

## 等式の両辺の単位が等しいかチェックするプログラムの実装\*

奥原 智史<sup>†</sup>岩見 宗弘<sup>‡</sup>

## 1 はじめに

物理学の計算をする際、立式は正しくても、途中の計算を間違え、 $F = v$ のような明らかに間違った等式を得ることがある。等式  $F = v$  において、 $F$  は力、 $v$  は速度を意図している。力は速度にはなりえないので、この等式は明らかに間違っている。このとき、どこで間違えたのか調べるため、最初の等式から1行ずつ改めて式変形を確認する必要がある。ここで、間違いを発見するために、等式の単位を用いる方法がある。両辺の単位が異なっていれば明らかにその等式は誤りであり、同じであれば誤りはないかもしれないと推測できる。実際、著者が高校時代に物理学を勉強していたとき、式変形は手計算で行うしかなく、非常に面倒だった。また、計算ミスも発生することがあった。計算ミスが発生した際、どこで間違えたか調べるために式変形前後で単位の計算をしていた。このため、単位の計算を自動で行うソフトウェアがあればよいと考え、この研究を行うことにした。本稿では、LaTeX で書かれた等式とその中に含まれる変数の単位から、その等式の両辺の単位が等しいか調べるプログラムをJavaを用いて実装する。このプログラムでは、四則演算および平方根と整数乗に対応することができる。なお、本研究のプログラムは三角関数とベクトルを含む等式は対象としていない。

本稿は文献 [3] の改良版である。

## 2 プログラムの実装

以下では、本稿で実装したプログラムをTCP(TeX Certification Program)とよぶ。TCPは、LaTeXで

書かれた等式と、それに含まれる変数の単位を入力とし、両辺の単位およびそれらが等しいかどうかを出力とする。まず、TCPで単位の計算をするためには、単位を扱うためのデータ構造が必要である。それが本研究のUnitクラスに相当し、単位のデータ構造や単位同士の計算をUnitクラス内のフィールドおよびメソッドとして定義している。Unitクラス内で単位は文字列と整数からなるハッシュマップとして定義されている。キーとなる文字列は基本単位であり、値となる整数はそれぞれ基本単位の指数である。例えば、 $m/s$  という単位は  $\{m:1, s:-1\}$  と表される。TCPは、LaTeXの等式を解析し、その単位を計算する。次に、その過程を6ステップに分割し、それぞれのステップに対応するメソッドをJavaで実装した。また、実装を容易にするためのメソッドも別に用意した。(1) 第1ステップは、LaTeXで書かれた等式を表す文字列をトークン(意味のある最小単位)ごとに区切る。(2) `\frac` のような多引数のコマンドは、どこまでが引数であるかを先に識別しておくとその後のプログラムでの処理が簡単になる。このため、第2ステップは`\frac{a}{b}` のような多変数のコマンドを`\frac{a ' b}` のように変形する。(3) 後の第5ステップでは、ユーザに各変数の単位を入力させる。そのために、トークンのうちどれが変数であるか識別する必要がある。すなわち、第3ステップでは等式で使われている変数を取り出す。(4) LaTeXの数式には、 $a+b$  のように中置記法により表現される演算と`\sqrt{a}` のように前置記法により表現される演算がある。第4ステップは、これらの演算をコンピュータで扱いやすくするために後置記法に置き換える。このステップのアルゴリズムは文献[2]の操作場アルゴリズムを参考にして実装した。(5) 第5ステップは、ユーザに第3ステップで取り出した各変数に対する単位を入力するように求める。(6) 第6ステップは等式の両辺の単位を実際に計算し、それらが等しいか調べる。また、第4ステップの実装を容易にするため、省略

\*Implementation of program to check if the units on both sides of an equation are equal

<sup>†</sup>島根情報処理センター, Satoshi Okuhara, Shimane Information Processing Center

<sup>‡</sup>島根大学総合理工学部知能情報デザイン学科, Munehiro Iwami, Department of Information Systems Design and Data Science, Shimane University

された \* と単項の - に対する処理を行うためのメソッドを別に用意した。さらに、入出力のために独自の GUI を用意した。これは GUI を作成するためのパッケージである swing を用いて実装した。

### 3 検証とその結果

高校物理の公式集 [5] に掲載されている公式とその公式で使われている変数の単位を、第 2 節において作成したプログラムに入力し、どれだけの割合の公式に対してプログラムが正しく動作するか調べた。なお、他の公式集ではなく文献 [5] を使用したのは、我々が文献 [5] よりまとまった高校物理の公式集を発見できなかったからである。また、本研究を大学の物理学に対応させようとする、線形代数と微積分のライブラリを作成する必要がある。これらが、1 年間の卒業研究では不可能であると判断し、高校物理の範囲のみを扱うことにした。その結果、公式集中の 55.1% を占める本プログラムで対象としているすべての公式で正しく動作することが確認された。また、残りの 44.9% の公式はもともとプログラムを作成する際に対象としないとした公式なので、このプログラムからバグが検出されることはなかった。以下に検証結果の一部を示す。これらは TCP が正しく動作した公式である：1.  $y = \frac{1}{2}gt^2$ , 2.  $f = \frac{1}{2\pi\sqrt{LC}}$ , 3.  $\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$ . ここで、 $y = \frac{1}{2}gt^2$  は四則演算と整数乗からなる基本的な公式である。また、 $f = \frac{1}{2\pi\sqrt{LC}}$  は分数と平方根の両方を用いた公式である。さらに、 $\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$  は等式の両辺がともに変数の形ではない公式である。一方、以下は TCP が正しく動作しなかった公式である：4.  $m\vec{a} = \vec{F}$ , 5.  $f = qvB \sin \theta$ . ここで、公式  $m\vec{a} = \vec{F}$  はベクトルを使用しており、 $f = qvB \sin \theta$  は三角関数を使用しているため、TCP の対象外の公式である。

### 4 関連研究との比較

関連研究として、Python で物理量の計算を行うためのパッケージである physipy が存在する [4]。しかしながら、physipy は LaTeX の等式を構文解析できず、また GUI も存在しない。そのため、ユーザは Python を用いてプログラミングを行う必要がある。一方、本研究ではユーザは GUI のウィンドウから等式を入力すればよいので、プログラミング

を行う必要はない。また、物理量の次元を計算したり、次元をもとに式や現象を考察することを次元解析という [1, 6]。本研究では計算に単位を用いており、次元は用いていない。

### 5 おわりに

本稿では、LaTeX で書かれた等式とその中に含まれる変数の単位から、その等式の両辺の単位が等しいか調べるプログラムを Java を用いて実装した。このプログラムでは、四則演算および平方根と整数乗に対応することができた。また、高校物理の公式集を用いた検証を行うことによって、プログラムが正しく動作することを確かめることができた。

本稿では、三角関数を含む等式を対象範囲から除外した。しかしながら、対象から除外した主な理由は時間が足りなかったためであり、今後の課題は三角関数を扱えるようにすることである。また、ベクトルを含む等式への対応も今後の課題である。また、等式を LaTeX で入力するのではなく、関数電卓のように GUI 上のボタンをクリックして入力できるように改良することも今後の課題である。これらの改良により、LaTeX の基礎的事項 (`\frac`, `\sqrt`) を知らなくてもプログラムを使用することができるようになるため、このプログラムのユーザが増えると考えられる。

### 参考文献

- [1] 宮本 雲平, 微分方程式と次元解析をテーマとした物理の高大接続授業 II, 秋田県立大学ウェブジャーナル A, Vol. 9, pp.139–147, 2021.
- [2] T. Norvell, Parsing Expressions by Recursive Descent, [https://www.engr.mun.ca/theo/Misc/exp\\_parsing.htm](https://www.engr.mun.ca/theo/Misc/exp_parsing.htm), 1999.
- [3] 奥原 智史, 卒業論文, 島根大学, 総合理工学部, 数理・情報システム学科, 2021.
- [4] physipy, Manipulate physical quantities in Python, <https://pypi.org/project/physipy/>
- [5] Rotton, 高校物理公式一覧, <https://wakariyasui.sakura.ne.jp/a/kousiki.html>, 2019.
- [6] 矢崎 成俊, 次元解析入門, 共立出版, 2022.