計算科学特論

有限要素法演習

Linux環境/Windows環境にログインして、
授業情報ウェブサイトにアクセスしてください

URL: http://comp.cs.ehime-u.ac.jp/
~okano/computation/

今日の予定

今日のテーマ:

Laplace方程式の境界値問題に有限要素法を適用し、得られた近似解を差分法の近似解や厳密解と比較して有限要素法の特徴を知ります。

具体的には、Scilab (サイラボ)のプログラムで差分法演習と同様の問題を解きます。

今日の予定

サンプルプログラム解説

サンプルプログラムを変更して
差分法との比較から有限要素法の特長を考える
厳密解が判る場合を利用して誤差を評価する

サンプルプログラム解説

授業資料とサンプルプログラム

授業情報ウェブサイトからダウンロードしてください。

http://comp.cs.ehime-u.ac.jp/~okano/computation/

授業資料の大半はサンプルプログラムリストです。

サンプルプログラムは2種類あります。
Linux用：　　fem-linux.sci
Windows用：　fem-windows.sci
※コメントと改行コードが違うだけです。

サンプルプログラム(fem-*.sci)

```
funcprot(0);
//stacksize('max');
```

ファイル実行時の警告を抑制する

利用可能なスタックサイズを最大化する
(ver. 5.5以前のscilabで必要)

関数定義、以降のコメント部分の例として考えてください。

```
function [v,bv,uv,fe,np] = SetupSquare(z1,z2,xn,yn);
// SetupSquare 矩形領域の問題に必要な設定
// z1 z2 領域の対角頂点、xn yn x方向・y方向の分割数
// (格子点数=(xn+1)×(yn+1)に注意)
// v 格子点座標、bv 境界格子点番号、uv 未知格子点番号
// fe 三角形有限要素の格子点番号、np 格子点を含む有限要素番号
```

宣言部

機能説明

入力要素説明

出力要素説明

～途中省略～

```
endfunction
```

サンプルプログラム（ fem-*.sci ）

```
funcprot(0);

//stacksize('max');
```

ファイル実行時の警告を抑制する

利用可能なスタックサイズを最大化する
（ver. 5.5以前のscilabで必要）

関数

```
func
//  S
//  Z
//
//  v
//  t
```

説明

説明

番号

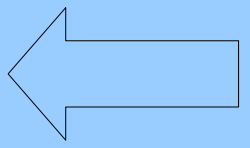
```
endfunction
```

Scilab ver. 6.0以前を利用する場合の注意事項:

第6版以前と以降でメモリ割り当ての方式が変更になっています。
ver. 表記が6未満（5.5がインストールされていることがあります。）の場合に
ではスタックサイズを指定しないと容量が足りなくなるかもしれません。
その際は stacksize 関数を使用してください。

※ 今日のサンプルプログラムであれば、右例から左例に変更してください。

```
funcprot(0);
stacksize('max');
```



```
funcprot(0);
//stacksize('max');
```

```
function [v,bv,uv,fe,np] = SetupSquare(z1,z2,xn,yn);
// SetupSquare 矩形領域の問題に必要な設定
〜
// v 頂点番号別座標(複素表現)
〜
// bv(j) j番目の境界頂点の頂点番号
〜
// uv(j) j番目の未知頂点(=境界じゃない頂点)の頂点番号
〜
// fe(k,:) k番目の(三角形)有限要素の 3つの頂点の頂点番号
〜
// np(k,1) k番目の頂点を含む有限要素の数
〜
// np(k,2〜np(k,1)+1) k番目の頂点を含むnp(k,1)個の有限要素
〜
endfunction
```

問題設定に応じた変数定義が主な機能です。

関係する全ての頂点座標を格納します。

境界上の頂点をvの番号で格納します。

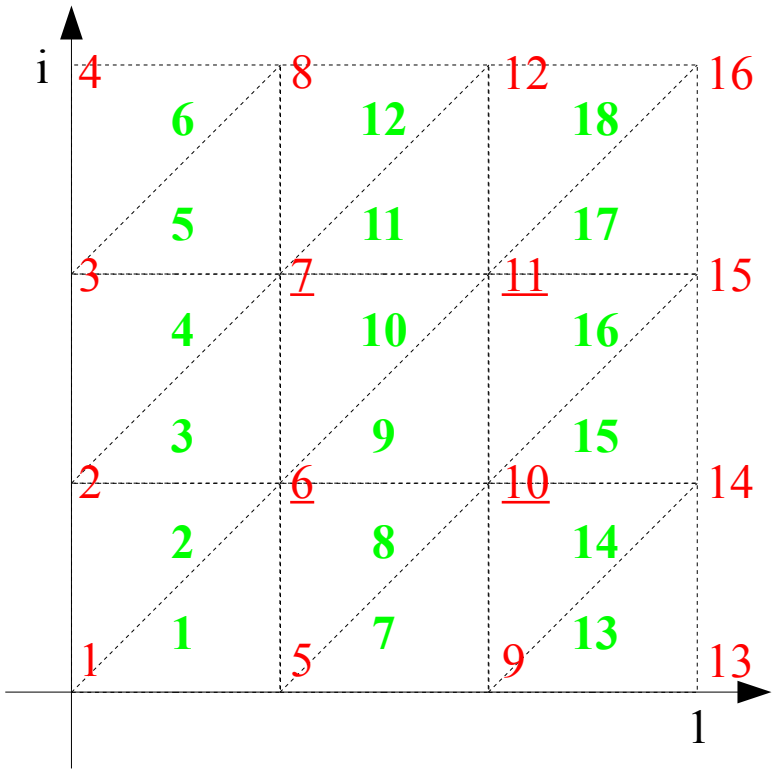
境界以外の頂点はこちら

有限要素を構成する頂点

np()はfe()の逆向きのインデックスになります

```
// メインルーチン
[v,bv,uv,fe,np] = SetupSquare(0,1+%i,3,3);
～以下略
```

0と1+iを結ぶ直線を対角線とする正方形を3×3に分割して有限要素を配置します。



赤字 は頂点番号
下線 は未知頂点
緑字 は有限要素番号

v: 頂点座標;
例では 0,i/3,2i/3,i,...,1+iの値を持つベクトル。
vに引数として頂点番号を与えて複素座標を得る。

bv: 境界頂点(番号);
例では1,2,3,4,5,8,9,12,13,14,15,16の値を持つベクトル。
uv: 未知頂点(番号);
例では 6,7,10,11 の値を持つベクトル。
bv(j),uv(j)で頂点番号、v を介して複素座標を得る。

fe: 有限要素毎の頂点番号;
各行に有限要素毎の3つの頂点番号を持つ行列。
例では [1,5,6;1,7,2;...;11,16,12]の値を持つ。

np: 頂点を含む有限要素の数と番号;
各行1列に頂点を含む有限要素数、2列以降に有限要素番号を持つ行列。例では[2,1,2,0,...;3,2,3,4,...]

```
～中略
bn = size(bv,2); // 境界頂点の数
ub = zeros(1,bn); // 境界値
for i = 1:bn
  ub(i) = Bfunc(v(bv(i))); // 境界条件の設定
end
```

メインルーチンの境界条件設定箇所。
境界頂点毎に境界値を別関数 Bfunc で求めて代入しています。

```
function r = Bfunc(z);
if imag(z) == 0;
  r = real(z)*(1-real(z));
else
  r = 0;
end
endfunction
```

実際の境界関数を与える関数。
←例では複素座標の虚部=0の場合のみ、x(1-x)を値として返しています。

```
～中略
un = size(uv,2); // 未知頂点の数
uu = zeros(1,un); // 未知変数


A = zeros(un,un);
for j = 1:un
  for k = 1:un
    A(k,j) = Prod(uv(k),uv(j));
  end
end
```

メインルーチンの係数行列設定箇所。
未知頂点が基底関数に対応します。

連立方程式 AX=B の行列Aを代入します。
近似関数のj番目の基底関数にk番目の基底関数をかけて内積を取ります。実際の計算はProd()関数で行います。

```
function r = Prod(j,k);
// Prod 頂点 j と k 周りの基底関数による重み付残差(弱形式)
function r = Prod0(i,j,k);
// Prod0 頂点 j,k を基点とする基底関数の有限要素 i 上での積分
```

こちらの詳細は省略。

Prod の下請け関数

```
～中略
B = zeros(un,1);
for j = 1:bn
  for k = 1:un
    B(k) = B(k) - ub(j)*Prod(uv(k),bv(j));
  end
end
uu = A\B;
u = zeros(v); // 近似関数の頂点における値
for i = 1:size(bv,2)
  u(bv(i)) = ub(i);
end
for i = 1:size(uv,2)
  u(uv(i)) = uu(i);
end
```

メインルーチンの連立方程式右辺設定箇所。

境界条件から定まる部分を代入します。

連立方程式 AX=B を解けば未知頂点における近似値が解ります。

求まった未知頂点の近似値に境界値を併せて全頂点での値を u に代入します。
u(j) で頂点 j の値が参照できます。

fec
三角メッシュ上に定義された関数の擬似カラープロット
呼び出し手順
fec(x,y,triangles,func)

パラメータ
x,y　　大きさ nのベクトル, (x(i),y(i)) は ノード iの座標を定義します
func　　大きさ nのベクトル
func(i) は,擬似カラープロットを行う関数のノード i における値を定義します.
triangles　[:,5] 行列
trianglesの各行は,メッシュの三角形
triangle(j) = [number,node1,node2,node3,flag] を定義します.
node1,node2,node3 は三角形を構成するノードの番号です. number は三角形の番号で,flag は fec 関数では使用されない整数です.

レポート(12)

学籍番号・氏名を記し提出してください。

- Scilabによる有限要素法演習を実施して、サンプルスクリプトの問題設定方法を理解してください。

  プログラム中で解く連立一次方程式のサイズが、前回の差分法演習のサンプルプログラムと同程度になるようなパラメタ設定をして、その結果を比較してください。

授業レポート用紙：氏名（　　）学籍番号(　　)

2020年1月27日（月）

できれば授業の感想も書いてください。