TD Décomposition de Domaine

Véronique Martin
veronique.martin@u-picardie.fr

LAMFA, Université de Picardie Jules Verne, France

Outline

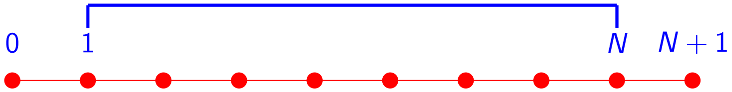
Outline

Exercice 1

Exercice 2

Exercice 3

Ex. 1 : Schwarz alterné 1D



$AU = b$ avec:

$$A = \frac{1}{h^2} \begin{pmatrix} 2+\alpha h^2 & -1 & & \\ -1 & 2+\alpha h^2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2+\alpha h^2 \end{pmatrix}$$

$$U = \left(\; U_1, U_2, \cdots, U_{N-1}, U_N \; \right)^t$$

$$b = \left(\; f(x_1) + \frac{1}{h^2} g_0, f(x_2), \cdots, f(x_{N-1}), f(x_N) + \frac{1}{h^2} g_1, \; \right)^t$$

Ex. 1 : Schwarz alterné 1D

Commandes Octave

Définir une matrice sparse:

```
e=ones(n,1);
A=spdiags([-e 2*e -e], [-1,0,1], n, n);
```

Définir une fonction:

```
f = @(x) (9+alpha)*sin(3*x);
```

Fonction lap1d

```
function u=lap1d(f,alpha,x,g0,g1)

n=length(x);
h=x(2)-x(1);
%-- Matrice
e = ones(n,1);
A = -spdiags([e -2*e e], -1:1, n-2, n-2)/h/h ...
     + alpha*speye(n-2);

%-- Second membre
b=f(x(2:end-1)');
b(1)=b(1)+g0/h/h;
b(end)=b(end)+g1/h/h;

u=A\b;
u=[g0;u;g1];
```

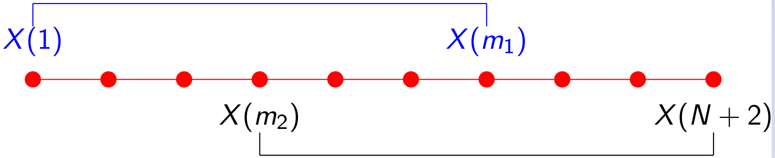
Fonction lap1d

```
function u=lap1d(f,alpha,x,g0,g1)

n=length(x);
h=x(2)-x(1);
%-- Matrice
e = ones(n,1);
A = -spdiags([e -2*e e], -1:1, n-2, n-2)/h/h ...
     + alpha*speye(n-2);

%-- Second membre
b=f(x(2:end-1)');
b(1)=b(1)+g0/h/h;
b(end)=b(end)+g1/h/h;

u=A\b;
u=[g0;u;g1];
```

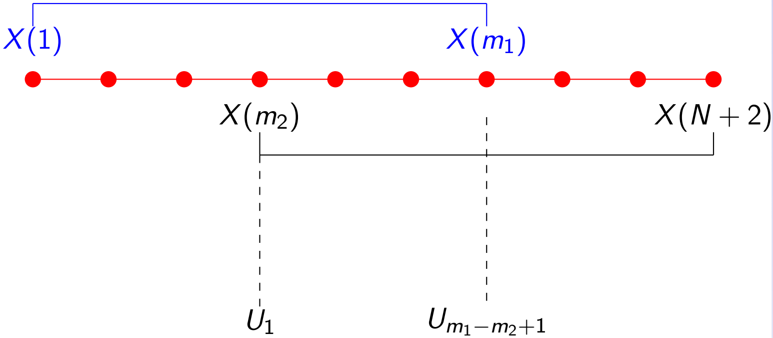
Test de la fonction lap1d

```
clear all
close all

% -- Parametres physiques
alpha=1;
f=@(x) (9+alpha)*sin(3*x);
g0=0;
g1= sin(3);

% -- Maillage
N=100;
x=linspace(0,1,N+2);

u=lap1d(f,alpha,x,g0,g1);
plot(x,u,'r')
```

Algo de Schwarz: échange de données



Algo de Schwarz: échange de données



```
%-- Maillage
N=100;
x=linspace(0,1,N+2);
m1=60;
m2=50;
x1=x(1:m1);
x2=x(m2:end);


val2=0;
val1=0;
for i=1:20
  u1=lap1d(f,alpha,x1,g0,val2);
  val1=u1(m2);
  u2=lap1d(f,alpha,x2,val1,g1);
  val2=u2(m1-m2+1);
  plot(x1,u1,'m')
  hold on
  plot(x2,u2,'k')
end
```

```
%-- Maillage
N=100;
x=linspace(0,1,N+2);
m1=60;
m2=50;
x1=x(1:m1);
x2=x(m2:end);


val2=0;
val1=0;
for i=1:20
  u1=lap1d(f,alpha,x1,g0,val2);
  val1=u1(m2);
  u2=lap1d(f,alpha,x2,val1,g1);
  val2=u2(m1-m2+1);
  plot(x1,u1,'m')
  hold on
  plot(x2,u2,'k')
end
```

```
err=1;
k=1;
E=[];
while (err>1e-6)

...
  val2=u2(m1-m2+1);
  err=abs(u1(end)-val2);
  E=[E;err];
  k=k+1;
...
end

figure(2)
semilogy(E)
```

Alterné/ Parallèle

Alterné

```
u1=lap1d(f,alpha,x1,g0,val2);
val1=u1(m2);
u2=lap1d(f,alpha,x2,val1,g1);
val2=u2(m1-m2+1);
```

Parallèle

```
u1=lap1d(f,alpha,x1,g0,val2);
u2=lap1d(f,alpha,x2,val1,g1);
val1=u1(m2);
val2=u2(m1-m2+1);
```

Multidomaine

Maillages

```
d=5;
x1=x(1:n/4);
x2=x(n/4-d:n/2);
x3=x(n/2-d:3*n/4);
x4=x(3*n/4-d:end);
```

Résolution

```
u1=lap1d(f,alpha,x1,a,val2g);
u2=lap1d(f,alpha,x2,val1d,val3g);
u3=lap1d(f,alpha,x3,val2d,val4g);
u4=lap1d(f,alpha,x4,val3d,b);

val2g=u2(d+1); val2d=u2(end-d);
val3g=u3(d+1); val3d=u3(end-d);
val4g=u4(d+1); val4d=u4(end-d);
val1d=u1(end-d);
```

Multidomaine

Maillages

```
d=5;
x1=x(1:n/4);
x2=x(n/4-d:n/2);
x3=x(n/2-d:3*n/4);
x4=x(3*n/4-d:end);
```

Résolution

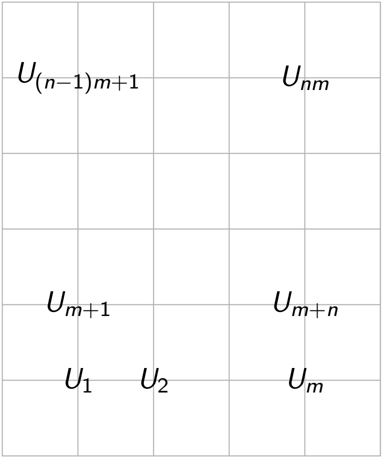
```
u1=lap1d(f,alpha,x1,a,val2g);
u2=lap1d(f,alpha,x2,val1d,val3g);
u3=lap1d(f,alpha,x3,val2d,val4g);
u4=lap1d(f,alpha,x4,val3d,b);

val2g=u2(d+1); val2d=u2(end-d);
val3g=u3(d+1); val3d=u3(end-d);
val4g=u4(d+1); val4d=u4(end-d);
val1d=u1(end-d);
```

Outline

Maillage

```
N=100;
x=linspace(0,1,N+2);
y=linspace(1,0,N+2);
[X,Y]=meshgrid(x,y);
```

$$X = \begin{pmatrix} x_0 & x_1 & \cdots & x_{N+1} \\ x_0 & x_1 & \cdots & x_{N+1} \\ & \dots & & \\ x_0 & x_1 & \cdots & x_{N+1} \end{pmatrix} \quad Y = \begin{pmatrix} y_0 & y_0 & \cdots & y_0 \\ y_1 & y_1 & \cdots & y_1 \\ & \dots & & \\ y_{N+1} & y_{N+1} & \cdots & y_{N+1} \end{pmatrix}$$


$U_{(n-1)m+1}$
U_{nm}
U_{m+1}
U_{m+n}
U_1
U_2
U_m

Matrice du Laplacien 2D discret

$$A = \begin{pmatrix} B & C & & & \\ C & B & C & & \\ & \ddots & \ddots & \ddots & \\ & & C & B & C \\ & & & C & B \end{pmatrix}$$

- B et C sont de taille $m \times m$.
- Il y a n blocs en ligne et n blocs en colonne.

Matrice du Laplacien 2D

```
function A=mat_lap2d(alpha,n,m,h)

e=ones(m,1);
B=spdiags([-e,(4+alpha*h*h)*e,-e],[-1,0,1],m,m);
C=speye(m);
e=ones(n,1);
T=spdiags([-e,-e],[-1,1],n,n);

A=(kron(speye(n,n),B)+kron(T,C))/h/h;
```

Matrice du Laplacien 2D

```
function A=mat_lap2d(alpha,n,m,h)


e=ones(m,1);
B=spdiags([-e,(4+alpha*h*h)*e,-e],[-1,0,1],m,m);
C=speye(m);
e=ones(n,1);
T=spdiags([-e,-e],[-1,1],n,n);

A=(kron(speye(n,n),B)+kron(T,C))/h/h;
```

Fonction lap2d

```
function U=lap2d(A,f,G1,G2,G3,G4,X,Y,h)

[n,m]=size(X); n=n-2; m=m-2;

% Second membre (calcul\'e sur les noeuds interieurs)
F=f(X(2:end-1,2:end-1),Y(2:end-1,2:end-1));

F(end,:)=F(end,:)+G1(2:end-1)/h/h;
F(1,:)=F(1,:)+G3(2:end-1)/h/h;
F(:,1)=F(:,1)+G4(2:end-1)/h/h;
F(:,end)=F(:,end)+G2(2:end-1)/h/h;
F=reshape(flipud(F)',n*m,1);

% Resolution du systeme
U=A\F;
U=flipud(reshape(U,n,m)');
U=[G4,[G3(2:end-1);U;G1(2:end-1)],G2];
```

Fonction lap2d

```
function U=lap2d(A,f,G1,G2,G3,G4,X,Y,h)

[n,m]=size(X); n=n-2; m=m-2;

% Second membre (calcul\'e sur les noeuds interieurs)
F=f(X(2:end-1,2:end-1),Y(2:end-1,2:end-1));

F(end,:)=F(end,:)+G1(2:end-1)/h/h;
F(1,:)=F(1,:)+G3(2:end-1)/h/h;
F(:,1)=F(:,1)+G4(2:end-1)/h/h;
F(:,end)=F(:,end)+G2(2:end-1)/h/h;
F=reshape(flipud(F)',n*m,1);

% Resolution du systeme
U=A\F;
U=flipud(reshape(U,n,m)');
U=[G4,[G3(2:end-1);U;G1(2:end-1)],G2];
```

Fonction lap2d

```
function U=lap2d(A,f,G1,G2,G3,G4,X,Y,h)

[n,m]=size(X); n=n-2; m=m-2;

% Second membre (calcul\'e sur les noeuds interieurs)
F=f(X(2:end-1,2:end-1),Y(2:end-1,2:end-1));

F(end,:)=F(end,:)+G1(2:end-1)/h/h;
F(1,:)=F(1,:)+G3(2:end-1)/h/h;
F(:,1)=F(:,1)+G4(2:end-1)/h/h;
F(:,end)=F(:,end)+G2(2:end-1)/h/h;
F=reshape(flipud(F)',n*m,1);

% Resolution du systeme
U=A\F;
U=flipud(reshape(U,n,m)');
U=[G4,[G3(2:end-1);U;G1(2:end-1)],G2];
```

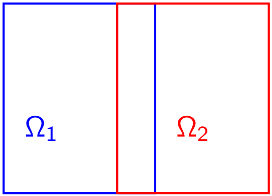
Résolution du problème monodomaine 1/2

```
% Maillage
N=100;
x=linspace(0,1,N+2);
y=linspace(1,0,N+2);
[X,Y]=meshgrid(x,y);
h=x(2)-x(1);

% Donn\'ees physiques
f= @(x,y) 2*sin(x).*sin(y);
g1= @(x) zeros(size(x));
g3= @(x) sin(1)*sin(x);
g4= @(y) zeros(size(y));
g2= @(y) sin(1)*sin(y);
alpha=0;
```

Résolution du problème monodomaine 2/2


```
% Resolution du probleme
A=mat_lap2d(alpha,N,N,h);
G1=g1(x);
G3=g3(x);
G4=g4(y');
G2=g2(y');
U=lap2d(A,f,G1,G2,G3,G4,alpha,X,Y,h);

surf(X,Y,U)
```

Décomposition du domaine




```
m1=60;
m2=50;
X1=X(:,1:m1+2);
Y1=Y(:,1:m1+2);
X2=X(:,m2+2:end);
Y2=Y(:,m2+2:end);
```

Schwarz alterné

```
A=mat_lap2d(alpha,N,N,h);
A1=mat_lap2d(alpha,N,m1,h);
A2=mat_lap2d(alpha,N,N-m2-1,h);

val2=zeros(N+2,1);
val1=zeros(N+2,1);

for i=1:20
  U1=lap2d(A1,f,G1(1:m1+2),val2,G3(1:m1+2),G4,X1,Y1,h);
  val1=U1(:,m2+2);
  U2=lap2d(A2,f,G1(m2+2:end),G2,G3(m2+2:end),val1,X2,Y2,h);
  val2=U2(:,m1-m2+1);

  surf(X1,Y1,U1)
  hold on
  surf(X2,Y2,U2)
  view([170 20])
  hold off
  pause
end
```

Avec factorisation LU

Dans le prog principal

```
A1=mat_lap2d(alpha,N,m1,h);
A2=mat_lap2d(alpha,N,N-m2-1,h);
[R1, S1] = lu(A1);
[R2, S2] = lu(A2);
```

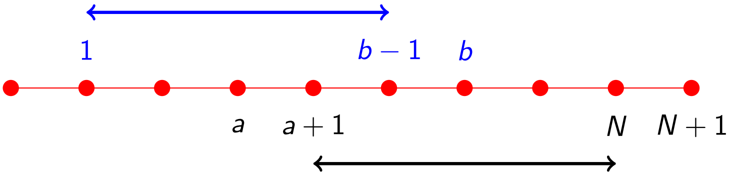
La fonction lap2d

```
function U=lap2dLU(R,S,f,G1,G2,G3,G4,X,Y,h)
Z=R\F;
U=S\Z;
```

Outline

Schwarz Multiplicatif

$$\begin{cases} U^{n+1/2} = U^n + R_1^t A_1^{-1} R_1 (b - AU^n) \\ U^{n+1} = U^{n+1/2} + R_2^t A_2^{-1} R_2 (b - AU^{n+1/2}) \end{cases}$$



$$R_1 : \mathbb{R}^N \rightarrow R^{b-1}$$

$$R_2 : \mathbb{R}^N \rightarrow R^{N-a}$$

Multiplicativ Schwarz

```
b=70; a=50;
% Restrictions
R1=[speye(b-1) sparse(b-1,N-b+1)];
R2=[sparse(N-a,a) speye(N-a)];
A1=R1*A*R1';
A2=R2*A*R2';

% Iterations
u=zeros(N,1);
for i=1:10
  r=F-A*u;
  u=u+R1'*(A1\(R1*r));
  r=F-A*u;
  u=u+R2'*(A2\(R2*r));
  plot(x,[g0;u;g1],'b','linewidth',2);
  hold on
  plot([x(a+1),x(a+1)],[0,0.1],'m-.')
  plot([x(b+1),x(b+1)],[0,0.1],'m-.')
  drawnow
end
```

Additiv Schwarz

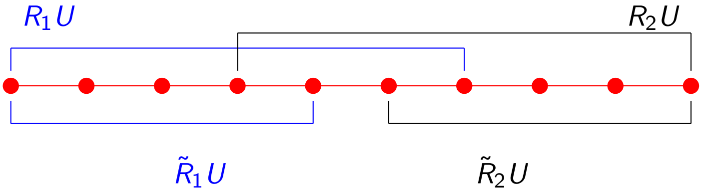
Additiv Schwarz

$$U^{n+1} = U^n + (R_1^t A_1^{-1} R_1 + R_2^t A_2^{-1} R_2)(b - AU^n)$$

En Octave

```
u=u+R1'*(A1\(R1*r))+R2'*(A2\(R2*r));
```

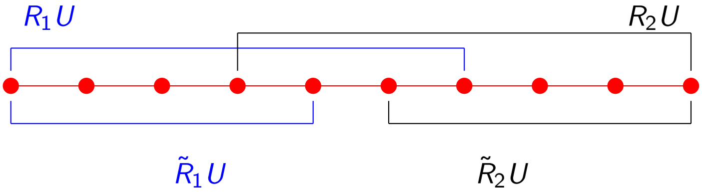
Restricted Additiv Schwarz



RAS

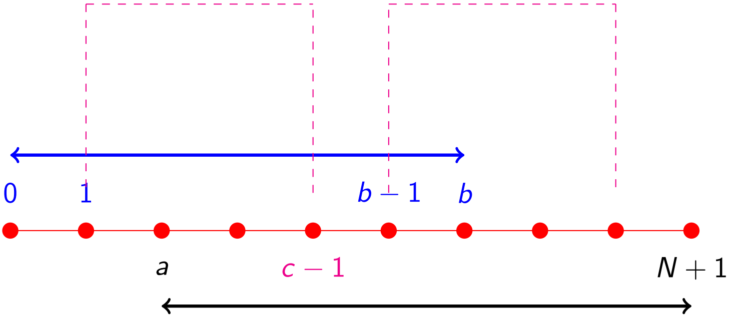
$$U^{n+1} = U^n + (\tilde{R}_1^t A_1^{-1} R_1 + \tilde{R}_2^t A_2^{-1} R_2)(b - AU^n)$$

Restricted Additiv Schwarz



RAS

$$U^{n+1} = U^n + (\tilde{R}_1^t A_1^{-1} R_1 + \tilde{R}_2^t A_2^{-1} R_2)(b - AU^n)$$



En Octave

```
c=round((a+b)/2);
R1t=R1;R1t(c:b-1,c:b-1)=0;
R2t=R2;R2t(1:c-a-1,a+1:c-1)=0;
...
u=u+R1t'*(A1\(R1*r))+R2t'*(A2\(R2*r));
```

Accélération par Krylov


```
P=@(x) R1'*(A1\(R1*x))+R2'*(A2\(R2*x));
[U, fl, relres, iter, resvec] = gmres(A,F,[],[],[],P);
semilogy(resvec)
```