

# 常微分方程式の解をコンピュータで検算する

大信田 丈志

2022年10月6日

この文書は、書籍『常微分方程式入門：物理を使うすべての人へ』（日本評論社）について <https://www.nippy.co.jp/shop/book/8914.html> で提供する補足資料の一部です。式番号は上記の書籍の数式を参照しています。

## 目次

1	Mathematica を用いる場合	2
2	Python を用いる場合	4

## 注意事項および謝辞

- この文書は、なるべく有用であるように努めましたが、内容の完全な正確さは保証できません。特に、ソフトウェアのバージョンアップなどに伴い、記載内容が実態に合わなくなる可能性があります。そのほか意図しない間違いが含まれる可能性もあります。
- この文書は、対応するソフトウェアの種類を増やすなどの理由で、予告なく更新する場合があります。最新版を日本評論社のウェブサイトでご確認ください。
- 説明のなかで用いている会社名・製品名などは、各社の登録商標または商標です。
- Python による検算の基本的なアイデアは、鳥取大学工学部の星健夫先生にご教示いただきました。ここに謝意を表します。

## 1 Mathematica を用いる場合

### 1.1 準備

Mathematica がインストールされているパソコンで、セッションを新規に起動します。

既に起動されているセッションを利用する場合には、使う予定の変数に何かが入っているとエラーの原因になるので、必要に応じて変数をクリアしておきます。以下の例では

```
Clear[t, u, y]
```

とすればいいでしょう。

### 1.2 検算の例：方程式を定義する方法

方程式をまず定義し、次に未知関数に検算対象を代入する方法で検算を行う例を示します。

たとえば、常微分方程式

$$\frac{du}{dt} = \frac{2u}{1+t} \quad (1.1.28)$$

に対し、 $u = A(1+t)^2$  が解となることを確かめましょう（ここで  $A$  は任意定数）。まずは方程式 (1.1.28) を、Mathematica の定義の形で

```
ode[u_] := (D[u,t] == 2u/(1+t))
```

のように入力します。次に、

```
ode[A (1+t)^2] // Simplify
```

と入力すると、方程式の  $u$  に  $A(1+t)^2$  を代入した式の成否が分かります。結果が `True` なら検算成功で、 $t$  の条件式が返ってきたり `False` と表示されたりした場合は検算失敗です。

上記の例で `ode` となっている箇所は、式番号など、区別のための文字を追加し

```
ode28[u_] := (D[u,t] == 2u/(1+t))
ode28[A (1+t)^2] // Simplify
```

とすることもできます。

同じ方法で初期値問題の検算も行ってみましょう。たとえば  $y = 2 - e^{-4t}$  が

$$\frac{dy}{dt} + 4y = 8 \quad (1.1.16)$$

$$y|_{t=0} = 1 \quad (2.3.9)$$

という初期値問題の解になっていることを確かめるには、まず、この初期値問題を

```
ode[y_] := (D[y,t] + 4 y == 8)
problem[y_] := {ode[y], (y /.t->0)==1}
```

のように入力し、次に

```
problem[2 - Exp[-4t]] // Simplify
```

として  $y$  に  $2 - e^{-4t}$  を代入します。これで結果が `{True,True}` になれば検算成功です。

### 1.3 検算の例：関数を定義する方法

上記の方法とは逆に、関数を Mathematica での定義の形で入力してから、そのあとで方程式の等号の成否を調べる方法もあります。

なお、あらかじめ（“準備”の項目に書いたように）必要な変数をクリアしておきましょう。

たとえば  $u = A(1+t)^2$  が方程式 (1.1.28) を満たすかどうかを調べるには、関数  $u(t)$  を

```
u[t_] := A (1+t)^2
```

のように入力して定義し、そのあと

```
u'[t] == 2u[t]/(1+t) // Simplify
```

として方程式 (1.1.28) の等号の成否を検証します。これにより `True` が表示されれば、等号が成立しているということで検算は成功です。

初期値問題 (1.1.16)(2.3.9) の場合は、検算し

たい関数を

```
y[t_] := 2 - Exp[-4 t]
```

のようにして定義し、そのあと

```
{y'[t] + 4 y[t] == 8,
 y[0] == 1} // Simplify
```

とします。結果が {True, True} なら検算成功ということになります。

## 1.4 数式の入力に関する補足

コンピュータにとって人間の言葉はまだまだ難しいので、人間の意思をコンピュータに正しく伝えるには工夫が必要です。たとえば、上記の例をよく見ると、人間の世界では単に“=”で済むような等号も、コンピュータの都合に合わせて区別して書いていることが分かります。

- 方程式や命題の等号は ==
- 関数などを定義するときの等号は :=

そのほか `z = 3t` のような単純な代入を示す等号もあります。こういった様々な“等号”の正確な意味を知りたい場合は、Mathematica の解説書を見てください。

微分演算子は D で入力します。たとえば

```
D[t^3, t]
```

と入力すれば

$$\frac{d(t^3)}{dt} = 3t^2$$

により  $3t^2$  が出力され、

```
D[t^3, {t, 2}]
```

と入力すれば  $(d/dt)^2(t^3)$  の計算結果である  $6t$  が出力されます。

他方、プライムを用いた導関数の記法も利用可能です。関数  $f(t)$  を

```
f[t_] := t^3
```

のように定義しておけば、 $f''[t]$  と入力すると  $6t$  が出力されます。他方、

```
Clear[f]
D[f[t], {t, 2}]
```

とすると  $f''[t]$  つまり  $f''(t)$  が出力されます。

さらに、Mathematica には、Sin[], Cos[], Exp[], Sqrt[] など、様々な関数があらかじめ組み込まれています。ただし、これらの関数を Mathematica に入力する際には、頭文字が大文字になっていることに注意しましょう。

## 2 Python を用いる場合

### 2.1 準備

何らかの方法で Python の実行環境を準備します。現時点で最も手軽な方法としては、適当なブラウザで Google アカウントにログインし、Google Colaboratory にアクセスするのがいいようです。

Python を入力する準備ができれば、まずは

```
from sympy import *
```

と入力します。これで、Python で文字式を扱うための SymPy というライブラリが使えるようになります。

### 2.2 検算の例

常微分方程式

$$\frac{du}{dt} = \frac{2u}{1+t} \quad (1.1.28)$$

に対し、 $u = A(1+t)^2$  が解となることを確かめましょう ( $A$  は任意定数)。それには、まず検算すべき関数  $u$  を以下のように入力します：

```
t = Symbol('t')
A = Symbol('A')
u = A * (1+t)**2
```

続いて、方程式 (1.1.28) の左辺と右辺を別々に計算します。左辺 (LHS) は  $du/dt$  ですから

```
LHS = diff(u,t); print(LHS)
```

とすればよく、その結果  $A*(2*t + 2)$  が出力されます。右辺 (RHS) は

```
RHS = 2*u/(1+t); print(RHS)
```

により  $2*A*(t + 1)$  が出力されます。

最後に、左辺と右辺が等しいかどうかを

```
print(simplify(LHS-RHS)==0)
```

として検証します。結果が True になれば検算をパスしたことになります。

続いて、初期値問題の例を見てみましょう。たとえば  $y = 2 - e^{-4t}$  が

$$\frac{dy}{dt} + 4y = 8 \quad (1.1.16)$$

$$y|_{t=0} = 1 \quad (2.3.9)$$

という初期値問題の解になっていることを確かめるには、まず、検算すべき関数  $y = y(t)$  を

```
y = 2 - exp(-4*t)
```

のように入力します (既に  $t$  は設定済みだとします)。論点は、この関数が、式 (1.1.16) の等号と式 (2.3.9) の等号を両方とも成立させるものになっているか否かです。

方程式 (1.1.16) の等号成立を検証するには

```
LHS = diff(y,t) + 4*y; print(LHS)
RHS = 8; print(RHS)
```

としたあと

```
print(simplify(LHS-RHS)==0)
```

とすればいいでしょう。他方、式 (2.3.9) は

```
print(y.subs(t,0)==1)
```

とすれば検証できます。上記の両方の式について True が返ってくれば成功です。

## 2.3 Python での数式の入力に関する補足

とりあえずは以下の表を見てください。

Python での入力	数式	備考
<code>p = q</code>	$p = q$	$p$ に $q$ を代入する
<code>p == q</code>	$p = q$	$p$ と $q$ は等しいか (方程式)
<code>y = sqrt(1+t**2)</code>	$y = \sqrt{1 + t^2}$	
<code>diff(x,t)**2</code>	$\left(\frac{dx}{dt}\right)^2$	
<code>diff(x,t,2)</code>	$\frac{d^2x}{dt^2}$	
<code>1/3</code>	0.333333333333	小数に変換され近似値となる
<code>Rational(1,3)</code>	$\frac{1}{3}$	

ここに載っていないものについては、“SymPy 使い方”などのキーワードでウェブ検索してみれば、さまざまな入力例や解説などが見つかると思います。