

# Calcul scientifique et optimisation

*Cours ED SIM*

**Objectif**

Résoudre

$$\frac{\partial T}{\partial t} + \mathcal{L}(T) = 0$$

## **Méthodes**

- **Calcul Stochastique**
- **SPH**
- **Calcul par éléments finis (DF, VF)**
- **Décomposition orthogonale**
- **Réduction dimensionnelle 'a priori'**

3

## **Exemple**

$$\frac{\partial T}{\partial t} + \underline{V} \text{grad} T - D \Delta T = 0$$

### **Cas 1D**

$$\frac{\partial T}{\partial t} + V \frac{\partial T}{\partial x} - D \frac{\partial^2 T}{\partial x^2} = 0$$

avec la condition initiale

$$T(|x| \leq 1, t = 0) = 1,$$

$$T(|x| > 1, t = 0) = 0.$$

*Pe, We*

4

## Calcul Stochastique

### Équation et algorithme d'évolution

#### Diffusion :

- processus aléatoire
- suit une loi normale de moyenne **nulle** et de variance  $\langle dX^2 \rangle = 2.D.\Delta t$

$$dX_i^{\Delta t} = V_i \Delta t + \sqrt{2.D.\Delta t} \sigma_i$$

- $\sigma_i$  : loi normale, moyenne nulle, variance unitaire

5

## Calcul Stochastique

### Équation et algorithme d'évolution

```
Vit = 1; % vitesse
Dr = 1; % diffusion

DT = 0.01; % pas de temps
X = [-1:0.001:1]'; % 0.01 % 0.001 % 0.0001 % 0.00001
N=max(size(X));

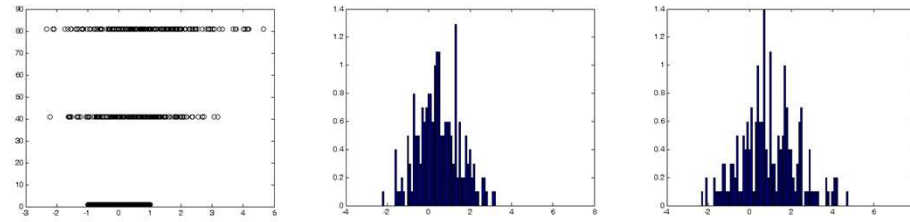
% pour le post traitement
pr = 0.2;
pos = [-6:pr:10];

for iter=1:80
    if mod(iter,20) == 1
        figure(1); plot(X,ones(N,1)*iter,'ko'); hold on
        figure; hist(X,pos);
    end

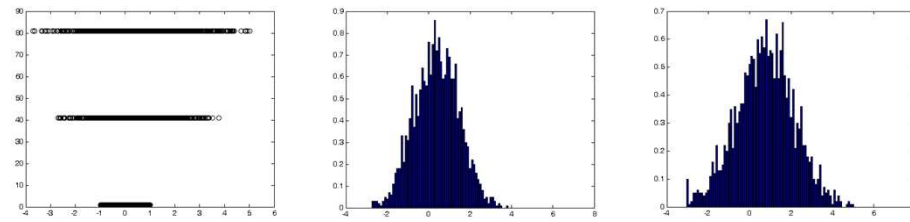
    X = X + Vit * DT + sqrt(2*Dr*DT)*randn(N,1);
end
```

6

**V=1, D =1, N=200**

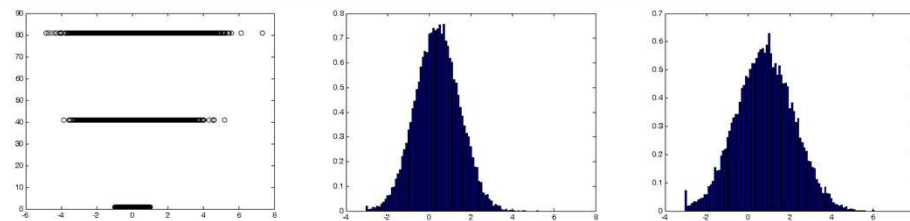


**V=1, D =1, N=2000**

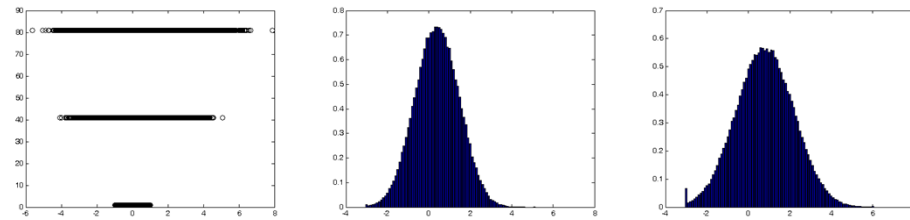


7

**V=1, D =1, N=20 000**

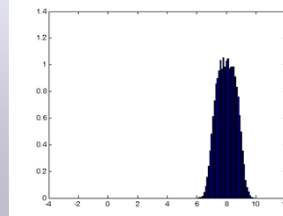
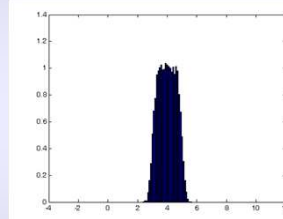
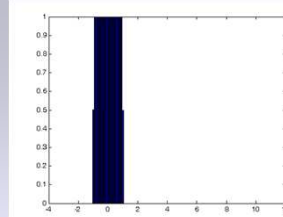
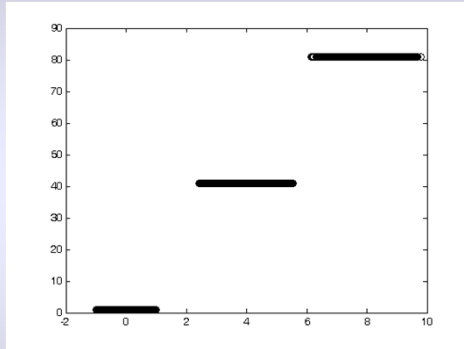


**V=1, D =1, N=200 000**



8

**V=10, D =0.5, N=20 000**



9

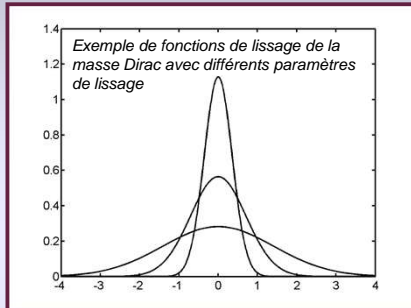
## Discussion

- (-) Grand nombre de particules**
- (-) Fluctuation de la solution**
- (-) Difficultés en forte diffusion**
- (-) Convergence en  $1/\sqrt{N}$**

- (+) Problèmes fortement convectifs**
- (+) Simplicité de mise en œuvre**

10

## Techniques déterministes particulières (SPH)



$$\int_{x=-\infty}^{x=+\infty} \zeta_{\epsilon}(x) dx = 1$$

### 1 – Changement de variable

$$\underline{\dot{X}}^* = \underline{\dot{X}} - D \frac{\partial T}{\partial \underline{X}}$$

### 2 – Forme purement convective de l'équation de convection-diffusion

$$\frac{dT}{dt} = - \frac{\partial}{\partial \underline{X}} (T \underline{\dot{X}}^*)$$

### 3 - Expression de la grandeur $T$

$$T(\underline{X}) \approx \sum_{i=1}^{i=N} \alpha_i \zeta_{\epsilon}(\underline{X} - \underline{X}_i) = \alpha \cdot \zeta_{\epsilon}(\underline{X})$$

Poids de probabilité

Fonction de lissage

11

## Techniques déterministes particulières (SPH)

```

DT = 0.01;
Xini=[-1:0.04:1]' ; X=Xini;
N=max(size(X)) ; alpha = 2/N;

% pour le post traitement
pr = 0.1; pos = [-3:pr:6];
taille = max(size(pos));
T = zeros (taille,1);

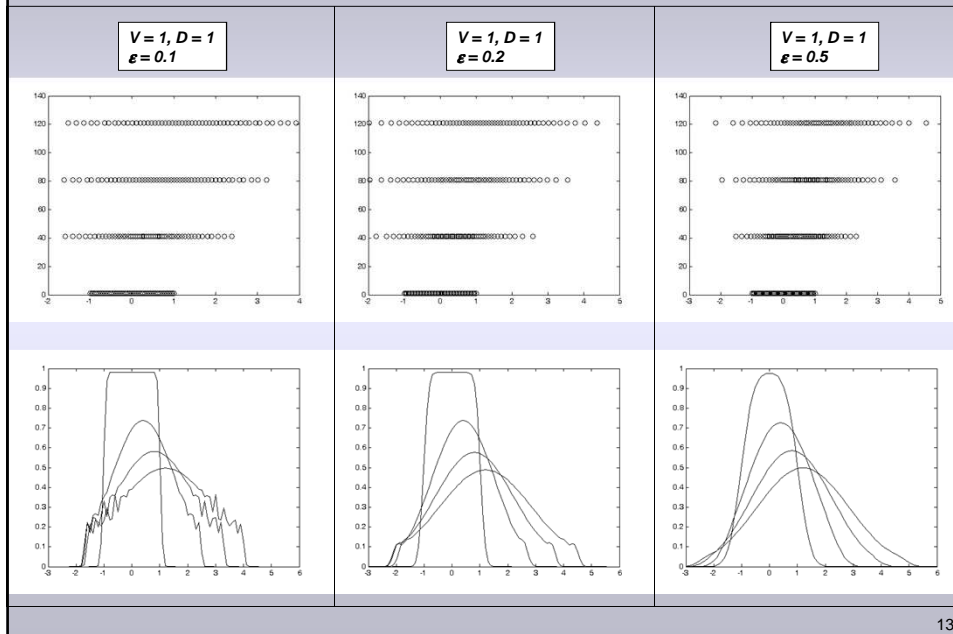
for iter=1:121
    for ii=1:taille
        T(ii) = alpha*sum (foncDzeta(pos(ii)-X));
    end
    if mod(iter,40)==1
        figure(1); plot(pos,T,'k'); hold on;
        figure(2); plot(X,ones(N,1)*iter,'ko'); hold on;
    end

    DeriXdT = zeros(N,1) ; % vecteur vitesse équivalent
    for particule = 1:N
        ValT = 0;
        ValdTdx = 0;
        for ii = 1:N
            ValT = ValT + alpha * foncDzeta(X(particule) - X(ii));
            ValdTdx = ValdTdx + alpha * deriFoncDzeta (X(particule) - X(ii));
        end
        DeriXdT(particule) = Vit - Dr * ValdTdx / ValT;
    end

    X = X + DT*DeriXdT;
end
    
```

12

## Influence du paramètre de lissage



13

## Discussion

**(+) Méthode déterministe sans maillage et sans bruit stochastique**

**(+) Aspect visuel**

**(-) Influence significative du paramètre de lissage**

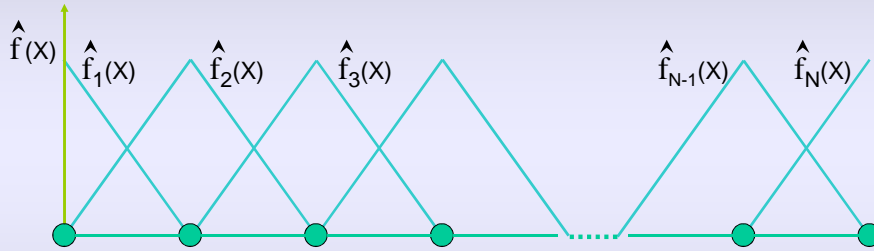
**(-) La densité de particules doit être contrôlée et le lissage doit être adapté**

**(-) Difficulté pour traiter les fortes anisotropies de distribution**

14

### Méthode des éléments finis

$$\frac{\partial T}{\partial t} + \mathcal{L}(T) = 0$$



$$T(X) = \left( \hat{f}_1(X) \quad \dots \quad \hat{f}_N(X) \right) \cdot \underbrace{\begin{pmatrix} T_1 \\ \vdots \\ T_N \end{pmatrix}}_{\underline{T}}$$

15

### Méthode des éléments finis

$$\int T^* \left( \frac{\partial T}{\partial t} + \mathcal{L}(T) \right) = 0$$

$$\underline{\underline{M}} = \int \begin{pmatrix} \hat{f}_1(X) \\ \vdots \\ \hat{f}_N(X) \end{pmatrix} \left( \hat{f}_1(X) \quad \dots \quad \hat{f}_N(X) \right) dX$$

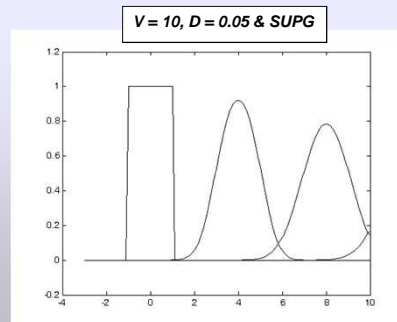
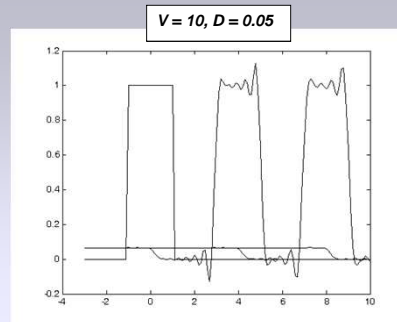
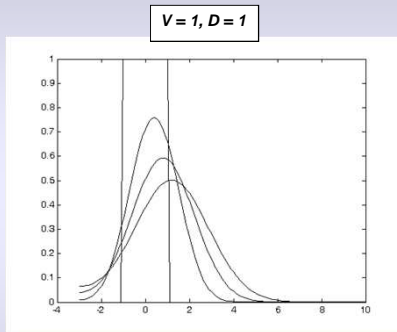
$$\underline{\underline{G}} = \int \begin{pmatrix} \hat{f}_1(X) \\ \vdots \\ \hat{f}_N(X) \end{pmatrix} \left( \mathcal{L}(\hat{f}_1(X)) \quad \dots \quad \mathcal{L}(\hat{f}_N(X)) \right) dX$$

$$\longrightarrow \underline{\underline{M}} \underline{\underline{\dot{T}}} + \underline{\underline{G}} \underline{\underline{T}} = \underline{\underline{0}} \quad \underline{\underline{M}}, \underline{\underline{G}}(N, N)$$

16



## Éléments finis : Résultats



17

## Discussion

- (+) Méthode déterministe avec maillage fixe
- (+/-) Utilisation d'un algorithme incrémental
- (-) Temps CPU fonction du nombre de DDL :  $t=N^k$
- (-) Difficile (CPU) en multi-échelles
- Choix de fonctions test appropriées (pour le cas de forte convection)
- Même cadre général que les méthodes DF, VF

18

## Décomposition de Karhunen-Loève

$$\underline{\underline{Q}} = [\underline{T}^1, \underline{T}^2, \dots, \underline{T}^p]$$

Snapshots

$$\lambda = \frac{\sum_{n=1}^p [\underline{F}^T \cdot \underline{T}^n]^2}{\underline{F}^T \cdot \underline{F}}$$

Objectif



$$\begin{aligned} \underline{\underline{A}} &= \underline{\underline{Q}} \underline{\underline{Q}}^T \\ \underline{\underline{A}} \underline{F} &= \lambda \underline{F} \rightarrow \underline{\underline{F}} = [\underline{F}_1, \underline{F}_2, \dots, \underline{F}_n] \end{aligned}$$

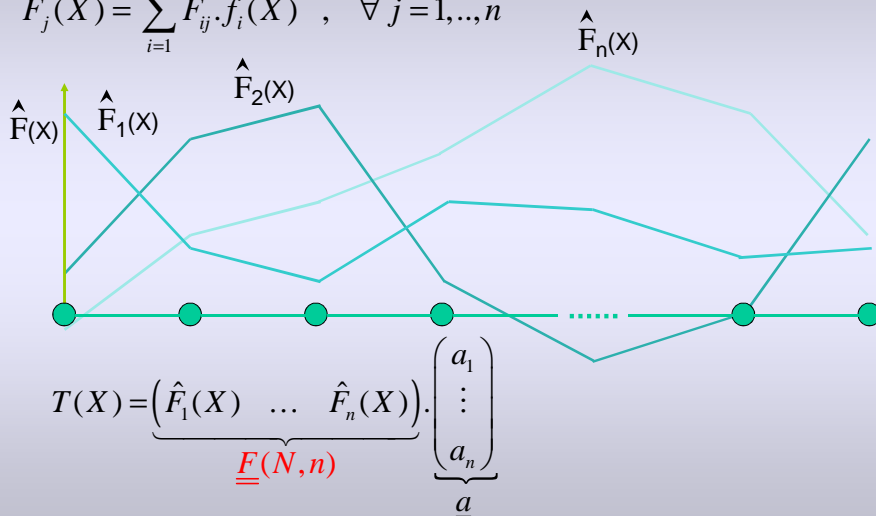
Résolution d'un problème à valeurs propres

19

## Espace réduit

'n' fonctions exprimées sur tout l'espace ( $n \ll N$ )

$$\hat{F}_j(X) = \sum_{i=1}^N F_{ij} \cdot \hat{f}_i(X) \quad , \quad \forall j = 1, \dots, n$$



20

## Formulation en base réduite

$$\int T^* \left( \frac{\partial T}{\partial t} + \mathcal{L}(T) \right) = 0$$

$$\underline{\underline{m}} = \int \begin{pmatrix} \hat{F}_1(X) \\ \vdots \\ \hat{F}_n(X) \end{pmatrix} \begin{pmatrix} \hat{F}_1(X) & \dots & \hat{F}_n(X) \end{pmatrix} dX$$

$$\underline{\underline{g}} = \int \begin{pmatrix} \hat{F}_1(X) \\ \vdots \\ \hat{F}_n(X) \end{pmatrix} \begin{pmatrix} \mathcal{L}(\hat{F}_1(X)) & \dots & \mathcal{L}(\hat{F}_n(X)) \end{pmatrix} dX$$

$$\underline{\underline{m}} \dot{\underline{\underline{a}}} + \underline{\underline{g}} \underline{\underline{a}} = 0 \quad \underline{\underline{m}}, \underline{\underline{g}} (n, n) \quad \begin{pmatrix} \underline{\underline{m}} = \underline{\underline{F}}^T \underline{\underline{M}} \underline{\underline{F}} \\ \underline{\underline{g}} = \underline{\underline{F}}^T \underline{\underline{G}} \underline{\underline{F}} \end{pmatrix}$$

21

## Algorithme de calcul en base réduite

```

DD = 0.5;
VV = 0.5;
dt = 0.001;

X = [-3:0.1:10]';

[N_NT, N_OP_NT] = calc_EF(X, VV, DD);

T0 = double(X.^2<=1);
Implicite = inv(N_NT + dt * N_OP_NT)*N_NT;
T = T0;

Base_Reduite = [];
for iter=1:3001
    if round((iter-1)/100) == (iter-1)/100
        plot(X,T,'k'); hold on
        Base_Reduite = [Base_Reduite, T];
    end
    T = Implicite * T ;
end

[vec_p, val_p] = eig(Base_Reduite * Base_Reduite' );
dim = (size(Base_Reduite,1));
val_p=diag(val_p);

if val_p(dim) > val_p(1)
    dim_nouvelle = sum ( abs(val_p/val_p(dim)) >= 1e-4 );
    Base_Reduite = vec_p(:, dim - dim_nouvelle + 1 : dim);
else
    dim_nouvelle = sum ( abs(val_p/val_p(1)) >= 1e-4 );
    Base_Reduite = vec_p(:, 1:dim_nouvelle);
end

figure; plot(X, Base_Reduite);

% exploitation de la réduction
t0 = Base_Reduite* T0;
DD = 1;
VV = 1;
[N_NT, N_OP_NT] = calc_EF(X, VV, DD);

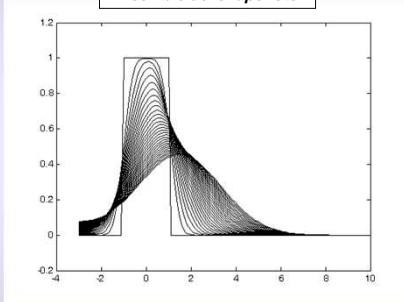
n_nt = Base_Reduite' * N_NT * Base_Reduite;
n_op_nt= Base_Reduite' * N_OP_NT * Base_Reduite;

implicite = inv(n_nt + dt * n_op_nt)*n_nt;
t=t0;
figure;
for iter=1:2401
    if round((iter-1)/400) == (iter-1)/400
        plot(X,Base_Reduite*t,'k'); hold on
    end
    t = implicite * t ;
end
    
```

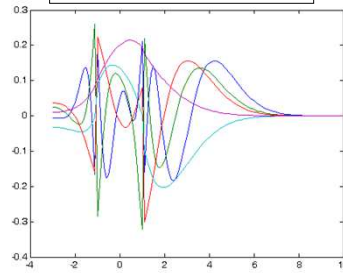
22

## Exemple 1

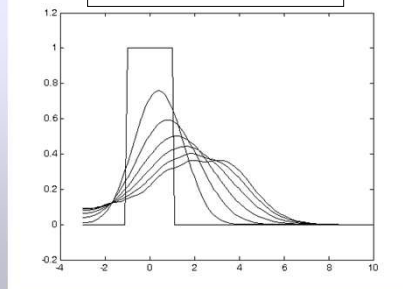
$V = 0.5, D = 0.5$   
Ensemble de 'snapshots'



Fonctions caractéristiques (5)  
de la base réduite



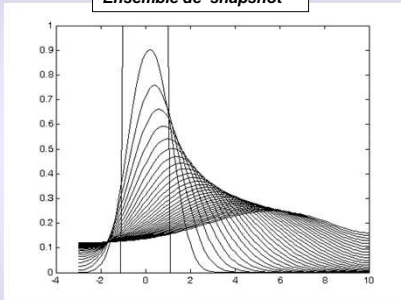
$V = 1, D = 1$   
Simulation avec base réduite



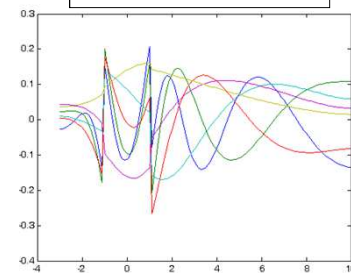
23

## Exemple 2

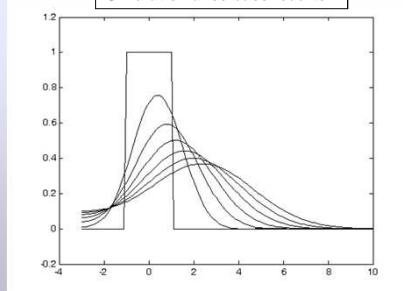
$V = 2, D = 2$   
Ensemble de 'snapshot'



Fonctions caractéristiques (5)  
de la base réduite



$V = 1, D = 1$   
Simulation avec base réduite



24

## Discussion

- (+) Méthode déterministe avec maillage fixe
- (+) Coût de calcul réduit
- (-) Difficultés à prédire des nouvelles évolutions
- (-) Pas de contrôle sur la qualité de la solution
- (-) Calcul incrémental

25

## Réduction dimensionnelle 'a priori'

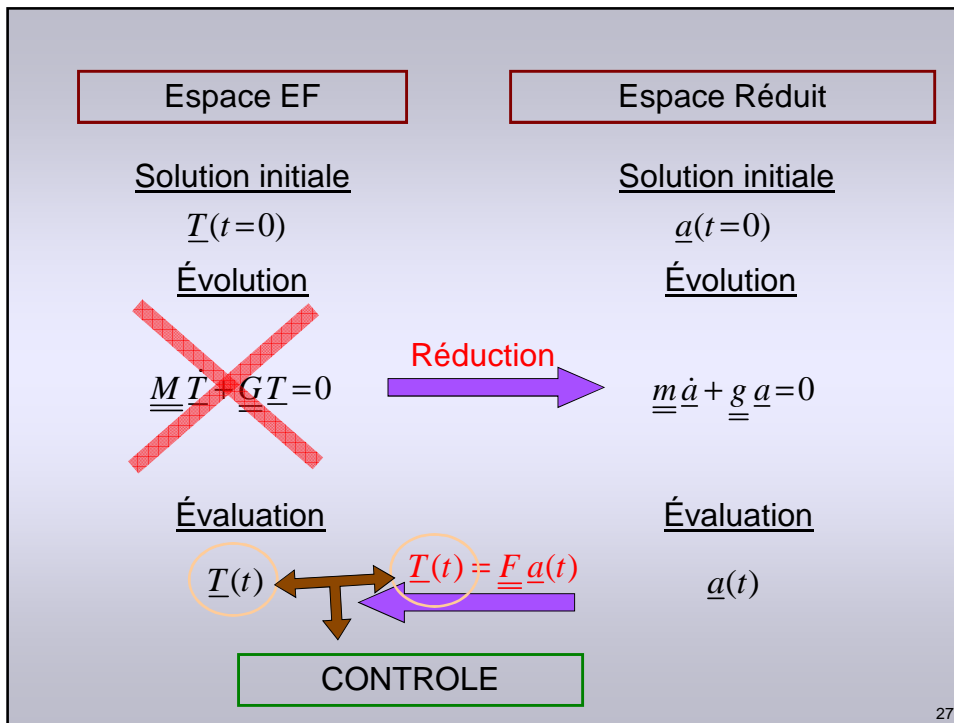
### Question 1

- Comment trouver l'espace réduit ?

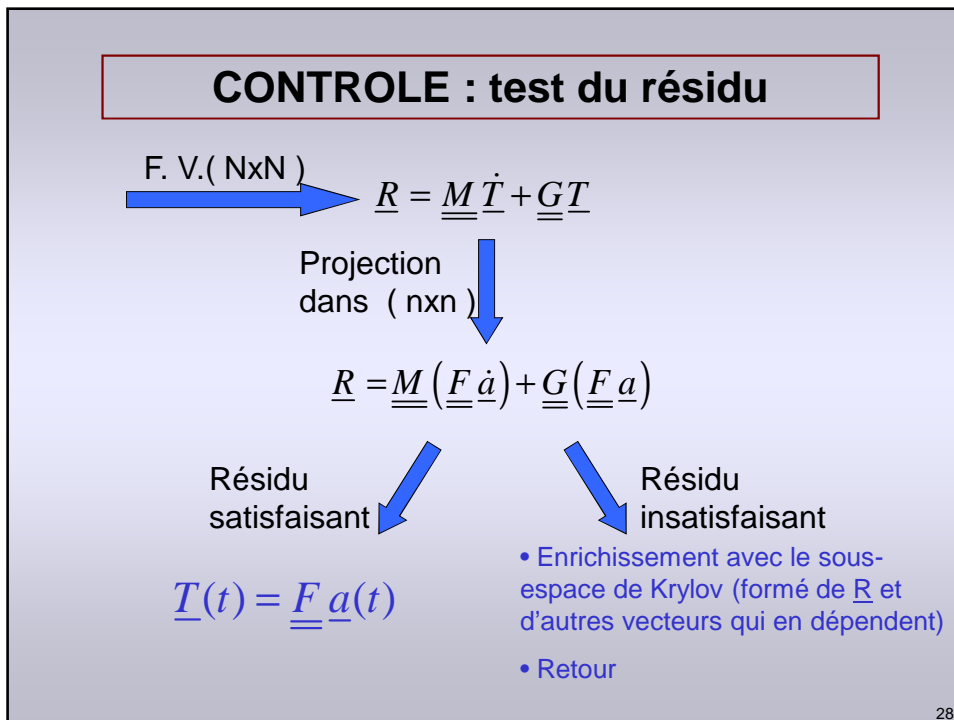
### Question 2

- Comment être sûr de son optimalité (taille et précision) ?

26



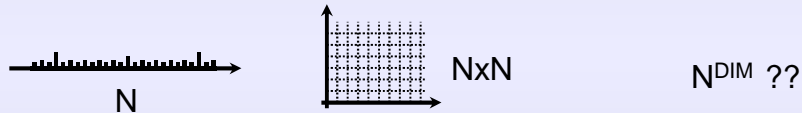
27



28

## Discussion

- (+) Méthode déterministe avec maillage fixe
- (+) Temps CPU
- (+) La taille de l'espace réduit ne dépend que de la régularité de la solution (et non de la dimension de l'espace de résolution)



- (-) Temps de résolution (pour les dimensions raisonnables) ?
- (-) Comment définir des éléments de référence et des interpolations de référence ?
- (-) Comment stocker la forme discrète de la solution ?

29

## Représentation en somme de produits tensoriels

Recherche de la solution de l'équation sous la forme d'un produit de fonctions

$$T(\underline{x}, t) = \sum_{j=1}^{\infty} \alpha_j F_j(\underline{x}) G_j(t)$$

- **Étape 1 :**

Trouver le meilleur ensemble de coefficients de pondération  $\alpha$

- **Étape 2 :**

Enrichir la description par un nouveau produit de fonctions

$$T(\underline{x}, t) = \sum_{j=1}^n \alpha_j F_j(\underline{x}) G_j(t) + R(\underline{x}) S(t)$$

30

## Démarche : Étape 1

Objectif : Trouver le meilleur ensemble de coefficients de pondération  $\alpha$

$$T(\underline{x}, t) = \sum_{j=1}^n \alpha_j F_j(\underline{x}) G_j(t)$$

Formulation variationnelle sur les ' $\alpha$ '

$$T(\underline{x}, t) = [F_1 G_1 \quad F_2 G_2 \quad \dots \quad F_n G_n] \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

$$T^*(\underline{x}, t) = [\alpha_1^* \quad \alpha_2^* \quad \dots \quad \alpha_n^*] \cdot \begin{bmatrix} F_1^T G_1^T \\ F_2^T G_2^T \\ \vdots \\ F_n^T G_n^T \end{bmatrix}$$

Obtention du système linéaire

$$\underline{H} \underline{\alpha} = \underline{0}$$

31

## Outils matriciels

$$\int T^* \left( \frac{\partial T}{\partial t} + \mathcal{L}(T) \right) = 0$$

$$\mathbb{N}_t = \int \mathbf{N}_t \mathbf{N}_t^T dt, \quad \mathbb{N}_x = \int \mathbf{N}_x \mathbf{N}_x^T dx$$

$$\mathbb{L}_t = \int \mathbf{N}_t \frac{d\mathbf{N}_t^T}{dt} dt, \quad \mathbb{L}_x = \int \mathbf{N}_x \mathcal{L} \mathbf{N}_x^T dx$$

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i^* H_{ij} \alpha_j = 0$$

$$H_{ij} = \mathbf{F}_i^T \mathbf{N}_x \mathbf{F}_j \cdot \mathbf{G}_i^T \mathbb{L}_t \mathbf{G}_j + \mathbf{F}_i^T \mathbb{L}_x \mathbf{F}_j \cdot \mathbf{G}_i^T \mathbf{N}_t \mathbf{G}_j$$

32



## Démarche : Étape 2

Objectif : Trouver le meilleur nouveau terme sous la forme

$$T(\underline{x}, t) = \sum_{j=1}^n \alpha_j F_j(\underline{x}) G_j(t) + \mathbf{R}(\underline{x}) S(t)$$

Formulation variationnelle sur R et S

$$T^*(\underline{x}, t) = R^*(\underline{x}) S(t) + \mathbf{R}(\underline{x}) S^*(t) \quad \Rightarrow \quad T^*(\underline{x}, t) = \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \begin{bmatrix} S \\ R \end{bmatrix}$$

$$\begin{bmatrix} \mathcal{L}(T) \\ \frac{\partial T}{\partial t} \end{bmatrix} = \sum_{j=1}^n \alpha_j \begin{bmatrix} \mathcal{L}(F_j) \cdot G_j \\ F_j \cdot \frac{dG_j}{dt} \end{bmatrix} + \begin{bmatrix} S \cdot \mathcal{L} & 0 \\ 0 & R \cdot \frac{d}{dt} \end{bmatrix} \begin{bmatrix} R \\ S \end{bmatrix}$$

Obtention du système non linéaire

$$\begin{pmatrix} \mathbb{L}_x \mathbf{S}^T \mathbb{N}_t \mathbf{S} & \mathbb{N}_x \mathbf{R} \mathbf{S}^T \mathbb{L}_t \\ \mathbb{N}_t \mathbf{S} \mathbf{R}^T \mathbb{L}_x & \mathbf{R}^T \mathbb{N}_x \mathbf{R} \mathbb{L}_t \end{pmatrix} \begin{pmatrix} \mathbf{R} \\ \mathbf{S} \end{pmatrix} = - \sum_{i=1}^n \alpha_i \begin{pmatrix} \mathbb{L}_x \mathbf{F}_j \mathbf{S}^T \mathbb{N}_t \mathbf{G}_j + \mathbb{N}_x \mathbf{F}_j \mathbf{S}^T \mathbb{L}_t \mathbf{G}_j \\ \mathbf{R}^T \mathbb{L}_x \mathbf{F}_j \mathbb{N}_t \mathbf{G}_j + \mathbf{R}^T \mathbb{N}_x \mathbf{F}_j \mathbb{L}_t \mathbf{G}_j \end{pmatrix}$$

33

## Illustration

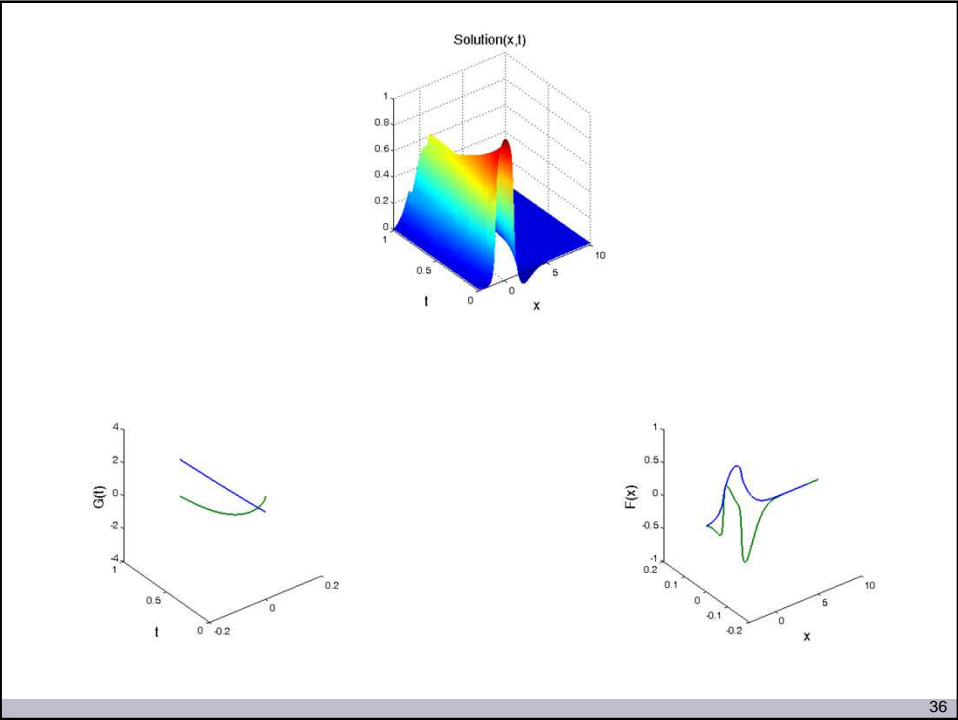
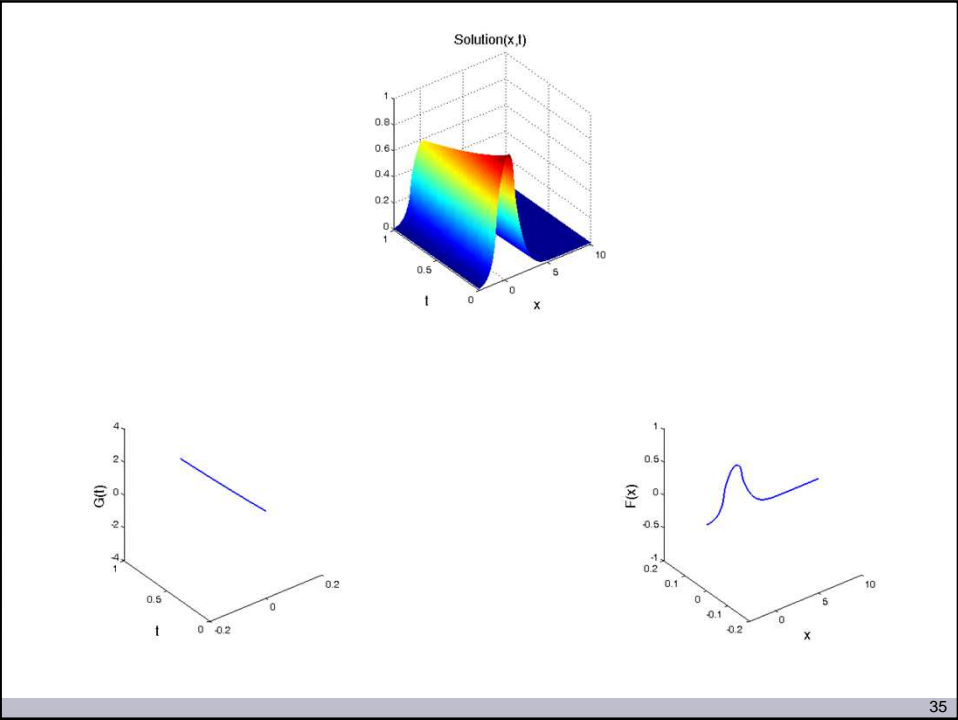
$$T(x, t) = \sum_{j=1}^n \alpha_j F_j(x) G_j(t)$$

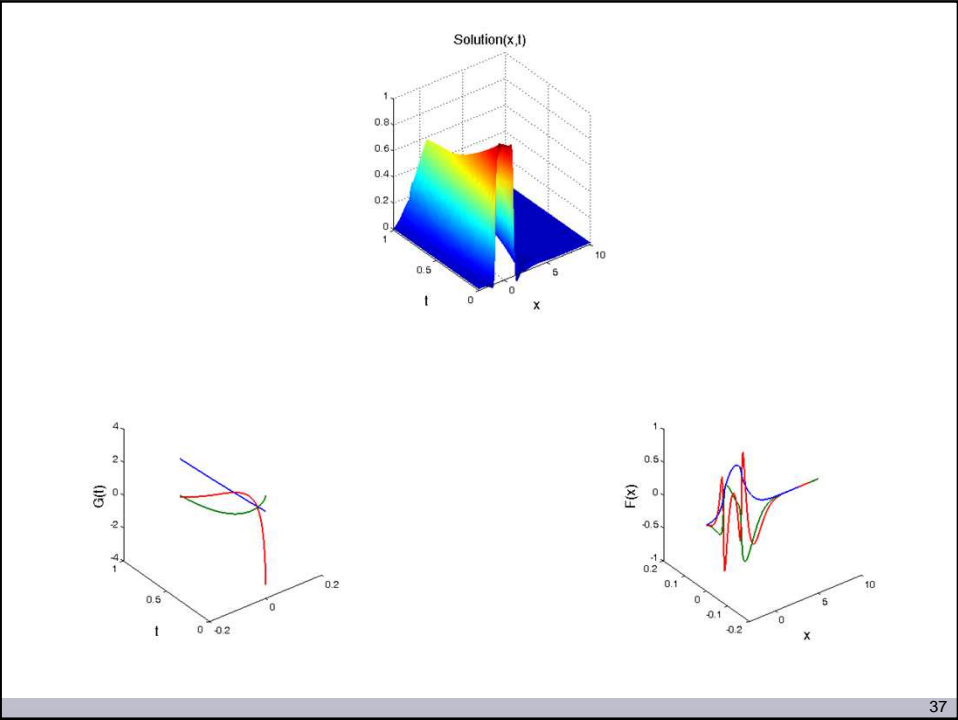
*Problème*  $\frac{\partial T}{\partial t} + V \frac{\partial T}{\partial x} - D \frac{\partial^2 T}{\partial x^2} = 0$

*Domaine*  $(x, t) \in ]-3, 10[ \times ]0, 1]$

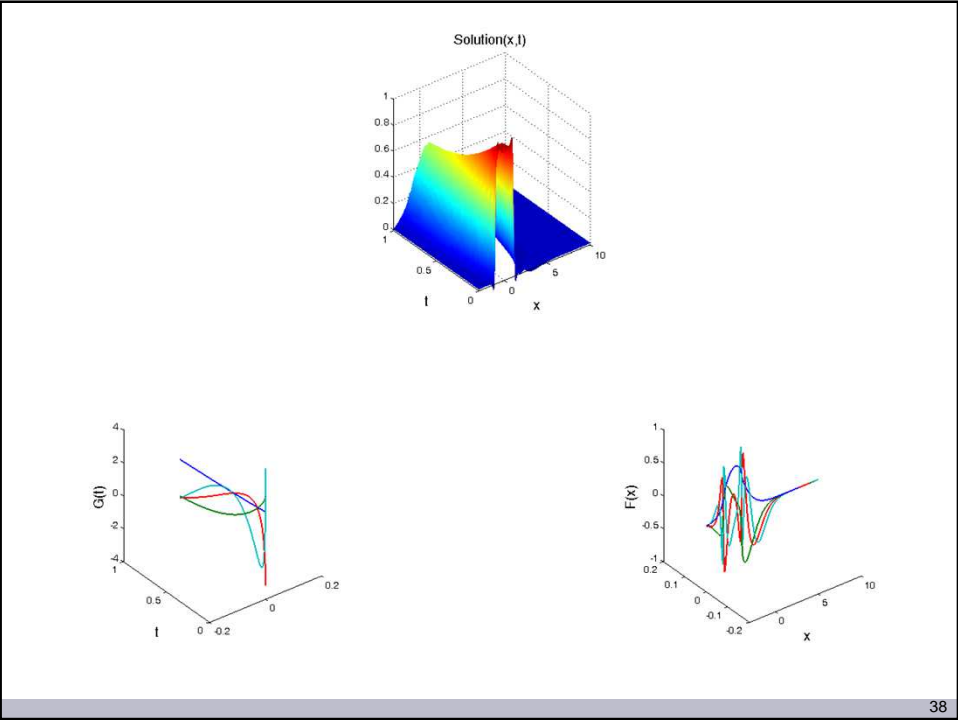
*Condition initiale*  $T(|x| \leq 1, t = 0) = 1;$   
 $T(|x| > 1, t = 0) = 0.$

34

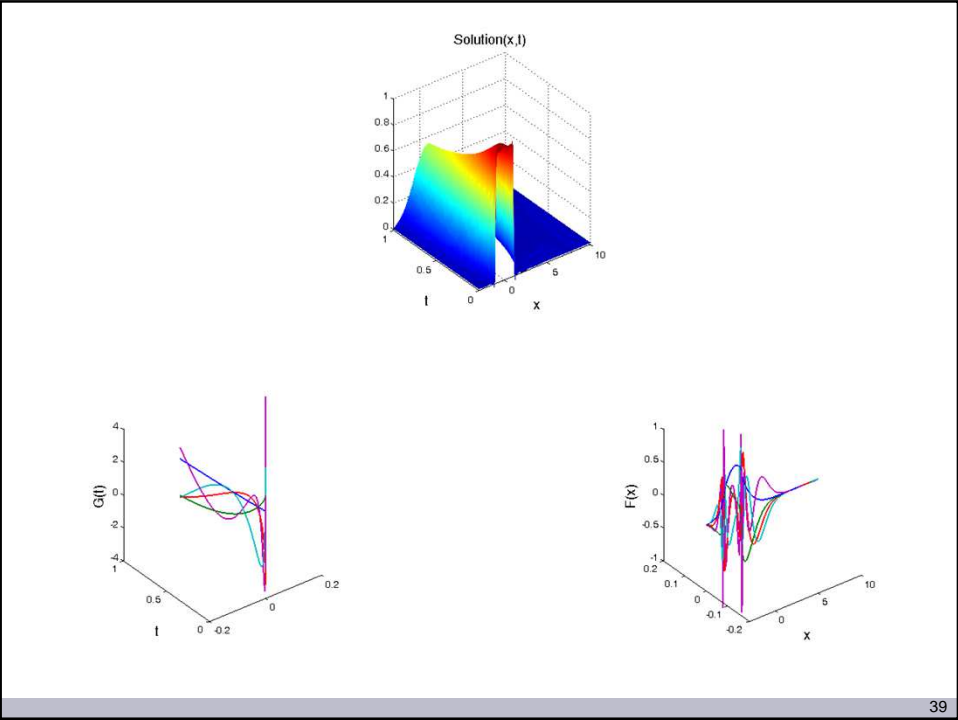




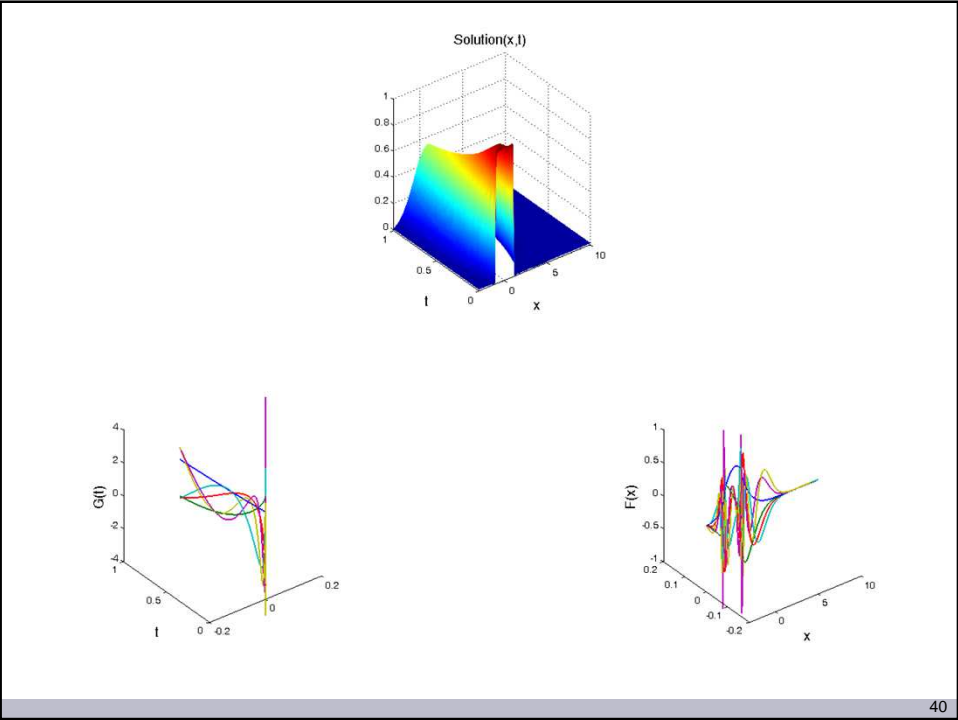
37



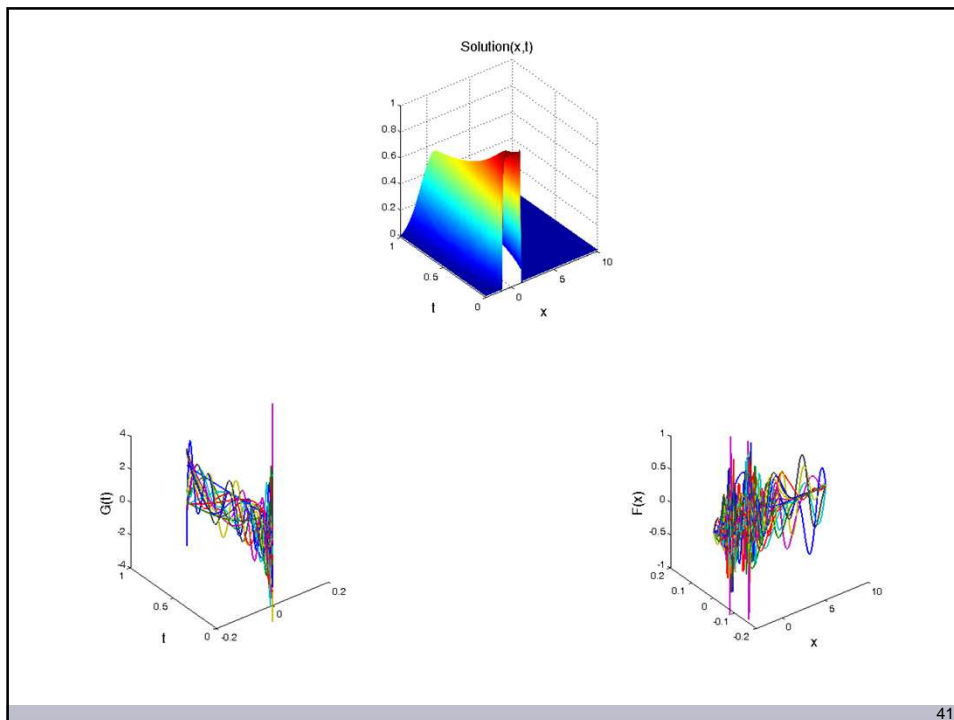
38



39



40



## Discussion

- (+) Gain d'un facteur 100 sur le temps CPU
- (+) Problème à grand incrément de temps
- (+) Disparition de la notion de stabilité transitoire
- (+) Indépendance des formulations dans les différents sous-espaces ( $EF$  pour  $(x,y)$  et  $EF SU$  pour le temps)  $\rightarrow$  autres formulations ??
- (+) Toutes les intégrales sont dans les sous-espaces
- (+) Ordre de la méthode  $O(\text{Nombre\_sous\_espaces} \cdot N)^2$
- (+) Optimalité
- (+) Permet de traiter des problèmes multidimensionnels
- (+) Permet de traiter des conditions aux limites non homogènes