計算材料科学の課題への数学・数理科学からのアプローチ:数理ベースプログラミング

檜貝, 信一
アーク・イノベーション

溝口, 佳寛
九州大学マス・フォア・インダストリ研究所

深作, 亮也
九州大学大学院数理学研究院

出版情報：アンサンブル. 23 (4), pp.242-248, 2021-10-31. The Molecular Simulation Society of Japan
バージョン：

[2-4]計算材料科学の課題への数学・数理科学からのアプローチ：数理ベースプログラミング

アーク・イノベーション
檜貝　信一　higai@ark-i.com
九州大学　マス・フォア・インダストリ研究所
溝口　佳寛　ym@imi.kyushu-u.ac.jp
九州大学　大学院数理学研究院
深作　亮也　fukasaku@math.kyushu-u.ac.jp

概要

現在，計算材料科学技術は，日本の材料産業分野の研究開発において，データ科学技術の急速な展開，および，世の中のデジタルトランスフォーメーションの巨大な潮流とも相俟って，実活用化は一段と加速し，既にコモディティー化，デファクトスタンダード化しつつある．しかし，その高精度ゆえの実活用限界の低さ，計算プログラムの膨大化・複雑化，実装技術の古さ，主流である手続き（命令）型プログラミングの限界，レガシーなプログラム言語，他，数多くの重要な課題が顕在化している．我々は，現状からのブレークスルーの可能性を探るべく，計算材料科学への数学・数理科学からの新たなアプローチを試みている．本稿では，手続き型プログラミングに代わる，数学・数理科学の知識と技術を積極的に関数型プログラミングに取り入れた数理ベースプログラミングによるアプローチについて紹介する．

キーワード：
数理ベースプログラミング，計算材料科学，関数型プログラミング，数学，数理科学，算盤，電卓，関数電卓

1. はじめに

日本の材料産業分野の研究開発において，計算材料科学技術の実活用化は，近年のデータ科学技術の急速な展開，そして，世の中のデジタルトランスフォーメーション（DX）という巨大な潮流とも相俟って，一段と加速し，既にコモディティー化，デファクトスタンダード化も進んでいる．しかし，計算材料科学技術については，数多くの重要な課題も顕在化している [1-6]. 計算材料科学の主力である量子化学計算，および第一原理計算は，量子論に基づく，非経験的な，現在求め得る最も高精度な計算だが，その高精度ゆえに，現実的な系を取り扱う際には，膨大な計算コストを要してしまい，実活用での低い限界に直ちに阻まれてしまう．また，計算プログラムも，新たな機能の追加に伴い，膨大化・複雑化の一途にあり, 実装技術も古いまま今日に至っている．さらに，主流である手続き（命令）型プログラミングは，上述の課題の要因ともなっており，その限界に達しつつある．そして，計算材料科学で主に用いられているプログラム言語も，今やレガシーな部類に属するものとなっている．さらに，将来に向けた計算の高速化，大規模化については，主にハードウェアの進歩を頼みとしており，そのやり方も力業的な感が否めない．

我々は，以上の現状からのブレークスルーの可能性を探るべく，計算材料科学への数学・数理科学からの新たなアプローチを試みている [6]. 先に，古典分子動力学計算などを題材として，上述の課題の要因である手続き型プログラミングに着目し，それに代わり，数学・数理科学の知識と技術を積極的に関数型プログラミングに取り入れた数理ベースプログラミング（mathematics-based programming）によるアプローチを実施した．その結果，未だ精査できていない点や，解決すべき課題はあるものの，計算プログラムの簡素化，明瞭化，強固化，エラー・バグの削減など，多くのメリットや可能性が確認できている．本稿では，その数理ベースプログラミングについて，その考え方などを分かりやすく紹介したい．

2. 数理ベースプログラミングとは

プログラムを手続きではなく関数と考えることによって，手続きの持つ性質自体を理解し，複雑な手続きへ誤りなく拡張できる．そして，高効率，かつ，高信頼，すなわち，誤りの少ないプログラムを実現することもできる．関数型プログラム言語とは，そうした立場から考えられたプログラム言語である．近年では，純粋な関数型プログラム言語だけでなく，一般のプログラム言語においても，関数定義や再帰呼出しなどのような関数型プログラム言語の機能が内包されている．本稿では，この考え方を言語に特定せずに関数型プログラミングと呼び，その考え方の大切さについて述べたい．そして，関数型プログラミングを有効活用する上で，数式・数理的思考が非常に重要となる．数式により導かれた再帰方程式，合成関数の性質を記述する公式などを用いることで，プログラムそのものや，プログラミングの過程も洗練される．我々は，そうした数理的な考え方による洗練された関数型プログラミングを，次のように数理ベースプログラミングと呼ぶことにした．

$\boxed{\text{数理ベース = 関数型プログラム + 数式の活用}}$

ここで数理ベースプログラミングの考え方を，「算盤」「電卓」「関数電卓」の発展過程と例えながら考えてみる．「算盤」は計算機の原点であるが，その中には，数学での計算の重要性を強調する多くの基本的な考え方が含まれている．そうした理由もあり，国際数学者会議では，情報科学の数学的側面における優れた貢献を称える賞の名前が，2022 年から「アバカス・メダル」という呼称になった [7]．「アバカス」は「算盤」を含む古くから考えられた計算のための器具であるが，日本や中国の「算盤」は非常に優れたアバカスである．そして，アバカスは数学と計算の関わりの象徴であり，計算のための単純なアイディアによる装置でもあり，更には人々と数学との関わり方に革命さえももたらした．ハードウェアとソフトウェアを駆使し，数学的アイディアを組み合わせることで，世界を変える現代のアバカスに相当するアイディアを創造し，そのアイディアを発展させた研究者に 4 年に 1 度の国際数学者会議でアバカス・メダルが 1 万ユーロの賞金とともに授与される．

プログラムの作成法は様々な文献で紹介されているが，本稿では，「算盤」が「電卓」「関数電卓」と発展したような身近な装置の発展例を使いながら，数学と計算の関わりも含め，数理ベースプログラミングを紹介する．特に，次の 3 つの視点から考察したい．

- **簡単に使える**プログラム
- **分担開発可能な**プログラム
- **数式を活用した**プログラム

「算盤」「電卓」「関数電卓」を，「**簡単に使える**」立場から整理すると，以下のように考えられる．

算盤	方法（珠の動かし方）を覚えれば，様々な演算を計算できる
電卓	数字と演算記号を押せば，簡単に答えを得ることができる
関数電卓	複雑な数式による計算でも，関数名のボタンを押すだけで計算できる

プログラミング作法において，手続き型プログラミングと関数型プログラミングという方法がある．この 2 つを「算盤」と「関数電卓」に例えると，算盤の珠を動かすことが**手続き型**に対応し，関数名のボタンを押すことが**関数型**に対応する．

手続き型	手続きの書き方を覚えれば，様々な計算手続きを実現することができる
関数型	手続きを行う関数の名前がわかればその関数を呼出して，計算できる

次に，「**分担開発可能な**」立場から整理すると，次のようになる．

電卓	行いたい計算に対して，その都度，ボタンを押さねばならない
計算機	決まった順番に押すボタンの列をまとめて，ひとつのボタンにできる

関数型プログラミングにおいて，関数として手続きを作成することが，関数電卓においてひとつのボタンで計算できることに対応する．関数電卓において，複数のボタンを順番に押すことが，次のように関数の計算結果を関数の引数とすることに対応する．

$$b = f(a); \qquad c = g(b)$$
$$\Rightarrow h(x) \coloneqq g\big(f(x)\big); \;\; c = h(a) \qquad (1)$$

複数のボタン f, g を順番に押すことをひとつのボタン h で実現すれば，さらに複雑な計算をひとつの操作で実現できる．そのときに大切なことは，hの作成者は，f, g のボタンを押すだけで，その中の計算の実現方法については感知しなくて良いということである．そうした開発こそが**分担開発可能**である．

一方，$g(f(x))$ を計算する新しい関数 $k(x)$を得られ，数式 $g(f(x)) = k(x)$ を数学的に証明できたとすると，関数 $h(x)$ の計算を f, g なしで直接 $h(x) \coloneqq k(x)$ から計算できるようになる．そうした数学的背景を持ったプログラミングが，**数式を活用した**プログラミングである．

上述の内容は，次のようにまとめることができる．

- **算盤**
  - 複雑な加減算演算も，珠を移動しながら決められた手続きで計算できる
  - 演算に対応する個々の手続きや個々の手順は，人が覚えておく必要がある
  - 手順が長くなれば，珠の移動で人為的ミスが起きやすい
- **関数電卓**
  - 機能ごとにボタンひとつになっている関数電卓を使うと，人為的ミスが減る
  - それでも，機能ごとのボタンの種類が増えると，ボタンと機能が覚えられなくなる
  - 複数機能をまとめたひとつの機能を作り，ひとつのボタンにして，ボタンを減らす
  - それでも，どんどんボタンが増える
- **数理ベースプログラミング**
  - あるとき，手順が異なっても，結果が同じボタン操作（数式）の存在に気が付く
  - 実は，そのボタンの機能が同じであることが，数学の等式から証明できる
  - 最初から数学の等式を活用して，短いボタン数での計算の実現を行うと良い
  - そのための基礎となるプログラムの考え方が，計算に名前をつけて関数にする関数型プログラミングである

3. 数理ベースプログラミング事例紹介

本節では，数理ベースプログラミングの事例として，物体の運動シミュレーションプログラム作成において解析力学の考え方を導入したアプローチを紹介する．特に，数式の活用としての解析力学，数式処理の活用について述べる．

古典力学においては，定められた座標系における物体の運動を，加速度と力の関係式によるニュートンの運動方程式と呼ばれる微分方程式で記述する．その計算手続きは，座標系の決定，運動方程式の構成，微分方程式の求解アルゴリズムの実現により完成される．すなわち，**数式**（微分方程式）の求解アルゴリズムを実現するプログラム作成が重要であることがわかる．

様々な運動の記述が，座標系に依存して（最初の座標系を変えると）運動方程式や微分方程式の形が変わる．そのため，求解アルゴリズムの実現プログラムを再構成することが必要となり，課題となる．

さらには，その再構成は，座標系の変更まで遡るので，過去のプログラムの遺産を引き継ぐことができない．もちろん，過去のプログラムの作成者は，プログラム作成方法は記憶しているので，時間をかければ作成することができるが，その手順は，算盤の珠を丁寧に繰り返して動かす動作と同様，実現可能であるが，長く大変な作業となり，その過程での人為的ミスが起きやすい．

一方，物理学において，**解析力学**の考え方は，座標系に依存しない一般的な物理法則の記述を与えてくれる．解析力学においては，運動エネルギーとポテンシャルエネルギーの差を表すラグランジュ関数を用い，運動方程式はラグランジュ関数の作用汎関数に対する最小作用の原理から導かれる．ここで注意したい点は，座標系によらず一般的な運動方程式の導出手順があることである．運動エネルギーT，ポテンシャルエネルギーVに対して，ラグランジアンLは，$L = T - V$で定められる．そして，その系の運動はラグランジュ方程式

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\frac{\partial L}{\partial x'} = 0 \qquad (2)$$

で決まる．

本稿では，簡単のために放物運動と振子運動の 2 つの運動方程式を例に紹介する．

<u>**放物運動：**</u>

質点の質量をm，高さをx，重力加速度gとするときの運動を考える．その運動エネルギーは$T = \frac{1}{2}m(x')^2$，ポテンシャルエネルギーは $V = mgx$，ラグランジアン（$L = T - V$）に対するラグランジュ方程式は，$-mx'' - mg = 0$となり，これはニュートンの運動方程式$x'' = -g$に等しい．

<u>**振子運動：**</u>

質点の質量m，重力加速度g，弦の長さℓ，弦の鉛直方向とのなす角をxとするときの運動を考える． その運動エネルギーは $T = \frac{1}{2}m(\ell x')^2$，ポテンシャルエネルギーは $V = -mg\ell\cos x$，ラグランジアン（$L = T - V$）に対するラグランジュ方程式は$mg\ell\sin x + mg\ell^2 x'' = 0$となり，これはニュートンの運動方程式$x'' = -\frac{g}{\ell}\sin x$に等しい．

放物運動，振子運動の 2 つのシミュレーションプログラムを考えるとき，異なる座標系で表現された異なる条件の運動であり，その運動方程式は全く異なる．一方で，ラグランジアンを用いた解析力学による求解を考えるとき，どちらもラグランジュ関数の求解アルゴリズムの実現と捉えることができる．**関数型プログラミング**において，関数の引数は数を表す変数だけでなく，関数を引数とすることができる．すなわち，ラグランジュ関数を引数とする求解アルゴリズムを作成しておけば，その引数として，

$$L = \frac{1}{2}m(x')^2 - mgx \qquad (3)$$

とすると放物運動のシミュレーションの計算となり，

$$L = \frac{1}{2}m(\ell x')^2 + mg\ell\cos x \qquad (4)$$

とすると振子運動のシミュレーション計算となる．

ここで注意すべき点は，ラグランジアンは複雑な高階微分方程式となるため，その一般的な求解アルゴリズムの実現は容易ではない．一方，近年の**数式処理**システムの発展により，数式レベルでの式の展開や簡約化，パターン分類による高速処理アルゴリズムなども考案されている．

次に，**二重振子運動**の例を紹介する．長さℓの棒の先に，質量mの質点がある振子を図 1 のように 2 つ接続する．重りを横に動かし，放すと往復運動を始める．この運動を，微分方程式を解くことでシミュレーションする．ここで固定端を原点とし，鉛直方向からの振子の偏角をθ_1, θ_2とする．運動エネルギーT，位置エネルギーV，ラグランジアンLは，

$$T = \frac{1}{2}m\ell^2\left(2(\theta_1')^2 + (\theta_2')^2 + 2\theta_1'\theta_2'\cos(\theta_1 - \theta_2)\right)$$

$$V = -mg\ell(2\cos\theta_1 + \cos\theta_2)$$

$$L = T - V \qquad (5)$$

となる．そして，次のラグランジュ方程式を解けば，2 つの振子の角度 θ_1, θ_2 の時間変化が得られる．

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \theta_1'}\right) - \frac{\partial L}{\partial \theta_1} = 0, \qquad \frac{d}{dt}\left(\frac{\partial L}{\partial \theta_2'}\right) - \frac{\partial L}{\partial \theta_2} = 0 \qquad (6)$$

このシミュレーションは，数式処理機能を持つ関数型言語 Mathematica では，以下のように十行程度のプログラムで実装できる [8].

```
Module[{s10,s20,v10,v20,g,l,L,sol},
  s10=(4 Pi)/3;s20=Pi/2;v10=0;v20=0;g=9.8;l=10;
  L=1/2*l*l*(2(s1'[t])^2+(s2'[t])^2+2(s1'[t])(s2'[t])*Co
s[s1[t]-s2[t]])+g*l*(2*Cos[s1[t]]+Cos[s2[t]]);
  sol=NDSolve[{
       D[D[L,s1'[t]],t]-D[L,s1[t]]==0,
       D[D[L,s2'[t]],t]-D[L,s2[t]]==0,
       s1[0]==s10,s1'[0]==v10,s2[0]==s20,s2'[0]==v20},
       {s1,s2},{t,0,100},MaxSteps->Infinity];
  ParametricPlot[
    Evaluate[{
      Sin[s1[t]]+Sin[s2[t]],
      -Cos[s1[t]]-Cos[s2[t]]}/.sol],
      {t,0,100},PlotStyle->Blue]]
```

変数 L にラグランジアンを与え，`NDSolve` 関数でラグランジュ方程式を解き，`ParametricPlot` 関数で質点の軌道を描画しており，上のプログラムは非常にシンプルである．図 2 は，この二重振子の先端の青色の質点の軌跡を図示したものであり，この二重振子の先端は，カオスと呼ばれる複雑な非周期的な運動をすることが知られている．

質点の数（微分方程式中の変数の数）が多くなるほど，求解アルゴリズムの計算時間は長くなるが，本稿で述べたいことは，関数型プログラミングにより，役割を分離し，それぞれの関数の中で最適アルゴリズムの実現を考えるアプローチの紹介である．数の計算のための関数だけでなく，数式処理のためのアルゴリズムを実現した関数たちも多々開発されている．さらには，単純な数式変換ではない，新たな等式や式変形の理論が，別の視点から，数学者らにより研究され証明されている．これらを活用した数だけの計算だけではなく，数式の計算も意識したプログラミング作法が，**数理ベースプログラミング**である．

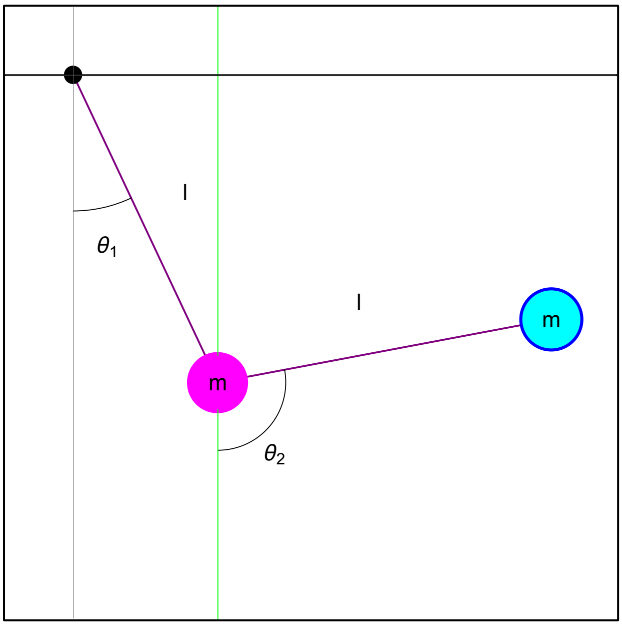

図 1. 二重振子の模式図.

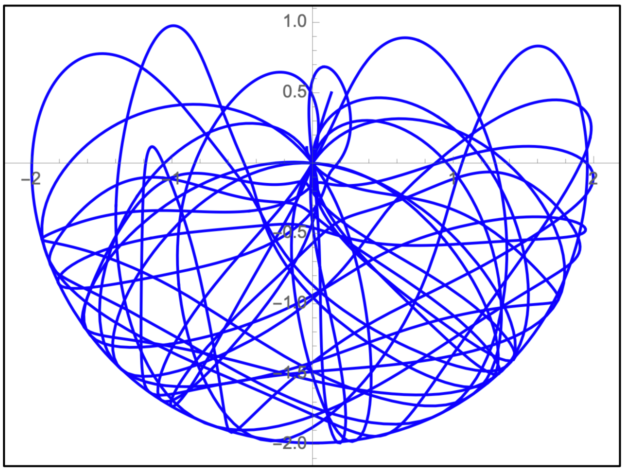

図 2. 図 1 の二重振子運動のシミュレーションにおける，先端の青質点の軌跡を示したグラフ.

4. おわりに

本稿では，関数型プログラミングへ数学・数理科学の知識と技術の蓄積である数式を活用するアプローチを提言し，具体例を用いて，その考え方を紹介した．特に，解析力学と数式処理を活用した物理シミュレーションの実現方法について述べた．本稿で

は紙面の都合上述べていないが，解析力学の活用は，解析空間をユークリッド空間のみでなく，周期的な空間，トーラスにおいても考察が可能である．周期的な空間での解は，実空間の周期解に対応する．このアプローチについては，文献 [6] を参照して欲しい．また，我々は，プログラム言語 Haskell [9]を用いた純関数型プログラミング，汎用言語を用いた関数型プログラミング，そして，数学・数式を活用した数理ベースプログラミングの教材開発に着手している（図 3 参照）．

図 3. 「数理ベースプログラミングを学ぼう！」

今後の課題として，具体的な計算科学のライブラリを関数型プログラミングによるライブラリとして再構築し，数理ベースのプログラミングの事例を拡張することを検討している．

最近，国内のあらゆる産業分野において，急に，「ソフトウェアファースト」が唱えられるようになってきた．ハードウェアの進歩の速度が大幅に鈍り，ソフトウェアの進歩の速度がこれをはるかに上回ってきたこと，そして，自動車の自動運転をはじめとする人工知能技術の実用化，さらに DX と，これらが目前の課題となったことが，その背景である．これまでの日本は，「ハードウェア至上主義」にあり，ソフトウェアを「タダ」，「オマケ」程度のものと軽視する傾向にあった．そのことが反映されてか，計算材料科学分野でも，主な計算プログラムは，ほとんどが外国製である．ソフトウェアの開発力の重要性がさらに高まる今後に向け，我々が大きな可能性を有すると期待している数理ベースプログラミングに，多くの方が興味・関心を持たれることを切に願っている．

謝辞

本稿で紹介した我々の取組について，その方向性を考案する上での重要なポイントをご教示頂いた神坂英幸博士に，深く感謝致します．本取組について，数多くの有意義な議論をして頂いたコンピュータによる材料開発・物質設計を考える会（CAMM フォーラム）[10] の皆様に，心より感謝致します．数理ベースプログラミングを，大変親しみやすいグラレコで表現して下さった TSUBU 氏と藤田ハルノ氏に，厚くお礼を申し上げます．本取組は，AIMaP 事業 [11] の一環によりコーディネートされた共同研究のものである．

参考文献

[1] 檜貝信一，化学と工業，特集「マテリアルズ・インフォマティクス」，**71**, 659 (2018).

[2] 檜貝信一，神坂英幸，CREST・さきがけ・AIMaP 合同シンポジウム「数学パワーが世界を変える 2019」，東京，3/10, 11, 2019.

[3] 檜貝信一，応用数理，**29**, 170 (2019).

[4] 檜貝信一，23 の先端事例がつなぐ計算科学のフロンティア 計算で物事を理解する予測する，理化学研究所 中村特別研究室 編，近代科学社，p.193 (2019).

[5] 檜貝信一，マテリアリズ インフォマティクス総合ウェビナー「Whole view of MI 日本における MI の現状, これから, そして MI を超えて」, アーク・イノベーション主催，東京，5/27, 28, 2021.

[6] R. Fukasaku, Y. Mizoguchi and S. Higai, “Approaches to molecular dynamics calculations from mathematics / mathematical sciences”, J. Comput. Chem. Jpn., to be published (2021).

[7] International Mathematical Union (IMU) Abacus Medal, 公式 web サイト：https://scilogs.spektrum.de/hlf/imu-abacus-medal/.

[8] 溝口佳寛,「数学処理ソフト Mathematica で数学の問題を解く」，九州数学教育会 算数・数学教育研修会，福岡，6/12, 2016, https://www.slideshare.net/yoshihiromizoguchi/mathematica-62981039.

[9] Haskell, 公式 web サイト：https://www.haskell.org/.

[10] コンピュータによる材料開発・物質設計を考える会（CAMM フォーラム），公式 web サイト：https://www.bri.or.jp/camm/.

[11] 文部科学省科学技術試験研究委託事業 数学アドバンストイノベーションプラットフォーム（Advanced Innovation powered by Mathematics Platform: AIMaP），公式 web サイト：https://aimap.imi.kyushu-u.ac.jp/wp/.

著者紹介

檜貝信一（博士（工学））：〔経歴〕1997 年筑波大学大学院工学研究科物質工学専攻博士課程修了, 同年金属材料技術研究所（現 物質・材料研究機構）入所, 2005 年村田製作所入社. 2019 年より現所属. 〔専門〕計算材料科学，コンサルティング. 〔趣味〕サイクルスポーツ，モーターサイクル，ドラムス.

溝口佳寛（博士（理学））：〔経歴〕1988 年九州大学大学院理学研究科博士後期課程数学専攻単位取得退学，同年九州工業大学情報工学部講師，1992 年博士（理学）九州大学，九州大学大学院数理学研究院助教授を経て, 2016 年より現所属. 〔専門〕理論計算機科学. 〔趣味〕音楽鑑賞，電子工作.

深作亮也（博士（理学））：〔経歴〕2016 年東京理科大学大学院理学研究科数理情報科学専攻博士後期課程修了，同年同理学部第一部助教，2019 年から現所属. 〔専門〕計算機代数. 〔趣味〕読書.