Multigrid Formulation for Finite Elements.

Trach-Minh Tran, Stephan Brunner

v0.2, October 2012

A multigrid formulation for finite elements is derived, using variational principles. More specifically the grid transfer operators will be derived and tested in 1D Cartesian, cylindrical and spherical geometry for arbitrary order B-Splines.

Contents

1 The discretized problem

Consider the one-dimensional linear integro-differential problem

$$\mathcal{L}(u) = f, \qquad 0 \leq x \leq L, \tag{1}$$

with suitable boundary conditions. On an *equidistant* mesh with interval $h = L/N$ and using the *weak form* of Eq. (1), the linear system to be solved on this grid (which will be referred as the *fine* grid) can be written as (see [1], [2]):

$$\sum_{i'=1}^{N+p} A^h_{ii'} u^h_{i'} = b^h_i, \qquad A^h_{ii'} = \int_0^L \Lambda^h_i \mathcal{L}(\Lambda^h_{i'})\, x^\alpha dx, \qquad b^h_i = \int_0^L f \Lambda^h_i \, x^\alpha dx, \tag{2}$$

where p is the order the Splines Λ^h_i and $\alpha = 0, 1, 2$ for Cartesian, cylindrical and spherical coordinates respectively. It should be noted that the unknowns u^h_i of this linear system are the *expansion coefficients* of the discretized solution of the problem $u^h(x)$

$$u^h(x) = \sum_{i'=1}^{N+p} u^h_{i'} \Lambda^h_{i'}(x) \tag{3}$$

and the right hand sides b^h_i are defined as the *projection* of $f(x)$ on the same basis functions, in contrast with the Finite Differences (FD) or Finite Volume (FV) formulations where u^h_i and b^h_i are the *nodal values* of u and f.

On the *coarser* mesh with interval $2h = 2L/N$, the discretized linear system can be written as

$$\sum_{i'=1}^{N/2+p} A^{2h}_{ii'} u^{2h}_{i'} = b^{2h}_i, \qquad A^{2h}_{ii'} = \int_0^L \Lambda^{2h}_i \mathcal{L}(\Lambda^{2h}_{i'})\, x^\alpha dx, \qquad b^{2h}_i = \int_0^L f \Lambda^{2h}_i \, x^\alpha dx. \tag{4}$$

2 Transfer operators

For simplicity let consider the two-grid procedure [3] which can be summarized as follow:

1. Obtain an approximation $\mathbf{u}^h$ on the *fine* grid, using a Gauss-Seidel (GS) or a weighted Jacobi scheme. This procedure is also called *smoothing* or *relaxation.*
2. Compute the *residuals*: $\mathbf{r}^h = \mathbf{b}^h - \mathbf{A}^h \mathbf{u}^h$.
3. Obtain the residuals on the coarse mesh $\mathbf{r^{2h}}$ by *restriction* of $\mathbf{r}^h$.
4. Direct solve $\mathbf{A}^{2h}\mathbf{e}^{2h} = \mathbf{r^{2h}}$ to obtain the error $\mathbf{e}^{2h} = \mathbf{u} - \mathbf{u}^{2h}$
5. Interpolate (*prolong*) the error to obtain $\mathbf{e}^h$.
6. Correct the approximation obtained on the fine grid: $\mathbf{u}^h \leftarrow \mathbf{u}^h + \mathbf{e}^h$.
7. Relax on $\mathbf{A}^h\mathbf{u}^h = \mathbf{f^h}$, using the previously computed $\mathbf{u}^h$ as a guess.

Steps 3 and 5 are called *grid transfers* and are detailed in the following. It should be noted that the fine to coarse transfer (restriction) applies to the right hand side $\mathbf{b}^h$ while the prolongation applies to the expansion coefficients $\mathbf{u}^{2h}$.

2.1 Fine to coarse grid transfer (restriction)

The right hand side on the fine and coarse grid can be written as

$$b_i^h = \int_0^L f\Lambda_i^h\, x^\alpha dx = \sum_{i'=1}^{N+p} f_{i'}^h \underbrace{\int_0^L \Lambda_i^h \Lambda_{i'}^h\, x^\alpha dx}_{M_{ii'}^{h,h}},$$

$$b_i^{2h} = \int_0^L f\Lambda_i^{2h}\, x^\alpha dx = \sum_{i'=1}^{N+p} f_{i'}^h \underbrace{\int_0^L \Lambda_i^{2h} \Lambda_{i'}^h\, x^\alpha dx}_{M_{ii'}^{2h,h}}$$

where the expansion $f(x) = \sum_{i=1}^{N+p} f_i^h \Lambda_i^h(x)$ has been used. Elimination of $\mathbf{f^h}$ leads to the definition of the *restriction* matrix:

$$\mathbf{b}^{2h} = \mathbf{R}_h^{2h}\mathbf{b}^h, \qquad \boxed{\mathbf{R}_h^{2h} = \mathbf{M}^{2h,h}(\mathbf{M}^{h,h})^{-1}}. \tag{5}$$

Note that the computation of the *mass matrices* $\mathbf{M}^{h,h}$ and $\mathbf{M}^{2h,h}$ can be done *exactly* using a Gauss integration with $N_G = \lceil p + (\alpha+1)/2 \rceil$ points.

Another way to derive the restriction operator $\mathbf{R}_h^{2h}$ is by noting that the basis functions Λ_i^{2h} are *piecewise* C_h^{p-1} *polynomials* with *breaks* on the fine grid points $x_i = ih$, and thus can be expressed *uniquely* as

$$\Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} c_{ii'} \Lambda_{i'}^h(x), \quad i = 1 \dots N/2 + p. \tag{6}$$

Projecting this equation on the basis Λ_j^h then leads to

$$\sum_{i'=1}^{N+p} c_{ii'} \int_0^L \Lambda_{i'}^h \Lambda_j^h\, x^\alpha dx = \int_0^L \Lambda_i^{2h} \Lambda_j^h\, x^\alpha dx, \qquad i = 1 \dots N/2 + p, \quad j = 1, \dots N + p$$

$$\Longrightarrow \mathbf{c} \cdot \mathbf{M}^{h,h} = \mathbf{M}^{2h,h} \Longrightarrow \mathbf{c} = \mathbf{M}^{2h,h}(\mathbf{M}^{h,h})^{-1} = \mathbf{R}_h^{2h}$$

and finally

$$\boxed{\Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} \left(\mathbf{R}_h^{2h}\right)_{ii'} \Lambda_{i'}^h(x), \quad i = 1 \dots N/2 + p} \tag{7}$$

Because the expansion coefficients $c_{ii'}$ of $\Lambda_i^{2h}(x)$ (rows of the restriction matrix $\mathbf{R}_h^{2h}$) on the fine mesh basis are *unique*, $\mathbf{R}_h^{2h}$ should be independent of the geometry exponent α or more generally, of the definition of the *projection* (or scalar product) used to calculate the restriction matrix. Furthermore, since the supports of both Λ_i^h and Λ_i^{2h} are *compact*, the matrix $\mathbf{R}_h^{2h}$ should be *sparse.*

One can show that, using (7), the restriction of the fine mesh FE matrix $\mathbf{A}^h$ is given by

$$\mathbf{A}^{2h} = \mathbf{R}_h^{2h}\mathbf{A}^h\left(\mathbf{R}_h^{2h}\right)^T. \tag{8}$$

2.2 Coarse to fine grid transfer (prolongation)

Let denote the discretized solution on the coarse mesh of $\mathbf{A}^{2h}\mathbf{u}^{2h} = \mathbf{R}_h^{2h}\mathbf{b}^h$ by

$$u^{2h}(x) = \sum_{i=1}^{N/2+p} u_i^{2h}\Lambda_i^{2h}(x),$$

and seek for an approximated solution on the fine mesh $\mathbf{u}^h$

$$u^h(x) = \sum_{i=1}^{N+p} u_i^h \Lambda_i^h(x).$$

by *prolongation* of $\mathbf{u}^{2h}$ (instead of solving $\mathbf{A}^h\mathbf{u}^h = \mathbf{b}^h$). A reasonable solution is to *minimize* the square of the error norm defined as

$$\epsilon^2 = \|u^h(x) - u^{2h}(x)\|^2 \equiv \int_0^L [u^h(x) - u^{2h}(x)]^2\, x^\alpha dx,$$

$$\frac{\partial\epsilon^2}{\partial u_i^h} = 0 \Longrightarrow \sum_{i'=1}^{N+p} u_i^h \int_0^L \Lambda_i^h \Lambda_{i'}^h\, x^\alpha dx = \sum_{i'=1}^{N/2+p} u_i^{2h} \int_0^L \Lambda_i^h \Lambda_{i'}^{2h}\, x^\alpha dx.$$

This yields the prolonged (or interpolated) *coarse grid* solution on the *fine grid*

$$\mathbf{u}^h = \mathbf{P}_{2h}^h \mathbf{u}^{2h}, \qquad \boxed{\mathbf{P}_{2h}^h = (\mathbf{M}^{h,h})^{-1}\mathbf{M}^{h,2h} = (\mathbf{R}_h^{2h})^T} \tag{9}$$

and the coarse FE matrix can be finally expressed as

$$\boxed{\mathbf{A}^{2h} = \mathbf{R}_h^{2h}\mathbf{A}^h\mathbf{P}_{2h}^h} \tag{10}$$

2.3 An alternative derivation of grid transfer operators

Starting from the inter grid transformation of the basis functions Eq.(7), the restriction of $\mathbf{b}^h$ and the prolongation of $\mathbf{u}^{2h}$ can be derived as follow

$$b_i^{2h} = \int_0^L f\Lambda_i^{2h}\, x^\alpha dx = \sum_{i'=1}^{N+p} \left(\mathbf{R}_h^{2h}\right)_{ii'} \int_0^L f\Lambda_{i'}^h\, x^\alpha dx = \sum_{i'=1}^{N+p} \left(\mathbf{R}_h^{2h}\right)_{ii'} b_{i'}^h,$$

$$u^{2h}(x) = \sum_{i=1}^{N/2+p} u_i^{2h}\Lambda_i^{2h} = \sum_{i'=1}^{N+p} \underbrace{\left[\sum_{i=1}^{N/2+p} \left(\mathbf{R}_h^{2h}\right)_{ii'} u_i^{2h}\right]}_{u_{i'}^h} \Lambda_{i'}^h(x) \Longrightarrow \mathbf{u}^h = \left(\mathbf{R}_h^{2h}\right)^T \mathbf{u}^{2h} = \mathbf{P}_{2h}^h\mathbf{u}^{2h}.$$

3 Numerical results for the transfer operators

The prolongation matrix as defined in Eq. (9) was calculated using the BSPLINES module. A Gauss integration with $N_G = \lceil p + (\alpha+1)/2 \rceil$ points is used to carry out the numerical integrations. In the following, the results are presented for linear, quadratic and cubic Splines. Since the restriction matrix is just the transpose of the prolongation matrix, only the latter is shown. As expected, all the obtained matrices are found to be *independent* of α and *sparse*.

During the calculations, it was checked that

- The coarse matrix computed using Eq. (10) and the transfer matrix, is identical to the matrix assembled directly on the coarse grid.
- The sum of each row of the prolongation matrix is 1, since a constant function ($\mathbf{u}^{2h} = 1$) should remain constant after the grid transfer.

3.1 Linear Splines

For $N = 8$, the prolongation is a 9×5 matrix given by

$$\mathbf{P}_{2h}^{h} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

As expected, the prolongation matrix for linear Splines is identical to the one obtained for first order FD discretization, where a linear interpolation is used. One can easily check that

$$\begin{aligned} \Lambda_1^{2h}(x) &= \Lambda_1^h(x) + \frac{1}{2}\Lambda_2^h(x), \\ \Lambda_2^{2h}(x) &= \frac{1}{2}\Lambda_2^h(x) + \Lambda_3^h(x) + \frac{1}{2}\Lambda_4^h(x), \end{aligned}$$

as expected from (7).

3.2 Quadratic Splines

For $N = 8$, the prolongation is a 10×6 matrix given by

$$\mathbf{P}_{2h}^{h} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 3/4 & 1/4 & 0 & 0 & 0 \\ 0 & 1/4 & 3/4 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 3/4 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

3.3 Cubic Splines

For $N = 10$, the prolongation is a 13×8 matrix given by

$$\mathbf{P}_{2h}^{h} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3/4 & 1/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3/16 & 11/16 & 1/8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/8 & 3/4 & 1/8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/8 & 3/4 & 1/8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/8 & 11/16 & 3/16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad (13)$$

Note that from the results shown above, it is straightforward to derive the prolongation matrix for other number of intervals N.

4 Practical Considerations

4.1 Boundary conditions

The *essential Dirichlet boundary conditions* are imposed by zeroing the column and row (first column and first row for the left boundary and last column and last row for the right boundary) of the FE matrix $\mathbf{A}^h$ and putting 1 on the diagonal. The same operation should be also performed on the prolongation matrix, preserving thus the relation (10). For non-homogeneous Dirichlet boundary conditions, the elements of the column should be saved before the zeroing operation (for example $A_{21}^h, A_{31}^h, \ldots$ for the left boundary condition). They will be used later to modify the right hand side:

$$b_i^h \leftarrow b_i^h - A_{i1}^h u_1^h, \quad i = 2, \ldots$$

Nothing has to be done for *natural boundary conditions*.

4.2 Residual norm and error

The residual norm is simply defined as the Euclidean norm of the residue:

$$\|r\|_2 = \|\mathbf{b} - \mathbf{Au}\|_2 = \sqrt{\sum_i \left(b_i - \sum_{i'} A_{ii'} u_{i'} \right)^2}. \qquad (14)$$

When the *exact* solution $u(x)$ is known, the *discretization error* can defined as

$$\|e\|_2 = \sqrt{\int x^\alpha dx \left[\sum_i u_i \Lambda_i(x) - u(x) \right]^2} \qquad (15)$$

and computed using a Gauss quadrature. Note that for Splines of order p, $\|e\|_2(h)$ converges to zero as $O(h^{p+1})$.

5 The Model Problems

5.1 Cartesian geometry

The following second-order boundary value problem is considered:

$$\begin{aligned} -\frac{d^2}{dx^2}u(x) + \sigma u(x) &= \sin(\pi k x), \qquad 0 \le x \le 1 \\ u(0) = u(1) &= 0 \\ \Rightarrow u(x) &= \frac{\sin(\pi k x)}{\pi^2 k^2 + \sigma}. \end{aligned} \tag{16}$$

Using the weak form, the FE discretized matrix and right hand side can be computed as

$$A_{ii'} = \int_0^1 dx \left[\Lambda_i'(x)\Lambda_{i'}'(x) + \sigma\Lambda_i(x)\Lambda_{i'}(x)\right], \qquad b_i = \int_0^1 dx \sin(\pi k x)\Lambda_i(x). \tag{17}$$

For Splines of order p, the integration is done with a $\lceil p + 1/2 \rceil$ point Gauss quadrature which is *exact* for the matrix $\mathbf{A}$ if σ is constant.

The boundary conditions are simply imposed by setting

$$A_{ki} = A_{ik} = \delta_{ik} \qquad \text{and} \qquad b_k = 0$$

for $k = 1$ (the first equation) and $N + p$ (the last equation).

5.2 Cylindrical geometry

The following second-order boundary value problem is considered:

$$\begin{aligned} -\frac{1}{r}\frac{d}{dr}r\frac{d}{dr}u(r) + \frac{m^2}{r^2}u(r) &= j_{ms}^2 J_m(j_{ms}r), \qquad 0 \le r \le 1, \quad j_{ms} = s^{th} \text{ zero of } J_m, \\ u(1) &= 0 \\ \Rightarrow u(r) &= J_m(j_{ms}r). \end{aligned} \tag{18}$$

Using the weak form, the FE discretized matrix and right hand side can be computed as

$$A_{ii'} = \int_0^1 rdr \left[\Lambda_i'(r)\Lambda_{i'}'(r) + \frac{m^2}{r^2}\Lambda_i(r)\Lambda_{i'}(r)\right], \qquad b_i = \int_0^1 rdr j_{ms}^2\, J_m(j_{ms}r)\Lambda_i(r). \tag{19}$$

The boundary condition has only to be imposed on the last equation, using the same procedure described for the Cartesian geometry.

It should be noted here that for $m \neq 0$, the matrix elements A_{1i} and A_{i1} *diverge* since $\Lambda_1(r)$ is not equal to zero at $r = 0$. However, using a *direct solver*, one can observe that the resulting *discretization errors* as defined by Eq.(15) converge for number of Gauss points N_G slightly larger than $p + 1$, as shown in Table 1. Then, using $N_G = 4$ and 6 for the linear and cubic splines respectively, the discretization error as a function of the number of grid intervals (Fig 1) show the expected quadratic and quartic scaling respectively for the linear and cubic Splines.

6 The Multigrid Schemes

The two grid procedure described in section (2) can be generalized as follow. Let ν_1, ν_2 and μ be three iteration parameters. Given a guess $\mathbf{u}^h$ and right hand side $\mathbf{b}^h$ at the *finest* level, a MG cycle represented by

$$\boxed{\mathbf{u}^h \leftarrow MG^h(\mathbf{u}^h, \mathbf{b}^h)}$$

will compute a *new* $\mathbf{u}^h$ and is defined recursively by the following steps:

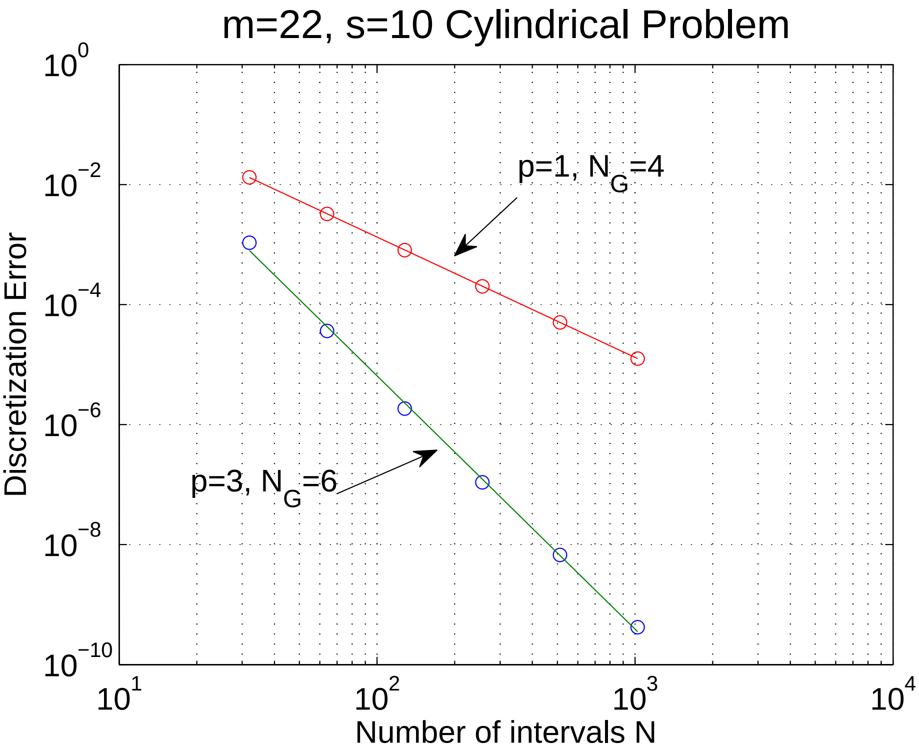

Figure 1: Discretization errors $\|e\|_2$ obtained by a *direct solver* versus the number of grid intervals N. A linear fit yields a quadratic scaling ($\sim N^{-2.0}$) for the linear Splines and a quartic convergence ($\sim N^{-4.3}$) for the cubic Splines.

Number of Gauss points	$p = 1$	$p = 3$
2	8.319E-04	
4	9.277E-04	5.799E-07
6	9.276E-04	5.936E-07
8	9.276E-04	5.936E-07

Table 1: Convergence of the *discretization error* with respect to the number of Gauss points for the cylindrical problem with $m = 1$, $s = 10$ on a 128 interval grid.

1. If h is the coarsest mesh size, direct solve $\mathbf{A}^h\mathbf{u}^h = \mathbf{b}^h$ and return.
2. Else
   - Relax $\mathbf{u}^h$ ν_1 times.
   - $\mathbf{b}^{2h} \leftarrow \mathbf{R}_h^{2h}(\mathbf{b}^h - \mathbf{A}^h\mathbf{u}^h), \quad \mathbf{u}^{2h} \leftarrow 0.$
   - $\mathbf{u}^{2h} \leftarrow MG^{2h}(\mathbf{u}^{2h}, \mathbf{b}^{2h})$ μ times.
   - $\mathbf{u}^h \leftarrow \mathbf{u}^h + \mathbf{P}_{2h}^h\mathbf{u}^{2h}.$
   - Relax $\mathbf{u}^h$ ν_2 times.

The standard V-cycle is obtained for $\mu = 1$ while $\mu = 2$ results in the W-cycle. Usually the number of *pre-smooth* and *post-smooth* sweeps ν_1 and ν_2 is limited to 1 or 2. In the following a V-cycle will be denoted by $V(\nu_1, \nu_2)$.

Another multigrid algorithm called *Full Multigrid* or FMG does not require an input guess $\mathbf{u}^h$ but solves first the problem on coarser grids and uses one or many MG cycles to obtain the problem solution. It can be represented by

$$\boxed{\mathbf{u}^h \leftarrow FMG^h(\mathbf{b}^h)}$$

and defined recursively by the following steps:

1. If h is the coarsest mesh size, direct solve $\mathbf{A}^h\mathbf{u}^h = \mathbf{b}^h$ and return.
2. Else
   - $\mathbf{b}^{2h} \leftarrow \mathbf{R}_h^{2h}(\mathbf{b}^h).$
   - $\mathbf{u}^{2h} \leftarrow FMG^{2h}(\mathbf{b}^{2h}).$
   - $\mathbf{u}^h \leftarrow \mathbf{P}_{2h}^h\mathbf{u}^{2h}.$
   - $\mathbf{u}^h \leftarrow MG^h(\mathbf{u}^h, \mathbf{b}^h)$ ν_0 times.

Note that while the MG process is an iterative process (started by setting for example the initial guess $\mathbf{u}^h = 0$), the FMG is more like a *direct solver* with appropriate values of ν_1, ν_2 and ν_0 determined experimentally.

7 Numerical Experiments

The residual norm $\|r\|_2$ and error $\|e\|_2$ defined previously are reported after each V-cycle in Table 2 for the Cartesian model problem and in Table 3 for the cylindrical one . The ratio between successive cycle $\|r\|_2$ and $\|e\|_2$ are shown in columns labeled *ratio* and measure the rate of iteration convergence. The *asymptotic* ratio of $\|r\|_2$ is called the *convergence factor.*

In all the cases shown, one can note that $\|e\|_2$ level off quickly to the discretization error obtained by using the *direct solver* on the finest grid, while the residual norms $\|r\|_2$ continue to decrease until the machine zero is eventually reached. One can also verify that the *final* discretization errors scale approximately as 8^2 and 8^4 respectively for linear and cubic Splines, as N is increased from 128 to 1024.

Most interestingly, the *iterative performance* depends very weakly on the problem size N, for both the Cartesian and the cylindrical cases. Moreover, the multigrid seems to be less efficient when linear Splines are used for the problem discretization. This iterative performance can be further improved by increasing the *iteration parameters* ν_1, ν_2 and μ, as shown in Table 4. One can also observe in the same table that the Jacobi relaxation is systematically less efficient than Gauss Seidel relaxation.

Linear B-Splines $p = 1$								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	6.219E-02		7.164E-04		2.210E-02		7.164E-04	
1	2.169E-02	0.35	5.880E-05	0.08	9.699E-03	0.44	3.622E-05	0.05
2	3.801E-03	0.18	7.806E-06	0.13	1.790E-03	0.18	1.965E-06	0.05
3	5.061E-04	0.13	3.666E-06	0.47	2.923E-04	0.16	1.583E-07	0.08
4	6.762E-05	0.13	3.564E-06	0.97	4.055E-05	0.14	6.197E-08	0.39
5	8.902E-06	0.13	3.585E-06	1.01	5.586E-06	0.14	5.655E-08	0.91
6	1.199E-06	0.13	3.589E-06	1.00	7.122E-07	0.13	5.622E-08	0.99
7	1.585E-07	0.13	3.590E-06	1.00	9.815E-08	0.14	5.620E-08	1.00
8	2.089E-08	0.13	3.590E-06	1.00	1.320E-08	0.13	5.619E-08	1.00
9	2.746E-09	0.13	3.590E-06	1.00	1.887E-09	0.14	5.619E-08	1.00
10	3.741E-10	0.14	3.590E-06	1.00	2.533E-10	0.13	5.619E-08	1.00

Cubic B-Splines $p = 3$								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	6.187E-02		7.164E-04		2.209E-02		7.164E-04	
1	1.948E-04	0.00	1.893E-06	0.00	1.685E-05	0.00	4.292E-08	0.00
2	4.316E-06	0.02	3.927E-09	0.00	1.241E-07	0.01	7.156E-11	0.00
3	1.554E-07	0.04	2.374E-09	0.60	4.184E-09	0.03	6.198E-13	0.01
4	5.750E-09	0.04	2.373E-09	1.00	1.560E-10	0.04	5.635E-13	0.91
5	2.153E-10	0.04	2.373E-09	1.00	5.912E-12	0.04	5.635E-13	1.00
6	8.122E-12	0.04	2.373E-09	1.00	2.258E-13	0.04	5.635E-13	1.00
7	3.079E-13	0.04	2.373E-09	1.00	8.777E-15	0.04	5.635E-13	1.00
8	1.173E-14	0.04	2.373E-09	1.00	1.758E-15	0.20	5.635E-13	1.00
9	4.489E-16	0.04	2.373E-09	1.00	1.709E-15	0.97	5.635E-13	1.00
10	9.571E-17	0.21	2.373E-09	1.00	1.761E-15	1.03	5.635E-13	1.00

Table 2: The multigrid $V(1,1)$ performance with Gauss-Seidel relation for a *Cartesian* problem with $k = 10$ and $\sigma = 0$, discretized on a grid with $N = 128$ and 1024 intervals, using linear and cubic B-splines. For both grid sizes, a total of 6 grid levels were considered.

Linear B-Splines $p = 1$								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	1.789E+01		9.354E-02		6.400E+00		9.354E-02	
1	3.373E+00	0.19	3.068E-03	0.03	1.826E+00	0.29	3.036E-03	0.03
2	4.895E-01	0.15	8.064E-04	0.26	3.133E-01	0.17	1.624E-04	0.05
3	6.160E-02	0.13	6.704E-04	0.83	4.581E-02	0.15	1.411E-05	0.09
4	8.013E-03	0.13	6.811E-04	1.02	5.959E-03	0.13	1.062E-05	0.75
5	9.871E-04	0.12	6.844E-04	1.00	8.098E-04	0.14	1.069E-05	1.01
6	1.283E-04	0.13	6.847E-04	1.00	1.048E-04	0.13	1.070E-05	1.00
7	1.613E-05	0.13	6.847E-04	1.00	1.504E-05	0.14	1.070E-05	1.00
8	2.097E-06	0.13	6.847E-04	1.00	2.050E-06	0.14	1.070E-05	1.00
9	2.639E-07	0.13	6.847E-04	1.00	3.008E-07	0.15	1.070E-05	1.00
10	3.500E-08	0.13	6.847E-04	1.00	4.074E-08	0.14	1.070E-05	1.00

Cubic B-Splines $p = 3$								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	1.768E+01		9.354E-02		6.399E+00		9.354E-02	
1	4.243E-02	0.00	4.727E-05	0.00	4.975E-03	0.00	6.588E-06	0.00
2	1.378E-03	0.03	1.897E-06	0.04	7.835E-05	0.02	6.578E-09	0.00
3	4.773E-05	0.03	1.814E-06	0.96	2.797E-06	0.04	4.125E-10	0.06
4	2.174E-06	0.05	1.814E-06	1.00	1.041E-07	0.04	4.092E-10	0.99
5	4.816E-07	0.22	1.814E-06	1.00	3.935E-09	0.04	4.092E-10	1.00
6	1.942E-07	0.40	1.814E-06	1.00	1.499E-10	0.04	4.092E-10	1.00
7	8.887E-08	0.46	1.814E-06	1.00	5.757E-12	0.04	4.092E-10	1.00
8	4.449E-08	0.50	1.814E-06	1.00	2.517E-13	0.04	4.092E-10	1.00
9	2.377E-08	0.53	1.814E-06	1.00	1.360E-13	0.54	4.092E-10	1.00
10	1.328E-08	0.56	1.814E-06	1.00	1.384E-13	1.02	4.092E-10	1.00

Table 3: The multigrid $V(1,1)$ performance with Gauss-Seidel relation for a one-dimensional *cylindrical* problem with $m = 22$ and $s = 10$, discretized on a grid with $N = 128$ and 1024 intervals, using linear and cubic B-splines. For both grid sizes, a total of 6 grid levels were considered.

	Cartesian problem		Cylindrical problem	
	$N = 128$	$N = 1024$	$N = 128$	$N = 1024$
$\nu_1 = 1, \nu_2 = 1, \mu = 1$	0.13	0.14	0.13	0.14
$\nu_1 = 1, \nu_2 = 2, \mu = 1$	0.08	0.08 (*0.10*)	0.08	0.08 (*0.09*)
$\nu_1 = 2, \nu_2 = 1, \mu = 1$	0.08	0.08	0.08	0.08
$\nu_1 = 2, \nu_2 = 2, \mu = 1$	0.04	0.04 (*0.08*)	0.02	0.03 (*0.08*)
$\nu_1 = 1, \nu_2 = 1, \mu = 2$	0.12	0.11	0.12	0.11

Table 4: The *convergence factor* (averaged over the last 5 cycles) for different iteration parameters ν_1, ν_2 and μ, using the linear Splines for both Cartesian ($k = 10$, $\sigma = 0$) and cylindrical ($m = 22$, $s = 10$) problems. The last entry is usually called a W-cycle while the first four designate a $V(\nu_1, \nu_2)$ cycle. Gauss Seidel relaxation is used except for the results enclosed in parenthesis which are obtained with the Jacobi (weighted with $\omega = 2/3$) relaxation.

	FMG(1,1)		FMG(2,1)	
N	$\lVert e\rVert_2$	$\lVert e\rVert_2/\lVert e\rVert_d$	$\lVert e\rVert_2$	$\lVert e\rVert_2/\lVert e\rVert_d$
4	1.011E-01	0.968	1.012E-01	0.969
8	7.781E-02	1.031	7.679E-02	1.018
16	3.332E-02	1.310	2.808E-02	1.104
32	1.516E-03	1.421	1.098E-03	1.030
64	5.168E-05	1.443	3.652E-05	1.019
128	2.012E-06	1.109	1.818E-06	1.002
256	1.125E-07	1.053	1.069E-07	1.001
512	6.819E-09	1.037	6.576E-09	1.000
1024	4.224E-10	1.032	4.093E-10	1.000
2048	2.634E-11	1.031	2.556E-11	1.000

Table 5: The discretization errors $\lVert e\rVert_2$ obtained from a FMG(ν_1, ν_2) sweep with $\nu_0 = 1$ for different grid sizes N. The columns $\lVert e\rVert_2/\lVert e\rVert_d$ display their ratio with the discretization errors obtained from a *direct* solver. The cylindrical problem with $m = 22$ and $s = 10$ using cubic Splines is considered here.

The next experiment is shown on Table 5, where two FMG(ν_1, ν_2) schemes are applied to the $m = 22$, $s = 10$ cylindrical problem with grid sizes up to $N = 2048$. Note that the problem is solved to the level of discretization for $N \geq 128$ with FMG$(2, 1)$ but not with FMG$(1, 1)$. Solving the same problem with the $V(2, 1)$ cycle required 3 iterations for all the values of N shown. Since the cost of one FMG(2,1) is ~ 2 the cost of one $V(2, 1)$ (see Appendix A), it appears that FMG is more efficient for $N \geq 128$.

Finally, in all the cases shown here, the equality (10) is verified numerically, except for the cylindrical case with $m \neq 0$. This is expected since as noted earlier, the matrix elements A_{i1} and A_{1i} diverge unless $m = 0$ in the cylindrical problem.

8 Periodic Case

8.1 Transfer operators

For periodic problems, we use *periodic* Splines [2] which satisfy the periodic boundary condition $\Lambda^h_{i+N}(x) = \Lambda^h_i(x - Nh)$. As a result, both the expansion coefficients and the right hand sides are periodic with periodicity N ($u^h_{i+N} = u^h_i$, $b^h_{i+N} = b^h_i$) and he rank of all matrices should be N instead of $N + p$ as in the non-periodic case.

The *prolongation* matrix $\mathbf{P}^h_{2h}$ as given by (9) are computed numerically and the results for $N = 8$ are given below for linear, quadratic and cubic Splines.

- Linear Splines

$$\mathbf{P}^h_{2h} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix} \tag{20}$$

- Quadratic Splines

$$\mathbf{P}_{2h}^{h} = \begin{pmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/4 & 3/4 & 0 & 0 \\ 0 & 3/4 & 1/4 & 0 \\ 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 3/4 & 1/4 \\ 0 & 0 & 1/4 & 3/4 \\ 1/4 & 0 & 0 & 3/4 \\ 3/4 & 0 & 0 & 1/4 \end{pmatrix} \tag{21}$$

- Cubic Splines

$$\mathbf{P}_{2h}^{h} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/8 & 3/4 & 1/8 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/8 & 3/4 & 1/8 \\ 0 & 0 & 1/2 & 1/2 \\ 1/8 & 0 & 1/8 & 3/4 \\ 1/2 & 0 & 0 & 1/2 \\ 3/4 & 1/8 & 0 & 1/8 \end{pmatrix} \tag{22}$$

The restriction matrix is simply $\mathbf{R}_{h}^{2h} = (\mathbf{P}_{2h}^{h})^T$. Generalization for any other number of intervals N should be straightforward.

8.2 Numerical Experiments

In order to test the grid transfer operators obtained above, the same second-order problem (16) but with the periodic boundary condition $u(x+1) = u(x)$ is considered. It should be noted that in that case, if $\sigma = 0$, the problem is singular since the solution is not *unique*! But we have observed that this problem can be avoided for a slightly non zero σ,

With $\sigma = 0.01$ and $k = 10$ and using linear and cubic Splines, we recover the same multigrid iterative performances shown in Table 2 obtained previously for non-periodic Dirichlet boundary conditions. Table 6 also shows similar iterative efficiencies for *quadratic* non-periodic and periodic problems.

The identity (10) is numerically verified in all the cases considered.

9 Conclusion

Using the variational principle, we have derived the expressions of the grid transfer matrices for Finite Elements using Splines of any order. It is found that:

- The grid transfer matrices do not depend of the geometries characterized by the Jacobian as defined in $dV = x^{\alpha} dx$.
- The standard grid transfer operator used for first order finite difference (FD) discretization for Cartesian geometry is recovered when linear Spline finite elements (FE) are used.
- Applying these transfer matrices, we have solved Cartesian, cylindrical as well as periodic one dimensional problems, and obtained essentially the same multigrid iterative performances as found for standard first order FD Cartesian problems.
- No performance *degradation* is observed when the order of Splines FE is increased from 1 to 3, or when the cylindrical geometry is considered.

Cartesian problem with quadratic splines								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	6.203E-02		7.164E-04		2.209E-02		7.164E-04	
1	8.114E-04	0.01	6.375E-06	0.01	1.003E-04	0.00	4.509E-07	0.00
2	1.891E-05	0.02	6.079E-08	0.01	1.769E-06	0.02	8.061E-10	0.00
3	1.103E-06	0.06	5.220E-08	0.86	7.018E-08	0.04	9.970E-11	0.12
4	8.148E-08	0.07	5.220E-08	1.00	5.620E-09	0.08	9.958E-11	1.00
5	6.368E-09	0.08	5.220E-08	1.00	4.772E-10	0.08	9.958E-11	1.00
6	4.969E-10	0.08	5.220E-08	1.00	4.101E-11	0.09	9.958E-11	1.00
7	3.874E-11	0.08	5.220E-08	1.00	3.548E-12	0.09	9.958E-11	1.00
8	3.081E-12	0.08	5.220E-08	1.00	3.081E-13	0.09	9.958E-11	1.00
9	2.489E-13	0.08	5.220E-08	1.00	2.690E-14	0.09	9.958E-11	1.00
10	1.986E-14	0.08	5.220E-08	1.00	3.212E-15	0.12	9.958E-11	1.00

Periodic problem with quadratic splines								
	$N = 128$				$N = 1024$			
V-cycle	$\|r\|_2$	ratio	$\|e\|_2$	ratio	$\|r\|_2$	ratio	$\|e\|_2$	ratio
0	6.203E-02		7.164E-04		2.209E-02		7.164E-04	
1	1.285E-03	0.02	1.294E-05	0.02	3.116E-04	0.01	5.862E-06	0.01
2	7.878E-05	0.06	6.569E-07	0.05	2.893E-05	0.09	1.626E-07	0.03
3	6.573E-06	0.08	6.511E-08	0.10	2.691E-06	0.09	5.631E-09	0.03
4	5.681E-07	0.09	5.224E-08	0.80	2.385E-07	0.09	2.400E-10	0.04
5	4.890E-08	0.09	5.219E-08	1.00	2.097E-08	0.09	9.997E-11	0.42
6	4.198E-09	0.09	5.219E-08	1.00	1.828E-09	0.09	9.958E-11	1.00
7	3.607E-10	0.09	5.219E-08	1.00	1.584E-10	0.09	9.958E-11	1.00
8	3.103E-11	0.09	5.219E-08	1.00	1.370E-11	0.09	9.958E-11	1.00
9	2.674E-12	0.09	5.219E-08	1.00	1.184E-12	0.09	9.958E-11	1.00
10	2.307E-13	0.09	5.219E-08	1.00	1.025E-13	0.09	9.958E-11	1.00

Table 6: The multigrid $V(1,1)$ performance with Gauss-Seidel relation for *Cartesian* problem ($k = 10$, $\sigma = 0$) and *periodic* problem ($k = 10$, $\sigma = 0.01$), discretized on a grid with $N = 128$ and 1024 intervals, using quadratic B-splines. For both grid sizes, a total of 6 grid levels were considered.

For two dimensional problems, notice that for both Cartesian ($dV = dxdy$) and standard curvilinear geometries ($dV = r^\alpha drd\theta$), the Jacobian is *separable.* Using this property, one can show that the two dimensional grid transfer consists of simply applying successively one dimensional grid transfer on each of the x and y (or r and θ) grids. With the solution $\mathbf{u}^h = [u^h_{ij}]$ and right hand side $\mathbf{b}^h = [b^h_{ij}]$ defined by

$$u(x,y) = \sum_{ij} u^h_{ij}\Lambda^h_i(x)\Lambda^h_j(y), \qquad b^h_{ij} = \int dx\Lambda^h_i(x)\int dy\Lambda^h_j(y)f(x,y), \tag{23}$$

the two dimension grid transfers can be expressed as (see Appendix B)

$$\begin{aligned} \mathbf{u}^h &= {}_x\mathbf{P}^h_{2h}\,\mathbf{u}^{2h}\left({}_y\mathbf{P}^h_{2h}\right)^T, \\ \mathbf{b}^{2h} &= {}_x\mathbf{R}^{2h}_h\,\mathbf{b}^h\left({}_y\mathbf{R}^{2h}_h\right)^T. \end{aligned} \tag{24}$$

For more general curvilinear coordinates such as found in tokamak magnetic coordinates defined by $dV = J(s,\theta)dsd\theta$, we will assume that the grid transfer operators derived above are still applicable. The validity of this assumption will be the object of the next task.

A Multigrid Cost Estimation

Assuming that the *coarsest* grid is fixed to 2, the total number of grid levels L is given by $N/2^{L-1} = 2$ or $L = \log_2(N)$, where N is the number of intervals in the *finest grid.* Since both *relaxation* and intergrid transfer are proportional to the number of problem unknowns, the cost of the V-cycle can be estimated as:

$$\begin{aligned} \mathrm{MG}(N) &= c\left[(N+p) + (N/2+p) + \ldots + (N/2^{L-2}+p)\right] \\ &= c\left[2N - 4 + (L-1)p\right], \end{aligned} \tag{25}$$

where p is the order of Splines used for the discretization. The FMG can then be deduced, assuming $\nu_0 = 1$ as

$$\begin{aligned} \mathrm{FMG}(N) &= \mathrm{MG}(N) + \mathrm{MG}(N/2) + \ldots + \mathrm{MG}(N/2^{L-2}) \\ &= c\left[4N - 8 + (L-1)(pL/2 - 4)\right]. \end{aligned} \tag{26}$$

As expected a single FMG cycle (with $\nu_0 = 1$) costs about two V-cycles.

B Two dimensional Grid Transfer

On the fine and the coarse grids, the problem solution $u(x,y)$ can be written as:

$$u(x,y) = \sum_{i'j'} u^h_{i'j'}\Lambda^h_{i'}(x)\Lambda^h_{j'}(y) = \sum_{i'j'} u^{2h}_{i'j'}\Lambda^{2h}_{i'}(x)\Lambda^{2h}_{j'}(y).$$

Projecting these two expansions on the two dimensional basis functions $\Lambda^h_i(x)\Lambda^h_j(y)$ yields

$$\begin{aligned} \sum_{i'j'} u^h_{i'j'} \underbrace{\int dx\Lambda^h_i(x)\Lambda^h_{i'}(x)}_{M^{h,h}_{ii'}}\underbrace{\int dy\Lambda^h_j(x)\Lambda^h_{j'}(y)}_{N^{h,h}_{jj'}} &= \sum_{i'j'} u^{2h}_{i'j'} \underbrace{\int dx\Lambda^h_i(x)\Lambda^{2h}_{i'}(x)}_{M^{h,2h}_{ii'}}\underbrace{\int dy\Lambda^h_j(x)\Lambda^{2h}_{j'}(y)}_{N^{h,2h}_{jj'}} \\ \Longrightarrow \quad \mathbf{M}^{h,h}\,\mathbf{u}^h\,\left(\mathbf{N}^{h,h}\right)^T &= \mathbf{M}^{h,2h}\mathbf{u}^{2h}\left(\mathbf{N}^{h,2h}\right)^T \\ \Longrightarrow \quad \mathbf{u}^h &= \left(\mathbf{M}^{h,h}\right)^{-1}\mathbf{M}^{h,2h}\,\mathbf{u}^{2h}\,\left[\left(\mathbf{N}^{h,h}\right)^{-1}\mathbf{N}^{h,2h}\right]^T. \end{aligned}$$

The right hand side can be written on the fine and coarse grids as

$$
\begin{aligned}
b^h_{ij} &= \int dx\Lambda^h_i(x)\int dy\Lambda^h_j(y)\; f(x,y) = \sum_{i'j'} M^{h,h}_{ii'} f^h_{i'j'} N^{h,h}_{jj'} \quad\Longrightarrow\quad \mathbf{b}^h = \mathbf{M}^{h,h}\,\mathbf{f}^h\left(\mathbf{N}^{h,h}\right)^T, \\
b^{2h}_{ij} &= \int dx\Lambda^{2h}_i(x)\int dy\Lambda^{2h}_j(y)\; f(x,y) = \sum_{i'j'} M^{2h,h}_{ii'} f^h_{i'j'} N^{2h,h}_{jj'} \quad\Longrightarrow\quad \mathbf{b}^{2h} = \mathbf{M}^{2h,h}\,\mathbf{f}^h\left(\mathbf{N}^{2h,h}\right)^T,
\end{aligned}
$$

where the expansion of $f(x,y)$ on the *fine* mesh has been used. Elimination of $\mathbf{f}^h$ then yields

$$
\mathbf{b}^{2h} = \mathbf{M}^{2h,h}\left(\mathbf{M}^{h,h}\right)^{-1}\;\mathbf{b}^h\;\left[\mathbf{N}^{2h,h}\left(\mathbf{N}^{h,h}\right)^{-1}\right]^T.
$$

References

[1] `The Solvers in BSPLINES`, `https://crppsvn.epfl.ch/repos/bsplines/trunk/docs/solvers.pdf`

[2] `BSPLINES` Reference Guide, `https://crppsvn.epfl.ch/repos/bsplines/trunk/docs/bsplines.pdf`

[3] W.L. Briggs, V.E. Henson and S.F. McCormick, A Multigrid Tutorial, Second Edition, Siam (2000).