した効率的数値解析手法の開発を目的とした研究が少ない点である。

境界値問題の数値解法の多くは任意の幾何学形状の物体に対して利用することを目的とした汎用的な手法である。代表的な境界値問題の数値解法である有限要素法を用いた汎用解析パッケージ<sup>[1.5]</sup>は数多くあるが、幾何学的特徴が高速化に活かされているのは2次元要素や軸対称要素を用いた解析<sup>[1.6]</sup>が可能であるところぐらいである。他にもコイルバネ等のらせん状の物体に特化した高速化手法<sup>[1.7]</sup>なども研究されているが、物体の幾何学的特徴を有限要素解析の高速化に活かす研究例は数少ない。

一方、構造物の多くは簡単な形状をした物体の組合せである。中でも棒状の物体もしくはその組み合わせによって形作られた構造物は数多くあり、しばしば数値解析の対象となっている。各種材料試験などでは棒状のシンプルな試験片が用いられることが多く、材料特性値の推定の為には大規模な繰り返し計算が必要となる場合もある。特に断面が一様である棒材はあらゆる構造物に用いられており、その弾性解析を有限要素法などにより高速に行うことは工学上重要である。しかし、現在までに断面が一様な棒材の弾性解析に対して、その幾何学的特徴を利用した効率的数値解析手法は開発されていない。

第3点目は、一般的な数値シミュレーションの解法が全ての未知数に対して同じ精度で解を求めることを暗黙の内に前提としてしまっている点である。

実際的な問題の解析要望には、全ての未知数の解を求めることを必要とせず、ある特定の未知数の解のみが必要である場合が少なくない。例えば境界値問題を対象とする場合、領域内の全て

の領域での解が必要である事はむしろ珍しく，ある注目領域の解だけが必要であるという状況が頻繁にある．この様な状況下では注目領域の解だけを高速に求められると都合が良い．しかし現在までに，注目する未知数の解だけを高速に求める，というコンセプトの研究は行われていない様に見受けられる．

境界値問題を境界要素法を用いて解く際の高速解法として，多重極展開法<sup>[1.8]</sup>が有名である．しかし，多重極展開法を用いた場合でも全ての未知数の解を同じ精度で求めることに変わりはない．境界積分法を基本解法として使い，注目領域の解だけを高速に求めることの出来る手法があれば有用であると考えられる．

### §1.3 研究目的

本論文は前述の3つの問題点を踏まえ，その中でもある程度広範囲の実際的な問題に対して適用可能であると考えられる3つの具体的な問題を克服する事を目的としている．本文はそれぞれの具体的研究テーマについて3つの章に分けて書かれている．

#### 第2章 CPPLapack/CPPScaLapack：大規模数値解析のための行列計算ライブラリ

大規模行列を簡便に取り扱え，なおかつ高速に計算出来るライブラリの開発が目的である．計算時間だけでなく開発時間も短縮することにより，全体としての解析時間の短縮を図ることを念頭に設計されている．開発言語にはオブジェクト指向言語であるC++<sup>[1.9]</sup>言語を用いている．本ライブラリには(1)高速行列計算，(2)直感的なユーザーインターフェイス，(3)並列プログラムへの簡単な移行，(4)一時オブジェクトへの不必要なデータコピーの抑制の4つの特徴があり，それらを順に解説する．特に特徴(4)は本ライブラリ独自の手法であり重点的に説明する．

#### 第3章 一様断面弾性体に対する動的有限要素解析の効率化

断面が一様である棒状の弾性体の動解析を陰解法の有限要素法で効率的に行う手法の開発が目的である！「形が棒状で断面が一様である」という幾何学的特徴を有限要素法で取り扱う行列の代数的特徴に変換して活用することにより，計算の高速化および記憶容量の低減を図っている．幾つかの解析例を示すと共に，実用例としてセラミック等の脆性材料の高温下での衝撃破壊靱性を測定する手法である一点曲げ衝撃試験に対して本手法を用いて行った数値シミュレーションについて詳しく解説する．

#### 第4章 ポテンシャル問題に対する注目領域解の境界要素高速解法

Laplace方程式で支配される2次元の境界値問題に対して，境界要素法を用いてある注目領域の解だけを高速に求める方法の開発が目的である．Laplace領域において，ある注目領域から十分な距離をおいた非注目領域の境界値の高周波成分の影響は注目領域の解に影響を与えないことを利用する．この性質を用いることにより，本手法ではすべての領域の解を同精度に求めることをせず，注目領域の解を高い精度で，非注目領域の解を低い精度で求めている．手法の解説および幾つかの解析例を示し，手法の有効性を確認する．

各研究テーマ毎の背景および研究目的の詳細は，各章の緒言に記載している．

## 参考文献

- [ 1.1 ] C. Lawson et al., “Basic Linear Algebra Subprograms for FORTRAN usage”, *ACM Trans. Math. Soft.*, 5(3), pp.308–323, 1979
- [ 1.2 ] E. Anderson et al., “LAPACK 利用の手引”, (小国力訳), 丸善, 1995
- [ 1.3 ] L. S. Blackford et al., “ScaLAPACK User’s Guide”, *SIAM*, 1997
- [ 1.4 ] J. Choi et al., “A Proposal for a Set of Parallel Basic Linear Algebra Subprograms”, *UT Tech Report CS-95-292*, LAPACK Working Note 100, 1995
- [ 1.5 ] Hibbitt, Karlsson, Sorensen, “ABAQUS/Standard Users’ Manual Version 6.3”, *HKS Inc.*, 2002;
- [ 1.6 ] 鷲津久一郎ほか, “有限要素法ハンドブック”, 培風館, 1981
- [ 1.7 ] Jiang WG, Henshall JL, “A novel finite element model for helical springs” *FINITE ELEMENTS IN ANALYSIS AND DESIGN*, 35 (4), pp.363–377, 2000
- [ 1.8 ] Greengard L, Rokhlin V, “A Fast Algorithm for Particle Simulations”, *Journal of Computational Physics*, 73, (1987), pp.325–348
- [ 1.9 ] B. Stroustrup, “プログラミング言語 C++ 第 3 版”, (株式会社ロングテール, 長尾高弘訳), アスキー, 1998

## 第 2 章 CPPLapack/ CPPScaLapack : 大規模数値解析のための行列計算ライブラリ

大西有希, 居相政史, “CPPLapack/ CPPScaLapack : 大規模数値解析のための行列計算ライブラリ”, 第 9 回計算工学講演会論文集, No.2, (2004), pp.663–666 より出典

### §2.1 緒言

数値解析による物理現象のシミュレーションの多くは最終的に数学的な行列計算の問題に帰着される。特に大規模な数値解析の場合, または最適化や逆解析など繰り返し計算を必要とする場合, 行列計算の効率は数値解析の効率そのものを左右する重要な要素となる。従って高性能な行列計算ライブラリの開発は数値解析技術の発展に不可欠である。

行列計算ライブラリが高性能である為には以下の 4 つの特徴を満たすことが必要であると考えられる。第 1 に計算が高速であること, 第 2 に直感的なコード記述が可能であること, 第 3 にソースコードをほとんど変更せずに単一計算機と並列計算機の両方の環境で動作すること, 第 4 に一時オブジェクトへの不必要なデータコピーを無くすこと, 以上 4 つである。既存の行列計算ライブラリには Blitz++<sup>[2.1]</sup>, Matrix Template Library(MTL)<sup>[2.2]</sup>, LAPACK++<sup>[2.3]</sup>, Template Numerical Toolkit(TNT)<sup>[2.4]</sup>, Java Matrix Package(JAMA) などがあるが, いずれも上記 4 つの特徴全てを満たしていない。特に, 第 4 の一時オブジェクトに関する問題は処理時間および記憶容量を不必要に消費する大きな問題である<sup>[2.11]</sup>が, 既存の行列計算ライブラリでは解決されていない様に見受けられる。

本研究では, 汎用的なプログラミング言語である C++言語を用いて上記 4 つの特徴を満たす行列計算ライブラリ CPPLapack/ CPPScaLapack を開発した。本稿では CPPLapack/ CPPScaLapack の実装について上記の特徴ごとに説明する。また, CPPLapack/ CPPScaLapack を用いた簡単なコード例, ならびにその有効性を示す。

### §2.2 CPPLapack/ CPPScaLapack の特徴とその実装

#### §2.2.1 高速行列計算

行列とベクトルの高速演算および計算用サブルーチンとして Basic Linear Algebra Subprograms (BLAS)<sup>[2.6]</sup> および Linear Algebra Package (LAPACK)<sup>[2.7]</sup> が有名である。スカラー並列環境で動作するサブルーチンとして Parallel Basic Linear Algebra Subprograms (PBLAS)<sup>[2.9]</sup> および Scalar

Linear Algebra Package (ScaLAPACK)<sup>[2.8]</sup>も開発されている。これらのパッケージは数多くのプラットフォームで最適化されたものが提供されており、汎用性が高い。

CPPLapack/PPScaLapackは行列およびベクトルの演算にBLAS/PBLASを、連立一次方程式の求解等の行列計算にLAPACK/ScaLAPACKを内部から呼び出して用いることにより高速化が図られている。

### §2.2.2 直感的なユーザーインターフェイス

ソースコードを直感的に記述できることは開発時間を短縮しバグの発生を低減する上で都合が良い。CPPLapack/PPScaLapackはC++言語を用いたクラスライブラリとして設計されている為、オブジェクト指向に基づいた直感的なコード記述が可能である<sup>[2.10]</sup>。具体的には密行列、帯行列、対称行列などに対し別々のクラスが用意されており、各種演算は $+$ ,  $-$ ,  $*$ ,  $/$ などの演算子を用いて一般的な数式と同じ表記で記述できる。行列やベクトルのクラスは内部にサイズの情報を保持しているので、BLAS/PBLASおよびLAPACK/ScaLAPACKを直接用いる場合に必要な数多くの引数を省いた形で使用できる。簡単なコード例を次節に記載しているので参照されたい。

### §2.2.3 並列プログラムへの簡単な移行

LAPACK等を利用することにより大規模行列計算を高速に行えるようになるが、さらなる大規模化と高速化のためにはScaLAPACK等を用いたスカラー並列化が必要不可欠である。しかし、従来の行列計算ライブラリの多くは単一計算機のための計算コードとスカラー並列計算機のための計算コードに互換性が無く、別々に開発する必要があった。

CPPLapack/PPScaLapackでは、単一計算機とスカラー並列計算機でのコードの違いをクラスの内部で吸収し、両方の環境で同じユーザーインターフェイスを提供している。すなわち、同じ名前の同じ引数を取る関数が、CPPLapackでは単一計算機による処理に適したコードで、PPScaLapackではスカラー並列計算機による処理に適したコードでそれぞれ書かれている。従って、次節に示す極めて簡単な手続きだけで、単一計算機のための計算コードからスカラー並列計算機のための計算コードへと移行することができる。この高い互換性により、計算コードの開発効率が大幅に改善されることが期待できる。

なお、ベクトル計算機や共有メモリ型並列計算機による並列化はそのプラットフォーム用に最適化されたBLASおよびLAPACKを用いることにより可能となる場合がある。その場合はCPPLapackを用い、適切なオプションを付けてコンパイルすればよい。

## §2.2.4 一時オブジェクトへの不必要なデータコピーの抑制

### §2.2.4.1 一時オブジェクト生成過程とその問題点

C++で行列計算ライブラリを作る際、必ず問題となるのが一時オブジェクトに関わる問題である<sup>[2.11]</sup>。計算式をC++で処理する場合、各演算子関数が内部で作成した計算結果は局所オブジェクトに蓄えられ、その後return文を通じて一時オブジェクトにデータがコピーされる。2項以上の式を処理する場合、各部分式が生成した一時オブジェクトは原則として全体式を処理し終えるまで消滅しない。一時オブジェクトへのデータコピーは本来不必要な処理であり、また、消滅せずに残り続ける一時オブジェクトはメモリ使用量を増大させる。特に大規模な計算の場合、計算速度低下やメモリ不足を引き起こしかねない重要な問題となる。

### §2.2.4.2 Smart Temporary システムの概要

この問題を解決するため、CPPLapack/CPPScaLapackでは、本研究で開発したSmart Temporary システムを利用している。ユーザーが直接利用する行列クラスの他に、それと機能的には等価だが関数の返値のみに利用するクラス、Smart Temporary クラスを用意し、引数として渡されたオブジェクトが一時オブジェクトか否かを判断できるようにする仕組みである。

Smart Temporary クラスは、ユーザーが直接利用する行列クラスと異なる次の2つの性質をもつ。デストラクタが呼ばれてもメモリを解放しないことと、コピーコンストラクタがコピー元からコピー先へポインタを移す浅いコピーを行うこと、である。これらの性質は、計算結果を関数の返値として返す際に効果を発揮する。さらに、引数として渡されたオブジェクトが一時オブジェクトか否かを関数内で判断できることは、加算、減算、スカラー倍の演算においてメモリの有効利用を図る効果がある。

具体例として、4つの行列オブジェクトが関連する $Y = AB + C$ という計算を考える。通常の行列計算ライブラリを用いた場合のメモリ操作の流れ図を図2.1に示す。積演算では $L$ 、加算演算では $M$ 、と演算子関数ごとに新しい局所オブジェクトを生成しメモリを確保している。それぞれの演算子関数を抜けると返値の行列を保持するために一時オブジェクト $P, Q$ が生成されるが、この一時オブジェクトも新しくメモリを確保する。この時、局所オブジェクトのメモリの値を一時オブジェクトのメモリ領域へとコピーする深いコピーが行われる。また、代入演算においても一時オブジェクト $Q$ からユーザーが宣言したオブジェクト $Y$ へ深いコピーが行われている。つまり、この計算において合計で3回の深いコピーが行われていることになる。一時オブジェクトが所有するメモリ領域は一連の演算が終了するまで解放されずに残るので、結果的に瞬間最大で行列3個分のメモリが余分に必要になっている。

次に、CPPLapack/CPPScaLapackを用いた場合のメモリ操作の流れ図を図2.2に示す。積演算においては、計算結果を保存するために局所オブジェクト $L$ を生成し、新しくメモリを確保する。演算子関数を抜けた時点で一時オブジェクト $P$ が生成されることは先ほどの例と同様だが、Smart Temporary システムでは浅いコピー、すなわちポインタの受け渡しのみが行われる。加算演算では引数の一方である $P$ がSmart Temporary オブジェクトとなるので、 $P$ に対して $+=$ 演算を行い

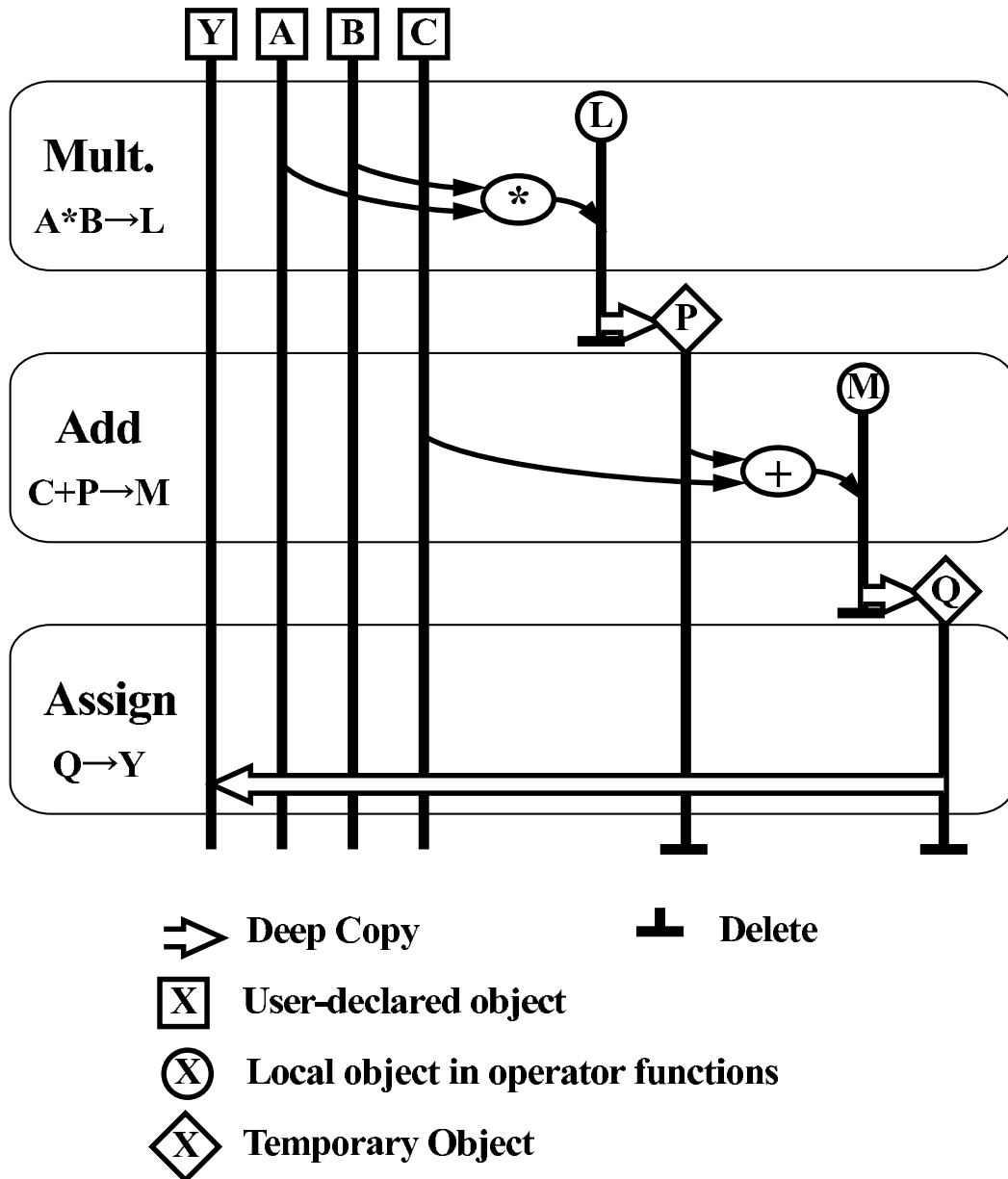


Fig. 2.1: Memory Operation Flowchart during  $Y = AB + C$  with ordinary library

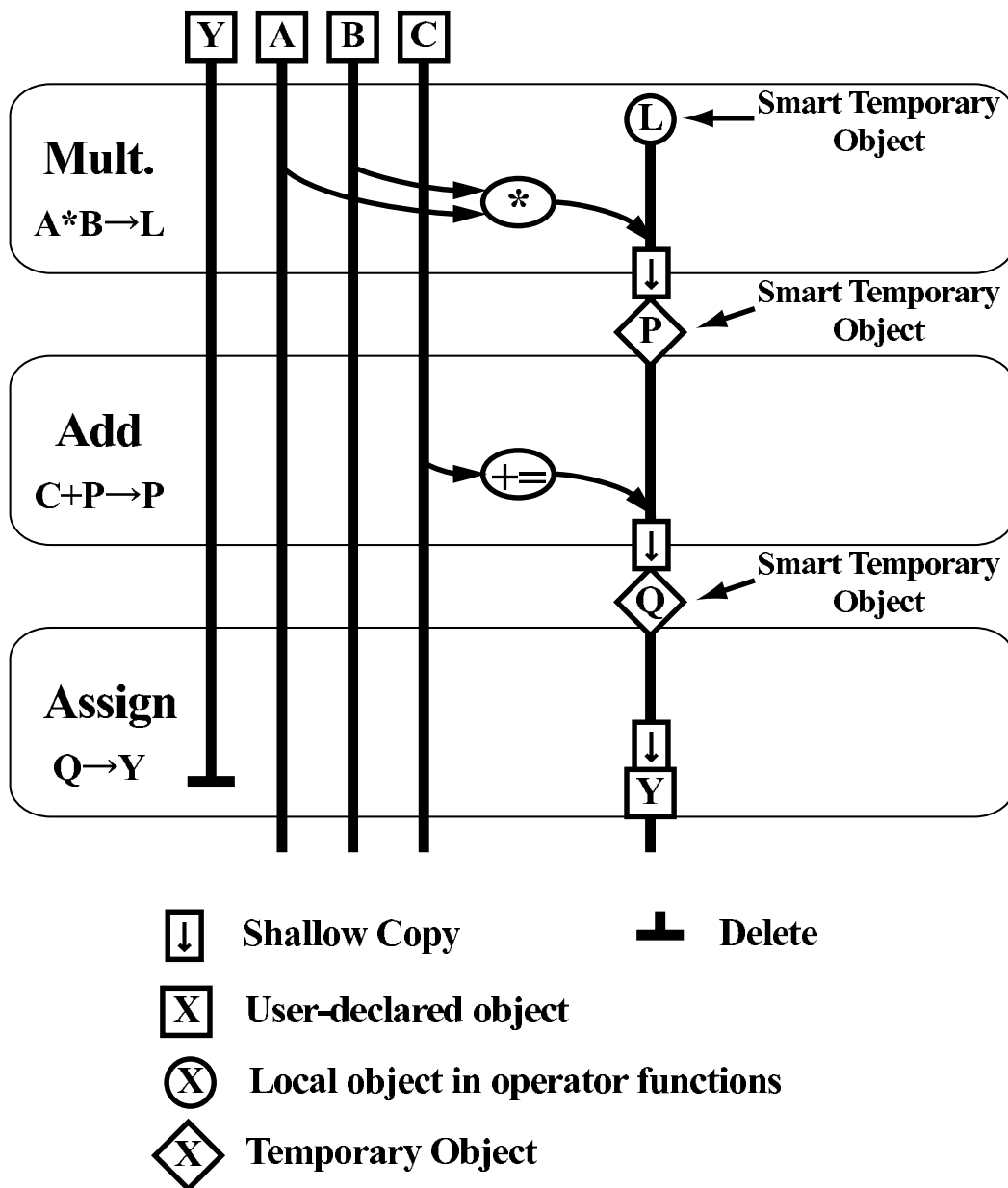


Fig. 2.2: Memory Operation Flowchart during  $Y = AB + C$  with CPPLapack/PPScaLapack

計算結果を上書きしている．これにより，新たに局所オブジェクトを生成することも新たにメモリを確保することも必要がない．代入演算においても，浅いコピーのみで処理が完了している．最終的に， $Y$  は積演算子関数の局所オブジェクト  $L$  のために確保されたメモリ領域に関連づけられる．Smart Temporary システムでは不必要な深いコピーは一度も行われず，またこの例では瞬間最大でも余分に必要なメモリは行列 1 個分のみで済んでいる．

もし，関数の返値として Smart Temporary クラスでなくユーザーが直接利用する行列クラスを用い，一時オブジェクトにポインタのみを渡す浅いコピーを行うと不都合が生じる．先述の例で  $L$  が Smart Temporary クラスでないとして浅いコピーが行われたとすると，積演算子を抜けた後  $L$  に対してデストラクタが呼び出された時点で  $L$  が参照しているメモリ領域が解放され， $P$  に渡されたポインタは無効（ダングリングポインタ）となってしまう．Smart Temporary システムを利用することは，この不都合を避けるための方法のひとつである．

### §2.2.4.3 Smart Temporary システムの諸注意点

Smart Temporary クラスのオブジェクトはデストラクタによってメモリが解放されることはないが，演算中に必要がなくなった時点で演算子関数内でメモリの解放が行われる．例えば加算演算の両方の引数が Smart Temporary クラスのオブジェクトであった場合，どちらか一方が計算結果として返され，もう一方は演算子関数内でメモリの解放が行われる．

Smart Temporary システムでは，計算式の最終的な計算結果となる Smart Temporary オブジェクトがユーザーの宣言した行列オブジェクトに代入演算子等で関連づけられることを前提に設計されている．逆に言えば，行列計算を行っても結果をユーザーが宣言した行列オブジェクトに代入しないまま放置するという意味のないコードを書くと，メモリが解放されずに残ってしまう．本ライブラリを利用する上での唯一の注意点である．

なお，高度に最適化されたコンパイラを使用して名前付き返値最適化<sup>[2,12]</sup>などの技法を利用すれば不必要な深いコピーをある程度抑制することは可能である．しかし，この方法では先に示した加算，減算を  $+=$  演算， $-=$  演算のみで済ますといった深いコピーの完全な抑制には到らない．CPPLapack/CPPScaLapack を利用すればコンパイラに依存すること無く深いコピーを完全に抑制出来る．

## §2.3 コード例

### §2.3.1 連立一次方程式の直接解法

CPPLapack を用いて連立一次方程式  $Ax = y$  を直接法を用いて解く為のコード例を次に示す．行列  $A$  およびベクトル  $y$  のデータをコンストラクタでファイルから読み込み，直接解法で解いた後，解をファイルに書き込んでいる．

```
#include "cpplapack.h"

int main(int argc, char** argv)
{
    CPPL::dgematrix A("matrix_data.txt"); //一般密行列の生成
    CPPL::dcovector y("vector_data.txt"); //縦ベクトルの生成

    A.dgesv(y); //LAPACK 関数 dgesv による直接解法
    y.write("solution.txt"); //結果出力

    return 0;
}
```

ここで CPPL は CPPLapack の名前空間であり，`dgematrix` および `dcovector` はそれぞれ実数の一般密行列および縦ベクトルのクラスである．

次に，CPPScaLapack を用いて同じ処理を行う為のコード例を次に示す．

```
#include "cppscalapack.h"
#include "cppscalapack_init.h"

int main(int argc, char** argv)
{
    CPPSL::initialize(argc,argv, 2,4, 64,64); //並列開始のための初期化処理

    CPPSL::dgematrix A("matrix_data.txt"); //一般密行列の生成
    CPPSL::dcovector y("vector_data.txt"); //縦ベクトルの生成

    A.dgesv(y); //ScaLAPACK 関数 pdgesv による直接解法
    y.write("solution.txt"); //結果出力

    CPPSL::finalize(); //並列終了処理
    return 0;
}
```

ここで CPPSL は CPPScaLapack の名前空間である．`initialize` 関数の後ろ 4 つの引数はそれぞれ，プロセス格子の行数と列数，データブロックの行数と列数を意味しており，環境に合わせて適当な値に定める必要がある．

CPPLapack/CPPScaLapack いずれの場合でも計算には LAPACK または ScaLAPACK が呼び出されるので，計算は高速である．CPPScaLapack の並列コードでも実質 10 行足らずで済んでおり，また直感的に理解しやすい．並列プログラムへの移行に必要な手続きは，

- include するファイルを追加，変更する
- main 関数の先頭と末尾にそれぞれ `initialize` 関数と `finalize` 関数を書き加える
- 名前空間 “CPPL” を “CPPSL” にする

の 3 つの単純な作業だけであり，スクリプト処理による自動化も可能である．

### §2.3.2 連立一次方程式の反復解法

特に大規模な連立一次方程式の解法には反復解法がよく用いられる．ここでは代表的な反復解法である Bi-CGStab 法<sup>[2,13]</sup>を行う関数のコード例を示す．

```
bool bicgstab(const CPPL::dgematrix& A, CPPL::dcovector& x,
             const CPPL::dcovector& y, const double& eps,
             int maxit)
{
    double alpha(0.0), beta,
           omega(1.0), rho_1(1.0), rho_2, eps_r(x.l*eps);
    CPPL::dcovector R(y-A*x), r(R), s, p(x.l), v(x.l), t;
    p.zero(); v.zero(); //ゼロベクトルに設定

    while(nrm2(r)>eps_r && (maxit--)>0){
        rho_2 = R*r;
        beta = (alpha/omega)*(rho_2/rho_1);
        p = r+beta*(p-omega*v);
        v = A*p;
        alpha = rho_2/(R*v);
        s = r-alpha*v;
        t = A*s;
        omega = (t*s)/(t*t);
        x += alpha*p +omega*s;
        r = s-omega*t;
        rho_1 = rho_2;
    }

    if(nrm2(r)>eps_r){ return 1; }
    else{ return 0; }
}
```

ここで演算子%は内積を意味する．

この例では LAPACK の関数は使われていないが，一般的な数式と同様の表記で記述された計算式の各種演算で BLAS 関数が使われている．なお，このコードを PBLAS を用いた並列化可能なコードに書き換えるには，名前空間を CPPL から CPPSL に書き換えるだけで良い．

## §2.4 計算速度についての検討

CPPLapack/CPPScaLapack は BLAS/PBLAS および LAPACK/ScaLAPACK のラッパーであり，行列計算の計算速度は LAPACK/ScaLAPACK を直接利用した場合と同等と考えてよい．ただし，演算子を用いた CPPLapack/CPPScaLapack の計算は BLAS/PBLAS を直接利用した場合に比べると厳密には計算速度が低下する場合がある．

例えば CPPLapack の実行列と実ベクトルの積の演算子関数の内部では BLAS の dgemv 関数を

呼び出しているが，`dgemv` 関数自体は  $y = \alpha Ax + \beta y$  の計算を行う関数であるので，CPPLapackの演算子関数では  $\alpha = 1, \beta = 0$  として  $Ax$  の計算を行っている．そのため， $Ax + y$  という計算を行う場合を考えると，`dgemv` 関数を直接用いて1回で計算を済ませる方がCPPLapack/CPPScaLapackを利用するより速いことになる．しかし， $Ax$  の積演算が  $O(N^2)$  の計算量であるのに対しベクトルの加算演算や定数倍演算は  $O(N)$  の計算量であり，大規模行列を対象とする際には計算速度にほとんど影響を及ぼさない．

また，BLAS/PBLAS および LAPACK/ScaLAPACK の関数を直接呼び出す必要がある場合のために，本ライブラリは行列およびベクトルのデータ配列のポインタを取得しやすく設計してあるので，ユーザーは容易にこれらの関数を直接利用することが出来る．

## §2.5 結言

行列計算ライブラリ CPPLapack/CPPScaLapack を開発した．本ライブラリは次の4つの特徴を持つ．(1)BLAS/PBLAS および LAPACK/ScaLAPACK を用いた高速行列計算，(2)C++言語の特徴を活かした直感的なユーザーインターフェイス，(3)統一されたクラス仕様による並列プログラムへの簡単な移行，(4)Smart Temporaryシステムを用いた一時オブジェクトへの不必要なデータコピーの抑制，以上4つである．本稿では4つの特徴について，その実装を含めて説明を行った．

コード例として同じ処理を行う単一計算機用コードと並列計算機用コードの両方を示し，単一計算機用から並列計算機用への移行が3つの単純な作業だけでよいことを述べた．また，速度についての検討も行い，充分大きな行列計算に対して本ライブラリを用いた場合とBLAS/PBLAS および LAPACK/ScaLAPACK の関数を直接用いた場合とで，計算速度にほとんど差位が無くなることにも言及した．

近年の数値解析の適応分野は多岐にわたる為，汎用数値計算パッケージでは解析できない問題も多く，プログラムを自作せざるを得ない場合が増えている．プログラムを自作する多くの場合，開発時間と計算時間の和を最小にすることが求められる．行列計算は種々のシミュレーションで必要であり，本ライブラリを用いることにより，高速かつ並列化が容易なプログラムを素早く開発することが可能となる．また，本ライブラリは汎用的なプログラミング言語であるC++言語で書かれているため，対応するプラットフォームが多く，行列計算以外の各種処理を行うコードと容易に一体化することも出来る．

なお，CPPLapack/CPPScaLapack は現在下記 URL にて開発を行っている．ソースコードおよびドキュメントが公開されているので参照して頂きたい．

<http://cpplapack.sourceforge.net/>

<http://cppscalapack.sourceforge.net/>

## 参考文献

- [ 2.1 ] Veldhuizen TL, “Arrays in Blitz++”, *Proceedings of the 2nd International Scientific Computing in Object Oriented Parallel Environments (ISCOPE'98)*, 1998
- [ 2.2 ] Siek JG, Lumsdaine A, “The Matrix Template Library: Generic components for high-performance scientific computing”, *Computing in Science and Engineering*, 1(6), 1999
- [ 2.3 ] Dongarra J, Pozo R, Walker D, “LAPACK++: A Design Overview of Object-Oriented Extensions for High Performance Linear Algebra”, *Proceedings of Supercomputing*, pp.162–171, 1993
- [ 2.4 ] <http://math.nist.gov/tnt/>
- [ 2.5 ] <http://math.nist.gov/javanumerics/jama/>
- [ 2.6 ] C. Lawson et al., “Basic Linear Algebra Subprograms for FORTRAN usage”, *ACM Trans. Math. Soft.*, 5(3), pp.308–323, 1979
- [ 2.7 ] E. Anderson et al., “LAPACK 利用の手引”, (小国力訳), 丸善, 1995
- [ 2.8 ] L. S. Blackford et al., “ScaLAPACK User’s Guide”, SIAM, 1997
- [ 2.9 ] J. Choi et al., “A Proposal for a Set of Parallel Basic Linear Algebra Subprograms”, *UT Tech Report CS-95-292*, LAPACK Working Note 100, 1995
- [ 2.10 ] W. H. Press et al, “Numerical Recipes in C++”, *Cambridge University Press*, 2002
- [ 2.11 ] B. Stroustrup, “プログラミング言語 C++ 第 3 版”, (株式会社ロングテール , 長尾高弘訳), アスキー, 1998
- [ 2.12 ] S. B. Lippman, “C++オブジェクトモデル 内部メカニズムの詳細”, (三橋二彩子ほか訳) トッパン, 1997
- [ 2.13 ] H. A. Van Der Vorst, “Bi-CGStab: A Fast and Smoothly Converging variant of Bi-CG for the Solution of Nonsymmetric Linear Systems”, *SIAM J. Sci. Stat. Comput.*, 13(2), pp.631–644, 1992

## 第3章 一様断面弾性体に対する動的有限要素解析の効率化

大西有希, 青木繁, 天谷賢治, 浦郷正隆, “一様断面弾性体に対する動的有限要素解析の効率化”, 日本機械学会論文集A編, v67, n662, 2001, pp.1583–1588,

青木 繁, 天谷賢治, 大西有希, 水谷隆之, “一点曲げ衝撃試験の数値シミュレーション”, 日本機械学会論文集A編, v68, n670, 2002, pp.924–929,

Onishi Y, Urago M, Aoki S, Amaya K, “Efficient Dynamic Finite Element Method for 3D Objects with Uniform Cross-Sections”, *International Journal for Numerical Methods in Engineering*, (掲載決定) より出典

### §3.1 緒言

衝撃荷重を受ける機械や構造物の安全性評価のためばかりでなく, 応力の伝播挙動を利用した各種の逆解析<sup>[3.23][3.5]</sup>のために, 一様断面を持つ棒(梁)や板の動的解析がしばしば必要となる. この解析に有限要素法を適用する場合, 特に最適化問題や逆問題では繰り返し計算が必要となることが多く, 効率的な方法が望まれる.

動的有限要素解析では, 質量マトリックスと減衰マトリックスが共に対角マトリックスであれば陽解法の中心差分法を用いて計算でき, 逆行列を求める必要が無くなる. 陽解法は計算速度, 記憶容量の点で優れている反面, 安定して解が得られない, 精度が良くない等の問題点があるため, 計算時間が多くても陰解法を用いることが望ましいとされる.

有限要素解析で細長い物体を扱う際, 上手に離散化することにより剛性マトリックス等がブロック3重対角となる特徴を持つことは良く知られており, そのことを考慮に入れた効率的手法<sup>[3.24][3.25][3.26]</sup>は幾つか提案されている. しかし, 断面が一様な細長い物体に対するより効率的な手法については研究がなされていない.

著者らは以前, 断面(形状は任意)が母線方向に一様であるような弾性体の静的有限要素解析に対する効率化手法<sup>[3.27]</sup>を開発した. 本研究ではこの方法を陰解法を用いた動的有限要素解析に拡張し, 計算時間, 記憶容量の節約を図る.

## §3.2 効率的な動的有限要素解析手法

弾性体の動解析の基礎方程式は次で与えられる．<sup>[3.3]</sup> 簡単のため，体積力の項を省略して議論する．

$$\rho \ddot{u}_i = \mu \nabla^2 u_i + (\lambda + \mu) u_{k,ki} \quad (\text{Navier's equation}), \quad (3.1)$$

$$u_i = g_i \quad (\text{on Dirichlet boundary}), \quad (3.2)$$

$$\sigma_{ij} n_j = h_i \quad (\text{on Neumann boundary}), \quad (3.3)$$

ここで  $\rho$  は密度， $\mu, \lambda$  は Lamé 定数， $\sigma$  は応力テンソル， $u$  は変位ベクトル， $n$  は法線ベクトル， $g$  は変位境界における変位ベクトル， $h$  は荷重境界における荷重ベクトルを表す．

### §3.2.1 要素分割

図 3.1 に示すような断面 (形状は任意) が母線方向に一様であるような弾性体を考える．この弾性

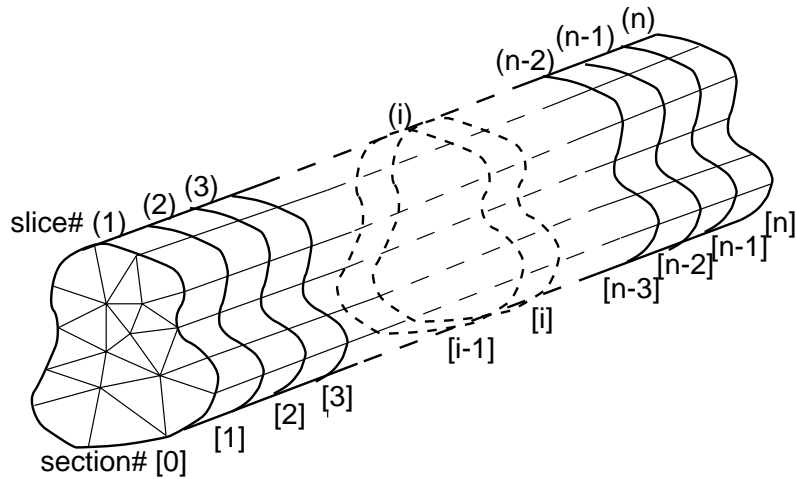


Fig. 3.1: Discretization of object with uniform cross-section

体を図 3.1 のように母線方向に  $n$  個のスライスに等分し，各スライスを同形のメッシュで要素分割する．これにより，全ての節点は端面を含む  $n + 1$  個の断面上にのみ一様に存在することになる．

### §3.2.2 1スライスの方程式

ある1つのスライスのみを考える．このスライスに対する変位型の有限要素式は次式で与えられる．

$$[M^s] \{\ddot{u}^s\} + [C^s] \{\dot{u}^s\} + [K^s] \{u^s\} = \{f^s\}. \quad (3.4)$$

ここで， $[M]$  は質量マトリックス， $[C]$  は減衰マトリックス， $[K]$  は剛性マトリックス， $\{u\}$  は節点変位ベクトル， $\{f\}$  は節点力ベクトルであり， $\dot{\phantom{x}}$  は時間微分を表す．また，上付き添え字  $s$  は，スライスに関する量であることを意味する．

このスライスが  $i$  番目のスライスであるとし，左断面上 ( $i - 1$  番目の断面上) と右断面上 ( $i$  番目の断面上) の節点に関する量にそれぞれ下添字  $i - 1$  と  $i$  を付けて表すこととする．式 (3.4) のマトリックスの行を入れ替える操作を行うことにより，有限要素式を次の様に書き換えることが出来る．

$$\begin{bmatrix} [M_1] & [M_2] \\ [M_3] & [M_4] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}_{i-1}\} \\ \{\ddot{u}_i\} \end{Bmatrix} + \begin{bmatrix} [C_1] & [C_2] \\ [C_3] & [C_4] \end{bmatrix} \begin{Bmatrix} \{\dot{u}_{i-1}\} \\ \{\dot{u}_i\} \end{Bmatrix} + \begin{bmatrix} [K_1] & [K_2] \\ [K_3] & [K_4] \end{bmatrix} \begin{Bmatrix} \{u_{i-1}\} \\ \{u_i\} \end{Bmatrix} = \begin{Bmatrix} \{f_{i-1}\} \\ \{f_i\} \end{Bmatrix}. \quad (3.5)$$

### §3.2.3 弾性体全体の方程式

$i$  番目のスライスの右断面上の節点は， $i + 1$  番目のスライスの左断面上の節点と同じものを意味しているということに注意して，前小節で求めた有限要素方程式を， $i = 1 \sim n$  までのすべてのスライスについて合成させることにより，弾性体全体の有限要素方程式が次式のように求められる．

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{f\}. \quad (3.6)$$

ただし，

$$[M] = \begin{bmatrix} [M_1] & [M_2] & & & & \\ [M_3] & [M_5] & [M_2] & & & \\ & \ddots & \ddots & \ddots & & \\ & & [M_3] & [M_5] & [M_2] & \\ & & & [M_3] & [M_4] & \end{bmatrix}, \quad (3.7)$$

$$[M_5] = [M_4] + [M_1], \quad (3.8)$$

$$[C] = \begin{bmatrix} [C_1] & [C_2] & & & & \\ [C_3] & [C_5] & [C_2] & & & \\ & \ddots & \ddots & \ddots & & \\ & & [C_3] & [C_5] & [C_2] & \\ & & & [C_3] & [C_4] & \end{bmatrix}, \quad (3.9)$$

$$[C_5] = [C_4] + [C_1], \quad (3.10)$$

$$[K] = \begin{bmatrix} [K_1] & [K_2] & & & & \\ [K_3] & [K_5] & [K_2] & & & \\ & \ddots & \ddots & \ddots & & \\ & & [K_3] & [K_5] & [K_2] & \\ & & & [K_3] & [K_4] & \end{bmatrix}, \quad (3.11)$$

$$[K_5] = [K_4] + [K_1], \quad (3.12)$$

$$\{u\} = \begin{Bmatrix} \{u_0\} \\ \{u_1\} \\ \vdots \\ \{u_{n-1}\} \\ \{u_n\} \end{Bmatrix}, \quad \{f\} = \begin{Bmatrix} \{f_0\} \\ \{f_1\} \\ \vdots \\ \{f_{n-1}\} \\ \{f_n\} \end{Bmatrix}. \quad (3.13)$$

上述のような規則的な要素分割を行ったために，弾性体全体に対するマトリックス  $[M]$ ， $[C]$ ， $[K]$  が1スライスに対する小さなマトリックスの規則的な繰り返しにより構成されたブロック3重対角マトリックスとなる．

### §3.2.4 方程式の解法

#### §3.2.4.1 陰解法による定式化

陰解法による動的有限要素運動解析には幾つかの計算手法があるが，ここでは Newmark の  $\beta$  法を用いた場合について述べることにする．その他の陰解法，例えば Wilson の  $\theta$  法などを用いる場合も，アルゴリズムは全く同じである．

時刻0での節点変位ベクトル  $\{u^0\}$ ，節点速度ベクトル  $\{\dot{u}^0\}$ ，及び節点加速度ベクトル  $\{\ddot{u}^0\}$  が既知である場合に，時刻1での変位ベクトル  $\{u^1\}$  を計算することを考える．Newmark の  $\beta$  法を用いるとすると， $\{u^1\}$  は次式で表される．<sup>[3.24]</sup>

$$\begin{aligned} \left( [K] + \frac{\delta}{\beta dt} [C] + \frac{1}{\beta dt^2} [M] \right) \{u^1\} = \{f^1\} + [M] & \left( \left( \frac{1}{2\beta} - 1 \right) \{\ddot{u}^0\} + \frac{1}{\beta dt} \{\dot{u}^0\} + \frac{1}{\beta dt^2} \{u^0\} \right) \\ & + [C] \left( \left( \frac{\delta}{2\beta} - 1 \right) dt \{\ddot{u}^0\} + \left( \frac{\delta}{\beta} - 1 \right) \{\dot{u}^0\} + \frac{\delta}{\beta dt} \{u^0\} \right). \end{aligned} \quad (3.14)$$

ただし， $dt$  は時間刻み， $\beta$ ， $\delta$  は解の安定性に関わる定数である．

式 (3.14) 右辺の各項は，前小節で示したマトリックス  $[M]$ ， $[C]$  の特徴 (小マトリックスの規則的繰り返しの性質) を用いることにより，効率的に計算できる．例えば  $\{\alpha\} = [M]\{u^0\}$  と置くと， $\{\alpha\}$  は次の様に計算される．

$$\begin{aligned} \{\alpha_0\} &= [M_1]\{u_0^0\} + [M_2]\{u_1^0\}, \\ \{\alpha_i\} &= [M_3]\{u_{i-1}^0\} + [M_5]\{u_i^0\} + [M_2]\{u_{i+1}^0\} \quad (1 \leq i \leq n-1), \\ \{\alpha_n\} &= [M_3]\{u_{n-1}^0\} + [M_4]\{u_n^0\}. \end{aligned} \quad (3.15)$$

式 (3.14) の左辺の括弧内を足し合わせたマトリックスを  $[A]$ ，右辺の全ての項を足し合わせたベクトルを  $\{b^1\}$  と置くと，方程式 (3.14) は次式の様に表される．

$$[A]\{u^1\} = \{b^1\}. \quad (3.16)$$



### §3.2.4.3 $[A']$ の収束特性

次式で漸化的に表されるマトリックス列  $[X_i]$  は，収束する性質があることが知られている<sup>[3.4]</sup>．

$$\begin{aligned} [X_0] &= [P], \\ [X_{i+1}] &= [Q] - [R]^*[X_i]^{-1}[R] \quad (i = 0, 1, 2, \dots). \end{aligned} \quad (3.22)$$

ここで  $[R]$  は任意の正方行列， $[P]$  および  $[Q]$  は正値対称行列であり， $[Q] - [P]$  も正値対称行列であるとする．

一般に質量マトリックス  $[M]$ ，粘性マトリックス  $[C]$ ，および剛性マトリックス  $[K]$  はすべて正値対称行列である．<sup>[3.3]</sup> 式 (3.14) に示す通り， $[A]$  は  $[M], [C]$ ，および  $[K]$  を正の係数で重ね合わせたものである為， $[A]$  もまた正値対称行列である． $[A_1]$  および  $[A_4]$  は  $[A]$  のブロック対角成分であるから，これらも正値対称行列である． $[A_2]$  および  $[A_3]$  は  $[A]$  の対称なブロック成分であるから， $[A_2]^T = [A_3]$  なる関係がある．従って，式 (3.19) は式 (3.22) と全く同じ形をしていることになり，式 (3.19) で定義された  $[A'_i]$  は収束列であることが示される．

以上により， $[A'_{k-1}] \neq [A'_k] \simeq [A'_{k+1}] \simeq [A'_{k+2}] \simeq \dots$  を満たす整数  $k$  が存在する．その整数が  $k < n$  を満たす場合， $[A'_{k+1}]$  から  $[A'_{n-1}]$  までのマトリックスは計算する必要もメモリ上に記憶しておく必要も無い． $[A'_i]$  の計算は初期化処理中に行われるだけであるので， $[A'_i]$  の収束性を用いても全体の解析時間はあまり減少しない．しかし， $[A'_i]$  の記憶容量は解析に必要な記憶容量のほとんどを占めているため， $[A'_i]$  の収束性を用いることにより記憶容量は劇的に減少する．

本手法では次式によって  $[A'_i]$  の収束を判断している．

$$\max(|[A'_{i+1}] - [A'_i]|) < \varepsilon \max(|[A'_0]|), \quad (3.23)$$

ここで  $\varepsilon$  は小さな正の定数 (例えば  $10^{-8}$ ) であり， $\max(|[X]|)$  は  $[X]$  の成分で絶対値が最大のものを意味する．本手法による誤差を直接的に決定することは困難であるが， $\varepsilon$  を段階的に変えて計算して解を比較することによって十分に小さい  $\varepsilon$  を決定することは可能である．

Guoらの論文<sup>[3.4]</sup>によると，収束速度は1次である．しかしながら，マトリックス列はメッシュサイズ，時間刻み，断面形状，およびメッシュパターン等によって変化する為， $k$  を解析的に推定する方法は現在のところ未開発であり，今後の課題である．

### §3.2.4.4 Dirichlet 境界を取扱う際の修正方法

上の議論では，すべての境界で変位  $u$  が未知で荷重  $f$  が既知であるかの様に計算を進めて来た．しかし，一般的には荷重指定境界 (変位が未知で荷重が既知) と変位指定境界 (変位が既知で荷重が未知) とが混在する．本手法は，荷重指定境界と変位指定境界が全スライスにおいて同様に混在する場合にも，以下に示す操作を施すことにより適用することができる．

仮に，式 (3.16) において，変位指定境界が  $q$  行目に与えられているとすれば，マトリックス  $[A]$  の  $(q, q)$  成分を除いて  $q$  行目の成分をすべて 0 とし，ベクトル  $\{b^1\}$  の  $q$  行目の値を指定された変位を満足する値に設定する．この操作により，指定された変位を満足する解が得られる．

荷重指定境界と変位指定境界が全スライスにおいて同様に混在するならば，上の操作を施す行が繰り返し現れることになる．操作後のマトリックスを  $[B]$  とおけば，やはり  $[B]$  も  $[B_1] \sim [B_5]$  の規則的な繰り返しの性質を持つことになる．従って，先に説明した漸化的計算法をそのまま用いて次時刻における変位ベクトル  $\{u^1\}$  を求めることが出来る．

### §3.2.4.5 $\{u\}$ の更新

$\{\dot{u}^1\}$ ,  $\{\ddot{u}^1\}$  は，次式から求められる．<sup>[3.24]</sup>

$$\{\dot{u}^1\} = (1 - \frac{\delta}{\beta})\{\dot{u}^0\} + (1 - \frac{\delta}{2\beta})dt\{\ddot{u}^0\} + \frac{\delta}{\beta dt}(\{u^1\} - \{u^0\}), \quad (3.24)$$

$$\{\ddot{u}^1\} = (1 - \frac{1}{2\beta})\{\ddot{u}^0\} - \frac{1}{\beta dt}\{\dot{u}^0\} + \frac{1}{\beta dt^2}(\{u^1\} - \{u^0\}). \quad (3.25)$$

以上のステップを  $\{u^2\}$ ,  $\{u^3\}$ , ... と繰り返すことにより，弾性体の動的挙動を解析することが出来る．

## §3.3 解析例

### §3.3.1 片持ち梁の衝撃曲げ

片持ち梁の衝撃曲げ問題を通常の方法と本手法で解析する．図 3.2 に示すような正方形断面を持つ角棒 ( $10 \times 10 \times 300$ [mm]) の弾性体の一側の端面を固定し，もう一方の断面に時刻  $t = 0$  からデルタ関数状に一樣な衝撃力を  $-x$  方向に作用させることを考える．棒の物性値を表 3.1 に示す．

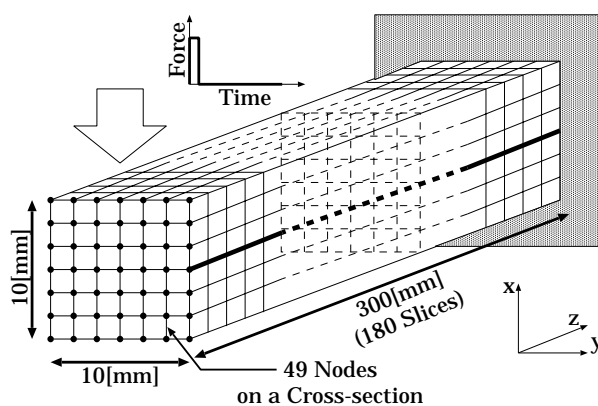


Fig. 3.2: Cantilever subject to lateral impulsive load

角棒を母線に沿って 180 等分し，各スライスを，66 個の要素に分割した．要素はすべて 8 節点六面体要素 (HEXL8)<sup>[3.28]</sup>とした．従って，全節点数は 8869，全自由度は 26607 である．時間刻み

Table 3.1: Material properties

Young's Modulus	$E$	205 [GPa]
Poisson's Ration	$\nu$	0.25
Density	$\rho$	7800 [kg/m <sup>3</sup> ]

$dt$  は 60[ns] で一定とした．図 3.2 に黒丸で示した節点に  $t = 0 \sim dt$  の間にだけ衝撃力を与え，その力積は 1[mNs] である．

この問題を本手法を用いた自作のプログラムと，汎用有限要素プログラム ABAQUS/Standard の 2 通りで解析した．本手法の陰解法には  $\beta = 0.25, \delta = 0.5$  とした Newmark $\beta$  法を用い， $[A'_i]$  の収束判定は  $\varepsilon = 10^{-8}$  として行った．ABAQUS/Standard は初期設定では Hilber-Hughes-Taylor 法を用いるが，入力ファイルにおいて手法のパラメータである  $\alpha$  をゼロとして本手法と全く同じ時間積分式を用いるようにしている．

時刻  $t = 60[\mu s]$  および  $180[\mu s]$  について，梁の側面中央 (図 3.2 の太線) における横方向変位 ( $u_x$ ) の解析結果を図 3.3 に示す．両時刻において，本手法による結果は ABAQUS/Standard と良く一致していることが分かる．

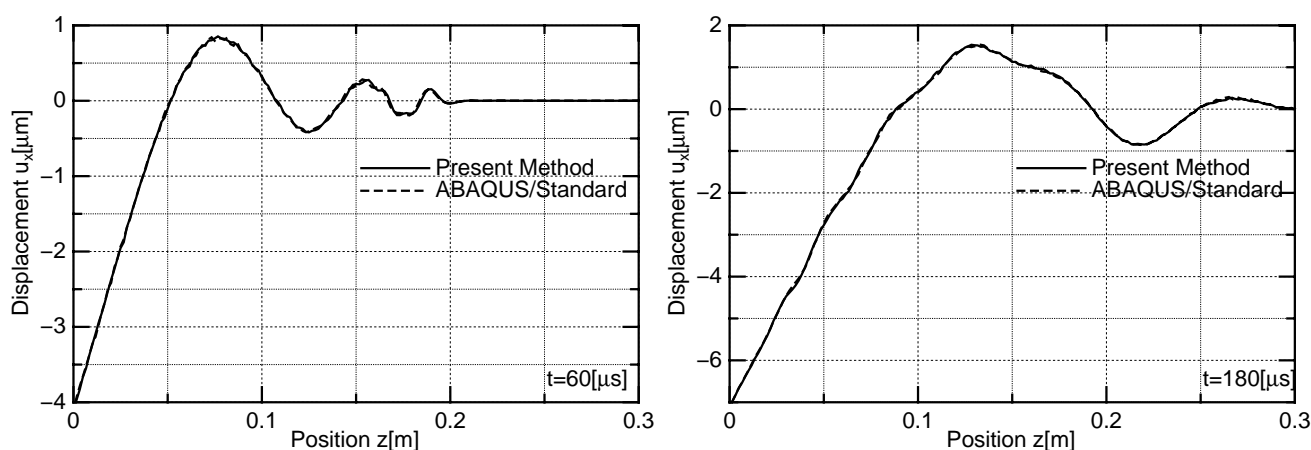


Fig. 3.3: Distribution of displacement( $u_x$ ) of the cantilever( $t=60[\mu s], 180[\mu s]$ )

双方の計算を SGI 社製 Origin2000 上で 3000 時間ステップの計算を本手法と ABAQUS/Standard で行い，計算時間，および記憶容量の比較を行った．本手法，ABAQUS/Standard 共に BLAS<sup>[3.8]</sup> や LAPACK<sup>[3.9]</sup> は用いていない．計算時間および記憶容量の比較結果を表 3.2 に示す．計算時間，記憶容量共に本手法によって節約されており，計算時間は約 1/80，記憶容量も約 1/30 に節約されている．ちなみに， $[A'_i]$  は  $i = 5$  で収束に到った．構造物の節点数が多くなる程，本手法を用いることによる節約量は大きくなるものと考えられる．

Table 3.2: Comparison of CPU time and memory

	CPU Time	Memory
Present Method	0.4 [h]	3.6 [Mbyte]
ABAQUS/Standard	33.5 [h]	109 [Mbyte]

### §3.3.2 側面を拘束した角棒の横衝撃

側面を拘束した角棒に横衝撃が加わる問題を本手法と通常的手法とで解析する．解析対象の概観を図3.4に示す．角棒の形状やメッシュは前例題と全く同じであるが，境界条件が異なっている．1つの側面が壁に完全に固定されており，反対側の側面の一部に衝撃力を受ける境界条件となっている．従って，Dirichlet境界条件とNeumann境界条件が混在する問題となっている．

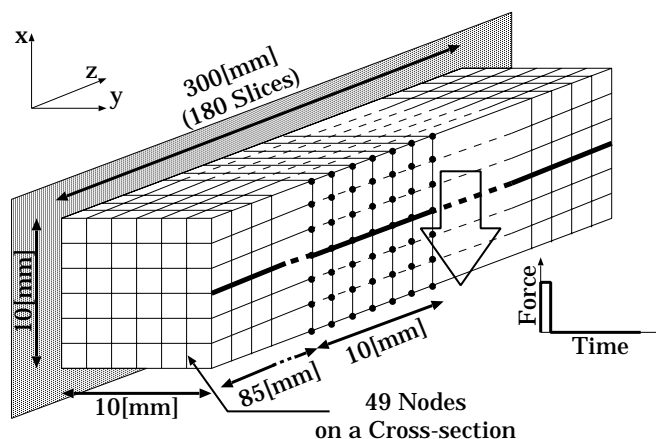


Fig. 3.4: Beam with fixed side surface subjected to impulsive loading

衝撃力は  $-x$  方向に  $t = 0$  から  $\Delta t$  の間のみを与え，与えられる力積は 1[mNs] である．材料定数は前例題に示した表 3.1 の通りである．また解析に用いた陰解法とそのパラメータも前例題と全く同じある．

解析結果として，時刻  $t=60[\mu s]$  と  $180[\mu s]$  において得られた図 3.4 中の太線で示した線上での  $x$  方向の変位を図 3.5 に示す．本手法と ABAQUS/Standard 双方の結果を重ねて描いてある．時刻  $t=60[\mu s]$  では良く一致おり，時刻  $t=180[\mu s]$  では部分的にずれが生じている．本例題での境界条件は非常に大きな高周波成分を含む例となっている点が前例題と異なっており，打ち切り誤差や浮動小数点誤差などの影響が拡大しやすい問題になっている．時刻  $t=180[\mu s]$  でのずれはこれらが原因となって生じているものと考えられる．しかしながら，本手法はこの問題の現象を解析する上で必要な精度は持ち合わせていると言える．

計算時間および記憶容量については前例題とほとんど同じである．マトリックス列  $[A'_i]$  は  $i = 8$  で収束した．この結果は，Dirichlet境界条件とNeumann境界条件とが母線方向に一樣に混在している問題に大しても本手法が有効であることを示している．

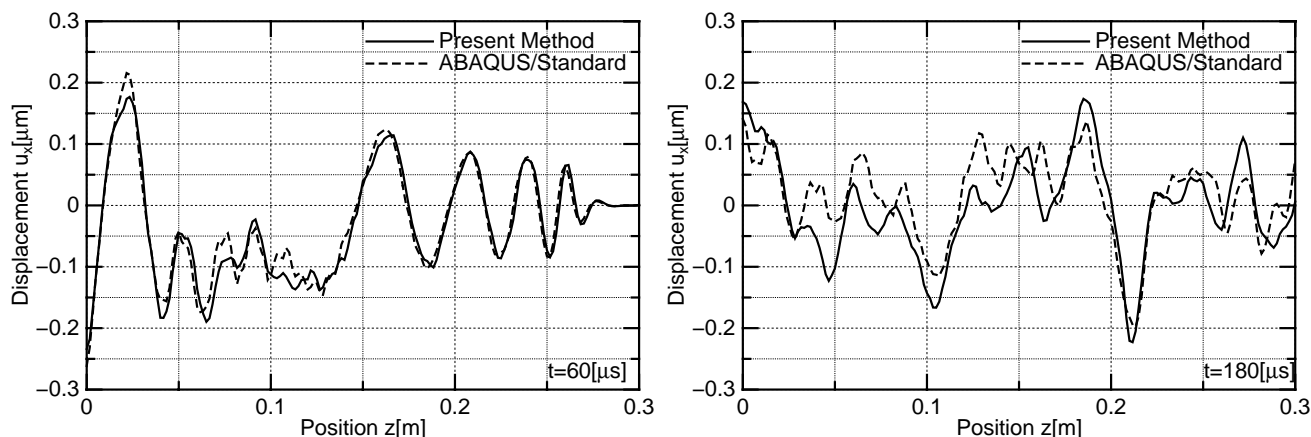


Fig. 3.5: Distribution of displacement( $u_y$ ) of the beam ( $t=60[\mu s]$ ,  $180[\mu s]$ )

### §3.3.3 Hopkinson-bar の全体解析

本手法の高効率性を示すため，パーソナルコンピュータを用いての Hopkinson-bar<sup>[3.5, 6]</sup>の全体解析の例を示す．Hopkinson-barは材料の高速ひずみ域での応力 - ひずみ線図を測定する為の実験装置である．試験片から Hopkinson-bar へと伝播する応力波により生じるひずみを，Hopkinson-bar の表面に張り付けたひずみゲージにより測定する．ひずみゲージの張り付け位置が試験片に近すぎる場合，試験片との接触面付近の応力の分散の影響が現れる為，適切なひずみ時刻歴が測定出来ない．特に試験片の設置位置が Hopkinson-bar に対して中心からずれている場合，応力の分散の影響は大きくなる．これらの影響を定量的に見積もる為，以下の数値解析を行う．

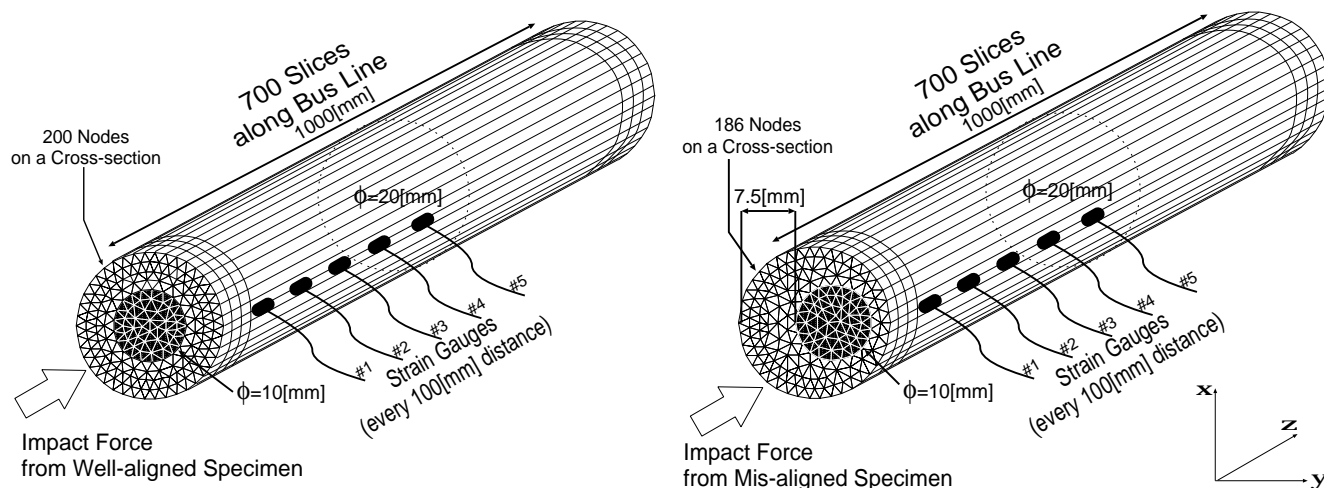


Fig. 3.6: Illustration of Hopkinson-bars (long circular cylinders) receiving impact force from well-aligned specimen and mis-aligned specimen

解析対象となる円形一様断面の Hopkinson-bar の概観図を図 3.6 に示す．黒塗りの面は試験片との接触面を表しており，左図では試験片が中心に正しく接触している場合，右図では試験片が中心からずれて接触している場合を示している．Hopkinson-bar はすべて 6 節点三角形プリズム要素で離散化されている．スライス数は双方とも 700 個であり，断面内の節点数は「正しく接触している場合」が 200 点，「ずれて接触している場合」が 186 点ある．従って，「正しく接触している場合」は全節点数が 140200 点で全自由度が 420600 であり，「ずれて接触している場合」は全節

点数が 130386 点で全自由度が 391158 である。

Hopkinson-bar の材料定数は先に示した解析例と同じで，表.3.1 に示す通りである．入力された応力は接触面内で均一であると仮定し，その時刻歴は図 3.7 に示すグラフのように与える．図 3.6 に示す通り，ひずみゲージは試験片との接触面から 100[mm] おきに 5 個張り付けてあるものとし，ゲージの各位置におけるひずみ時刻歴を解析の出力とした．解析の時間刻み  $\Delta t$  は 0.2[ $\mu$ s] とし，全部で 1000 ステップの計算を行った．陰解法には  $\beta = 0.25$ ， $\delta = 0.6$  とした Newmark $\beta$  法を用い，収束判定は  $\varepsilon$  を  $10^{-8}$  とした．

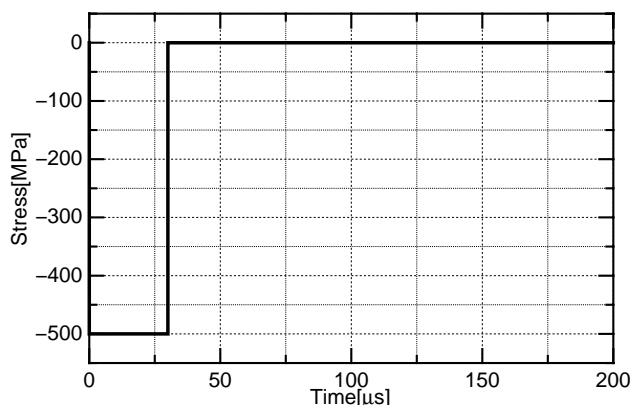


Fig. 3.7: Time history of applied stress to the Hopkinson-bar

解析により得られた各ひずみゲージ位置での応力時刻歴を図 3.8 に示す．#1,#2,#3 のゲージ位置では「正しく接触している場合」と「ずれて接触している場合」との応力時刻歴に差が生じている．#4,#5 のゲージ位置では双方の結果が良く一致しており，また分散の影響が緩和されて定常波となっていることが分かる．この結果から，直径 20[mm] の Hopkinson-bar の場合，試験片との接触面から最低 400[mm] 以上離れた位置にひずみゲージを張り付ける必要があることが結論付けられる．

この解析は CPU に Intel Pentium4 2.4[GHz] を搭載したパーソナルコンピュータで BLAS<sup>[3.8]</sup> および LAPACK<sup>[3.9]</sup> を内部で用いている行列計算ライブラリ CPPLapack で行った． $[A'_i]$  は「正しく接触している場合」と「ずれて接触している場合」の両方とも  $i = 11$  で収束した．計算時間は双方とも約 1.5[h]，記憶容量は双方とも約 180[MByte] であった． $[A'_i]$  ( $i = 0, 1, \dots, 11$ ) を計算するのに要した時間は約 10[s] であり， $[A'_i]$  の収束特性を用いなかった場合の計算時間は約 1.6[h]，記憶容量は 12,600[MByte] になると予想される．この結果は，本手法が Hopkinson-bar 全体の陰解法による有限要素動解析をパーソナルコンピュータを用いて行うことが実用的に可能であることを示している．

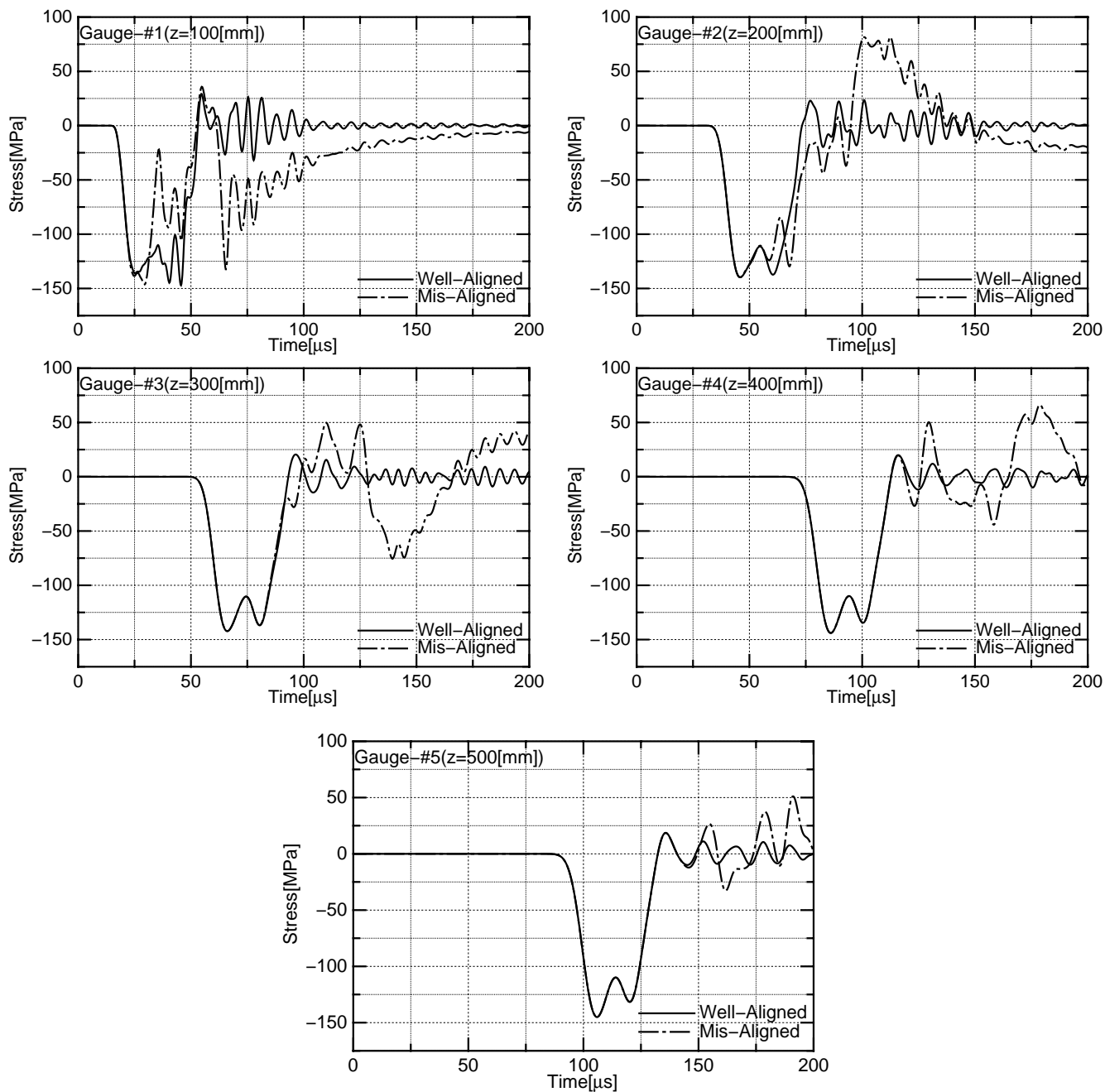


Fig. 3.8: Simulated time histories of stress at locations of strain gauges

## §3.4 実用例：一点曲げ衝撃試験の数値シミュレーションへの適用

### §3.4.1 緒言

セラミックスを初めとして様々な新しい材料が構造材料として広く使用されようとしているが、衝撃荷重に対する安全性を確保するためには、正確な衝撃破壊靱性や高速き裂進展靱性を知る必要がある。しかし、これらの靱性の測定には様々な誤差が含まれ易いために、未だに簡便で精度の高い試験法が模索されており、標準的な試験法の確立が急がれている。

Kalthoff<sup>[3.10]</sup>、Giovanola<sup>[3.11]</sup>、坂田ら<sup>[3.12]</sup>は、中央に片側き裂を持つ試験片の両端を細い糸で吊る等の方法により応力自由とし、中央部に落下棒等により衝撃力を加えて試験片を破壊させる一点曲げ衝撃試験法(後出の図3.13参照)を考案している。この試験法は、三点曲げ衝撃試験で問題となる支持点における摩擦、再衝突等の影響を避けることができるので、有効な方法と考えられる。

一般に、衝撃破壊靱性試験や高速き裂進展靱性試験においては動的応力拡大係数を求める必要があるが、通常の有限要素法を用いた汎用的な数値解法に頼るのは経済的でないと考えられて来た。藤野ら<sup>[3.13]</sup>は一点曲げ衝撃試験片を Euler-Bernoulli 梁としてモデル化して、静止き裂に対する動的応力拡大係数を簡便に求める方法を提案し、計算時間を通常の有限要素法の約 1 / 100 に低減している。黒田ら<sup>[3.14]</sup>は、回転慣性とせん断変形を考慮した Timoshenko 梁理論を用いることにより藤野らによる簡便法の精度を向上させている。

しかし、一点曲げ衝撃試験におけるき裂進展開始後の試験片の挙動については不明な点が多い。例えば、き裂進展開始の後衝撃荷重はどのように変化するのか、その変化にはき裂進展速度がどの程度影響するのか、などの問題に対しては研究がなされていない。これらの挙動が明らかになって始めて、一点曲げ衝撃試験の改良や高速き裂進展靱性試験への拡張などの可能性が検討できると考えられる。

そこで本研究では、黒田らの Timoshenko 梁理論による方法を拡張して、高速進展するき裂の挙動を模擬できるようにし、Hertz の接触理論を利用して衝撃棒と(最初は静止き裂、途中から高速進展き裂を持つ)試験片との衝突挙動をシミュレートする。さらに、一様断面弾性体に対する有限要素高速解法を用いたシミュレーションと比較して精度と計算速度の検証を行う。また、上記の一点曲げ衝撃試験の改良や拡張などの可能性を調べる。

### §3.4.2 静止き裂を持つ Timoshenko 梁モデル

この節では、詳細を黒田ら<sup>[3.14]</sup>に譲り、後の議論に必要な部分のみを記述することとする。

§3.4.2.1 基礎方程式

図1に示すように、静止き裂を持つ試験片を Timoshenko 梁でモデル化して、両端自由の試験片の中央に衝撃力  $F(t)$  が作用する場合を考える．き裂を Nash の方法<sup>[3,16]</sup>でモデル化すると，基礎方程式は次式で与えられる．

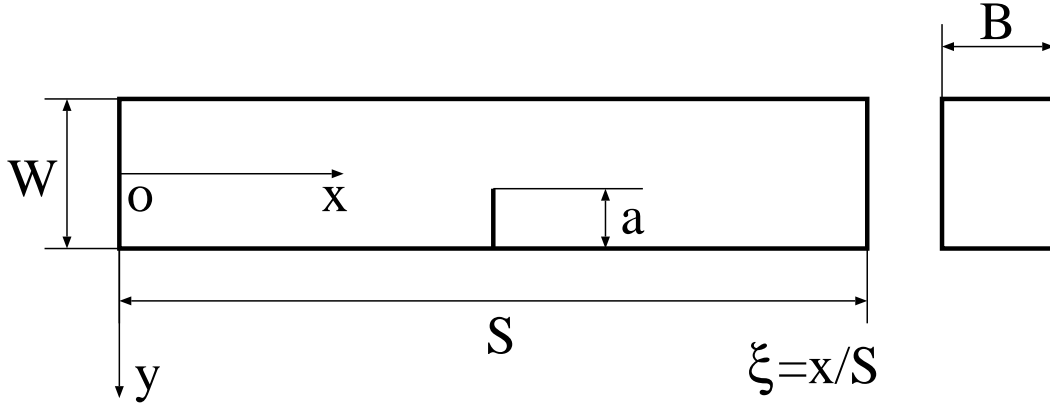


Fig. 3.9: One-point-bend specimen

$$\rho A \frac{\partial^2 u}{\partial t^2} = \frac{1}{S} \frac{\partial Q}{\partial \xi} + \frac{1}{S} F(t) \delta(\xi - \frac{1}{2}) \quad (3.26)$$

$$\rho I \frac{\partial^2 \psi}{\partial t^2} = Q + \frac{1}{S} \frac{\partial M}{\partial \xi} \quad (3.27)$$

$$Q = \kappa AG \left( \frac{1}{S} \frac{\partial u}{\partial \xi} - \psi \right) \quad (3.28)$$

$$M = \frac{E}{S\phi} \frac{\partial \phi}{\partial \xi} \quad (3.29)$$

$$\phi = \frac{1}{I} + \frac{D}{S} \delta(\xi - \frac{1}{2}) \quad (3.30)$$

$$\kappa = \frac{10(1 + \nu)}{12 + 11\nu} \quad (3.31)$$

ここで，曲げ変形に関して断面二次モ - メントの逆数を式 (3.30) のように仮定することにより，き裂の存在が考慮されている．

上式において  $\xi$  は，試験片のスパン長さ  $S$  によって無次元化した  $x$  座標であり， $\delta(\xi)$  は Dirac のデルタ関数， $u$  はたわみによる変位， $\psi$  は回転角， $Q$  はせん断力， $M$  はモ - メントである． $\rho$ ， $\nu$ ， $E$ ， $G$ ， $A$  および  $I$  はそれぞれ，試験片の密度，ポアソン比，縦弾性係数，横弾性係数，断面積 ( $= BW$ ) および断面二次モ - メント ( $= BW^3/12$ ) である． $\kappa$  は横断面形状に依存するせん断応

力分布係数である． $D$  はき裂の効果を表す係数であり，次式で与えられる．ただし， $\alpha = a/W$

$$D = \frac{2(1-\nu^2)W}{I} V(\alpha) \quad (3.32)$$

$$V(\alpha) = \left(\frac{\alpha}{1-\alpha}\right)^2 \times (5.58 - 19.57\alpha + 36.82\alpha^2 - 34.94\alpha^3 + 12.77\alpha^4) \quad (3.33)$$

両端でモーメントおよびせん断力がゼロなので，次の境界条件式 (3.34) が与えられる．また， $t = 0$  で静止しているので，初期条件は式 (3.35) となる．

$$\left(\frac{\partial \psi}{\partial \xi}\right)_{\xi=0,1} = \left(\frac{1}{S} \frac{\partial u}{\partial \xi} - \psi\right)_{\xi=0,1} = 0 \quad (3.34)$$

$$(u)_{t=0} = (\psi)_{t=0} = \left(\frac{\partial u}{\partial t}\right)_{t=0} = \left(\frac{\partial \psi}{\partial t}\right)_{t=0} = 0 \quad (3.35)$$

### §3.4.2.2 固有関数

式 (3.26) において  $F(t) = 0$  とし，変数分離解

$$u(\xi, t) = Y(\xi)e^{i\omega t} \quad (3.36)$$

$$\psi(\xi, t) = \Psi(\xi)e^{i\omega t} \quad (3.37)$$

を仮定する．ここで， $i = \sqrt{-1}$  である．

これらの式を式 (3.26) および (3.27) に代入し，得られた式を整理した後  $\xi$  についてラプラス変換し， $\xi = 0$  の境界条件を用いて  $Y$  および  $\Psi$  について解き，ラプラス逆変換を行う．さらに， $\xi = 1$  の境界条件を考慮すると，振動数方程式として次式が得られる．

(a)  $b^2 > 1/(q^2 r^2)$  の場合

$$\begin{aligned} 2 - 2 \cos \alpha \cos \beta - \frac{T_1^2 + T_2^2}{T_1 + T_2} \sin \alpha \sin \beta - Z \left( \left\{ \sin \alpha \cos \beta - 2 \sin \frac{\alpha}{2} \cos \frac{\beta}{2} (2 - \cos \frac{\alpha}{2} \cos \frac{\beta}{2}) \right\} T_1 \right. \\ \left. - \left\{ \cos \alpha \sin \beta - 2 \cos \frac{\alpha}{2} \sin \frac{\beta}{2} (2 - \cos \frac{\alpha}{2} \cos \frac{\beta}{2}) \right\} T_2 \right. \\ \left. + \frac{T_1^2}{T_2} \sin^2 \frac{\alpha}{2} \sin \beta - \frac{T_2^2}{T_1} \sin \alpha \sin^2 \frac{\beta}{2} \right) = 0 \quad (3.38) \end{aligned}$$

ただし，

$$b^2 = \frac{\rho A}{EI} S^4 \omega^2, \quad r^2 = \frac{I}{AS^2}, \quad q^2 = \frac{EI}{\kappa GAS^2} \quad (3.39)$$

$$\left[ \begin{array}{c} \alpha^2 \\ \beta^2 \end{array} \right] = \frac{b^2(q^2 + r^2) \mp \{b^4(q^2 - r^2)^2 + 4b^2\}^{\frac{1}{2}}}{2} \quad (3.40)$$

$$T_1 = \alpha(b^2 q^2 - \alpha^2), \quad T_2 = \beta(b^2 q^2 - \beta^2) \quad (3.41)$$

$$Z = \frac{DI}{(\alpha^2 - \beta^2)S} \quad (3.42)$$

角振動数  $\omega$  に関する方程式 (3.38) (式 (3.39)-(3.42) 参照) の第  $m$  番目の根を  $\omega_{Fm}$  とし, これに対応する  $b, \alpha$  および  $\beta$  の値をそれぞれ  $b_{Fm}, \alpha_{Fm}$  および  $\beta_{Fm}$  とすると, たわみによる変位および回転角に関する固有関数として次式が得られる.

$$Y_{Fm}(\xi) = A_c \left( \sin \alpha_{Fm} \xi - \frac{\beta_{Fm}}{\alpha_{Fm}} \sin \beta_{Fm} \xi \right) + C_c \left( \cos \alpha_{Fm} \xi - \frac{\beta_{Fm} T_1}{\alpha_{Fm} T_2} \cos \beta_{Fm} \xi \right) \quad (3.43)$$

$$(0 \leq \xi \leq \frac{1}{2}), (m = 1, 2, 3, \dots)$$

$$\Psi_{Fm}(\xi) = \frac{1}{S} \left\{ A_c \left( -\frac{T_1}{\alpha_{Fm}^2} \cos \alpha_{Fm} \xi + \frac{T_2}{\alpha_{Fm} \beta_{Fm}} \cos \beta_{Fm} \xi \right) + C_c \left( \frac{T_1}{\alpha_{Fm}^2} \sin \alpha_{Fm} \xi - \frac{T_1}{\alpha_{Fm} \beta_{Fm}} \sin \beta_{Fm} \xi \right) \right\} \quad (3.44)$$

$$(0 \leq \xi \leq \frac{1}{2}), (m = 1, 2, 3, \dots)$$

ここで,  $A_c/C_c$  は次式で表される.

$$\frac{A_c}{C_c} = \frac{T_1 \left( \cos \alpha_{Fm} - \cos \beta_{Fm} \right) + T_1 \left( \cos \frac{\alpha_{Fm}}{2} - \cos \frac{\beta_{Fm}}{2} \right) Z P}{-T_1 \sin \alpha_{Fm} + T_2 \sin \beta_{Fm} - \left( T_1 \sin \frac{\alpha_{Fm}}{2} - T_2 \sin \frac{\beta_{Fm}}{2} \right) Z P} \quad (3.45)$$

ただし,  $T_1, T_2$  および  $Z$  は式 (3.41) および (3.42) に  $\alpha_{Fm}$  および  $\beta_{Fm}$  を代入して得られる定数であり,  $P$  は次式から計算される定数である.

$$P = T_1 \sin \frac{\alpha_{Fm}}{2} - T_2 \sin \frac{\beta_{Fm}}{2} \quad (3.46)$$

(b)  $b^2 < 1/(q^2 r^2)$  の場合

式 (3.45) において  $\alpha$  が虚数になるが, 三角関数と双曲線関数の関係式 ( $\cos ix = \cosh x$  など)<sup>[3.17]</sup> を用いると, 式 (3.38)-(3.46) を実関数の式に書き換えることができる.

### §3.4.2.3 外力に対する応答

外力  $F(t)$  に対する  $u(\xi, t)$  および  $\psi(\xi, t)$  の応答が次式に示すように変数分離形で表されると仮定する.

$$u(\xi, t) = \sum_{m=0}^{\infty} Y_{Fm}(\xi) \Theta_m(t) \quad (3.47)$$

$$\psi(\xi, t) = \sum_{m=0}^{\infty} \Psi_{Fm}(\xi) \Theta_m(t) \quad (3.48)$$

式 (3.47) および (3.48) を式 (3.26) および (3.27) に代入して整理し, 直交条件式 (3.49)<sup>[3.18]</sup> を用い

ると，式 (3.50) が得られる．

$$\int_0^1 \left( Y_{Fm}(\xi) Y_{Fn}(\xi) + \frac{I}{A} \Psi_{Fm}(\xi) \Psi_{Fn}(\xi) \right) d\xi = 0 \quad (m \neq n) \quad (3.49)$$

$$\omega_{Fm}^2 W_{Fm} \Theta_m + W_{Fm} \ddot{\Theta}_m = \frac{1}{S} F(t) Y_{Fm}\left(\frac{1}{2}\right) \quad (3.50)$$

ただし， $(\cdot)$  は時間微分を表す．また，

$$\omega_{Fm}^2 = \frac{b_{Fm}^2 EI}{S^4 \rho A} \quad (3.51)$$

$$W_{Fm} = 2\rho A \int_0^{1/2} \left( Y_{Fm}^2(\xi) + \frac{I}{A} \Psi_{Fm}^2(\xi) \right) d\xi \quad (3.52)$$

式 (3.50) より次式が得られる．

$$\Theta_m(t) = \frac{Y_{Fm}\left(\frac{1}{2}\right)}{\omega_{Fm} W_{Fm} S} \int_0^t F(\tau) \sin \omega_{Fm}(t - \tau) d\tau \quad (m = 1, 2, \dots) \quad (3.53)$$

式 (3.53) を (3.47) および (3.48) に代入すると  $u(\xi, t)$  および  $\psi(\xi, t)$  の応答が求められる．試験片各部のモーメントは  $\psi(\xi, t)$  を式 (3.29) に代入することにより，試験片の変位は  $u(\xi, t)$  に剛体運動を加えることにより求めることができる．(後出の式 (3.71) 参照)

#### §3.4.2.4 動的応力拡大係数の簡便式

動的応力拡大係数  $K_I(t)$  が試験片中央近傍のモーメントに比例すると仮定し，その比例定数  $k_m$  は静的な場合と等しいと仮定すると，

$$K_I(t) = k_m \lim_{\varepsilon \rightarrow 0} M\left(\frac{1}{2} - \varepsilon, t\right) \quad (3.54)$$

$$k_m = K_s / F(t) \frac{EI}{S} \sum_{m=1}^{\infty} \frac{Y_{Sm}\left(\frac{1}{2}\right) \Psi'_{Sm}\left(\frac{1}{2}\right)}{\omega_{Sm}^2 W_{Sm} S} \quad (3.55)$$

が得られる．ここで，静的な応力拡大係数  $K_s$  はハンドブックの値<sup>[3.19]</sup>を用いる．下添字  $S$  の付いたその他の量は三点曲げに関する量であり，文献<sup>[3.14]</sup>に評価式が与えられている(文献<sup>[3.14]</sup>の表示では下添字  $S$  は付いていない)．式 (3.54) より， $K_I(t)$  の簡便式として次式が得られる．

$$K_I(t) = k_m \frac{EI}{S} \sum_{m=1}^{\infty} \Psi'_{Fm}\left(\frac{1}{2}\right) \Theta_m(t) \quad (3.56)$$

ただし， $(\cdot)$  は  $\xi$  による微分を表す．

### §3.4.3 高速進展き裂を持つ Timoshenko 梁モデル

#### §3.4.3.1 外力に対する応答

き裂長さ  $a$  が時間  $t$  の関数となることを除けば，基礎方程式は式 (3.26)-(3.35) と同じである．また，剛体運動も静止き裂の場合と同じである．しかし，式 (3.53) のような単純なたたみこみ積分を用いることができなくなる．き裂が進展すると，それに伴って角振動数が時間  $t$  の関数となるため， $u$  および  $\psi$  をつぎのように置くこととする．

$$u(\xi, t) = \sum_{m=1}^{\infty} Y_{Fm}(\xi, t) \Theta_m(t) \quad (3.57)$$

$$\psi(\xi, t) = \sum_{m=1}^{\infty} \Psi_{Fm}(\xi, t) \Theta_m(t) \quad (3.58)$$

ここで， $Y_{Fm}(\xi, t)$  および  $\Psi_{Fm}(\xi, t)$  は，それぞれ時刻  $t$  のき裂長さに対する，たわみによる変位および回転角の固有関数である．この場合には式 (3.49) の直交条件が成立しないので，剛体運動の他は  $m = 1$  のみを考えることとする．そこで， $m = 1$  として，式 (3.57) および (3.58) を基礎式 (3.26) および (3.27) に代入すると次式が得られる．

$$\rho A \left( Y_{F1} \ddot{\Theta}_1 + 2\dot{Y}_{F1} \dot{\Theta}_1 + \ddot{Y}_{F1} \Theta_1 \right) + \rho A \omega_{F1}^2 Y_{F1} \Theta_1 = \frac{1}{S} F(t) \delta\left(\xi - \frac{1}{2}\right) \quad (3.59)$$

$$\rho I \left( \Psi_{F1} \ddot{\Theta}_1 + 2\dot{\Psi}_{F1} \dot{\Theta}_1 + \ddot{\Psi}_{F1} \Theta_1 \right) + \rho I \omega_{F1}^2 \Psi_{F1} \Theta_1 = 0 \quad (3.60)$$

ただし，

$$\omega_{F1}^2 = \frac{b_{F1}^2 EI}{S^4 \rho A} \quad (3.61)$$

上式のように，2つの方程式が得られるので， $m = 2$  まで本節の方法で解析可能であるが，後述の一点曲げ衝撃試験のシミュレーションにおいて，静止き裂を持つ試験片の動的応力拡大係数は  $m = 5$  まで計算した値と  $m = 1$  のみで計算した値の間に大きな差異はない (図 3.14 参照) ことから， $m = 1$  のみを考えた．この場合には，式 (3.59) のみからでも解が得られるが，ここでは静止き裂の場合との整合性から両方の式を用いた (両者の解の間に数値的にはほとんど差異がなかった)．式 (3.59) および (3.60) にそれぞれ  $Y_{F1}$  および  $\Psi_{F1}$  を掛けて，両式の和をとり， $\xi$  について 0 から 1 まで積分すると次式が得られる．

$$W_{AF1} \ddot{\Theta}_1 + 2W_{BF1} \dot{\Theta}_1 + \omega_{F1}^2 W_{AF1} \Theta_1 + W_{CF1} \Theta_1 = \frac{1}{S} F(T) Y_{F1}\left(\frac{1}{2}\right) \quad (3.62)$$

ここで，

$$W_{AF1} = 2\rho A \int_0^{1/2} \left( Y_{F1}^2(\xi, t_n) + \frac{I}{A} \Psi_{F1}^2(\xi, t_n) \right) d\xi \quad (3.63)$$

$$W_{BF1} = 2\rho A \int_0^{1/2} \left( \dot{Y}_{F1}(\xi, t_n) Y_{F1}(\xi, t_n) + \frac{I}{A} \dot{\Psi}_{F1}(\xi, t_n) \Psi_{F1}(\xi, t_n) \right) d\xi \quad (3.64)$$

$$W_{CF1} = 2\rho A \int_0^{1/2} \left( \ddot{Y}_{F1}(\xi, t_n) Y_{F1}(\xi, t_n) + \frac{I}{A} \ddot{\Psi}_{F1}(\xi, t_n) \Psi_{F1}(\xi, t_n) \right) d\xi \quad (3.65)$$

式(3.62)において、時刻0から $t$ までの区間を微小時間 $\Delta t$ に分割し、各時間間隔 $\Delta t$ 内では $\Theta, \dot{\Theta}$ および $\ddot{\Theta}$ の各係数が一定であると仮定する。式(3.64)中の $\dot{Y}_{F1}$ および $\dot{\Psi}_{F1}$ はき裂長さの変化による $\alpha_{F1}$ および $\beta_{F1}$ の変化の影響を考慮して、解析的に求められる(式が長くなるのでここでは記述を省略する)が、式(3.65)中の $\dot{Y}_{F1}$ および $\dot{\Psi}_{F1}$ は式が煩雑となるため、 $\dot{Y}_{F1}$ および $\dot{\Psi}_{F1}$ の一階差分により求める。ある時間間隔( $t_{n-1} - t_n$ )において、式(3.34)の境界条件の下で式(3.62)をラプラス変換法により $\Theta_1$ について解くと次式が得られる。

$$\Theta_1(t_n) = \frac{Y_{F1}(\frac{1}{2})}{\beta W_{AF1} S} \int_{t_{n-1}}^{t_n} F(\tau) e^{\alpha(\tau-t_n)} \sinh \beta(t_n - \tau) d\tau + \Theta_1(t_{n-1}) e^{-\alpha \Delta t} \left( \cosh \beta \Delta t + \frac{\alpha}{\beta} \sinh \beta \Delta t \right) + \frac{1}{\beta} \dot{\Theta}_1(t_{n-1}) e^{-\alpha \Delta t} \sinh \beta \Delta t \quad (3.66)$$

$$\dot{\Theta}_1(t_n) = \frac{Y_{F1}(\frac{1}{2})}{\beta W_{AF1} S} \int_{t_{n-1}}^{t_n} F(\tau) e^{\alpha(\tau-t_n)} \left( \beta \cosh \beta(t_n - \tau) - \alpha \sinh \beta(t_n - \tau) \right) d\tau + \Theta_1(t_{n-1}) e^{-\alpha \Delta t} \left( \beta - \frac{\alpha^2}{\beta} \right) \sinh \beta \Delta t + \dot{\Theta}_1(t_{n-1}) e^{-\alpha \Delta t} \left( \cosh \beta \Delta t - \frac{\alpha}{\beta} \sinh \beta \Delta t \right) \quad (3.67)$$

ただし、

$$\alpha = \frac{W_{BF1}}{W_{AF1}} \quad (3.68)$$

$$\beta = \left( \frac{W_{BF1}^2}{W_{AF1}^2} - \omega^2 - \frac{W_{CF1}}{W_{AF1}} \right)^{\frac{1}{2}} \quad (3.69)$$

式(3.66)および(3.67)を時間 $t$ まで繰り返し計算し、その結果を式(3.57)および(3.58)に代入し、剛体運動を加えることにより試験片の応答が求められる。また、これらの結果を用いれば、式(3.56)により高速進展き裂の動的応力拡大係数が求められる。ただし、式(3.55)により係数 $k_m$ を求める際に変化するとき裂長さに対応した値を用いる必要がある。なお、高速き裂進展に対するこの方法は、回転慣性とせん断変形を無視した、より簡便な Euler-Bernoulli 梁モデルにも適用することができる。

### §3.4.3.2 有限要素法および Euler-Bernoulli 梁モデルとの比較

表 3.3 下段に示すセラミックス試験片に図 3.10 上図に示す衝撃力が作用し、き裂が $t = 10\mu s$ ( $\equiv t_0$ )から $v = 200\text{m/s}$ の一定速度で進展する場合について、本方法により求められる変位を有限要素法および Euler-Bernoulli 梁モデルによる結果と比較した。有限要素法では平面応力状態を仮定し、4 節点アイソパラメトリック要素を用いて、図 3.10 下図のように要素分割した。き裂の進展は節点力開放法<sup>[3.20]</sup>によって模擬した。有限要素解析には一様断面弾性体に対する高速解法を使用した。

Timoshenko 梁モデル、Euler-Bernoulli 梁モデル、および有限要素法による試験片(対称性から左半分)の変位の計算結果を、 $2.0\mu s$ ごとに $t = 20\mu s$ までプロットした結果を図 3.11 に示す。参考のためにき裂が進展しない場合の計算結果を図 3.12 に示す。Timoshenko 梁モデルの結果(一点鎖線)はき裂進展の有無によらず、有限要素解(実線)と良く一致していることがわかる。また、Timoshenko 梁モデル(一点鎖線)と Euler-Bernoulli 梁モデルによる結果(破線)の差異はこの解

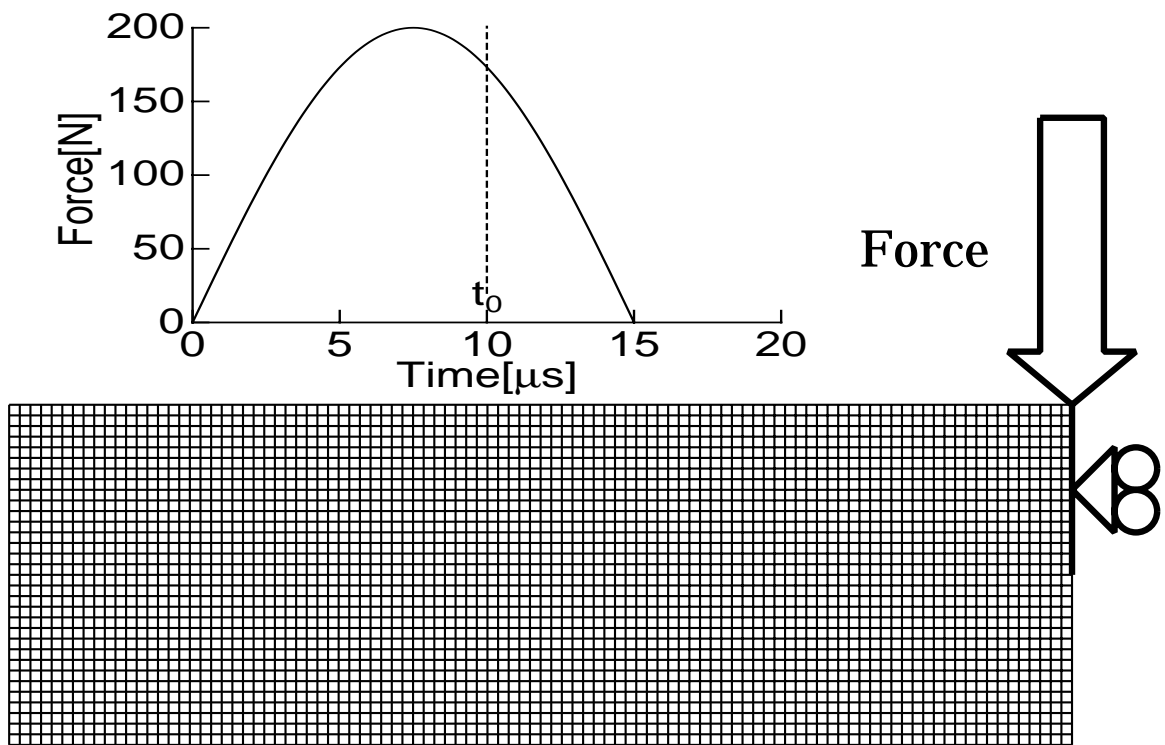


Fig. 3.10: Impact force and finite element mesh

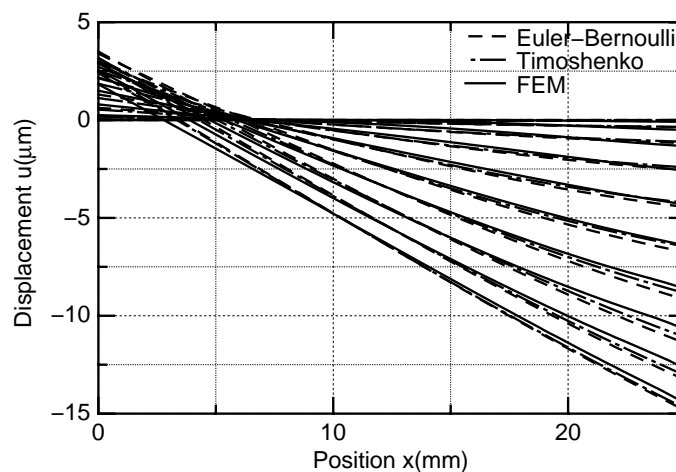


Fig. 3.11: Displacement of one-point-bend specimen at every  $2\mu\text{s}$  (with crack growth)

析条件ではあまり大きくないことがわかる．き裂進展開始後は，リガメント部の変形のために変位が試験片中央でシャープに変化することが良く模擬されている．なお，この解析をパーソナルコンピュータ計算した場合に要する時間は，Timoshenko 梁モデルでは数十秒，一様断面弾性体に対する有限要素高速解法では数分であった．

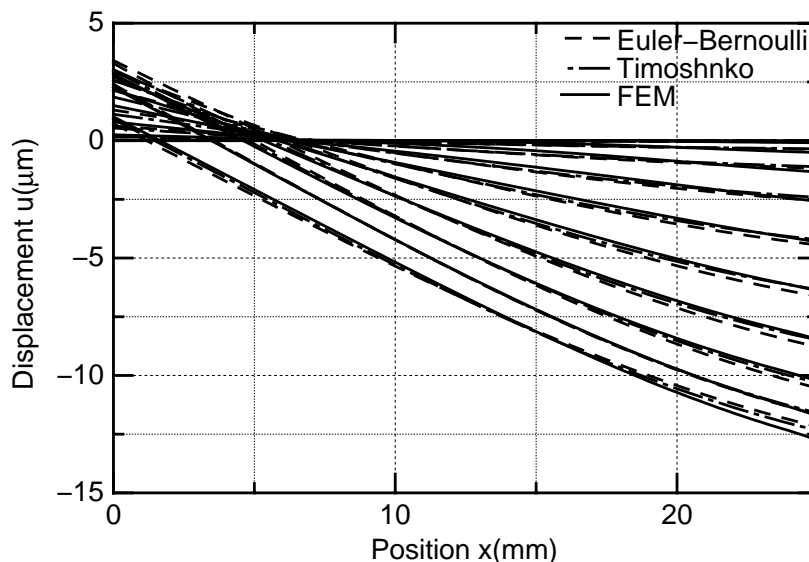


Fig. 3.12: Displacement of one-point-bend specimen at every  $2\mu s$  (without crack growth)

### §3.4.4 一点曲げ衝撃試験のシミュレーション

#### §3.4.4.1 衝撃力

本方法を用いて，図 3.13 および表 3.3 に示すような落下棒式一点曲げ衝撃試験の数値シミュレーションを行った．衝撃力は Hertz の接触理論に基づいて次の積分方程式<sup>[3,22]</sup>を数値的に解くことにより近似的に求めた．

$$F(t) = k_o(v_0 t - \eta_C(t) - \eta_B(t))^{\frac{3}{2}} \quad (3.70)$$

ただし，

$$\eta_B(t) = \int_0^t F(\tau) \frac{(t - \tau)}{M_B} d\tau + u\left(\frac{1}{2}, t\right) \quad (3.71)$$

$$\eta_C(t) = \int_0^t F(\tau) \frac{C_0}{A_C E_C} d\tau \quad (3.72)$$

$$k_o = \frac{4}{3} \frac{\sqrt{R}}{\frac{1-\nu_B^2}{E_B} + \frac{1-\nu_C^2}{E_C}} \quad (3.73)$$

ここで， $v_0$  は衝撃速度， $A_C, C_0, R, E$  および  $\nu$  はそれぞれ落下棒の断面積，落下棒中の縦波の伝播速度，落下棒下端の曲率半径，縦弾性係数およびポアソン比であり，下添字  $B$  および  $C$  はそれぞれ試験片および落下棒を表す．

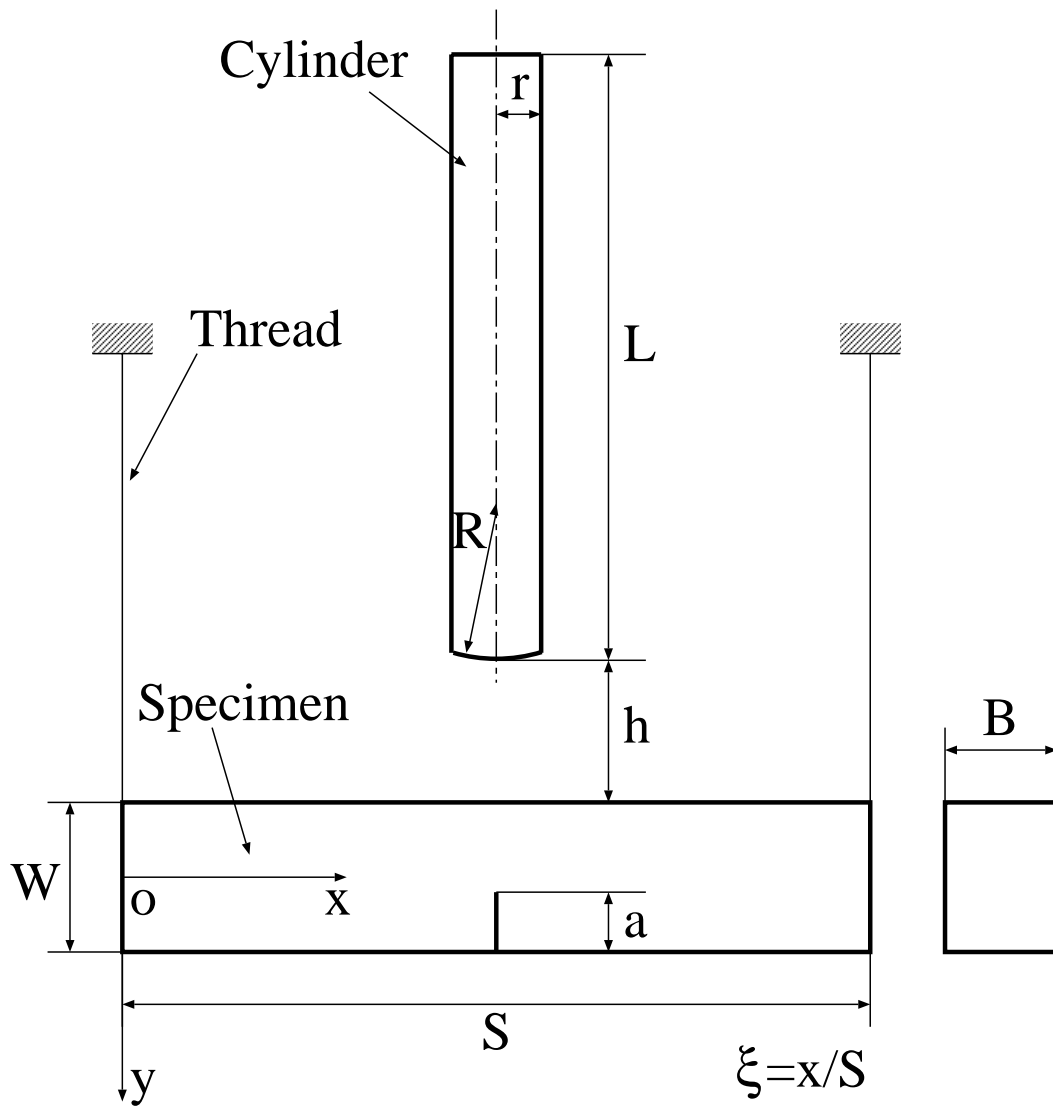


Fig. 3.13: Configurations of one-point-bend specimen and drop cylinder

Table 3.3: Material constants and dimensions of specimen

	$\rho(\text{g/cm}^3)$	$E(\text{GN/m}^2)$	$\nu$	Dimension(mm)
Cylinder	7.8	206	0.30	L=1500, r=3, R=100, h=50
Specimen	3.1	397	0.15	S=50, W=8, b=6, a=4

式 (3.71) の右辺第一項は試験片の剛体運動を表し，第2項の  $u_{(1/2),t}$  は衝撃点における試験片のたわみによる変位であり3章の方法により求められる．式 (3.72) の右辺は，衝撃点における落下棒の衝撃応答関数が  $C_0 H(t)/(A_C E_C)$  ( $H(t)$  はヘヴィサイド関数) であるので，これと  $F(t)$  のたたみこみ積分となっている．ただし，一次元波動理論を用い，衝撃持続時間が応力波が落下棒を往復する時間より短いので無限長棒の衝撃応答関数を用いている．

図 3.14 はき裂進展がない時の衝撃力  $F(t)$  を  $m = 1$  までと  $m = 5$  まで考慮して計算した結果である．両者に大きな差がないことがわかる．

図 3.15 は，種々の進展開始時刻  $t_0 = 5, 10, 15\mu\text{s}$  から，速度  $v = 200\text{m/s}$  でき裂が進展した場合の  $F(t)$  を示す．き裂進展の影響が  $F(t)$  にき裂進展直後にはほとんど現れず，徐々に現れることがわかる．

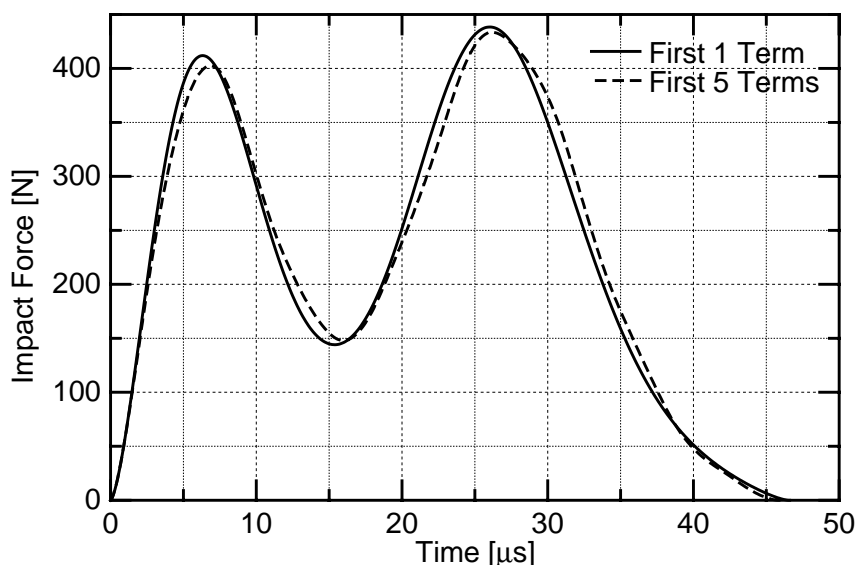


Fig. 3.14: Impact force calculated by taking the first 1 or 5 terms (Timoshenko beam model without crack growth)

#### §3.4.4.2 一点曲げ衝撃試験の改良・拡張に対する検討

何らかの方法でき裂進展開始時刻  $t_0$  を検出できれば，式 (3.56) から衝撃破壊靱性が求められる．試験片毎にき裂先端近傍にひずみゲージを貼ることにより  $t_0$  を検出することが考えられる．しかし，高温衝撃破壊靱性を測定する場合，試験片を高温炉の中に入れるので，き裂進展開始を光学

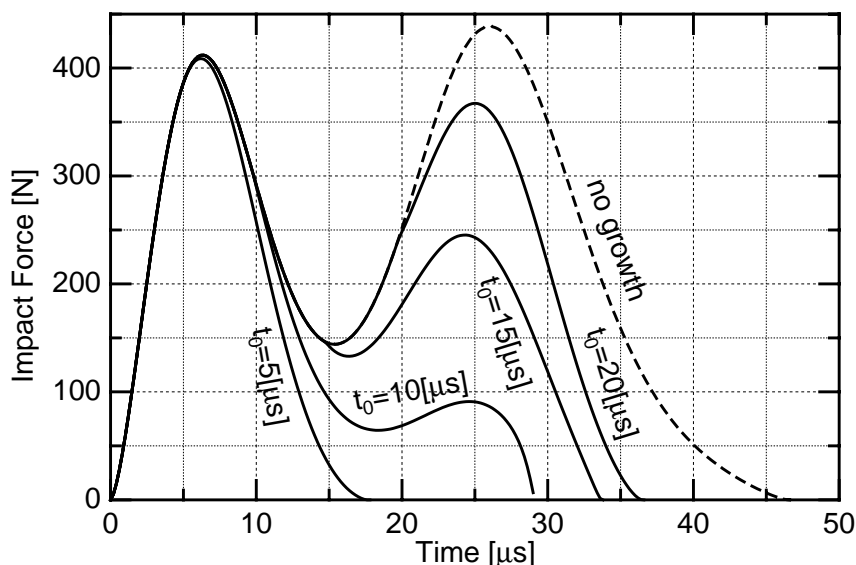


Fig. 3.15: Impact force calculated by Timoshenko beam model with or without crack growth

的測定装置や試験片に貼ったひずみゲージにより検出することができない．そこで，複数の試験片に種々の衝撃力を与える<sup>[3.14]</sup>か，一本の試験片に次第に増加する衝撃力を数回加えて<sup>[3.12]</sup>，試験片が破壊した時の衝撃力から衝撃破壊靱性を求めている．

一本の一点曲げ衝撃試験片に一回の衝撃力を加えて破壊させ，衝撃力  $F(t)$  の時間的变化を（落下棒等の）衝撃棒の炉外の部分に貼ったひずみゲージにより測定し，そのデータから逆解析により，き裂進展開始時刻を推定出来れば好都合である．しかし，図 3.15 のシミュレーション結果から， $F(t)$  はき裂進展直後では，き裂進展のない場合とほとんど同じように推移しているため，この逆問題は極めて悪条件であることがわかる．したがって， $t_0$  を精度よく推定するためには，何か有効な補助情報が必要であることが推察される．

次に，高速き裂進展脆性の計測について考える．前節までの議論により，もし，衝撃速度  $v_0$  およびき裂長さの時間変化  $a(t)$  が測定できれば，衝撃力  $F(t)$  が計算でき，この  $a(t)$  と  $F(t)$  から高速き裂の動的応力拡大係数  $K_I(t)$  が求められる．き裂進展速度  $v$  が  $a(t)$  から求められるので，その  $v$  に対する高速き裂進展脆性  $K_{ID}$  が計測できることが分かる．衝撃速度  $v_0$  の代わりに衝撃力  $F(t)$  を直接測定しても，同様である．

しかし，この方法による高速き裂進展脆性の計測を実用化するためには，未だ多くの研究が必要である．まず，本方法による  $F(t)$  や  $K_I(t)$  の計算結果を実測値や他の数値解法（有限要素法，境界要素法など）による解と比較することにより，計算精度のチェックを行わなければならない．次に，実際に  $K_{ID}$  を測定し，他の方法による計測結果と比較することも必要であろう．また現状では一様断面弾性体に対する有限要素高速解法は Timoshenko 梁理論による方法に比べて計算速度は遅い．しかし通常の有限要素法と比較すれば格段に高速になっており，有限要素高速解法を用いた繰り返し計算によってより高精度な衝撃破壊靱性の逆解析を行うことも実用上可能となって来ていると考えられる．以上は今後の課題としたい．

## §3.5 結言

一様断面 (断面形状は任意) を持つ弾性体の動的有限要素法解析を効率化する手法を提案した。まず、弾性体を母線に沿ってスライス状に等分割し、各スライスに同一の要素分割を行う。この要素分割法によって、陰解法による有限要素方程式のマトリックスは5つの小マトリックスの繰り返しからなるブロック3重対角マトリックスとなる。弾性体全体に対する大きなマトリックスを直接取り扱わないので、計算時間と記憶容量の節約を図ることが可能となった。

さらにこの繰り返しからなるブロック3重対角マトリックスに対してガウスの前進消去を行うと漸化的に定義されたマトリックス列が現れるが、そのマトリックス列が収束列であることが Guo らによって示されている。よって漸化的な計算は途中で打ち切ることができ、計算に必要な記憶容量はより一層節約することが可能になった。

幾つかの例題を代表的な陰解法である Newmark $\beta$ 法を用いて解析した。さらに本手法と通常の有限要素法とで計算時間および記憶容量の比較を行い、本手法の有効性を示した。

近年では部材の組み合わせからなる構造物を領域分割法により並列化してより高速に解く方法も盛んに用いられるようになって来ている。また、一様断面弾性体を組み合わせて作られた構造物は非常に多く、本手法と領域分割法とを組み合わせるといけば広範囲に利用可能であると考えられる。

本手法は動的弾性問題ばかりでなく、幾何学的非線形を含まない動的弾塑性問題にも利用することが可能である。さらに、非定常熱伝導や動電磁場の問題をなど様々な問題に利用できると考えられる。

## 参考文献

- [ 3.1 ] Yun JH, “Parallel performance of block ILU preconditioners for a block-tridiagonal matrix”, *Journal of Supercomputing*, 2003; **24** (1): 69–89
- [ 3.2 ] Rossi T, Toivanen J, “A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary” *SIAM Journal on Scientific Computing*, 1999; **20** (5): 1778–1796
- [ 3.3 ] Hughes TJR, “The Finite Element Method: Linear Static and Dynamic Finite Element Analysis” *Dover Publications*, 2000; 490–513
- [ 3.4 ] Guo CH, Lancaster P, “Iterative Solution of Two Matrix Equations”, *Mathematics of Computation*, 1999; **68** (228): 1589–1603
- [ 3.5 ] Gong JC, Malvern LE, Jenkins DA, “Dispersion Investigation in the Split Hopkinson Pressure Bar” *Journal of Engineering Materials and Technology*, 1990; **112**: 309–314
- [ 3.6 ] Kolsky H, “Stress Waves in Solids”, *Dover Publications*, 1963; 87–91
- [ 3.7 ] Hibbitt, Karlsson, Sorensen, “ABAQUS/Standard Users’ Manual Version 6.3”, *HKS Inc.*, 2002;
- [ 3.8 ] Lawson CL et.al., “Basic Linear Algebra Subprograms for FORTRAN usage”, *ACM Transactions on Mathematical Software*, 1979; **5** (3): 308–323
- [ 3.9 ] Anderson E, “LAPACK Users’ Guide Third Edition”, *Society for Industrial and Applied Mathematics*, 1999;
- [ 3.10 ] Kalthoff, J.K., *Int.J.Fract.*, 27(1985)227-298
- [ 3.11 ] Giovanola, J.H., *ASTM STP*, 905(1986)307-328
- [ 3.12 ] 坂田・ほか4名, 材料, 37 – 419(1988)910-915
- [ 3.13 ] 藤野・ほか3名, 機論, 54A – 501(1988)1101-1106
- [ 3.14 ] 黒田・ほか3名, 機論, 50A – 453(1984)988-995
- [ 3.15 ] 高橋・青木, 機論, 57A – 537(1991)1073-1078
- [ 3.16 ] Nash, G.E., *Int.J.Fract.*, 5(1969)269-286
- [ 3.17 ] 森口・ほか2名, 数学公式 II, (1957) 岩波
- [ 3.18 ] Huang, T.C., *J.Appl.Mech.*, 12(1961)579-584
- [ 3.19 ] Tada, H. et al., *The Stress Analysis of Cracks Handbook*, (1973) Del Research Corpora-

tion

- [ 3.20 ] 岸本・ほか3名, 機論, 52A – 474(1986)440-448
- [ 3.21 ] 青木・ほか3名, 機論, 67A – 662(2001)1583-1588
- [ 3.22 ] Goldsmith,W., *Impact*, (1960)Edward Arnold
- [ 3.23 ] 井上裕嗣, 青木繁, 岸本喜久雄, “衝撃問題における逆解析” 日本機械学会論文集, A編, 63-616, (1997-12), pp.2478–2484
- [ 3.24 ] 矢川元基, 吉村忍, “10章 時間依存型問題の解析”, 有限要素法, 培風館, (1991), p.173–176
- [ 3.25 ] 大坪英巨, 久保田晃弘, “2章 流体–構造相関問題”, 相関問題, 培風館, (1991), p.56–67
- [ 3.26 ] 鷺津久一郎, 宮本博, 山田嘉昭, 山本善之, 川井忠彦, “4章 動的問題”, 有限要素法ハンドブック II 応用編, 培風館, (1983), p.3–64
- [ 3.27 ] 青木繁, 天谷賢治, 矢作徹夫, “セラミックス中のき裂面に働く接触力の同定法”, 日本機械学会論文集, A編, 64-621, (1998-5), pp.46–52
- [ 3.28 ] 鷺津久一郎, 宮本博, 山田嘉昭, 山本善之, 川井忠彦, “3章 各種有限要素法”, 有限要素法ハンドブック I 基礎編, 培風館, (1981), p.363–450

## 第4章 ポテンシャル問題に対する注目領域解の境界要素高速解法

Onishi Y, Amaya K, “A new technique for high-speed boundary element analyses of Laplace equations to obtain solutions at target regions”, *Engineering Analysis with Boundary Elements*, 28-7, (2004), pp.791-799 より出典

### §4.1 緒言

微分方程式の境界値問題の解法として境界要素法が有効であることは広く知られている。さらに、境界要素法の高速解法として多重極展開法<sup>[4.1][4.2][4.3]</sup>があり、大規模解析の効率的な手法として頻繁に用いられている。

しかし、実際の問題において、領域内すべての場所での解ではなく、ある注目領域の解だけが必要であるという場合は少なくない。例えば、定常熱伝導の問題において領域内の最高温度を求めたいが、最高温度をとる場所はあらかじめ予想が付いている、といった場合である。このような問題に対し多重極展開法を用いると、通常境界要素法<sup>[4.4]</sup>に比べるとはるかに高速に計算することが出来るが、注目領域と非注目領域の解とを同精度で求めてしまうことに変わりはない。従って、注目領域の解のみを正確に、且つより高速に求める手法があれば非常に有用であると考えられるが、今日までにそのような研究はなされていない。

そこで、本研究では2次元 Laplace 方程式の境界値問題に対し、境界要素法を用いて注目領域の解だけを高速に求める方法を開発した。本手法では、Laplace 領域において注目境界から十分な距離が離れた非注目境界での境界変数の高周波成分が注目境界の解に影響を与えないことに着目、利用している。

本論文では、まず本手法の概要について説明した後、高速解法の数学的詳細について記述する。さらに、具体的な例題を用いて本手法の有効性を示す。

### §4.2 手法概要

Laplace 方程式で支配されている2次元領域について考える。例えばすべての境界で Dirichlet 境界条件、すなわちポテンシャルが与えられると、領域内のポテンシャルの場は一意に決まる。この時、この境界条件に少し変更を加え、境界の一部に空間的に高周波なポテンシャルの揺らぎを足し合わせたとする。当然、場の状態は別の状態へと変化するが、変更を加えた境界から十分に離れた領域においては、場の状態はほとんど変化しない。

例えば図4.1に示すようなドーナツ形状の2次元 Laplace 領域を考える．内周の境界を注目境界，外周の境界を非注目境界とし，ポテンシャルを  $\phi$ ，境界上のフラックスの外向き法線成分を  $i$  とおく．

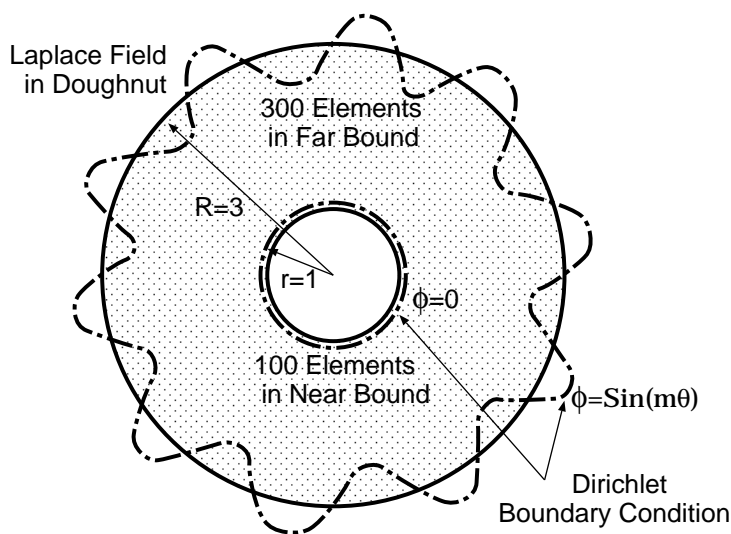


Fig. 4.1: Doughnut model for numerical experiment

すべて Dirichlet 境界とし，内周の境界では常に  $\phi = 0$ ，外周の境界では三角関数状の境界条件  $\phi = \sin(m\theta)$  (ただし  $m$  はモードを表す自然数) を与えるものとする (図4.2 参照) 求める変数は内周の  $i$  であり， $m = 1, 2, 4, 8$  と変化させた時の内周における  $i$  の値を図4.3に示す．

$m$  が大きくなるにつれ，内円の  $i$  の値は0に近づいていくことが分かる．つまり，遠くの境界の境界条件からの影響はその周波数が十分に高ければ無視できるようになると言える．

すなわち，注目領域からある程度の距離が離れた領域で与えられた境界条件に高周波な擾乱があったとしても，その擾乱が注目領域の解に与える影響は小さなものとなる (図4.4 参照) 故に，注目領域が境界に近い場合，その近傍にある境界要素の解を正確に，それ以外の解を荒く求めれば，注目領域の解は充分正確に求めることが出来る．

そこで，境界要素を注目領域に近い要素と遠い要素の2種類に分離して扱うこととする (以後，この2種類の境界を単に近傍境界，遠方境界と呼ぶことにする)．先に述べた通り，遠方境界要素の境界変数に空間的な振動があった場合，その高周波成分は近傍境界にほとんど影響を与えず，低周波成分のみが近傍境界に影響を与える．従って，近傍境界の解だけを求めたいのであれば，遠方境界の解空間は低周波関数の重ね合わせのみで表現するだけで充分である (図4.5 参照)<sup>(i)</sup>

本手法では，まず遠方境界の境界変数を Fourier 級数展開で表現し，次にその Fourier 係数と近傍の境界変数を用いた境界積分方程式を立てる．Fourier 級数展開を低次で打ち切れば，その式は遠方を荒く，近傍を細かく表現した式となり，これを解くことによって遠方の概略的な解と近傍の詳細な解を求めることが可能となる．遠方は通常境界要素法に比べて非常に少ない個数の変数で記述される為，解くべき方程式の次元が下がり，従って高速に近傍の解を求めることが出来る (図4.6 参照)

<sup>(i)</sup>多重極展開法は遠方から近傍への「影響」を低周波成分のみで表現する近似解法である．

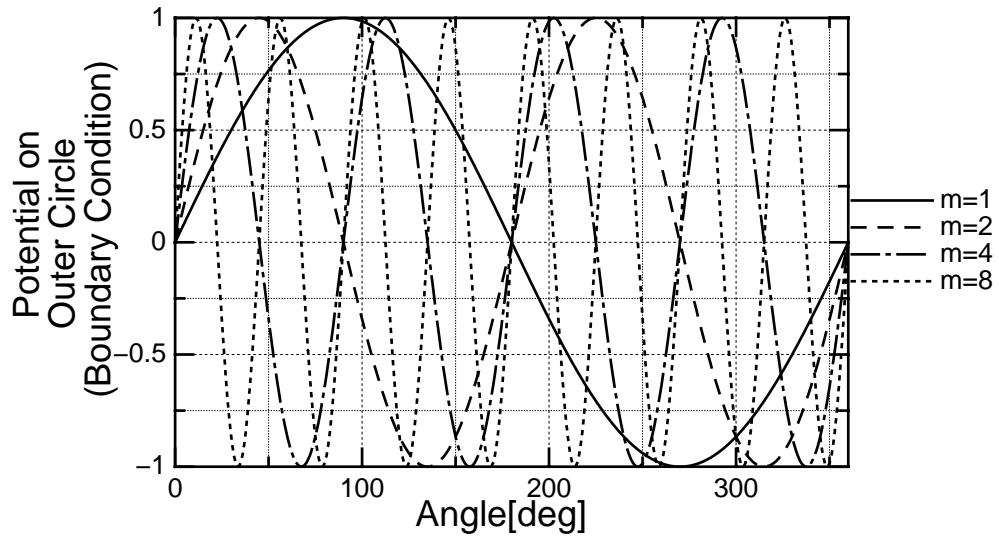


Fig. 4.2: Boundary condition in non-target bound varied by frequency

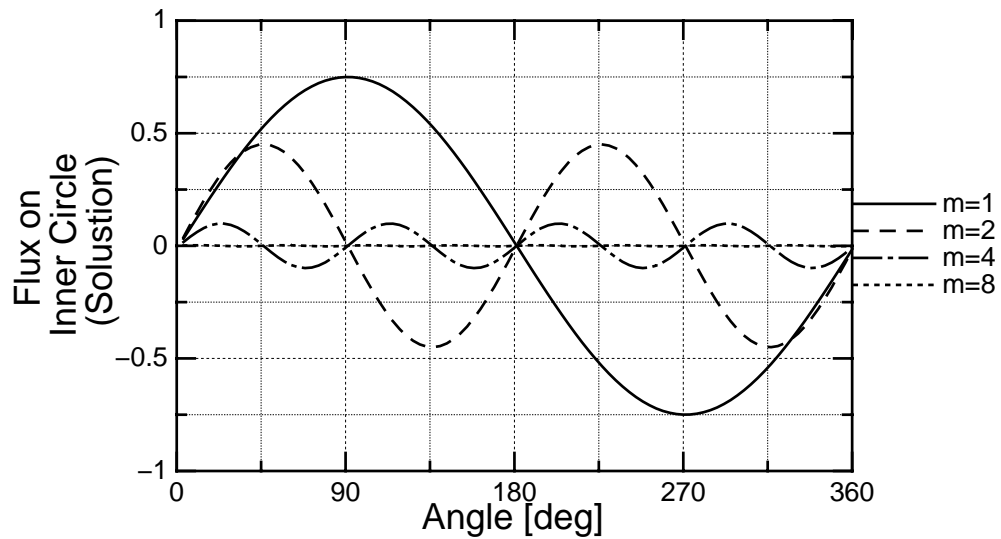


Fig. 4.3: Flux in target bound infected by boundary condition in non-target bound varied by frequency

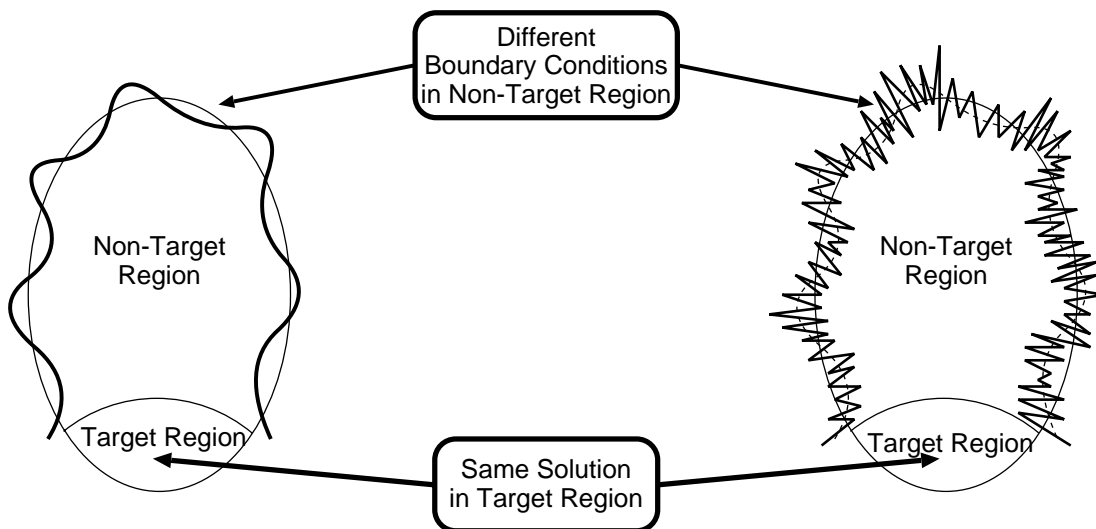


Fig. 4.4: Little influences for target region from high frequency disturbance of boundary conditions in non-target bound

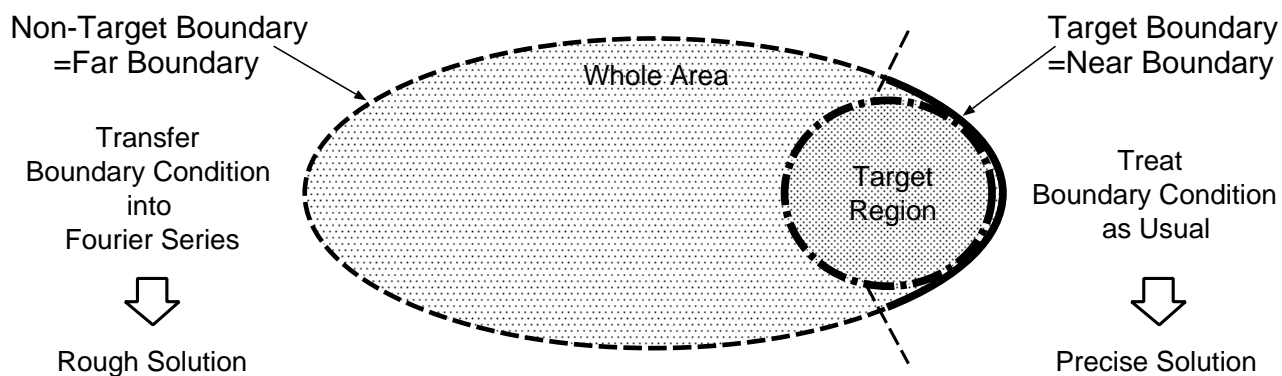


Fig. 4.5: Division of bound into near and far bounds

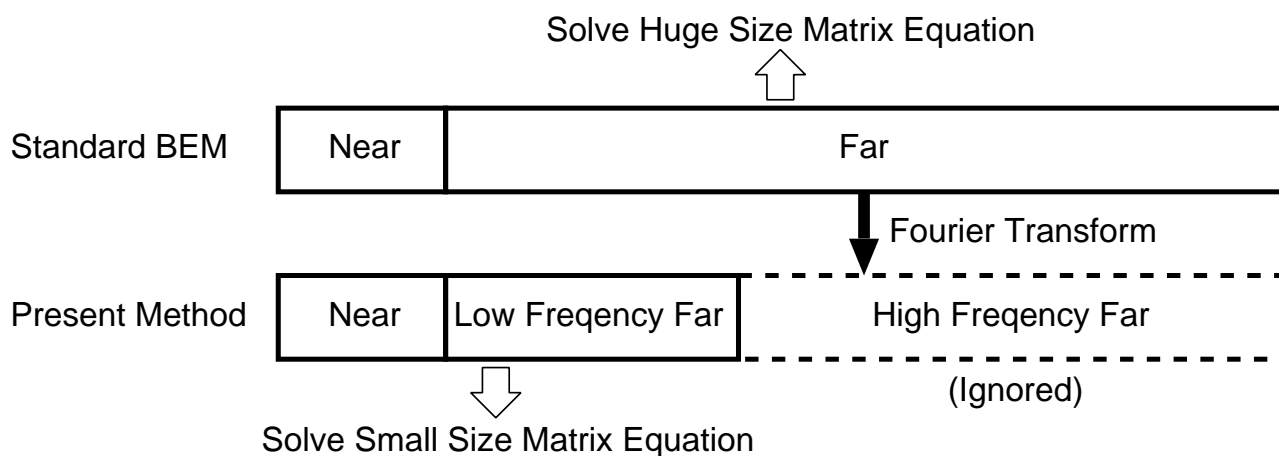


Fig. 4.6: Reduction of number of parameters

### §4.3 高速解法

解析モデルとして、図4.7に示すような領域及び境界を考える。領域内ではLaplace方程式  $\nabla^2\phi = 0$  を満たし、ポテンシャルとフラックスは  $\nabla\phi = \vec{i}$  の関係で結ばれているものとする。

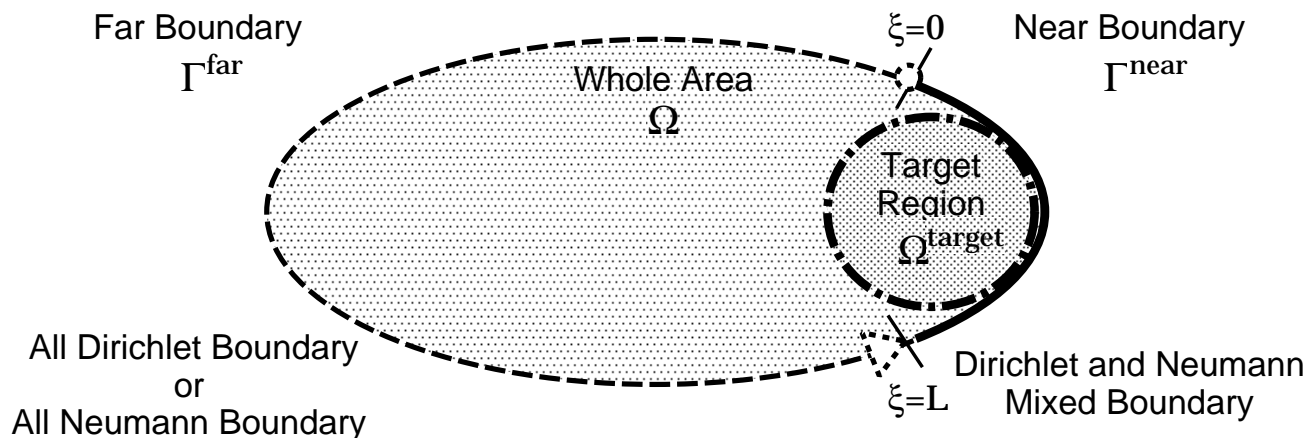


Fig. 4.7: Model of analysis

#### §4.3.1 遠方境界の Fourier 級数展開

遠方境界上の点を  $x$  と置く。まず、遠方境界の一端を始点に取り、境界に沿ってもう一端の終点へと続く長さ  $L$  の経路上に曲線座標  $\xi$  を定める。次に、その区間  $\xi = 0 \sim L$  で定義された関数  $\phi(x(\xi))$  (ポテンシャル) 及び  $i(x(\xi))$  (フラックスの外向き方向成分) を Fourier 級数展開することにより次式が得られる。<sup>[4.5]</sup>

$$\phi(x(\xi)) = \text{Re} \left[ \sum_{k=0}^{\infty} \Phi_k e^{i\frac{2\pi k}{L}\xi} \right] \quad (4.1)$$

$$i(x(\xi)) = \text{Re} \left[ \sum_{k=0}^{\infty} I_k e^{i\frac{2\pi k}{L}\xi} \right] \quad (4.2)$$

ここで、 $e$  は自然対数の底、 $i$  は虚数、 $\Phi_k$  及び  $I_k$  は複素数の Fourier 係数であり、Fourier 級数の基底の直交性から、

$$\Phi_0 = \frac{1}{L} \int_0^L \phi(x(\xi)) d\xi, \quad \Phi_k = \frac{2}{L} \int_0^L \phi(x(\xi)) e^{-i\frac{2\pi k}{L}\xi} d\xi \quad [\text{for } k \text{ in } 1 \sim \infty] \quad (4.3)$$

$$I_0 = \frac{1}{L} \int_0^L i(x(\xi)) d\xi, \quad I_k = \frac{2}{L} \int_0^L i(x(\xi)) e^{-i\frac{2\pi k}{L}\xi} d\xi \quad [\text{for } k \text{ in } 1 \sim \infty] \quad (4.4)$$

で表される。(ii)

<sup>(ii)</sup>  $\Phi_0, I_0$  も計算上複素数として扱うが、実際にはこれらの虚部は常にゼロである。

### §4.3.2 境界積分方程式の導出

本手法では、大きく分けて2種類の境界積分方程式を用いる。すなわち、2種類の重み関数を用いて直接法による境界積分方程式の定式化を行う。

1種類の積分方程式は、重み関数として近傍要素の節点に基本解を1つづ置いた式で、通常の境界要素法で用いる境界積分方程式と基本的には同じである(図4.8参照)<sup>(iii)</sup>

2種類の積分方程式は、遠方に基本解を連続して置き、その湧き出し量(高さ)が場所によって波打った山脈状の重み関数を置いた式である(図4.9参照)<sup>(iv)</sup>

なお、この小節では離散化を行わない為、すべて近似を含まない解析的な式のみで議論を進める(離散化による近似は次小節で行う。)

#### §4.3.2.1 境界積分方程式 1

2次元 Laplace 方程式の基本解を  $\phi^*$ 、第2基本解を  $i^*$  とする。重み関数として、近傍境界上の点  $y$  に置いた基本解  $\phi^*(y, x)$ <sup>(v)</sup> を用いて重み付き残差法を行うと次式が得られる。

$$c(y)\phi(y) = \int_{\Gamma} i(x)\phi^*(y, x) - \phi(x)i^*(y, x)d\Gamma_x \quad (4.5)$$

遠近分離

$$= \int_{\Gamma^{\text{near}}} i(x)\phi^*(y, x) - \phi(x)i^*(y, x)d\Gamma_x + \int_{\Gamma^{\text{far}}} i(x)\phi^*(y, x) - \phi(x)i^*(y, x)d\Gamma_x \quad (4.6)$$

<sup>(iii)</sup>実際の基本解は特異性を持つため極値は無限大に発散するが、上図では極値の山を有限に抑えて描画してある。

<sup>(iv)</sup>Weight function 1 同様、極値の山を有限に抑えて描画してある。

<sup>(v)</sup> $\phi^*(y, x)$  の物理的意味は相反定理から2通りの解釈が可能であるが、本論文では「点  $y$  に置いた湧き出しが点  $x$  に与える影響」を意味するものとする。

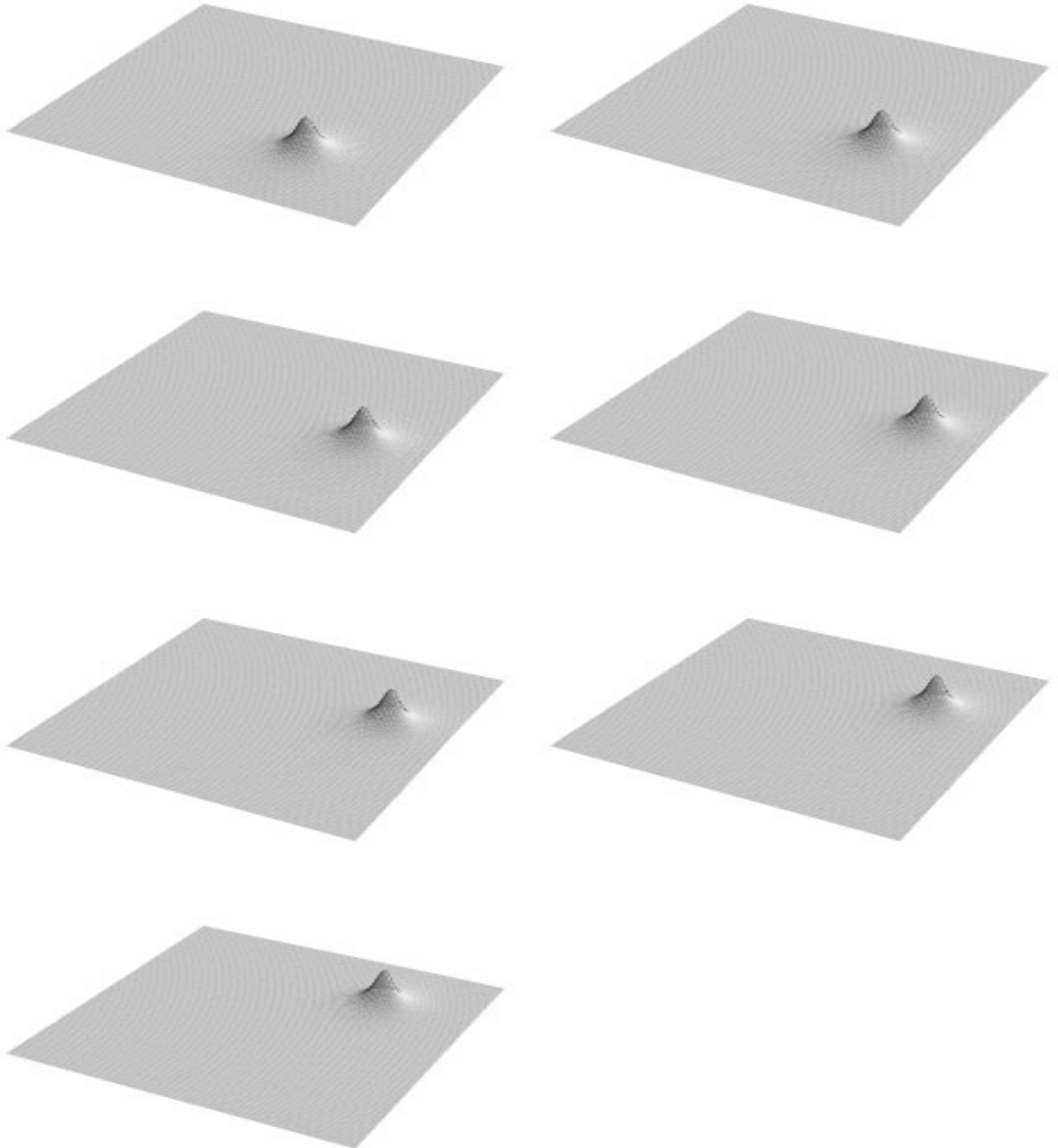


Fig. 4.8: Image of Weight function 1 ( fundamental solution at each node of near element )

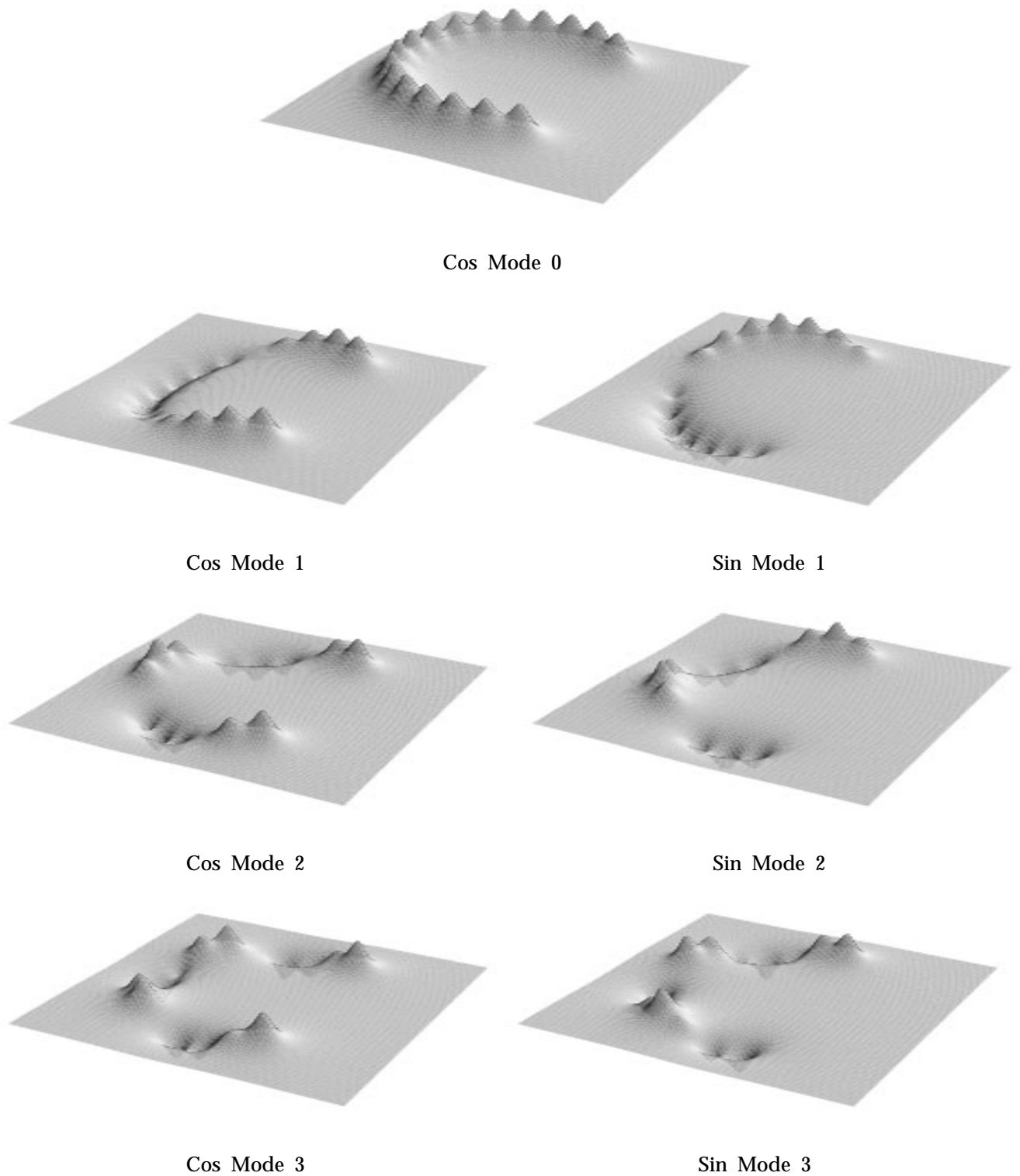


Fig. 4.9: Weight function 2 ( trigonometrically disposed fundamental solutions at each node of far element )

ここで，遠方の  $\phi(x)$  及び  $i(x)$  に式 (4.1),(4.2) を代入すると，遠方  $\Gamma^{\text{far}}$  の境界積分項は，

$$\int_{\Gamma^{\text{far}}} i(x)\phi^*(y, x) - \phi(x)i^*(y, x)d\Gamma_x$$

式 (4.1), (4.2) を代入

$$= \int_{\Gamma^{\text{far}}} \text{Re} \left[ \sum_{k=0}^{\infty} I_k e^{i\frac{2\pi k}{L}\xi} \right] \phi^*(y, x) - \text{Re} \left[ \sum_{k=0}^{\infty} \Phi_k e^{i\frac{2\pi k}{L}\xi} \right] i^*(y, x) d\Gamma_x \quad (4.7)$$

積分と総和の順序入れ替え

$$= \sum_{k=0}^{\infty} \text{Re} \left[ I_k \int_{\Gamma^{\text{far}}} e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x) d\Gamma_x - \Phi_k \int_{\Gamma^{\text{far}}} e^{i\frac{2\pi k}{L}\xi} i^*(y, x) d\Gamma_x \right] \quad (4.8)$$

$x$  から  $\xi$  に変数変換

$$= \sum_{k=0}^{\infty} \text{Re} \left[ I_k \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x(\xi)) d\xi - \Phi_k \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y, x(\xi)) d\xi \right] \quad (4.9)$$

$\text{Re}[\dots]$  の分解

$$= \sum_{k=0}^{\infty} \text{Re}[I_k] \text{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x(\xi)) d\xi \right] - \sum_{k=0}^{\infty} \text{Im}[I_k] \text{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x(\xi)) d\xi \right]$$

$$- \sum_{k=0}^{\infty} \text{Re}[\Phi_k] \text{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y, x(\xi)) d\xi \right] + \sum_{k=0}^{\infty} \text{Im}[\Phi_k] \text{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y, x(\xi)) d\xi \right] \quad (4.10)$$

と書き換えることが出来る．

結局，上記の書き換えの後得られる積分方程式，すなわち積分方程式 1 は次式となる．

$$c(y)\phi(y) = \int_{\Gamma^{\text{near}}} i(x)\phi^*(y, x) - \phi(x)i^*(y, x)d\Gamma_x$$

$$+ \sum_{k=0}^{\infty} \text{Re}[I_k] \text{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x(\xi)) d\xi \right] - \sum_{k=0}^{\infty} \text{Im}[I_k] \text{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y, x(\xi)) d\xi \right]$$

$$- \sum_{k=0}^{\infty} \text{Re}[\Phi_k] \text{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y, x(\xi)) d\xi \right] + \sum_{k=0}^{\infty} \text{Im}[\Phi_k] \text{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y, x(\xi)) d\xi \right] \quad (4.11)$$

#### §4.3.2.2 境界積分方程式 2

重み関数を，遠方境界上に連続して基本解を置いた山脈上の複素数関数  $W_K(x)$ ，すなわち，

$$W_K(x) = \int_{\Gamma^{\text{far}}} \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\Gamma_y = \int_0^L \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\eta \quad (4.12)$$

として重み付き残差法を行うと次式が得られる。(vi)

$$\int_{\Omega} \phi(x) \nabla_x^2 W_K(x) d\Omega_x = \int_{\Gamma} i(x) W_K(x) - \phi(x) \frac{\partial W_K(x)}{\partial n(x)} d\Gamma_x \quad (4.13)$$

遠近分離

$$\begin{aligned} &= \int_{\Gamma^{\text{near}}} i(x) W_K(x) - \phi(x) \frac{\partial W_K(x)}{\partial n(x)} d\Gamma_x \\ &+ \int_{\Gamma^{\text{far}}} i(x) W_K(x) - \phi(x) \frac{\partial W_K(x)}{\partial n(x)} d\Gamma_x \end{aligned} \quad (4.14)$$

ただし,  $\frac{\partial W_K(x)}{\partial n(x)}$  は,

$$\begin{aligned} \frac{\partial W_K(x)}{\partial n(x)} &= \frac{\partial}{\partial n(x)} \int_0^L \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\eta \\ &= \int_0^L \frac{\partial}{\partial n(x)} \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\eta \\ &= \int_0^L i^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\eta \end{aligned} \quad (4.15)$$

と書き表せる.

まず, 左辺の領域積分  $\Omega$  の項を整理すると,

$$\begin{aligned} &\int_{\Omega} \phi(x) \nabla_x^2 W_K(x) d\Omega_x \\ &\text{式 (4.12) の代入と被積分関数の移動} \\ &= \int_{\Omega} \int_0^L \phi(x) \nabla_x^2 \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\eta d\Omega_x \end{aligned} \quad (4.16)$$

積分順序の入れ替え

$$= \int_0^L \left( \int_{\Omega} \phi(x) \delta(y) d\Omega_x \right) e^{i\frac{2\pi K}{L}\eta} d\eta \quad (4.17)$$

主値積分結果の代入

$$= \int_0^L c(y(\eta)) \phi(y(\eta)) e^{i\frac{2\pi K}{L}\eta} d\eta \quad (4.18)$$

となる.

<sup>(vi)</sup>  $W_K(x)$  の実部が cos 状, 虚部が sin 状の山脈を表している.

次に，遠方の  $\phi(x)$  及び  $i(x)$  に式 (4.1),(4.2) を代入すると，右辺遠方  $\Gamma^{\text{far}}$  の境界積分項は，

$$\int_{\Gamma^{\text{far}}} i(x)W_K(x) - \phi(x)\frac{\partial W_K(x)}{\partial n(x)}d\Gamma_x$$

式 (4.1), (4.2) の代入

$$= \int_0^L \operatorname{Re}\left[\sum_{k=0}^{\infty} I_k e^{i\frac{2\pi k}{L}\xi}\right]W_K(x)d\xi - \int_0^L \operatorname{Re}\left[\sum_{k=0}^{\infty} \Phi_k e^{i\frac{2\pi k}{L}\xi}\right]\frac{\partial W_K(x)}{\partial n(x)}d\xi \quad (4.19)$$

$\operatorname{Re}[\dots]$  の分解

$$= \int_0^L \sum_{k=0}^{\infty} \left( \operatorname{Re}[I_k]\operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right] - \operatorname{Im}[I_k]\operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right] \right)W_K(x)d\xi$$

$$- \int_0^L \sum_{k=0}^{\infty} \left( \operatorname{Re}[\Phi_k]\operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right] - \operatorname{Im}[\Phi_k]\operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right] \right)\frac{\partial W_K(x)}{\partial n(x)}d\xi \quad (4.20)$$

積分と総和の順序入れ替え

$$= \sum_{k=0}^{\infty} \operatorname{Re}[I_k] \left( \int_0^L \operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right]W_K(x)d\xi \right) - \sum_{k=0}^{\infty} \operatorname{Im}[I_k] \left( \int_0^L \operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right]W_K(x)d\xi \right)$$

$$- \sum_{k=0}^{\infty} \operatorname{Re}[\Phi_k] \left( \int_0^L \operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right]\frac{\partial W_K(x)}{\partial n(x)}d\xi \right) + \sum_{k=0}^{\infty} \operatorname{Im}[\Phi_k] \left( \int_0^L \operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right]\frac{\partial W_K(x)}{\partial n(x)}d\xi \right) \quad (4.21)$$

と書き換えることが出来る．

結局，上記の書き換えの後得られる積分方程式，すなわち積分方程式 2 は次式となる．

$$\int_0^L c(y(\eta))\phi(y(\eta))e^{i\frac{2\pi K}{L}\eta}d\eta$$

$$= \int_{\Gamma^{\text{near}}} i(x)W_K(x) - \phi(x)\frac{\partial W_K(x)}{\partial n(x)}d\Gamma_x$$

$$+ \sum_{k=0}^{\infty} \operatorname{Re}[I_k] \left( \int_0^L \operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right]W_K(x)d\xi \right) - \sum_{k=0}^{\infty} \operatorname{Im}[I_k] \left( \int_0^L \operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right]W_K(x)d\xi \right)$$

$$- \sum_{k=0}^{\infty} \operatorname{Re}[\Phi_k] \left( \int_0^L \operatorname{Re}\left[e^{i\frac{2\pi k}{L}\xi}\right]\frac{\partial W_K(x)}{\partial n(x)}d\xi \right) + \sum_{k=0}^{\infty} \operatorname{Im}[\Phi_k] \left( \int_0^L \operatorname{Im}\left[e^{i\frac{2\pi k}{L}\xi}\right]\frac{\partial W_K(x)}{\partial n(x)}d\xi \right) \quad (4.22)$$

### §4.3.3 離散化とマトリックス方程式

簡単のため，境界要素はすべて一定要素を使い，要素積分の際の三角関数の項は要素内で一定として扱うこととする．<sup>(vii)</sup>

近傍を  $n$  個，遠方を  $m$  個の要素で分割したものとし，遠方の Fourier 級数展開は低次の  $l$  項を考慮するものとする．境界変数は，近傍境界上の節点のポテンシャル  $\phi_j$  [ $j = 0 \sim n-1$ ] とフラックス外向き法線方向成分  $i_j$  [ $j = 0 \sim n-1$ ]，及び式 (4.3),(4.4) により与えられた遠方のフーリエ

<sup>(vii)</sup>もっと厳密に計算，特に積分計算を行うことも可能であるが，計算式が非常に煩雑になるだけでなく，計算コストもかかってしまうこととなる．研究の本義である高速化のメリットを損なわないため，この様な離散近似を行った．

係数  $\Phi_k, I_k$  [ $k = 0 \sim l - 1$ ] である．境界条件が与えられているので，境界変数の半分は既知数であり，残る半分が未知数である．

#### §4.3.3.1 離散化された Fourier 級数展開

式 (4.3),(4.4) で表される Fourier 係数は，積分内の  $\phi, i$  が各要素ごとに一定，三角関数の項は要素の midpoint での値で一定とすると，要素積分の外に出すことができ，

$$\Phi_0 = \frac{1}{L} \int_0^L \phi(x(\xi)) d\xi = \frac{1}{L} \sum_{p=0}^{m-1} \phi_p^{\text{far}} L_p \quad (4.23)$$

$$\Phi_k = \frac{2}{L} \int_0^L \phi(x(\xi)) e^{-i\frac{2\pi k}{L}\xi} d\xi = \frac{2}{L} \sum_{p=0}^{m-1} \phi_p^{\text{far}} e^{-i\frac{2\pi k}{L}\xi_p} L_p \quad [\text{for } k \text{ in } 1 \sim l - 1] \quad (4.24)$$

$$I_0 = \frac{1}{L} \int_0^L i(x(\xi)) d\xi = \frac{1}{L} \sum_{p=0}^{m-1} i_p^{\text{far}} L_p \quad (4.25)$$

$$I_k = \frac{2}{L} \int_0^L i(x(\xi)) e^{-i\frac{2\pi k}{L}\xi} d\xi = \frac{2}{L} \sum_{p=0}^{m-1} i_p^{\text{far}} e^{-i\frac{2\pi k}{L}\xi_p} L_p \quad [\text{for } k \text{ in } 1 \sim l - 1] \quad (4.26)$$

となる．遠方境界がすべて Dirichlet 条件の時は  $\Phi$  は既知数， $I$  は未知数となり，すべて Neumann 条件の時はその逆となる．

#### §4.3.3.2 離散化された境界積分方程式 1

境界積分方程式 1，すなわち式 (4.11) を基にして考える．まず，基本解を置く位置  $y$  を近傍要素のそれぞれの節点  $y_j^{\text{near}}$  とする．

すべて一定要素で離散化するため，左辺の  $c(y_j^{\text{near}}) = 1/2$  である．従って，式 (4.11) の左辺は，

$$\frac{1}{2} \phi(y_j^{\text{near}}) \quad (4.27)$$

となる．

右辺については，近傍の境界積分項は，

$$\begin{aligned} & \int_{\Gamma^{\text{near}}} i(x) \phi^*(y_j^{\text{near}}, x) - \phi(x) i^*(y_j^{\text{near}}, x) d\Gamma_x \\ & \quad \text{積分の離散化} \\ & \simeq \sum_{j=0}^{n-1} \left( i(x_j^{\text{near}}) \int_{\Gamma_j^{\text{near}}} \phi^*(y_j^{\text{near}}, x) d\Gamma_x - \phi(x_j^{\text{near}}) \int_{\Gamma_j^{\text{near}}} i^*(y_j^{\text{near}}, x) d\Gamma_x \right) \quad (4.28) \end{aligned}$$

となり，通常の境界要素法と同様の手法で計算できる．ただし， $\Gamma_j^{\text{near}}$  は近傍の  $j$  番目の境界要素

を意味する．遠方の境界積分項は，各項それぞれ，

$$\sum_{k=0}^{\infty} \operatorname{Re}[I_k] \operatorname{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y_J^{\text{near}}, x(\xi)) d\xi \right] \simeq \sum_{k=0}^{l-1} \operatorname{Re}[I_k] \operatorname{Re} \left[ \sum_{p=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_p} \int_{\Gamma_p^{\text{far}}} \phi^*(y_J^{\text{near}}, x) d\Gamma_x \right] \quad (4.29)$$

$$\sum_{k=0}^{\infty} \operatorname{Im}[I_k] \operatorname{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} \phi^*(y_J^{\text{near}}, x(\xi)) d\xi \right] \simeq \sum_{k=0}^{l-1} \operatorname{Im}[I_k] \operatorname{Im} \left[ \sum_{p=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_p} \int_{\Gamma_p^{\text{far}}} \phi^*(y_J^{\text{near}}, x) d\Gamma_x \right] \quad (4.30)$$

$$\sum_{k=0}^{\infty} \operatorname{Re}[\Phi_k] \operatorname{Re} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y_J^{\text{near}}, x(\xi)) d\xi \right] \simeq \sum_{k=0}^{l-1} \operatorname{Re}[\Phi_k] \operatorname{Re} \left[ \sum_{p=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_p} \int_{\Gamma_p^{\text{far}}} i^*(y_J^{\text{near}}, x) d\Gamma_x \right] \quad (4.31)$$

$$\sum_{k=0}^{\infty} \operatorname{Im}[\Phi_k] \operatorname{Im} \left[ \int_0^L e^{i\frac{2\pi k}{L}\xi} i^*(y_J^{\text{near}}, x(\xi)) d\xi \right] \simeq \sum_{k=0}^{l-1} \operatorname{Im}[\Phi_k] \operatorname{Im} \left[ \sum_{p=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_p} \int_{\Gamma_p^{\text{far}}} i^*(y_J^{\text{near}}, x) d\Gamma_x \right] \quad (4.32)$$

となる．これにより，実際に積分計算が必要となる部分は通常の境界要素法と同じになる（解析的に積分することが可能である．<sup>[4.6]</sup>）

この離散化により，近傍には  $n$  個の要素があることから，境界積分方程式 1 として  $n$  本の式を立てることが出来る．

#### §4.3.3.3 離散化された境界積分方程式 2

境界積分方程式 1，すなわち式 (4.11) を基にして考える．まず，重み関数  $W_K(x)$  およびその境界での勾配  $\frac{\partial W_K(x)}{\partial n(x)}$  を，

$$W_K(x) = \int_{\Gamma^{\text{far}}} \phi^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\Gamma_y \simeq \sum_{p=0}^{m-1} \left( \phi^*(y_p^{\text{far}}, x) e^{i\frac{2\pi K}{L}\eta_p} L_p \right) \quad (4.33)$$

$$\frac{\partial W_K(x)}{\partial n(x)} = \int_{\Gamma^{\text{far}}} i^*(y(\eta), x) e^{i\frac{2\pi K}{L}\eta} d\Gamma_y \simeq \sum_{p=0}^{m-1} \left( i^*(y_p^{\text{far}}, x) e^{i\frac{2\pi K}{L}\eta_p} L_p \right) \quad (4.34)$$

の様に離散近似する．

遠方もすべて一定要素で離散化するため，左辺の  $c(y) = 1/2$  である．従って左辺は，

$$\int_0^L c(y(\eta)) \phi(y(\eta)) e^{i\frac{2\pi K}{L}\eta} d\eta \simeq \frac{1}{2} \int_0^L \phi(y(\eta)) e^{i\frac{2\pi K}{L}\eta} d\eta = \begin{cases} \frac{L}{2} (\operatorname{Re}[\Phi_0] - i\operatorname{Im}[\Phi_0]) & [\text{for } K = 0]^{(\text{viii})} \\ \frac{L}{4} (\operatorname{Re}[\Phi_K] - i\operatorname{Im}[\Phi_K]) & [\text{for } K \geq 1] \end{cases} \quad (4.35)$$

となる．

<sup>(viii)</sup>先に述べた通り  $\operatorname{Im}[\Phi_0] = 0$  なので，実際には  $\frac{L}{2} \operatorname{Re}[\Phi_0]$  のみである．

右辺については，近傍の境界積分項は

$$\int_{\Gamma^{\text{near}}} i(x)W_K(x) - \phi(x) \frac{\partial W_K(x)}{\partial n(x)} d\Gamma_x$$

近傍積分の離散化

$$\simeq \sum_{j=0}^{n-1} \left( i_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} W_K(x) d\Gamma_x - \phi_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} \frac{\partial W_K(x)}{\partial n(x)} d\Gamma_x \right) \quad (4.36)$$

離散近似された  $W$  の代入

$$\simeq \sum_{j=0}^{n-1} \left( i_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} \sum_{p=0}^{m-1} \left( \phi^*(y_p^{\text{far}}, x) e^{i \frac{2\pi K}{L} \eta_p L_p} \right) d\Gamma_x - \phi_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} \sum_{p=0}^{m-1} \left( i^*(y_p^{\text{far}}, x) e^{i \frac{2\pi K}{L} \eta_p L_p} \right) d\Gamma_x \right) \quad (4.37)$$

積分と総和の順序入れ替え

$$= \sum_{j=0}^{n-1} \left( i_j^{\text{near}} \sum_{p=0}^{m-1} \left( e^{i \frac{2\pi K}{L} \eta_p L_p} \int_{\Gamma_j^{\text{near}}} \phi^*(y_p^{\text{far}}, x) d\Gamma_x \right) - \phi_j^{\text{near}} \sum_{p=0}^{m-1} \left( e^{i \frac{2\pi K}{L} \eta_p L_p} \int_{\Gamma_j^{\text{near}}} i^*(y_p^{\text{far}}, x) d\Gamma_x \right) \right) \quad (4.38)$$

となる．

遠方の境界積分項は，まず第1項を取り上げると，

$$\sum_{k=0}^{\infty} \text{Re}[I_k] \left( \int_0^L \text{Re}[e^{i \frac{2\pi k}{L} \xi}] W_K(x) d\xi \right)$$

Fourier 級数打ち切りと，離散近似された  $W$  の代入

$$\simeq \sum_{k=0}^{l-1} \text{Re}[I_k] \left( \int_{\Gamma^{\text{far}}} \text{Re}[e^{i \frac{2\pi k}{L} \xi}] \sum_{p_1=0}^{m-1} \left( \phi^*(y_{p_1}^{\text{far}}, x) e^{i \frac{2\pi K}{L} \eta_{p_1} L_{p_1}} \right) d\Gamma_x \right) \quad (4.39)$$

積分と総和の順序入れ替え

$$= \sum_{k=0}^{l-1} \text{Re}[I_k] \left( \sum_{p_1=0}^{m-1} e^{i \frac{2\pi K}{L} \eta_{p_1} L_{p_1}} \int_{\Gamma^{\text{far}}} \text{Re}[e^{i \frac{2\pi k}{L} \xi}] \phi^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right) \quad (4.40)$$

遠方積分の離散化

$$\simeq \sum_{k=0}^{l-1} \text{Re}[I_k] \left( \sum_{p_1=0}^{m-1} e^{i \frac{2\pi K}{L} \eta_{p_1} L_{p_1}} \sum_{p_2=0}^{m-1} \text{Re}[e^{i \frac{2\pi k}{L} \xi_{p_2}}] \int_{\Gamma_{p_2}^{\text{far}}} \phi^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right) \quad (4.41)$$

Re を総和と積分の外に移動

$$= \sum_{k=0}^{l-1} \text{Re}[I_k] \left( \sum_{p_1=0}^{m-1} e^{i \frac{2\pi K}{L} \eta_{p_1} L_{p_1}} \text{Re} \left[ \sum_{p_2=0}^{m-1} e^{i \frac{2\pi k}{L} \xi_{p_2}} \int_{\Gamma_{p_2}^{\text{far}}} \phi^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right] \right) \quad (4.42)$$

と出来る．同様にして遠方積分の他項についても近似を行うと，

$$\begin{aligned} & \sum_{k=0}^{\infty} \operatorname{Re}[I_k] \left( \int_0^L \operatorname{Re}[e^{i\frac{2\pi k}{L}\xi}] W_K(x) d\xi \right) \\ & \simeq \sum_{k=0}^{l-1} \operatorname{Re}[I_k] \left( \sum_{p_1=0}^{m-1} e^{i\frac{2\pi K}{L}\eta_{p_1}} L_{p_1} \operatorname{Re} \left[ \sum_{p_2=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_{p_2}} \int_{\Gamma_{p_2}^{\text{far}}} \phi^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right] \right) \end{aligned} \quad (4.43)$$

$$\begin{aligned} & \sum_{k=0}^{\infty} \operatorname{Im}[I_k] \left( \int_0^L \operatorname{Im}[e^{i\frac{2\pi k}{L}\xi}] W_K(x) d\xi \right) \\ & \simeq \sum_{k=0}^{l-1} \operatorname{Im}[I_k] \left( \sum_{p_1=0}^{m-1} e^{i\frac{2\pi K}{L}\eta_{p_1}} L_{p_1} \operatorname{Im} \left[ \sum_{p_2=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_{p_2}} \int_{\Gamma_{p_2}^{\text{far}}} \phi^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right] \right) \end{aligned} \quad (4.44)$$

$$\begin{aligned} & \sum_{k=0}^{\infty} \operatorname{Re}[\Phi_k] \left( \int_0^L \operatorname{Re}[e^{i\frac{2\pi k}{L}\xi}] \frac{\partial W_K(x)}{\partial n(x)} d\xi \right) \\ & \simeq \sum_{k=0}^{l-1} \operatorname{Re}[\Phi_k] \left( \sum_{p_1=0}^{m-1} e^{i\frac{2\pi K}{L}\eta_{p_1}} L_{p_1} \operatorname{Re} \left[ \sum_{p_2=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_{p_2}} \int_{\Gamma_{p_2}^{\text{far}}} i^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right] \right) \end{aligned} \quad (4.45)$$

$$\begin{aligned} & \sum_{k=0}^{\infty} \operatorname{Im}[\Phi_k] \left( \int_0^L \operatorname{Im}[e^{i\frac{2\pi k}{L}\xi}] \frac{\partial W_K(x)}{\partial n(x)} d\xi \right) \\ & \simeq \sum_{k=0}^{l-1} \operatorname{Im}[\Phi_k] \left( \sum_{p_1=0}^{m-1} e^{i\frac{2\pi K}{L}\eta_{p_1}} L_{p_1} \operatorname{Im} \left[ \sum_{p_2=0}^{m-1} e^{i\frac{2\pi k}{L}\xi_{p_2}} \int_{\Gamma_{p_2}^{\text{far}}} i^*(y_{p_1}^{\text{far}}, x) d\Gamma_x \right] \right) \end{aligned} \quad (4.46)$$

となる．先程と同じく，実際に積分計算が必要となる部分は通常の境界要素法と同じになる．

この離散化により，遠方の Fourier 係数の低次  $l$  項を用いることから，境界積分方程式 2 として  $l$  本の式を立てることが出来る．しかし，境界積分方程式 2 は複素数の式であるので，実部と虚部を別々の式と捉えれば  $2l$  本の式が立っていることとなる．

#### §4.3.3.4 マトリックス方程式

最終的に得られるマトリックス方程式は， $n$  本の境界積分方程式 1 と， $l$  本の境界積分方程式 2 の実部の式，及び  $l$  本の境界積分方程式 2 の虚部の式を連立させたものとなり，通常の境界要素法と同様の形をしたマトリックス方程式となる．

$$[H]\{u\} = [G]\{q\} \quad (4.47)$$

ただし,  $\{u\}, \{q\}$  は  $n + 2l$  行のベクトルであり, 境界変数が次のような順で並んでいる.

$$\{u\} = \begin{pmatrix} \phi_0^{\text{near}} \\ \vdots \\ \phi_{n-1}^{\text{near}} \\ \text{Re}[\Phi_0] \\ \vdots \\ \text{Re}[\Phi_{l-1}] \\ \text{Im}[\Phi_0] \\ \vdots \\ \text{Im}[\Phi_{l-1}] \end{pmatrix}, \quad \{q\} = \begin{pmatrix} i_0^{\text{near}} \\ \vdots \\ i_{n-1}^{\text{near}} \\ \text{Re}[I_0] \\ \vdots \\ \text{Re}[I_{l-1}] \\ \text{Im}[I_0] \\ \vdots \\ \text{Im}[I_{l-1}] \end{pmatrix} \quad (4.48)$$

また,  $[H], [G]$  は  $n + 2l$  行  $n + 2l$  列の正方マトリックスであり, マトリックスの成分は境界積分方程式 1 と 2 から作られる.  $\{u\}, \{q\}$  が先に述べたような順に並んでいることから,  $[H], [G]$  マトリックスには次のような物理的意味を持った成分が並んでいることになる.

積分方程式 1	near to near	near to far	(4.49)
積分方程式 2	far to near	far to far	

このマトリックスの解法については, 通常境界要素法と全く同じであるので省略する.

### §4.3.4 注目領域解の計算

前小節の方法により, 近傍での境界値と遠方境界の Fourier 係数, すなわち  $\{u\}, \{q\}$  が求まったとする. この時, 境界値ではなく領域内部の点におけるポテンシャル値を求めることを考える.

重み関数を領域内の点  $Y$  に置いた基本解として重み付き残差法を行うと，次式が得られる．

$$\phi(Y) = \int_{\Gamma} i(x)\phi^*(Y, x) - \phi(x)i^*(Y, x)d\Gamma_x \quad (4.50)$$

遠近分離

$$= \int_{\Gamma^{\text{near}}} i(x)\phi^*(Y, x) - \phi(x)i^*(Y, x)d\Gamma_x + \int_{\Gamma^{\text{far}}} i(x)\phi^*(Y, x) - \phi(x)i^*(Y, x)d\Gamma_x \quad (4.51)$$

遠方を Fourier 級数で近似

$$\begin{aligned} &\simeq \int_{\Gamma^{\text{near}}} i(x)\phi^*(Y, x) - \phi(x)i^*(Y, x)d\Gamma_x \\ &+ \int_{\Gamma^{\text{far}}} \text{Re} \left[ \sum_{k=0}^{l-1} I_k e^{i\frac{2\pi k}{L}\xi} \right] \phi^*(Y, x) - \text{Re} \left[ \sum_{k=0}^{l-1} \Phi_k e^{i\frac{2\pi k}{L}\xi} \right] i^*(Y, x) d\Gamma_x \end{aligned} \quad (4.52)$$

離散化

$$\begin{aligned} &\simeq \sum_{j=0}^{n-1} \left( i_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} \phi^*(Y, x) d\Gamma_x - \phi_j^{\text{near}} \int_{\Gamma_j^{\text{near}}} i^*(Y, x) d\Gamma_x \right) \\ &+ \sum_{p=0}^{m-1} \left( \text{Re} \left[ \sum_{k=0}^{l-1} I_k e^{i\frac{2\pi k}{L}\xi_p} \right] \int_{\Gamma_p^{\text{far}}} \phi^*(Y, x) d\Gamma_x - \text{Re} \left[ \sum_{k=0}^{l-1} \Phi_k e^{i\frac{2\pi k}{L}\xi_p} \right] \int_{\Gamma_p^{\text{far}}} i^*(Y, x) d\Gamma_x \right) \end{aligned} \quad (4.53)$$

上式は領域内部でのポテンシャルが  $\phi^{\text{near}}$ ,  $i^{\text{near}}$  及び  $\Phi, I$  から求められることを意味しており特に  $Y$  が注目領域内にあるとすれば， $\phi(Y)$  は充分正確に求めることができる．

## §4.4 解析例

この節では幾つかの形状に対する本手法での解析例を示す．まず，本手法の計算精度について通常の境界要素法との比較を交えて考察し，本手法が注目領域における解を充分正確に求めていることを示す．続いて，解析に要した計算時間についてやはり本手法と通常の境界要素法とで比較を行い，本手法を用いることで通常の境界要素法に比べて非常に高速に注目領域解を求めることができることを示す．

### §4.4.1 計算精度比較

#### §4.4.1.1 ドーナツ領域での例

最も単純な例として，図 4.10 に示すようなドーナツ形状の 2 次元 Laplace 領域を考える．内周の境界を注目境界，外周の境界を非注目境界と設定し，図 4.10 に示すような極座標をとる．ポテンシャルを  $\phi$ ，境界上のフラックスの外向き法線成分を  $i$  とおく．

内周の境界では図 4.11 に示す様な常に  $i = 0$  の Neumann 境界条件とし，外周の境界では図 4.12 の破線で示す様な Dirichlet 境界条件を与えた．求める未知変数は近傍境界（内周）の Dirichlet

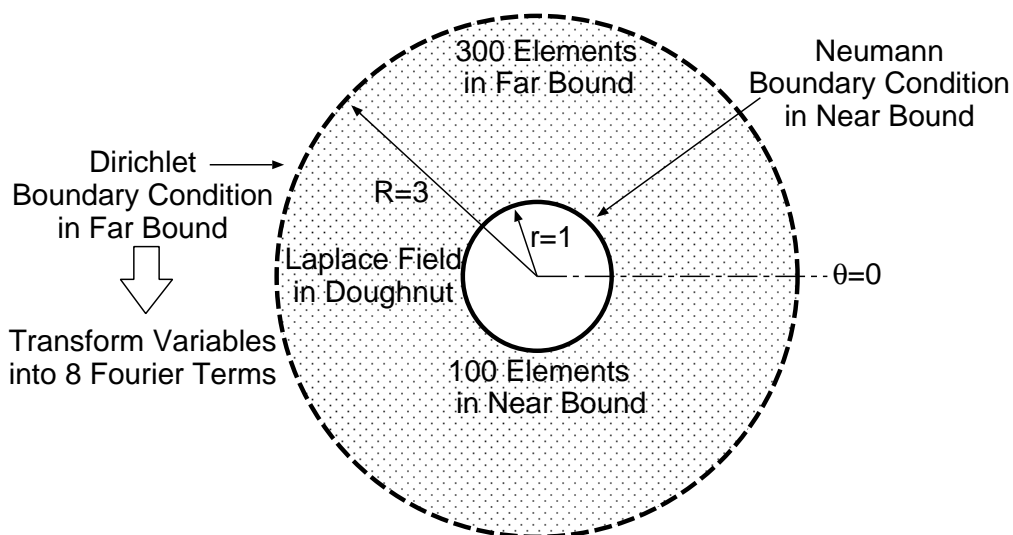


Fig. 4.10: Model of doughnut-shaped example

データ  $\phi$  と遠方境界（外周）の Neumann データ  $i$  であり、これを通常の境界要素法と Fourier 級数展開項数  $l = 8$  とした本手法とで計算精度の比較を行った。ただし、本手法は近傍境界（内周）の Dirichlet データのみを正確に求めることを目的としている。

図 4.11 は近傍境界の境界条件であり、本手法でも通常の境界要素法と同様に扱うため全く差がない。しかし図 4.12 に示す遠方境界の境界条件は、本手法では Fourier 級数の低次 8 項しか取り扱わない為、高周波成分の項分が差となって現れている。

図 4.13 および図 4.14 は両手法で求めた解であり、図 4.13 が近傍境界での解を、図 4.14 が遠方境界での解を表している。遠方境界においては、本手法の解は通常の境界要素法の解の低周波成分のみを表現していることが読みとれるが、近傍境界では両者の解がグラフの上で完全に一致している。

この解析例から、ドーナツ領域のように境界が 2 つ以上に明確に分断されているような形状の解析において、いずれか 1 つの境界の近傍の解のみが必要であるといった場合に本手法は非常に有効であり、十分な精度で近傍（注目）領域の解を求めることが可能であることが示された。

#### §4.4.1.2 円形領域での例

ドーナツ領域と異なり、近傍境界と遠方境界とが分離出来ずに繋がっている例として、図 4.15 に示すような単位円形状の 2 次元 Laplace 領域を考える。図 4.15 の通りに角度座標  $\theta$  をとり、 $0 < \theta < \pi/2$  を近傍境界、 $\pi/2 < \theta < 2\pi$  を遠方境界とした。なお、ポテンシャルを  $\phi$ 、境界上のフラックスの外向き法線成分を  $i$  とおく。

境界条件はすべて Dirichlet 条件とし、図 4.16 の破線の通りに与えた。求める未知変数は Neumann データ  $i$  であり、通常の境界要素法と Fourier 級数展開項数  $l = 8$  とした本手法を用いて計算した  $i$  の値を図 4.17 および図 4.18 に示す。ただし、本手法は近傍境界（ $\theta = 0 \sim 90$ ）の Neumann

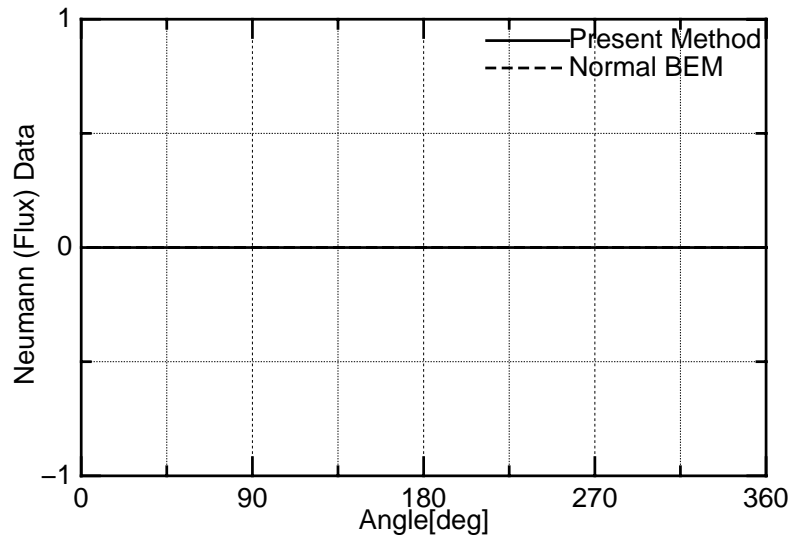


Fig. 4.11: Boundary condition (Neumann data) of doughnut-shaped example in near bound (inner circle)

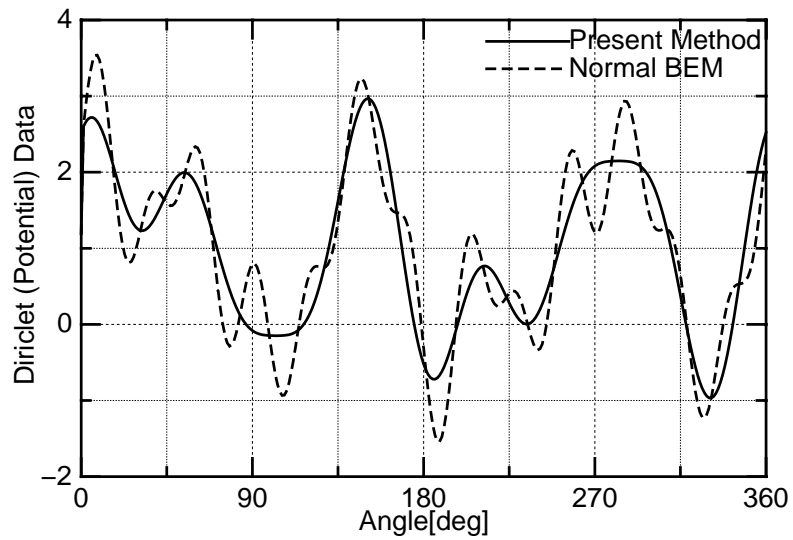


Fig. 4.12: Boundary condition (Dirichlet data) of doughnut-shaped example in far bound (outer circle)

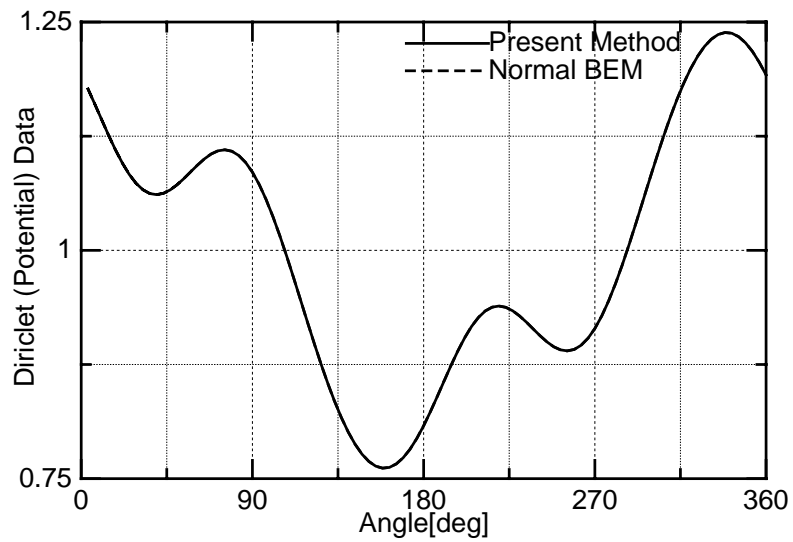


Fig. 4.13: Solution(Dirichlet data) of doughnut-shaped example in near bound (inner circle)

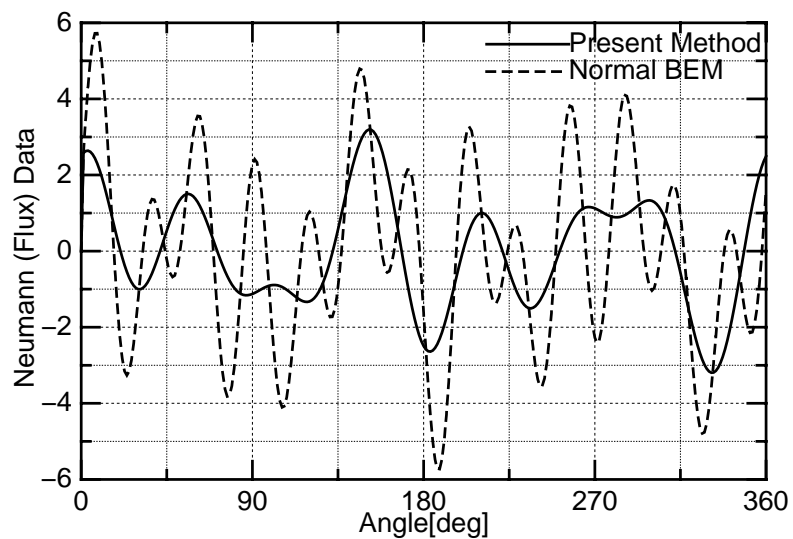


Fig. 4.14: Solution(Newmann data) of doughnut-shaped example in far bound (outer circle)

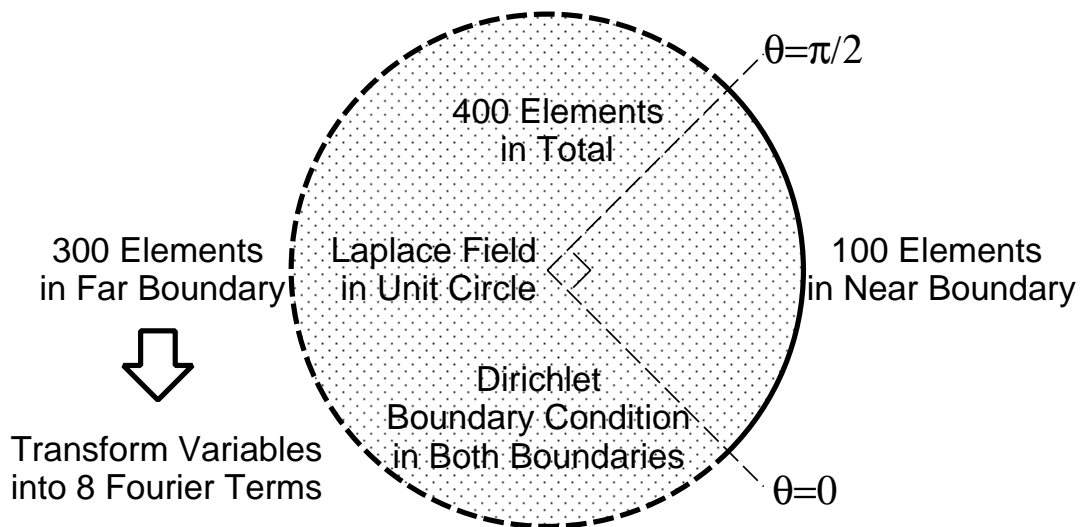


Fig. 4.15: Model of circle-shaped example

データのみを正確に求めることを目的としている。

図 4.16 は境界条件のグラフであり， $0 \sim 90[\text{deg}]$  の範囲は近傍境界なので両グラフは同じものとなるが， $90 \sim 360[\text{deg}]$  の遠方境界の範囲では本手法は遠方境界の Fourier 級数展開の低次 8 項分のデータしか取り扱わない為両者に差が生じる。

図 4.17 が両手法で求めた両境界での解であり，遠方境界にあたる横軸  $90 \sim 360[\text{deg}]$  の範囲では本手法で得られた遠方境界の  $i$  は通常の境界要素法の解，すなわち正解の低周波成分のみを表現している。しかし図 4.18 (図 4.16 の横軸  $0 \sim 90[\text{deg}]$  範囲と同じ) に示す通り，近傍境界の  $i$  は両端を除いて<sup>(ix)</sup> 正解とほとんど一致しており，本手法が注目領域の解だけを正確に求めていることが読みとれる。

この解析例から，円形領域のように境界がすべて 1 本に繋がっていて分断されていないような形状の解析においても，境界のある一部分の近傍の解のみを本手法により精度良く求めることが可能であることが示された。

#### §4.4.1.3 花形領域での例

最後に形状が入り組んでいる場合の例として，図 4.19 に示すような花形領域の 2 次元 Laplace 領域を考える。花の外周を遠方領域，花の内周 (単位円周) を近傍領域とし，内周の中心を原点とした極座標をとる。花の内周，外周共に極座標において等角度に要素を分割を行った。

境界条件はすべて Dirichlet 条件とし，内周は図 4.19 に示すように常に  $\phi = 0$ ，外周は図 4.21 の

<sup>(ix)</sup> 本手法では境界変数の周波数特性分解に Fourier 変換を用いている為，近傍境界と遠方境界の境目点，すなわち図 4.16 および 4.17 の  $90[\text{deg}], 360[\text{deg}] (=0[\text{deg}])$  の位置での境界変数の値は境界条件のデータと一致せず，従って値が不連続になってしまうという欠点がある。それが為，本手法では境目点で境界条件が不連続であると見なしてしまい，本手法の解と正解とが一致していない。

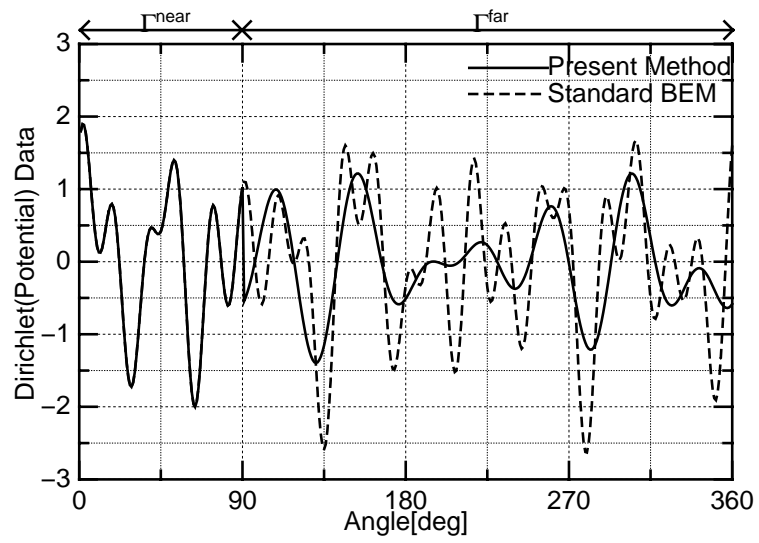


Fig. 4.16: Boundary condition of circle-shaped example

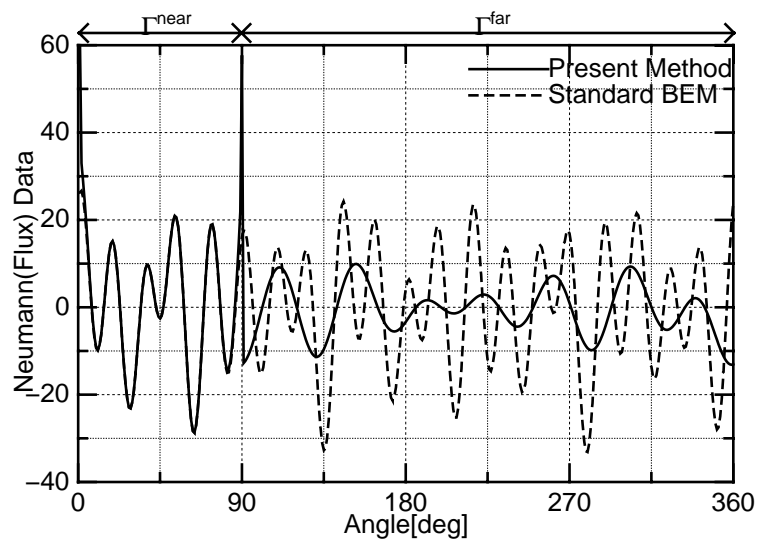


Fig. 4.17: Solution of circle-shaped example

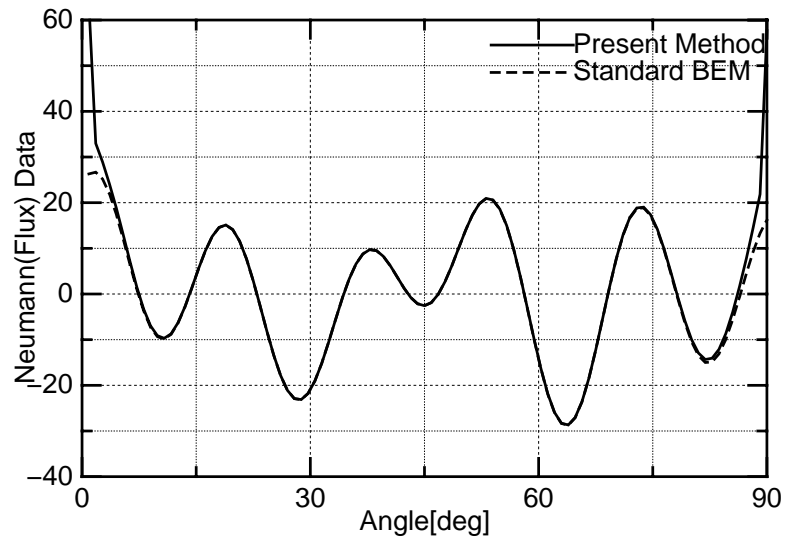


Fig. 4.18: Solution of circle-shaped example in near bound

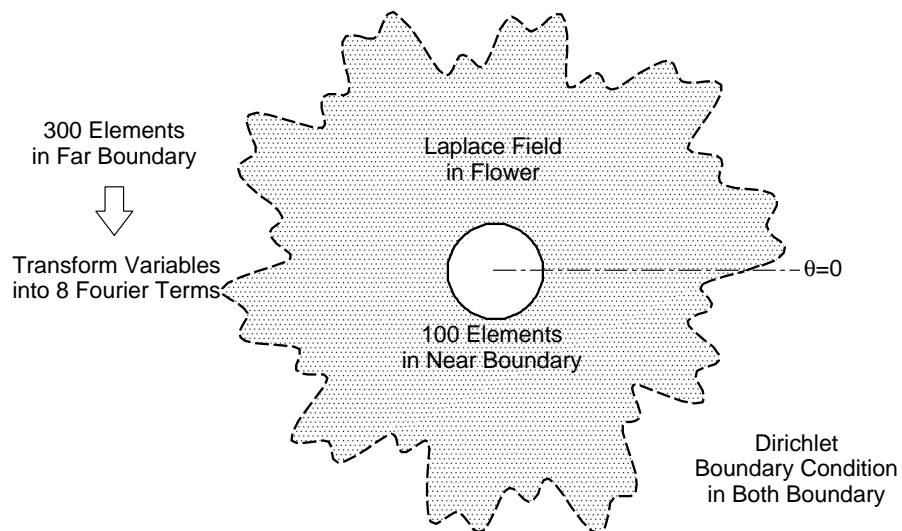


Fig. 4.19: Model of flower-shaped example

破線で示すような境界条件を与えた．求める未知数は両境界の Neumann データ  $i$  であり，やはり Fourier 級数展開項数  $l$  を 8 とした本手法と通常の境界要素法とで計算精度の比較を行った．ただし，本手法は近傍境界である内周の Neumann データのみを正確に求めることを目的としている．

図 4.22 および図 4.23 は両手法で求めた解であり，図 4.22 が近傍境界での解を，図 4.23 が遠方境界での解を表している．この例においても，遠方境界における本手法の解は通常の境界要素法の解の低周波成分を表現している．近傍境界における本手法の解は通常の境界要素法の解に非常に近づいていることが確認できる．

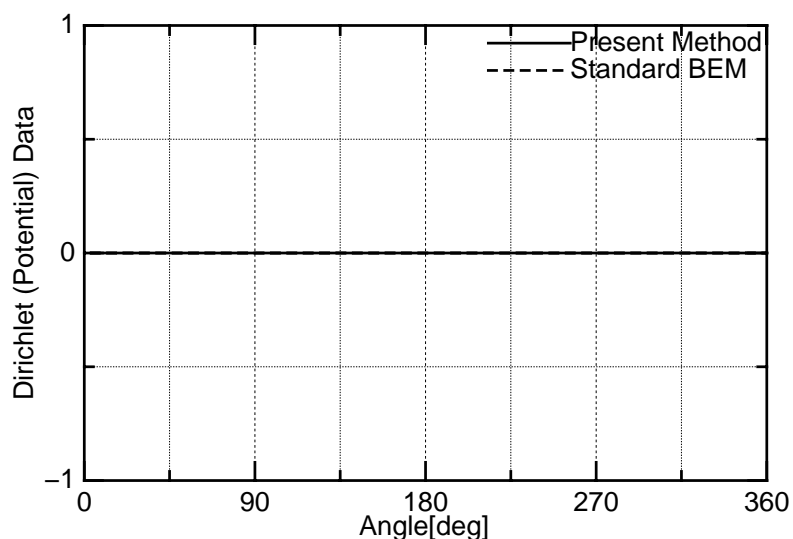


Fig. 4.20: Boundary condition of flower-shaped example in near bound (inner circle)

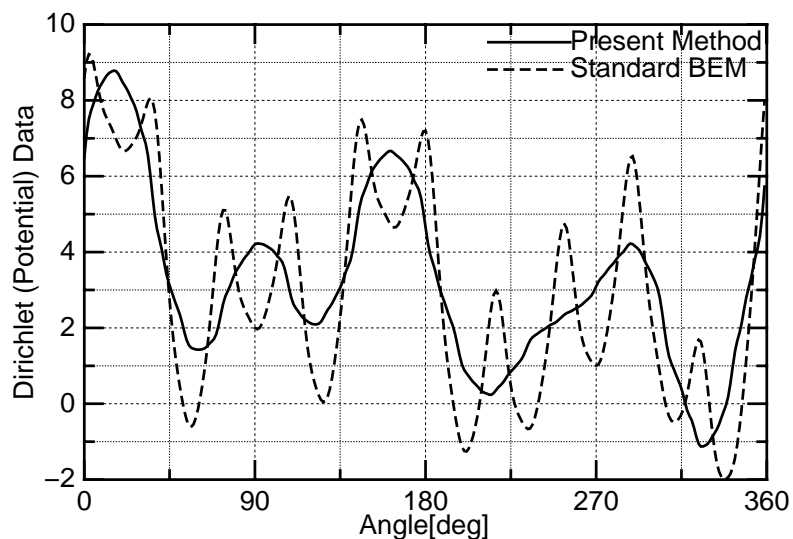


Fig. 4.21: Boundary condition of flower-shaped example in far bound (outer circle)

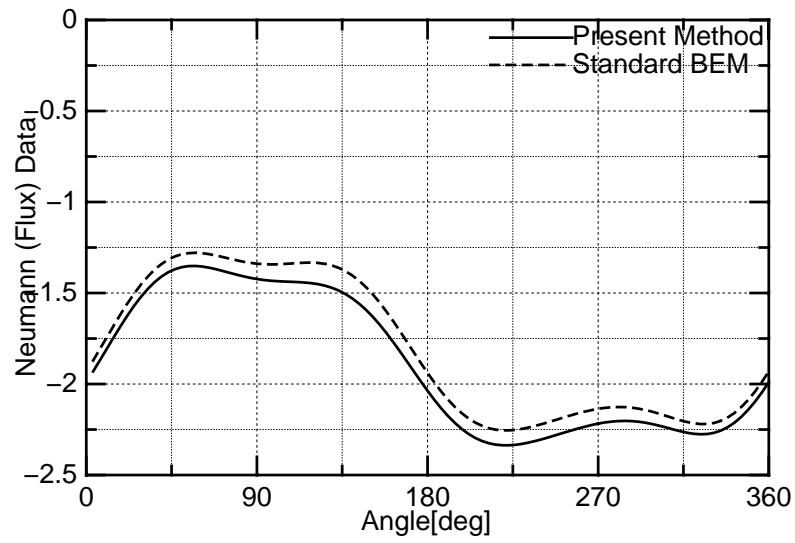


Fig. 4.22: Solution of flower-shaped example in near bound (inner circle)

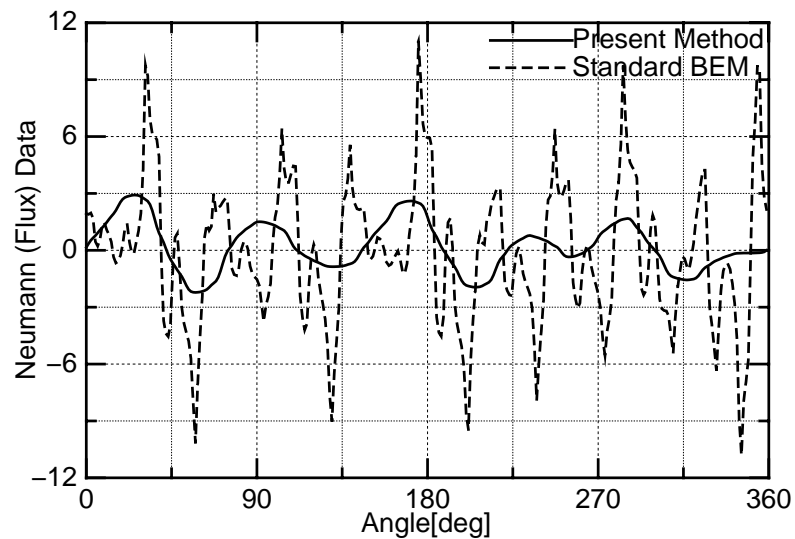


Fig. 4.23: Solution of flower-shaped example in far bound (outer circle)

### §4.4.2 計算時間比較

続いて計算時間の比較を行う．前小節で例に挙げたドーナツ形もしくは花形領域の問題に対し，境界要素数を変化させて計算にかかった時間を実際の計算機上で比較した．

内周を近傍境界とし，要素数は100要素で固定とする．外周を遠方境界とし，要素数を900～11900要素まで1000要素刻みに変化させて計算を行った．また，Fourier展開項数 $l$ は8で固定とする．<sup>(x)</sup>なお，計算に使用したCPUは731MHz Alpha 21264であり，計算時間計測はtcshのtimeコマンドを用いた．

図4.24，4.25は共に横軸に全要素数，縦軸に計算時間をとった同じグラフであり，図4.24は線形スケールで，図4.25はLogスケールで描いてある．総要素数が5000近辺で計算時間が逆転して本手法の方が高速に計算できていることが分かる．近似曲線が示す通り，総要素数を $N$ とすると通常の境界要素法が $N^3$ ，本手法が $N^2$ に比例して計算が増加しており，この解析例に限らず要素数がある程度以上多くなれば本手法の方が通常の境界要素法よりも高速に計算できることとなる．さらに，Fourier級数展開等の計算ルーチンにFFTを用いれば $N \log N$ に比例する計算時間で計算することが可能となるはずであり，さらなる高速化が可能である．

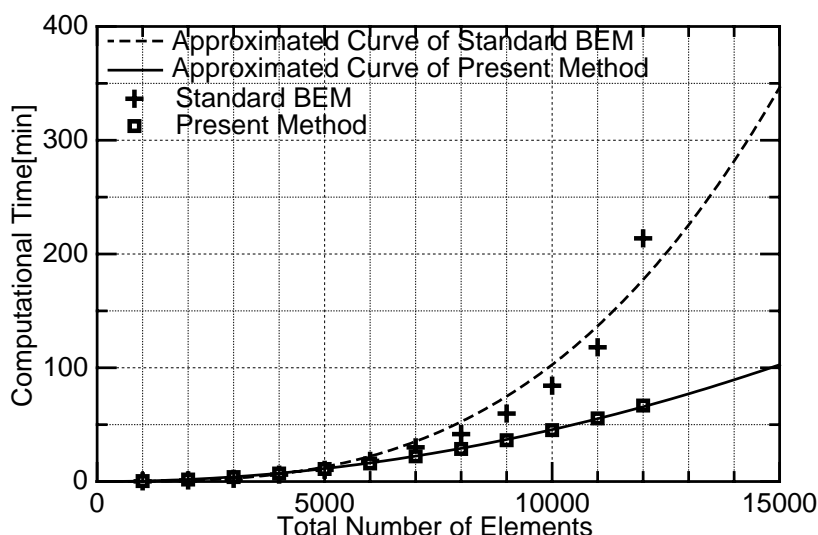


Fig. 4.24: Comparison of CPU time between present method and normal BEM (linear scale)

## §4.5 結言

本研究では2次元ラプラス方程式の境界値問題に対し，境界要素法を用いて特定の注目領域の解だけを高速に求める方法を開発した．非注目領域にある境界変数が空間的に高周波に振動していたとしても，注目領域にはその影響がほとんど現れないことに着目している．具体的には，遠方境界変数をFourier級数展開で表し，その高周波成分を無視した境界積分方程式を用いる．こ

<sup>(x)</sup>本手法の計算時間は要素数が充分多い範囲においてはFourier展開項数に比例する．

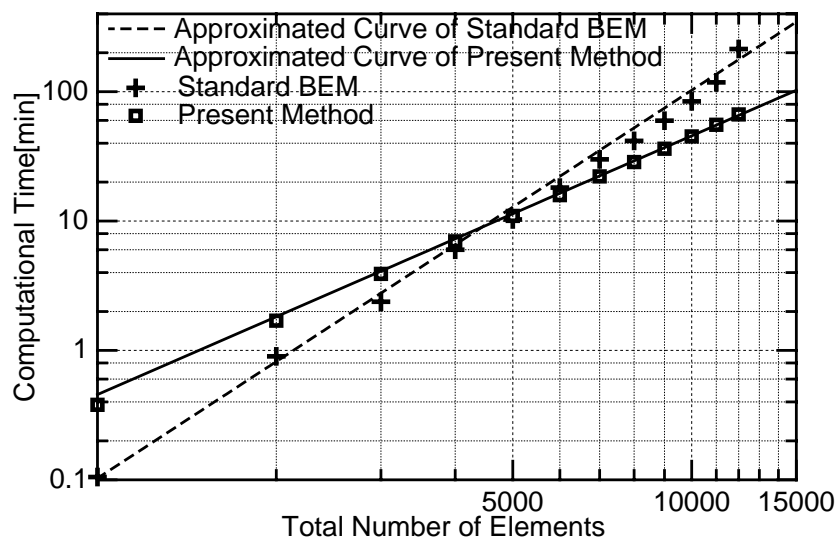


Fig. 4.25: Comparison of CPU time between present method and normal BEM (log scale)

の式は非注目領域を荒く，注目領域を正確に表現した式となっており，注目領域解を高速かつ十分な精度で求めることを可能にした．

実際の問題においても，領域内すべての場所での解ではなく，ある注目領域の解だけが必要であるという場合は少なくない．従って，この手法を用いることにより，特に大規模な電場，伝熱等の解析を必要とする構造設計を非常に効率化する事が出来ると考えられる．

## 参考文献

- [ 4.1 ] V.Rokhlin, “Rapid Solution of Integral Equations of Classical Potential Theory”, *Journal of Computational Physics*, **60**, (1983), pp.187–207
- [ 4.2 ] L.Greengard and V.Rokhlin, “A Fast Algorithm for Particle Simulations”, *Journal of Computational Physics*, **73**, (1987), pp.325–348
- [ 4.3 ] 福井卓雄, 服部純一, 土居野優, “高速多重極法の境界要素解析への応用”, 土木学会 構造工学論文集, **43A**, (1997)
- [ 4.4 ] 田中正隆, 松本敏郎, 中村正行, “計算力学と CAE シリーズ 2 境界要素法”, 培風館, (1991), pp.23–48
- [ 4.5 ] 高木貞治, “解析概論”, 岩波書店, (1983), pp.268–293
- [ 4.6 ] 水本久夫, 原平八郎, “FORTRAN による境界要素法の基礎”, サイエンス社, (1985), pp.38–47
- [ 4.7 ] 神谷紀生, 北栄輔, “トレフツ法入門”, コロナ社, (2000)

## 第5章 結論

本論文では大規模数値シミュレーションの高速化に関する3つの問題に取り組み、次のような研究成果を得た。

- (1) 大規模行列を簡便かつ高速に取り扱うためのC++クラスライブラリを開発し、大規模行列計算を必要とする種々の数値シミュレーションの開発時間の短縮を実現した。
- (2) 断面が母線方向に一様である棒状の弾性体に対し、その幾何学的特徴を手法に活かした動的有限要素解析の高速解法を開発した。
- (3) ラプラス方程式の境界値問題に対し、注目していない領域周辺の境界値の解を不必要に正確に求めることを止め、代わりに注目する領域付近の境界値の解を精度良く高速に求める方法を開発した。

ところで、本論文の緒論において大規模数値シミュレーションの高速化に関する従来研究の3つの大きな問題点を指摘した。

- [1] 数値シミュレーションの高速化の研究では往々にして「計算時間」の短縮についてのみに主眼が置かれており、「開発時間」の短縮があまり重要視されていない点。
- [2] 物体の幾何学的特徴を利用した効率的数値解析手法の開発を目的とした研究が少ない点。
- [3] 一般的な数値シミュレーションの解法が全ての未知数に対して同じ精度で解を求めることを暗黙の内に前提としてしまっている点。

本論文に記載した3つの研究成果は、これら3つの大きな問題点を持つ特定の具体的な問題に対する解決法を示したものである。従って、これら3つの大きな問題点が本論文によって全て克服された訳ではない。同種の問題点を持つ研究は、既出の研究および本論文で取り扱った研究以外にも未着手のものが多く残されているものと想像される。今後も本論文と同じ視点を持って諸種の問題の解決法の研究に取り組むことが必要であると考えられる。