

数学を楽しく学ぶコツは「イメージ」をつかむこと

葛谷 直規

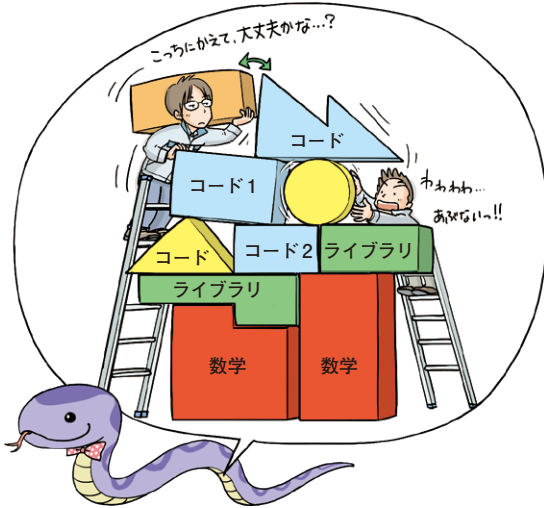


図1 ソフトウェアの根幹は数学

読者の方は数学にどのようなイメージをお持ちでしょうか。筆者は、長年数学に苦手意識を持っていました。高校までの数学は、とにかく問題の解き方だけを覚える、それが大学に入ると急に複雑な定義が出てきて、ちんぷんかんぷんになったことを覚えています。

本稿ではエンジニアなりの数学との向き合い方をお話しします。

エンジニアが数学を学ぶ意義

なぜ数学を勉強する必要があるのでしょうか。基本的に理系の業務というのは背後に数学があることがほとんどです(図1)。特に、近年のAIの発展により、これまで数学にあまり触れる必要のなかった方も、その裏にある数学に触れる必要性も増えてきているのではないのでしょうか。

● ライブラリの中身の数学を知って危険回避

中身の数学を覆い隠してライブラリとして提供しているツールも多くあります。例えば、線形代数だとPython言語のNumPyやSciPy、ソフトウェア開発環境のMATLABや、オープンソースのOctaveやScilab

リスト1 solve() 関数を使って方程式を解く

```
>>> A = np.array([[1.0, 2.0], [-2.0, -4.0]])
>>> b = np.array([1.0], [-2.0])
>>> x = np.linalg.solve(A, b)
Traceback (most recent call last):
...
numpy.linalg.LinAlgError: Singular matrix
```

エラーが発生して解けない

リスト2 pinv() 関数を使って方程式を解く

```
>>> x = np.linalg.pinv(A).dot(b)
>>> print(x)
array([[0.2], [0.4]])
```

解けたように見えるが...

など、多くのツールが用意されています。では、それらのツールがあれば中身の数学を知らなくても済むのでしょうか。

済む場合もあるでしょうが、使いこなそうとするほど済まない場合が増えてきます。また、中身を知らないでツールを使っているとかわぬやけどをこうむることもありえます。

例えば行列による連立1次方程式(つまり行列とベクトルの積による方程式)をPythonで解く場合、NumPyのlinalg.solve()という関数を使うことができます。次の方程式があるとします。

$$Ax = b \dots\dots\dots(1)$$

ただし、

$$A = \begin{bmatrix} 1 & 2 \\ -2 & -4 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \dots\dots\dots(2)$$

で、 x は未知の2次元ベクトルです。solve()関数で解こうとすると、リスト1のエラーが発生してしまいます。

そこで、調べてみると、pinv()関数を使えば、解を求めることができる、という情報を得たとします。そこでリスト2のように実行します。これで問題は解決でしょうか。解決できている場合もあれば、できていない場合もあるでしょう(もともと解きかかった問題次第)。

この行列は本来2次元を表現できるだけの情報を持っていないため、この方程式の解は無限に存在しま