

機器分析演習

1. e の値を求める
2. π の値を求める
3. 一次反応のシミュレーション
4. 二次反応のシミュレーション
5. 逐次反応のシミュレーション
6. ウサギとキツネ
7. ニュートン法
8. 線形最小二乗法
9. 非線形最小二乗法
10. 台形公式による数値積分
11. モンテカルロ法による数値積分

級数の和を求める

1. e の値を求める

e^x のテーラー展開 (マクローリン展開) は

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

なので、 $(0! = 1)$

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \dots$$

【課題】

この級数がどの程度の速度で収束するのか、計算する。収束の様子をグラフ化する。これ以後いちいち書かないが、できる限り課題の結果はグラフ化すること。グラフ化しないでシミュレーションをするのは非常に危険である。

- ・ 連続データの作成
- ・ 関数の利用
- ・ セルの間の計算
- ・ 計算式のコピー
- ・ 項を順に足していくこと
- ・ 漸化式・シミュレーション
- ・ グラフによる可視化

2. π の値を求める

π を計算するには Arcsin 関数 (\sin 関数の逆関数) か、 Arctan 関数 (\tan 関数の逆関数) を利用する。 Arcsin 関数は $|x| < 1$ で

$$\text{Arcsin}(x) = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1}$$

のように、 Arctan 関数は $|x| \leq 1$ で

$$\text{Arctan}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}$$

テーラー展開 (マクローリン展開) される。まず、 \arcsin 関数を利用して π を計算してみよう。

$$\sin\left(\frac{\pi}{6}\right) = \frac{1}{2}$$

だから、

$$\text{Arcsin}\left(\frac{1}{2}\right) = \frac{\pi}{6}$$

なので、

$$\sum_{n=0}^{\infty} \frac{(2n)! \times 0.5^{2n+1}}{4^n (n!)^2 (2n+1)} = \frac{\pi}{6}$$

ここから π を計算してみる。 $(\sin(\pi/2) = 1$ からでも計算できるが、収束は非常に遅い)

一方、

$$\tan\left(\frac{\pi}{4}\right) = 1$$

だから、

$$\text{Arctan}(1) = \frac{\pi}{4}$$

なので、

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

ここから π を計算してみる。Arctan からの計算は収束が遅いが、計算が簡単なのでより正確に計算できる。

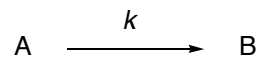
【課題】

それぞれの級数がどの程度の速度で収束するのか比較する。

微分方程式を差分方程式に置き換えてシミュレーションする

3. 一次反応のシミュレーション

反応速度とは、ある物質の濃度が変化する速度である。これは通常微分方程式の形で書かれる。例えば、



という反応（反応速度定数 k ）があると、反応速度とは、 A の濃度 $[A]$ の変化する（減少する）速度であって、

$$-\frac{d}{dt}[A]$$

である。反応が一次反応（ A の一分子だけで起こる反応）だとすると、 $[A]$ の減少速度は $[A]$ に比例するので

$$-\frac{d[A]}{dt} = k \cdot [A]$$

である。これを積分すると、 $[A]$ の初期値（ $t=0$ の時の $[A]$ ）を $[A]_0$ とすると

$$\ln\left(\frac{[A]}{[A]_0}\right) = -k \cdot t$$

つまり

$$[A] = [A]_0 \cdot \exp(-k \cdot t) \quad \text{ただし、} \exp(x) = e^x$$

となるのであるが、これを数値計算でやってみよう。微分というのは

$$-\frac{d[A]}{dt} = \lim_{\Delta t \rightarrow 0} \left(-\frac{\Delta[A]}{\Delta t} \right)$$

つまり、微小時間 (Δt) 当たりの濃度 $[A]$ の変化量 ($\Delta[A]$) を意味する。その「微小時間」が十分短ければ、その時間の中での $[A]$ の変化量は微々たるものだから、初期濃度 $[A]_0$ から、「微小時間後の A の濃度」を計算することができる。それを繰り返せば、 $[A]$ の変化する様子をシミュレーションできる。この作業は、 e や π の計算で、級数のそれぞれの項を足していったのと同様である。

具体的には、反応速度式を

$$-\frac{\Delta[A]}{\Delta t} = k \cdot [A]$$

と置き換えて、これから $[A]$ の値を次々と計算して行く。このような方程式を差分方程式という。微分方程式は、 $d \rightarrow \Delta$ として、差分方程式に置き換えられる。

【課題】

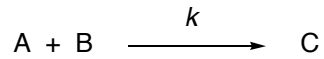
データセットで与えられた初期濃度 $[A]_0$ (任意単位) と反応速度定数 k (任意単位) を用いて、一次反応をシミュレーションする。時間 t における $[A]$ の値は解析的に求めることができるから、シミュレーションした値と比較する。

反応の半減期 $\tau_{1/2}$ ($[A]$ が $[A]_0$ の $1/2$ 、 $1/4$ 、 $1/8 \dots$ となる時間) をシミュレーションで求める。解析的には、 $\tau_{1/2} = \frac{1}{k} \cdot \ln 2$ となるはずであるから、微小時間 Δt (単位 s) をいろいろ設定して、 Δt によって誤差がどのように変わるか調べる。 Δt が小さい方が誤差は小さくなるが、その分、大規模なシートを作らなければならない。

半減期が一定であるということは「反応はいつまでも終わらない」ことを意味する。

- ・ 相対参照と絶対参照

4. 二次反応のシミュレーション



という反応があったとすると、反応速度は、

$$-\frac{d}{dt}[A] \quad [A] \text{の減少（消費）速度}$$

あるいは

$$-\frac{d}{dt}[B] \quad [B] \text{の減少（消費）速度}$$

あるいは

$$\frac{d}{dt}[C] \quad [C] \text{の増加（生成）速度}$$

で表わされる。結果が一致するとは限らないが、どれで表わしても構わない。反応が二次反応（A と B が衝突することによって起こる二分子反応）だとすると、例えば次のように反応速度を表わすことができる。

$$-\frac{d[A]}{dt} = k \cdot [A] \cdot [B]$$

もし $[A]_0 = [B]_0$ であるならば、 $[A] = [B]$ であるから反応速度式は

$$-\frac{d[A]}{dt} = k \cdot [A]^2$$

となつて、簡単に積分ができ、

$$\frac{1}{[A]} - \frac{1}{[A]_0} = k \cdot t$$

となるが、 $[A]_0 \neq [B]_0$ であると

$$\ln\left(\frac{[A]_0 \cdot [A] + [B]_0 - [A]_0}{[B]_0 [A]}\right) = ([B]_0 - [A]_0) \cdot k \cdot t$$

となり、解析は容易ではない。実験的には、解析を可能にするために $[B]_0 \gg [A]_0$ のように条件を設定する（擬一次条件）。

しかし、数値計算なら簡単である。反応速度式を差分方程式に書き直せば、

$$-\frac{\Delta[A]}{\Delta t} = k \cdot [A] \cdot [B]$$

B の濃度に関するパラメータが増えるだけで、一次反応と同様にシミュレーションを行なうことができる。

【課題】

データセットで与えられた初期濃度 $[A]_0$ および $[B]_0$ （任意単位）と反応速度定数 k （任意単位）、を用いて、 $[A]$ および $[B]$ の変化を追跡しよう。

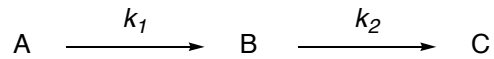
半減期を求めてリストにする。一次反応と異なり、半減期は一定とならない。これは何を意味するか？

データセットには、二次反応のデータ例がある。この濃度変化のデータに合うような k の値を探すことで、反応速度定数を決定してみよう。まず適当な k を与えてシミュレーションを行う。次に、 k の値を変えていって、データによく一致するような k の値を探す。必要な濃度だけが見えるようにシートを作っておかなければならない。

・ VLOOKUP 関数によるデータの参照

5. 逐次反応のシミュレーション

反応が複雑であっても、素反応が明らかであれば反応速度式を書くことはできる。例えば、



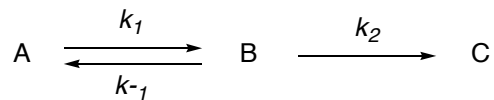
であれば、

$$\frac{d[A]}{dt} = -k_1 \cdot [A]$$

$$\frac{d[B]}{dt} = k_1 \cdot [A] - k_2 \cdot [B]$$

$$\frac{d[C]}{dt} = k_2 \cdot [B]$$

だし、



であれば、

$$\frac{d[A]}{dt} = -k_1 \cdot [A] + k_{-1} \cdot [B]$$

$$\frac{d[B]}{dt} = k_1 \cdot [A] - k_{-1} \cdot [B] - k_2 \cdot [B]$$

$$\frac{d[C]}{dt} = k_2 \cdot [B]$$

となる。このように複雑な場合、反応速度式（連立微分方程式）を解析的に解くことはできないかもしれない。しかし、差分方程式に書き直せば、これまでと同様に容易にシミュレーションできる。

【課題】

データセットで与えられた逐次反応用の初期濃度 $[A]_0$ (任意単位) と反応速度定数 k_1 および k_2 (任意単位)、を用いて、 $[A]$ 、 $[B]$ および $[C]$ の変化を追跡しよう。

$[B]$ はいったん増加した後減少していくことを確認する。

反応によって分子数が増加していないので、 $[A] + [B] + [C]$ は反応の最中に変化しないはずである。このことを確かめる。

適当な k_{-1} を設定して、 A と B の間に平衡が存在する時の反応のシミュレーションも行ない、比較してみよう。

6. ウサギとキツネ

「微分方程式→差分方程式」に基づくシミュレーションは反応だけに適用されるものではない。例えばフィードバック制御は温度調節や方向制御を行なうための基本制御であるが、目標値からのズレに基づく微分方程式の形で制御される。多くの場合この微分方程式を解析的に解くことはできないが、差分方程式としてシミュレーションすることは容易である。ここでは、ウサギとキツネの例をとってフィードバック制御の簡単な例をシミュレーションしてみよう。

次のような状況を考える。

- (1) ウサギ (A 匹) とキツネ (B 匹) が棲んでいる島があり、外界から途絶されている。
- (2) キツネはウサギを食べて生きている。
- (3) ウサギは一定の割合 (p_1) で子どもを産み、増えていく。
→ウサギの増加速度は $p_1 \cdot A$
- (4) ウサギが死ぬのはキツネに食べられた時だけであるとする。キツネは一定の頻度 (p_{-1}) でウサギを食べ、ウサギはその数を減らす。
→ウサギの減少速度は $p_{-1} \cdot A \cdot B$
- (5) キツネもある割合 (r_1) で子どもを産み、増えていく。ウサギがたくさんいると子どももたくさん産まれる。
→キツネの増加速度は $r_1 \cdot A \cdot B$
- (6) キツネは一定の割合 (r_{-1}) で死亡する。
→キツネの減少速度は $r_{-1} \cdot B$

したがって、次のような差分方程式が書ける。

$$\frac{\Delta[A]}{\Delta t} = p_1 \cdot A - p_{-1} \cdot A \cdot B$$

$$\frac{\Delta[B]}{\Delta t} = r_1 \cdot A \cdot B - r_{-1} \cdot B$$

【課題】

キツネもウサギも絶滅しないようなパラメータを探してみよう。ウサギあるい

はキツネの数が 0 以下になるのは、絶滅を意味する。ウサギとキツネの数は典型的なフィードバック制御の様子を示しながら変化し、安定な状態を保つことは非常に難しい（たぶん不可能）。

試しにキツネの数の初期値を 0 にしてみよう。このシミュレーションでのウサギの数の振る舞いは、捕食者のいない人類が今後地球の中でどのような状況になるかを示している。

方程式の数値解を求める

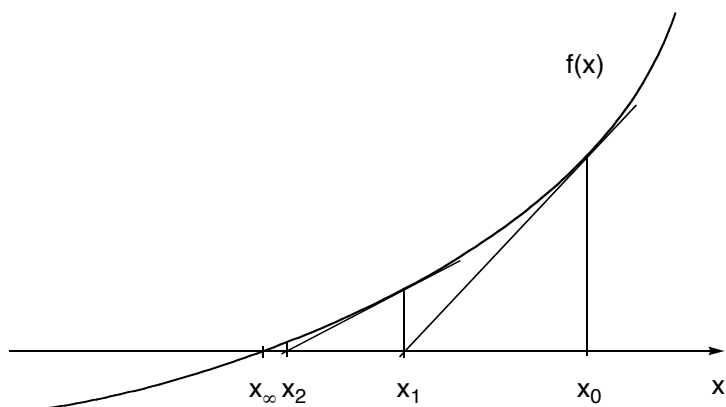
7. ニュートン法

例えば $\cos(x) + x = 0$ という方程式の解を解析的に求めることはできない。しかし、

$$f(x) = \cos(x) + x$$

のグラフを書いてみると（書いてみよう）、解を持つことが分かる。このように解析的に解を求められない場合でも、数値計算をすれば解を求めることができる。

ある関数 $f(x)$ があって、 $f(x) = 0$ を満たすような解 x_{∞} を求めたいとする。 x_{∞} の近傍の x_0 を選び、 $(x_0, f(x_0))$ から接線を引き、 x 軸との交点を x_1 とすると、 $(f(x))$ の凹凸によるが x_1 は x_0 よりも x_{∞} に近いはずである。このことを繰り返して x_{∞} を求める。



$(x_0, f(x_0))$ における接線は、 $y - f(x_0) = f'(x_0) \cdot (x - x_0)$ である。この接線の $y = 0$ の時の値が x_1 となるので、これを解くと

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

になる。以下同様で、一般に

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

となる。

例えば $f(x) = \cos(x) + x$ であれば、 $f'(x) = -\sin(x) + 1$ であるから、

$$x_{n+1} = x_n - \frac{\cos(x_n) + x_n}{-\sin(x_n) + 1}$$

これを順に計算して解が数値的に求められる。 x_0 が x_∞ に十分近くないと解は求まらない (ことがある)。

ゴールシークの利用

2つのセルに x と $f(x)$ を代入する。 x に適当な値 (x_0 に相当する) を代入しておく。ツールからゴールシークを選び、

「数式入力セル」に $f(x)$ を定義したセルを

「目標値」に、この場合ならば 0 を

「変化させるセル」に x を定義したセルを

それぞれ選び、実行する。ニュートン法を利用した解と比較する。

ソルバーを利用した解

ソルバーは複数のセルの値を最適化したり、関数の値を最小にしたり、最大にするパラメータをする時に利用できる。ゴールシークと同様にも利用できるが、この程度の計算をするには高級過ぎる。

関数が複数の解を持つ場合、ニュートン法では初期値に近い解を与えることになる。したがって、まず関数の形を把握しておいてから初期値を適切に設定しなければならない。

【課題】

データセットに与えられた $f(x)$ と $g(x)$ を使い、方程式 $f(x)+g(x)=0$ を解いてみよう。まず x_0 にいくつを代入すればよいかあたりをつけてから解き始めること。

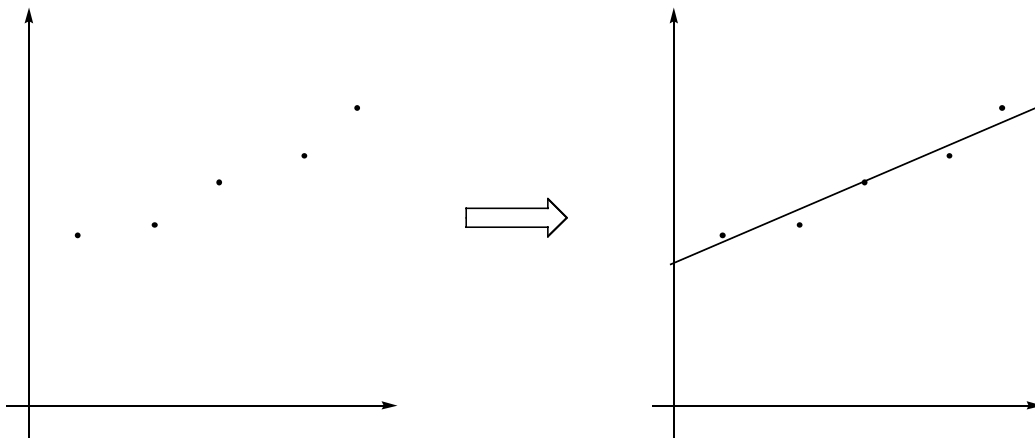
Newton 法は x_0 にどのような値を用いてもよいわけではない（確認せよ）。

ゴールシークを利用して同じ関数を解いてみる。この場合も x_0 に何を選ぶかに気をつけなければならないことは変わらない。

回帰分析

8. 線形最小二乗法

(x, y)のデータの組がいくつか与えられている。それぞれのデータの間にある「関係」が期待される時、「最も確からしい」関数を探すことを回帰分析という。最もよく用いられるのは、その「関係」が直線関係である、すなわち、「最も確からしい」関数が一次関数である、時である。



「最も確からしい」ということを、普通は、データと関係式との差 Δy を用いて、 $\sum (\Delta y)^2$ を最小にすること、と定義する。

そのようにして、データの組から「最も確からしい」直線関係を導くことを線形最小二乗法という。Excel は線形最小二乗法を行なうための関数を持っている。 $y=ax+b$ という関係式に回帰分析をするのであれば

slope	データセットから傾き a を与える関数
intercept	データセットから y 切片 b を与える関数
rsq	データセットの回帰直線からのズレの程度 r^2 を与える関数

という 3 つの関数を利用する。

データは多くの場合「質の悪いデータ」を含む。レベルの高い回帰分析では、データに重みをつけ、反復計算をしながら、回帰直線に近いデータを重視するように分析する。しかし、Excel で重み付きの回帰分析をするのはちょっと面倒である。次善の策は、外れたデータを棄却することである。そのためにも、単に関数を当てはめて回帰直線を求めるだけではなく、必ずグラフを描いて、「質の悪いデータ」を見つけ出さなければならない。

グラフを描くと、グラフメニューが現われる。ここから線形最小二乗法を行なうことができる。「近似曲線の追加」から、線形近似を行なう。回帰直線が現われるので、データとの相関を見て、質の悪いデータを棄却する。

回帰直線は「オプション」からその様子をコントロールする。

- ・ 回帰直線が必ず原点を通る (y 切片は 0) と分かっている時 (検量線など)、切片を指定する。
- ・ 回帰直線はデータのある範囲でしか描かれないので、必要に応じて補外して回帰直線を延長する。
- ・ 回帰直線の式、ズレをグラフに表示させることができる。

【課題】

データセットに与えられたデータに対して回帰直線を引き、その直線の式を求めてみよう。

9. 非線形最小二乗法

データセットの組が与えられていて、そのデータの中に曲線関係が予測される時、その回帰曲線を最小二乗法で求める方法を非線形最小二乗法という。

Excel では、単純な非線形最小二乗法をグラフから使うことができる。

- ・ 対数近似 $y = a \ln x + b$
- ・ 多項式近似 $y = a + bx + cx^2 + dx^3 + \dots$
- ・ 累乗近似 $y = ax^b$
- ・ 指数近似 $y = ae^{bx}$

である。このうち、多項式近似以外はいずれも適当な式の変形で直線関係に置き換えることができる。このような場合、直線関係に置き換えて線形最小二乗法を行なうべきである（その方が精度がよい）。

より複雑な式、より多変数の式へ非線形最小二乗法を行なうためにはソルバーを利用する。

いずれの場合も（特にソルバーを使った時には）、グラフを描き、データセットと回帰直線がよい一致を示していることを確認しなければならない。

一次反応のシミュレーション結果は対数近似されるはずである。自分のシミュレーション結果を対数近似してみよう。一方、二次反応のシミュレーション結果は対数近似では精度が悪いはずである。確かめてみる。

数値積分

10. 台形公式による数値積分

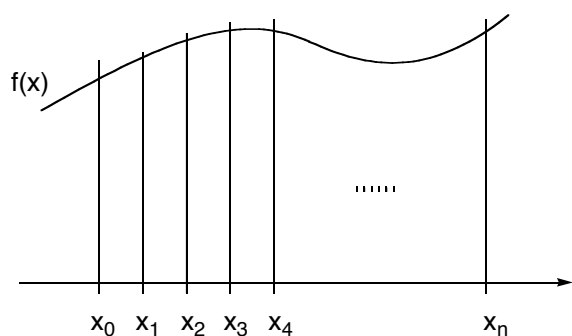
ある関数が与えられていた時、その関数を積分するのはしばしば難しい。しかし、積分とは要するに面積を求めることなのであるから、解析的に積分できなくても、積分値を知ることはできるはずである。

今、関数 $f(x)$ を区間 $[x_0, x_n]$ の間で積分して、 $\int_{x_0}^{x_n} f(x) dx$ を求めたいとする。区間 $[x_0, x_n]$ を n 等分して、それぞれの x の値をそれぞれ x_1, x_2, x_3, \dots とする。等分した幅を Δx とすると、

$$\Delta x = x_1 - x_0 = x_2 - x_1 = x_3 - x_2 = \dots$$

$$x_n - x_0 = n \cdot \Delta x$$

で、積分値＝面積はそれぞれの短冊の面積の合計になる。



Δx が十分に小さければ、それぞれの短冊の形は台形に近いので、面積は簡単に計算できる。すなわち、最初の台形の面積は

$$\frac{(f(x_0) + f(x_1)) \times \Delta x}{2}$$

で、 i 番目の台形の面積は

$$\frac{(f(x_{i-1}) + f(x_i)) \times \Delta x}{2}$$

となる。台形は全部で n 個あるので、全ての台形の面積の合計は

$$\sum_{i=1}^n \frac{(f(x_{i-1}) + f(x_i)) \times \Delta x}{2}$$

関数として $f(x) = \frac{1}{1+x^2}$ を例にとり、区間 $[0, 1]$ で積分してみよう。

この計算はシミュレーションの形にならない。あらかじめ分割数分の行数を確保しておく必要がある。

実は、この関数は積分すると $\text{Arctan}(x)$ になる。 π の計算でやったように、この積分値は $\frac{\pi}{4}$ となるはずである。

台形公式による積分値の計算は、簡単ではあるが効率が悪い。より高級な方法としてシンプソンの方法、ガウスの方法などが知られている。高級ではあっても、式に従って Excel に計算させるだけなので、式の入力さえ間違えなければ簡単に積分をすることができる。

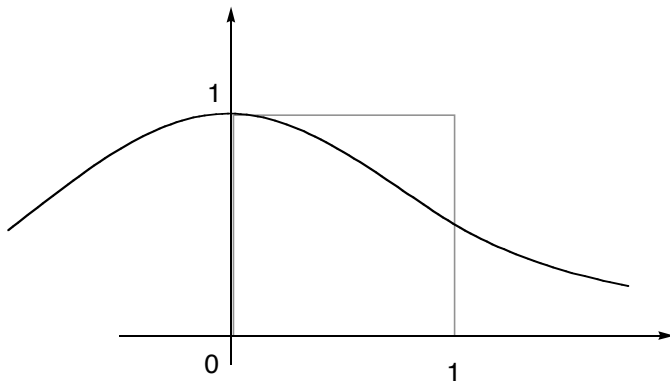
【課題】

区間分割数 n によって誤差がどの程度になるか調べてみよう。

1.1. モンテカルロ法による数値積分

モンテカルロはカジノの街である。カジノは確率に支配されている。確率を利用した計算方法をモンテカルロ法と呼ぶ。確率は、試行回数が多ければ多いほど正確な値を与える。コンピュータのような疲れを知らない機械であれば、確率を正確に求めることで、例えば積分をすることができる。

関数として $f(x) = \frac{1}{1+x^2}$ を例にとり、区間 $[0, 1]$ で積分してみよう。この関数の概形は次のようになっている。



求めたい面積は、原点から $x=1$ 、 $y=1$ までの範囲内にある。この正方形（面積は 1）の中にランダムに点を取った時、それが $f(x) = \frac{1}{1+x^2}$ よりも小さい確率は、求めたい面積に正確に等しい。

0 から 1 の間の乱数 (`rand()` という特殊な関数で発生させることができる) 2 組で、正方形の中の点をランダムに指定し、それが $f(x) = \frac{1}{1+x^2}$ の上下どちらにあるかを判定する。全試行数のうち、条件を満たすものの割合が確率になるので、そこから積分値を計算する。前項のように、この値は $\frac{\pi}{4}$ となるはずであるから、 π を計算する。

【課題】

全試行数と誤差の関係を調べてみよう。試行数を 10 倍にするのと、10 回試行して平均するのでは、結果は同じだろうか？

- ・ if 関数による判定