FDM流れ解析　No.3
CG法で結果の対称性を保つ工夫

2007年6月

後　保範 (東京工芸大学)

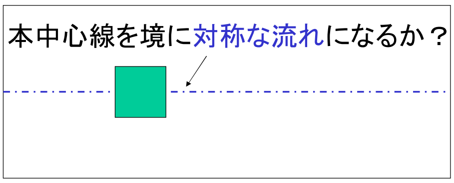
1

目次

2

1. 結果の対称性とは

- 「下記の様な角柱周りの流れ解析で、初期条件、境界条件がy軸の中心で対称なら流れも対称になるか？」が最終目標



3

1.1 CG法の解の対称性

流れ解析の圧力方程式

$\Delta p = -div(v \cdot \nabla)v + div(v)/\Delta t$

$\Downarrow$ 差分法で離散化

$Ax = b$；A, bとも領域のy軸で対称

$\Downarrow$ CG法で解xを計算

圧力解(x)はy軸で対称にできるか？

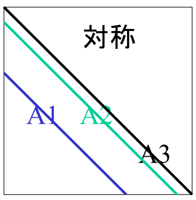
4

2. 対称性が崩れる項

- CG法は数学的には場の対称性を保つが、丸め誤差のため下記計算で対称性が崩壊

q=Apの計算

q(i,j) = A1(i,j)*p(i,j-1)
　+ A2(i,j)*p(i-1,j)
　+ A3(i,j)*p(i,j)
　+ A2(i+1,j)*p(i+1,j)
　+ A1(i,j+1)*p(i,j+1)



5

2.1 CG法プログラム

反復計算で$Ax = b$の解xを求める.
$p = r = b - Ax$,　　xは初期値
$\lambda_0 = (r, r)$
以下$\|r\|_2 \le \varepsilon$(収束条件)となるまで反復する.

$q = Ap$,　$\alpha = \lambda_0 / (p, q)$
$x = x + \alpha p$,　$r = r - \alpha q$
$\lambda_1 = (r, r)$,　$\beta = \lambda_1 / \lambda_0$
$p = r + \beta p$,　$\lambda_0 = \lambda_1$

6

3. 浮動小数点演算の丸め誤差

- 計算機で使用する浮動小数点(Cでは float,doubleで指示)演算では下記の計算順序の変更で結果が異なる場合がある。

実数(無限桁数)

$$(x_1+x_2)+x_3 \;=\; (x_3+x_2)+x_1$$

浮動小数点(有限桁数)

$$(x_1+x_2)+x_3 \;\neq\; (x_3+x_2)+x_1$$

7

3.1 結果が異なる例

- 10進3桁で四捨五入丸め方式の例を示す

$x_1=21.5,\quad x_2=1.24,\quad x_3=0.236$

(1) $(x_1+x_2)+x_3$の計算

$x_1+x_2=21.5+1.24=22.7 \leftarrow 22.74$

$(x_1+x_2)+x_3=22.7+0.236=22.9 \leftarrow 22.936$

(2) $(x_3+x_2)+x_1$の計算

$x_3+x_2=0.236+1.24=1.48 \leftarrow 1.476$

$(x_3+x_2)+x_1=1.48+21.5=23.0 \leftarrow 22.98$

不一致

8

4. 対称性が崩れる原因

- 係数行列Aとベクトルpは場の対称性のため下記の対応関係がある。

H_5

H_2 — H_3 — H_4

H_1 (i,ny-j)=(i,k)

L_5

L_2 — L_3 — L_4

L_1 (i,j)

格子L_3とH_3のq計算するAとpの等号関係

H_1格子= L_5格子

H_2格子= L_2格子

H_3格子= L_3格子

H_4格子= L_4格子

H_5格子= L_1格子

9

4.1 係数行列の等号関係

- H_1格子とL_5格子：(k = ny - j)
  A1(i,k) = A1(i,j+1) ,　　p(i,k-1) = p(i,j+1)
- H_2格子とL_2格子
  A2(i,k) = A2(i,j),　　p(i-1,k) = p(I-1,j)
- H_3格子とL_3格子
  A3(i,k) = A3(i,k),　　p(i,k) = p(i,j)
- H_4格子とL_4格子
  A2(i+1,k) = A2(i+1,j),　　p(i+1,k) = p(i+1,j)
- H_5格子とL_1格子
  A1(i,k+1) = A1(i,j),　　p(i,k+1) = p(i,j-1)

10

4.2 数学的に等価な計算

- L_3格子(i,j)点とH_3格子(i,ny-j)=(i,k)点では下記の計算は数学的に同じである。
- L_3格子(i,j)点

q(i,j) = A1(i,j)*p(i,j-1) + A2(i,j)*p(i+1,j) + A3(i,j)*p(i,j)
　+ A2(i+1,j)*p(i,j+1) + A1(i,j+1)*p(i,j+1)

- H_3格子(i,k)点

q(i,k) = A1(i,k)*p(i,k-1) + A2(i,k)*p(i+1,k) + A3(i,k)*p(i,k)
　+ A2(i+1,k)*p(i,k+1) + A1(i,k+1)*p(i,k+1)

11

4.3 q(i,j)=q(i,k)が成立しない原因

- 下記のq(i,j)とq(i,k)は数学的に同じであるが、丸め誤差のため値が異なる場合がある。
- L_3格子(i,j)点

q(i,j) = A1(i,j)*p(i,j-1) + A2(i,j)*p(i+1,j) + A3(i,j)*p(i,j)
　+ A2(i+1,j)*p(i,j+1) + A1(i,j+1)*p(i,j+1)

- H_3格子(i,k)点

q(i,k) = A1(i,k)*p(i,k-1) + A2(i,k)*p(i+1,k) + A3(i,k)*p(i,k)
　+ A2(i+1,k)*p(i,k+1) + A1(i,k+1)*p(i,k+1)

12

5. 丸め誤差を同一にする工夫

- 2項の加減算の丸め誤差は計算順序に依存しないことを利用。( )を付けて計算順序を固定
- L_3格子(i,j)点

q(i,j) = (A3(i,j)*p(i,j) + (A1(i,j)*p(i,j-1) + A1(i,j+1)*p(i,j+1)))
+ (A2(i,j)*p(i+1,j) + A2(i+1,j)*p(i,j+1))

- H_3格子(i,k)点 (L_3格子と同一になる)

q(i,k) = (A3(i,k)*p(i,k) +(A1(i,k)*p(i,k-1)+A1(i,k+1)*p(i,k+1)))
+ (A2(i,k)*p(i+1,k) + A2(i+1,k)*p(i,k+1))

13

5.1 ループを含め同一にする工夫

- 下記の様に()を付けて対応するものをペアーとし、計算順序を固定する。

```
do j=1,ny- 1
  do i=1,nx-1
    q(i,j) = (A3(i,j)*p(i,j)
           + (A1(i,j)*p(i,j-1) + A1(i,j+1)*p(i,j+1) )
           + (A2(i,j)*p(i-1,j) + A2(i+1,j)*p(i+1,j) )
```

14

6. CG法での計算結果

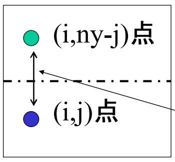
- 2次元CG法 (対称性保持工夫版)

Un-symmetric solution points = 0

k=1.0, f=100.0

-div(k.grad(u)) = f

10×10分割で比較



解が一致するか？

15

6.1 CG法での計算結果(未工夫)

- 2次元CG法 (対称性保持の工夫なし)

Un-symmetric solution points = 7

i	j	X(i,j)	X(i,NY-j)
1	1	3FF4803EBC75E90E	3FF4803EBC75E90D
2	1	4000803EBC75E90E	4000803EBC75E90F
4	1	40066648496100C4	40066648496100C5
1	2	4000803EBC75E90E	4000803EBC75E90F
3	2	40112F66F7F2AB28	40112F66F7F2AB29
4	2	40131BE1958B67EC	40131BE1958B67ED
4	3	4018533208EF2E81	4018533208EF2E82

16