

Python の基本と 振動・制御工学への応用

工学博士 松下 修己
博士（情報科学） 藤原 浩幸 【共著】
博士（工学） 保手浜 拓也

コロナ社

まえがき

大学などの工業数学を考えた場合、少なくとも二つの側面がある。その代表例は数値ベースで計算することで、すべての数値が既知である Excel のような数値 (numerical) 計算である。他方は、 $ax^2+bx+c=0$ のような文字変数 (symbol) を含みながら数式を解くことである。後者の symbolic 数学では、変数にある値を代入し、残りは未知変数として変数名のまま数式展開を行い一般的に解析することができる。数値のみを扱う電卓と似て非なるところである。Python では、この二つの世界は整然と区別され、numpy と sympy のコマンド群で実行する。両群のコマンドを協調させながら一つの目的に向かってプログラムが作成される。競合ソフトウェア Mathematica も symbolic 数学を得意とするが、無料という点で Python が多くの方に歓迎されている。

板書で秩序立てて数式や数値を交えながら説明する理工系授業などでは、本ソフトウェアの利用で教育効率は各段に向上する。局所的ではあるが大学での理系工業数学の教育方法が一変しつつある。また、Excel を駆使しているような職場でも、複素数や変数を交えての計算なども簡単に扱える本ソフトウェアの導入は、仕事の仕方を変貌させる可能性がある。

工場の設計者として効率向上至上主義を教え込まれ、教育者に転じた著者自身、この Python を駆使した講義の高効率化が目下の自身の開発テーマである。例えば、理工系力学において、数式・計算に悩まされることより、図表を用いてもっと上流側の物理的考察や力学的解釈に注力した講義を展開したい。そのツールとして、Python をダウンロード (DL) し、それを活用する入門書を著した。

本書の構成は全 7 章よりなる。

第 1 章は、Python の DL およびインストール方法の手引きである。

第 2 章では、簡単な電卓計算から始まり、表計算、グラフの描画、ファイルの出し入れなど通論で 10 講義を用意。これがわかれば即実践へという入門編である。第 3 章では、フローチャートに似たソフトウェア設計図である PAD (Problem Analysis Diagram) を紹介し、論理思考過程の視覚化を説く。第 4 章では、厳選した 4 テーマ「微

分方程式、ラプラス変換、フーリエ変換、固有値問題」について、大学で学ぶ工業数学を例題中心に学ぶ。鉛筆に代わり、ソフトウェアで数学を処理する威力を体感してほしい。以上の準備のもと、筆者らの専門範囲である機械・電気系ダイナミクスの基本として「機械力学」（振動工学）を第5章に、また「制御工学」を第6章にそれぞれを述べた。最終第7章では、我々が仕事や教育で使っている重要な私設関数を紹介するので、多いに活用されたい。

Pythonなどのソフトウェア習得のコツはプログラムのコピーにある。自分の課題に近いプログラムをまずコピー、それを変形し、自分流に書き換えていく方法が最も近道であることは、等しく余人の認めるところである。大いにコピーされたい。紙面数の制約で本書では中途となったプログラムが多々あり、（詳細はWEB）のサインが入っている。コロナ社のWebページ[†]にはプログラムの全文が載っているので、DL・活用をお願いします。

書き終えてみると、ソフトウェア紹介本の執筆の大変さを思い知った。特にプログラミングでは、間違いなきことは当然であるが、紹介コマンドの取捨選択、プロミンクは短く・分かりやすくなど、推敲に推敲を重ねた。著者らの筆力の不足、浅学非才や独断にもとづく誤った説明の所在が不安でもある。万全の体制で執筆につとめたが、専門の範囲も広く、また奥も深く、山の高さを痛感している。読者の皆様の批判や叱正を得ることができれば著者の望外の喜びとするところである。

最後に、本書の図版作成などご便宜を賜った新川文登様（新川電機株式会社 社長）および同社広報関係者の皆様のご協力に感謝申し上げます。また、出版に際し格段のご支援を賜ったコロナ社に深く感謝の意を表す。

2023年12月

著者代表 松下 修己

[†] コロナ社 Web ページ <https://www.coronasha.co.jp/np/isbn/9784339032468/>

目 次

第 1 章 Python 事始め 1

1.1 Anaconda パッケージについて	1
1.2 Anaconda のインストール	1
1.3 Jupyter ノートブックと実行方法	2
1.4 簡単な計算	3
1.5 メニューの活用	4
1.6 本書を読む上での留意点	6

第 2 章 Python 速習十講 7

2.1 加 減 乗 除	7
2.2 関 数 解 析	8
2.2.1 numpy, sympy ライブラリ読み込み	8
2.2.2 数 値 化 関 数	8
2.2.3 平 方 根	9
2.2.4 対 数	9
2.3 複 素 数 計 算	10
2.3.1 虚 数 単 位	11
2.3.2 複素数の実部と虚部	11
2.3.3 複素数の絶対値	12
2.3.4 複素数の位相角	12
2.3.5 複素数の log 計算	13
2.3.6 オイラーの公式	13
2.4 関数の Plot 表現	14
2.4.1 直線・曲線のプロット	14
2.4.2 波形のプロット	16
2.5 数式による解析	17
2.6 数値式の解析	19

2.6.1	代数方程式	19
2.6.2	微積分	20
2.7	配列の階級 (class) と配列要素の操作	21
2.7.1	配列	22
2.7.2	型について	25
2.8	行列	26
2.8.1	行列の記述	26
2.8.2	ベクトルの演算	27
2.8.3	行列の演算	28
2.8.4	行列と配列の操作	29
2.8.5	線形代数方程式	31
2.9	グラフィックスの離散系 Plot 表現	32
2.10	データの入出力	35

第3章 PADによるプログラミング技法 39

3.1	PADとプログラム設計	39
3.1.1	構造化プログラム PAD	39
3.1.2	プログラム設計例	41
3.2	PADとプログラミング	42
3.2.1	プログラミング	42
3.2.2	PADに基づくプログラミング演習	45
3.2.3	PADの利用効果	47

第4章 Pythonで解く工業数学 48

4.1	微分方程式	48
4.1.1	定係数の常微分方程式の解法	49
4.1.2	変係数の常微分方程式の解法	53
4.1.3	微分方程式の数値解	55
4.1.4	1階微分方程式の図式解	57
4.1.5	2階微分方程式の図式解	59
4.2	ラプラス変換	60

4・2・1	関数とラプラス変換	60
4・2・2	微積分とラプラス変換	64
4・2・3	ラプラス変換による微分方程式の解法	65
4・3	フーリエ級数展開の実際	68
4・3・1	内在する \cos/\sin 波の検出	68
4・3・2	フーリエ級数展開	70
4・3・3	離散フーリエ級数展開	75
4・3・4	離散フーリエ級数展開のパラメータ	80
4・4	固有値解析	83
4・4・1	微分方程式と固有値問題	84
4・4・2	係数行列と固有値	86
4・4・3	固有ベクトルの直交性	86

第5章 機械力学 91

5・1	1自由度系の振動	91
5・1・1	自由振動	91
5・1・2	力加振時の強制振動	93
5・1・3	不つり合い振動	95
5・1・4	入出力の線形関係	98
5・1・5	影響係数法バランス	98
5・1・6	3点法バランス (4 run balance)	103
5・2	ラグランジュの方程式	105
5・3	多自由度系の運動方程式	114
5・3・1	多自由度系	114
5・3・2	2自由度系の運動方程式	114
5・3・3	多自由度系運動方程式の行列	115
5・4	多自由度系のモード分離	117
5・4・1	固有値問題と固有ベア	117
5・4・2	直交性	117
5・4・3	モード分離	118
5・5	モード解析	121
5・5・1	モーダルモデル	121
5・5・2	周波数応答	123

5.5.3	過 渡 応 答	126
5.6	非線形振動解析	129
5.6.1	非線形系の平均法による近似解	129
5.6.2	時刻歴応答シミュレーション	130
5.6.3	等傾斜法による図式解	130

第 6 章 制 御 工 学 136

6.1	伝達関数と時間応答	136
6.1.1	動的システムの表現	136
6.1.2	ブロック線図の構成	139
6.1.3	インパルス応答とステップ応答	141
6.2	周 波 数 応 答	144
6.2.1	周波数応答とは	144
6.2.2	ボ ー ド 線 図	145
6.2.3	ベクトル軌跡 (ナイキスト線図)	146
6.3	フィードバック制御	153
6.3.1	フィードバック制御系の特性	153
6.3.2	フィードバック制御の PID 制御	155
6.3.3	実用的な位相進み回路	159
6.4	フィードバック制御系の安定性	161
6.4.1	安定性の概念	161
6.4.2	特性方程式・特性根	162
6.4.3	フルビッツの安定規範	163
6.4.4	極配置と根軌跡	164
6.4.5	開ループ特性	166
6.4.6	ボード線図による固有振動数および安定余裕判別	167
6.4.7	ベクトル軌跡による安定性解析	169
6.4.8	閉ループ特性	173

第 7 章 私 設 関 数 177

1

Python 事始め

1・1

Anaconda パッケージについて

本節では Python を使用するための準備について解説する。特に算術計算で使用するためには数多くのライブラリを準備する必要がある。本書では算術計算に必要なライブラリや実行環境をまとめてインストールできる **Anaconda** というパッケージを利用している。おもなライブラリとして、算術計算 **Numpy**、シンボリック変数計算 **Sympy**、グラフ描画 **matplotlib**、実行環境 **Jupyter Notebook** などが入っている。機械・電機・制御などのシステム工学計算に必要なものがおよそそろっている。

1・2

Anaconda のインストール

Anaconda のソフトウェアは下記の web サイト[†]からダウンロードできる。

<https://www.anaconda.com/products/individual>

上記サイトを表示したら、一番下までスクロールし、自分の OS に合わせてクリックしてインストーラをダウンロード (DL) を実行すると実行環境ができる。本書ではインストールの詳細は省略するが、下記のサイトで解説されているので、参考にされたい。

<https://www.python.jp/install/anaconda/windows/install.html>

[†] 本書に示す URL はすべて 2023 年 12 月現在。

1.3 Jupyter ノートブックと実行方法

インストールが完了したら、プログラムの起動を確認する。図 1.1 に示すように、①のスタートメニューをクリックすると、②の Anaconda3 のフォルダが見える。そのフォルダをクリックすると③のように中身のソフトウェアが見える。この中から④の“Jupyter Notebook”を選択し起動する。

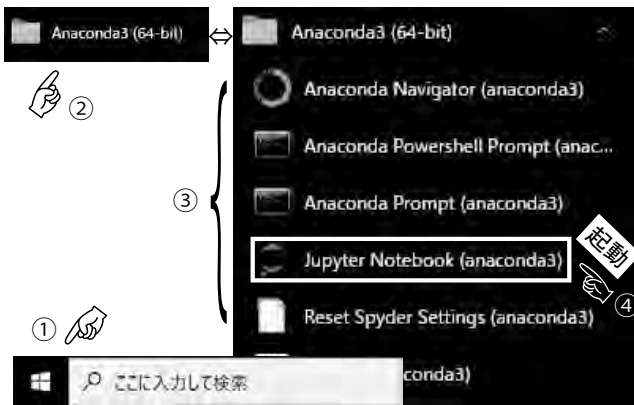


図 1.1 スタートメニュー

Jupyter Notebook は標準ブラウザ (Windows のときは Edge) で動作するようになっており、図 1.2 (左上) は Jupyter Notebook が起動した直後の画面である。表示されているフォルダはユーザーの“Home”フォルダであり、標準の Desktop などのフォルダが見えるはずである。ここから、⑤“Documents”、⑥“Python Scripts”と順にクリックする。ここから既存の Notebook ファイル⑦を選択して開くことができる。ここでは、fujiwara.ipynb を選択している。既存ファイルがない場合は新規にプログラムを書き始めるとして、右上に表示されている⑧“New ▼”をクリックする。その後、⑨“Python3 (ipykernel)”メニューを選択すると新しく⑩ Notebook “Untitled1.ipynb”が開かれる。⑩のUntitled1をクリックすると名前の変更ができる。この段階で計算用の Notebook が準備され、⑪ではプロンプトが点滅し入力のタイプインを待っている。これで準備完了である。

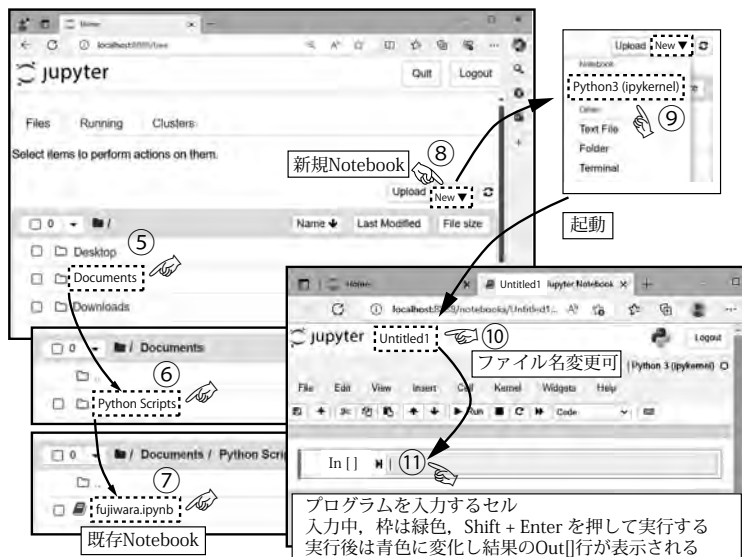


図 1.2 Jupyter Notebook の起動直後の画面

これから計算を開始していく。In [] の右枠にカーソルが点滅するので、ここにプログラムを記述して、**SHIFT** + **ENTER** を押すと計算が実行される。この枠一つが実行単位となり、**セル (cell)** と呼ばれる。実行が完了すると直下に計算結果として Out [] が現れる。[] 内の数字はこの Notebook でセルが実行された順番であり、数字が表示されていない場合は未実行、[*] が表示されているときは実行中であることを示す。

1.4 簡単な計算

早速、実行してみよう。簡単な計算であれば、数値計算のライブラリを **Import** せずに実行できるので、下記の例に従って進めてほしい。


[1] 足し算 +

- ・整数値 6+3 をセルに入力
[Shift] + [Enter] を押す。
- ・結果 9 が表示される。

In [1]: 6+3

Out [1]: 9 **整数で答えが表示**

〔2〕引き算 -

<ul style="list-style-type: none"> ・例 6.0-3 ・結果 3.0 	In [2]: <code>6. -3</code> Out [2]: 3.0  入力が小数のとき答えが小数値
---	---

〔3〕掛け算 *, 割り算 / と四則演算

加減乗除を使った練習をしてみよう。

<ul style="list-style-type: none"> ・例 1+2, 4-3, 4*6, 10/5 ・複数行の結果を表示するときは Print() 関数を使用 ・結果 3,1,24,2.0 	In [3]: <code>print (1 + 2, 4 - 3) print (4 * 6, 10/5)</code> Out [3]: 3 1 24 2.0 掛け算 '*'(アスタリスク) 割り算 '/'(スラッシュ)
<ul style="list-style-type: none"> ・例 $1+2\times 3\times 4 - 6\div 3 = 23$ ・結果 23.0 	In [4]: <code>1+2*3*4-6/3</code> Out [4]: 23.0 一般的な優先順位 掛け算, 割り算が優先

〔4〕割り算の商 // と除り %

<ul style="list-style-type: none"> ・例 7 割る 3 の商と除り ・結果 2,1 	In [5]: <code>print (7//3, 7% 3)</code> Out [5]: 2 1 商 '/'(スラッシュ,スラッシュ) 除り '%'
--	---

〔5〕べき乗 **

<ul style="list-style-type: none"> ・例 2^8 ・結果 256 	In [6]: <code>2**8</code> Out [6]: 256 べき乗は '**'(アスタリスクアスタリスク)
--	---

1・5 メニューの活用

記載したプログラムの保存や暴走したプログラムの中止などはメニューから選択する。図 1・3 におもなものを記載する。

① 実行したファイルのオープンと保存

①(a) 既存のファイルを開くとき **File → Open**

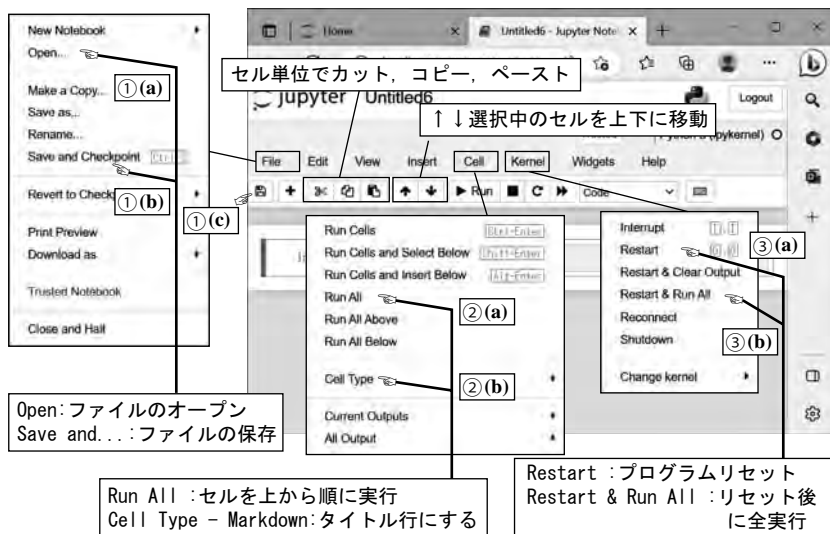


図 1-3 Jupyter notebook のメニューと操作

- ①(b) 保存するとき **File** → **Save and Checkpoint** または
 ①(c) 保存アイコンを押す
- ② セルの操作
- 既存のファイルを開いた直後などにすべてのセルを実行するときは
- ②(a) **Cell** → **RunAll**
- プログラムの区切りとしてタイトル、コメントを入れる場合はセルを選択した状態で
- ②(b) **Cell** → **Cell Type** → **Markdown**
- を選ぶ。文字の先頭に '#' を付けてから、[SHIFT] + [ENTER] を押すと文字列がフォントの大きいゴシック体で表示される。先頭#の数を '##' と増やすとフォントが小さくなっていく。
- ③ カーネル操作
- 何らかの理由でプログラムが止まったり、暴走したときは
- ③(a) **Kernel** → **Restart**
- で、計算をリセットし、変数を初期化する。また、プログラム全体を始めから終

索引

索引は以下の順に掲載する。

1. コマンド一覧 (計 115) np. および sp. の後がアルファベット順に並んでいる。
ただし、私設関数 `my_○○○○` は第 7 章 p. 177 に掲載。
2. 専門用語一覧 (計 281)

1. コマンド一覧

記号コマンド	解 説	掲載ページ
<code>mat[n], mat[:, n]</code>	要素 <code>mat</code> の n 行目 / n 列目を取り出す	30, 149
<code>{a:X, b:Y}</code>	<code>dict</code> 辞書 <code>{a:X, b:Y}</code>	25
<code>[a, b, c]</code>	<code>list</code> リスト <code>[a, b, c]</code>	24
<code>{a, b, c}</code>	<code>set</code> セット 配列定義	26
<code>(a, b, c)</code>	<code>tuple</code> タプル <code>(a, b, c)</code>	25
<code>//, %, **</code>	割り算の商, 余り, べき乗	4
<code>_</code> (アンダーバー)	前の実行結果	13
<code>+, -, *, /</code>	加減乗除	3, 4
<code>i+=a, i-=a, i*=a, i/=a</code>	<code>i</code> に <code>a</code> を足す, 引く, かける, 割る	45
コマンド	解 説	掲載ページ
<code>np.abs()</code> , <code>sp.Abs()</code>	絶対値	12, 75, 78, 96, 104
<code>np.angle(z)</code> , <code>sp.arg(z)</code>	複素数の位相角	12
<code>sp.apart()</code>	部分分数展開	121
<code>list.append(a)</code> , <code>np.append(list, a)</code>	<code>list</code> に要素 <code>a</code> を追加	72, 74
<code>np.arange(ns, ne, nd)</code>	<code>ns</code> から <code>ne</code> まで <code>nd</code> 刻みのベクトル作成	72, 134
<code>np.array([1, 2])</code>	数値リスト <code>[1, 2]</code> を NumPy 用行列定義	22, 46
<code>Axes3D(fig)</code>	3D グラフの描画準備	34
<code>plt.bar(x, y)</code>	リスト <code>x, y</code> を棒グラフで表示	45, 72
<code>sp.BlockMatrix([[m1, m2], [m3, m4]])</code>	行列を組み合わせて大きな行列作成	31
<code>np.conjugate(z)</code>	複素数の共役	85
<code>np.cos(x)</code> , <code>sp.cos(x)</code>	<code>cos</code> 関数	34, 71, 186
<code>vec1.cross(vec2)</code>	<code>vec1</code> と <code>vec2</code> の外積	28

<code>pd.DataFrame(dat)</code>	<code>dat</code> を <code>pandas</code> class に変更	38
<code>def ... return</code>	私設関数定義	56
<code>sp.deg(rad)</code> , <code>np.deg2rad(deg)</code>	radian を度に変換, 度を radian に変換	12, 13
<code>sp.Derivative(f(x), x).doit()</code>	<code>f</code> を <code>x</code> で微分	20
<code>sp.det(mat)</code>	<code>mat</code> の行列式	164
<code>np.diag([x1, x2, ...])</code>	対角行列	29, 119
<code>sp.diff(f(x), x)</code> , <code>sp.diff(f(x), x, 2)</code> <code>eq.diff(x)</code>	<code>f</code> を <code>x</code> で微分, <code>f</code> を <code>x</code> で 2 階微分 式 <code>eq</code> を <code>x</code> で微分	20, 50
<code>display()</code>	数式, 行列表示	13
<code>np.dot(mat1, mat2)</code>	行列の掛け算	188
<code>vec1.dot(vec2)</code>	ベクトルの内積	28
<code>sp.dsolve()</code>	微分方程式を解く	50
<code>np.linalg.eig()</code>	固有値問題 (固有値, 固有ベクトル)	85
<code>sp.Eq(siki)</code>	方程式 <code>siki=0</code> の定義	18, 50
<code>np.exp(x)</code> , <code>sp.exp(x)</code>	e^x	9, 82, 97
<code>sp.expand()</code>	数式を展開	19, 52
<code>zu1.extend(zu2)</code>	<code>sp.plot</code> の描画 <code>zu1</code> にグラフ <code>zu2</code> を重ねる	52, 67
<code>sp.eye(n)</code>	$n \times n$ の単位行列	29
<code>sp.factor(多項式)</code> , <code>f.factor()</code>	因数分解, 多項式 <code>f</code> の因数分解	19, 121, 163
<code>np.fft.fft()</code>	高速フーリエ変換	78, 128
<code>np.fft.fftfreq()</code>	離散フーリエ用の周波数軸を設定	128
<code>plt.figure(figsize=(3, 3))</code>	図のサイズ指定 (横幅の最大値 10)	105
<code>sp.flatten(mat)</code>	多次元リストを 1 次元に平坦化	165
<code>float(x)</code>	<code>x</code> を浮動小数点数として扱う	37
<code>np.floor()</code>	数値の整数部分	79
<code>sp.fourier_series(関数).truncate(n)</code>	関数の n 次までのフーリエ級数展開	72
<code>sp.fraction()</code>	分数を (分子 [0], 分母 [1]) として出力	163, 188
<code>f=sp.Function()</code>	<code>f</code> を関数として定義	56, 106
<code>plt.grid()</code>	グラフの格子を ON, OFF する	75
<code>sp.Heaviside()</code>	ヘビサイド関数	63
<code>if</code> 条件 :	条件が真のとき, つぎの行を実行	46
<code>x.imag</code> , <code>sp.im(x)</code>	複素数 <code>x</code> の虚部 (<code>np</code> では変数の後ろに <code>.imag</code>)	11, 12, 178
<code>int()</code> , <code>np.int64()</code>	整数指定, 64 ビット整数	79, 128

<code>sp.integrate()</code>	積分	21, 134
<code>np.linalg.inv(mat), mat.inv()</code>	行列 mat の逆行列	90, 112
<code>sp.inverse_laplace_transform()</code>	逆ラプラス変換	66, 142
<code>1j, sp.I</code>	虚数単位 (np のときは <code>1j</code> のように数字必須)	11, 94, 146
<code>len(vec), len(mat), len(mat[0])</code>	vec の要素数, mat の行数, mat の列数	30
<code>eq.lhs, eq.rhs</code>	方程式 eq の左辺, 右辺	50
<code>sp.linsolve((mat, vec))</code>	線形代数式 mat*x=vec の x を解く	31
<code>np.linspace(ns, ne, nd)</code>	ns から ne まで nd 刻みのベクトル作成	58, 82
<code>np.log(x)</code> <code>np.log2(x), np.log10(x)</code>	底が e (自然対数) 底が 2, 10 (常用対数) の対数	9
<code>sp.log(x), sp.log(x, a)</code>	自然対数, 底が a の対数	10, 178
<code>sp.Matrix(mat)</code>	行列定義	22, 121
<code>sp.matrix_multiply_elementwise(v1, v2)</code>	v1 と v2 の要素同士の積	28
<code>np.meshgrid(xx, yy)</code>	メッシュデータ作成 → <code>plot_surface</code> で描画	34, 58
<code>np.mod(a, n)</code>	数値 a を除数 n で割り算したときの余り	45
<code>sp.N(x)</code>	数値で表現	8, 153
<code>np.ones((n, m))</code>	要素 1 の n 行 m 列の行列	58
<code>pass</code>	ループをパスする	47
<code>np.pi, sp.pi</code>	円周率	8
<code>sp.plot(fx, (t, 0, 5))</code>	関数のグラフ表示	15, 52
<code>plt.plot(mat[:, 0], mat[:, 3])</code>	行列 mat の 0 列目を x 軸, 3 列目を y 軸とするグラフ表示	32
<code>ax.plot_surface(X, Y, h)</code>	3D グラフ作成	34
<code>sp.plotting.plot3d(h(x, y))</code>	高さ h(x, y) の 3D 図を描画	35
<code>print()</code>	印字	7
<code>%pwd</code>	現在のディレクトリを表示	36
<code>np.radians(rad), sp.rad(deg)</code>	度をラジアン単位にする	14, 97
<code>np.random.randint(0, 6)</code>	0 ~ 6 までの整数の乱数発生	45
<code>pd.read_table("dat.txt")</code>	dat.txt からデータ読み込み	36
<code>x.real, sp.re(x)</code>	複素数 x の実部 (np では変数の後ろに real)	11, 12, 97
<code>np.round()</code>	四捨五入	78

<code>mat.rows, mat.cols</code>	<code>mat</code> 行列の行数, 列数	30
<code>zu.show(), plt.show()</code>	図 <code>zu</code> を表示, <code>matplotlib</code> のグラフ表示	16, 52
<code>sp.sign()</code>	<code>sign</code> 関数	97
<code>sp.simplify()</code>	数式を簡略化	12, 50
<code>np.sin(), sp.sin()</code>	<code>sin</code> 関数	16, 50, 63
<code>sp.solve()</code>	方程式を解く	18, 50, 141
<code>solve_ivp()</code>	微分方程式 (時刻歴応答) を解く	56, 132
<code>np.sqrt(), sp.sqrt()</code>	平方根関数	9
<code>plt.streamplot()</code>	流線	58
<code>f.subs(x, xin), f.subs(para)</code>	<code>f</code> の <code>x</code> に <code>xin</code> を代入, <code>f</code> にパラメータ <code>para</code> を代入	9, 50, 93, 159
<code>sum()</code>	要素の合計	71
<code>x=sp.Symbol('x')</code> <code>x1, x2, x3=sp.symbols('x1, x2, x3')</code>	<code>x</code> をシンボルとして扱う <code>x1, x2, x3</code> をシンボルとして扱う	8, 50, 106, 142, 178
<code>plt.text</code>	指定した座標に文字を表示	184
<code>dat.to_csv('dat.txt')</code>	データ <code>dat</code> を <code>dat.txt</code> として CSV 形式で出力	38
<code>np.transpose(mat), mat.T</code>	行列 <code>mat</code> の転置行列	30, 85
<code>type(a)</code>	変数の型	8, 25, 36,
<code>dat.values</code>	<code>pandas</code> class のデータ <code>dat</code> を <code>np</code> class の値に変換	37
<code>sp.var('x'), sp.var('x1:3')</code>	<code>x, (x1, x2, x3)</code> をシンボルとして扱う	93, 141
<code>while</code>	繰り返し計算	45
<code>plt.xlabel("x"), plt.ylabel("y")</code>	図の下に " <code>x</code> ", 左横に " <code>y</code> " を表示	17, 34
<code>plt.xlim(0, 100), plt.ylim(0, 5)</code>	横軸を 0 ~ 100 に, 縦軸を 0 ~ 5 にする	83
<code>plt.xscale('log')</code> <code>plt.yscale('log')</code>	横軸を対数にする 縦軸を対数にする	83
<code>np.zeros([n, m]), sp.zeros([n, m])</code> <code>np.zeros([n, m], dtype=complex)</code>	<code>n</code> 行, <code>m</code> 列ゼロ行列作成 <code>n</code> 行, <code>m</code> 列ゼロ行列作成 (複素数)	30, 47
ライブラリ	解 説	掲載ページ
<code>import numpy as np</code>	数値演算用パッケージ	8, 45, 48, 92, 140, 178
<code>import matplotlib.pyplot as plt</code>	グラフ表示パッケージ	8, 45, 48, 93, 140, 178
<code>plt.rcParams['figure.figsize'] = (x, y)</code> <code>plt.rcParams['axes.grid'] = True</code>	グラフの標準の大きさ (<code>x, y</code>) を横幅最大 10 で指定, グラフに格子を標準で入れる	8, 45, 48, 93, 140, 178

<code>from scipy.integrate import solve_ivp</code>	微分方程式パッケージ	93
<code>import sympy as sp</code>	数式処理パッケージ	8, 45, 48, 92, 140, 178
<code>import pandas as pd</code>	統計処理パッケージ	35

2. 専門用語一覧

<p>【あ】</p> <p>余り (mod, %) 4, 44</p> <p>安定限界 161, 166</p> <p>安定性 161</p> <p>【い】</p> <p>位相角 12</p> <p>位相差, 位相曲線 145</p> <p>位相進み回路 149, 159</p> <p>位相 (平) 面 59, 131</p> <p>位置エネルギー 106</p> <p>一巡伝達関数 → 開ループ伝達関数</p> <p>一般解 49</p> <p>一般化座標 106</p> <p>一般化力 106</p> <p>因数分解 19</p> <p>インストール 1</p> <p>インパルス応答 127, 142</p> <p>インパルス関数 62</p> <p>インパルス振動波形 81</p> <p>インピーダンス 139</p> <p>【う】</p> <p>宇宙 54</p> <p>運動エネルギー 106</p> <p>【え】</p> <p>影響係数法バランス 98</p> <p>衛星 157</p> <p>円周率 np.pi, sp.pi 8</p> <p>【お】</p> <p>オイラーの公式 10, 13</p> <p>折り返し周波数 81</p>	<p>【か】</p> <p>カーネル 5</p> <p>階 級 22, 25, 182</p> <p>階乗計算 45</p> <p>外 積 28</p> <p>解析解 18</p> <p>回転数, 回転角速度 95</p> <p>外 乱 157</p> <p>外 力 115</p> <p>外力振動 (応答) 66, 67</p> <p>開ループ伝達関数 156</p> <p>開 (ループ) 特性 156, 166</p> <p>加減乗除 7</p> <p>重ね描き 181</p> <p>加振周波数 93</p> <p>加振力 → 調和加振力 93</p> <p>型 type 25</p> <p>過渡応答 126</p> <p>簡単化 simplify 12, 109</p> <p>【き】</p> <p>規格化 85</p> <p>奇関数 71</p> <p>規準座標系 → モード座標</p> <p>基本周期 70</p> <p>基本周波数 (成分) 68, 80</p> <p>逆行列 90</p> <p>求 解 50</p> <p>求 根 19</p> <p>共振曲線 94, 126</p> <p>共振振幅 (ピーク値) 95, 126</p> <p>強制振動 93</p> <p>行 列 26, 115</p> <p>行列操作 (和, 差, 長さ, 抽出など) 29, 30</p> <p>極配置 164</p>	<p>虚数単位 11</p> <p>虚 部 11</p> <p>「距離」ベクトル <i>D</i> 172</p> <p>【く】</p> <p>偶関数 71</p> <p>組立行列 31</p> <p>クリホフ・ボゴリユエウボフの解析法 129</p> <p>【け】</p> <p>係数行列 85</p> <p>ゲイン曲線 145</p> <p>ゲイン交差周波数 167</p> <p>ゲイン余裕 167</p> <p>減衰行列 115</p> <p>減衰自由振動 92</p> <p>減衰定数 (係数) 106</p> <p>減衰比 91</p> <p>【こ】</p> <p>効果ベクトル 100</p> <p>高次多項式 19</p> <p>剛性行列 115</p> <p>——の重量 → 重量</p> <p>構造化プログラム 39</p> <p>高速フーリエ変換 → FFT</p> <p>交代行列 86, 88</p> <p>剛体ロータ 111</p> <p>合同変換 89</p> <p>固有角振動数 91</p> <p>固有値 83, 117</p> <p>固有値解析 83</p> <p>固有値問題 117</p> <p>固有ベア 84, 117</p> <p>固有ベクトル 83, 117</p> <p>——の直交性 86</p>
--	---	--

固有モード
→ 固有ベクトル
根軌跡 164

【さ】

最大表示周波数 f_{\max} 81
最大値検出 46
三角波 (加振) 63, 67
サンプリング 75, 82
サンプリング個数 80
サンプリング周期 80
サンプリング周波数 80

【し】

時間平均 69
時間窓 80
時間領域 61, 80, 136
磁気浮上制御系 162
軸受座標 111
時刻歴応答・時間応答
130, 136, 188

辞書 **dict** 25
実部 11
質量 (行列) 91, 115
写像 169
周期関数 69, 70
集合のベン図 26
重心座標 111
自由振動 91
周波数応答 144
周波数領域 80
出力行列 115
巡回行列 46
状態変数 84
状態方程式 84
常微分 20
常微分方程式 49
初期値 49
初期値振動 (応答) 66, 67
自励振動 130
シンクロスコープ図 97
振幅比 145

【す】

水位 (制御) 系 55, 138, 155
数式による解析 17

数値化 8
図式解 20, 57
ステップ応答 143, 154, 175
ステップ関数 62
ストロボ図 97

【せ】

正弦半波 (加振) 63, 67
静たわみ 93
積分制御動作 155
積分定数 49
絶対値 12
節点 115
セット **set** 26
セル 3, 5
セルを実行 5
全応答 66, 67
線形近似 109
線形代数方程式 31
選択 **if** 文 42, 43

【そ】

相似変換 89, 90
属性定義 17

【た】

対角化 89
対角行列 29
対称行列 86
対数 9
代数方程式 19
ダイナミックダンパ 107
ダウンロード 1
多項式展開 19
多自由度系 114
タプル **tuple** 25
試し重り 99
単位行列 29
タンク → 水位系

【ち】

置換コマンド 19
地球脱出速度 59
調和加振力 93
調和関数振幅 69
重量 69, 115

直流成分 68
直交行列 86, 89
直交条件 87
直交性 86, 117, 122

【つ】

ツールボタン 6

【て】

定係数の常微分方程式 49
定常振幅 136
定常偏差 155, 157, 176
定積分 21
ディレクトリ 35, 36
データの入出力 35
テキストファイル 36
デシベル (dB) 145, 178
電気回路 140
伝達関数 65, 94, 136, 151, 188

【と】

等傾斜法 130
同次微分方程式 49
動的システム 136
特殊解 49
特性根 162
特性方程式 162
特解 49

【な】

ナイキスト線図 99, 146, 152
内積 28
流れ図 40

【に】

入力行列 115

【ね】

粘性減衰係数 → 減衰定数

【の】

のこぎり波 73, 76

【は】

媒介変数 181
配列 21

配列操作 (抽出, 結合, 整形など) 23, 24, 29
 ばね定数 91, 106
 バランス修正 100
 反復 (リピート) 39, 43

【ひ】

微積分 20
 非線形 (係数) (項) 109, 130
 非線形振動 129
 左固有ベクトル 90
 左半面 169
 非同次微分方程式 65
 微分操作 20
 微分方程式 48, 84, 136

【ふ】

ファイルのオープン 4
 フィードバック制御 153
 プーリ 109
 フーリエ級数展開 68, 70
 フーリエ係数 70
 複素振幅 94, 95
 複素数 10
 複素フーリエ係数 70
 不減衰系 117
 不つり合い振動 (応答) 95, 174
 不定積分 21
 フーリエ (振幅) スペクトル 71
 フルピッツの安定規範 163
 ふれまわり 97
 ブロック線図 139
 分解能 81

【へ】

平均法 129
 平方根 9
 閉 (ループ) 特性 156, 173
 ベクトル 27
 ベクトル軌跡
 → ナイキスト線図
 ベル 54
 偏角 145, 178
 変係数の常微分方程式 53

偏重心 98

【ほ】

ボード線図 145
 ポーラ線図 183

【ま】

窓時間 71

【み】

右固有ベクトル 90
 ミラー現象 79

【も】

モーダルモデル 121
 モード解析 121
 モード (質量, 剛性, 減衰) 行列 122
 モード座標 118, 121, 123
 モードパラメータ 118
 モード分離 118
 モード別伝達関数 119, 123
 問題分析図 → PAD

【ら】

ライン数 L 81
 ラグランジュの方程式 105
 ラプラス逆変換 61, 141
 ラプラス変換 60, 136, 141
 ラプラス領域 (s 領域) 61, 137

【り】

離散データ 75
 離散フーリエ級数展開 (DFT) 75, 76
 離散系 Plot 表現 32
 リスト **list** 25
 リニア表示 81
 リピート 39, 43
 リミットサイクル 133
 流線 57
 流速 58

【れ】

連接 39

連立方程式 18

【ろ】

ロータ 96
 ログ表示 81

【数字】

1 次遅れ系 139, 146
 1 次遅れ要素 146
 2 次遅れ要素 147
 3 点法バランス 103
 4 run blance
 → 3 点法バランス

【英字】

Anaconda 1
 class → 階級
 CSV (カンマ区切りファイル) 35, 38
 CFT → フーリエ級数展開
dB polar, db ポーラ 171, 183
 DiracDelta
 → (遅れ) インパルス関数
 Excel 38
 FFT (高速フーリエ変換) 76, 80, 127
 FSE → フーリエ級数展開
 Hermitian 行列 86
 Heviside 関数
 → (遅れ) ステップ関数
 Jupyter Notebook 2
 Kernel 5
 log 13
 Mathematica 42
 PAD 39
 PID 制御 155
 Plot 表現 14
 Python 42
 Q 値 173
 skew - Hermitian 86
 s 領域 → ラプラス領域
 unitary 86
 Van der Pol 方程式 130

— 著 者 略 歴 —

松下 修己 (まつした おさみ)

1972年 東京大学大学院工学系研究科博士
課程修了 (機械工学専攻)
工学博士
1972年 株式会社日立製作所機械研究所勤務
1993年 防衛大学校教授
2010年 防衛大学校名誉教授

保手浜 拓也 (ほてはま たくや)

2005年 神戸大学大学院自然科学研究科博士
後期課程修了 (システム機能科学専攻)
博士 (工学)
2005年 独立行政法人 産業技術総合研究所
にて学振特別研究員や産総研特別研
究員など博士研究員として勤務
2016年 三菱電機株式会社先端技術総合研
究所勤務
現在に至る

藤原 浩幸 (ふじわら ひろゆき)

1998年 東北大学大学院情報科学研究科博士
後期課程修了 (システム情報科学専攻)
博士 (情報科学)
1998年 防衛大学校助手
2005年 防衛大学校講師
2007年 防衛大学校准教授
2020年 防衛大学校教授
現在に至る

Python の基本と振動・制御工学への応用

Fundamentals of Mechanical Vibration-Control Engineering with Python

© Matsushita, Fujiwara, Hotehama 2024

2024年 2月 22日 初版第1刷発行



検印省略

著 者 松 下 修 己
藤 原 浩 幸
保 手 浜 拓 也
発 行 者 株式会社 コロナ社
代 表 者 牛 来 真 也
印 刷 所 壮 光 舎 印 刷 株 式 会 社
製 本 所 株式会社 グリー ン

112-0011 東京都文京区千石 4-46-10

発 行 所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-03246-8 C3053 Printed in Japan

(柏原)



ICOPY < 出版者著作権管理機構 委託出版物 >

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構 (電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp) の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。