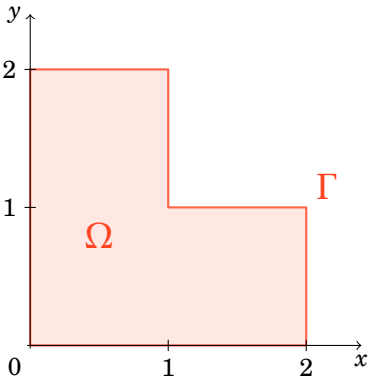
Discrétisation aux différences finies. Calcul numérique

Il s'agit de résoudre le problème suivant :

$$\boxed{\begin{cases} -\triangle(u) = f \text{ dans } \Omega \\ u = g \text{ sur } \Gamma \end{cases}}$$



i.e., pour le problème plan considéré ici,

$$\begin{cases} \forall (x,y) \in \Omega,\ \dfrac{\partial^2 u}{\partial x^2}(x,y) + \dfrac{\partial^2 u}{\partial y^2}(x,y) = -f(x,y) \\ \forall (x,y) \in \Gamma,\ u(x,y) = g(x,y) \end{cases}$$

1° On cherche à évaluer u sur les nœuds du quadrillage, qui permet de discrétiser la géométrie. On discrétise alors le problème continu, de sorte à faire apparaître le laplacien et donc en obtenir une approximation discrète. En supposant u suffisamment différentiable et régulière, la formule de Taylor-Young donne :

$$\begin{aligned}
u(x+h,y) &= u(x,y) + h\frac{\partial u}{\partial x}(x,y) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}(x,y) + \mathrm{O}(h^4) \\
u(x-h,y) &= u(x,y) - h\frac{\partial u}{\partial x}(x,y) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x,y) - \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}(x,y) + \mathrm{O}(h^4) \\
u(x,y+h) &= u(x,y) + h\frac{\partial u}{\partial y}(x,y) + \frac{h^2}{2}\frac{\partial^2 u}{\partial y^2}(x,y) + \frac{h^3}{6}\frac{\partial^3 u}{\partial y^3}(x,y) + \mathrm{O}(h^4) \\
u(x,y+h) &= u(x,y) - h\frac{\partial u}{\partial y}(x,y) + \frac{h^2}{2}\frac{\partial^2 u}{\partial y^2}(x,y) - \frac{h^3}{6}\frac{\partial^3 u}{\partial y^3}(x,y) + \mathrm{O}(h^4)
\end{aligned}$$

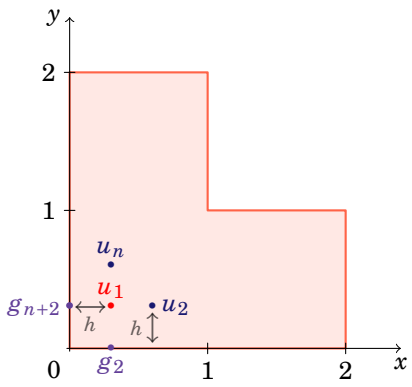
On a alors :

$$u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) = 4u(x,y) + h^2 \underbrace{\left(\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y)\right)}_{\triangle(u)(x,y)} + \mathrm{O}(h^4)$$

On obtient $\triangle(u)(x,y) = \dfrac{1}{h^2}[u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) - 4u(x,y)] + \mathrm{O}(h^2)$, ce qui donne, en reprenant le problème sur Ω, l'approximation d'ordre 2 suivante :

$$\boxed{4u(x,y) - u(x+h,y) - u(x-h,y) - u(x,y+h) - u(x,y-h) \simeq h^2 f(x,y)}$$

On maille la géométrie avec un pas constant $h := 1/n$, on calcule alors u à chaque nœud du maillage ainsi effectué de Ω, en suivant la méthode ci-dessus.

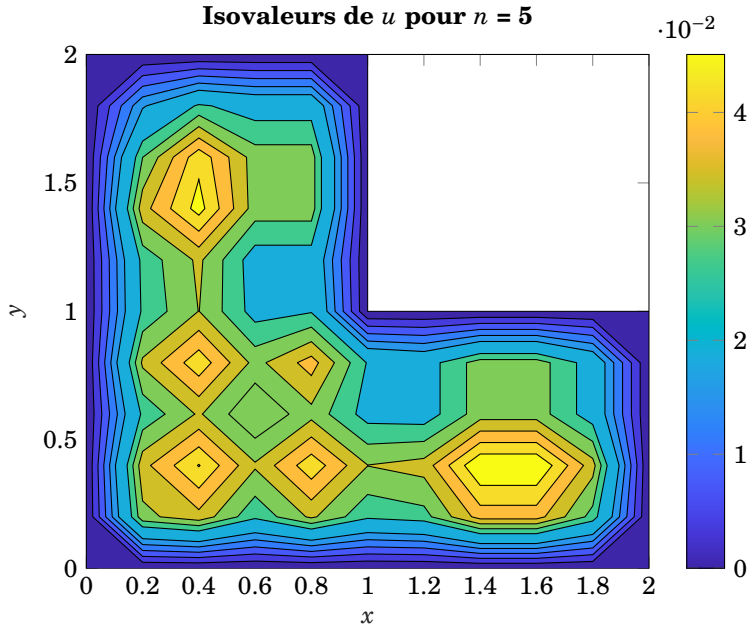


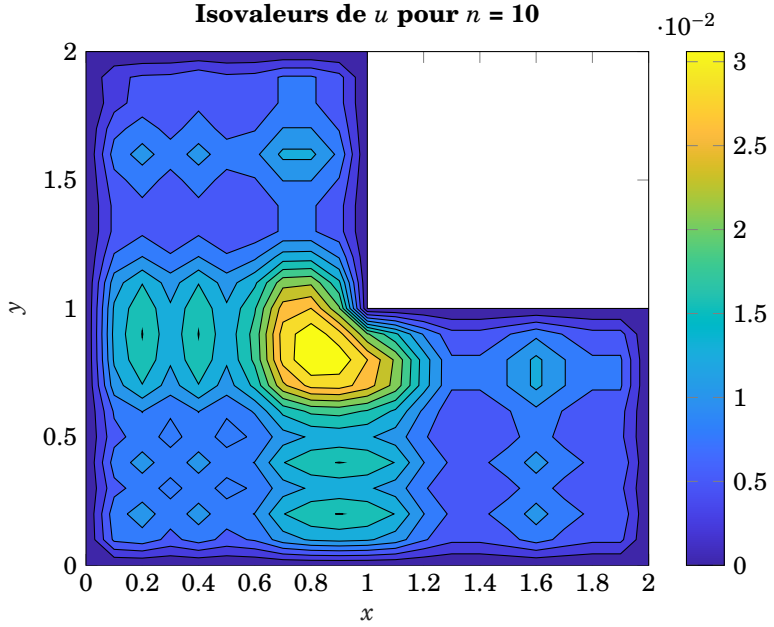
Comme illustré sur cette figure, on a la relation $4u_1 - u_2 - g_{n+2} - u_n - g_2 = h^2 f_1$. On considère alors la matrice A remplie, ligne par ligne, par les relations entre les u_i : par exemple ici, $A_{1,1} = 4$, $A_{1,2} = A_{1,n} = -1$.

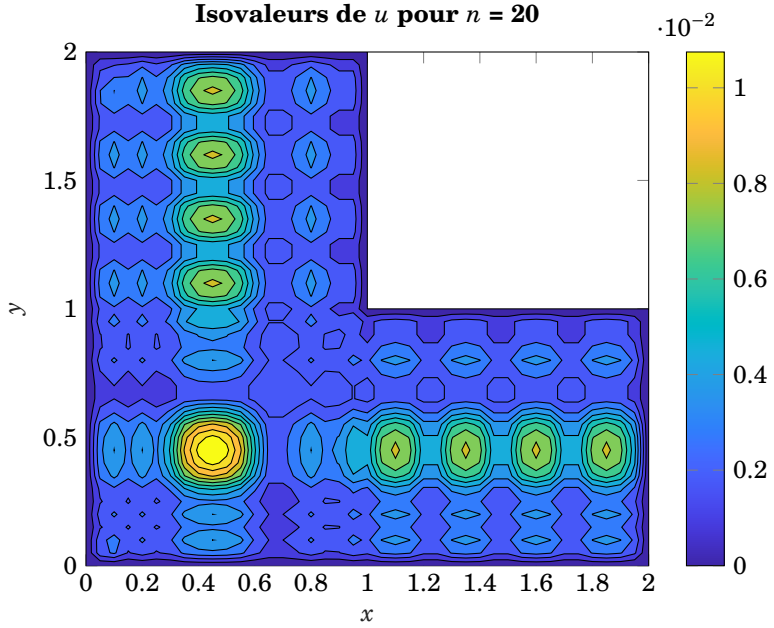
Quant au maillage, les côtés de longueur 2 sont divisés en $2n-1$ points et les côtés de longueur 1 en $n-1$ points, c'est ainsi que le nombre total de points pour mailler Ω vaut $(2n-1)(n-1) + n(n-1) = (3n-1)(n-1)$, nombre que l'on notera N.

En notant $U = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}$, $F = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}$, $G = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$ avec β_i nul si u_i a tous ses voisins dans Ω ou pouvant valoir la valeur du voisin (resp. la somme des valeurs des deux voisins) de u_i dans Γ, on a à résoudre le système linéaire $AU = h^2 F + G$, ce qui donne, sous réserve que A soit inversible, $U = A^{-1}\left(\frac{1}{n^2}F + G\right)$.

2° On considère ici $\begin{cases} \triangle(u)(x,y) = -1 \text{ dans } \Omega \\ u(x,y) = 0 \text{ sur } \Gamma \end{cases}$. Les courbes isovaleurs sont visibles ci-dessous :

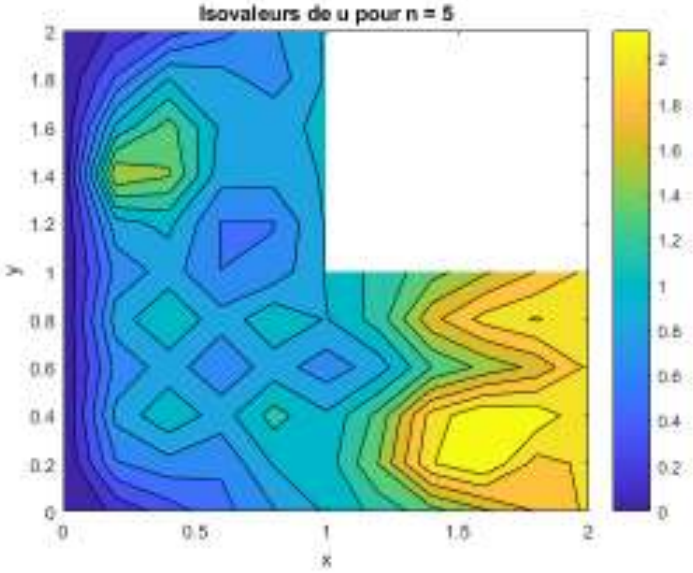


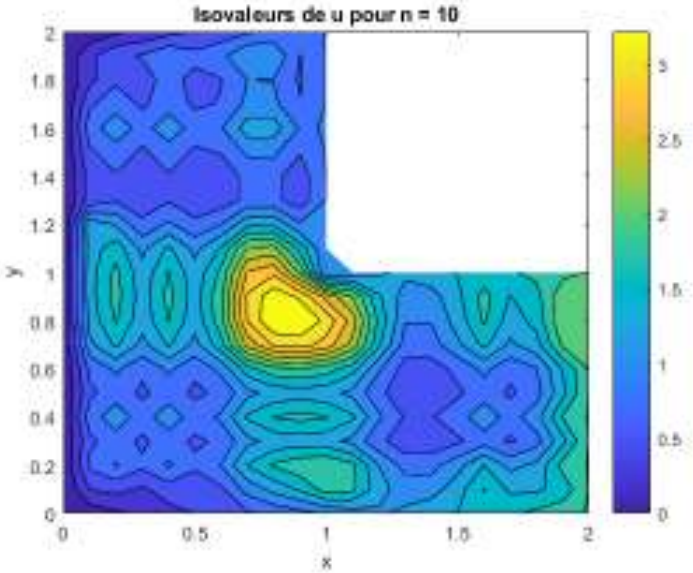


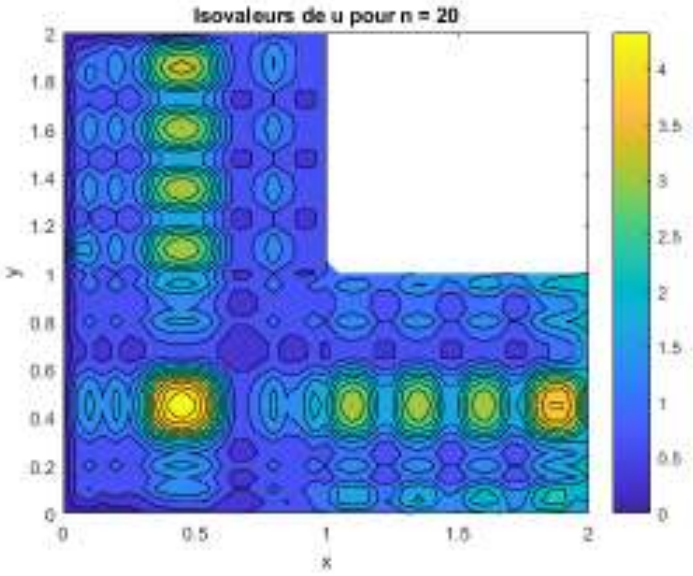


Le code est donné en annexe.

3° On considère ici $\begin{cases} \triangle(u)(x,y) = -\sup(x,y) \text{ dans } \Omega \\ u(x,y) = x \text{ sur } \Gamma \end{cases}$. $(x,y) \longmapsto \sup(x,y)$ est symétrique par rapport à la première bissectrice. Les courbes isovaleurs sont visibles ci-dessous :






Isovaleurs de u pour n = 20
y
x
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0.5
1.5
0.5
2.5
3
3.5
4

Annexe : programme informatique

Méthode :

— on a commencé par calculer les coordonnées (x, y) de chaque point du maillage dans l'ordre qui nous était imposé, grâce à la fonction `Discretisation` ;

— chaque point du maillage est numéroté, comme on l'a vu précédemment. Grâce à la fonction `matM`, on crée une matrice $M \in \mathcal{M}_{N,4}(\mathbb{N})$ qui contient l'indice des quatre voisins de chaque point. Chaque ligne i de M contient les quatre indices des voisins du point du maillage i ;

— la matrice A est créée grâce à la fonction `matA`. Sur chaque ligne de la matrice A, le coefficient $a_{i,j}$ prend la valeur -1 si j est dans la ligne i de M. On met évidemment des 4 sur la diagonale de A ;

— le script `scriptdessinq2.m` (resp. `scriptdessin3.m`) donne les résultats et affiche les isovaleurs de u pour la question 2 (resp. la question 3).

```
function [Px,Py] = Discretisation(n)
% Renvoie les coordonnées de chaque point du maillage dans l'ordre imposé.

Px=[];
Py=[];

Px=(1/n);
Py=(1/n);

c=0;

while c~=n-1
    for i=1:2*n-1
        Px((2*n-1)*c+i)=i*(1/n);
        Py((2*n-1)*c+i)=(c+1)*(1/n);
    end
    c=c+1;
end

while c~=2*n-1
    for i=1:(n-1)
        Px((2*n-1)*c+i)=i*(1/n);
        Py((2*n-1)*c+i)=(c+1)*(1/n);
    end
    c=c+1;
end

m=1;

while m==1
```

```
    for i = 1 : (3*n-1)*(n-1)
        if Px(i)==0
            Px(i)=[];
        elseif Py(i)==0
            Py(i)=[];
        end
    end
    if min(Px)~=0
        m=0;
    else
        m=1;
    end
end

end
```

```
function [M] = matM(X,Y,n)
% Renvoie les indices des voisins de chaque point du maillage.

h=1/n;
N=length(X);
M=zeros(N,4);
for k=1:N
    for i=1:N
        if i~k
            if X(i)==X(k)+h && Y(i)==Y(k)
                M(k,1)=i;
            elseif X(i)==X(k)-h && Y(i)==Y(k)
                M(k,2)=i;
            elseif Y(i)==Y(k)+h && X(i)==X(k)
                M(k,3)=i;
            elseif Y(i)==Y(k)-h && X(i)==X(k)
                M(k,4)=i;
            end
        end
    end
end
end
```

```
function [A] = matA(M,n)
% Renvoie la matrice clé du problème, qui donne les relations entre les u_i.

for i=1:size(M,1)
    A(i,i)=4;
    for j=M(i,:)
```

```
        if j~0
            A(i,j)=-1;
        end
    end
end

end
```

scriptdessinq2.m est donné ci-dessous :

```
n=20; % à changer
h=1/n;
[X,Y] = Discretisation(n);
M = matM(X,Y,n);
A = matA(M,n);
N=length(X);
F=ones(N,1);

U=inv(A)*(1/n^2)*F;

x1=linspace(0,2,2*n+1);
y1=linspace(0,2,2*n+1);

[Px,Py] = meshgrid(x1,y1);

Z = zeros(2*n+1,2*n+1);

U2 = zeros((2*n-1)*(n-1),1) ;
for i = 1:(2*n-1)*(n-1)
U2(i)=U(i);
end
U2 = reshape(U2,2*n-1,n-1)';

U3 = zeros(n*(n-1),1);
for i=((2*n-1)*(n-1)+1):N
U3(i-((2*n-1)*(n-1))) = U(i);
end
U3 = reshape(U3,n-1,n)';

Z(2:n,2:(2*n)) = U2;
Z((n+1):(2*n),2:n) = U3;
Z((n+2):(2*n+1),(n+2):(2*n+1)) = ones(n,n)*NaN;


axis('equal')
contourf(Px, Py, Z, 12)
```

```
colorbar()
xlabel("x")
ylabel("y")
title(['Isovaleurs de u pour n = ',num2str(n)])
```

`scriptdessinq3.m` est donné ci-dessous :

```
n=5; % à changer
h=1/n;
[X,Y] = Discretisation(n);
M = matM(X,Y,n);
A = matA(M,n);
N = (3*n-1)*(n-1);
cotedroit = linspace(1, 2-1/n, n);
cotehaut = linspace(1/n, 1-1/n, n-1);
cotehaut1 = linspace(0, 1, n+1);
cotehaut2 = linspace(1, 2, n+1);
F=zeros(N,1);
G = zeros(N,1);
for i=1:(2*n-2) % côté bas
G(i) = (i+1)/n;
end
for j=1:n % côté droit en bas
G(j*(2*n-1)) = 2 + G(j*(2*n-1));
end
for i=((n-1)*(2*n-2)):((2*n-1)*(n-1)) % côté [1,2] au milieu
G(i) = G(i) + cotedroit(i-(n-1)*(2*n-2)+1);
end
for j=n:N % côté droit en haut
G(j) = 1 + G(j);
end
for i=(N-(n-1)):(N-1) % côté haut
G(i) = G(i) + cotehaut(i-(N-(n-1))+1);
end

for i=1:N
F(i)=max([X(i);Y(i)]);
end

U=inv(A)*((1/n^2)*F+G);

x1=linspace(0,2,2*n+1);
y1=linspace(0,2,2*n+1);

[Px,Py] = meshgrid(x1,y1);

```

```
40      Z = zeros(2*n+1,2*n+1);
41
42      U2 = zeros((2*n-1)*(n-1),1) ;
43      for i = 1:(2*n-1)*(n-1)
44      U2(i)=U(i);
45      end
46      U2 = reshape(U2,2*n-1,n-1)';
47
48      U3 = zeros(n*(n-1),1);
49      for i=((2*n-1)*(n-1)+1):N
50      U3(i-((2*n-1)*(n-1))) = U(i);
51      end
52      U3 = reshape(U3,n-1,n)';
53
54      Z(2:n,2:(2*n)) = U2;
55      Z((n+1):(2*n),2:n) = U3;
56      Z((n+2):(2*n+1),(n+2):(2*n+1)) = ones(n,n)*NaN;
57
58      Liste = linspace(0, 2, 2*n+1);
59      for i=1:(2*n)
60      Z(1,i) = Liste(i);
61      end
62      for i=1:n
63      Z(n+1,n+i) = cotehaut2(i);
64      end
65      for j=1:(n+1)
66      Z(j,2*n+1) = 2;
67      end
68      for j=(n+1):(2*n+1)
69      Z(j,n+1) = 1;
70      end
71      Z((2*n+1),1:(n+1)) = linspace(0,1,n+1); % côté du haut, on aurait pu é
            viter les boucles for précédentes en utilisant cette méthode, tant
            pis pour nous !
72
73      axis('equal')
74      contourf(Px, Py, Z, 13)
75      colorbar()
76      xlabel("x")
77      ylabel("y")
78      title(['Isovaleurs de u pour n = ',num2str(n)])
```