# 実験数学1（6/15）

担当：宮武 勇登

PCを立ち上げてログインしてください

CLEから資料をダウンロードしてください

とりあえず何かoctaveのプログラムを実行して

グラフを描きましょう

(コマンドウィンドウで plot([0,1]) とかでよいかも)

# レポートのコメントに対して



- 実はプログラミング以前にPC（コンピュータ）について良くわかっていない
  - ハード面はともかく，フォルダの階層構造など
  - 一定の知識は必要です
- C言語に慣れるとOctaveに違和感
  - おっしゃる通り… CとmatlabのC違いを調べてみるとよい
  - 例えば http://u0u1.net/SOYv （プログラミング以外の要素も学べる）
  - Pythonユーザーは https://cheatsheets.quantecon.org/

# レポートのコメントに対して



- Macへのインストールがうまくいかない
  - 宮武もhomebrewからは上手くインストールできませんでした
- Octaveは肌に合わないからPythonでやってみた
  - Octaveを強要するつもりは一切ないので
  - 好きな言語を使ってください（もちろんCでもよい）
- 数学の内容が難しい
  - がんばりましょう！
- 課題の答えは（翌週ではなく）その時間中に…
  - 気持ちは分かります…
  - 一方で，3〜4時間はじっくり考えてほしい…

# レポートのコメントに対して



- プログラムの意味がよくわからない
  - まずは，総和などの単純なプログラムを書いてみる
  - 自分で考えることはしないが，単純な計算ができ，かつ
    指示は守る子供に対して，どのような手順で指示を出しますか？
- 計算速度が遅いのになぜOctave（そもそもなぜ遅い？）
  - 阪大の計算機で容易に使える
  - 最初にコンパイルをしないので，メモリの確保などが
    行き当たりばったり
- 小レポートの頻度（A4一枚だとすぐにうまってしまう）

  あと1,2回だします（もちろんたくさん書いてOKです）

# レポートのコメントに対して



- zerosや配列の定義

  zeros(n,m) : n x m の配列のすべての要素に0が入ったもの

- MathematicaやLaTeXの方が使いやすい？
  - Mathematica: 数式処理がメイン
  - LaTeX: 文書作成がメイン
- 一からプログラミングするのは難しい
  - おっしゃる通りです…
  - 特に新しい言語をはじめるときは，相当な（時間の）勉強が必要

# レポートのコメントに対して



- おすすめの本？

  webにも情報はたくさんありますし，まずはどんなことを勉強したい（調べたい）かを（漠然とでいいので）考えてみることが大事かも

- IT系で働きたいが学んでおくべき学問や講義
  - 微積・線形代数・微分方程式・フーリエ解析 etc.
  - （一つでもよいので）得意な言語があると良い
  - 計算機やwebについての一定の知識

    （GitHubが使えるとか，簡単なwebページは公開できるとか）

# レポートのコメントに対して（昨年）



- %はコメント．　% と %%に特に違いはない．

  宮武は，%はその行の説明，%%はブロックの説明に利用

- 【重要】エラーはコマンドウィンドウに表示される

- 「x」を２つの意味で使っていました．
  - 無名関数「@(x)～」の引数
  - 配列（ベクトル）

  別の文字にした方が混乱がなく適切です（@(t)～とか）

- $f(x) = x^2$のとき，定数分だけずれた人

  →おそらく$a_0/2$の項を忘れています

# レポートのコメントに対して（昨年）



- Octaveはどういう意味で「簡単」なのか？
  - 型の定義が不要
  - 数学関数 や アルゴリズムの多く が既に実装済み
- 「一様収束」の確認は「各点収束」の確認でOK？
  - ダメ です
  - 各点収束しても一様収束するとは限りません.
  - フーリエ級数は（ある条件下で）不連続点以外で各点収束はしますが，一様収束するとは限りません.
  - さらに詳しく知りたいひとは「ノルムの同値性」について調べてみるとよいでしょう
    「ノルムの同値性」は関数解析や数値解析で必須の概念です

# ノルムの同値性



ラフにいえば

- 有限次元（例えばベクトル）
  - ふたつのベクトル間の距離を定義するとき，定義の仕方はいろいろある
  - ある定義のもとで，二つのベクトルが十分近い場合，別の定義のもとでも，必ず近い関係にある
- 無限次元（例えば関数）
  - ある定義のもとで，二つの関数が十分近い場合でも，別の定義のもとでは，必ずしも近い関係にない

# Tips：検索は英語でも試してみる



たとえば，無名関数がよくわからないとき，

- Googleで「“octave” “無名関数”」と検索
  →207件
- とりあえず，「無名関数」と検索して，例えばwikipediaを見てみると，無名関数は英語では「anonymous function」であることがわかる
- Googleで「 “octave” “anonymous function” 」と検索
  →71,500件（情報量が桁違い）

# Tips：検索は英語でも試してみる



この場合，おそらく一番上にでてくるのは

Octave 本家のドキュメント

https://octave.org/doc/v4.2.0/Anonymous-Functions.html

例もあって分かりやすい（ように思う）

( https://octave.org/doc/v4.2.0/ に様々なことが書いてあることがわかる)

# Tips：検索は英語でも試してみる



## 一般に，英語の方が桁違いに情報量が多い

- 日本語の検索では絶対に分からないような情報が得られることが多い
- 一方で，情報に溺れないように，適切なキーワードで検索したい（だんだんとコツをつかめるはず）

なお，4年生になると，数学の洋書を読むことも増えるし（辞書は使ってよいが英語は100%理解できないと苦しい），プログラミングを通して少しずつ数学で使う英語に慣れていくと近い将来役に立つはず.

# 今日の内容



- フーリエ級数展開
- 常微分方程式の数値解法

**やりたいこと**

$f(x),\ x \in (-\pi, \pi)$ を 三角関数を使った無限級数で近似

つまり

$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + \sum_{k=1}^{\infty} b_k \sin(kx)$$

Q. 係数$\{a_k\}_{k=0}^{\infty}, \{b_k\}_{k=1}^{\infty}$をどう定めればよいか？

# フーリエ級数展開（続き）



$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + \sum_{k=1}^{\infty} b_k \sin(kx)$$

右辺の部分和を$S_n(x)$とする：

$$S_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos(kx) + \sum_{k=1}^{n} b_k \sin(kx)$$

次の性質が成り立つように定める：

$$\lim_{n\to\infty} \int_{-\pi}^{\pi} (f(x) - S_n(x))^2 dx = 0$$

# フーリエ級数展開（続き）



$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(kx)\,dx$$

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(kx)\,dx$$

直ちにわかること：

奇関数 $\Rightarrow a_k = 0$ （sin 関数だけで展開できる）

偶関数 $\Rightarrow b_k = 0$ （cos 関数だけで展開できる）

# 理由（証明ほど厳密ではない）



$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + \sum_{k=1}^{\infty} b_k \sin(kx)$$

≈を=だとして両辺に$\sin(ix)$をかけて区間$(-\pi, \pi)$で積分

$$\int_{-\pi}^{\pi} f(x) \sin(ik) dx$$

$$= \frac{a_0}{2} \int_{-\pi}^{\pi} \sin(ik)\, dx + \sum_{k=1}^{\infty} a_k \int_{-\pi}^{\pi} \sin(ix) \cos(kx)\, dx$$

$$+ \sum_{k=1, k\neq i}^{\infty} b_k \int_{-\pi}^{\pi} \sin(ix) \sin(kx)\, dx + b_i \int_{-\pi}^{\pi} \sin^2(ix) dx$$

$$= \pi b_i$$

青い積分 ＝ 0

$f(x) = x, x \in (-\pi, \pi)$を例に
$f(x)$ と$S_n(x)$ をプロットするプログラム

```
%% 元の関数
function y = original_func (x)
    y = x;
end
% 奇関数なのでsin関数の重ね合わせだけで表現できる
```

# o2.m（続き）



%% パラメータなど

```
x = [-pi:pi/1000:pi];
```

%[a:b:c]で[a,a+b,a+2b,a+3b,...,c]というベクトルを表現

```
n = 100; % sin(nx)まで級数を求める

b = zeros(n,1); % 係数保持ようの配列
```

# o2.m（続き）

```
%% フーリエ級数：係数の計算

for k = 1 : n

  % b(k,1) = 1/pi * quad(@(x) x * sin(k*x),-pi,pi);

  b(k,1) = 2*(-1)^(k+1)/k;

end

% quad(f(x),a,b) は f(x) を区間[a,b]で数値積分

%「@(x) xの関数」で関数を定義したことになっている（無名関数）

% もちろん，厳密に計算できる場合は，そうした方がよい
```

# o2.m（続き）



```
%% フーリエ級数：実際の級数

function y = fourier_func(x,n,b)

  y = 0;

  for i = 1 : n

    y += b(i,1) * sin(i*x);

  end

end
```

要するに y = b(1,1)*sin(x) + b(2,1)*sin(2x) + …+b(n,1)*sin(nx)

# 課題



- $n$を色々と変えてみて収束の雰囲気を実感する
- プログラムを少し修正し

$$err(n) := \int_{-\pi}^{\pi} (f(x) - S_n(x))^2 dx$$

を横軸$n$としてプロットし，収束を確かめる

- $\sup_{x \in (-\pi, \pi)} |f(x) - S_n(x)|$ は収束するか？

（要するに一様収束しているか？）→していない

- 他の関数（例えば $f(x) = x^2$）でも試してみる

（不連続関数でもよい）

- $err(n) := \int_{-\pi}^{\pi} (f(x) - S_n(x))^2 dx$

$$= \int_{-\pi}^{\pi} (x - \sum_{k=1}^{n} b_k \sin(kx))^2 dx$$

$$= \int_{-\pi}^{\pi} \left( x^2 + \sum_{k=1}^{n} b_k^2 \sin^2(kx) - 2 \sum_{k=1}^{n} b_k x \sin(kx) \right) dx$$

$$= \frac{2\pi^3}{3} + \sum_{k=1}^{n} b_k^2 \pi - 2 \sum_{k=1}^{n} b_k \frac{2\pi(-1)^{k+1}}{k}$$

$$= \frac{2\pi^3}{3} + \sum_{k=1}^{n} \left( \pi b_k^2 + \frac{4\pi(-1)^k b_k}{k} \right)$$

$$= \frac{2\pi^3}{3} - \sum_{k=1}^{n} \frac{4\pi}{k^2} \quad \left( \because b_k = \frac{2(-1)^{k+1}}{k} \right)$$

- これを実装すればよい

$$err(n) = \frac{2\pi^3}{3} - \sum_{k=1}^{n} \frac{4\pi}{k^2}$$

実験してみると，　$err(n) \to 0$ が観察される

実際，これは正しい（Fourier解析の議論より）

すなわち

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
err = zeros(n,1); % 誤差保持用の配列
%% errの計算
for k = 1 : n
  err(k,1) = 2*pi^3/3;
  for i = 1 : k
    err(k,1) += b(i,1)^2*pi + 4*pi*b(i,1)*(-1)^i/i;
  end
end

semilogy(err) %プロット
```

$$err(k)= \frac{2\pi^3}{3} + \sum_{i=1}^{k} \left( \pi b_i^2 + \frac{4\pi(-1)^i b_i}{i} \right)$$

# 今日の内容



- フーリエ級数展開
- 常微分方程式の数値解法

# 常微分方程式の初期値問題



- 常微分方程式の初期値問題

$$\frac{d}{dt}y = f(y),\ y(0) = y_0 \in \mathbb{R}^n \quad (f: \mathbb{R}^n \to \mathbb{R}^n)$$

- $y(t)$を知りたいが，一般に初等関数の組み合わせで解析解を表現できない

$$y(t) = y_0 + \int_0^t f\big(y(s)\big)ds$$

$y(t)$が分からないと計算できない

積分をうまく近似することが，
常微分方程式の数値解法の基本

# 陽的Euler法



$$y(\Delta t) = y_0 + \int_0^{\Delta t} f\big(y(s)\big)ds$$

$$\approx y_0 + \Delta t f(y_0)$$

以前の講義の
Algorithm1
を用いた

- $y(\Delta t)$の近似を $y_1$ と書くことにすると

$$y_1 = y_0 + \Delta t f(y_0)$$

で近似を定義できる.

- 一般には

$$y_{n+1} = y_n + \Delta t f(y_n), \qquad n = 0,1,2, \ldots$$

とすれば, $y_n \approx y(n\Delta t)$

## 微分方程式 $\frac{d}{dt}y = -y,\ y(0) = 1$

```
%% 微分方程式の右辺の定義
function f = func (x)
   f = -x;
end

%% パラメータなど
T = 1; %目標時刻
n = 10; %分割数
dt = T/n; %時間刻み
t = [0:dt:T]; %ベクトル
```

```
%% 初期値
y0 = 1;

%% 数値解を格納するベクトル
yEE = zeros(1,n+1); %Euler法用
yEE(1,1) = y0; %初期値
yRK = zeros(1,n+1); %RK法法用
yRK(1,1) = y0; %初期値

%% 厳密解：あとで比較するため
function y = exact(t,c)
  y = c*exp(-t);
end
```

```
%% オイラー法の一ステップ
function y = ExEuler(x,dt)
  y = x + dt*func(x)
end

%% Runge-Kutta法の一ステップ
function y = RK(x,dt)
  % ここを書く
end

%% 反復
for i = 1:n
  yEE(i+1,1) = ExEuler(yEE(i,1),dt);
  % yRK(i+1,1) = RK(yRK(i,1),dt);
end
```

# その他の解法



- 陰的Euler法

$$y_1 = y_0 + \Delta t f(y_1)$$

- Runge-Kutta法

$$Y_1 = y_0$$

$$Y_2 = y_0 + \frac{\Delta t}{2} f(Y_1)$$

$$Y_3 = y_0 + \frac{\Delta t}{2} f(Y_2)$$

$$Y_4 = y_0 + \Delta t f(Y_3)$$

$$y_1 = y_0 + \frac{\Delta t}{6}\left(f(Y_1) + 2f(Y_2) + 2f(Y_3) + f(Y_4)\right)$$

- 課題：RK法を実装する

# 予告



次週，レポート課題を出題します