One dimensional PDE

Philippe Grandclément

Laboratoire de l'Univers et de ses Théories (LUTH)
CNRS / Observatoire de Paris
F-92195 Meudon, France

philippe.grandclement@obspm.fr

Collaborators
Silvano Bonazzola, Eric Gourgoulhon, Jérôme Novak

November 14-18, 2005

Outline

1 Introduction

2 One-domain methods

3 Multi-domain methods

4 Some LORENE objects

INTRODUCTION

Type of problems

We will consider a differential equation :

$$Lu(x) = S(x) \qquad x \in U \tag{1}$$
$$Bu(y) = 0 \qquad y \in \partial U \tag{2}$$

where L are B are linear differential operators.
In the following, we will only consider one-dimensional cases $U = [-1; 1]$.
We will also assume that u can be expanded on some functions :

$$\tilde{u}(x) = \sum_{n=0}^{N} \tilde{u}_n \phi_n(x). \tag{3}$$

Depending on the choice of expansion functions ϕ_k, one can generate :

- finite difference methods.
- finite element method.
- spectral methods.

The weighted residual method

Given a scalar product on U, one makes the residual $R = Lu - S$ small in the sense :

$$\forall k \in \{0, 1, ...N\}, \quad (\xi_k, R) = 0, \tag{4}$$

under the constraint that u verifies the boundary conditions.
The ξ_k are called the test functions.

Standard spectral methods

The expansion functions are global orthogonal polynomials functions, like Chebyshev and Legendre.
Depending on the choice of test functions :

Tau method

The ξ_k are the expansion functions. The boundary conditions are enforced by an additional set of equations.

Collocation method

The $\xi_k = \delta\left(x - x_k\right)$ and the boundary conditions are enforced by an additional set of equations.

Galerkin method

The expansions and the test functions are chosen to fulfill the boundary conditions.

Optimal methods

Definition :

A numerical method is said to be optimal iff the resolution of the equation does not introduce an error greater than the one already done by interpoling the exact solution.

- u_{exact} is the exact solution.
- $I_N u_{\text{exact}}$ is the interpolant of the exact solution.
- $u_{\text{num.}}$ is the numerical solution.

The method is optimal iff $\max_\Lambda \left(|u_{\text{exact}} - I_N u_{\text{exact}}|\right)$ and $\max_\Lambda \left(|u_{\text{exact}} - u_{\text{num.}}|\right)$ have the same behavior when $N \to \infty$.

ONE-DOMAIN METHODS

Matrix representation of L

The action of L on u can be given by a matrix L_{ij}

If $u = \sum_{k=0}^{N} \tilde{u}_k T_k$ then

$$Lu = \sum_{i=0}^{N} \sum_{j=0}^{N} L_{ij} \tilde{u}_j T_i$$

L_{ij} is obtained by knowing the basis operation on the expansion basis. The k^{th} column is the coefficients of LT_k.

Example of elementary operations with Chebyshev

If $f = \sum_{n=0}^{\infty} a_n T_n(x)$ then $Hf = \sum_{n=0}^{\infty} b_n T_n(x)$

H is the multiplication by x

$$b_n = \frac{1}{2}\left(\left(1+\delta_{0n-1}\right)a_{n-1} + a_{n+1}\right) \text{ with } n \geq 1$$

H is the derivation

$$b_n = \frac{2}{\left(1+\delta_{0n}\right)} \sum_{p=n+1,\, p+n \text{ odd}}^{\infty} p a_p$$

H is the second derivation

$$b_n = \frac{1}{\left(1+\delta_{0n}\right)} \sum_{p=n+2,\, p+n \text{ even}}^{\infty} p\left(p^2-n^2\right) a_p$$

Tau method

The test functions are the T_k

$(T_k|R) = 0$ implies : $\displaystyle\sum_{j=0}^{N} L_{kj}\tilde{u}_j = \tilde{s}_k$ ($N+1$ equations).

The $\tilde{s}_k$ are the coefficients of the interpolant of the source.

Boundary conditions

- $u\,(x=-1) = 0 \Longrightarrow \displaystyle\sum_{j=0}^{N} (-1)^j\,\tilde{u}_j = 0$
- $u\,(x=+1) = 0 \Longrightarrow \displaystyle\sum_{j=0}^{N} \tilde{u}_j = 0$

One considers the $N-1$ first residual equations and the 2 boundary conditions. The unknowns are the $\tilde{u}_k$.

Collocation method

The test functions are the $\delta_k = \delta(x - x_k)$

$(\delta_n|R) = 0$ implies that : $Lu(x_n) = s(x_n)$ ($N+1$ equations).

$$\sum_{i=0}^{N} \sum_{j=0}^{N} \tilde{u}_j L_{ij} T_i(x_n) = s(x_n) \quad \forall n \in [0, N]$$

Boundary conditions

- Like for the Tau-method they are enforced by two additional equations.
- One has to relax the residual conditions in x_0 and x_N.

Galerkin method : choice of basis

We need a set of functions that

- are easily given in terms of basis functions.
- fulfill the boundary conditions.

Example

If one wants $u(-1) = 0$ and $u(1) = 0$, one can choose :

- $G_{2k}(x) = T_{2k+2}(x) - T_0(x)$
- $G_{2k+1}(x) = T_{2k+3}(x) - T_1(x)$

Let us note that only $N - 1$ functions G_i must be considered to maintain the same order of approximation (general feature).

Transformation matrix

Definition

The G_i are given in terms of the T_i by a transformation matrix M
M is a matrix of size $N+1 \times N-1$.

$$G_i = \sum_{j=0}^{N} M_{ji} T_j \quad \forall i \leq N-2 \tag{5}$$

Example

$$M_{ij} = \begin{pmatrix} -1 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The Galerkin system (1)

Expressing the equations $(G_n|R)$

- u is expanded on the Galerkin basis.

$$u = \sum_{i=0}^{N-2} \tilde{u}_i^G G_i(x). \quad (6)$$

- The expression of Lu is obtained in terms of T_i via M_{ij} and L_{ij}.
- $(G_n|Lu)$ is computed by using, once again M_{ij}
- The source is NOT expanded in terms of G_i but by the T_i.
- $(G_n|S)$ is obtained by using M_{ij}
- This is $N-1$ equations.

The Galerkin system (2)

$(G_n|R) = 0 \quad \forall n \leq N-2$

$$\sum_{k=0}^{N-2} \tilde{u}_k^G \sum_{i=0}^{N} \sum_{j=0}^{N} M_{in} M_{jk} L_{ij} \left(T_i|T_i\right) = \sum_{i=0}^{N} M_{in} \tilde{s}_i \left(T_i|T_i\right), \quad \forall n \leq N-2 \tag{7}$$

The $N-1$ unknowns are the coefficients $\tilde{u}_n^G$.
The transformation matrix M is then used to get :

$$u(x) = \sum_{k=0}^{N} \left(\sum_{n=0}^{N-2} M_{kn} \tilde{u}_n^G \right) T_k$$

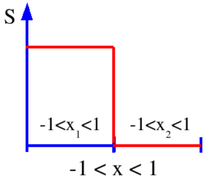
MULTI-DOMAIN METHODS

Multi-domain decomposition

Motivations

- We have seen that discontinuous functions (or not $\mathcal{C}^\infty$ functions) are not well represented by spectral expansion.
- However, in physics, we may be interested in such fields (for example the surface of a strange star can produce discontinuities).
- We also may need to use different functions in various regions of space.
- In order to cope with that, we need several domains in such a way that the discontinuities lies at the boundaries.
- By doing so, the functions are $\mathcal{C}^\infty$ in every domain, preserving the exponential convergence.

Multi-domain setting



- $x = \frac{1}{2}(x_1 - 1)$
- $x = \frac{1}{2}(x_2 + 1)$

Spectral decomposition with respect to x_i

- Domain 1 : $u(x<0) = \sum_{i=0}^{N} \tilde{u}_i^1 T_i(x_1(x))$
- Domain 2 : $u(x>0) = \sum_{i=0}^{N} \tilde{u}_i^2 T_i(x_2(x))$
- Same thing for the source.

Note that $\frac{\mathrm{d}}{dx} = 2\frac{\mathrm{d}}{dx_i}$

A multi-domain Tau method

Domain 1

- $(T_k|R) = 0 \Longrightarrow \sum_{j=0}^{N} L_{kj}\tilde{u}_j^1 = \tilde{s}_k^1$
- $N+1$ equations and we relax the last two. (N-1 equations)
- Same thing in domain 2.

Additional equations :

- the 2 boundary conditions.
- matching of the solution at $x = 0$.
- matching of the first derivative at $x = 0$.

A complete system

- 2N-2 equations for residuals and 4 for the matching and boundary conditions.
- 2N+2 unknowns, the $\tilde{u}_i^1$ and $\tilde{u}_i^2$

Homogeneous solution method

This method is the closest to the standard analytical way of solving linear differential equations.

Principle

- find a particular solution in each domain.
- compute the homogeneous solutions in each domain.
- determine the coefficients of the homogeneous solutions by imposing :
  - the boundary conditions.
  - the matching of the solution at the boundary.
  - the matching of the first derivative.

Homogeneous solutions

In general 2 in each domain and they can be known either :

- by numerically solving $Lu = 0$.
- or, most of the time, they can be found analytically.

The number of homogeneous solutions can be modified for regularity reasons.

Particular solution

In each domain, we can seek a particular solution g by a Tau residual method.

$$(T_k|R) = 0 \Longrightarrow \sum_{j=0}^{N} L_{kj}\tilde{g}_j = \tilde{s}_k$$

However, due to the presence of homogeneous solutions, the matrix L_{ij} is degenerate.

More precisely, L_{ij} is more and more degenerate as $N \to \infty$, the homogeneous solution being better described by their interpolant.

$$\sum_{j=0}^{N} L_{kj}\tilde{h}_j \to 0 \text{ when } N \to \infty$$

The non-degenerate operator

A non-degenerate operator O can be obtained by removing :

- the m first columns of L_{ij} (imposes that the first m coefficients of g are 0).
- the m last lines of L_{ij} (relaxes the last m equations for the residual).
- m is the number of homogeneous solutions (typically $m = 2$).

The matrix O is, generally, non-degenerate, and can be inverted.(true as long as the m first coefficients of the HS are not 0...)

Matching system

Example

- 2 domains.
- 2 homogeneous solutions in each of them.

The system (4 equations)

- two boundary conditions (left and right).
- matching of the solution across the boundary.
- matching of the first radial derivative.

The unknowns are the coefficients of the homogeneous solutions (4 in this particular case).

Variational formulation

Warning : this method is easily applicable only when using Legendre polynomials because it requires that $w(x) = 1$.
We will write Lu as $Lu \equiv -u'' + Fu$, F being a first order differential operator on u.

Starting point

- weighted residual equation :

$$(\xi|R) = 0 \Longrightarrow \int \xi \left(-u'' + Fu\right) \mathrm{d}x = \int \xi s \mathrm{d}x$$

- Integration by part :

$$[-\xi u'] + \int \xi' u' \mathrm{d}x + \int \xi F u \mathrm{d}x = \int \xi s \mathrm{d}x$$

Test functions

As for the collocation method : $\xi = \delta_k = \delta(x - x_k)$ for all points but $x = -1$ and $x = 1$.

Various operators

Derivation in configuration space

$$g'(x_k) = \sum_{j=0}^{N} D_{kj} g(x_j) \tag{8}$$

First order operator F in the configuration space

$$Fu(x_k) = \sum_{j=0}^{N} F_{kj} u(x_j) \tag{9}$$

Expression of the integrals

$[-\xi u'] + \int \xi' u' \mathrm{d}x + \int \xi F u \mathrm{d}x = \int \xi s \mathrm{d}x$

- $$\int \xi_n s \mathrm{d}x = \sum_{i=0}^{N} \xi_n(x_i)\, s(x_i)\, w_i = s(x_n)\, w_n$$
- $$\int \xi_n F u \mathrm{d}x = \sum_{i=0}^{N} \xi_n(x_i)\, Fu(x_i)\, w_i = \left[\sum_{j=0}^{N} F_{nj} u(x_j)\right] w_n$$
- $$\int \xi_n' u' \mathrm{d}x = \sum_{i=0}^{N} \xi_n'(x_i)\, u'(x_i)\, w_i = \sum_{i=0}^{N}\sum_{j=0}^{N} D_{ij} D_{in} w_i u(x_j)$$

Equations for the points inside the domains

$[-\xi u'] = 0$ so that, in each domain :

$$\sum_{i=0}^{N}\sum_{j=0}^{N} D_{ij} D_{in} w_i u\left(x_j\right) + \left[\sum_{j=0}^{N} F_{nj} u\left(x_j\right)\right] w_n = s\left(x_n\right) w_n$$

In each domain : $0 < n < N$, i.e. 2N-2 equations.

Equations at the boundary

In the domain 1 :

$n = N$ and $[-\xi u'] = -u'^1 (x_1 = 1; x = 0)$

$$u'^1 (x_1 = 1) = \sum_{i=0}^{N} \sum_{j=0}^{N} D_{ij} D_{iN} w_i u^1 (x_j) + \left[\sum_{j=0}^{N} F_{Nj} u^1 (x_j) \right] w_N - s^1 (x_N) w_N$$

In the domain 2 :

$n = 0$ and $[-\xi u'] = u'^2 (x_2 = -1; x = 0)$

$$u'^2 (x_2 = -1) = -\sum_{i=0}^{N} \sum_{j=0}^{N} D_{ij} D_{i0} w_i u^2 (x_j) - \left[\sum_{j=0}^{N} F_{0j} u^2 (x_j) \right] w_0 + s^2 (x_0) w_0$$

Matching equation

$u'^1(x_1 = 1; x = 0) = u'^2(x_2 = -1; x = 0) \Longrightarrow$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} D_{ij} D_{iN} w_i u^1(x_j) + \left[\sum_{j=0}^{N} F_{Nj} u^1(x_j)\right] w_N$$

$$+ \quad \sum_{i=0}^{N}\sum_{j=0}^{N} D_{ij} D_{i0} w_i u^2(x_j) + \left[\sum_{j=0}^{N} F_{0j} u^2(x_j)\right] w_0$$

$$= s^1(x_N) w_N + s^2(x_0) w_0$$

Additional equations

- Boundary condition at $x = -1$: $u^1(x_0) = 0$
- Boundary condition at $x = 1$: $u^2(x_N) = 0$
- Matching at $x = 0$: $u^1(x_N) = u^2(x_0)$

We solve for the unknowns $u^i(x_j)$.

Why Legendre ?

Suppose we use Chebyshev : $w(x) = \frac{1}{\sqrt{1-x^2}}$.

$$\int -u'' f w \mathrm{d}x = [-u' f w] + \int u' f' w' \mathrm{d}x$$

Difficult (if not impossible) to compute u' at the boundary, given that w is divergent there $\Longrightarrow$ difficult to impose the weak matching condition.

SOME LORENE OBJECTS

Array of double : the Tbl

- Constructor : `Tbl::Tbl(int ... )`. The number of dimension is 1, 2 or 3.
- Allocation : `Tbl::set_etat_qcq()`
- Allocation to zero : `Tbl::annule_hard()`
- Reading of an element : `Tbl::operator()(int ...)`
- Writing of an element : `Tbl::set(int...)`
- Output : operator cout

Matrix : `Matrice`

- Constructor : `Matrice::Matrice(int, int)`.
- Allocation : `Matrice::set_etat_qcq()`
- Allocation to zero : `Matrice::annule_hard()`
- Reading of an element : `Matrice::operator()(int, int)`
- Writing of an element : `Matrice::set(int, int)`
- Output : operator cout
- Allocation of the banded form : `Matrice::set(int up, int down)`
- Computes the LU decomposition : `Matrice::set_lu()`
- Inversion of a system $AX = Y$: `Tbl Matrice::inverse(Tbl y)`. The LU decomposition must be done before.

Tuesday directory

What it provides

- Routines to computes collocation points, weights, and coefficients (using Tbl).
- For Chebyshev (cheby.h and cheby.C)
- For Legendre (leg.h and leg.C)
- The action of the second derivative in Chebyshev space (solver.C)

What should I do ?

- Go to `Lorene/School05` directory.
- type `cvs update -d` to get todays files.
- compile `solver` (using make).
- run it ... (disappointing isnt'it ?).
- write what is missing.