

作りながら学ぶ… シミュレーションとソルバ

新井 正敏

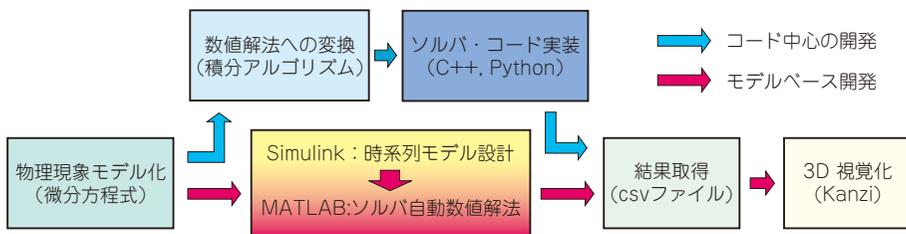


図1 コード中心開発とモデルベース開発

● シミュレーション計算はSimulinkで実行しているわけではない

Simulinkで構築したモデルは、第1章で示したように時間関数としての構造がブロック線図の中に埋め込まれるので、人間が直観的に扱いやすいというメリットがあります。

しかし、Simulinkは数値解法そのものを内部に持つわけではないので、Simulink単独でシミュレーション計算を実行することはできません。シミュレーションを行うためには、MATLABに用意されたソルバを用いて、時間積分に基づく数値計算を行う必要があります。

ソルバとは、微分方程式、代数方程式、および制約条件などから構成される数理モデルを、時間軸に沿って数値的に解くためのアルゴリズムです。このようなソルバは、MATLABに限らず、C++やPythonといったプログラミング言語においても、必要に応じて自ら実装することができます。

● 本章でやること…ソルバの仕組みを解き明かす

本章では、コンピュータ・シミュレーションの裏側で稼働しているソルバの仕組みについて、その考え方を1つずつ解き明かしていきます。ルンゲ=クッタ法は代表的なソルバの1つですが、そのアルゴリズムは比較的複雑であり、直観的に理解できる教材は多くありません。MATLABには複数のソルバが用意されていますが、初学者の多くは、それぞれの特性を十分に理解しないまま、経験的あるいは慣習的にソルバを選

択して使用しているのが現状だと思います。

そこで本章では、数値計算に共通する基本的な考え方を理解するための導入として、まず数値積分の考え方を取り上げます。その後、時間積分を行う各種ソルバの特性について整理していきます。

本章では微分積分を多用し、計算もやや複雑になりますので、他の章を一通り理解した後に、振り返りとして読むことも可能です。ただし、モデルベース開発を行う際には、ソルバが計算エンジンとして裏方で稼働していることを、常に意識しておいてください。

開発の流れとソルバの必要性

ここでは、シミュレーション計算を行う上で、なぜソルバが必要となるのかについて、コード中心の開発とモデルベース開発の流れを対比しながら示します。また、本章で紹介するソルバの順序についても示し、次節以降では、MATLABのコードによる実装とSILSによる検証を行っていきます。

● コード中心の開発とモデルベース開発

図1に、微分方程式で表された物理モデルを出発点として、コード中心の開発とモデルベース開発のそれぞれの流れ、およびその検証方法を示します。

下段に示すモデルベース開発では、第1章で示したように、Simulinkを用いてブロックを接続することで、時間関数の構造を持つモデルを直観的に設計することができます。