

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-111749

(P2017-111749A)

(43) 公開日 平成29年6月22日 (2017.6.22)

(51) Int. Cl.		F I			テーマコード (参考)	
<b>G06F</b>	<b>9/44</b>	<b>(2006.01)</b>	G06F	9/06	620A	5B081
<b>G06F</b>	<b>9/45</b>	<b>(2006.01)</b>	G06F	9/44	320F	5B376

審査請求 未請求 請求項の数 10 O L (全 26 頁)

(21) 出願番号	特願2015-247693 (P2015-247693)	(71) 出願人	504132272 国立大学法人京都大学 京都府京都市左京区吉田本町36番地1
(22) 出願日	平成27年12月18日 (2015.12.18)	(74) 代理人	100101454 弁理士 山田 卓二
		(74) 代理人	100081422 弁理士 田中 光雄
		(74) 代理人	100125874 弁理士 川端 純市
		(72) 発明者	嶋吉 隆夫 京都府京都市左京区吉田本町36番地1 国立大学法人京都大学内

最終頁に続く

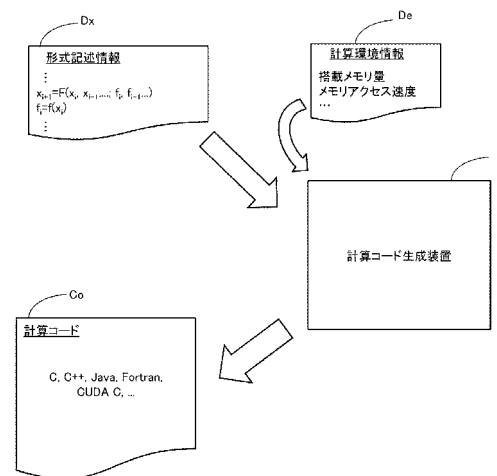
(54) 【発明の名称】 計算コード生成装置、方法及びプログラム

(57) 【要約】 (修正有)

【課題】数値計算プログラムの生産性を向上することができる計算コード生成装置、方法及びプログラムを提供する。

【解決手段】計算コード生成装置1は、計算対象の数学的問題に対する数値計算が実行される計算コードCoを自動生成する。計算コード生成装置は、記憶部と、制御部とを備える。記憶部は、計算対象の数学的問題に対する数値計算法を数式で宣言的に記述する形式記述情報Dxを格納する。制御部は、記憶部に格納された形式記述情報に基づく演算処理を行う。制御部は、形式記述情報に基づいて、形式記述情報における数式及び数式が含む変数の関係を解析する。制御部は、解析結果に基づいて、所定の手続き型プログラミング言語により計算対象の関数に対する数値計算を実行する計算コードを生成する。形式記述情報は、反復型数値計算法を漸化式で規定する情報を含む。

【選択図】図1



## 【特許請求の範囲】

## 【請求項 1】

計算対象の数学的問題に対する数値計算が実行される計算コードを自動生成する計算コード生成装置であって、

数値計算法を数式で宣言的に記述する形式記述情報を格納する記憶部と、  
前記記憶部に格納された形式記述情報に基づく演算処理を行う制御部とを備え、  
前記制御部は、

前記形式記述情報に基づいて、前記形式記述情報における数式及び前記数式が含む変数の関係を解析して、

解析結果に基づいて、所定のプログラミング言語により前記数値計算を実行する計算コードを生成し、

前記形式記述情報は、反復型数値計算法を漸化式で規定する情報を含む計算コード生成装置。

## 【請求項 2】

前記制御部は、前記形式記述情報における漸化式を、前記漸化式に含まれる変数を介して関連する一連の数式に展開することにより、前記解析を行う

請求項 1 に記載の計算コード生成装置。

## 【請求項 3】

前記制御部は、前記記憶部に格納された形式記述情報における数式毎に各数式に含まれる変数を分類して、前記漸化式で規定する情報を解釈する

請求項 1 又は 2 に記載の計算コード生成装置。

## 【請求項 4】

前記制御部は、

前記計算コードが実行される計算環境を示す計算環境情報を取得し、

前記計算環境情報に応じて最適化して、前記計算コードを出力する

請求項 1 ~ 3 のいずれか 1 項に記載の計算コード生成装置。

## 【請求項 5】

前記形式記述情報における漸化式は、少なくとも 3 回にわたる反復計算回数にそれぞれ対応する少なくとも 3 つの変数を含む

請求項 1 ~ 4 のいずれか 1 項に記載の計算コード生成装置。

## 【請求項 6】

前記形式記述情報は、前記数値計算法をモジュール化して記述される

請求項 1 ~ 5 のいずれか 1 項に記載の計算コード生成装置。

## 【請求項 7】

前記生成された計算コードの外部から、前記数学的問題の具体形が指定される

請求項 1 ~ 6 のいずれか 1 項に記載の計算コード生成装置。

## 【請求項 8】

前記生成された計算コードの内部に、前記数学的問題の具体形が組み込まれている

請求項 1 ~ 6 のいずれか 1 項に記載の計算コード生成装置。

## 【請求項 9】

コンピュータが、計算対象の数学的問題に対する数値計算が実行される計算コードを自動生成する方法であって、

前記コンピュータの記憶部には、前記計算対象の数学的問題に対する数値計算法を数式で宣言的に記述する形式記述情報が格納されており、

前記コンピュータの制御部が、前記形式記述情報に基づいて、前記形式記述情報における数式及び前記数式が含む変数の関係を解析するステップと、

前記制御部が、解析結果に基づいて、所定のプログラミング言語により前記計算対象の関数に対する数値計算を実行する計算コードを生成するステップとを含み、

前記形式記述情報は、反復型数値計算法を漸化式で規定する情報を含む方法。

## 【請求項 10】

請求項 9 に記載の方法をコンピュータに実行させるためのプログラム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、数値計算法が実行されるプログラムコードを自動生成する計算コード生成装置、方法及びプログラムに関する。

## 【背景技術】

## 【0002】

従来、数値計算アルゴリズムにおいては、計算性能が重要視されるため、特定の手続型言語による参照実装が一般的に公開されている。しかし、この手法には移植性や保守性の問題があることから、機械的に数値計算法のプログラムコードを生成するための技術が提案されている（例えば特許文献 1，非特許文献 1）。

10

## 【0003】

特許文献 1 は、数値計算アルゴリズムに関する問題定義情報とプログラム様式情報の入力により、数値計算プログラムを生成する数値計算プログラム生成システムを開示している。特許文献 1 の数値計算プログラム生成システムによると、生成される数値計算プログラムの数値計算アルゴリズムを疑似コードの順序列で表現したものからプログラム生成を行う。これにより、既に定義されている疑似コードを利用して新たな数値計算アルゴリズムに対応する疑似コードの順序列を作り、数値計算プログラム生成システムが生成可能な数値計算アルゴリズムを拡張できる。

20

## 【0004】

非特許文献 1 は、細胞生理学モデルを記述したファイルと ODE（常微分方程式）解法スキームを記述したファイルから時間発展計算を行うシミュレーションプログラムの自動生成手法を開示している。非特許文献 1 の自動生成手法によると、計算機が、上記の各ファイルの入力によって時間発展を計算する数式セットを自動生成し、生成された数式セットから実行コードを生成する。

## 【先行技術文献】

## 【特許文献】

## 【0005】

【特許文献 1】特開 2004 - 86760 号公報

## 【非特許文献】

## 【0006】

【非特許文献 1】山下義陽、外 7 名、「形式的に記述された ODE 解法スキームに基づく CellML シミュレーションコード生成システム」、生体医工学、50(1)、p.68 - 77、2012 年

30

【非特許文献 2】Bunus P., Fritzon P., "Methods for Structural Analysis and Debugging of Modelica Models", 2nd International Modelica Conference, Proceedings, pp. 157 - 165.

【非特許文献 3】J. Unger, et al., "STRUCTURAL ANALYSIS OF DIFFERENTIAL - ALGEBRAIC EQUATION SYSTEMS - THEORY AND APPLICATIONS", Computer s chem., Engng Vol. 19, No. 8, pp. 867 - 882, 1995.

40

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0007】

特許文献 1 の数値計算プログラム生成システムは、システム内で既に定義された疑似コードを利用して、入力される問題定義情報に応じた数値計算プログラムを生成している。このため、所望の数値計算法のアルゴリズム自体を入力として指定することで、数値計算

50

プログラムを生成させることはできない。

【0008】

非特許文献1の方法では、実行コードを自動生成する対象の数値計算法が、常微分方程式の初期値問題に対する修正Euler法や陽的Runge-Kutta法などの一段階法の一部に限定されている。非特許文献1の方法は、常微分方程式の初期値問題の解法以外の数値計算法や、常微分方程式の初期値問題の解法の中でも例えば陰的Runge-Kutta法や線形多段階法には適用できず、種々の数値計算法の実行コードを自動生成することができなかった。

【0009】

本発明の目的は、数値計算プログラムの生産性を向上することができる計算コード生成装置、方法及びプログラムを提供することである。

10

【課題を解決するための手段】

【0010】

本発明に係る計算コード生成装置は、計算対象とする数学的問題に対する数値計算が実行される計算コードを自動生成する計算コード生成装置である。計算コード生成装置は、記憶部と、制御部とを備える。記憶部は、計算対象の問題に対する数値計算法を数式で宣言的に記述する形式記述情報を格納する。制御部は、記憶部に格納された形式記述情報に基づく演算処理を行う。制御部は、形式記述情報に基づいて、形式記述情報における数式及び数式が含む変数の関係を解析する。制御部は、解析結果に基づいて、所定のプログラミング言語により計算対象とする問題に対する数値計算を実行する計算コードを生成する。形式記述情報は、反復型数値計算法を漸化式で規定する情報を含む。

20

【発明の効果】

【0011】

本発明に係る計算コード生成装置によると、形式記述情報に記述された漸化式に基づき、種々の反復型数値計算法を実行する計算コードが生成可能であり、数値計算プログラムの生産性を向上することができる。

【図面の簡単な説明】

【0012】

【図1】本発明の実施形態1に係る計算コード生成装置による計算コードの自動生成方法の概念を説明するための図

【図2】実施形態1に係る計算コード生成装置の構成を示すブロック図

30

【図3】実施形態1に係る計算コード生成処理を示すフローチャート

【図4】実施形態1における形式記述情報の一例を示す図

【図5】図4に続く形式記述情報の例を示す図

【図6】図5に続く形式記述情報の例を示す図

【図7】計算コード生成処理における形式記述解釈処理を示すフローチャート

【図8】形式記述解釈処理で生成される各種情報を説明するための図

【図9】計算コード生成処理における数式構造解析処理を示すフローチャート

【図10】数式構造解析処理における数式の展開を説明するための図

【図11】数式構造解析処理における二部グラフを説明するための図

【図12】数式構造解析処理で解析される諸情報を説明するための図

40

【図13】計算コード生成処理におけるコード生成処理を示すフローチャート

【図14】実施形態1における形式記述情報の別例を示す図

【図15】図14に続く形式記述情報の例を示す図

【図16】計算コード生成処理における最適化前後の計算コードを示す図

【図17】形式記述情報の変形例を示す図

【図18】変形例の形式記述情報における第1の参照情報を示す図

【図19】変形例の形式記述情報における第2の参照情報を示す図

【発明を実施するための形態】

【0013】

以下、添付の図面を参照して本発明に係る計算コード生成装置、方法及びプログラムの

50

実施の形態を説明する。なお、以下の各実施形態において、同様の構成要素については同一の符号を付している。

【0014】

(実施形態1)

1. 構成

1-1. 概要

本発明の実施形態1に係る計算コード生成装置による計算コードの自動生成方法の概念について、図1を参照して説明する。図1は、本実施形態に係る計算コード生成装置1による計算コードの自動生成方法の概念を説明するための図である。

【0015】

本実施形態に係る計算コード生成装置1は、例えばユーザが指定する数値計算法を記述した形式記述情報Dxを入力し、指定された数値計算法を実行する計算コードCoを自動生成する。

【0016】

形式記述情報Dxは、任意の数値計算法の完全な定義が、形式仕様記述において宣言的に記述された情報である(図4~6参照)。形式記述情報Dxにおいて、数値計算法のアルゴリズムは数式を用いて記述され、数式は例えばMathML形式やOpenMath形式などの形式記述書式により記述される。また、数値計算法が計算対象とする数学的問題の一般形を表す方程式(方程式系)における当該問題を定める関数(以下、「問題定義関数」という。)は、形式記述情報Dx中では変数として記述される。例えば、常微分方程式の初期値問題の一般形「 $dy/dt = f(y, t)$ 」において、問題定義関数はfである。

【0017】

計算コードCoは、所定のプログラミング言語において、形式記述情報Dxで指定された数値計算法による数値計算が行われるように記述されたプログラムコードである。所定のプログラミング言語には、例えばC言語、C++、Java(登録商標)、Fortran, CUDA C, Haskell, Scalaなどが含まれる。本実施形態では、問題定義関数(数学的問題)の具体形は、別途、計算コードCoに対する入力として指定される。

【0018】

本実施形態に係る計算コード生成処理の対象は、常微分方程式の初期値問題の解法や境界値問題の解法、偏微分方程式の解法、(非線形)連立方程式の求根アルゴリズム、最適化問題の解探索アルゴリズム、数値連続法、数値微分法、数値積分法など、数値計算法全般にわたる。特に、数値計算法の内部で反復計算処理を行う反復型数値計算法は、形式記述情報Dxにおいて、反復計算回数を添字集合とする列を有する漸化式として明示的に表現する。計算コード生成処理において漸化式の諸構造を解析することにより、例えば線形多段階法などの複雑な解法を用いる場合でも、計算コードCoを自動生成することができる。

【0019】

また、本実施形態に係る計算コード生成装置1は、計算環境情報Deに応じて、計算コードCoを最適化して出力する。計算環境情報Deは、計算コードCoが実行される計算環境における各種スペック(例えば搭載メモリ量やCPU数)に関する情報である。また、計算コードCoを記述するプログラミング言語も、計算環境情報Deにおいて指定される。本実施形態では、形式記述情報Dxに記述された数式を解析することで、機械的に判定可能な範囲で最適化された計算コードCoが得られる。

【0020】

以下、本実施形態に係る計算コード生成装置1のハードウェア構成について説明する。

【0021】

1-2. ハードウェア構成

図2は、計算コード生成装置1の構成を示すブロック図である。計算コード生成装置1

10

20

30

40

50

は、例えばPC（パーソナルコンピュータ）などの情報処理装置で構成される。計算コード生成装置1は、図2に示すように、制御部10と、記憶部11と、操作部12と、表示部13と、機器インタフェース14と、ネットワークインタフェース15とを備える。

#### 【0022】

制御部10は、例えばソフトウェアと協働して所定の機能を実現するCPUやGPU、MPUで構成され、計算コード生成装置1の全体動作を制御する。制御部10は、記憶部11に格納されたデータやプログラムを読み出して種々の演算処理を行い、各種の機能を実現する。例えば、制御部10は、本実施形態に係る計算コードの自動生成方法が実現される計算コード生成処理を実行する。計算コード生成処理を実行するためのプログラムは、パッケージソフトウェアであってもよい。また、制御部10は、所定の機能を実現するように設計された専用の電子回路や再構成可能な電子回路などのハードウェア回路であってもよい。制御部10は、CPU、MPU、マイコン、DSP、FPGA、ASIC等の種々の半導体集積回路で構成されてもよい。

10

#### 【0023】

記憶部11は、計算コード生成装置1の機能を実現するために必要なプログラム及びデータを記憶する記憶媒体であり、例えばハードディスク（HDD）や半導体記憶装置（SSD）を備える。また、記憶部11は、例えば、DRAMやSRAM等の半導体デバイスを備えてもよく、データを一時的に記憶するとともに制御部10の作業エリアとしても機能する。記憶部11において、形式記述情報Dxや計算環境情報De、計算コード入力リストLin、数値リストL1、数式変数リストL2（後述）などが格納される。

20

#### 【0024】

操作部12は、ユーザが操作を行うユーザインタフェースである。操作部12は、例えば、キーボード、タッチパッド、タッチパネル、ボタン、スイッチ、及びこれらの組み合わせで構成される。操作部12は、ユーザによって入力される諸情報を取得する取得部の一例である。

#### 【0025】

表示部13は、例えば、液晶ディスプレイや有機ELディスプレイで構成される。表示部13は、例えば操作部12から入力された情報など、種々の情報を表示する。

#### 【0026】

機器インタフェース14は、計算コード生成装置1に他の機器を接続するための回路（モジュール）である。機器インタフェース14は、所定の通信規格にしたがい通信を行う。所定の規格には、USB、HDMI（登録商標）、IEEE1395、WiFi、Bluetooth（登録商標）等が含まれる。

30

#### 【0027】

ネットワークインタフェース15は、無線または有線の通信回線を介して計算コード生成装置1をネットワークに接続するための回路（モジュール）である。ネットワークインタフェース15は所定の通信規格に準拠した通信を行う。所定の通信規格には、IEEE802.3、IEEE802.11a/11b/11g/11ac等の通信規格が含まれる。

40

#### 【0028】

### 2. 動作

以下、本実施形態に係る計算コード生成装置1の動作について説明する。

#### 【0029】

### 2-1. 計算コード生成処理

本実施形態に係る計算コード生成装置1によって実行される処理について、図3～6を参照して説明する。図3は、本実施形態に係る計算コード生成処理を示すフローチャートである。図4、5、6は、形式記述情報Dxの一例を示す図である。

#### 【0030】

計算コード生成処理は、所望の数値計算法が形式的に記述された形式記述情報Dxに基づき、数値計算法が計算環境情報Deにより指定される計算環境下で実行される計算コー

50

ドC oを生成する処理である。本処理は、計算コード生成装置1の制御部10によって実行される。

【0031】

まず、計算コード生成装置1の制御部10は、形式記述情報D xに基づいて、形式記述解釈処理を行う(S 1)。形式記述解釈処理では、計算コード生成装置1に入力される形式記述情報D xを取得し、形式記述情報D xにおける数値計算法の入力/出力の指定や反復条件、数値計算法のアルゴリズムを規定する数式の記述の解釈(構文解析)を行う。

【0032】

ここで、形式記述情報D xについて説明する。形式記述情報D xには、所望の数値計算法に対する入力や出力の指定、及び数値計算法のアルゴリズムを規定する数式などの各種情報が、宣言的に記述されている。図4~6に、形式記述情報D xの一例を示す。

10

【0033】

形式記述情報D xは、例えばXML形式で記述される。図4~6に示すように、形式記述情報D xにおいて、記述XAは計算コードC oに対する入力を指定する部分であり、記述XBは計算コードC oによる出力を指定する部分であり、記述XCは計算コードC oで実行する反復計算の反復条件を指定する部分である。また、記述XDは、形式記述情報D xにおいて反復型数値計算法のアルゴリズムを規定する数式を記述したアルゴリズム部分である。記述XEは、数値計算時に使用するパラメータを指定する部分である。記述XA~XC, XEの指定内容については、後述する。

【0034】

20

図4~6においては、反復型数値計算法が常微分方程式( $dy/dt = f(y, t)$ )の初期値問題に対する線形多段階法の一つである2次Adams-Bashforth法(以下、「2次AB法」と略記する。)の場合の一例を示している。本例では、2次AB法にEuler法を組み合わせたアルゴリズムを用いている。本例の反復型数値計算法のアルゴリズムを規定する漸化式は、以下のように表される。

$$y_{i+1} = y_i + h \times (3g_i - g_{i-1}) / 2 \quad (1)$$

$$g_i = f(y_i, t_i) \quad (2)$$

$$t_{i+1} = t_i + h \quad (3)$$

$$y_1 = y_0 + h \times g_0 \quad (4)$$

【0035】

30

上式(1)~(3)は、2次AB法のアルゴリズムを規定している。また、上式(4)において、 $y_1$ の計算にEuler法を用いている。図4~6に示すように、形式記述情報D xのアルゴリズム部分XDにおいて、記述XD1に上式(1)が記述され、記述XD2に上式(2)が記述され、記述XD3に上式(3)が記述され、記述XD4に上式(4)が記述されている。

【0036】

上記の漸化式の組(1)~(4)において、 $i$ は、反復計算における反復回数に対応する添字(インデックス)である。例えば、変数 $y_i$ は、関数 $f(y, t)$ の引数変数 $y$ に対する反復計算 $i$ 回目の値に対応する変数であり、変数 $y_i$ の組 $\{y_0, y_1, \dots, y_i, \dots\}$ は $y$ の反復計算に関する数列を構成する。各変数は、数値(スカラー)であってもよいし、ベクトル、行列、又は集合などであってもよい。また、これに応じて、数列は、各項が数値で構成される列だけでなく、ベクトル列など、あらゆる列を含むものとする。以下、変数 $y_i$ のように、添字が付された変数であって添字に基づく数列の一項を構成する変数を、「添字付き変数」という。

40

【0037】

上式(1)中の添字付き変数 $y_{i+1}$ と $y_i$ のように、同じ数列に属する添字付き変数は、計算コードC o上では別々の変数として扱われることとなるが、計算コードC oの生成時には、反復計算の繰り返し前後の計算ステップ等で関連付けをする必要もある。このため、本実施形態の形式記述解釈処理では、形式記述情報D xにおける漸化式に含まれる種々の変数や、各変数が属する数列に関する情報の分類を行う。形式記述解釈処理の詳細

50

については後述する。

【 0 0 3 8 】

形式記述解釈処理 ( S 1 ) の後に、制御部 1 0 は、形式記述解釈処理 ( S 1 ) の解析結果に基づいて、数式構造解析処理を行う ( S 2 )。数式構造解析処理では、数値計算法のアルゴリズムを規定する数式の組における数式構造 ( 変数を介した数式間の関係 ) を解析し、数値計算法を種々のプログラミング言語で記述可能にするための情報 ( 中間データ ) を生成する。

【 0 0 3 9 】

例えば、2次 A B 法の場合、上記の漸化式の組 ( 1 ) ~ ( 4 ) が数式構造解析処理の対象となる。漸化式は、上式 ( 1 ) ~ ( 4 ) に示すように、あらゆる  $i$  (  $i = 1, 2, \dots$  ) について共通の式である。ここで、漸化式の組における数式間で、添字の相対的な関係を考慮する必要がある場合がある。例えば、上式 ( 1 ) , ( 2 ) において、式 ( 1 ) の右辺の  $g_i$  は式 ( 2 ) により定義されることが明示的だが、式 ( 1 ) の右辺の  $g_{i-1}$  も式 ( 2 ) により定義されることは字句上では明示的でない。そこで、本実施形態の数式構造解析処理では、反復型数値計算法のアルゴリズムを規定する漸化式に対して、数値計算に使用される別々の添字付き変数が明示されるように、数式の展開を行う。数式構造解析処理の詳細については後述する。

【 0 0 4 0 】

次に、制御部 1 0 は、数式構造解析処理 ( S 2 ) の解析結果及び計算環境情報 D e に基づいて、コード生成処理を行う ( S 3 )。コード生成処理では、数式構造解析処理で得られた中間データを、計算環境情報 D e で指定されるプログラミング言語で変換し、計算コード C o を生成する。本実施形態のコード生成処理では、計算環境情報 D e が指定する計算環境に応じて自動化可能な範囲で最適化された計算コードが生成される。

【 0 0 4 1 】

本実施形態では、形式記述情報 D x において数値計算法のアルゴリズムが数式で指定されるため、数式中で並列計算可能な部分や計算結果が共通する部分を容易に判定可能である。特に、反復型数値計算法のアルゴリズムを指定する漸化式では、上記の漸化式 ( 1 ) において反復計算 (  $i - 1$  ) 回目で計算される  $g_{i-1}$  のように、反復計算の過去の計算ステップの計算結果が再利用可能な部分を特定可能である。ここで、 $g_{i-1}$  について計算結果を記憶することと再計算することのいずれが適切であるかは、計算環境に依存する。本実施形態のコード生成処理では、計算環境に応じて上記の部分等の最適化を行う。コード生成処理の詳細については後述する。

【 0 0 4 2 】

以上の各処理の詳細について、以下、説明する。

【 0 0 4 3 】

2 - 1 - 1 . 形式記述解釈処理

図 3 のフローチャートにおける形式記述解釈処理 ( ステップ S 1 ) について、図 4 ~ 8 を参照して説明する。図 7 は、形式記述解釈処理を示すフローチャートである。図 8 は、形式記述解釈処理で生成される各種情報を示す図である。

【 0 0 4 4 】

以下では、形式記述解釈処理について、入力の数値計算法が上述の 2 次 A B 法の一例の場合について説明する。また、以下では、形式記述情報 D x ( 図 4 ~ 6 参照 ) において、漸化式の組 ( 1 ) ~ ( 4 ) を示す宣言的記述 ( X D ) とともに、以下の項目 ( I ) ~ ( I V ) が記述されている例で説明する ( X A ~ X C , X E )。

( I ) 計算コード C o の入力は、変数  $y_0, t_0, f$  である。

( I I ) 計算コード C o の出力は、全反復における変数の組 (  $y_n, t_n$  ) である。

( I I I ) 反復計算の反復条件は、継続条件  $t < T$  である。

( I V ) パラメータ定数は、 $h, T$  である。

【 0 0 4 5 】

なお、反復条件は、反復型数値計算アルゴリズムによる反復計算を繰り返すか否かを判

10

20

30

40

50



定するための条件である。反復条件は、反復計算の継続を判定する継続条件で指定されてもよいし（例えば  $t = T$ ）、反復計算の終了を判定する終了条件で指定されてもよい（例えば  $t > T$ ）。

【0046】

図7のフローチャートにおいて、まず、制御部10は、記憶部11から形式記述情報Dxを取得する(S11)。形式記述情報Dxは、例えば本処理の開始前にユーザによって操作部12(図2)から入力されて、記憶部11に格納されている。なお、形式記述情報Dxにおいて他の情報を参照する記述がある場合、制御部10は参照されている情報も読み出し、形式記述情報Dxの記述に従ってこれらの情報を合成する(図17~19参照)。

10

【0047】

次に、制御部10は、読み出した形式記述情報Dxにおいて計算コードCoに対する入力を指定する記述(項目(I))を解釈し、計算コードCoに対して入力される情報を抽出する(S12)。例えば、上述の2次AB法の一例の形式記述情報Dxには、図4に示すように、計算コードCoの入力の記述XAがある。制御部10は、この記述XAに基づいて、図8(a)に示すように、計算コード入力リストLinを生成する。計算コード入力リストLinは、形式記述情報Dxに基づき生成される計算コードCoの使用時に入力されることとなる情報を管理するリストである。

【0048】

図8(a)に示すように、計算コード入力リストLinは、「入力変数」と、「変数属性」とを関連付けている。入力変数は、計算コードの使用時に入力として指定される変数である。変数属性は、変数の型を示す属性情報である。

20

【0049】

図8(a)に示すように、計算コード入力リストLinには、入力変数 $y_0$ 、 $t_0$ が「定数添字付き変数」という型の変数であり、入力変数fが「関数」を表す変数であることが記録されている。以下、定数だけからなる添字を「定数添字」といい、添字が定数添字である添字付き変数を「定数添字付き変数」という。また、変数を含む添字を「変数添字」といい、添字が変数添字である添字付き変数を「変数添字付き変数」という。

【0050】

次に、制御部10は、形式記述情報Dxにおいて計算コードCoの出力を指定する記述(項目(II))を解釈し、計算コードCoによる数値計算結果として出力される情報を抽出する(S13)。例えば、2次AB法の形式記述情報Dxには、図4に示すように、計算コードCoの出力の記述XBがある。この記述XBから、制御部10は、2次AB法の数値計算結果として、全反復における変数の組( $y_n$ 、 $t_n$ )の評価値が出力されることを判断する。

30

【0051】

次に、制御部10は、形式記述情報Dxにおいて計算コードCoの数値計算アルゴリズムで用いられる反復条件やパラメータ定数を指定する記述(項目(III)、(IV))を解釈する(S14)。制御部10は、例えば計算コード入力リストLinと同様に、形式記述情報Dxに記述されたパラメータ定数をリスト化する。

40

【0052】

例えば、2次AB法の形式記述情報Dxでは、図4に示すように、反復条件の記述XCがある。この記述XCから、制御部10は、反復計算が、 $t = T$ である限り継続されることを判断する。また、図6に示すように、計算コードCoのパラメータ定数h、Tの記述XEがある。制御部10はこの記述XEを抽出して、2次AB法の数値計算時にパラメータ定数h、Tが用いられることを判断する。

【0053】

次に、制御部10は、ステップS15~S18において、形式記述情報Dxにおいて数値計算法のアルゴリズムが記述されたアルゴリズム部分の解釈を行う。

【0054】

50

まず、制御部 10 は、形式記述情報 D x におけるアルゴリズム部分の記述 X D に基づいて、抽象構文木や D O M ツリーなどで数値計算法のアルゴリズムを規定する数式を表現する数式情報を生成する ( S 1 5 )。

【 0 0 5 5 】

次に、制御部 10 は、生成された数式情報から、反復型数値計算アルゴリズムにおける数列を示す情報を抽出し、数列リスト L 1 を生成する ( S 1 6 )。数列リスト L 1 は、反復型数値計算アルゴリズムを規定する 1 つ又は複数の漸化式中の添字付き変数が構成する数列を分類するリストである。図 8 ( b ) に、ステップ S 1 6 で生成される数列リスト L 1 の一例を示す。

【 0 0 5 6 】

図 8 ( b ) に示すように、数列リスト L 1 は、「数列」と、「添字付き変数」と、「変数添字」と、「初期項フラグ」とを関連付けている。初期項は、計算コード C o に対する入力として初期値が設定される数列の最初の 1 つ又は複数の項であり、初期項フラグは、各数列について初期項の入力があるか否かを示すフラグである。

【 0 0 5 7 】

図 8 ( b ) は、上述の 2 次 A B 法の例の数列リスト L 1 を例示しており、この数列リスト L 1 は漸化式の組 ( 1 ) ~ ( 4 ) の記述に基づき生成される。数列リスト L 1 では、例えば変数添字 i を有する添字付き変数  $y_i$  で構成される数列  $\{y_i\}$  に関して、計算コード入力リスト L i n ( 図 8 ( a ) ) の初期項  $y_0$  に応じて、初期項フラグが「ON」に設定されている。添字付き変数  $t_i, g_i$  の数列  $\{t_i\}, \{g_i\}$  についても同様に管理される ( 例えば、計算コード入力リスト L i n に  $g_0$  等がないため、数列  $\{g_i\}$  の初期項フラグは「OFF」に設定されている。 )。

【 0 0 5 8 】

図 7 に戻り、制御部 10 は、漸化式の組に含まれる添字付き変数の変数添字の正規化を行う ( S 1 7 )。変数添字の正規化では、漸化式中の添字付き変数の変数添字の中で最小の変数添字が「0」に設定されるとともに、同式中の変数添字間の相対的な関係 ( 添字相対値 ) を維持するように他の添字付き変数の変数添字が設定される。例えば、漸化式 ( 1 ) は、変数添字の最小値が「i - 1」であるため、以下のように正規化される。

$$y[2] = y[1] + h(3g[1] - g[0]) / 2 \quad (1')$$

【 0 0 5 9 】

次に、制御部 10 は、正規化された漸化式の組における数式毎に、添字付き変数を抽出してリスト化し、数式変数リスト L 2 を生成する ( S 1 8 )。数式変数リスト L 2 は、数値計算法のアルゴリズム部分を記述する漸化式の組の各数式に含まれる別々の変数を分類するリストである。図 8 ( c ) に、ステップ S 1 8 で生成される数式変数リスト L 2 の一例を示す。

【 0 0 6 0 】

図 8 ( c ) に示すように、数式変数リスト L 2 は、「数式」と、複数の変数 ( 「変数 1」、「変数 2」、... ) とを関連付けている。数式変数リスト L 2 では、正規化された漸化式の組において、各の数式に含まれる全ての変数が、添字の違いも考慮して、1 つずつ記録される。例えば、漸化式 ( 1 ) については、ステップ S 1 7 で正規化されていることから ( 上式 ( 1' ) 参照 )、図 8 ( c ) に示すように、4 つの変数  $y[2], y[1], g[1], g[0]$  が記録されている。

【 0 0 6 1 】

以上のように、制御部 10 は形式記述情報 D x 中のアルゴリズム部分の解釈を行う ( S 1 5 ~ S 1 8 )。ステップ S 1 5 で生成された数式情報は、ステップ S 1 6 ~ S 1 8 で得られた各種情報に基づき、制御部 10 によって適宜、更新される。

【 0 0 6 2 】

これにより、制御部 10 は、図 3 のフローチャートにおける形式記述解釈処理 ( ステップ S 1 ) を終了し、数式構造解析処理 ( ステップ S 2 ) に進む。

【 0 0 6 3 】

10

20

30

40

50

以上の処理によると、反復計算回数に対応する添字付き変数が構成する数列や、数式毎に含まれる添字付き変数など、制御部 10 が形式記述情報  $Dx$  において反復型数値計算法を表す漸化式を解析するために必要な情報が整理される。

【0064】

以上の各処理において、制御部 10 は、数式情報や計算コード入力リスト  $L_{in}$ 、数列リスト  $L_1$ 、数式変数リスト  $L_2$  などの各種情報を生成すると、随時、記憶部 11 に保持し、保持した情報を適宜、以降の処理に利用する。

【0065】

なお、以上の説明では、数式構造解析処理（図 3 のステップ  $S_2$ ）よりも前の形式記述解釈処理（ステップ  $S_1$ ）において各種情報を抽出したが、上記情報を事前に抽出せず、数式構造解析処理（ステップ  $S_2$ ）以降の処理において必要の都度、抽出してもよい。

10

【0066】

また、以上の説明では、変数添字の種類が「 $i$ 」の 1 種類のみについて説明したが、これに限らず、複数種類の変数添字があってもよい。例えば、偏微分方程式の解法における有限差分法のように空間的方向を示す添字が、反復計算回数（時間方向）に対応する添字とは別に用いられてもよい。また、変数添字について、反復計算処理が 2 重ループや多重ループになっていてもよい。

【0067】

また、以上の説明では、数列  $\{y_i\}$ 、 $\{t_i\}$  がそれぞれ 1 つの初期項  $y_0$ 、 $t_0$  を有する例について説明したが、初期項は、数列の最初の 1 項に限らず、複数項であってもよい。例えば、数列  $\{y_i\}$  に関して初期項として  $y_0$  と  $y_1$  の値が入力されてもよい。

20

【0068】

また、以上の説明では、2 次  $AB$  法に Euler 法を組み合わせた一例の式 (1) ~ (4) が記述された形式記述情報  $Dx$  に対する形式記述解釈処理について説明した。形式記述情報  $Dx$  には、任意の反復型数値計算を記述することができ、例えば複数の反復型数値計算法を組み合わせで記述されてもよい。また、適宜、式 (2) の  $g_i$  のように、中間変数を導入して、形式記述情報  $Dx$  が記述されてもよい。

【0069】

2 - 1 - 2 . 数式構造解析処理

図 3 のフローチャートにおける数式構造解釈処理（ステップ  $S_2$ ）について、図 9 ~ 11 を参照して説明する。図 9 は、数式構造解析処理を示すフローチャートである。図 10 は、数式構造解析処理における数式の展開を説明するための図である。図 11 は、数式構造解析処理における二部グラフを説明するための図である。

30

【0070】

以下では、解析対象の漸化式が陽的に記述されていても陰的に記述されていても統一的に数式構造を解析可能に実現される数式構造解析処理の一例について説明する。

【0071】

まず、制御部 10 は、数列リスト  $L_1$  及び数式変数リスト  $L_2$  に基づいて、初期項が入力として設定されない数列に属する添字付き変数の添字の正規化後の値を、数式間で揃えるように、漸化式の組を展開する ( $S_{21}$ )。本処理は、数列の添字の値が数式間で特定の相対的關係を有する漸化式の組を、数値計算時に使用され得る数列内の別変数が明示された一連の数式に展開するために行われる。

40

【0072】

ここで、図 10 (a) ~ (c) を用いて、ステップ  $S_{21}$  の処理の一例を説明する。本例では、図 10 (a) の数列リスト  $L_1 A$  に示すように、2 つの数列  $\{x_k\}$ 、 $\{z_k\}$  があり、数列  $\{x_k\}$  の初期項フラグは「ON」である一方、数列  $\{z_k\}$  の初期項フラグは「OFF」である。また、図 10 (b) の数式変数リスト  $L_2 A$  に示すように、2 つの数式  $Eq_1$ 、 $Eq_2$  がある。数式変数リスト  $L_2 A$  によると、（正規化後の）第 1 の数式  $Eq_1$  は添字付き変数  $x[1]$ 、 $z[1]$ 、 $z[0]$  間の関係を表し、（正規化後の）第 2 の数式  $Eq_2$  は添字付き変数  $x[0]$ 、 $z[1]$  間の関係を表していることがわかる

50

。

## 【0073】

ここで、数式変数リストL2Aからは、第1の数式Eq1中に存在する数列 $\{z_k\}$ に属する添字付き変数 $z[1]$ 、 $z[0]$ のうちで、 $z[1]$ が第2の数式Eq2に関連していることは即座に読み取れるが、 $z[0]$ が第2の数式Eq2に関連していることは字句上からだけでは読み取れない。しかし、第2の数式Eq2の実際（正規化前）の変数は $z_{k+1}$ 、 $x_k$ であり（図10(a)）、実際の第2の数式Eq2は $z[0]$ にも関連している。そこで、ステップS21の処理において、制御部10は、第1の数式Eq1が含む初期項フラグが「OFF」である添字付き変数 $z[0]$ の添字の値「0」に揃えるように、第2の数式Eq2を展開する。

10

## 【0074】

具体的には、数式変数リストL2A（図10(b)）において第2の数式Eq2を示すデータを、各添字の値を「-1」だけシフトして複製し、図10(c)に示すように、二つの数式Eq2-1、Eq2-2として数式変数リストL2A'に記録する。これに合わせて、制御部10は、形式記述処理で得られた数式情報等も適宜、更新する。数式（漸化式）Eq1、Eq2を展開した一連の数式Eq1、Eq2-1、Eq2-2の内の数式Eq2-2には、 $z[0]$ とともに $x[-1]$ も含まれる（図10(c)）。このように、数式変数リストL2A'で示される展開後の一連の数式には、各数列中で数値計算時に使用され得る別々の変数が、全て出揃うこととなる。

20

## 【0075】

図9に戻り、制御部10は、数式変数リストL2A'で示される展開後の一連の数式に基づき、例えば各数式と添字付き変数との間の二部グラフを用いて、数式間の変数を介した依存関係を解析する（S22）。ステップS21、S22の処理により、各数式における添字の値の範囲が解析される。

## 【0076】

図11(a)、(b)は、ステップS22の処理で用いられる二部グラフの例を示す。図11(a)に示すように、二部グラフは、第1の点群P1と、第2の点群P2と、第1及び第2の点群P1、P2の間に引かれる線とで構成される。図11(a)に示す二部グラフにおいて、第1の点群P1は、図10(c)に示す展開後の数式変数リストL2A'における各数式Eq1、Eq2-1、Eq2-2対応する3つの点からなる。また、第2の点群P2は、数式変数リストL2A'において互いに異なる全ての添字付き変数 $x[1]$ 、 $z[1]$ 、 $z[0]$ 、 $x[0]$ 、 $x[-1]$ に対応する5つの点からなる。

30

## 【0077】

制御部10は、ステップS22の処理において、数式変数リストL2A'に基づいて、図11(a)の二部グラフを配列等で生成する。図11(a)の二部グラフにおいて、第1及び第2の点群P1、P2の間の線は、数式Eq1、Eq2-1、Eq2-2と、各数式を構成する変数 $x[1]$ 、 $z[1]$ 、 $z[0]$ 、 $x[0]$ 、 $x[-1]$ とを結んだものである。例えば、数式Eq1には、添字付き変数 $x[1]$ 、 $z[1]$ 、 $z[0]$ が現れるため（図10(c)参照）、数式Eq1に対応する点と、上記変数に対応する3点との間に線が引かれている。図11(a)の二部グラフによると、例えば、第1の数式Eq1における $x[1]$ は、数式Eq2-1、Eq2-2を介して、 $x[0]$ 及び $x[-1]$ （計算ステップが1ステップ前及び2ステップ前のデータ）を用いて計算されることが判明する。

40

## 【0078】

次に、制御部10は、形式記述処理で得た計算コード入力リストLin等の情報に基づいて、図11(b)に示すように、第1の点群P1に、初期項を含む定数添字付き変数に対応する点 $(x_0, x_1)$ を追加する。これとともに、制御部10は、第1の点群P1に追加した点と、追加した点に対して第2の点群P2内で所定の関係を有する点（例えば、追加した点に対応する定数添字付き変数と同じ数列に属する変数添字付き変数に対応する点）との間に線を引くように、二部グラフを更新する。図11(b)の例では、第1の点

50

群 P 1 に追加した 2 点の初期項  $x_0$  ,  $x_1$  と同じ数列  $\{x_k\}$  の変数  $x[1]$  ,  $x[0]$  ,  $x[-1]$  の内の、添字の値が小さい 2 つ ( $x[-1]$  ,  $x[0]$ ) の点と、追加した 2 点 ( $x_0$  ,  $x_1$ ) との間に線が引かれている。

【0079】

次に、制御部 10 は、更新した二部グラフの最大マッチングを求め、最大マッチングに基づき、展開後の数式の中で連立方程式として解く必要がある数式の集合を示す連立方程式情報と、計算を行う順序に関する数式間の半順序関係を示す計算順序情報とを導出する (S 23)。計算順序情報が示す数式間の半順序関係は、数値計算時に順番どおりに計算する必要がある数式間の順序の関係だけでなく、独立に計算可能な数式の組み合わせの関係も含む。上記の二部グラフの最大マッチングを求めることにより、連立方程式情報及び計算順序情報は既存の手法で導出することができる (例えば、非特許文献 2 , 3 参照)。

10

【0080】

次に、制御部 10 は、漸化式中の添字の範囲の解析結果に基づき、変数添字付き変数を算出可能な数式を抽出することにより、ループ部分情報を取得する (S 24)。ループ部分情報は、反復計算において計算の繰り返しになるループ部分を示す情報である。例えば、制御部 10 は、数式変数リスト  $L_{2A}$  ,  $L_{2A}'$ 、更新された二部グラフ (図 10 (b) , 11 (b) 参照) 及び数式間の半順序関係に基づき、一連の数式において数式 Eq 1、Eq 2 をループ部分情報として抽出する。なお、ループ部分情報は、ステップ S 23 で計算順序情報と同時に抽出されてもよい。ループ部分情報は、既存の手法 (例えば、非特許文献 2 , 3 参照) を利用し、半順序関係から添字をずらすことで反復計算できる部分として抽出できる。

20

【0081】

次に、制御部 10 は、連立方程式情報、計算順序情報及びループ部分情報並びに形式記述解釈処理で得た各種情報に基づき、種々のプログラミング言語に変換可能な中間形式で数値計算法を表す中間データを生成する (S 25)。中間データは、連立方程式情報、計算順序情報及びループ部分情報を含む。

【0082】

具体的に、制御部 10 は、展開後の数式中で、連立方程式情報が示す数式の集合を特定し、中間形式において、特定した数式の集合を連立方程式として解くことを設定する。さらに、制御部 10 は、計算順序情報に基づく計算順序において、一連の数式の半順序関係を中間データとして保存する。さらに、制御部 10 は、ループ部分情報及び数列リスト  $L_1$  に基づき、反復計算のためのループ部分において、繰り返し前後の計算ステップ間の各変数を対応付け、中間データを生成する。

30

【0083】

これにより、制御部 10 は、図 3 のフローチャートにおける数式構造解析処理 (ステップ S 2) を終了し、コード生成処理 (ステップ S 3) に進む。

【0084】

以上の処理によると、数列中の変数間で共通の漸化式の組を展開することで、その数式構造を解析でき、反復型数値計算法を実現する中間データを得ることができる。また、計算環境に依存しない中間形式の中間データを生成することで、種々の計算環境に応じた計算コード  $C_0$  が生成可能になる。

40

【0085】

例えば、ステップ S 23 で連立方程式として解く必要がある数式の集合が存在した場合、制御部 10 は、コード生成処理 (図 3 のステップ S 3) において中間データに基づいて、非線形方程式の求根アルゴリズムを用いてこの漸化式を解くことを判断できる。この場合、例えば、制御部 10 は、予め記憶部 11 に記録された非線形方程式の求根アルゴリズムを記述する情報を参照して、上記の漸化式を当該求根アルゴリズムの計算対象とするように、適宜、計算コードに組み込む。

【0086】

また、ステップ S 21 の処理により漸化式が陰的であっても陽的であっても同様に展開

50

され、陰的な漸化式を含む漸化式の組に対する数値計算が可能な中間データを生成することができる。

【0087】

以上の説明では、漸化式が陽的に記述されていても陰的に記述されていても統一的に数式構造を解析可能な方法について説明した。数式構造解析処理では、添字が最大の添字付き変数に対して陽的に記述可能である陽的な漸化式と、陽的に記述不可能である陰的な漸化式とについて、異なる方法で数式構造が解析されてもよい。以下、この方法の一例を説明する。

【0088】

まず、数式構造解析処理の簡略化のため、形式記述情報  $Dx$  において、陽的な漸化式は下記の式(5)のように記述され、陰的な漸化式は下記の式(6)のように記述されることとする( $F$ は任意の関数)。

$$x_{i+1} = F(x_i, x_{i-1}, \dots) \quad (5)$$

$$F(x_{i+1}, x_i, \dots) = 0 \quad (6)$$

【0089】

この場合、制御部10は、例えば、形式記述情報  $Dx$  における漸化式の組の中に、右辺が「0」の数式があるか否かによって、陰的な漸化式があるか否かを判断することができる。

【0090】

ここで、式(5)の陽的な漸化式の記述によると、右辺を計算することにより左辺の変数の値が算出できることが明らかである。このため、漸化式の展開(ステップS21)において、制御部10は、陽的な漸化式に対しては右辺の中で、添字付き変数の添字の正規化後の値を合わせる必要がある変数があるか否かを判断すればよい。また、制御部10は、特定の変数に揃えるように数式を展開する必要がある場合、陽的な漸化式については左辺の変数が特定の変数と添字の値だけずれている場合にのみ、添字の値をシフトするように展開すればよい。

【0091】

例えば、図7の形式記述解釈処理(図3のS1)の説明で用いた2次AB法の例では、数式構造解析処理(S2)において、陽的な漸化式の組(1)~(4)は以下のように展開される。

$$y_{i+1} = y_i + h(3g_i - g_{i-1})/2 \quad (1)$$

$$g_i = f(y_i, t_i) \quad (2-1)$$

$$g_{i-1} = f(y_{i-1}, t_{i-1}) \quad (2-2)$$

$$t_{i+1} = t_i + h \quad (3)$$

$$y_1 = y_0 + hg_0 \quad (4)$$

【0092】

上式では、式(1)の右辺の  $g_i, g_{i-1}$  に合わせて、左辺が  $g_i$  であった式(2)が式(2-1), (2-2)に展開されている。以上のように、陽的な漸化式に対しては右辺と左辺との役割を設定して展開を行うことで、数値計算に用いることがない無駄な展開を行わずに、効率的に数式構造を解析することができる。

【0093】

図12は、式(1)~(4)の例において、数式構造解析処理で解析される諸情報(計算順序情報  $Dorder$  及びループ部分情報  $Dloop$ )を説明するための図である。同図が示す情報は、展開後の式(1), (2-1), (2-2), (3), (4)に対してステップS23, S24と同様に最大マッチングを求めることなどによって得られる。

【0094】

図12では、式(1)~(4)における正規化後の各変数の依存関係を矢印で示している。例えば、 $y[2]$ の計算(すなわち式(1)の計算)は、 $y[1], g[1], g[0]$ に依存していることを表している。これらの矢印で示された(枝分かれを含む)順番が、数式間の半順序関係すなわち計算順序情報  $Dorder$  に対応している。例えば、 $y$

10

20

30

40

50

[ 2 ] の計算 ( 式 ( 1 ) ) の前に、 $g [ 1 ]$  の計算 ( 式 ( 2 - 1 ) ) 及び  $g [ 0 ]$  の計算 ( 式 ( 2 - 2 ) ) をする必要があるので計算順序情報  $D o r d e r$  に含まれている。

【 0 0 9 5 】

また、図 1 2 において、破線で囲んだ範囲がループ部分情報  $D l o o p$  に対応している。つまり、破線で囲まれた範囲に含まれる計算が反復計算において繰り返されることとなる。ループ部分情報  $D l o o p$  は、 $y [ 1 ]$  ,  $t [ 0 ]$  ,  $y [ 0 ]$  が定数添字変数  $y_1$  ,  $g_0$  ,  $t_0$ 、或いは以前の計算ステップにおいて計算された変数から添字をずらしたものである、ということから抽出することができる。

【 0 0 9 6 】

以上の解析結果を含む中間データが、制御部 1 0 により、ステップ S 2 5 で生成される。

【 0 0 9 7 】

また、以上のように生成された中間データや各処理における解析結果は、随時、記憶部 1 1 に保持され、以降の処理に適宜、利用される。

【 0 0 9 8 】

2 - 1 - 3 . コード生成処理

図 3 のフローチャートにおけるコード生成処理 ( ステップ S 3 ) について、図 1 3 ~ 1 6 を参照して説明する。コード生成処理 ( S 3 ) では、数式構造解析処理 ( S 2 ) により生成された中間データを計算環境に応じて最適化して、最終的な計算コード  $C o$  を生成する。以下では、手続型プログラミング言語で計算コードが生成され、最適化される例について説明する。

【 0 0 9 9 】

図 1 3 は、コード生成処理を示すフローチャートである。図 1 4 , 1 5 は、本実施形態における形式記述情報の別例を示す図である。図 1 6 は、コード生成処理における最適化前後の計算コードを示す図である。

【 0 1 0 0 】

以下では、コード生成処理について、入力の数値計算法が、非線形連立方程式の求根アルゴリズムの一種の割線法である例で説明する。図 1 4 , 1 5 に、数値計算法が割線法である例の形式記述情報  $D x a$  を示す。図 1 4 , 1 5 に示す形式記述情報  $D x a$  には、割線法を規定する以下の漸化式を示す情報が記述されている。

$$x_{i+1} = x_i - f(x_i)(x_i - x_{i-1}) / (f(x_i) - f(x_{i-1}))$$

( 7 )

【 0 1 0 1 】

以下では、図 1 4 , 1 5 の形式記述情報  $D x a$  に基づき図 3 のステップ S 1 , S 2 の形式記述解釈処理及び数式構造解析処理が行われた後に、ステップ S 3 のコード生成処理が開始したこととする。

【 0 1 0 2 】

図 1 3 のフローチャートにおいて、まず、制御部 1 0 は、計算環境情報  $D e$  ( 図 1 ) を取得する ( S 3 0 ) 。

【 0 1 0 3 】

計算環境情報  $D e$  は、計算コード  $C o$  が実行される計算環境に応じた種々の指定を含む情報である。種々の指定は、例えば、計算コード  $C o$  を記述する手続型プログラミング言語の指定や、計算コード  $C o$  の実行時に ( 最大 ) 幾つに並列化して処理を実行するかの指定 ( 並列化の指定 )、メモリの使用と計算処理のいずれを優先するのかの指定 ( メモリ使用 / 計算処理の優先指定 ) を含む。並列化の指定は、例えば計算環境における CPU 数や分散ネットワーク環境を考慮して設定される。メモリ使用 / 計算処理の優先指定は、例えばメモリの搭載量及びメモリのアクセス速度、並びに搭載 CPU の性能に対するメモリ性能の相対評価等を考慮して設定される。

【 0 1 0 4 】

10

20

30

40

50

なお、計算環境情報  $D_e$  は、計算コード  $C_o$  が実行される計算環境の各種スペックを示す情報であってもよく、制御部 10 が計算環境情報  $D_e$  における各種スペックに基づき上記の指定を判断するようにしてもよい。また、計算環境情報  $D_e$  は、図 3 の計算コード生成処理の開始前からあらかじめ記憶部 11 に格納されていてもよいし、図 3 のステップ S2 の数式構造解析処理後に、ユーザによって操作部 12 (図 2) から入力されてもよい。

#### 【0105】

次に、制御部 10 は、数式構造解析処理 (S2) で生成された中間データ内の記述を、計算環境情報  $D_e$  において指定された手続き型プログラミング言語の記述に置き換えて、(最適化前の) 計算コード  $C_o'$  を生成する (S31)。図 16 (a) に、ステップ S30 の処理で生成される計算コード  $C_o'$  の一例を示す。

10

#### 【0106】

次に、制御部 10 は、計算環境情報  $D_e$  において、メモリ使用 / 計算処理の優先指定のうちで、メモリ使用の優先指定がされているか否かを判断する (S32)。

#### 【0107】

制御部 10 は、メモリ使用の優先指定がされている場合 (S32 で Yes)、数式構造解析処理の解析結果に基づいて、計算コード中で、計算結果が共通する共通部分に対する一時保存コードを生成する (S33)。一時保存コードは、中間変数等を用いて共通部分の計算結果を一時的に保存するコードである。

#### 【0108】

具体的には、制御部 10 は、中間データを構成する (展開後の) 全数式間または各数式の内部で、共通の部分式を検出する。共通の部分式は、添字付き変数の添字相対値まで完全に一致する部分式だけでなく、添字付き変数の添字相対値が全体的にシフトしている部分式も含む。制御部 10 は、検出した部分式の評価値が格納される中間変数を有する一時保存コードを生成する。一時保存コードは、添字相対値がシフトしている共通部分に対しては、シフト幅の分、反復計算の計算ステップを跨いで一時保存するように記述される。

20

#### 【0109】

例えば、式 (7) においては 2 箇所の  $f(x_i)$  が共通しており、図 16 (a) に示すように、最適化前の計算コード  $C_o'$  上では、11 行目において 2 回、同じ評価値となる関数呼び出しを行っている。また、式 (7) には、 $f(x_i)$  から添字の値がシフトした  $f(x_{i-1})$  があり、この評価値は、反復計算の計算ステップが 1 つ前の  $f(x_i)$  の評価値と同一である。このため、図 16 (b) に示すように、最適化後の計算コード  $C_o$  上で  $f(x_i)$  の関数呼び出しの戻り値を一時保存する一時保存コード (19 行目) が用いられることで、使用メモリ量を増やす代わりに計算量を削減することができる。

30

#### 【0110】

一方、実行環境や関数の内容によっては、関数呼び出しの方が有利な場合もある。このため、制御部 10 は、計算環境情報  $D_e$  において、メモリ使用の優先指定がされていない場合 (S32 で No)、ステップ S33 の処理を行うことなく、ステップ S34 の処理に進む。

#### 【0111】

次に、制御部 10 は、計算環境情報  $D_e$  において、(2 並列以上の) 並列化の指定がされているか否かを判断する (S34)。

40

#### 【0112】

制御部 10 は、並列化の指定がされている場合 (S34 で Yes)、数式構造解析処理の解析結果に基づいて、数式上で並列に計算可能な部分を判別し、計算コードを並列化する (S35)。例えば、計算順序情報において独立に計算できる数式同士や、数式情報中の数式内で「+」や「-」を介して結合している部分式同士は並列に計算可能であり、制御部 10 は、計算順序情報や数式内の算術演算子に基づき並列計算可能な部分式を判別する。また、制御部 10 は、数式構造解析処理で抽出された連立方程式情報に基づき、中間データを構成する全数式間で、連立方程式として解く必要のない、独立に計算可能な数式を判別する。制御部 10 は、判別した部分式や数式について、計算環境情報  $D_e$  において

50



指定された並列化の限度に応じて、計算コードの並列化を行う。

【0113】

一方、制御部10は、計算環境情報において並列化の指定がない場合(S34でNo)、ステップS35の処理を行うことなく、ステップS36の処理に進む。

【0114】

次に、制御部10は、計算環境情報Deに基づいて、計算コードCo中で省略可能な部分(不要部分)の整理を行う(S36)。例えば、制御部10は、計算コードCo中で使い回すことのない不要な中間変数の定義と、その中間変数を介した代入文を1文にまとめる。例えば、図16(a)では、初期項 $x_0$ 、 $x_1$ を示す入力変数 $i_0$ 、 $i_1$ の値は数値計算開始後に参照されないため、この値を保存する必要はなく上書き可能である。制御部10は、数式構造解析処理(S2)で生成された中間データに基づき変数 $i_0$ 、 $i_1$ が不要(省略可能)と判断する。制御部10は、代入文 $x_0 = i_0$ 、 $x = i_1$ (図16(a)の7、8行目)を削除して、直接、入力変数 $x_0$ 、 $x_1$ を反復計算に使用するように、計算コードCoを整理する(図16(b)3、14-17行目参照)。なお、計算環境におけるプログラミング言語やコンパイラによっては、引数変数を使い回さない方が有利な場合もあるため、本処理における不要部分の整理方法は、計算環境情報Deに応じて適宜、設定される。

10

【0115】

次に、制御部10は、計算順序情報と計算環境情報Deに基づいて、計算順序の最適化を行う(S37)。具体的には、計算順序に任意性のある部分についてメモリ局所参照性やパイプライン処理、多重処理、並列処理などを考慮して、計算コードCo上の計算順序の最適化を行う。本処理により、計算順序の最適化において実現可能な全ての計算順序について実行性能を比較評価するような手間を掛けることなく、既存の探索手法を用いて効率的に最適または準最適な計算順序を得ることができる。

20

【0116】

次に、制御部10は、計算環境情報Deに基づき、計算コードCo中で、形式記述情報Dxaの反復条件に対応する反復計算の終了判定を行う箇所を最適な箇所に変更する(S38)。例えば、図16(b)に示すように、反復計算のループ(11~20行)中で終了条件の判定を可能な限り前方(12行目)で行うことで、無駄な計算を削減することができる。なお、計算環境における数値計算の実行条件によっては、最適化前の位置(図16(a)の10~17行中の13行目)での評価が有利な場合もあるため、本処理の最適化方法は、計算環境情報Deで指定される条件に応じて適宜、設定される。

30

【0117】

その後、制御部10は、以上の処理により最適化された計算コードCoを記憶部11等に出力し(S39)、本処理を終了する。

【0118】

以上の処理により、計算環境情報Deで指定される種々の計算環境に応じて、数式構造解析処理の解析結果に基づき自動的に判定可能な範囲で最適化がなされた計算コードCoを、効率的に得ることができる。また、計算環境情報Deにおいて並列化の指定がある場合には、その指定に応じて並列化された計算コードCoが生成される。

40

【0119】

3.まとめ

以上のように、本実施形態に係る計算コード生成装置1は、計算対象の関数に対する数値計算が実行される計算コードCoを自動生成する。計算コード生成装置1は、記憶部11と、制御部10とを備える。記憶部11は、計算対象の関数に対する数値計算法を数式で宣言的に記述する形式記述情報Dxを格納する。制御部10は、記憶部11に格納された形式記述情報Dxに基づく演算処理を行う。制御部10は、形式記述情報Dxに基づいて、形式記述情報Dxにおける数式及び数式が含む変数の関係を解析する(S2)。制御部10は、解析結果に基づいて、所定のプログラミング言語により計算対象の関数に対する数値計算を実行する計算コードCoを生成する(S3)。形式記述情報Dxは、反復型

50

数値計算法を漸化式で規定する情報を含む（図4～6等参照）。

【0120】

本実施形態に係る計算コード生成装置1によると、形式記述情報Dxに記述された漸化式に基づき、あらゆる反復型数値計算法を実行する計算コードCoが生成可能である。また、入力された形式記述情報Dxでは、数値計算法全般を同一形式で統一的に記述することができる。これにより、数値計算プログラムの生産性を向上することができる。

【0121】

また、本実施形態において、制御部10は、形式記述情報Dxにおける漸化式を、漸化式に含まれる（数列中の）変数を介して関連する一連の数式に展開する（S23）。数列は、反復型数値計算法における反復計算回数に対応する添字を有する変数の列で構成される。これにより、計算コード生成装置1は、数列中の変数間で共通の漸化式の組における数式と変数との関係を解析でき、反復型数値計算法を実現するための情報を得ることができる。

10

【0122】

また、本実施形態において、制御部10は、形式記述情報Dxにおける数式毎に、数式に含まれる（数列中の）変数を分類して、漸化式を示す情報を解釈する（S1）。これにより、形式記述情報Dxにおいて反復型数値計算法を表す漸化式を解析するために必要な情報が整理される。

【0123】

また、本実施形態において、制御部10は、計算コードCoが実行される計算環境を示す計算環境情報Deを取得する。制御部10は、計算環境情報Deに応じて最適化して、計算コードCoを出力する。これにより、計算環境情報Deで指定される種々の計算環境に応じて、数式から自動的に判定可能な範囲で最適化がなされた計算コードCoを、効率的に得ることができる。

20

【0124】

また、本実施形態において、形式記述情報Dxにおける漸化式（1）～（4）は、少なくとも3回にわたる反復計算回数にそれぞれ対応する少なくとも3つの変数 $y_{i+1}$ 、 $y_i$ 、 $g_i$ 、 $g_{i-1}$ を含む。これにより、i回目の計算ステップで $y_{i+1}$ の評価値を計算するために、 $g_{i-1}$ において過去の計算結果を用いるのか、再計算するのか、適宜選択して計算コードCoを生成することができる。

30

【0125】

また、本実施形態において、生成部S3によって生成された計算コードCoの外部から、数学的問題（問題定義関数関数f）の具体形が指定される。これにより、生成された計算コードCoは、種々の問題定義関数を計算するための数値計算ライブラリ等として使用することができる。

【0126】

（他の実施形態）

上記の実施形態1では、計算コード生成装置1がPCなどの情報処理装置で構成される例について説明したが、これに限らず、例えば、計算コード生成装置1はASPサーバなどのサーバ装置であってもよい。例えば、計算コード生成装置1は、ネットワークを介して入力された形式記述情報Dxや計算環境情報Deをネットワークインタフェース（取得部の一例）により取得して、計算コード生成処理を実行してもよい。また、計算コード生成装置1は、計算コード生成処理において生成した計算コードCoを、ネットワークを介して送信してもよい。

40

【0127】

また、計算コードの自動生成方法は、クラスタPCやクラウドコンピューティングなどにおいて実行されてもよい。

【0128】

また、計算コードの自動生成方法が実現されるプログラムは、例えば、CAE、CADソフトウェアに組み込まれてもよいし、数式処理ソフトウェア、数値解析ソフトウェアに

50

において利用されてもよい。

【0129】

また、上記の実施形態1では、2次AB法や割線法などの一種の反復型数値計算法が記述された形式記述情報Dx, Dx aの例について説明した。反復型数値計算法においては、その内部で別の反復型数値計算法を用いることがあり、形式記述情報Dxは、数値計算法及びその一部を部分的に交換可能にするように、モジュール化して記述されてもよい。例えば、数値計算法及びその内部で用いられる数値計算法の種別毎に共通インタフェースを定義した形式言語を用いて形式記述情報Dxが記述されてもよい。

【0130】

例えば、形式記述情報Dxにおいて、非線形方程式の求根アルゴリズムの記述中で、数値微分を用いることを抽象的に記述し、具体的に利用する数値微分法は形式記述情報Dxの外部から指定するようにしてもよい。また、数値計算法の内部で別の数値計算法を利用することについて、形式記述情報Dx上で明示的には記述せず、計算コード生成処理において計算コード生成装置1が判断してもよい。図17~19を用いてこれを説明する。

10

【0131】

図17は、数値計算法がニュートン法(内部で数値微分法を用いる非線形方程式の求根アルゴリズムの一種)である例の形式記述情報Dxbを示す。形式記述情報Dxbには、図17に示すように、アルゴリズム部分を形式的に記述した第1の参照情報Dxb1(図18)と、反復計算の終了条件を形式的に記述した第2の参照情報Dxb2(図19)とを参照する記述がある。計算コード生成装置1の制御部10は、図7の形式記述解釈処理のステップS11において、上記の記述に基づき各参照情報Db1, Db2を読み出して、形式記述情報Dxbに合成する。

20

【0132】

また、図18に示すように、第1の参照情報Dxb1には、ニュートン法を規定する以下の漸化式を示す情報が記述されている。

$$x_{i+1} = x_i - f(x_i) / f'(x_i) \quad (8)$$

【0133】

上式(8)には、関数f(x<sub>i</sub>)の導関数f'(x<sub>i</sub>)(=df(x<sub>i</sub>)/dx)が含まれている。制御部10は、例えば図7のステップS11において、第1の参照情報Dxb1中の導関数の記述(図18)に基づき数値微分法を用いることを判断する。この場合、制御部10は、記憶部11に予め記憶された所定の数値微分法を示す形式記述情報を読み出して、形式記述情報Dxbに合成する。

30

【0134】

以上のように数値計算法の部分処理や部分解法(例えば収束判定法や数値微分)の記述をモジュール化することにより、入力する数値計算法の部分的変更を容易化することができる。また、多数の部分解法を組み合わせる複雑な数値計算法も容易に記述可能になる。

【0135】

また、上記の実施形態1では、計算対象の数学的問題を定める問題定義関数の具体形は、外部から計算コードCoに対する入力として指定された。しかし、これに限らず、問題定義関数(数学的問題)の具体形は、計算コードCoの内部に組み込まれてもよい。この場合、図3の計算コード生成処理において、問題定義関数の具体形は、形式記述情報Dxとともに、ステップS1において制御部10に入力される。また、ステップS2の数式構造解析処理では、図9のステップS21において、問題定義関数の方程式系についても数式が展開される。

40

【0136】

例えば、常微分方程式の初期値問題における問題定義関数の具体形として、生体機能モデルの一種のFitzHugh-Nagumoモデルは、次式のように表される。

$$dv/dt = v - v^3/3 - w + I \quad (= f^1(y)) \quad (9)$$

$$dw/dt = (v + a - b \times w) / \tau \quad (= f^2(y)) \quad (10)$$

【0137】

50

上式(9), (10)において、 $I, a, b,$  は定数変数である。また、関数変数  $f$  及び引数変数  $y$  は、2成分ベクトルであり、例えば  $y = (v, w)^T$  である ( $T$  は転置記号)。上式(9), (10)を示す情報が形式記述情報  $Dx$  とともに入力されることで、図9のステップS21において、例えば式(2)は、次式のように展開される。

$$g_i = (v_i - v_i^3 / 3 - w_i + I, (v_i + a - b \times w_i) / )^T \quad (2')$$

【0138】

上式(2')のように、問題定義関数の方程式系についても展開された一連の数式に対して図9のステップS22以降の解析が行われることで、図3のステップS3の計算コード生成処理では、問題定義関数の具体形に応じた最適化を行うことができる。

【0139】

また、上記の実施形態1では、図9の数式構造解析処理のステップS22において、二部グラフを用いて展開後の数式間の変数を介した依存関係を解析し、連立方程式情報、計算順序情報及びループ部分情報を取得する例について説明したが、二部グラフを用いなくてもよい。展開後の数式構造について、既存手法の組み合わせにより様々な解析方法で解析することで、上記の各種情報を取得してもよい。

【産業上の利用可能性】

【0140】

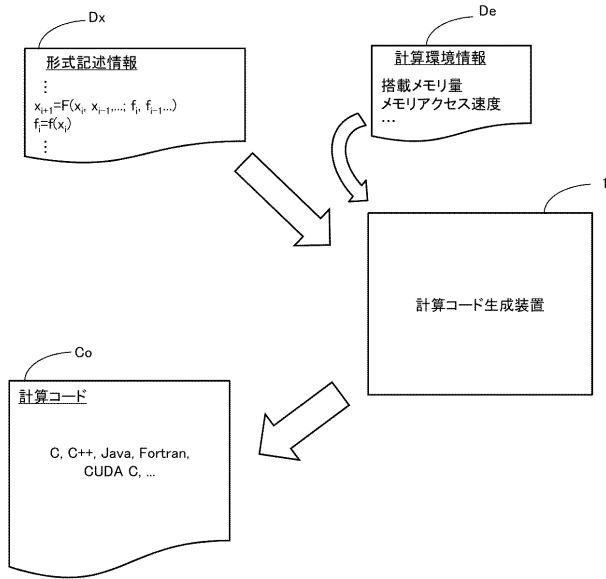
本発明は、数値計算を利用するあらゆる産業分野に適用可能であり、例えば、CAE, CADソフトウェアへの組み込みや、数式処理ソフトウェア、数値解析ソフトウェアにおける利用が可能である。

【符号の説明】

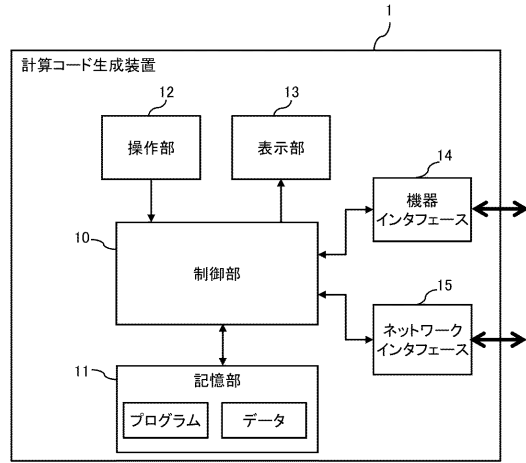
【0141】

- 1 計算コード生成装置
- 10 制御部
- 11 記憶部
- $Dx, Dx a, Dx b$  形式記述情報
- $De$  計算環境情報
- $Co$  計算コード

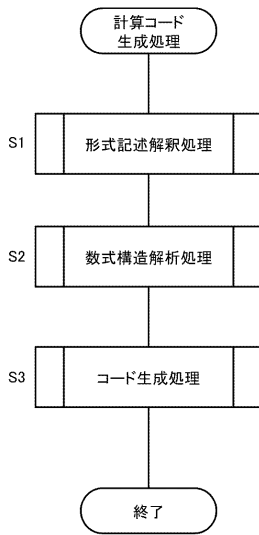
【 図 1 】



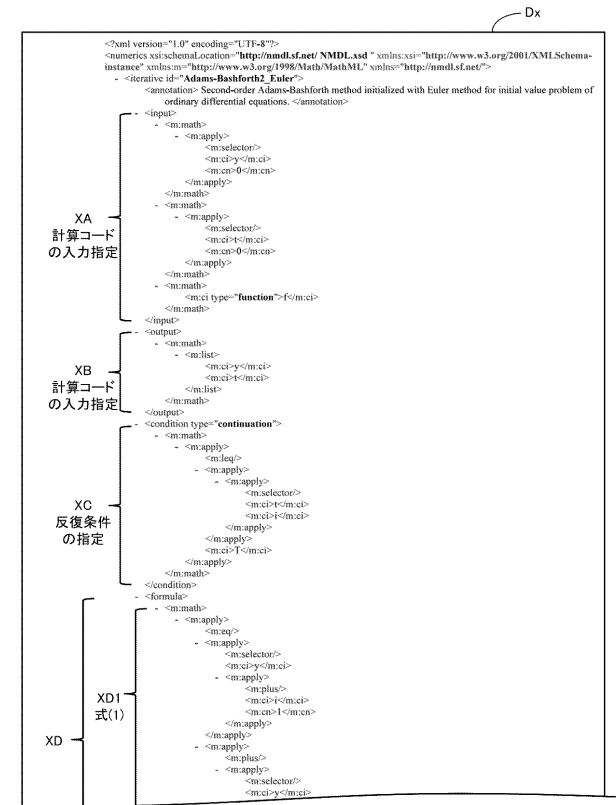
【 図 2 】



【 図 3 】

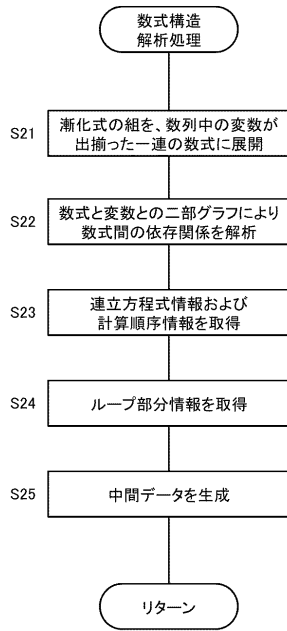


【 図 4 】

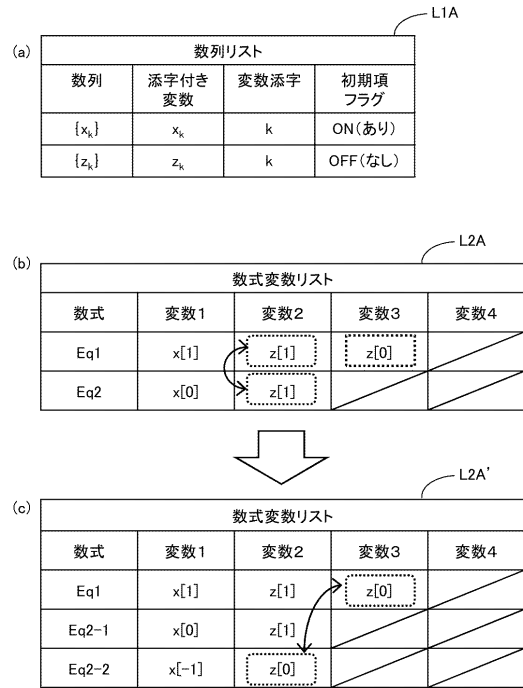




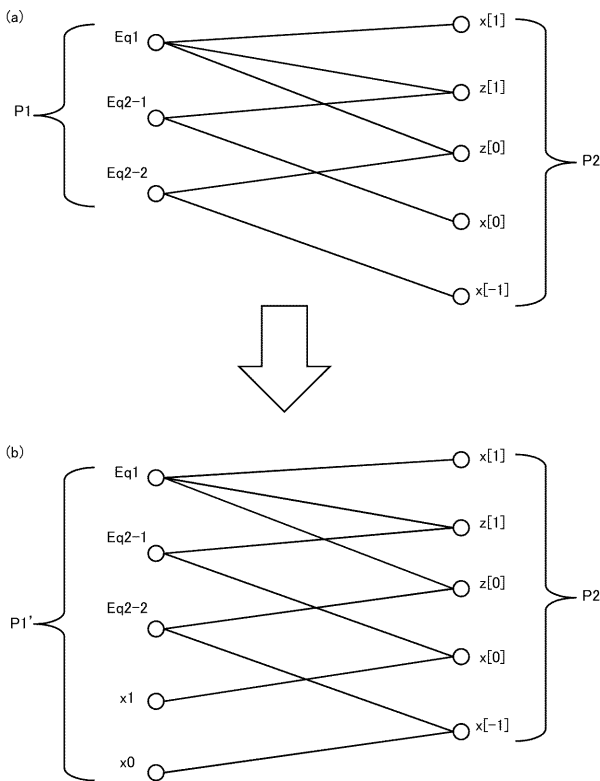
【 図 9 】



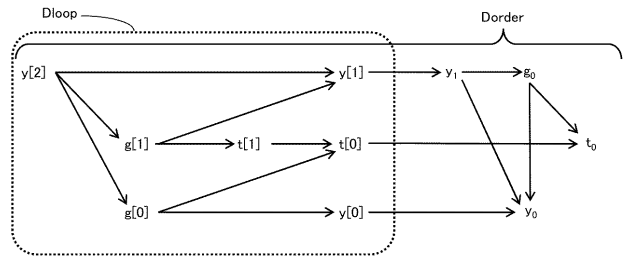
【 図 1 0 】



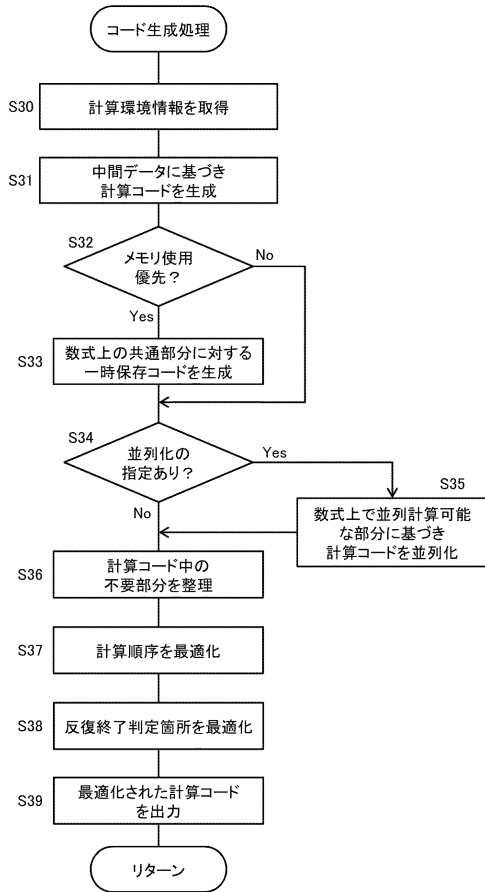
【 図 1 1 】



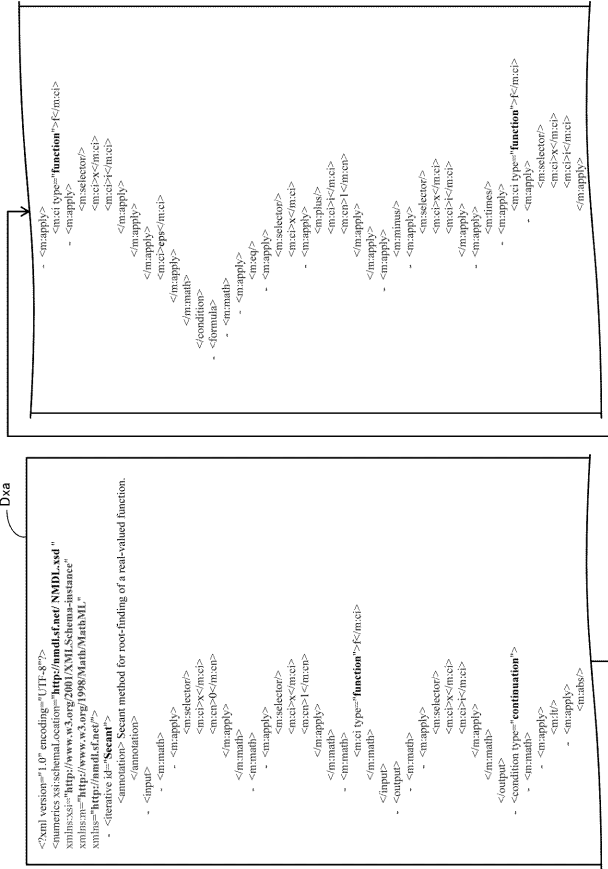
【 図 1 2 】



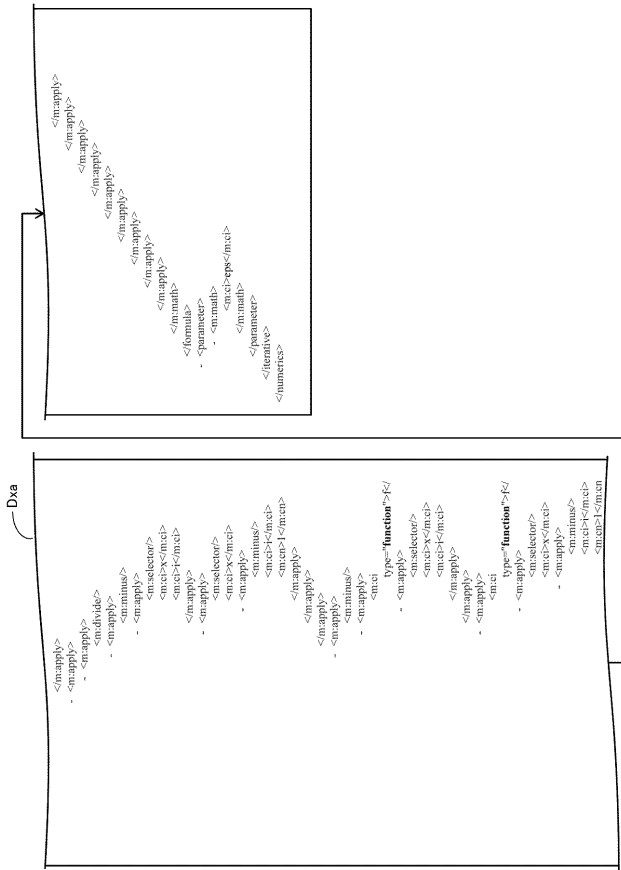
【 図 1 3 】



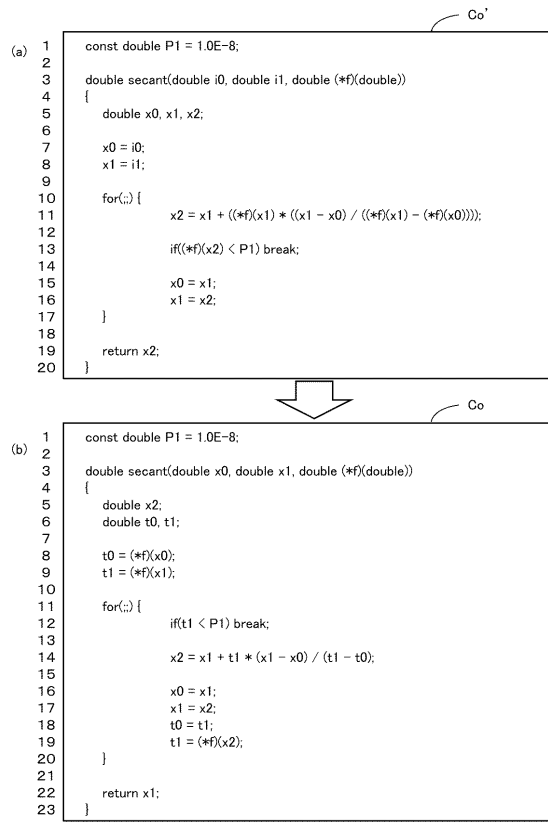
【 図 1 4 】



【 図 1 5 】

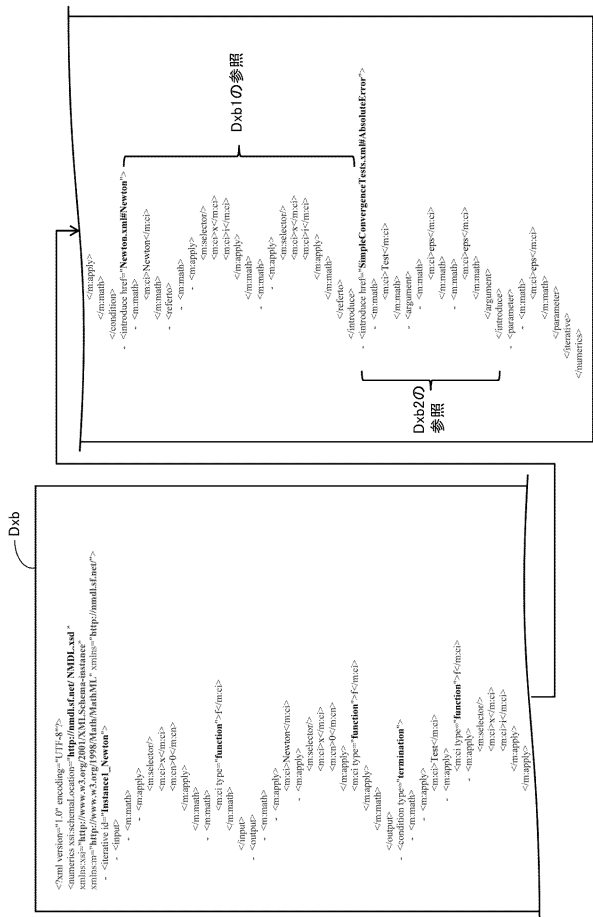


【 図 1 6 】

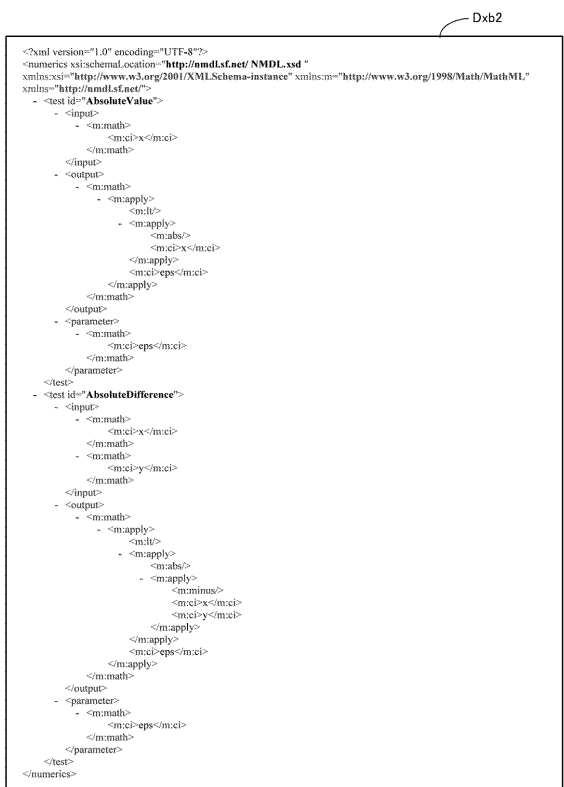




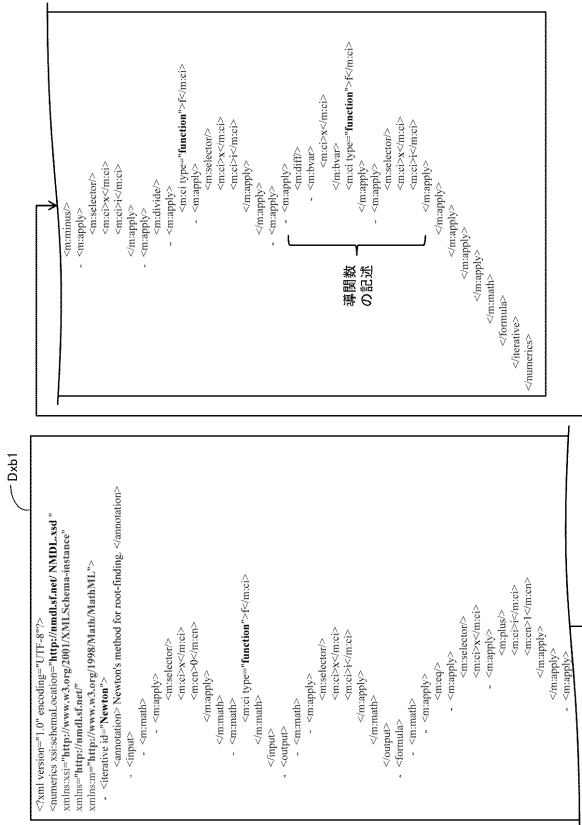
【 図 1 7 】



【 図 1 9 】



【 図 1 8 】



---

フロントページの続き

(72)発明者 天野 晃

滋賀県草津市野路東 1 - 1 - 1 立命館大学 びわこ・くさつキャンパス生命科学部内

Fターム(参考) 5B081 CC32

5B376 BC08 BC32 BC73