
データ
サイエンス
千葉大学ASCENT-6E
プログラミング講座
【改訂第2版】

第 1 部

データとサイエンス

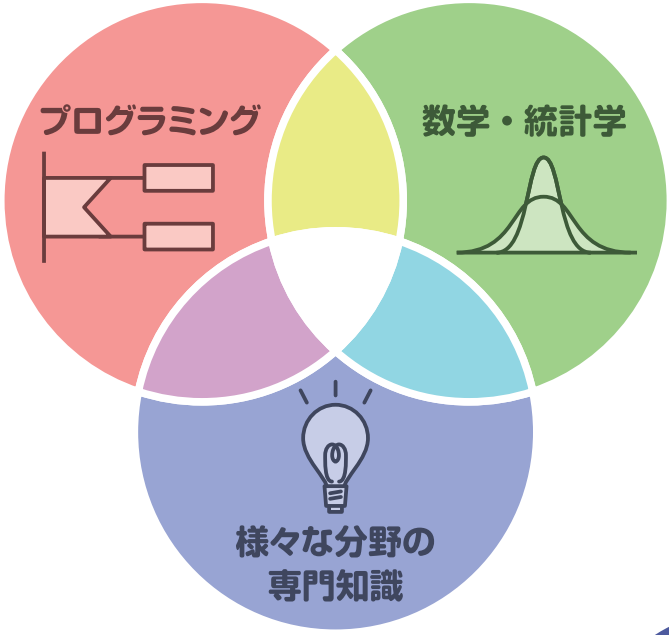
第 2 部

プログラミング

第 1 部

データとサイエンス

データサイエンスの 3 要素



いろいろなデータを数理的に扱う
「データサイエンス」は
科学研究の基本になっているんだ！

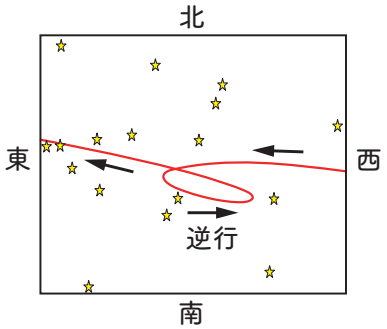
第1部
01

観測・実験に基づくサイエンス

近代科学では、観測や実験によって得られたデータを分析することによって新しい現象を発見したり、法則を見出したりします。その例として、惑星の観測からニュートンによる万有引力の法則の発見に至った歴史を振り返ってみます。

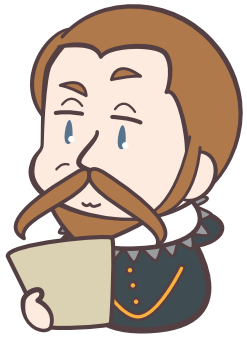
惑星の運動

夜空に輝く星の中には、北斗七星のように星座の形を変えずに日周運動をする恒星と、恒星との位置関係を刻々と変えながら運動する惑星があります。右図は地球から見たときの火星の位置です。恒星はこの図の上で静止していますが、火星は日々位置を変え、逆向きに運動することもあります。



ティコ・ブラーエによる観測

ティコ・ブラーエ（1546-1601）は、デンマーク王の援助を受けて天体の位置を精密に測定しました。特に、16年間にわたる火星の位置の観測データを残し、これが助手のケプラーに受け継がれました。



ケプラーの法則の発見

ドイツの天文学者ヨハネス・ケプラー（1571-1630）はティコ・ブラーエが残した火星の観測データを解析して以下の３つの法則を見出しました。

① 惑星は太陽を焦点とする楕円軌道上を運動している。
② 太陽と惑星を結ぶ線分は等しい時間には等しい面積を覆う。
③ 惑星の太陽からの平均距離の3乗は公転周期の2乗に比例する。

ケプラー以前には惑星の軌道は円だと思われていました。火星の軌道が円から少しずれた楕円であると発見されたことが、ニュートンによる大発見につながりました。

万有引力の法則の発見

アイザック・ニュートン（1642-1727）は、すべての物体に互いに引き合う力（万有引力）が働くという着想を得、この力の大きさが物体の質量に比例し、物体間の距離の2乗に反比例するなら、ケプラーの3法則を説明できることを数学的に示しました。この過程で見出された運動の法則を用いて、様々な物体の運動を理解し、予測することができるようになりました。

ニュートンはこれらの研究成果を「自然哲学の数学的諸原理」（プリンキピア）という著作にまとめ、古典力学を確立しました。

しっかりした観測データが残されていたから、法則が発見できたんだね

第1部
02

巨大データを用いたサイエンス

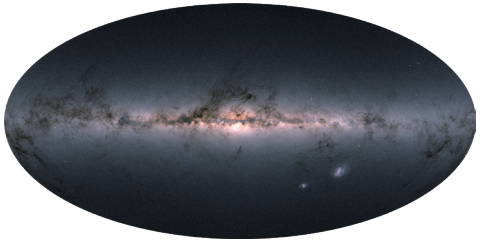
コンピュータネットワークの発達に伴い、膨大なデータを短時間で取得することが可能になってきました。たとえば、携帯電話を用いて、人の移動の様子を把握することができます。商品の購入情報なども瞬時に集めることができます。

科学研究データの巨大化

科学研究に用いられるデータも巨大化しています。20世紀末に行われたヒトゲノム計画では、人間ひとり分のDNA配列（約60億ビット）を10年間かけて解読しました。現在ではDNAシーケンサーを用いて1日で解読でき、病気の原因の解明や治療法の研究に活用されています。

天文観測の精密化

天文学においては、人工衛星を用いた大気圏外からの観測や、大気揺らぎを補正する技術の発達によって高い解像度が達成され、得られるデータ量が増大しています。

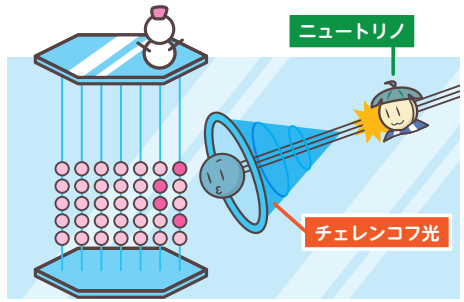


GAIA衛星のデータに基づく全天画像。光の帯は天の川。
https://www.cosmos.esa.int/web/gaia/gaiadr2_gaiaskyincolour

たとえば、GAIA衛星は10億個以上の恒星の位置を正確に測定しました。地球が太陽のまわりを回転するにつれて星の位置が変化することを用いて星までの距離を測定することができます。銀河系内の恒星の観測データをもとに、銀河系が過去に他の銀河と衝突した痕跡などが見つかっています。

高エネルギーニュートリノをとらえる

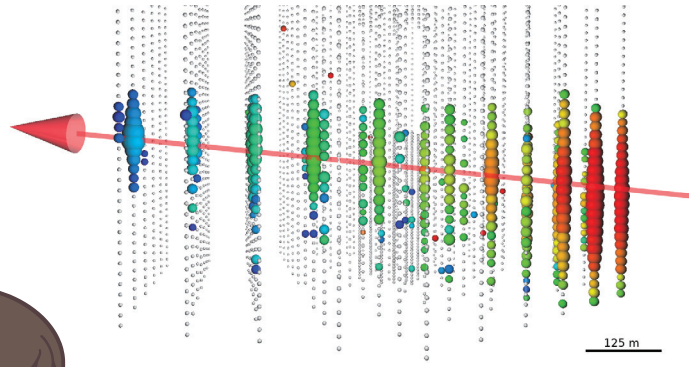
千葉大学は、南極の氷を用いて宇宙から飛来する高エネルギーニュートリノを捉えるIceCube(アイスキューブ）プロジェクトに参画しています。ニュートリノは他の物質とほとんど相互作用しないため、宇宙空間を直進します。これを観測することにより、高エネルギー粒子（宇宙線）の放射天体を明らかにできると期待されています。



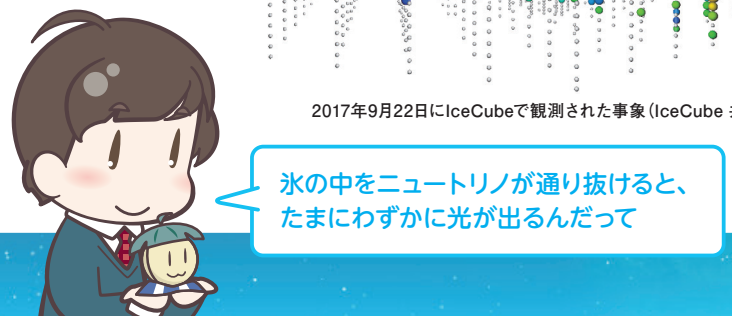
南極に建設されたニュートリノ観測装置IceCube。

IceCubeは、1立方キロメートルの氷の中に数珠つなぎに埋め込まれた5000個以上の検出器（光電子増倍管）を用いて、ニュートリノが氷と相互作用して発する青い光（チェレンコフ光）を捉えます。検出器が捉えた光子の個数や到達時間のデータを解析することで、到来方向やエネルギーを決定することができます。

千葉大学IceCubeチームは2012年に1PeV（PeVはエネルギーの単位で10の15乗電子ボルト）以上のエネルギーを持つ宇宙ニュートリノを捉えることに世界で初めて成功しました。下図に宇宙ニュートリノの観測例を示します。光を捉えた検出器に色をつけています。



2017年9月22日にIceCubeで観測された事象（IceCube 共同実験提供）



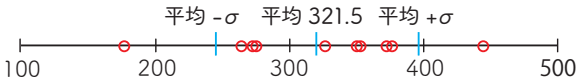
第1部
03

データ解析と計算機

データを活用するためには、データを整理して平均値や、ばらつきの大きさを計算したり、データどうしの関係を数値化したりすることが必要になり、計算機が用いられます。

ばらつきの大きさ

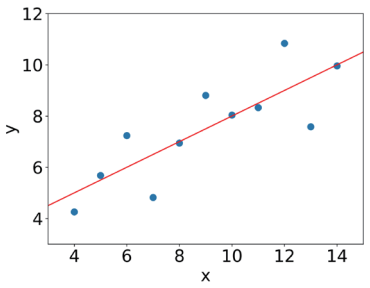
1日あたりの患者数が (371, 266, 350, 378, 445, 353, 327, 272, 178, 275) だったとします。この分布は、図のように平均321.5のまわりにばらついています。ばらつきの大きさは、平均との差を２乗して平均することによって数値化でき、その平方根 σ (シグマ) を標準偏差と言います。



データ間の関係を数値化する

気温と花粉の飛散量のようにふたつのデータの間に関係があることがあります。たとえばxとyの値 (x,y) が (10, 8.04), (8, 6.95), (13, 7,58), (9, 8.81), (11, 8.33), (14, 9.96), (6, 7.24), (4, 4.26), (12, 10.84), (7, 4.82), (5, 5.68) のとき、直線y=ax+bとデータ点の差の2乗の和が最小になるようにaとbを決めることができます。

この方法を最小二乗法、直線を回帰直線と呼びます。



散布図と回帰直線。青丸のデータはアンスコム (Anscombe) の数値例のひとつ。

計測データとノイズ

実験装置や観測装置を用いて測定したデータには様々なノイズが含まれます。たとえば、IceCubeでは宇宙線と地球大気の相互作用によって発生する大気ニュートリノが雑音源です。これらのノイズを除去したデータを解析します。

計算機ネットワークを用いたアラートシステム

身近な警報システムとしては、緊急地震速報があります。海底などに設置した地震計のデータを、計算機ネットワークを用いて集約し、地震の発生直後に強い揺れの到達時刻や震度を予測します。これらを地震による強い揺れが始まる前にテレビや携帯電話を通して通報します。

宇宙では様々な爆発現象が発生します。現在では、重力波、ニュートリノ、電波、可視光、X線、ガンマ線などの望遠鏡が連携して超新星爆発などの観測が行われています。IceCubeが捉えた宇宙ニュートリノの観測データは、計算機で即時解析され、その発生源の位置等が通報されます。

南極で宇宙ニュートリノをキャッチ！

2017年9月23日（日本時間）早朝、携帯電話に通報があり、さまざまな波長での観測がスタートしました。その結果、ニュートリノの発生源が活動銀河中心核であることが明らかになり、放射源が同定された最初の例になりました。

アラートが出されると、世界中の望遠鏡がその方向に向けられて観測が始まるよ

第1部
04

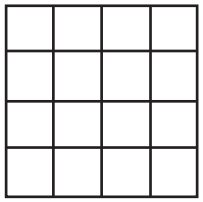
数値シミュレーション

様々な要因が複雑に絡みあって生じる現象を理解する際、数値シミュレーションが強力な手段になります。現実世界をモデル化して計算することにより、物質の性質や天体現象を計算機の中に再現して実験することができます。

数値シミュレーションの方法

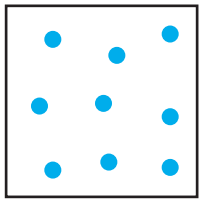
1 連続モデル

空間を碁盤目状の格子に分割し、それぞれのセル内の圧力、温度、速度などの時間変化を計算する方法です。天気予報、車や飛行機のまわりの空気の流れ、宇宙流体の計算などに用いられます。



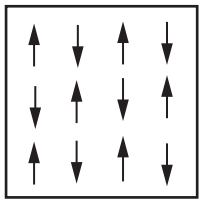
2 粒子モデル

粒子間に働く力を計算し、粒子の運動を追跡する方法です。銀河系内の星の運動や分子の運動、電荷を持った粒子集団（プラズマ）の運動の計算に用いられます。



3 確率モデル

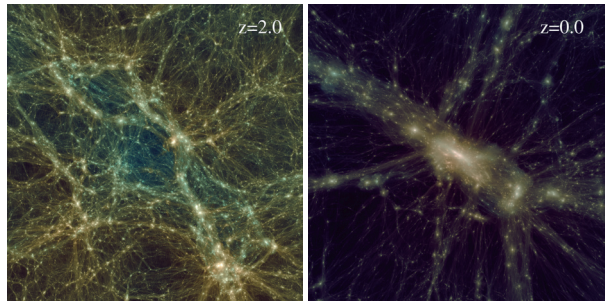
モンテカルロ法とも呼ばれ、ある確率で発生する現象を、でたらめ（ランダム）に生成される乱数を用いて計算します。たとえば、鉄原子は上向き、あるいは下向きのスピンを持ち、温度が高いときにはその方向はランダムに決まりますが、温度を下げていくと周囲の鉄原子と同じ方向を向く確率が高くなります。その結果、ある温度以下では多数の鉄原子のスピンが同じ方向を向くようになります。これが磁石です。



数値シミュレーションの例

ダークマターによる宇宙の大規模構造形成

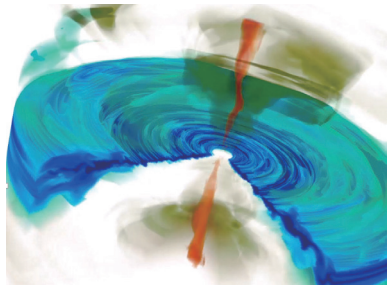
宇宙の物質のうち80%以上は見えない物質ダークマターです。ビッグバンで始まった膨張する宇宙の中で、ダークマターが互いの重力によって引き合って構造を作る過程を粒子シミュレーションによって計算した結果を右図に示します。左が100億年前、右が現在の姿です。銀河団サイズの構造が形成されています。



宇宙構造形成シミュレーション結果（石山智明氏提供）。一辺の長さは1億光年。

ブラックホール降着円盤とジェット

ブラックホールに回転しながら落下する物質によってできる円盤は降着円盤と呼ばれ、活動銀河中心核等のエネルギー源と考えられています。右図は連続モデルに基づく3次元シミュレーション結果で、中心に存在するブラックホールの重力に引かれてゆっくりと落下する回転円盤上下にジェットが噴出しています。



ブラックホール降着円盤のシミュレーション結果（五十嵐太一氏提供）

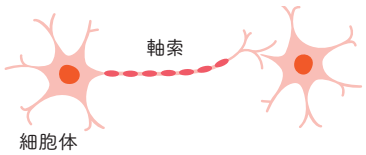
計算機の中で宇宙を創って爆発させる…なんだかSFみたい

第1部
05

大規模データを用いた学習

画像認識等の分野で、人間の脳神経細胞の働きを模倣したニューラルネットワークが広く利用されるようになってきました。とくに、深層学習（ディープラーニング）は、アルファ碁などにも適用され、人間に打ち勝つには数十年かかると言われていた碁の世界でも人間を超えました。

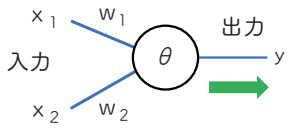
ニューロンの仕組み

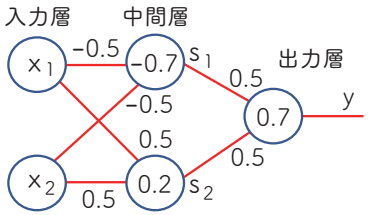


人間や生物の脳神経細胞（ニューロン）は、右図のように細胞体から細長い軸索が伸びて、他のニューロンと結合しています。この軸索内を電気パルスが伝わり、情報を伝達します。あるニューロンに伝わってきた電気パルスが閾値を超えるとこのニューロンが発火して、他のニューロンに電気パルスを送り出します。

多層ニューラルネットワーク

このようなニューロンの働きをモデル化したのが右図です。入力をx_1, x_2, 結合の強さ（重み）をw_1, w_2として$w_1x_1+w_2x_2 > \theta$なら1が出力され、そうでなければ0が出力されるとします。右図のように3層のニューロン間の重みとθを設定すると、入力（x_1, x_2）が(0,0), (1,1)のとき、yには0、(0,1), (1,0)のときに1が出力されます。





ディープラーニング

ニューロン間の結合の強さ（重みw）と、閾値θを調節することにより、ニューラルネットワークが入力に対して適切な値を出力するように教育することができます。多層ニューラルネットワークを用いた、このような学習のことを深層学習（ディープラーニング）と言います。

深層学習は、2012年にカナダのヒントン教授らのチームが、画像認識のコンテストで圧勝したことを契機にして広く利用されるようになりました。Google社のグループは、多数の猫の画像データを学習させることにより、猫の画像を見せると「ネコ」と答えてくれるニューラルネットワークを作成しました。ゴッホやミュシャなどの画風を学習させることにより、下図のように画風を変換することもできます。

ニューラルネットワークを用いた画風変換。
左:原図、右:ミュシャ風
https://tech.preferred.jp/ja/blog/chainer-gogh/

ディープラーニングの活用事例

手書き文字を認識するニューラルネットワークの学習に用いられるデータの例を右図に示します。深層学習は入国審査の際の顔認識にも利用されています。CTやMRIによる画像データを用いたがん検診、車の自動運転やお片付けロボットなどにも利用されつつあります。

手書き文字データMNISTの例
http://yann.lecun.com/exdb/mnist/

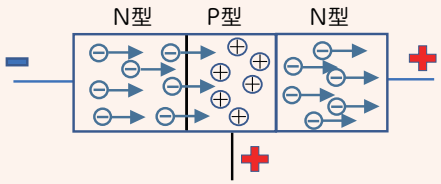
学習用データと正解データを使ってお勉強

豆知識

計算機の発達

計算のための道具は、そろばんや歯車式の計算機から電動式の計算機、電子の流れを制御できる素子を用いた電子計算機へと発達してきました。

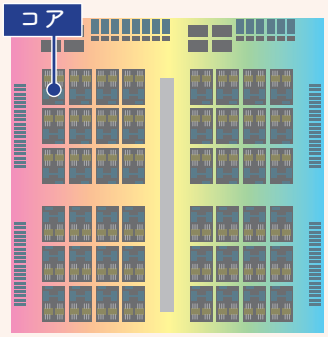
トランジスタの原理



図はシリコン結晶に不純物を加えてマイナス電荷を持つ電子を過剰にしたN型半導体とマイナス電荷が不足したP型半導体を組み合わせた半導体素子の例(トランジスタ)です。中央の領域に加える電圧をプラスにすると電流が流れ、マイナスにすると電流が流れなくなります。

1970年代には、電子計算機に与える命令を解釈して実行したり、計算を行ったりする中央処理装置(CPU)を数センチのシリコン基板上に作成したマイクロプロセッサが開発され、パーソナルコンピュータ(PC)が登場します。

メニーコアプロセッサ

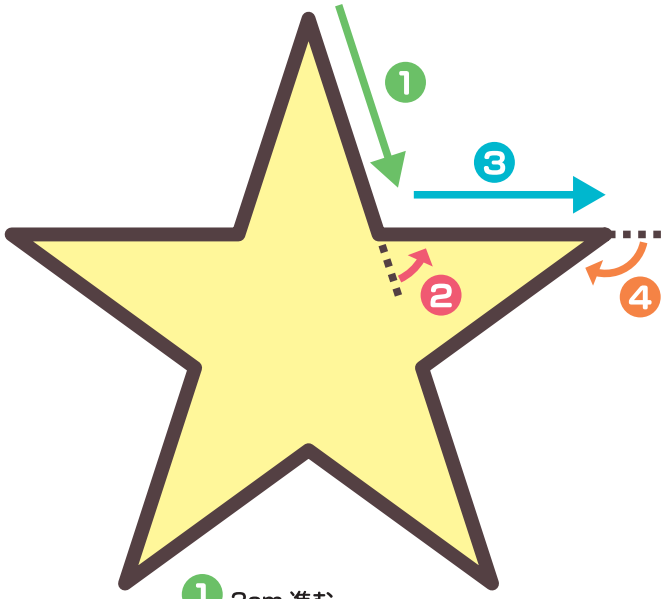


微細加工技術の進歩にともなって、マイクロプロセッサに搭載できる半導体素子の数が2年に2倍のペースで増え、演算速度も速くなりました。しかしながら、近年、速度は上昇しなくなり、演算装置(演算コア)を多数搭載したプロセッサが用いられるようになりました。

日本のフラグシップ計算機「富岳」では48個の演算コアを持つプロセッサが16万個使用され、これらが同時に命令を処理することにより、1秒間に50京回の計算が可能です。

第2部

プログラミング



この星を描くには

1 3cm 進む
2 反時計回りに 72°ターン
3 3cm 進む
4 時計回りに 144°ターン

を 5 回繰り返す

プログラムは問題を解く手順を
計算機向きの言語で表現したもの、
運動会のプログラムも手順書という意味だよ！

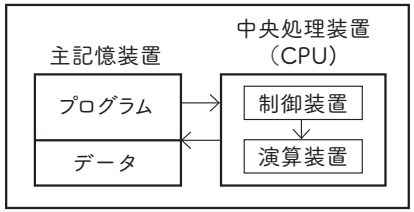
第2部
01

アルゴリズムとプログラム

問題を解く手順のことをアルゴリズムと言います。その語源は方程式の解き方などをまとめたアラビアの数学者アル＝フワーリズミーの名前です。アルゴリズムを計算機向きの言語で表現したものがプログラムです。電卓を用いた計算では人間が手順を記憶していますが、電子計算機では手順をプログラムとして与えます。

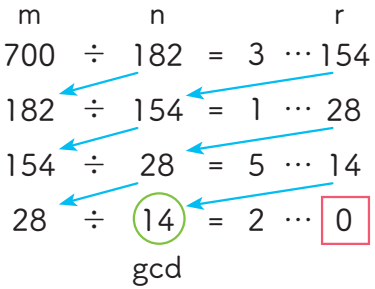
プログラム内蔵方式の計算機

大多数の計算機では、計算機が実行できる命令に変換されたプログラムが主記憶装置に読み込まれ、この命令が順次、中央処理装置（CPU）に取り出されて実行されます。



ユークリッドの互除法のアルゴリズム

有名なアルゴリズムに、ユークリッドの互除法があります。これは、ふたつの整数の最大公約数を求める手順です。ふたつの整数を割り算し、余りを求めます。次に除数を被除数に、余りを除数にして割り算を行い、余りを求めます。余りが0になったときの除数が最大公約数です。例えば、700と182の最大公約数をこの手順で求めると14になります。



余りが0になったときの除数が最大公約数gcd

アルゴリズムの基本パターン

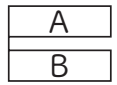
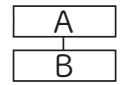
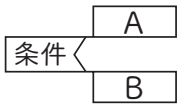
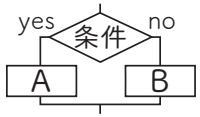
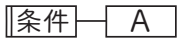
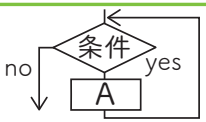
すべてのアルゴリズムは以下の３つの基本パターンの組み合わせで記述することができます。

① 連接：処理Ａをして、次に処理Ｂをする
② 選択：条件が満たされていれば処理Ａ
　　満たされていなければ処理Ｂをする
③ 反復：条件が満たされている間、処理Ａを繰り返す

例えばユークリッドの互除法は、「余りが０でない間、割り算をして余りを求める操作を繰り返す」と記述できます。

アルゴリズムの図式表現

アルゴリズムを表現するために流れ図が用いられます。しかし、流れ図では上記の３つの基本パターンが明確に区別できていません。そこで、連接、選択、反復に異なる記号を用いる図式が考案されました。一例として、PAD（Problem Analysis Diagram）を紹介します。

	PAD	流れ図
連接		
選択		
反復		

アルゴリズムを整理しておかないと、スパゲッティみたいに絡まったプログラムになっちゃうよ

第2部

02

プログラミング言語

計算機向きのプログラムを記述するために使われるのがプログラミング言語です。プログラミング言語にはCPUに備わった命令を2進数で表現した機械語、アルファベットで表現したアセンブリ言語などの低水準言語と、より抽象度が高く、数式なども扱うことができる高水準言語があります。

プログラミング言語の変遷

1950年代に式を扱うことができる高水準言語であるFORTRANが開発され、現在でも科学技術計算用言語として用いられています。1970年頃にはアルゴリズムの3基本パターンを明解に記述することができるPASCALやC言語が開発されます。その後、オブジェクト指向という考えを取り入れたC++, JAVAなどが開発されます。データサイエンス、特に機械学習にはPythonが広く使われるようになりました。

Python言語のプログラム

Python言語ではアルゴリズムの３基本パターンを右図のように記述します。

	PAD	Python
連接	A B	A B
選択	条件 ＜ A / B	if 条件： 　A else： 　B
反復	条件 ― A	while 条件： 　A

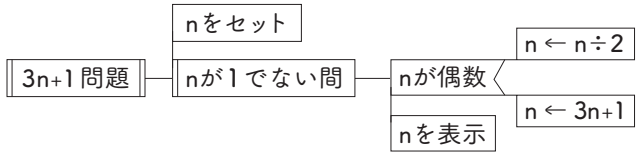
Python言語のプログラムを入力・実行する方法

例として、Google Colaboratory (Colab)　を紹介します。

1. https://colab.research.google.com/ に接続。
2. 「ノートブックを新規作成」をクリック。
3. 「＋コード」をクリックしてプログラムを入力する。
4. ▶ ボタンをクリックするか、Ctrlキーと Enter キーを同時に押すとプログラムが解釈されて実行される。
5. 間違いがあればプログラムを修正して再実行。
6. 「ファイル」から「保存」を選んでnotebookを保存。

例題：3n+1問題

ある整数が偶数なら2で割る。奇数なら3倍して１を加える。どの自然数から出発しても最後は１になるか？




```
n = 5
while n != 1:
    if n % 2 == 0:
        n = n//2
    else:
        n = 3*n+1
    print(n)
```

= は右辺の値を左辺の記憶場所に記憶するという「代入操作」を表す。
!= は「等しくない」という関係
== は「等しい」という関係。
%は割り算の余りを計算。
//　は整数どうしの割り算を表す。
5→16→8→4→2→1

いろいろな数から出発して、
ちゃんと１になるか試してみよう

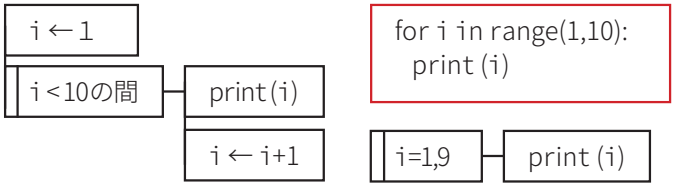
第2部
03

基本プログラミング

アルゴリズムの３基本パターンに加えて、単純な繰り返しを簡潔に記述する方法の紹介です。多数のデータをまとめて扱う方法、データを入力する方法についても説明します。

単純繰り返し：for文

ある回数繰り返すという手順を簡潔に表現できます。
以下はカウンター iの値を1から9まで1ずつ増やしながら表示するアルゴリズムとプログラムです。

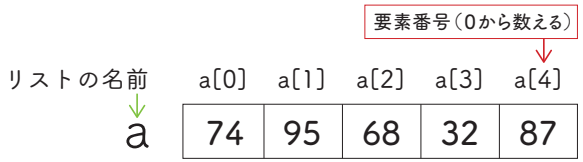



```
for i in range(1,10):
  print (i)
```

i=1,9　print (i)

多数のデータを扱う方法

複数のデータをまとめて扱う場合、「リスト」という表現形式を用いることができます。たとえば5人の学生の成績をa0, a1, a2, a3, a4のように異なる名前を用いて表現するよりも、ひとつのリストで表現した方が便利です。



リストを用いたプログラムの例

最大値を求める

```
a=[74,95,68,32,87]
max=a[0]
for i in range(5):
   if a[i] > max:
      max=a[i]
print('max = ',max)
```

実行結果
max = 95

合計を求める

```
a=[74,95,68,32,87]
s=0
for i in range(5):
   s=s+a[i]
print('sum = ',s)
```

sum = 356

データを入力する

```
a=[0]*3   # a=[0,0,0]
for i in range(3):
   a[i]=int(input('a = '))
print(a)
```

```
a = 1
a = 2
a = 3
[1,2,3]
```

inputという命令を用いて、データを入力することができます。
intは入力された文字列を整数に変換します

フィボナッチ数列

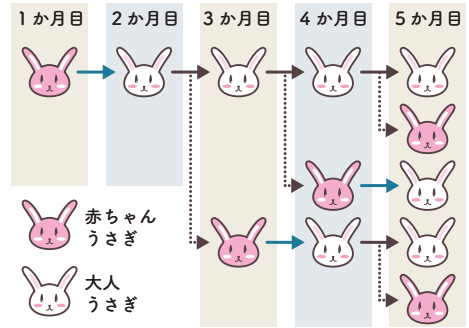
ウサギが1匹生まれました。2か月たつと毎月子供を1匹生むとします。10か月後にウサギは何匹になるでしょう？

nか月後のウサギの数をa[n] とします。a[n] は2か月前のウサギの数の2倍と１か月前に生まれたウサギの数の和ですから
a[n] = 2×a[n-2]+(a[n-1]-a[n-2]) = a[n-1] + a[n-2]。

このような数列をフィボナッチ数列と言います。

以下にプログラムを示します。計算してみましょう。

```
a=[1]*11
for n in range(2,11):
   a[n]=a[n-1]+a[n-2]
print(a)
```



フィボナッチ数は巻貝の形やヒマワリの種の数みたいに、世界のいろいろな場所で見つけることができるよ

第2部
04

モンテカルロシミュレーション

計算機を用いて、でたらめな数（乱数）を発生させ、確率的に起こる現象をシミュレートする方法（モンテカルロ法）を紹介します。三角関数の計算等に用いる数値計算ライブラリや、グラフの表示方法についても説明します。

乱数とシミュレーション

計算機の中でサイコロを振ることに相当するのが乱数です。ある規則に基づいて数を生成しますので、厳密な意味では乱数とは言えず、擬似乱数と言います。たとえば、整数にある数を掛けて、別の整数で割った余りを次の数とします。

5×123 ÷ 64 = 9…39　39×123 ÷ 64 = 74…61
61×123 ÷ 64 =117…15　15×123 ÷ 64 = 28…53
39, 61, 15, 53 …

余りは0から63の範囲の整数になります。これを乱数とみなします。

数値計算ライブラリnumpyを用いた乱数生成

`np.random.rand()`　0.0 以上、1.0 未満の一様乱数を生成
`np.random.randint(a,b)`　a から b-1 の範囲の整数を生成

```
# 1から6までの乱数を5個生成
import numpy as np
n=5
for i in range(n):
  print(np.random.randint(1,7))
実行結果
5
2
6
4
3
```

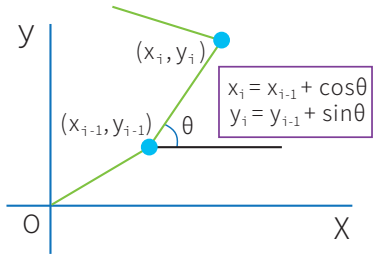
```
# まとめて10個生成して平均を求める
import numpy as np
n=10
saikoro=np.random.randint(1,7,n)
print(saikoro)
print(np.mean(saikoro))
実行結果
[5 6 6 1 2 6 1 1 4 2]
3.4
```

ランダムウォークとグラフ作成の例

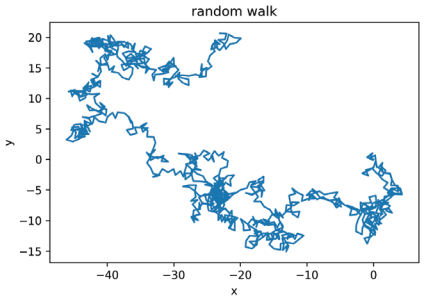
太陽内部で光子は電子によって何回も散乱されるため、なかなか表面に出てくることができません。このような現象のモデルとして、ある距離進むごとに進行方向がランダムに変わる運動を追跡してみます。

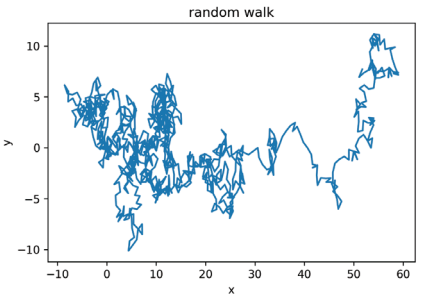
```
import matplotlib.pyplot as plt
import numpy as np
n=1000
x=np.zeros(n)
y=np.zeros(n)
seed=int(input('random seed :'))
np.random.seed(seed)
for i in range(1,n):
   theta=2.0*np.pi*np.random.rand()
   x[i]=x[i-1]+np.cos(theta)
   y[i]=y[i-1]+np.sin(theta)
plt.plot(x,y)
plt.title('random walk')
plt.xlabel('x')
plt.ylabel('y')
```

- 原点 (0,0) から出発してランダムな方向に1mずつ移動。
- 考え方：次の1歩がx軸となす角度θを乱数にする。



このプログラムではグラフ表示のために matplotlib というライブラリを呼び出しています。plt.plot(x,y) により、リスト x,y に記憶された点を線で結んで表示してくれます。





1000歩進んでも、出発点からあまり遠くには行けないんだね

第2部

05

データ解析への応用

インターネットで公開されているデータを読み込んで解析する方法を紹介します。たとえばe-Statというサイトでは国勢調査結果等が公開されます。天体観測データも多くの場合、ある期間が過ぎると一般公開されるため、これらをダウンロードして研究に活用することができます。

データ解析支援ライブラリ

pandas

Pythonからは、大規模データ解析を支援するライブラリpandas が利用できます。pandas は DataFrameというデータ形式を用いて、表形式のデータを扱うことができます。

天体観測データの例

国際宇宙ステーションの日本実験棟「きぼう」船外に取り付けられた全天X線監視装置MAXIによる観測データを例にして、公開データの解析方法を紹介します。

MAXIホームページ（http://maxi.riken.jp/）のBasic dataから、様々な天体のX線時間変動データを下表のような形式でダウンロードすることができます。ここで、MJDは修正ユリウス日、各列はキロ電子ボルト（keV）であらわしたエネルギー範囲の1秒1cm^{2}あたりの光子数の平均値です。

	MJD	2-20keV	2-4keV	4-10keV	10-20kev
0	58187.5	0.111142	0.028501	0.029337	0.040441
1	58188.5	0.232336	0.073869	0.076537	0.060910
2	58189.5	0.351457	0.130996	0.126267	0.074254
3	58190.5	0.427677	0.179219	0.158668	0.069805
4	58191.5	0.886044	0.394579	0.309640	0.153357

ブラックホール候補天体 MAXI J1820+070の観測データ。誤差データ列は省いた。
元データは http://maxi.riken.jp/pubdata/v7.7l/J1820+071/index.html　Binsize = 24.0h のCSVファイル

データ読み込み

観測データが MAXIJ1820.csv というファイルに保存されているとき、次のプログラムで読み込むことができます。

```
import pandas as pd
df = pd.read_csv('./MAXIJ1820.csv')
df.head()
```

import pandas as pd は、pandasを使うためのおまじないです。pd.read_csvは、データを読み込む命令です。読み込んだデータはdfという名前で参照できます。

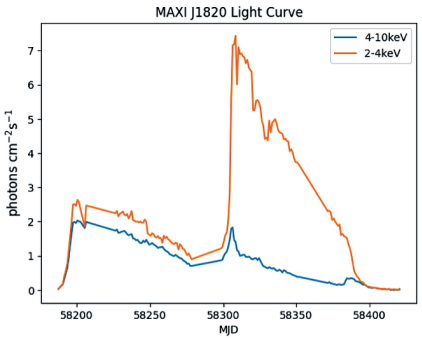
読み込んだデータの各列はdf['4-10keV']のように各列のラベルを用いて取り出せます。

解析結果のグラフ表示

matplotlibを用いて解析結果をグラフ表示できます。

```
import matplotlib.pyplot as plt
plt.plot(df['MJD'],df['4-10keV'],label='4-10keV')
plt.plot(df['MJD'],df['2-4keV'],label='2-4keV')
plt.xlabel('MJD')
plt.ylabel('photons cm$^{-2}$s$^{-1}$',fontsize=12)
plt.title('MAXI J1820 Light Curve')
plt.legend(loc='upper right')
plt.show()
```

このプログラムを実行すると、右図のように、低エネルギー（2-4keV）と高エネルギー（4-10keV）の光子数の時間変化が表示されます。



既に公開されているデータの解析からでも、
新しい発見があるかもしれないよ！

応用課題

ハビタブル惑星

応用例としてGSCアカデミックセミナー企画「観測データとシミュレーションから生存可能天体について考える」の課題とした、太陽系外惑星の観測データ解析を取り上げます。

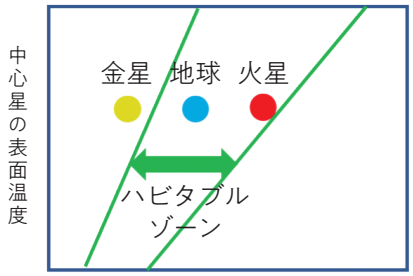
太陽系外惑星の発見

太陽以外の恒星のまわりを公転する惑星を発見することは天文学者の長年の夢でした。Mayor博士とQueloz博士は、1995年にペガスス座51番星の周りを4.2日の周期で公転する、木星の0.45倍の質量を持つ惑星（51 Peg b）を発見し、2019年ノーベル物理学賞を受賞しました。

ハビタブル惑星

太陽系外惑星の発見方法のひとつに、惑星が中心星の前を通過する際の中心星の光度変化を検出する「トランジット法」があります。2009年にNASAによってケプラー衛星が打ち上げられ、トランジット法によって多数の太陽系外惑星が発見されました。この中には、惑星表面の温度が0℃と100℃の間にあって水が液体として存在できる生存可能惑星（ハビタブル惑星）があります。

惑星の表面温度は、中心星の半径、表面温度、惑星までの距離から決まります。右図に太陽系内の惑星の例を示します。矢印が、水が液体として存在できる範囲を示します。

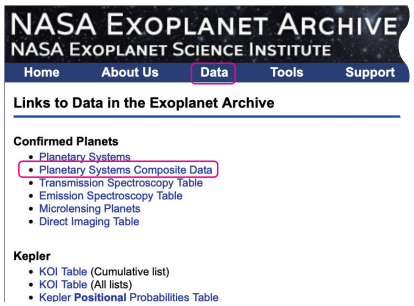


太陽系外惑星の観測データ

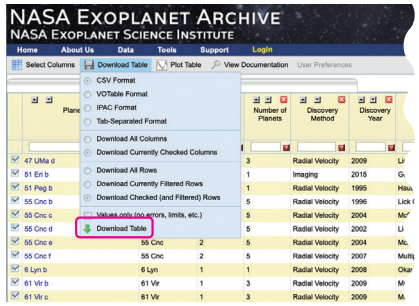
これまでに発見された太陽系外惑星のデータが以下のNASA/JPL-CaltechのWebページにまとめられています。

https://exoplanetarchive.ipac.caltech.edu/index.html

以下の手順でデータをダウンロードすることができます



メニューからDataを選び、Planetary Systems Composite Dataを選ぶと、惑星一覧が表示されます。



メニューの下にある Download Tableから下矢印をクリックするとデータファイルがダウンロードされます。

pandas を用いてこのデータファイルを読み込んで、以下の課題に取り組んでみましょう。

課題：ハビタブル惑星をみつける

① 太陽系外惑星のデータを読み込み、横軸に中心星からの距離、縦軸に中心星の表面温度をとったグラフ上に各惑星をプロットする。
② この中で、ハビタブルゾーンにある惑星を表示する。
③ 地球に最も近い恒星プロキシマ・ケンタウリの惑星プロキシマｂがハビタブルゾーン内にあることを確認する。

応用課題解説

1. 太陽系外惑星データの読み込みと描画

ダウンロードした惑星データファイルの名前を PS.csvに変更しておきます。

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv('PS.csv',comment='#')
df.plot.scatter(x='pl_orbsmax',y='st_teff',
            xlim=[0,1.5],ylim=[2000,8000])
plt.xlabel("Semi Major Axis [au]")
plt.ylabel("Surface Temperature [K]")
plt.show()
```

データファイル中、ラベル pl_orbsmax の列が惑星の軌道長半径（単位は地球と太陽の距離 au）、st_teff の列が中心星の表面温度（単位は絶対温度 K）です。軌道長半径をｘ軸、中心星の表面温度をｙ軸として ｘ＜1.5au、2000K ＜ｙ＜ 8000K の範囲にある惑星を表示します。

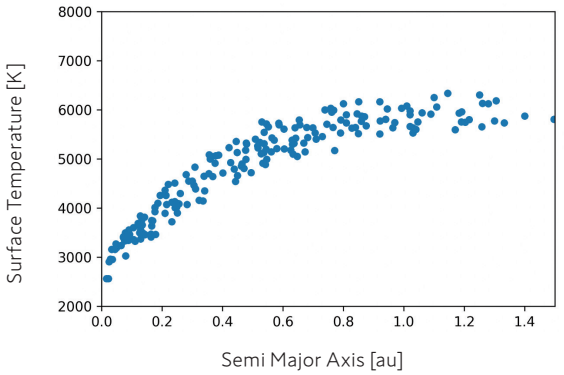
2. 惑星の表面温度の計算

惑星の表面温度T_pは、惑星が中心星から受けとるエネルギー（中心星の表面温度T、中心星の半径R、惑星までの距離ｄ に依存）と惑星表面から放射されるエネルギーを等しいとおくことによって求まります。中心星の半径を太陽半径$R_\odot$を単位とし、惑星までの距離ｄをau単位とすると近似的に $T_p=0.05\ (R/d)^{1/2}\ T$ になります。惑星の大気組成や表面の反射率などは、ここでは考慮していません。

3. ハビタブルゾーンにある惑星を表示する

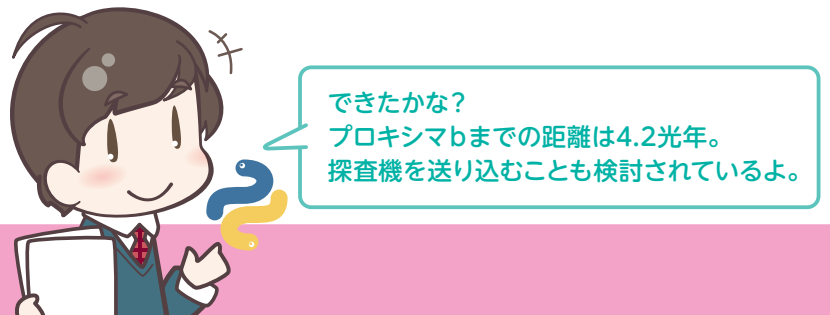
プログラム中、df.plot.scatter の前に以下を挿入します。惑星の表面温度（pl_eqt）を求め、273K ＜ pl_eqt ＜ 373Kの惑星だけを選んでいます。実行結果も示します。

```
df['pl_eqt']=0.05*np.sqrt(df['st_rad']
                  /df['pl_orbsmax'])*df['st_teff']
df=df[(df['pl_eqt'] > 273) & (df['pl_eqt'] < 373)]
```



4. プロキシマbがハビタブルゾーン内にあることを確認

プロキシマ・ケンタウリは地球から最も近い恒星でケンタウルス座α星と 連星をなす赤色矮星で、表面温度は3042K です。2016年にプロキシマから0.05auの距離を公転する地球サイズの惑星プロキシマbが発見されました。プロキシマｂをグラフ上にプロットするとハビタブルゾーン内にあることがわかります。



用語集

データサイエンス

データを用いて科学的あるいは社会的に有用な知見を得ようとする科学のこと。計算機ネットワークの発達によって集積されるデータ量が膨大になり、これらを活用する新たな科学的アプローチが必要になっている。

Society 5.0

狩猟社会、農耕社会、工業社会、情報社会に続く社会で、計算機ネットワークを通してヒトとモノがつながり、現実空間とサイバー空間が融合して人間とロボットが共存していく社会。

ディープラーニング（深層学習）

人間の脳神経細胞の働きを模した多層ニューラルネットワークを用いた学習。手書き文字などの入力に応じて正しい出力が得られるように、ニューロン間の結合の強さ等を調整する。膨大な学習データを用いて学習することにより、画像認識では人間の能力を超え、ロボットが眼を獲得した。この分野を先導したカナダのJ.Hinton教授は、2018年に情報科学分野のノーベル賞ともいえるチューリング賞を受賞、2024年にノーベル物理学賞を受賞した。

人工知能（Artificial Intelligence: AI）

人間特有の能力と考えられていたものが次々と計算機で実現され、当初は人工知能と言われた能力もすぐに陳腐な技術になってきた。機械学習と呼ばれることも多い。最近、画像や文章を生成する生成系AIが広く利用されるようになっている。

ニュートリノ

電気を帯びていない軽い素粒子で、物質とほとんど反応せず、地球を通り抜けることもできる。これを捉えるには大量の水や氷を必要とする。

ダークマター

重力以外の相互作用がほとんどなく、光らない暗黒物質。宇宙の物質の80%以上を占める。ダークマターが重力によって集積することによって宇宙の大規模構造や銀河団が形成される。

さらに勉強するには

データサイエンスについて

データサイエンス入門第3版　竹村彰通他：編、学術図書出版社
データサイエンスの基礎　濱田悦生：著、狩野 裕 編、講談社

アルゴリズム

Pythonで学ぶアルゴリズムとデータ構造　辻 真吾：著、下平英寿：編、講談社
アルゴリズムとデータ構造　大槻兼資：著、秋葉拓哉：監修、講談社
問題解決のための「アルゴリズム×数学」が基礎からしっかり身につく本
米田優峻：著、技術評論社

Python言語

みんなのPython　柴田 淳：著、Soft Bank Creative

情報処理技術者試験

https://www.ipa.go.jp/shiken/

情報オリンピック

https://www.ioi-jp.org/

情報処理推進機構　未踏事業

https://www.ipa.go.jp/jinzai/mitou/portal_index.html

千葉大学ハドロン宇宙国際研究センター

http://www.icehap.chiba-u.jp/

科学技術振興機構次世代科学技術チャレンジプログラム
未来の世界を創造する研究人財育成
千葉大学STELLAプログラム ASCENT-6E
https://stella.e.chiba-u.jp/
2025年3月25日発行
松元亮治：著　構成／装丁：荒木未来　イラスト：秋本祐希（ひっぐすたん）

```
a=[1,1]
for i in range(10):
 a.append(a[-1]+a[-2])
rint(a)
```

```
import matplotlib.pyplot as plt
import numpy as np
n=1000
x=np.zeros(n)
y=np.zeros(n)
seed=int(input('random seed :' ))
np.random.seed(seed)
for i in range(1,n):
 theta=2.0*np.pi*np.random.rand()
 x[i]=x[i-1]+np.cos(theta)
 y[i]=y[i-1]+np.sin(theta)
plt.plot(x,y)
plt.title('random walk')
plt.xlabel('x')
plt.ylabel('y')
```

```
import numpy as np
n=10
saikoro=np.random.randint(1,7,n)
print(saikoro)
print(np.mean(saikoro))
```

```
0   58187.5   0.111142   0.028501   0.029337   0.040441
1   58188.5   0.232336   0.073869   0.076537   0.060910
2   58189.5   0.351457   0.130996   0.126267   0.074254
3   58190.5   0.427677   0.179219   0.158668   0.069805
    58191.5   0.886044   0.394579   0.309640   0.153357
```