情報学基礎

11章　数式処理と計算ツール

管理工学科　篠沢 佳久
（代講　情報工学科 高田眞吾先生）

理工学を支える道具たち

- 高校までの数学には電卓で十分
  - 大学以降の数学のための道具は関数電卓で十分？
- 数式処理
  - 数式を記号のまま処理
  - 四則演算、微積分、因数分解、級数展開
  - 理論的・理学的な問題に適す
- 数値計算
  - 数式を具体的な値を用いて計算
  - ベクトル計算や行列計算の基盤
＋高度なプログラミング機能
  - 実際的・工学的な問題に適す
- 可視化処理
  - 網羅的な数値計算を行い、グラフィカルに表示

11.1 数式処理システム

- 有料／無料のソフトウェア
  - Reduce, Macsyma, Maple, Mathematica, MuPad
  - 各々のウェブにあるライセンス条項を読み、
    諸君が無料で利用可能なものを探してみると良い
- Mathematicaは有料であるが、
  慶應義塾大学は包括契約を締結している
  - ITCのパソコン室全室で利用可能！
  - 諸君のパソコンにもインストール可能！
    （個人で購入したら25,000円もします）
    - 日吉ITCのソフトウェアライセンス利用ページ参照

慶應義塾 日吉 ITC
Hiyoshi Information Technology Center

- TOP
- コンピュータ
- ネットワーク
- ソフトウェア
- keio.jp
- 利用案内
- お問い合わせ
- スケジュール
- 教職員向け情報

keio.jp 共通認証システム

慶應義塾大学 日吉キャンパス

トップ ＞トップ ＞ソフトウェア ＞ソフトウェアライセンス利用 ＞

Mathematica

ライセンス種別について

Mathematicaは、インストールしてご使用になる端末によって、以下のライセンスを使い分ける必要があります。

ライセンスの種別 と 導入PC例

シングルマシン用 (旧: マシン固有パスワード/セキュア 版)	• 慶應義塾が所有・管理するPC • 慶應義塾が賃貸借契約を結んで導入しているPC
教員用ホームユーズライセンス	• 教職員 本人が所有するPC
学生用ホームユーズライセンス	• 学生 本人が所有するPC [導入手順] バージョン 10 [PDF（.pdf）形式]

- 利用対象者/非対象者の詳細については、上記各ライセンスのページをご覧ください。
- 上記の表に掲載されていない「非対象者と共有利用しているPC」、「外部資金で整備され、その所有が義塾以外 PC」、「利用対象者の家族、親族と共有利用しているPC」にはインストールできません。

Mathematica

- Wolfram Research社が開発/販売している計算システムソフトウェア
- 数式処理が出発点（1988）
- 現在では数値計算機能、可視化機能を備えた数理系の万能ツールである
- オンラインヘルプが充実しており、使用方法の検索や閲覧が画面上で可能
  - ウェブでも公開されている http://www.wolfram.com/mathematica/
  - 自分で調べられるようになることが重要！
    - ドキュメントセンター、関数ナビゲータ、?コマンドなど

方程式の操作：展開

- 多項式を展開することができる

$$(x^3 - 1)(2x - 1)$$

In[1]:=

```
Expand[(x^3 - 1) (2 x - 1)]
```

Out[1]=

$$1 - 2\,x - x^3 + 2\,x^4$$

使い方は自分で調べられる

- ？に続けて知りたい内容を書く

In[26]:=

```
? Expand
```

Expand[*expr*] 式 *expr* における積と正の整数ベキを展開する.

Expand[*expr*, *patt*] パターン *patt* にマッチする項を含まない式 *expr* の要素の展開を避ける. ≫

- >> を押せば、さらに詳しく調べられる

方程式の操作:因数分解

- 代数式を既約因子の積の形式に書き直す

$$1 - 2x - x^3 + 2x^4$$

In[5]:=

```
Factor[1 - 2 x - x^3 + 2 x^4]
```

Out[5]=

$$(-1 + x)\ (-1 + 2\,x)\ \left(1 + x + x^2\right)$$

方程式の操作：簡約（テキストにない）

- 多項式の簡約 $(x-1)(x+1)(x^2+1)+1$

```
In[42]:=
Simplify[(x - 1) (x + 1) (x^2 + 1) + 1]
Out[42]=
x^4
```

- 仮定を用いた簡約や証明も可能 $x^2 > 3$

```
In[43]:=
Simplify[x^2 > 3, x > 2]
Out[43]=
True
```

関数の不定積分

- 標準的な規則が適用されて不定積分できる

$$\int 1 - 2x - x^3 + 2x^4 dx$$

In[2]:=

```
fp = Integrate[1 - 2 x - x^3 + 2 x^4, x]
```

Out[2]=

$$x - x^2 - \frac{x^4}{4} + \frac{2\,x^5}{5}$$

- 次で使いたいので **fp** と置いている

微分

- 先の不定積分の結果は **fp** に保存してある
  これを微分すれば元の式に戻るはず

$$\frac{d\,\mathbf{fp}}{dx} = \frac{d\left(x - x^2 - \frac{x^4}{4} + \frac{2x^5}{5}\right)}{dx}$$

In[3]:=

```
D[fp, x]
```

Out[3]=

$$1 - 2\,x - x^3 + 2\,x^4$$

微分

- 同じ式を**x**について3階微分させてみよう

$$\frac{d^3\,\mathbf{fp}}{dx^3} = \frac{d^3\left(x - x^2 - \frac{x^4}{4} + \frac{2x^5}{5}\right)}{dx^3}$$

In[4]:=

```
D[fp, {x, 3}]
```

Out[4]=

$-6\,x + 24\,x^2$

方程式の解法

- 複素数の厳密解を得ることができる

$$1 + x + x^2 = 0$$

In[6]:=

```
Solve[1 + x + x^2 == 0, x]
```

Out[6]=

$$\{\{x \to -(-1)^{1/3}\}, \{x \to (-1)^{2/3}\}\}$$

ReplaceAll

直前の入力

In[7]:=

```
ComplexExpand[x /. %]
```

Out[7]=

$$\left\{-\frac{1}{2} - \frac{i\sqrt{3}}{2}, -\frac{1}{2} + \frac{i\sqrt{3}}{2}\right\}$$

方程式の解法

- 近似解を得ることもできる

$$1 + x + x^2 = 0$$

```
In[8]:=
    NSolve[1 + x + x^2 == 0, x]
Out[8]=
    {{x → -0.5 - 0.866025 ⅈ},
     {x → -0.5 + 0.866025 ⅈ}}
```

三角関数の扱い（テキストにない）

- ExpandやFactorは三角関数を操作しない
- 三角関数式を操作する命令もある
  - 展開 TrigExpand
  - 分解 TrigFactor, TrigFactorList
  - 簡約 TrigReduce
  - 指数関数に変換 TrigToExp
  - 指数関数から変換 ExpToTrig

$$\cos 2x = \cos x^2 - \sin x^2$$

```
In[27]:=
        TrigExpand[Cos[2 x]]
Out[27]=
        Cos[x]^2 - Sin[x]^2
```

グラフ化

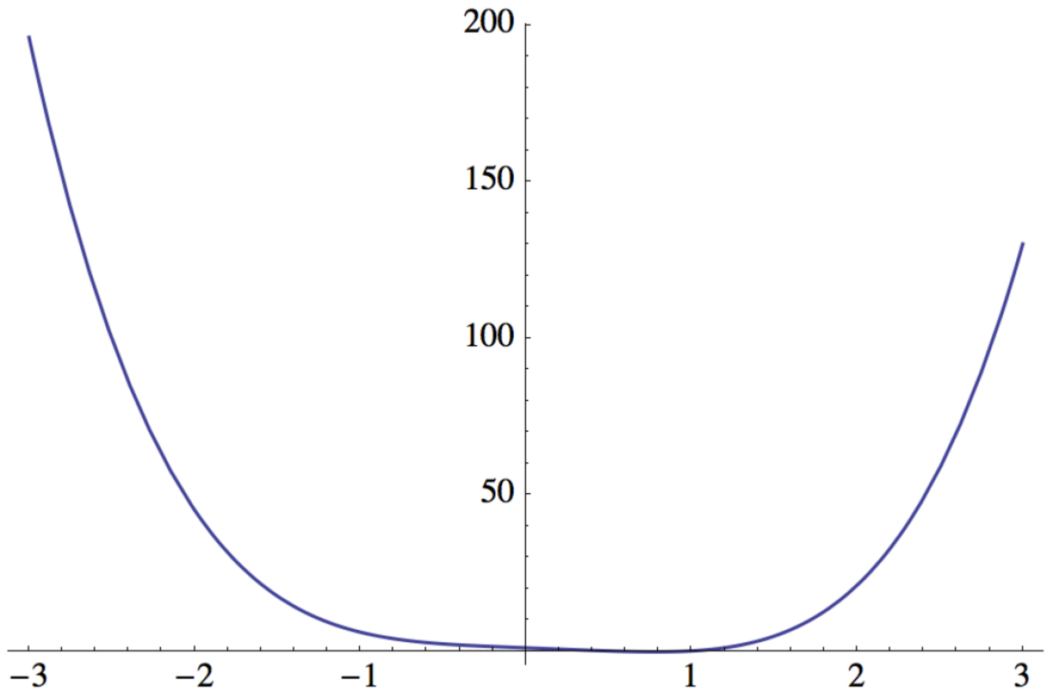
- 数値計算を行い、グラフにプロットできる

$$y = 1 - 2x - x^3 + 2x^4$$

In[12]:=

```
Plot[1 - 2 x - x^3 + 2 x^4 == 0, {x, -3, 3}]
```

Out[12]=

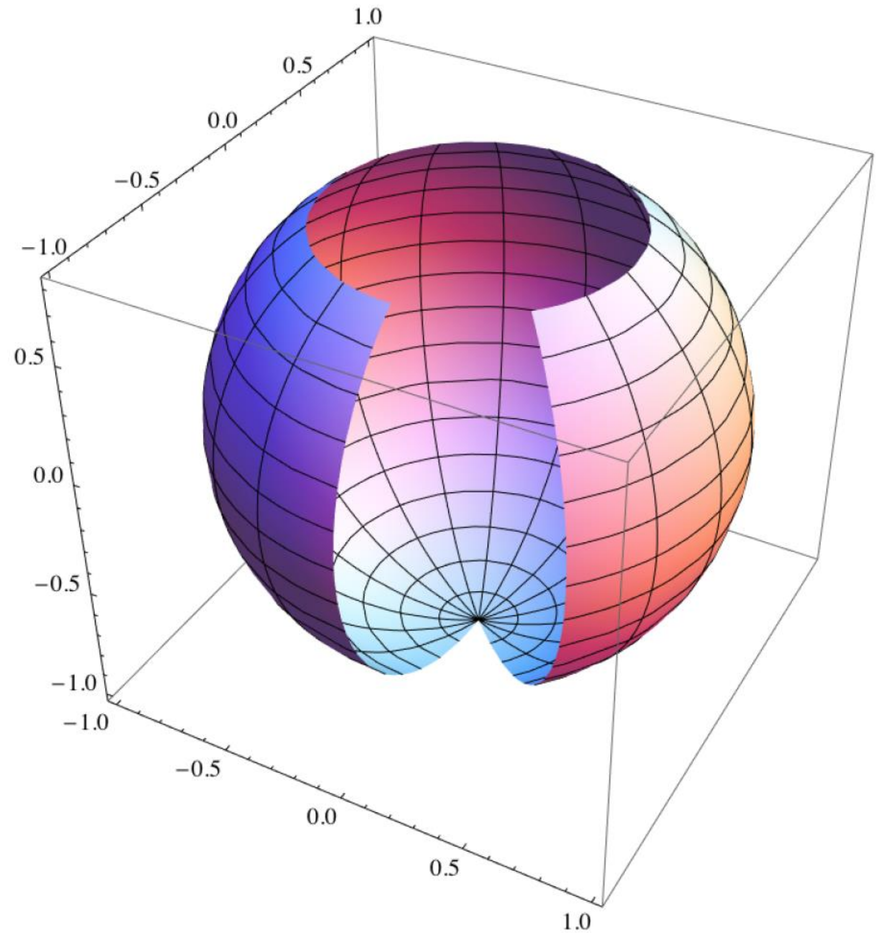


3次元グラフィックス

- 曲面をプロット

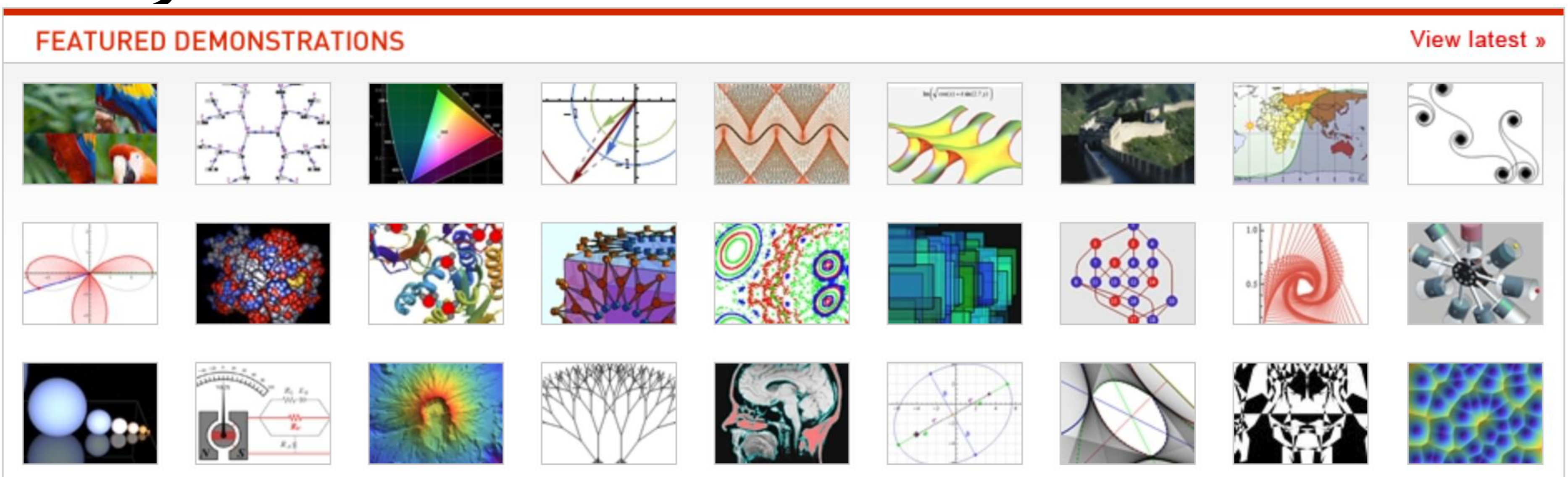
```
In[13]:= ParametricPlot3D[{Sin[u] Cos[v], Sin[u] Sin[v], Cos[u]},
  {u, Pi / 5, Pi}, {v, -Pi / 4, 3 Pi / 2}]
```

Out[13]=



Wolfram Demonstrations Project

- http://demonstrations.wolfram.com/
- 多種多様な動く資料を見ることができる
- 使いこなしたくなる動機づけになることでしょう



Mathematicaに慣れておこう

- 日々使って、慣れておくべき道具です
  - 今後の勉強の学習効率が上がること間違いなし
- 数学、物理、化学などの演習問題を解く際の検算用途で使い始めることを強く勧めます
  - 導出過程が得られないので、検算用途にのみ

11.2 数値計算のツール

- 有料／無料のソフトウェア
  - MATLAB, Scilab, Octave
  - 各々のウェブにあるライセンス条項を読み、諸君が無料で利用可能なものを探してみると良い
- MATLABは有料であるが、慶應義塾大学は包括契約を締結している
  - ITCの一部のパソコン室で利用可能！
  - 日吉は3室のみ、矢上は全室

慶應義塾 日吉 ITC
Hiyoshi Information Technology Center

- TOP
- コンピュータ
- ネットワーク
- ソフトウェア
- keio.jp
- 利用案内
- お問い合わせ
- スケジュール
- 教職員向け情報

keio.jp 共通認証システム

慶應義塾大学 日吉キャンパス

慶應義塾

慶應義塾の 電力使用状況

トップ ＞ソフトウェア ＞ソフトウェアライセンス利用 ＞

MATLAB

ライセンスオプションについて

MATLABは、インストールしてご使用になる端末によって、以下のライセンスオプションを使い分ける必要があります。

ライセンスオプション と 導入PC例

キャンパスオプション	• 常勤教職員本人が所有するPC • 研究室などに設置されているPC
スチューデントオプション	• 学生 本人が所有するPC

- 利用対象者/非対象者の詳細については、上記各ライセンスオプションのページをご覧ください。
- 上記の表に掲載されていない「非対象者と共有利用しているPC」、「外部資金で整備され、その所有が義塾以外の組織となるPC」、「利用対象者の家族、親族と共有利用しているPC」にはインストールできません。

MATLAB オンライントレーニング

オンラインで受講できる自己学習形式のトレーニングです。

MATLAB オンライントレーニング

最終更新日：2016年6月20日

MATLAB

- MathWorks社が開発/販売している数値計算ツール
- 行列計算に基づく高水準プログラミング言語が出発点(1984)
- 現在は数式処理を扱うMuPadが組み込まれている
- GUIを使ってプラントを設計し、シミュレーションを行ったり、実際の機器を操作することができる
- 多くのツールボックスにより、様々な分野のシミュレーションを容易に実現可能
  - 統計解析、最適化、信号処理、金融モデル、生命科学
- 使用法はオンラインヘルプで検索/閲覧できる
  - 自分で調べられるようになることが重要！ help, doc
  - オンラインチュートリアルも充実している

ベクトル・行列の入力方法

- 要素を各括弧[]で括る
- 列要素は
  空白かカンマ,で区切る
- 改行はセミコロン;で区切る

```
>> a=[1 2 4]

a =

     1     2     4

>> b=[18; 71; -13]

b =

    18
    71
   -13
```

```
>> A=[ 1 -5 1; 6 -14 5; 21 13

A =

     1    -5     1
     6   -14     5
    21    13   -10
```

行列計算に関連する関数コマンド

- 大きさ、ランク、転置、行列式、トレース

```
>> size(A)

ans =

     3     3

>> rank(A)

ans =

     3
```

```
>> A'

ans =

     1     6    21
    -5   -14    13
     1     5   -10
```

```
>> det(A)

ans =

 -378.0000

>> trace(A)

ans =

   -23
```

使い方は自分で調べられる

- help に続けて知りたいことを書く

```
>> help size
 size   Size of array.
    D = size(X), for M-by-N matrix X, returns the two-element row vector
    D = [M,N] containing the number of rows and columns in the matrix.
    For N-D arrays, size(X) returns a 1-by-N vector of dimension lengths.
    Trailing singleton dimensions are ignored.
```

- doc を使えば、さらに詳しく調べられる

行列計算に関連する関数コマンド

- 逆行列、固有ベクトル

```
>> inv(A)

ans =

   -0.1984    0.0979    0.0291
   -0.4365    0.0820   -0.0026
   -0.9841    0.3122   -0.0423
```

```
>> [X D]=eig(A)

X =

  -0.1367 - 0.1956i  -0.1367 + 0.1956i  -0.1497
  -0.3934 + 0.0345i  -0.3934 - 0.0345i  -0.4865
  -0.8872            -0.8872             0.8608


D =

  -1.0000 + 4.1231i        0                  0
        0            -1.0000 - 4.1231i        0
        0                  0            -21.0000
```

X(1:3,3)
で行列の一部を選べる

D(3,3)

行列・ベクトルの四則演算

- +-*/ 記号は行列演算・ベクトル演算を表す
- 要素毎の四則演算は演算子の前にピリオドを付ける

```
>> A*A

ans =

    -8     78    -34
    27    231   -114
  -111   -417    186

>> A.*A

ans =

     1     25      1
    36    196     25
   441    169    100
```

逆行列を確かめてみよう

- 元の行列と逆行列の積が単位行列になるか

```
>> A*inv(A)

ans =

    1.0000    0.0000   -0.0000
   -0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000
```

$$AA^{-1} = I$$

- -0.0000 とは？ 精度を上げて表示してみよう

```
>> format long
>> A*inv(A)

ans =

   1.000000000000000   0.000000000000000  -0.000000000000000
  -0.000000000000001   1.000000000000000  -0.000000000000000
   0.000000000000004  -0.000000000000000   1.000000000000000
```

固有値を確かめてみよう

- 固有ベクトルと行列の積は、
  固有ベクトルのスカラー倍になっているか

Ax

```
>> A*X(1:3,3)

ans =

    3.1436
   10.2168
  -18.0758
```

λx

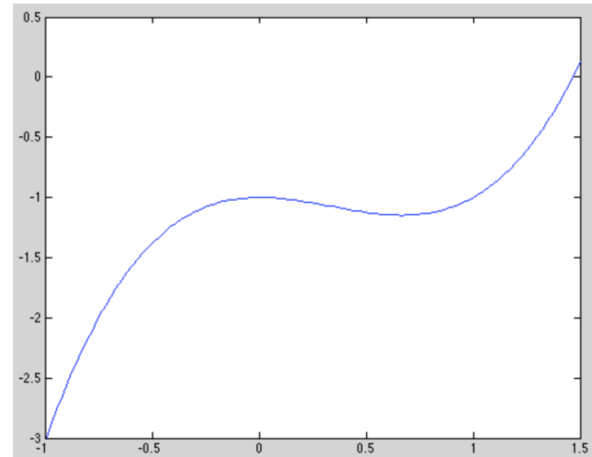
```
>> D(3,3)*X(1:3,3)

ans =

    3.1436
   10.2168
  -18.0758
```

関数のプロット

$$f(x) = 4x^3 - 4x^2 - x + 1$$

- 予め関数をMファイルに定義
  - f.m というファイル名で作成

```
function y=f(x)
y=4*x.^3-4*x.^2-x+1;
end
```



- プロットしたい範囲の値をxベクトルに用意
  - 例えば -1から1.5の範囲を
    - 0.01刻み `x=-1:0.01:1.5;`
    - 100点 `x=linspace(-1,1.5,100);`
- プロット命令を実行

```
plot(x, f(x))
```

ニュートン法について（p.131）

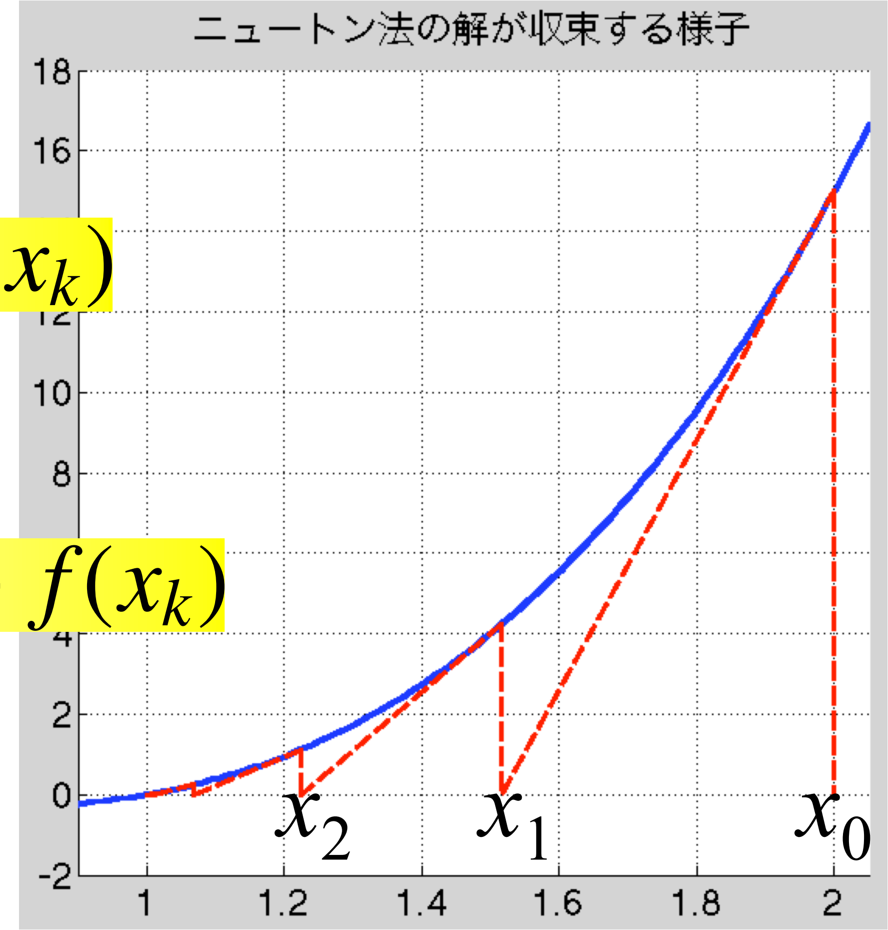
- 1階微分可能な代数方程式の求解法の1つ
- $(x, f(x))$を通り
  傾き$f'(x)$の直線

$$y = f'(x_k)(x - x_k) + f(x_k)$$

がx軸と交差する
点を次のxとする

$$0 = f'(x_k)(x_{k+1} - x_k) + f(x_k)$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



ニュートン法の準備

- 微分関数とニュートン法の処理方法をMファイルとして作成 $f'(x) = 12x^2 - 8x - 1$
  - fprime.m というファイル名で作成

```
function y=fprime(x)
y=12*x.^2-8*x-1;
end
```

  - newton.m というファイル名で作成

```
k=0;
while abs(x-xprev)>eps*abs(x)
    xprev=x;
    x=x-f(x)/fprime(x);
    k=k+1;
end
```

ニュートン法の使用例

- 初期値を与えて実行すれば解が求まる

```
>> xprev=0; x=2;
>> newton
>> x

x =

    1.0000
```

どの解が求まるかは
初期値次第

$$4x^3 - 4x^2 - x + 1 = 0$$

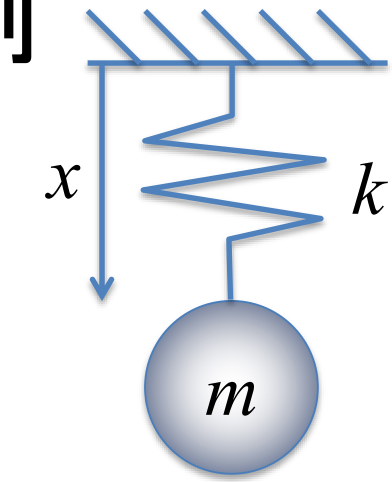
```
>> roots([4 -4 -1 1])

ans =

   -0.5000
    1.0000
    0.5000
```

- 組み込み関数rootsを使って検算してみる

シミュレーション（模擬実験）してみよう

- MATLABの真価は、シミュレーションの手軽さ
- Simulinkを使うと、GUIで実験環境を入力し、模擬実験ができる
- Simulinkのライブラリ（ツールボックス）は、様々な実験環境の構築を容易にしてくれる

シミュレーションの例



- バネで吊り下げた質点の運動
  - バネ定数k
  - バネの長さx
  - バネの自然長x_0
  - 質量m
  - 重力加速度g

$$f = ma = mg - k(x - x_0)$$

$$a = g - \frac{k}{m}(x - x_0)$$

$$v = \int a\, dt + v_0$$

$$x = \int v\, dt + x_0$$

MATLABによるシミュレーション

- MATLAB単体では、離散化し、数値積分に置き換えて、プログラムとして書く必要がある

```matlab
x0=0.1; m=1; k=10; g=9.8;
x=x0; v=0; dt=0.01;
hold on

for t=0:dt:10
    a=g-k*(x-x0)/m;
    v=v+a*dt;
    x=x+v*dt;
    plot(t, x, 'b-', 'LineWidth', 3)
end
```

$$a = g - \frac{k}{m}(x - x_0)$$

$$v = \int a\,dt + v_0$$

$$x = \int v\,dt + x_0$$

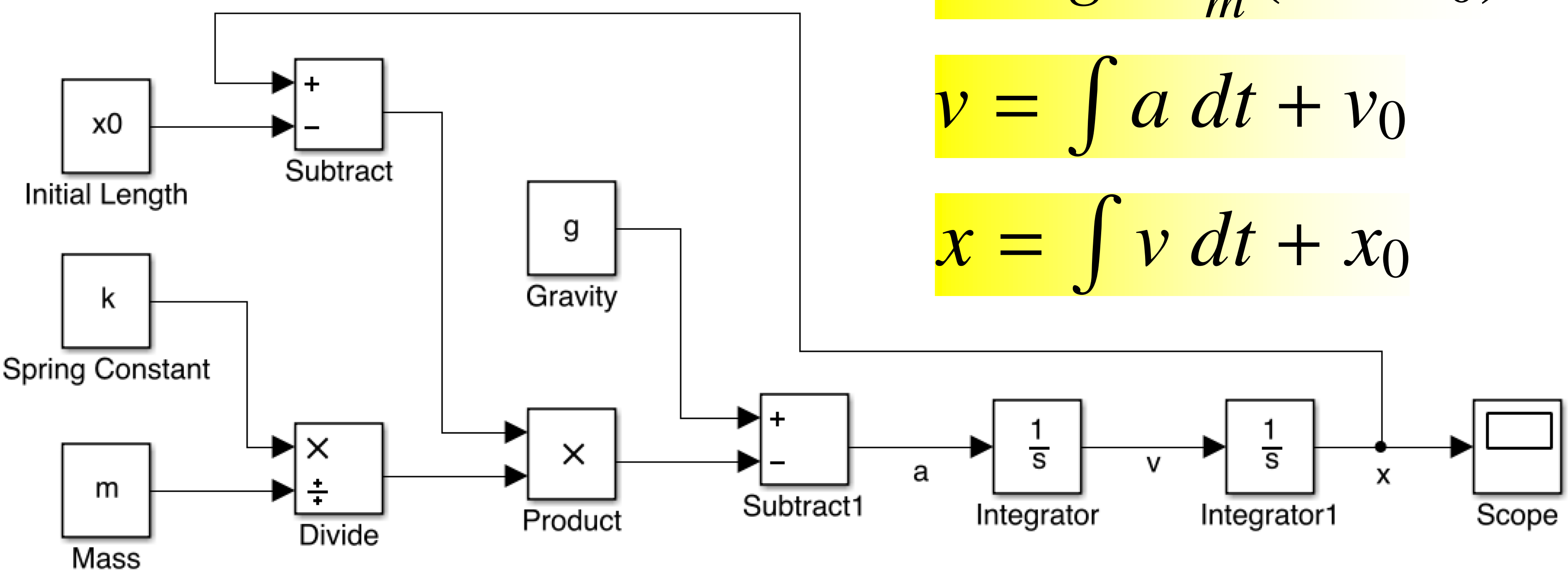
Simulinkによるシミュレーション

- 図形を配置し、結線するだけで実験できる
  - 積分は$1/s$、微分はsでOK

$$a = g - \frac{k}{m}(x - x_0)$$

$$v = \int a\,dt + v_0$$

$$x = \int v\,dt + x_0$$



MATLAB/Scilabに慣れておこう

- 日々使って、慣れておくべき道具です
  - 今後の学習効率が上がること間違いなし
- 実験結果の検証用途で使い始めることを強く勧めます

期末試験に関して

- MathematicaやMATLABの関数名を問うような問題は出題しません
- 試験対策よりも、自分のために使えるようになっておくことが大切です