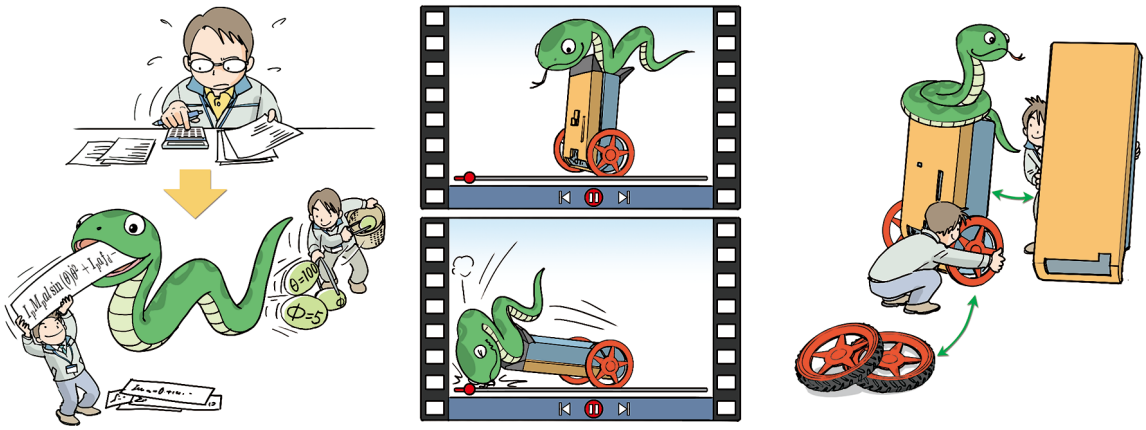
ダウンロード・データあります

第2章 複雑な計算はライブラリにお任せ！ わずか200行でグラフ表示までOK

シミュレーション①… 運動方程式のPython化

川村 聡



①計算が簡単になる　②アニメーション表示できる　③パラメータを変更できる

図1　Pythonで計算することのメリット
複雑な計算を任せられるだけでなく，アニメーション表示やパラメータ変更も楽に行えるようになる

Pythonを使い，エレガントに計算する

● 制御の数式…手計算するには複雑すぎる

第1部 第1章では，倒立車輪ロボットの挙動を手計算と表計算ソフトウェアExcelで求めてきました．しかし，これには限界があります．2自由度の簡単な振り子のモデルですら，最後は第1部 第1章 式(8)のように呪文のような数式ができてしまいました．もう少し複雑な計算になると，式があまりにも長くなり，手計算では追いつかなくなりそうです．

結果の確認には，Excelを使って簡易的にグラフ表示しましたが，できればもっと直感的に分かるようにアニメーション表示なども追加したいところです．

そこで，Pythonのライブラリを駆使した場合，先ほどの手計算と同様の結果を出すことが，どのくらい簡単になるかを試します(**図1**)．

● 複雑な計算はライブラリに任せる

解析の最初に行う数式導出には，Pythonの追加パッケージ`sympy.physics.mechanics`をメインで使います．これは物理シミュレーション用の式を自動で算出してくれる強力なモジュールです．

式の導出が終わったら，SciPyに用意されている運動方程式を解くための`scipy.integrate.solve_ivp`という関数を利用します．運動方程式の解析結果は`matplotlib.pyplot`というモジュールでグラフ描画やアニメーションの形で出力します．

● わずか200行でシミュレーションまで実装！

Pythonを使えば，こうした一連の流れをおよそ200行の短いコードで実装できます．まさに手計算の泥臭さの正反対で，エレガントな手法だと言えるでしょう．

● 開発環境は必要に応じて使い分ける

第1部で説明するPythonコードの開発環境には，JupyterLabとThonnyの2つを使っています．本章と第3章ではJupyterLabを使って数値シミュレーションを行います．第5章以降では，Thonnyを使って実機の制御を行います．開発環境は普段使い慣れたものを使っても構いません．