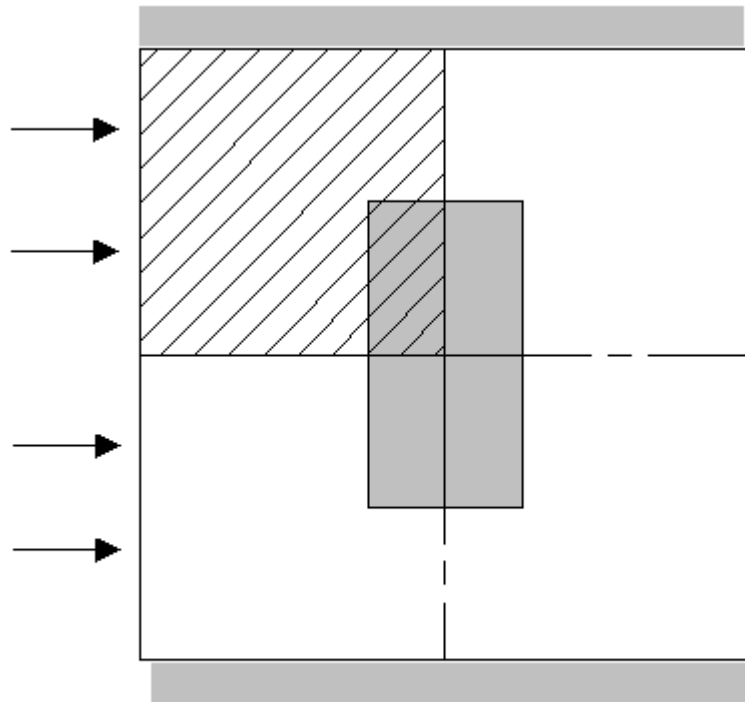
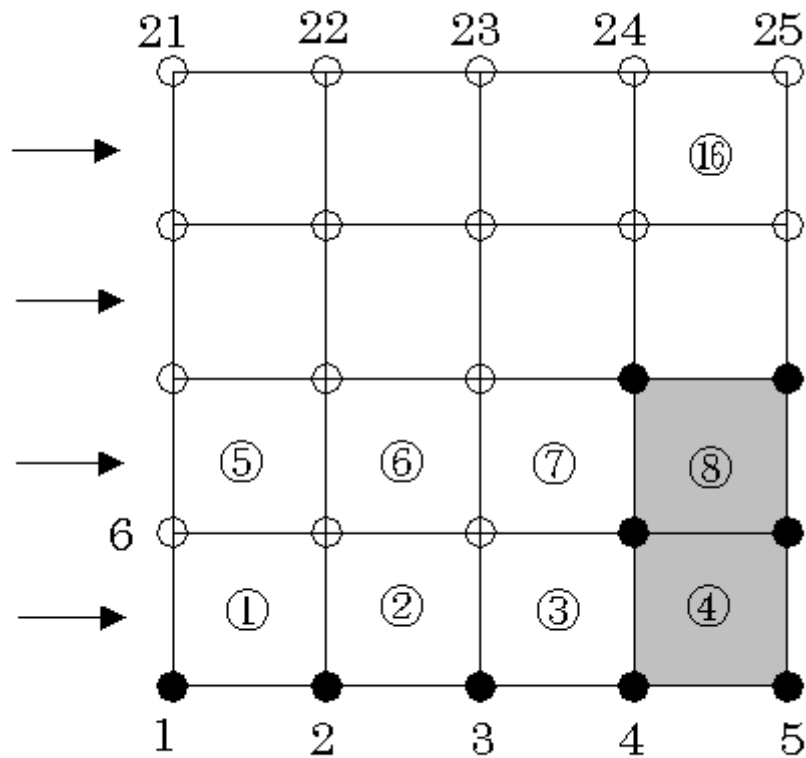


2次元ポテンシャル流れの有限要素法による数値解法



(a)



(b)

四角の障害物のある流れの流線を、もっとも原始的で短いプログラムでもとめてみよう。上図の(a)のように、上下の壁の間にはさまれて右方向へゆっくり流れるポテンシャル流れがあり、中央に灰色でしめした四角の障害物があるものとしよう。対称性から、上左の斜線をほどこした $1/4$ の領域について、(b)のように、16個の正方形のメッシュにわけ、1から2までの節点番号をつける。計算のための、各要素にたいする4個の節点の順番は左下からはじまり、反時計周りに数えるものとする。例えば要素 7 では1, 2, 7, 6という具合に。

(1) 各要素番号と対応する4個の節点番号の関係を示すトポロジーテーブルを作成する。

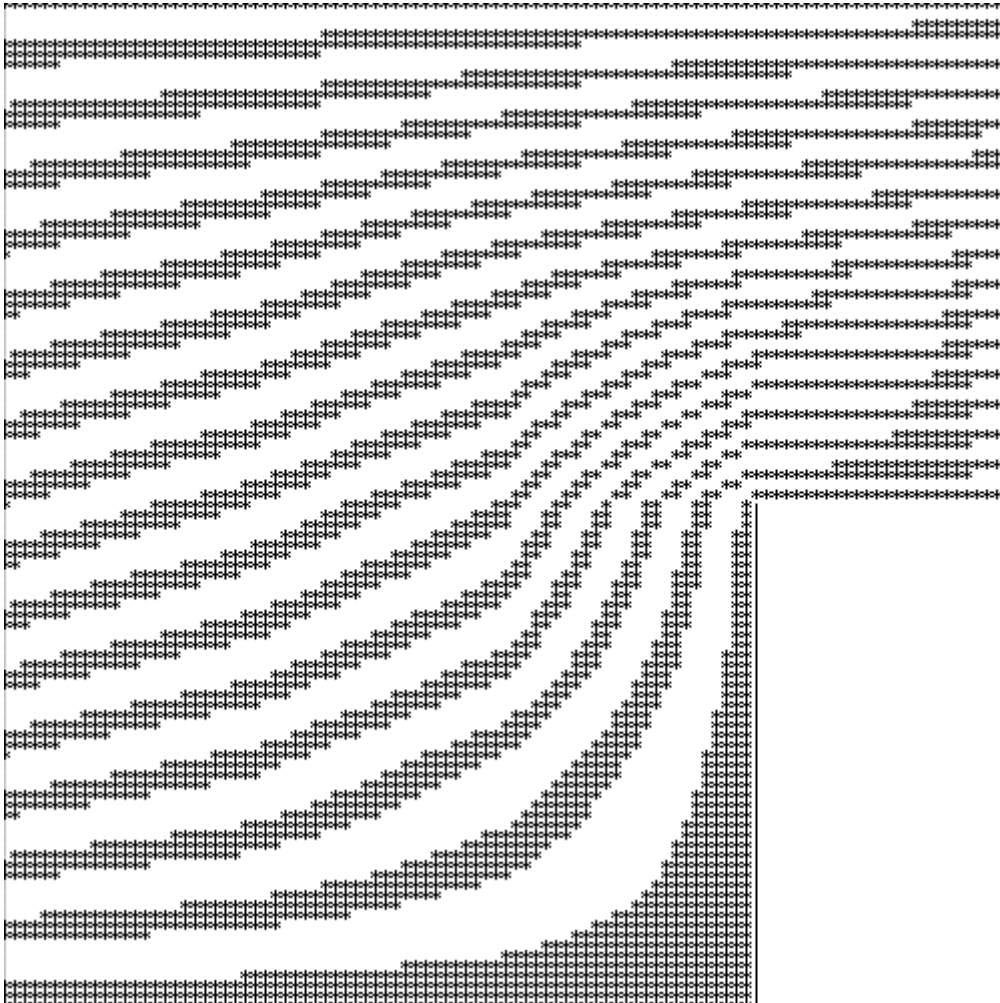
(2) 境界条件として、黒点をつけた節点の流れ関数の値を0とする。また節点21...25の流れ関数の値を4とする。節点6, 11, 16の流れ関数の値をそれぞれ1, 2, 3とする。

(3) ラプラスの微分方程式を数値計算するための要素剛性マトリクスを作成する。

(4) トポロジーテーブルを利用して全体剛性マトリクスを作成する。

- (5) 境界条件によって全体剛性マトリクスと荷重ベクトルを修正する。
- (6) 線形方程式を解く。
- (7) 求まった流れの関数の各節点の値をもとにして、流線を描く。

流線は次のようになる。 流れ関数の値から、各点の流速を求める方法などを工夫して下さい。



! Finite Element Method

!fem101.bas by T. Uede 00-8-12

set window 0,100,0,100

draw grid with scale (100,100)

option base 0

dim u(25),Kel(16,4,4),Top(16,4),Res(20,2),K(25,25),Kmod(25,25)

dim fmod(25),a(25,25),c(25)

for i=1 to 4 !-----topology

for j=1 to 4

let Top(i+4*(j-1),1)=i+4*(j-1)+j-1

let Top(i+4*(j-1),2)=i+4*(j-1)+j

let Top(i+4*(j-1),3)=i+4*(j-1)+j+5

let Top(i+4*(j-1),4)=i+4*(j-1)+j+4

next j

next i

for i=1 to 5 !-----boundary condition

let Res(i,1)=i

let Res(i,2)=0

next i

for i=1 to 5

let Res(i+5,1)=20+i

let Res(i+5,2)=4

next i

for i=1 to 3

let Res(i+10,1)=5*i+1

let Res(i+10,2)=i

next i

let Res(14,1)=9

let Res(14,2)=0

let Res(15,1)=10

let Res(15,2)=0

let Res(16,1)=14

let Res(16,2)=0

let Res(17,1)=15

let Res(17,2)=0

for ie=1 to 16 !-----element stiffness matrix

```

let Kel(ie,1,1)=2/6+2/6
let Kel(ie,1,2)=-2/6+1/6
let Kel(ie,1,3)=-1/6-1/6
let Kel(ie,1,4)=1/6-2/6

let Kel(ie,2,1)=Kel(ie,1,2)
let Kel(ie,2,2)=2/6+2/6
let Kel(ie,2,3)=1/6-2/6
let Kel(ie,2,4)=-1/6-1/6

let Kel(ie,3,1)=Kel(ie,1,3)
let Kel(ie,3,2)=Kel(ie,2,3)
let Kel(ie,3,3)=2/6+2/6
let Kel(ie,3,4)=-2/6+1/6

let Kel(ie,4,1)=Kel(ie,1,4)
let Kel(ie,4,2)=Kel(ie,2,4)
let Kel(ie,4,3)=Kel(ie,3,4)
let Kel(ie,4,4)=2/6+2/6
next ie

for ie=1 to 16 !-----global stiffness matrix
  for i=1 to 4
    for j=1 to 4
      let K(Top(ie,i),Top(ie,j))=K(Top(ie,i),Top(ie,j))+Kel(ie,i,j)
    next j
  next i
next ie

for i=1 to 25 !-----modifying the matrix
  for j=1 to 25
    let Kmod(i,j)=K(i,j)
  next j
next i
for i=1 to 25
  for j=1 to 25

```

```

    for ires=1 to 17
        if i=Res(ires,1) then
            let Kmod(i,j)=0
            let Kmod(j,i)=0
            let Kmod(i,i)=1
        end if
    next ires
next j
next i
for i=1 to 25  !-----modifying the force vector
    let fmod(i)=0
next i
for i=1 to 25
    for ires=1 to 17
        let fmod(i)=fmod(i)-K(i,Res(ires,1))*Res(ires,2)
        let fmod(Res(ires,1))=Res(ires,2)
    next ires
next i
for i=1 to 25  !-----solving the linear equations
    let c(i)=fmod(i)
    for j=1 to 25
        let a(i,j)=Kmod(i,j)
    next j
next i
for j=1 to 25
    for i=j+1 to 25
        let pp=a(i,j)/a(j,j)
        let c(i)=c(i)-c(j)*pp
        for ind=j to 24
            let a(i,ind)=a(i,ind)-a(j,ind)*pp
        next ind
    next i
next j
let u(25)=c(25)/a(25,25)
for j=24 to 1 step -1
    let s=0

```

```

for i=j+1 to 24
  let s=s+a(j,i)*u(i)
next i
let u(j)=(c(j)-s)/a(j,j)
next j

for x=0 to 100 !-----drawing the flow line
  for y=0 to 100
    for i=1 to 4
      for j=1 to 4
        if x>=25*(i-1) and x<=25*i then
          if y>=25*(j-1) and y<=25*j then
            let xx=x-25*(i-1)
            let yy=y-25*(j-1)
            let ie=i+4*(j-1)
            let uu=(1-xx/25)*(1-yy/25)*u(Top(ie,1))
            let uu=uu+xx/25*(1-yy/25)*u(Top(ie,2))
            let uu=uu+xx*yy/25/25*u(Top(ie,3))
            let uu=uu+(1-xx/25)*yy/25*u(Top(ie,4))
          end if
        end if
      next j
    next i
  next y

  for i=1 to 5 step 0.25
    if uu>=i-1 and uu<i-1+0.1 then
      if x<75 or y>50 then
        plot points: x,y
      end if
    end if
  next i
  let uu=0
next y
next x
plot lines :0,0;75,0;75,50;75,50
END

```