期末試験と小テストについて

追試験を実施する場合は次の日程とします。

- 追試験の対象者は2月１日までに掲示します。
- 追試験の実施予定日は
  2013年2月5日（火）2時限目

小テストの再提出期限を追試験日までとします。

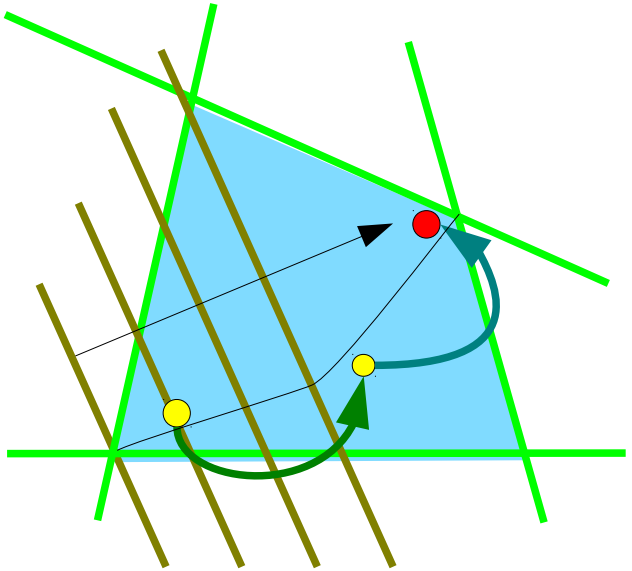
単位取得のために小テストの高評価が必要な場合は通知します。

修学支援システムのメッセージ機能を使います。

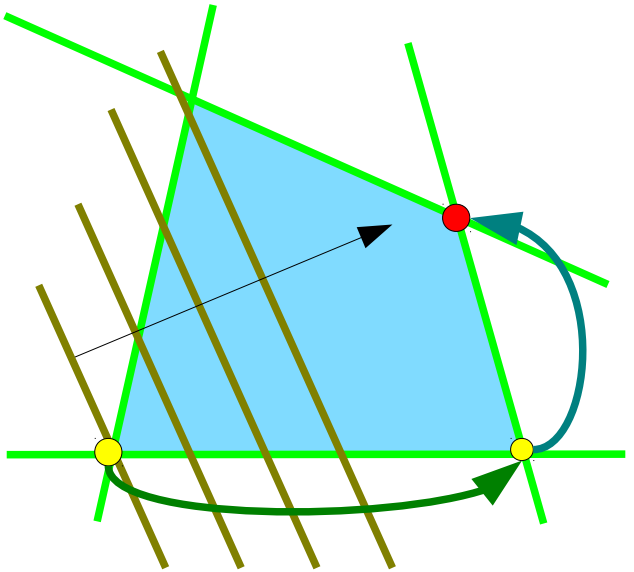
数理計画法

第15回：まとめ

自己双対型内点法



単体法(simplex法)



単体法は線形計画問題に特化した専用の解法であり、線形計画問題の特徴に対応した方法で、本質的に可算個の可能性を順番に辿る直接的方法である。
自己双対型内点法は、線形計画問題の解法ではあるが、実際の実行過程は非線形最適問題の反復解法になっている。

自己双対型内点法のNewton法

- 双対ギャップのゼロ点 $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{0}$ をニュートン法で求める
  制約式を満たすにはどうする？
- 双対ギャップを主・スラック変数を用いて表わせば
  $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}=(\boldsymbol{c}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})=\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{s}=\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{y}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{pmatrix}$
  制約式は変数の非負性に対応する。
- 逆に変数の非負性より双対ギャップのゼロ点を求める問題に同等な連立方程式を導くことができる

$$(\boldsymbol{t}^{\mathrm{T}}\ \boldsymbol{y}^{\mathrm{T}})\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{pmatrix}=0\Longleftrightarrow\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{pmatrix}(\boldsymbol{t}^{\mathrm{T}}\ \boldsymbol{y}^{\mathrm{T}})=\boldsymbol{0}$$

- 初期点 $\boldsymbol{x}_0,\boldsymbol{y}_0,\boldsymbol{s}_0,\boldsymbol{t}_0\geq\boldsymbol{0}$　を何らかの方法で定め、
  その近傍で関係式 $(\boldsymbol{X}+\delta\boldsymbol{X})(\boldsymbol{Y}^{\mathrm{T}}+\delta\boldsymbol{Y}^{\mathrm{T}})=\boldsymbol{0}$の解を
  一次近似式を緩和した式から反復法で求める;
  $\boldsymbol{X}\delta\boldsymbol{Y}^{\mathrm{T}}+\delta\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}=\lambda\boldsymbol{1}\qquad\lambda\to 0\qquad\boldsymbol{X}\equiv\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{pmatrix},\quad\boldsymbol{Y}\equiv\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{y}\end{pmatrix}$

復習:1変数Newton法のゼロ点探索

- 1変数の場合、初期点 $\tilde{x}$の近傍での Taylor 展開を考えて

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x}) + \frac{1}{2} f''(\tilde{x})(x - \tilde{x})^2 + \ldots$$

- 1次(=線形)近似を得る

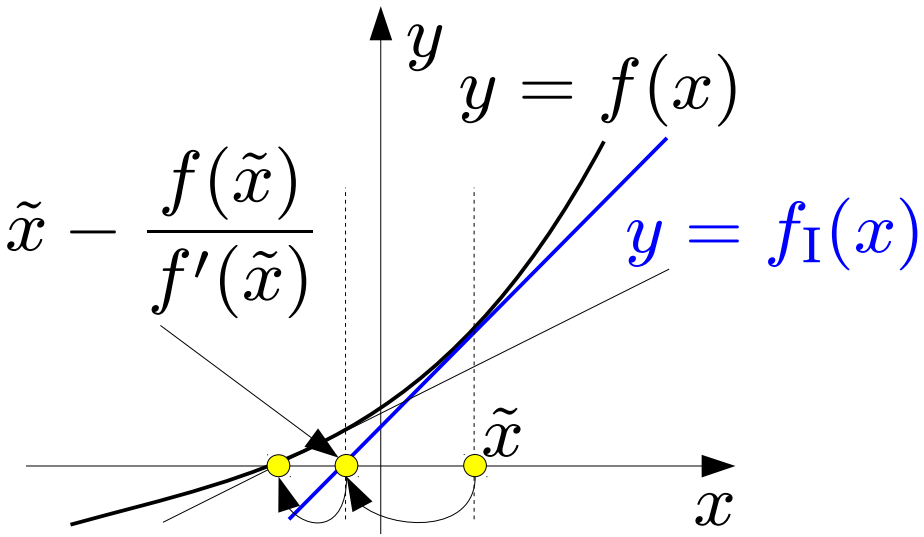
$$f(x) \sim f_{\mathrm{I}}(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x})$$

- 1次近似のゼロ点を求め

$$f_{\mathrm{I}}(x) = 0 \Leftrightarrow$$

$$x = \tilde{x} - f(\tilde{x})/f'(\tilde{x})$$

- 求めた x を $\tilde{x}$ として①に戻る
  ($f(x)$がゼロに近ければ終了)



復習：1変数Newton法の極点探索

- 1変数の場合、初期点 $\tilde{x}$ の近傍での Taylor 展開を考えて

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x}) + \frac{1}{2} f''(\tilde{x})(x - \tilde{x})^2 + \ldots$$

- 2次近似を得る

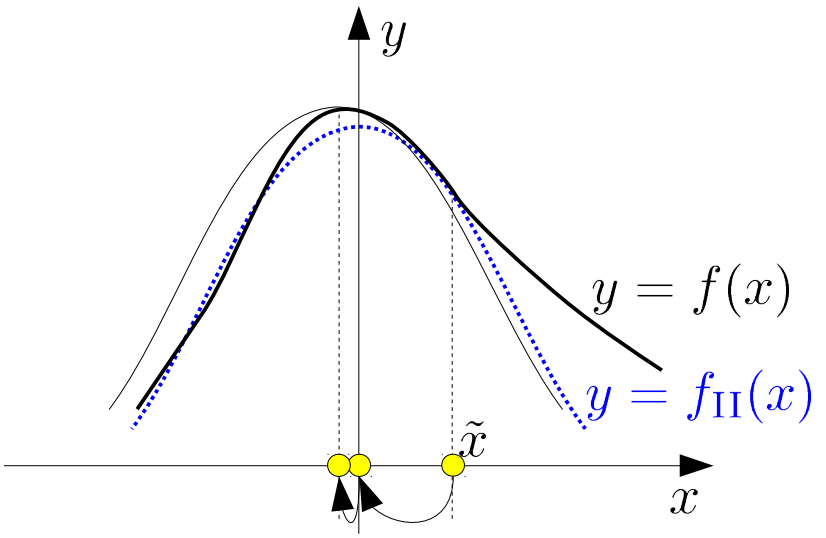
$$f_{\mathrm{II}}(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x}) + \frac{1}{2} f''(\tilde{x})(x - \tilde{x})^2$$

- 2次近似の極点を求め

$$f'_{\mathrm{II}}(x) = 0 \Leftrightarrow$$

$$\frac{\mathrm{d} f_{\mathrm{II}}}{\mathrm{d}(x - \tilde{x})} = 0 \Leftrightarrow$$

$$x = \tilde{x} - f'(\tilde{x}) / f''(\tilde{x})$$



- 求めた x を $\tilde{x}$ として①に戻る
（$f(x)$ がゼロに近ければ終了）

多変数Newton法によるゼロ点探索

- 初期点 $\tilde{\boldsymbol{x}}$ 近傍での Taylor 展開を考えて同様に、

$$f(\boldsymbol{x}) = \tilde{f} + \sum_j \tilde{f}_{x_j}\delta_j + \frac{1}{2}\sum_{j,k}\tilde{f}_{x_j x_k}\delta_j\delta_k + \dots$$

$$\tilde{f} = f(\tilde{\boldsymbol{x}}), \tilde{f}_{x_j} = \frac{\partial f}{\partial x_j}(\tilde{\boldsymbol{x}}), \tilde{f}_{x_j x_k} = \frac{\partial^2 f}{\partial x_j \partial x_k}(\tilde{\boldsymbol{x}}), \delta_j = x_j - \tilde{x}_j$$

一次近似: $f_{\mathrm{I}}(\boldsymbol{x}) = \tilde{f} + \sum_j \tilde{f}_{x_j}\delta_j = \tilde{f} + \tilde{\boldsymbol{f}}'^{\mathrm{T}}\boldsymbol{\delta}$

$$\tilde{\boldsymbol{f}}'^{\mathrm{T}} = \left(\frac{\partial f}{\partial x_1}(\tilde{\boldsymbol{x}}), \dots, \frac{\partial f}{\partial x_n}(\tilde{\boldsymbol{x}})\right), \boldsymbol{\delta}^{\mathrm{T}} = (\delta_1, \dots, \delta_n)$$

$f_{\mathrm{I}}(\boldsymbol{x}) = 0$ より、$\tilde{f} + \tilde{\boldsymbol{f}}'^{\mathrm{T}}\boldsymbol{\delta} = 0$ を解いて $\boldsymbol{\delta}$ を定める

反復公式: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta}$

多変数Newton法による極値探索

- 二次近似: $f_{\mathrm{II}}(\boldsymbol{x}) = \tilde{f} + \sum_j \tilde{f}_{x_j}\delta_j = \tilde{f} + \tilde{\boldsymbol{f}}'^{\mathrm{T}}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\delta}$

$$\tilde{\boldsymbol{f}}'^{\mathrm{T}} = \left(\frac{\partial f}{\partial x_1}(\tilde{\boldsymbol{x}}), \ldots, \frac{\partial f}{\partial x_n}(\tilde{\boldsymbol{x}})\right), \quad \boldsymbol{H} = \begin{pmatrix} f_{x_1x_1} \cdots f_{x_1x_n} \\ \vdots \quad \ddots \quad \vdots \\ f_{x_nx_1} \cdots f_{x_nx_n} \end{pmatrix},$$

$$\boldsymbol{\delta}^{\mathrm{T}} = (\delta_1, \ldots, \delta_n).$$

条件: $\left(\frac{\partial}{\partial \delta_1}, \ldots, \frac{\partial}{\partial \delta_n}\right)^{\mathrm{T}} (f_{\mathrm{II}} - \tilde{f}) = 0$

より連立方程式 $\tilde{\boldsymbol{f}}'^{\mathrm{T}} + \boldsymbol{H}(\tilde{x})\boldsymbol{\delta} = 0$ を得、$\boldsymbol{\delta}$ を求める

反復公式: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta}$

多変数Newton法の例

- 例題：$f(x) = x_1^3 + x_2^3 - 3x_1x_2$ の極小点を求めよ
- 二次近似: $f_{\mathrm{II}}(\boldsymbol{x}) = \tilde{f} + \sum_j \tilde{f}_{x_j}\delta_j = \tilde{f} + \tilde{\boldsymbol{f}}'^{\mathrm{T}}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\delta}$

  $\tilde{\boldsymbol{f}}'^{\mathrm{T}} = (f_{x_1}(\tilde{\boldsymbol{x}}), \ldots, f_{x_n}(\tilde{\boldsymbol{x}}))$, $\boldsymbol{H} = \begin{pmatrix} f_{x_1x_1} \cdots f_{x_1x_n} \\ \vdots \quad \ddots \quad \vdots \\ f_{x_nx_1} \cdots f_{x_nx_n} \end{pmatrix}$

  $(3x_1^2 - 3x_2, 3x_2^2 - 3x_1)$, $\begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}$

  $\boldsymbol{\delta}^{\mathrm{T}} = (\delta_1, \ldots, \delta_n)$.

- $\tilde{\boldsymbol{x}}^{\mathrm{T}} = (2,2)$ とすれば、$\tilde{\boldsymbol{f}}'^{\mathrm{T}} = (6,6), \boldsymbol{H} = \begin{pmatrix} 12 & -3 \\ -3 & 12 \end{pmatrix}$,

  $\tilde{\boldsymbol{f}}'^{\mathrm{T}} + \boldsymbol{H}(\tilde{x})\boldsymbol{\delta} = 0$ を解いて、$\boldsymbol{\delta}^{\mathrm{T}} = (-\frac{2}{3}, -\frac{2}{3})$.

  1回目の更新で $\boldsymbol{x}^{\mathrm{T}} = (1\frac{1}{3}, 1\frac{1}{3})$ となる.

演習問題

課題1: $x^2 = 2$ の解をニュートン法で求め、$\sqrt{2}$ の近似値を 4桁まで得るのに必要な反復回数を調べよ

課題2: 例題の続きを確認せよ

例題: $f(\boldsymbol{x}) = x_1^3 + x_2^3 - 3x_1x_2$ の極小値を求めよ

2次近似 $f_{\mathrm{II}}(\boldsymbol{x}) = \tilde{f} + \sum_j \tilde{f}_{x_j}\delta_j = \tilde{f} + \tilde{\boldsymbol{f}}'^{\mathrm{T}}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\delta}$

の与える漸化式: $\tilde{\boldsymbol{f}}'^{\mathrm{T}} + \boldsymbol{H}(\tilde{x})\boldsymbol{\delta} = 0, \quad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta},$

$$\tilde{\boldsymbol{f}}'^{\mathrm{T}} = (3x_1^2 - 3x_2, 3x_2^2 - 3x_1), \boldsymbol{H} = \begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}.$$

初期点: $\tilde{\boldsymbol{x}}^{\mathrm{T}} = (2, 2)$ →1回反復後: $\boldsymbol{x}^{\mathrm{T}} = (1\frac{1}{3}, 1\frac{1}{3})$

課題1: $x^2=2$ の解をニュートン法で求め、$\sqrt{2}$ の近似値を 4桁まで得るのに必要な反復回数を調べよ

関数 $f(x) = x^2-2$のゼロ点を求める問題なので、反復式は

$x_{k+1}=x_k-f(x_k)/f'(x_k)=x_k-(x^2-2)/(2x)$

例えば $x_0=1$のときと、$x_0=2$のときで$x_k\approx1.414$となる回数に違いはあるでしょうか。

課題2: 例題の続きを確認せよ

例題:$f(x)=x_1^3+x_2^3-3x_1x_2$の極小値を求めよ

2次近似 $f_{\mathrm{II}}(x)=f+\sum_j f_{xj}\delta_j=f+\boldsymbol{f}'^{\mathrm{T}}\boldsymbol{\delta}+(1/2)\boldsymbol{\delta}^{\mathrm{T}}H\boldsymbol{\delta}$

の与える漸化式: $\tilde{\boldsymbol{f}}'^{\mathrm{T}} + \boldsymbol{H}(\tilde{x})\boldsymbol{\delta} = 0, \quad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta},$

例題: $\tilde{\boldsymbol{f}}'^{\mathrm{T}}=(3x_1^2 - 3x_2, 3x_2^2 - 3x_1), \boldsymbol{H}=\begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}.$

初期値を$\tilde{\boldsymbol{x}}^{\mathrm{T}} = (2, 2)$とすれば、1回の更新で $\boldsymbol{x}^{\mathrm{T}} = (1\frac{1}{3}, 1\frac{1}{3})$

Scilabを使って実際に反復計算してみましょう。

Scilab, A Free Scientific Software Package
Scilab は工学・理学での応用に役立つ強力でオープンな計算環境を提供する数値計算のためのソフトウェアパッケージです。Scilabには何百もの数学関数が含まれ、加えて、様々なプログラミング言語(C, C++, Fortran・・・)で作成されたプログラムを必要に応じて取り込む機能を備えています。洗練されたデータ構造(リスト、多項式、有理関数、線形システム・・・)を扱うことができる仕組みと、インタプリタ、高度なプログラミング言語を備えています。

Scilabホームページ (http://www.scilab.org)から抜粋・直訳

よく似たソフトウェアに MATLAB や octave などがあります。scilab や octaveには、商用で非常に広く使われている MATLAB に対するフリー・オープンな代替品としての側面があります。ただし、スクリプト言語の仕様は完全に互換ではありません。また、scilabは積極的に独自の機能・仕様を採用し、MATLABからは徐々に離れつつあると言われています。その一方で octave はユーザインタフェースを含めてそっくりになるモードを備える等、MATLAB 互換路線を継続するようです。
興味があれば octave も試してみてください。

1、Scilab練習

起動と終了

起動（linuxでは端末からwindowsはメニューから）
linux: /usr/local/scilab-5.2.0/bin/scilab [Enter]
windows: スタートメニューから起動項目を探してください
※コマンドを入力できるコンソール画面が開きます。

終了（コマンドコンソールから）
quit [Enter]

基本操作

直接コマンドを入力すれば複素数や行列を扱うことのできる計算機として動作します。入力履歴の表示や編集も可能です。

計算

1＋2[Enter]
3＊3[Enter]
2 ^ 0.5[Enter]
※[Enter]キーを押すと式の評価値が表示されます。

変数

X=4.2[Enter]

※アルファベットで始まりアルファベット/数字/アンダースコアからなる変数名が使えます。

定数

%pi[Enter]

%e[Enter]

複素数

1＋%i*2[Enter]

Z=2+%i*3[Enter]

Z*Z[Enter]

%i ^ 0.5[Enter]

行列

[1 2 ; 3 4][Enter]

%eye(2,2)[Enter]

※列の区切りは空白またはカンマ(,)

変数

X=4.2[Enter]

※アルファベットで始まりアルファベット/数字/アンダースコアからなる変数名が使えます。

※Scilabの標準関数等の名前では大文字（頭文字）から始まる名前を使わないので、プログラムで用いる名前を大文字から始まるようにすれば、衝突を防ぐことができる。

定数

%pi[Enter]

%e[Enter]

複素数

1＋%i*2[Enter]

Z=2+%i*3[Enter]

Z*Z[Enter]

%i ^ 0.5[Enter]

行列

[1 2 ; 3 4][Enter]　　%eye(2,2)[Enter]
%zeros(2,2)[Enter]　　%ones(2,2)[Enter]
A=[1 2 ; 3 4][Enter]　　A'[Enter]
A'*A[Enter]　　%eye/A

※列の区切りは空白またはカンマ(,)

行列の演算

A=[1 2; 3 4][Enter]
B=A'[Enter]　　エルミート転置(共役転置)行列
A.*B[Enter]　　行列の要素同士の積
A./B[Enter]　　行列の要素同士の商
A+%i*B[Enter]
(A+%i*B)'[Enter]

※四則演算、割り算を使った連立方程式の解法・逆行列の計算等ができる。
※内部処理では全ての数値は複素数要素の行列として扱います。
※A'はAのエルミート転置です。実転置はA.'と表現します。

便利な機能

継続行

... を行末に置く

A=[1 2 3;...

4 5 6][Enter]

等間隔ベクトル

初期値:増加量:終値

A=1:0.5:3[Enter]

zeros()、ones()、eye()

引数が複数の整数：n1×n2×…行列、

引数が行列：引数と同じ大きさの行列、

A=zeros(2,4)[Enter]　　B=ones([1:4])[Enter]

C=eye(A' B')[Enter]

便利な機能

継続行

... を行末に置く

A=[1 2 3;...

4 5 6][Enter]

等間隔ベクトル

初期値:増加量:終値

A=1:0.5:3[Enter]

zeros()、ones()、eye()

引数が複数の整数：n1×n2×…行列、

引数が行列：引数と同じ大きさの行列、

A=zeros(2,4)[Enter]　　B=ones([1:4])[Enter]

C=eye(A' B')[Enter]

デリミタ「;」について

行末にセミコロンを置くと、結果出力が抑制されます。
プログラム中では計算結果を表示する必要がある場合以外は「;」を忘れずに。

A=zeros(2,4); [Enter]

※セミコロンを着けなかった場合と比較してください。

関数定義

```
function [x,y]=myfct(a,b)
x=a+b
y=a-b
endfunction
```

使用例

myfct(1,2)**[Enter]**

myfct([1 2; 3 4],[1 2; 3 4]') [Enter]

※別ファイルで定義し、getf()で読み込むこともできる。

getf('functiondeginition.sci') [Enter]

グラフィックス

2次元グラフ

```
t=[0:0.05:1]'[Enter]
ct=cos(2*%pi*t)[Enter]
plot2d(t,ct)[Enter]
square(-0.5,-1.5,1.5,1.5)[Enter]
tht=2*%pi*t[Enter]
plot2d(tht,ct)[Enter]
square(-%pi/2,-1.5,2.5*%pi,1.5)[Enter]
```

3次元グラフ

```
x1 = [-1:0.1:1];
x2 = [-1:0.1:1];
[xx,yy] = meshgrid(x,y);
plot3d(x,y,xx.^3+yy.^3-3*xx.*yy);
```

課題1: $x^2=2$ の解をニュートン法で求め、$\sqrt{2}$ の近似値を 4桁まで得るのに必要な反復回数を調べよ

関数 $f(x) = x^2-2$のゼロ点を求める問題なので、反復式は

$$x_{k+1}=x_k-f(x_k)/f'(x_k)=x_k-(x^2-2)/(2x)$$

例えば $x_0=1$のときと、$x_0=2$のときで$x_k\approx1.414$となる回数に違いはあるでしょうか。

例:

```
x = 1;
while abs(x*x -2) > 10^(-4)
  x = x-(x*x-2)/(2*x)
end
```

※回数表示は？

課題2: 例題の続きを確認せよ

例題:$f(x)=x_1^3+x_2^3-3x_1x_2$の極小値を求めよ

2次近似 $f_{II}(x)=f+\sum_j f_{xj}\delta_j=f+\boldsymbol{f}'^{\mathrm{T}}\boldsymbol{\delta}+(1/2)\boldsymbol{\delta}^{\mathrm{T}}H\boldsymbol{\delta}$

の与える漸化式: $\tilde{\boldsymbol{f}}'^{\mathrm{T}} + \boldsymbol{H}(\tilde{x})\boldsymbol{\delta} = 0, \quad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta},$

例題: $\tilde{\boldsymbol{f}}'^{\mathrm{T}}=(3x_1^2 - 3x_2, 3x_2^2 - 3x_1), \boldsymbol{H}=\begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}.$

初期値を$\tilde{\boldsymbol{x}}^{\mathrm{T}} = (2, 2)$とすれば、1回の更新で $\boldsymbol{x}^{\mathrm{T}} = (1\frac{1}{3}, 1\frac{1}{3})$

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

function dy=df(x)
  dy=[3*x(1)^2-3*x(2); 3*x(2)^2-3*x(1)];
endfunction

function h=H(x)
  h=[6*x(1), -3; -3, 6*x(2)];
endfunction
```

内点法：カーマーカー法

help karmarkar

Name karmarkar — karmarkar algorithm
Calling Sequence [x1]=karmarkar(a,b,c,x0)
Parameters
- a matrix (n,p)
- b n - vector
- c p - vector
- x0 initial vector
- eps threshold (default value : 1.d-5)
- gamma descent step 0<gamma<1, default value : 1/4
- x1 solution
- crit value of c'*x1

Description
Computes x which minimizes

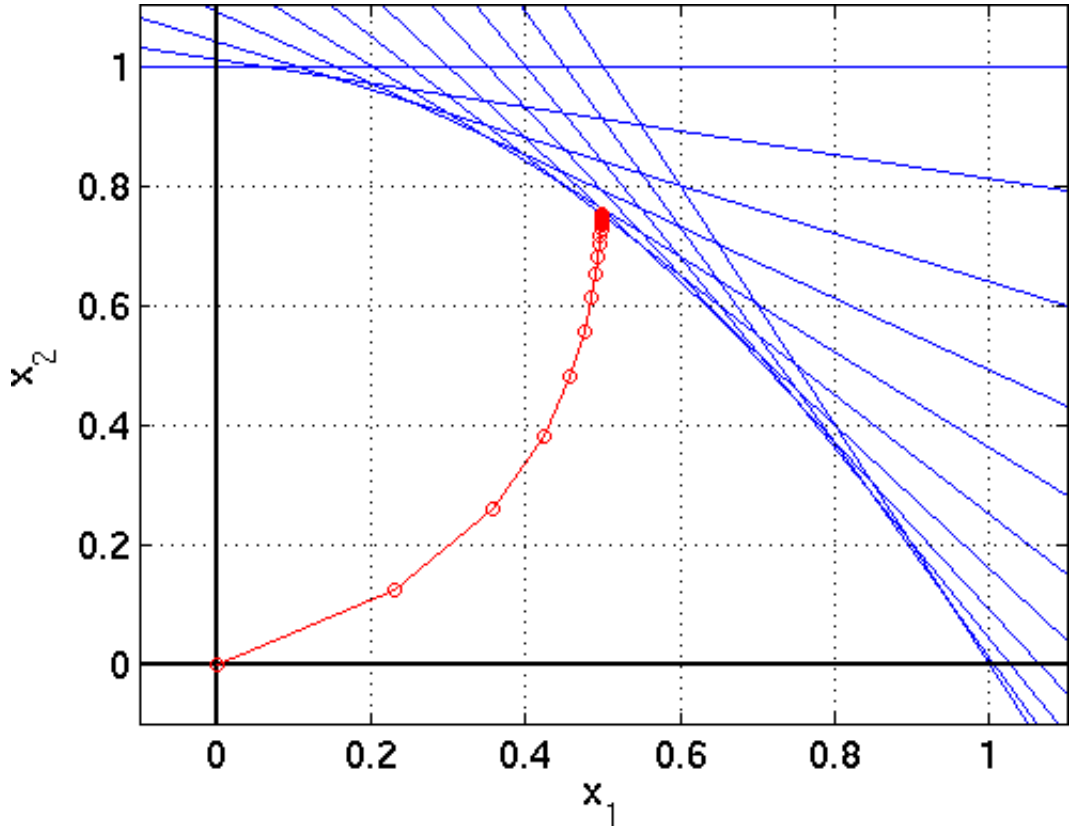
$\min c^{T}x$ with $ax=b,\ x \geq 0$

まず、ヘルプを見て、Examplesの例を実行してみてください。
次に、karmarkar関数呼び出し行の「;」を除いてみてください。

カーマーカーのアルゴリズム

カーマーカーのアルゴリズム（英: Karmarkar's algorithm）とは1984年、ナレンドラ・カーマーカーにより発見された線形計画問題の解法である。このアルゴリズムは、しばしば、カーマーカー法（Karmarkar's method）とも呼ばれる。また、本アルゴリズムを発明とする特許が米国や日本で出願されたため、請求特許は時折カーマーカー特許（Karmarkar's patent）とも呼称される。

wikipediaより



Scilab5.3の途中経過表示機能を利用する

```
function stop=myoutputfunction(xopt, optimValues, state)
    localmsg = gettext("Iter#%3.0f: %s (%s), "+..
        "f=%10.3e, x=[%s], gap=%10.3e\n")
    xstr = strcat(msprintf("%10.3e\n",xopt)'," ")
    mprintf(localmsg,optimValues.iteration,state,optimValues.procedure,..
        optimValues.fval,xstr,optimValues.dualgap)
    plot(xopt(1),xopt(2),"bo")
    stop = %f
endfunction
```

例：

```
n=11;
A = [2*linspace(0,1,n)',ones(n,1)];
b = 1 + linspace(0,1,n)'.^2;
c=[-1;-1];
// Plot the constraints
scf();
for i = 1 : n
  plot(linspace(0,1,100),b(i)-A(i,1)*linspace(0,1,100),"b-")
end
xopt=karmarkar([],[],c,[],[],[],[],myoutputfunction,A,b); //scilab 5.3 用
// Plot the starting and ending points
plot(xopt(1),xopt(2),"k*")
```

練習問題15

課題1: $x^2 = 2$ の解をニュートン法で求め、$\sqrt{2}$ の近似値を 10桁まで得るのに必要な反復回数を調べよ。結果とともに回数を表示するプログラムを作成せよ。

ヒント:

```
x = 1;
while abs(x*x -2) > 10^(-4)
  x = x-(x*x-2)/(2*x)
end
```