Quarterly Report Q1-2017

Trach-Minh Tran (SPC-EPFL), Ahmed Ratnani (IPP-Garching).

1 Introduction

The *POMS* project has two main objectives:

- Implement, test and improve the *GLT* smoother [1], in collaboration with NMPP, Max-Planck IPP, Garching, starting from the serial multigrid[2] solver developed during the HLST project *SOLVER++* in 2012.
- Parallelization of the MG+GLT solver, using both *distributed* as well as *shared* memory with `MPI+OpenMP` and `MPI+OpenACC` programming models to be run on clusters of multi-core processors, many integrated core (MIC) processors and GPU devices.

Since this report is the first for this project, some details on the Finite Element Method (FEM) using B-Splines [3], the Geometric Multigrid iterative procedure and GLT Smoother will be presented in sections [2-5] before the first results are presented in section 6.

2 The Splines

We start by defining a finite interval $[a, b]$ subdivided into N_x intervals:

$$a = t_0 \leq t_1 \leq \ldots \leq t_{N_x} = b. \tag{1}$$

The sequence $t_i, i = 0, \ldots, N_x$ can be irregularly spaced. The j^{th} Spline of degree p defined on this sequence of grid points (also called **knots**), is denoted by Λ_j^p and can be constructed using the following recurrence relation [3]. Starting with the *constant* $p = 0$ Spline

$$\Lambda_i^0(x) = \begin{cases} 1 & \text{if } t_i \leq x < t_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

the Splines of degree $p > 0$ for $t_i \leq x < t_{i+1}$ can be constructed from the following recurrence relation

$$\begin{aligned} \Lambda_i^p &= w_i^p \Lambda_i^{p-1} + (1 - w_{i+1}^p)\Lambda_{i+1}^{p-1}, \\ w_i^p &= \frac{x - t_i}{t_{i+p+1} - t_i}. \end{aligned} \tag{3}$$

2.1 Some Properties of Splines

It is well-known that $\left\{\Lambda_1^p, \ldots, \Lambda_{N_x}^p\right\}$ form a basis for the spline space of degree p. Moreover, the B-splines possess the following properties

a1) The Spline Λ_i^p is *strictly positive* in $]t_i, t_{i+p+1}[$: $\Lambda_i^p(x) > 0$,

a2) For $t_0 \leq x \leq t_{N_x}$: $\sum_j \Lambda_j^p(x) = 1$.

a3) The derivative of the Spline of degree p can be expressed in terms of the Spline of degree $p-1$ by:

$$\frac{d}{dx}\Lambda_i^p = p\left(\frac{\Lambda_i^{p-1}}{t_{i+p} - t_i} - \frac{\Lambda_{i+1}^{p-1}}{t_{i+p+1} - t_{i+1}}\right). \tag{4}$$

a4) With the proper normalization, all Splines of all degrees have unitary surface:

$$\frac{p+1}{t_{i+p+1} - t_i}\int \Lambda_i^p(x)dx = 1. \tag{5}$$

a5) Local support property:

$$\text{supp}(\Lambda_i^p) = [t_i, t_{i+p+1}], \quad i = 1, \ldots, N_x,$$

2.2 Boundary Conditions

Applying the recurrence relation to generate *all* the Splines on the finite domain $[t_0, t_{N_x}]$ yields the $N_x + p$ Splines of degree p:

$$\Lambda_{-p}^p, \Lambda_{-p+1}^p, \ldots, \Lambda_{N_x-1}^p. \tag{6}$$

Note that *additional* knots beyond both ends of $[t_0, t_{N_x}]$ have to be defined to generate all these Splines.

2.2.1 Periodic Splines

The extra knots are simply defined through periodicity.:

$$t_{-\nu} = t_{N_x-\nu} - (b-a), \tag{7}$$

$$t_{N_x+\nu} = t_\nu + (b-a), \qquad \nu = 0, \ldots, p. \tag{8}$$

The $p+1$ leftmost Splines in (6) are thus identical to the rightmost Splines:

$$\Lambda_{-\nu}^p = \Lambda_{N_x-\nu}^p, \qquad \nu = 0, \ldots, p. \tag{9}$$

2.2.2 Non-periodic Splines

The choice made in the implementation of these Splines is simply:

$$t_{-p} = \cdots = t_0 = a, \qquad b = t_{N_x} = \cdots = t_{N_x+p}. \tag{10}$$

Using the recurrence relation 3, it can be shown that for the *non-periodic boundary Splines*:

$$\Lambda_r^p(a) = \delta_{r,-p}, \qquad \Lambda_r^p(b) = \delta_{r,N_x-1} \tag{11}$$

2.2.3 Spline expansion

In summary, the approximation of a function f defined in the interval $[a, b]$ using a basis of Splines of degree p associated with the sequence of knots $t_i, i = -p, \ldots, N_x + p$ can be written as

$$f(x) = \sum_{j=-p}^{N_x-1} c_j \Lambda_j^p(x), \qquad \begin{array}{l}\text{support of } \Lambda_j^p\text{: } [t_j, t_{j+p+1}],\\ t_i \leq x < t_{i+1} \Longrightarrow \Lambda_{i-p}^p(x), \ldots, \Lambda_i^p(x) \geq 0.\end{array} \tag{12}$$

Note that the *last* Spline in the interval $[t_i, t_{i+1}]$, which can be written as

$$\Lambda_i^p(x) = w_i^p(x)\Lambda_i^{p-1}(x) = \ldots = w_i^p(x)w_i^{p-1}(x)\ldots w_i^1(x)\Lambda_i^0(x)$$

vanishes at the knot $x = t_i$. Thus at any position x, the sum involves $p+1$ terms except at the knots t_i where there are only p terms.

It is more convenient to *renumber the Spline index* j so that it starts from 0 instead of $-p$. With this new numbering, the Spline expansion becomes

$$f(x) = \sum_{j=0}^{N_x+p-1} c_j \Lambda_j^p(x), \qquad \begin{array}{l}\text{support of } \Lambda_j^p: \quad [t_{j-p}, t_{j+1}],\\ t_i \le x < t_{i+1} \Longrightarrow \Lambda_i^p(x), \ldots, \Lambda_{i+p}^p(x) \ge 0.\end{array} \tag{13}$$

In the *periodic* case, there are N_x *independent* Spline coefficients since

$$c_{N_x+\nu} = c_\nu, \qquad \nu = 0, \ldots, p-1. \tag{14}$$

In the *non-periodic* case, the first and the last Spline coefficients c_0, c_{N_x+p-1} are respectively the values of f at the end points a and b.

The basis functions for both non-periodic and periodic cubic Splines ($p = 3$) are shown in Fig .1 where this new numbering is used.

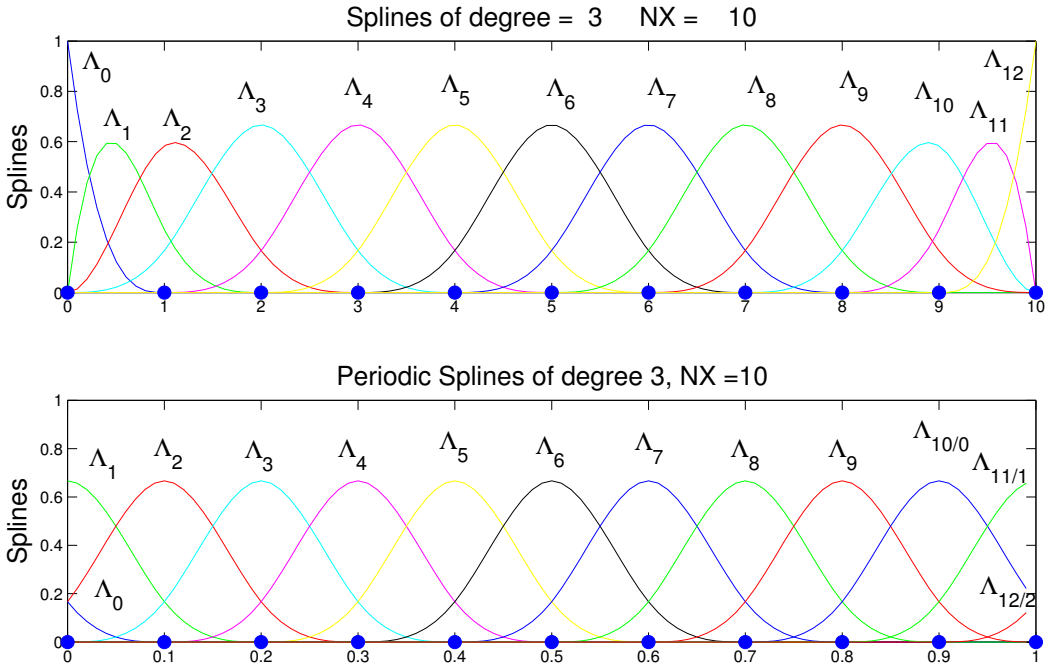

Figure 1: The basis of non-periodic and periodic cubic Splines. The periodic Splines Λ_{10}, Λ_{11}, Λ_{12} denote the same Splines as Λ_0, Λ_1, Λ_2 respectively.

2.3 Special case: Cardinal B-Splines

When the knots vector is $T = \{-p, \ldots, p+1\}$, for a given integer p, the generated spline is known as the *Cardinal Spline* of degree p. Another definition is based on a recursive convolution. In this section, we recall the basic properties of the Cardinal B-Splines, which we will be using in the construction of GLT symbols later.

Definition 2.1 *A cardinal B-spline of zero degree, denoted by ϕ_0, is the characteristic function over the interval $[0,1)$, i.e.,*

$$\phi_0(t) := \begin{cases} 1, & t \in [0,1) \\ 0, & otherwise \end{cases}.$$

A cardinal B-Spline of degree q, $q \in \mathbb{N}$, denoted by ϕ_q, is defined by convolution as

$$\phi_q(t) = (\phi_{q-1} * \phi_0)(t) = \int_{\mathbb{R}} \phi_{q-1}(t-s)\phi_0(s)\, ds.$$

A cardinal B-Spline of degree q has the following properties

b1) Local support property:
$$\mathrm{supp}(\phi_q) = [0, q+1];$$

b2) Regularity: $\phi_q \in \mathcal{C}^{q-1}$;

b3) Derivative expression: $\forall t \in [0, q+1]$ and $q \geq 1$, we have
$$\phi_q'(t) = \phi_{q-1}(t) - \phi_{q-1}(t-1);$$

b4) Recursive definition: $\forall t \in [0, q+1]$ and $q \geq 1$, we have
$$\phi_q(t) = \frac{t}{q}\phi_{q-1}(t) + \frac{q+1-t}{q}\phi_{q-1}(t-1);$$

b5) Symmetry: ϕ_q is symmetric on the interval $[0, q+1]$, i.e.
$$\phi_q(t) = \phi_q(q+1-t), \quad \forall t \in [0, q+1];$$

b6) Inner product:
$$\int_{\mathbb{R}} \phi_{q_1}^{(r_1)}(t)\phi_{q_2}^{(r_2)}(t+\tau)\, dt = (-1)^{r_1}\phi_{q_1+q_2+1}^{(r_1+r_2)}(q_1+1+\tau) = (-1)^{r_2}\phi_{q_1+q_2+1}^{(r_1+r_2)}(q_2+1-\tau), \quad \tau \in \mathbb{R}, \quad q_1, q_2, r_1, r_2 \geq 0.$$

Cardinal B-splines are of interest since the so-called central basis functions Λ_i^p, $i = p+1, \ldots, n$ are uniformly shifted and scaled versions of the cardinal B-splines ϕ_p. More precisely, we have

$$\Lambda_i^p(t) = \phi_p(nt - i + p + 1), \quad i = p+1, \ldots, n, \tag{15}$$

and then

$$\left(\Lambda_i^p(t)\right)' = n\phi_p'(nt - i + p + 1), \quad i = p+1, \ldots, n. \tag{16}$$

2.4 Tensor product B-splines

The definition of B-Splines can be extended to higher dimension, using a tensor product.

Definition 2.2 *For any pair of d-indexes $\boldsymbol{n} = (n_1, n_2, \ldots, n_d)$ and $\boldsymbol{p} = (p_1, p_2, \ldots, p_d)$, let us define the tensor product B-splines as follows*

$$\boldsymbol{\Lambda_i^p} : [0,1]^d \to \mathbb{R}, \quad \boldsymbol{\Lambda_i^p}(\boldsymbol{t}) = \prod_{j=1}^{d} \Lambda_{i_j}^{p_j}(t_j), \quad \boldsymbol{i} = \boldsymbol{1}, \ldots, \boldsymbol{n} + \boldsymbol{p}, \quad \boldsymbol{t} \in [0,1]^d,$$

where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{N}^d$.

For example, the 2D tensor-product B-spline space is defined as

$$\mathcal{S}^{p_1,p_2} := \mathrm{span}\left\{\Lambda_{i_1}^{p_1}(t_1)\Lambda_{i_2}^{p_2}(t_2)\right\}_{i_1,i_2}, \quad i_1 = 1, \ldots, N_1, \quad i_2 = 1, \ldots, N_2. \tag{17}$$

3 The Finite Element Discretization

Let us start with the one-dimensional Sturm-Liouville differential equation:

$$-\frac{d}{dx}\left[C_1(x)\frac{d\phi}{dx}\right] + C_2(x)\phi = \rho,$$

on the domain $0 \leq x \leq L$ with suitable boundary conditions. On a grid with N intervals, the discretized solution ϕ, using the splines $\Lambda_i(x)$ of order p can be written as

$$\phi(x) = \sum_{i=0}^{d-1} u_i \Lambda_i(x), \tag{18}$$

where from section 2.2.3:

$$d = \begin{cases} N & \text{if } \phi \text{ is periodic,} \\ N+p & \text{otherwise,} \end{cases}$$

and u_i are the unknowns of the following matrix equation:

$$\sum_{i'=0}^{d-1} A_{ii'} u_{i'} = b_i, \qquad i = 0, \ldots, d-1. \tag{19}$$

Here the matrix $A_{ii'}$ and the right-hand-side b_i are respectively given by:

$$\begin{aligned} A_{ii'} &= \int_0^L dx\, C_1 \Lambda_i^{'} \Lambda_{i'}^{'} + \int_0^L dx\, C_2 \Lambda_i \Lambda_{i'}, \\ b_i &= \int_0^L dx \rho \Lambda_i. \end{aligned} \tag{20}$$

The *periodicity* can be easily enforced while constructing the right-hand-side b_i and the matrix $A_{ii'}$. This results in a solution ϕ_i which is also N-periodic.

For *non-periodic* problems, the constructed *non-periodic* Splines are such that at the boundaries $x = 0$ and $x = L$:

$$\Lambda_i(0) = \delta_{i,0}, \qquad \Lambda_i(L) = \delta_{i,N+p-1}, \tag{21}$$

which imply that, using (11)

$$\phi(0) = \phi_0, \qquad \phi(L) = \phi_{N+p-1}. \tag{22}$$

It is thus possible to impose the Dirichlet boundary conditions by a simple modification of the matrix $A_{ii'}$. For example to impose the *left BC* $\phi(0) = u_0 = c$:

$$\begin{pmatrix} 1 & 0 & \cdots \\ 0 & A_{11} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N+p-1} \end{pmatrix} = \begin{pmatrix} u_0 \\ b_1 - cA_{1,0} \\ \vdots \\ b_N - cA_{N+p-1,0} \end{pmatrix} \tag{23}$$

It is important to modify the matrix *column* as well in order to preserve the *symmetry* of the original problem.

Notice that the *homogeneous Neumann* BC is already included in the construction of the matrix $A_{o,j}$, but for *non-homogeneous Neumann* BC, additional terms have to be included in the RHS b_i. For example, for the left Neumann BC condition: $\phi' 0) = d$:

$$b_0 \leftarrow b_0 + d \tag{24}$$

For the two-dimensional Helmholtz equation

$$\left[-\nabla\, C_1(x,y) \cdot \nabla + C_2(x,y)\right] \phi = \rho$$

the discretized solution and the right-hand side written as:

$$\begin{aligned} \phi(x,y) &= \sum_{i=1}^{N_x+p_x} \sum_{j=0}^{N_y+p_y} u_{ij}\Lambda_i(x)\Lambda_j(y) \\ b_{ij} &= \int_0^{L_x} dx \int_0^{L_y} dy\, \rho(x,y)\Lambda_i(x)\Lambda_j(y). \end{aligned} \tag{25}$$

The matrix equation to solve is

$$\sum_{i'=1}^{N_x+p_x} \sum_{j'=1}^{N_y+p_y} A_{iji'j'}u_{i'j'} = b_{ij}, \tag{26}$$

with the matrix $A_{iji'j'}$ expressed as

$$A_{iji'j'} = \int_0^{L_x}\int_0^{L_y} dxdy \left[C_1(x,y)\left(\nabla\Lambda_i^{p_x}(x)\Lambda_{i'}^{p_y}(y) \cdot \nabla\Lambda_j^{p_x}(x)\Lambda_{j'}^{p_y}(y)\right) + C_2(x,y)\,\Lambda_i^{p_x}(x)\Lambda_j^{p_y}(y)\,\Lambda_{i'}^{p_x}(x)\Lambda_{j'}^{p_y}(y)\right]. \tag{27}$$

The same technique used to impose the BC previously in the 1D problem can be extended in a straightforwards manner to this 2D case.

4 The Multigrid Method

4.1 The Multigrid V and W cycle

Let's denote the solution $\mathbf{u}^h$ and right hand side $\mathbf{b}^h$ from the matrix equation

$$\mathbf{A^h u^h} = \mathbf{b^h}, \tag{28}$$

defined at some grid level represented by the grid spacing h. For the 2D problem, the matrix $\mathbf{A^h} = (A^h_{kk'})$ and the vector $\mathbf{u^h} = (u^h_k)^t$ are formed with the index k defined by the *lexicographic numbering*

$$k = (j-1)(N_x + p_x) + i. \tag{29}$$

In `Fortran 2008` it is possible to map an array pointer to an array of *different* rank, for example:

```
1  REAL(rkind), ALLOCATABLE, TARGET :: v(:,:)
2  REAL(rkind), POINTER             :: v1d(:) => NULL()
3  ...
4  ALLOCATE(v(m,n))
5  v1d(1:m*n) => v
```

Using this feature, one needs to modify only either v or $v1d$.

The following MG algorithm for the $V(\nu_1,\nu_2)$ can be described as follows

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

computes a *new* $\mathbf{u}^h$. It is defined *recursively* by the following steps:

1. If h is the coarsest mesh size,
   - Direct solve $\mathbf{A}^h\mathbf{u}^h = \mathbf{b}^h$
   - Goto 3.
2. Else

- Relax $\mathbf{u}^h$ ν_1 times.
- $\mathbf{b}^{2h} \leftarrow \mathbf{R}(\mathbf{b}^h - \mathbf{A}^h\mathbf{u}^h)$.
- $\mathbf{u}^{2h} \leftarrow MG^{2h}(\mathbf{u}^{2h}, \mathbf{b}^{2h})$ μ times.
- $\mathbf{u}^h \leftarrow \mathbf{u}^h + \mathbf{P}\mathbf{u}^{2h}$.
- Relax $\mathbf{u}^h$ ν_2 times.

3. Return

The standard $V(\nu_1, \nu_2)$ cycle is obtained by calling this MG^h procedure with $\mathbf{b}^h$ defined at the *finest* grid level, a guess $\mathbf{u}^h = 0$ and $\mu = 1$. With $\mu > 1$ results in the $W_\mu(\nu_1, \nu_2)$ cycle.

Only the weighted Jacobi and the Gauss-Seidel are presently implemented.

4.2 Grid coarsening

Let start with the one-dimensional *fine* grid defined by x_i, $i = 0, \ldots, N$, assuming that N is even. The next coarse grid (with $N/2$ intervals) is obtained by simply discarding the grid points with *odd* indices.

In order to get a *smallest coarsest* grid (so that it is possible to solve *cheaply* the problem with a *direct* method), N should be $N = N_c 2^{L-1}$ where L the total number of grid levels and N_c is either 2 or a *small odd* integer. As an example, the fine grid with $N = 768$ can have up to 9 grid levels, and a coarsest grid with 3 intervals, see Table 1.

L	N				
1	2	3	5	7	9
2	4	6	10	14	18
3	8	12	20	28	36
4	16	24	40	56	72
5	32	48	80	112	144
6	64	96	160	224	288
7	128	192	320	448	576
8	256	384	640	896	1152
9	512	768	1280	1792	2304
10	1024	1536	2560	3584	4608

Table 1: Set of values of the *fine* grid number of intervals N to obtain a *coarsest* grid size at most equal to 9 with at most 10 grid levels.

For a two-dimensional grid, the same procedure is applied to both dimensions. The result of such procedure is illustrated in Fig. 2, for a 8×4 fine grid.

4.3 The Grid Transfer Operators

In the MG procedure described above, the operators $\mathbf{R}$ and $\mathbf{P}$ denote respectively the *restriction* (from *fine* to *coarse* grid) and the *prolongation* (from *coarse* to *fine* grid). Notice that in this multigrid procedure, $\mathbf{R}$ is only needed to restrict the *right hand side* while $\mathbf{P}$ is used only to prolong (or interpolated) the *solution*.

4.3.1 The Restriction Operator

Noting that the basis functions $\Lambda_i^{2h}(x)$, which are *piecewise* C^{p-1} polynomials with *breaks* on the *coarse* grid points $x_k^{2h} = (2h)k$ can be also considered as *piecewise* C^{p-1} polynomials with *breaks* on the *fine* grid

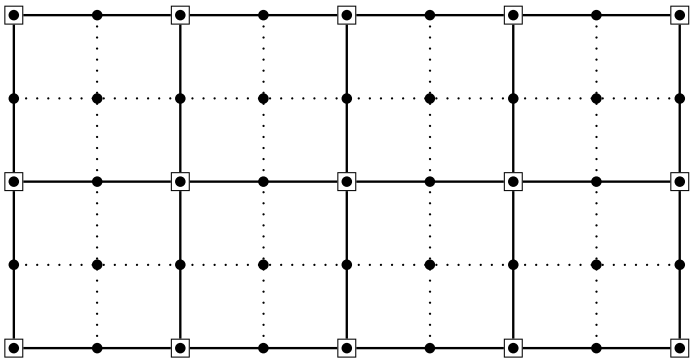

Figure 2: A *coarse* 4×2 grid ($\square$) obtained from a 8×4 fine grid ($\bullet$).

$x_k^h = kh$, they can be expressed *uniquely* as a linear combination of the *fine* grid basis functions:

$$\Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} c_{ii'} \Lambda_{i'}^h(x), \quad i = 1, \ldots, N/2 + p. \tag{30}$$

The (rectangular) matrix $c_{ii'}$ can be identified as the one-dimensional *restriction* $\mathbf{R}^x$ since

$$b_i^{2h} = \int dx\, \rho(x) \Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} c_{ii'}\ b_{i'}^h = \sum_{i'=1}^{N+p} R_{ii'}^x\ b_{i'}^h.$$

It can be computed by simply projecting Eq.(30) on the fine grid basis function $\Lambda_j^h(x)$:

$$\sum_{i'=1}^{N+p} R_{ii'}^x \underbrace{\int dx \Lambda_{i'}^h(x) \Lambda_j^h(x)}_{M_{i'j}^h} = \underbrace{\int dx \Lambda_i^{2h}(x) \Lambda_j^h(x)}_{M_{ij}^{2h,h}} \Longrightarrow \mathbf{R}^x = \mathbf{M}^{2h,h} \cdot (\mathbf{M}^h)^{-1}. \tag{31}$$

It should be stressed that the representation for $\Lambda_i^{2h}(x)$ in Eq.(30) is *unique*. This is checked by verifying that the same matrix $R_{ii'}^x$ is obtained by solving the *interpolation problem*, Eq.(30) on the $N/2 + p$ interpolation sites x_k for *odd* p and $x_{k+1/2} = (x_k + x_{k+1})$ for *even* p.

Denoting the restriction on x and y respectively by $\mathbf{R}^x$ and $\mathbf{R}^y$, the *two-dimensional restriction* of b_{ij}^h is defined as

$$b_{ij}^{2h} = \iint dxdy\, \rho(x,y) \Lambda_i^{2h}(x) \Lambda_j^{2h}(y) = \sum_{i'=1}^{N_x+p_x} \sum_{j'=1}^{N_y+p_y} R_{ii'}^x R_{jj'}^y b_{i'j'}^h,$$

and thus

$$\boxed{\mathbf{b}^{2h} = \mathbf{R}^x \cdot \mathbf{b}^h \cdot (\mathbf{R}^y)^T.} \tag{32}$$

4.3.2 The Prolongation Operator

Let denote the discretized solution on the coarse mesh of

$$\mathbf{A}^{2h}\mathbf{u}^{2h} = \mathbf{b}^{2h} = \mathbf{R}_h^{2h}\mathbf{b}^h$$

by

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

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

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

by *prolongation* of $\mathbf{u}^{2h}$ (and *not* by solving $\mathbf{A}^h\mathbf{u}^h = \mathbf{b}^h$). A reasonable solution is to *minimize* the square of the error norm defined as

$$\epsilon^2 = \|\tilde{\phi}^h(x) - \phi^{2h}(x)\|^2 \equiv \int [\tilde{\phi}^h(x) - \phi^{2h}(x)]^2\, dx,$$

$$\frac{\partial \epsilon^2}{\partial \tilde{u}_i^h} = 0 \Longrightarrow \sum_{i'=1}^{N+p} \tilde{u}_i^h \underbrace{\int \Lambda_i^h \Lambda_{i'}^h\, dx}_{M_{ii'}^h} = \sum_{i'=1}^{N/2+p} u_i^{2h} \underbrace{\int \Lambda_i^h \Lambda^{2h} dx_{i'}}_{M_{ii'}^{h,2h}}.$$

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

$$\tilde{\mathbf{u}}^h = \mathbf{P}^x \mathbf{u}^{2h}, \qquad \boxed{\mathbf{P}^x = (\mathbf{M}^{h,h})^{-1}\mathbf{M}^{h,2h} = (\mathbf{R}^x)^T} \tag{33}$$

It should be noted here that, while the restricted right hand side $\mathbf{b}^{2h}$ as defined in Eq.(32) is *exactly identical* to the assembled right hand side, the prolonged solution $\tilde{\mathbf{u}}^h$ defined in Eq.(33) is just a representation of $\phi^{2h}(x)$ on the fine mesh and *not* the solution $\phi^h(x)$ which can only be obtained by solving the problem on the fine mesh!

The well-known *linear interpolation* scheme is recovered by computing numerically the prolongation 9×5 matrix with (33), using *linear* Splines and $N = 8$ as shown in

$$\mathbf{P}^x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{34}$$

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

$$\mathbf{P}_{2h}^h = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3/4 & 1/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3/16 & 11/16 & 1/8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/8 & 3/4 & 1/8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/8 & 3/4 & 1/8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/8 & 11/16 & 3/16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{35}$$

From (32) and (33):

$$\mathbf{A}^{2h}\mathbf{u}^{2h} = \mathbf{b}^{2h} = \mathbf{R}^x\mathbf{b}^h = \mathbf{R}^x\mathbf{A}^h\mathbf{u}^h = \mathbf{R}^x\mathbf{A}^h\mathbf{P}^x\mathbf{u}^{2h} \Longrightarrow \boxed{\mathbf{A}^{2h} = \mathbf{R}^x\mathbf{A}^h\mathbf{P}^x} \tag{36}$$

Generalization to the two-dimensional case is straightforward by using the *Kronecker product* to calculate the two-dimensional restriction and prolongation operators. This yields:

$$\mathbf{A}^{2h} = \mathbf{R}\mathbf{A}^h\mathbf{P}, \tag{37}$$

$$\mathbf{R} = \mathbf{R}^y \otimes \mathbf{R}^x, \tag{38}$$

$$\mathbf{P} = \mathbf{P}^y \otimes \mathbf{P}^x = \mathbf{R}^T, \tag{39}$$

with the matrices $\mathbf{A}^h$, $\mathbf{A}^{2h}$ assembled using the numbering defined in (29).

5 The GLT Smoother

In this section we will introduce some preliminary approximation and spectral tools. Together with the relevant properties of the (cardinal) B-spline as collected in subsection 2.3, subsection 5.1 and Appendix A are devoted to spectral notions of multilevel block Toeplitz matrices and Generalized Locally Toeplitz (GLT) sequences, respectively. We will end this section collecting some spectral results on the matrices involved in the discretization of 1D and 2D elliptic problems which serve in section 6 as tests for the GLT *smoother*.

5.1 Unilevel and multilevel Toeplitz matrix-sequences

Definition 5.1 *A (unilevel) Toeplitz matrix is a real/complex valued $n \times n$ matrix $T_n = [t_{ij}]_{i,j=1}^n$, where $t_{ij} = t_{i-j}$, i.e.,*

$$T_n = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \dots & t_{-(n-1)} \\ t_1 & t_0 & t_{-1} & \dots & \\ t_2 & t_1 & t_0 & \dots & \vdots \\ \vdots & & & \ddots & \\ t_{n-1} & \dots & & \dots & t_0 \end{pmatrix}.$$

For any function $f \in L^1([-\pi,\pi])$, the existence of the Fourier series leads to

$$f(\theta) \sim \sum_{j\in\mathbb{Z}} \hat{f}_j e^{\mathbf{i}j\theta}, \quad \forall\theta \in [-\pi,\pi],$$

where

$$\hat{f}_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-\mathbf{i}j\theta}\, d\theta,$$

hence, the sequence $\{\hat{f}_j\}_j$ determines uniquely the function f and viceversa. Therefore, the function f, if it exists, is also uniquely determined by the sequence of the Toeplitz matrices $\{T_n(f)\}_n$ with

$$T_n(f) = \left[\hat{f}_{i-j}\right]_{i,j=1}^n.$$

When the function $f \in L^1([-\pi,\pi]^d)$, the associated sequence is made of the so called *multilevel Toeplitz matrices*, that is matrices which 'at each level' are Toeplitz matrices. For example, a 2-level matrix is a block Toeplitz whose blocks are still Toeplitz. A more general definition of Toeplitz sequences associated to a function is obtained when f is a matrix-valued function $f : [-\pi,\pi]^d \to \mathbb{C}^{s\times s}$ such that all its components $f_{ij} : [-\pi,\pi]^d \to \mathbb{C},\ i,j = 1,\dots,s$, belong to $L^1([-\pi,\pi]^d)$. In this case the associated sequence is made of the so called *multilevel block Toeplitz matrices*, that is multilevel Toeplitz matrices whose entries 'at the last level' are $s \times s$ matrices themselves.

Let $\boldsymbol{n} := (n_1,\dots,n_d)$ be a multi-index in $\mathbb{N}^d$ and set $N(\boldsymbol{n}) := \prod_{i=1}^d n_i$. The formal definition of d-level block Toeplitz sequence associated to f is the following.

Definition 5.2 *Let $f : [-\pi,\pi]^d \to \mathbb{C}^{s\times s}$ such that $f_{ij} \in L^1([-\pi,\pi]^d)$, $i,j = 1,\dots,s$, and let $\hat{f}_{\boldsymbol{j}}$ be its Fourier coefficients*

$$\hat{f}_{\boldsymbol{j}} := \frac{1}{(2\pi)^d}\int_{[-\pi,\pi]^d} f(\boldsymbol{\theta})\mathrm{e}^{-\mathbf{i}\langle \boldsymbol{j},\boldsymbol{\theta}\rangle}\, d\boldsymbol{\theta} \in \mathbb{C}^{s\times s}, \quad \boldsymbol{j} = (j_1,\dots,j_d) \in \mathbb{Z}^d, \quad \boldsymbol{\theta} = (\theta_1,\dots,\theta_d) \in [-\pi,\pi]^d, \tag{40}$$

where $\langle \boldsymbol{j},\boldsymbol{\theta}\rangle = \sum_{r=1}^d j_r\boldsymbol{\theta}_r$ and the integrals in (40) are computed componentwise. Then, the $\boldsymbol{n}$th Toeplitz matrix associated with f is the matrix of order $sN(\boldsymbol{n})$ given by

$$T_{\boldsymbol{n}}(f) = \left[\hat{f}_{\boldsymbol{i}-\boldsymbol{j}}\right]_{\boldsymbol{i},\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{n}} = \sum_{|j_1|<n_1} \cdots \sum_{|j_d|<n_d} \left[J_{n_1}^{(j_1)} \otimes \cdots \otimes J_{n_d}^{(j_d)}\right] \otimes \hat{f}_{\boldsymbol{j}},$$

where $\mathbf{1} = (1,\dots,1) \in \mathbb{N}^d, \boldsymbol{i} = (i_1,\dots,i_d) \in \mathbb{N}^d, \boldsymbol{j} = (j_1,\dots,j_d) \in \mathbb{N}^d$ and $\otimes$ denotes the (Kronecker) tensor product of matrices. The term $J_m^{(l)}$ is the matrix of order m whose (i,j) entry equals 1 if $i-j=l$ and zero otherwise. The set $\{T_{\boldsymbol{n}}(f)\}_{\boldsymbol{n}}$ is called the family of d-level block Toeplitz matrices generated by f, *that in turn is referred to as the* generating function or the symbol of $\{T_{\boldsymbol{n}}(f)\}_{\boldsymbol{n}}$.

5.2 B-Splines mass and stiffness matrices

Let us consider the following mass and stiffness matrices

$$M_n^p = \left[\int_0^1 \Lambda_{i_1}^p(t)\ \Lambda_{j_1}^p(t)\ dt\right]_{i_1,j_1=1}^{n+p}, \tag{41a}$$

$$S_n^p = \left[\int_0^1 \left(\Lambda_{i_1}^p(t)\right)'\ \left(\Lambda_{j_1}^p(t)\right)'\ dt\right]_{i_1,j_1=1}^{n+p}. \tag{41b}$$

Remark 5.1 *From [4] we know that the matrices M_n^p and S_n^p are Symmetric Positive Definite (SPD) matrices.*

Using the results of Subsection 2.3, these matrices up to a low-rank perturbation write as

$$(M_n^p)_{i_1j_1} = \frac{1}{n}\ \phi_{2p+1}(p+1-(i_1-j_1)), \tag{42a}$$

$$(S_n^p)_{i_1j_1} = -\,n\phi''_{2p+1}(p+1-(i_1-j_1)), \tag{42b}$$

that is, they are low-rank perturbations of Toeplitz matrices. Thanks to the results in Subsection A, the following theorems on the symbol of the mass and stiffness matrices in (41a)-(41b) hold.

Theorem 5.2 *We have $\{nM_n^p\}_n \sim_{\mathrm{GLT}} \mathfrak{m}_p$ and $\{nM_n^p\}_n \sim_{\sigma,\lambda} (\mathfrak{m}_p, [-\pi,\pi])$, where the symbol $\mathfrak{m}_p$ is given by*

$$\mathfrak{m}_p(x,\theta) := \mathfrak{m}_p(\theta) = \phi_{2p+1}(p+1) + 2\sum_{k=1}^p \phi_{2p+1}(p+1-k)\cos(k\theta). \tag{43}$$

Theorem 5.3 *We have $\{\frac{1}{n}S_n^p\}_n \sim_{\mathrm{GLT}} \mathfrak{s}_p$ and $\{\frac{1}{n}S_n^p\}_n \sim_{\sigma,\lambda} (\mathfrak{s}_p, [-\pi,\pi])$, where the symbol $\mathfrak{s}_p$ is given by*

$$\mathfrak{s}_p(x,\theta) := \mathfrak{s}_p(\theta) = -\phi''_{2p+1}(p+1) - 2\sum_{k=1}^p \phi''_{2p+1}(p+1-k)\cos(k\theta). \tag{44}$$

The symbols $\mathfrak{m}_p(\theta)$ and $\mathfrak{s}_p(\theta)$ satisfy the following properties for all $p \geq 1$ and $\theta \in [-\pi,\pi]$ (see [4, 5]):

c1) $\mathfrak{s}_p(\theta) = \mathfrak{m}_{p-1}(\theta)(2 - 2\cos(\theta))$;

c2) Let $M_{\mathfrak{s}_p} = \max_{[0,\pi]} \mathfrak{s}_p(\theta)$. Then

$$\frac{\mathfrak{s}_p(\pi)}{M_{\mathfrak{s}_p}} \leq 2^{2-p},$$

which means that $\frac{\mathfrak{s}_p(\pi)}{M_{\mathfrak{s}_p}}$ decreases exponentially to zero as $p \to \infty$;

c3) $\left(\frac{4}{\pi^2}\right)^p \leq \mathfrak{m}_p(\theta) \leq \mathfrak{m}_p(0) = 1$.

Remark 5.4 *As a result of previous properties $\mathfrak{s}_p(\theta)$ has a unique zero of order 2 at 0 (like the function $2-2\cos(\theta)$). On the other hand, from a numerical point of view, we can say that, for large p, the normalized symbol $\frac{\mathfrak{s}_p(\theta)}{M_{\mathfrak{s}_p}}$ has also an exponential numerical zero at $\theta = \pi$.*

6 First Serial Results with GLT Smoother

In this section, 1D and 2D test problems are considered. The resulting discretized matrix equation in both tests is assumed to include only a *stiffness* matrix:

$$\mathbf{Au} = \mathbf{Su} = \mathbf{b} \tag{45}$$

When high order Splines are used, it can be shown theoretically [1] and observed (see below) that the *convergence factor* of the standard multigrid method is close to 1, with the number of iterations increasing quickly with the order p of Splines. In order to solve this problem, a GLT *post smoother* is implemented as an additional step to be added at the end of each multigrid V cycle, consisting of performing $\nu_{\text{psm}} > 0$ iterations of Preconditioned Conjugate Gradient (PCG) or Generalized Minimum Residual (GMRES) [6] on the equivalent system

$$\mathbf{P}^{-1}\mathbf{Au} = \mathbf{P}^{-1}\mathbf{b}. \tag{46}$$

The precondition matrix $\mathbf{P}$ is the Toeplitz stiffness matrix (42b). Either *point Jacobi* or *SSOR* methods [6] are used to invert $\mathbf{P}$.

6.1 The 1D problem

The following second-order boundary value problem is considered:

$$-\frac{d^2}{dx^2}\phi = \rho, \quad 0 \leq x \leq L_x, \qquad \phi(0) = \phi(L_x) = 0, \tag{47}$$

where the right hand side ρ is computed assuming the *exact solution*

$$\phi_{ex}(x) = \sin\left(\frac{\pi k_1 x}{L_x}\right) + \sin\left(\frac{\pi k_2 x}{L_x}\right).$$

The *residual norm* r and the *discretization error* e are defined as:

$$r = \|\mathbf{b} - \mathbf{Au}\|_2 = \sqrt{\sum_i \left(b_i - \sum_{i'} A_{ii'} u_{i'}\right)^2}, \tag{48}$$

$$e = \|\phi - \phi_{ex}\|_2 = \sqrt{\int_0^{L_x} dx \left[\sum_i u_i \Lambda_i(x) - \phi_{ex}(x)\right]^2}. \tag{49}$$

ν_{psm}	$p=10$	$p=9$	$p=8$	$p=7$	$p=6$	$p=5$	$p=4$	$p=3$
0	396	174	83	41	21	11	6	5
1	94	74	29	35	16	9	5	4
2	53	30	16	12	8	6	4	4
3	40	24	14	10	8	5	4	3
4	27	19	12	8	5	4	3	3
5	20	11	7	5	4	4	3	3
6	15	9	5	4	4	4	3	3
7	13	8	5	4	4	3	3	3
8	9	5	4	3	3	3	3	3
9	6	4	3	3	3	3	3	3
10	7	4	3	3	3	3	3	3
11	6	4	3	3	3	3	3	3
12	5	3	3	3	3	3	3	3
13	4							
14	4							
15	4							
16	4							

Table 2: The number of iterations required for tol $= 10^{-12}$, using the `PCG` smoother for $N_x = 2048$, $k_1 = 3$, $k_2 = 400$. The precondition matrix is inverted using `SSOR` with an relaxation $\omega = 1.6$. The number of grid levels is 7.

ν_{psm}	$p=10$	$p=9$	$p=8$	$p=7$	$p=6$	$p=5$	$p=4$	$p=3$
0	396	174	83	41	21	11	6	5
3	136	54	26	17	10	7	4	3
6	59	34	16	9	6	4	3	3
9	44	21	11	7	4	3		3
12	32	15	9	5	4	3		
15	23	11	7	5	3	3		
18	15	9	6	4	3			
21	13	8	5	3	3			
24	12	7	5	3				
27	10	6	4	3				
30	9	6	4	3				
33	8	5	3					
36	7	5	3					
39	7	4						
42	7	4						
45	6	4						
48	6	4						
51	5	3						
54	5	3						
57	5	3						
60	4	3						

Table 3: The number of iterations required for tol $= 10^{-12}$, using the `PCG` smoother for $N_x = 2048$, $k_1 = 3$, $k_2 = 400$. The precobdition matrix is inverted using `Jacobi`. The number of grid levels is 7.

The integral in the definition of e is computed using the same Gauss quadrature as in the assembling of $\mathbf{A}$ and $\mathbf{b}$. The iterative MG process is stopped when $r^{(k)}/r^{(0)}$ decreases to less than $\texttt{tol} = 10^{-12}$. The *correctness* of the iterative MG is checked against the discretization error e obtained by solving the linear system using the direct Lapack solver `dgbtrf/dgbtrs`.

Using the Gauss-Seidel MG relaxations, the required number of $V(2,2)$ cycles versus the Spline degree p and the number of `GLT` *post smoother* sweeps ν_{psm} are shown in Table 2 when a `SSOR` *Precondition Conjugate Gradient* (PCG) is used for the `GLT` smoother and in Table 3 when the `Jacobi` PCG is used instead. A slight decrease in performance of the `GLT` post smoother can be observed in the latter case. In both cases, the optimum ν_{psm} increases almost linearly with p.

6.2 The 2D problem

The following second order boundary value problem is considered:

$$\begin{aligned} -\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right]\phi &= \rho \qquad 0 \le x \le L_x, \quad 0 \le y \le L_y, \\ u(0,y) = u(L_x,y) &= u(x,0) = u(x,L_y) = 0, \end{aligned} \tag{50}$$

where the right hand side ρ is computed assuming the *exact solution*

$$\phi_{ex}(x,y) = \sin\left(\frac{\pi k_x x}{L_x}\right)\sin\left(\frac{\pi k_y y}{L_y}\right).$$

The *residual norm* r and the *discretization error* e are computed as

$$r = \|\mathbf{b}-\mathbf{A}\mathbf{u}\|_2, \tag{51}$$

$$e = \|\phi-\phi_{ex}\|_2 = \sqrt{\iint dxdy\left[\sum_{ij} u_{ij}\Lambda_i(x)\Lambda_j(y)-\phi_{ex}(x,y)\right]^2}. \tag{52}$$

The double integral in the definition of e is computed using the same Gauss quadrature as in the assembling of $\mathbf{A}$ and $\mathbf{b}$. The iterative MG process is stopped when $r^{(k)}/r^{(0)}$ decreases to less than $\texttt{tol} = 10^{-12}$. The *correctness* of the iterative MG is checked against the discretization error e obtained by solving the linear system using the direct sparse `MUMPS` solver.

Using the Gauss-Seidel MG relaxations, the required number of $V(4,4)$ cycles versus the Spline degree p and the number of `GLT` *post smoother* sweeps ν_{psm} are shown in Table 4 when a `PCG` is used for the `GLT` smoother. The `GMRES` has been also tested with the same problem, although the stiffness matrix $\mathbf{A}$ and the precondition matrix $\mathbf{P}$ are both *SPD*. It is reported in Table 5 and shows a clear decrease of the required number of iterations. for $p > 6$!

Finally, to assess the performance of the GLT smoother using `PCG` and `GMRES`, the gain in the whole solver's elapsed time versus the number of smoother sweeps is reported in Fig.(3) for $p = 6, 8$ and 10. The convergence acceleration seems to be much improved by using for $p > 6$ `GMRES` instead of `PCG`.

7 Conclusions

- First tests show that the GLT post smoother can solve the convergence problem of the *plain* FEM MG solver for large Spline's degree p. Both `PCG` and `GMRES` are implemented for the GLT smoother. For large p, it is observed that GMRES implementation of the GLT smoother is more efficient than `PCG`.
- The present *serial* MG+GLT implementation is ready to be parallelized as planned.

ν_{psm}	$p=10$	$p=9$	$p=8$	$p=7$	$p=6$	$p=5$	$p=4$	$p=3$
0	>2000	1993	1160	1118	298	106	28	9
4	>2000	>2000	1302	242	51	18	7	4
8	>2000	1160	796	160	35	10	4	3
12	1892	1584	304	95	23	7	3	3
16	>2000	823	379	76	17	5	3	2
20	1346	540	288	55	12	5	3	2
24	668	863	166	42	10	4	2	
28	856	793	121	35	9	4		
32	1079	673	137	28	7	4		
36	863	468	100	24	6	3		
40	417	323	88	21	6	3		
50	585	287	66	15	5	3		
60	456	243	52	12	4	2		
70	366	178	40	10	4	2		

Table 4: The number of iterations required for tol $= 10^{-12}$, using a `PCG` smoother, for a 256×256 2D grid and $\mathbf{k} = (3, 40)$. The number of grid levels is 7.

ν_{psm}	$p=10$	$p=9$	$p=8$	$p=7$	$p=6$	$p=5$	$p=4$	$p=3$
0	>2000	1993	1160	1118	298	106	28	9
5	>2000	1629	849	226	49	15	7	3
10	1310	642	411	98	26	8	4	3
15	852	298	249	63	15	6	3	2
20	575	200	155	43	10	5	2	2
25	421	134	109	30	9	4	2	2
30	329	93	62	23	8	3	2	2
35	245	75	56	18	7	3	2	2
40	213	87	39	16	6	3	2	2
45	180	58	27	13	5			
50	146	50	25	12	5			
55	98	56	24	12	4			
60	91	56	22	10	4			
65	68	43	21	9	4			
70	59	42	19	8	3			

Table 5: The number of iterations required for tol $= 10^{-12}$, using a `GMRES` smoother, for a 256×256 2D grid and $\mathbf{k} = (3, 40)$. The number of grid levels is 7.

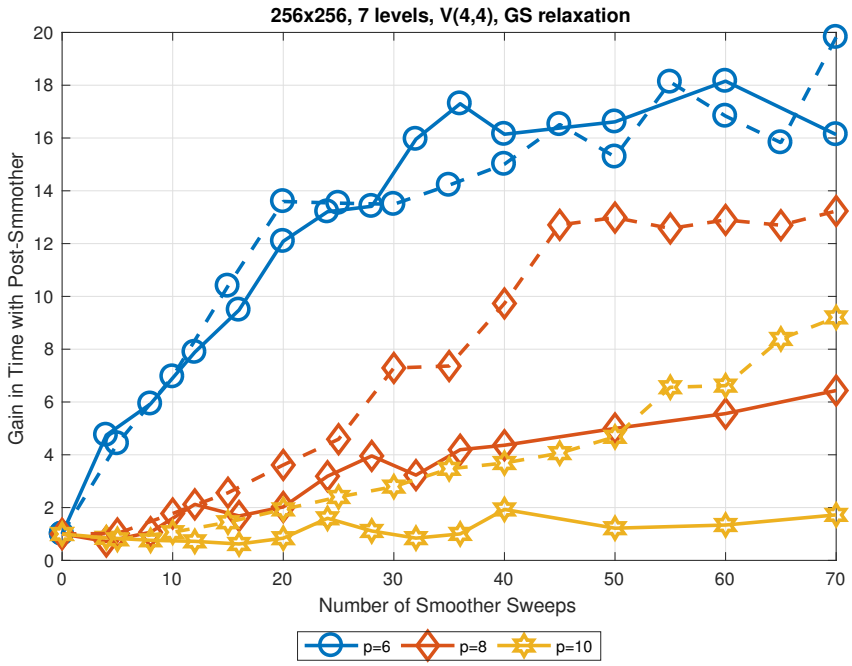

Figure 3: The gain in time provided by the GLT post smoother versus the number of smoother sweeps ν_{psm} for different degrees of Splines p. Solid and dashed lines show the results obtained with `PCG` and `GMRES` respectively.

References

[1] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers, "Robust and optimal multi-iterative techniques for iga galerkin linear systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 230 – 264, 2015. Isogeometric Analysis Special Issue.

[2] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial*. Miscellaneous Bks, Society for Industrial and Applied Mathematics, 2000.

[3] C. de Boor, *A Practical Guide to Splines*. Applied Mathematical Sciences, Springer New York, 2001.

[4] C. Garoni, C. Manni, F. Pelosi, S. Serra-Capizzano, and H. Speleers, "On the spectrum of stiffness matrices arising from isogeometric analysis," *Numerische Mathematik*, vol. 127, no. 4, pp. 751–799, 2014.

[5] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers, "Symbol-based multigrid methods for galerkin b-spline isogeometric analysis," *SIAM Journal on Numerical Analysis*, vol. 55, no. 1, pp. 31–62, 2017.

[6] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. Philadelphia, PA: SIAM, 1994.

[7] P. Tilli, "Locally toeplitz sequences: spectral properties and applications," *Linear Algebra and its Applications*, vol. 278, no. 1, pp. 91 – 120, 1998.

[8] S. Serra-Capizzano, "The glt class as a generalized fourier analysis and applications," *Linear Algebra and its Applications*, vol. 419, no. 1, pp. 180 – 233, 2006.

[9] S. Capizzano, "Generalized locally toeplitz sequences: spectral analysis and applications to discretized partial differential equations," *Linear Algebra and its Applications*, vol. 366, pp. 371 – 402, 2003.

[10] U. Grenander and G. Szegő, *Toeplitz Forms and Their Applications*. California monographs in mathematical sciences, University of California Press, 1958.

[11] P. Tilli, "A note on the spectral distribution of toeplitz matrices," *Linear and Multilinear Algebra*, vol. 45, no. 2-3, pp. 147–159, 1998.

[12] L. Golinskii and S. Serra-Capizzano, "The asymptotic properties of the spectrum of nonsymmetrically perturbed jacobi matrix sequences," *Journal of Approximation Theory*, vol. 144, no. 1, pp. 84 – 102, 2007.

A Summary of the theory of GLT sequences

In the sequel, we recall the basic properties of the Generalized Locally Toeplitz sequences. More details can be found in the pioneering work [7] by Tilli focused on the spectrum of one-dimensional differential operators and in [8, 9] containing a generalization to multivariate differential operators.
As described in [8, 9], a GLT sequence $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ is a sequence of matrices of increasing size. Before listing some crucial properties of the GLT sequences, we need to introduce the definition of spectral distribution in the sense of the eigenvalues and of the singular values for a generic matrix-sequence $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$.

Definition A.1 *Let $f : G \to \mathbb{C}^{s\times s}$ be a measurable function, defined on a measurable set $G \subset \mathbb{R}^l$ with $l \geq 1$, $0 < m_l(G) < \infty$, where m_l is the Lebesgue measure. Let $\mathcal{C}_0(\mathbb{K})$ be the set of continuous functions with compact support over $\mathbb{K} \in \{\mathbb{C}, \mathbb{R}_0^+\}$ and let $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ be a sequence of matrices with $\dim(A_{\boldsymbol{n}}) = d_{\boldsymbol{n}}$ and $d_{\boldsymbol{n}} \to \infty$ as $\boldsymbol{n} \to \infty$, i.e. $n_j \to \infty$, $j = 1, \ldots, d$.*

- $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ *is* distributed as the pair (f, G) in the sense of the eigenvalues, *in symbols*

$$\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\lambda (f, G),$$

if the following limit relation holds for all $F \in \mathcal{C}_0(\mathbb{C})$:

$$\lim_{\boldsymbol{n}\to\infty} \frac{1}{d_{\boldsymbol{n}}} \sum_{j=1}^{d_{\boldsymbol{n}}} F(\lambda_j(A_{\boldsymbol{n}})) = \frac{1}{m_l(G)} \int_G \frac{\mathrm{tr}(F(f(t)))}{s} dt, \tag{53}$$

where $\lambda_j(A_{\boldsymbol{n}})$, $j = 1, \ldots, d_{\boldsymbol{n}}$ are the eigenvalues of $A_{\boldsymbol{n}}$.

- $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ *is* distributed as the pair (f, G) in the sense of the singular values, *in symbols*

$$\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\sigma (f, G),$$

if the following limit relation holds for all $F \in \mathcal{C}_0(\mathbb{R}_0^+)$:

$$\lim_{\boldsymbol{n}\to\infty} \frac{1}{d_{\boldsymbol{n}}} \sum_{j=1}^{d_{\boldsymbol{n}}} F(\sigma_j(A_{\boldsymbol{n}})) = \frac{1}{m_l(G)} \int_G \frac{\mathrm{tr}(F(|f(t)|))}{s} dt, \tag{54}$$

where $\sigma_j(A_{\boldsymbol{n}})$, $j = 1, \ldots, d_{\boldsymbol{n}}$ are the singular values of $A_{\boldsymbol{n}}$.

Remark A.1 *Denote by $\lambda_1(f), \ldots, \lambda_s(f)$ and by $\sigma_1(f), \ldots, \sigma_s(f)$ the eigenvalues and the singular values of a $s \times s$ matrix-valued function f, respectively. If f is smooth enough, an informal interpretation of the limit relation* (53) *(resp.* (54)*) is that when the matrix-size of $A_{\boldsymbol{n}}$ is sufficiently large, then $d_{\boldsymbol{n}}/s$ eigenvalues (resp. singular values) of $A_{\boldsymbol{n}}$ can be approximated by a sampling of $\lambda_1(f)$ (resp. $\sigma_1(f)$) on a uniform equispaced grid of the domain G, and so on until the last $d_{\boldsymbol{n}}/s$ eigenvalues can be approximated by an equispaced sampling of $\lambda_s(f)$ (resp. $\sigma_s(f)$) in the domain.*

In the following, two well-known results on the spectral distribution of Toeplitz sequences. If f is a real-valued function, the following theorem due to Szegö holds:

Theorem A.2 ([10]) *Let $f \in L^1([-\pi, \pi]^d)$ be a real-valued function. Then, $\{T_{\boldsymbol{n}}(f)\}_{\boldsymbol{n}} \sim_\lambda (f, [-\pi, \pi]^d)$.*

In the case where f is a Hermitian matrix-valued function, previous theorem can be extended as follows:

Theorem A.3 ([11]) *Let $f : [-\pi, \pi]^d \to \mathbb{C}^{s\times s}$ with $f_{ij} \in L^1([-\pi, \pi]^d)$, $i, j = 1, \ldots, s$, be a Hermitian matrix-valued function. Then, $\{T_{\boldsymbol{n}}(f)\}_{\boldsymbol{n}} \sim_\lambda (f, [-\pi, \pi]^d)$.*

We are know ready to provide more details on the GLT sequences. Each GLT sequence is *associated* to a *complex-valued* Lebesgue-measurable function κ, which is known as the *symbol* of the sequence $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$. In this case, we note $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa$. The domain of definition G of the symbol is taken as $[0,1]^d \times [-\pi,\pi]^d$ while a point in G is denoted $(\boldsymbol{x},\boldsymbol{\theta})$, where $\boldsymbol{x} = (x_1,\ldots,x_d)$ are the *physical variables* and $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_d)$ are the *Fourier variables*.

We recall the following properties of a GLT sequence $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$:

GLT1 Let $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa$ with $\kappa : G \to \mathbb{C}$, $G = [0,1]^d \times [-\pi,\pi]^d$, then $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\sigma (\kappa, G)$. If the matrices $A_{\boldsymbol{n}}$ are definitely Hermitian, that is $A_{\boldsymbol{n}} - A_{\boldsymbol{n}}^*$ is 'small enough' (see Theorem 3.4 in [12]), then it holds also $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\lambda (\kappa, G)$.

GLT2 The set of GLT sequences form a $*$-algebra, i.e., it is closed under linear combinations, products, inversion, conjugation: hence, the sequence obtained via algebraic operations on a finite set of input GLT sequences is still a GLT sequence and its symbol is obtained by following the same algebraic manipulations on the corresponding symbols of the input GLT sequences. In symbols, let $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa_1$ and $\{B_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa_2$, then

- $\{\alpha A_{\boldsymbol{n}} + \beta B_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \alpha\kappa_1 + \beta\kappa_2, \quad \alpha,\beta \in \mathbb{C}$;
- $\{A_{\boldsymbol{n}} B_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa_1\kappa_2$;
- if κ_1 vanishes, at most, in a set of zero Lebesgue measure, then $\{A_{\boldsymbol{n}}^{-1}\}_{\boldsymbol{n}} \sim_{\text{GLT}} \kappa_1^{-1}$;
- $\{A_{\boldsymbol{n}}^*\}_{\boldsymbol{n}} \sim_{\text{GLT}} \bar{\kappa_1}$.

GLT 3 Any sequence of Toeplitz matrices $\{T_{\boldsymbol{n}}(f)\}_{\boldsymbol{n}}$ generated by a function $f \in L^1([-\pi,\pi]^d)$ is a GLT sequence with symbol $\kappa(\boldsymbol{x},\boldsymbol{\theta}) = f(\boldsymbol{\theta})$.

GLT 4 Any sequence of diagonal sampling matrices $\{D_{\boldsymbol{n}}(a)\}_{\boldsymbol{n}}$ containing the evaluations of a Riemann-integrable function $a : [0,1]^d \to \mathbb{C}$ over a uniform grid is a GLT sequence with symbol $\kappa(\boldsymbol{x},\boldsymbol{\theta}) = a(\boldsymbol{x})$.

GLT 5 Every sequence which is distributed as the constant zero in the singular value sense is a GLT sequence with symbol 0 and viceversa, i.e., $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\sigma 0 \iff \{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_{\text{GLT}} 0$.

Remark A.4 *Property* **[GLT4]** *is crucial when dealing with non-constant coefficient problems. In this paper we only focus on constant coefficient problems, so it has been mentioned just for the sake of completeness.*

We end this subsection, introducing the notion of zero-distributed matrix-sequences and giving a characterization for them.

Definition A.2 *Let $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ be a sequence of matrices with $\dim(A_{\boldsymbol{n}}) = d_{\boldsymbol{n}}$ and $d_{\boldsymbol{n}} \to \infty$ as $\boldsymbol{n} \to \infty$. We say that $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}}$ is a* zero-distributed matrix-sequence *if $\{A_{\boldsymbol{n}}\}_{\boldsymbol{n}} \sim_\sigma 0$.*

B Example of GLT symbol computation

Although the GLT is not restricted to elliptic partial differential equations, we shall consider the following special form

$$-\nabla \cdot (A(x,y)\nabla\phi(x,y)) + c(x,y)\phi(x,y) = f(x,y) \tag{55}$$

where A is a matrix function and c a scalar function of (x,y).
This equation can be written in a weak formulation as

$$\int_\Omega (A\nabla\phi)\cdot\nabla\psi + c\phi\psi \; d\Omega = \int_\Omega f\psi \; d\Omega \tag{56}$$

where ψ denotes a test function.
We also consider that the computational domain Ω is the *image* of the unit square $\hat{\Omega} = [0,1]^2$ (called a logical domain or a patch) by a *mapping* (geometric transformation) F, *i.e.* $(x,y) = F(\xi_1, \xi_2)$ (Fig. 4). The jacobian matrix associated to the inverse mapping, will be denoted $\mathcal{J}_{F^{-1}}$, while J_F will stand for determinant of the Jacobian matrix associated to the mapping F. Equation (Eq. 56) can be written as

$$\int_{\hat{\Omega}} \left(\nabla\psi \left(\mathcal{J}_{F^{-1}}^T A \mathcal{J}_{F^{-1}}\right) \nabla\phi + c\psi\phi\right) J_F \; d\hat{\Omega} = \int_{\hat{\Omega}} f\psi J_F \; d\hat{\Omega} \tag{57}$$

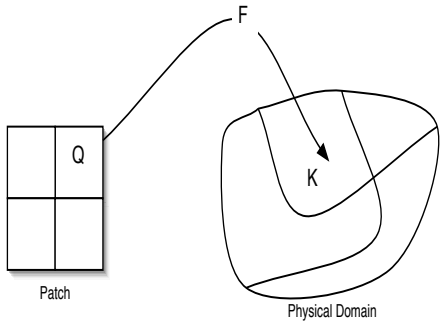

Figure 4: The computational domain is the image of a patch, *logical domain*, with a geometric transformation **F**. An element Q in the patch is *mapped* to an element $K = F(Q)$.

Notice that for sack of simplicity, we do not specify the arguments of the functions A and c. They are implicitly given, using the definition $A(x,y) := A \circ \mathcal{F}(\xi_1, \xi_2)$ for example.
Taking into account the form of the last weak formulation, we see that it is similar to the first one. Therefor, without losing in generality, we shall consider the computational domain to be a unit square *i.e.* $\Omega = [0,1]^2$. We will also consider the following weak formulation

$$\int_{[0,1]^2} (A\nabla\phi) \cdot \nabla\psi + c\phi\psi \; d\Omega = \int_{[0,1]^2} f\psi \; d\Omega \tag{58}$$

where we keep the same notation A and c even if in the case of a mapping, we should use the composition with the change of coordinates.
Using an expansion over the Spline space (for both ϕ and ψ), we have the following matrix form associated to the weak formulation (Eq. 58)

$$\Sigma_{\boldsymbol{i},\boldsymbol{j}} = \int_{\Omega} \{\nabla\Lambda_{\boldsymbol{i}} A \nabla\Lambda_{\boldsymbol{j}} + c\Lambda_{\boldsymbol{i}}\Lambda_{\boldsymbol{j}}\} \; d\Omega \tag{59}$$

Now let us use the multi-indices $\boldsymbol{i} = (i_1, i_2)$ and $\boldsymbol{j} = (j_1, j_2)$ and write the basis function in its tensor form, *i.e.* $\Lambda_{\boldsymbol{i}}(\xi_1, \xi_2) = \Lambda^1_{i_1}(\xi_1)\Lambda^2_{i_2}(\xi_2)$. We shall drop the exponent indices, to make our document easy to read. The reader can always know which direction (1 or 2) we are using, since it is implicitly specified by the i_1 and i_2 indices for example. Let's first start with the *mass* term. We have,

$$\int_{\Omega} c\Lambda_{\boldsymbol{i}}\Lambda_{\boldsymbol{j}} \; d\Omega = \int_{[0,1]^2} c\Lambda_{i_1}\Lambda_{i_2}\Lambda_{j_1}\Lambda_{j_2} \; d\Omega = \int_{[0,1]^2} c \left(\Lambda_{i_2}\Lambda_{j_2}\right)\left(\Lambda_{i_1}\Lambda_{j_1}\right) \; d\Omega \tag{60}$$

where we gathered together, terms related to the same direction. Let us assume for the moment that the function c is of the form $c(\xi_1, \xi_2) = c_1(\xi_1)c_2(\xi_2)$. We have,

$$\int_{\Omega} c\Lambda_{\boldsymbol{i}}\Lambda_{\boldsymbol{j}} \; d\Omega = \int_{[0,1]^2} \left(c_1\Lambda_{i_1}\Lambda_{j_1}\right)\left(c_2\Lambda_{i_2}\Lambda_{j_2}\right) \; d\Omega = \left(\int_{[0,1]} c_1\Lambda_{i_1}\Lambda_{j_1} \; d\xi_1\right)\left(\int_{[0,1]} c_2\Lambda_{i_2}\Lambda_{j_2} \; d\xi_2\right) \tag{61}$$

Under proper assumptions (cf. [1, 10]), the eigenvalues of the linear system associated to the the sequence of matrices $\left(\left[\int_{[0,1]} c_1\Lambda_{i_1}\Lambda_{j_1}\right]_{i_1,j_1=1}^{n_1+p_1}\right)_{n_1}$ is described by the symbol $c_1\mathfrak{m}_1$. To improve the readability of our

document, we will remove the parenthesis describing the sequence of the matrices, needed for the consistency of the GLT equivalence definition. Hence,

$$\left[\int_{[0,1]} c_1 \Lambda_{i_1} \Lambda_{j_1}\right]_{i_1,j_1=1}^{n_1+p_1} \sim_{\text{GLT}} \frac{1}{n_1} c_1 \mathfrak{m}_1 \tag{62}$$

Idem for

$$\left[\int_{[0,1]} c_2 \Lambda_{i_2} \Lambda_{j_2}\right]_{i_2,j_2=1}^{n_2+p_2} \sim_{\text{GLT}} \frac{1}{n_2} c_2 \mathfrak{m}_2 \tag{63}$$

Therefor, the *mass* term has the following symbol

$$\left[\int_\Omega c \Lambda_{\boldsymbol{i}} \Lambda_{\boldsymbol{j}} \, d\Omega\right]_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{n}+\boldsymbol{p}} \sim_{\text{GLT}} \frac{1}{n_1} c_1 \mathfrak{m}_1 \frac{1}{n_2} c_2 \mathfrak{m}_2 \tag{64}$$

$$\sim_{\text{GLT}} \frac{1}{n_1 n_2} c \; \mathfrak{m}_1 \mathfrak{m}_2 \tag{65}$$

where we used the assumption $c = c_1 c_2$. In general, the function c may not be a *separable* function with respect to its variables. However, under proper assumptions (cf. [1, 5]), we can write $c = \sum_k c_1^k c_2^k$ (the sum can be finite or infinite). Therefor, using the fact that the GLT is an (involutive) algebra (stability with respect to the sum), we have

$$\left[\int_\Omega c \Lambda_{\boldsymbol{i}} \Lambda_{\boldsymbol{j}} \, d\Omega\right]_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{n}+\boldsymbol{p}} = \left[\sum_k \int_\Omega c_1^k c_2^k \Lambda_{\boldsymbol{i}} \Lambda_{\boldsymbol{j}} \, d\Omega\right]_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{n}+\boldsymbol{p}} \sim_{\text{GLT}} \sum_k \frac{1}{n_1} c_1^k \mathfrak{m}_1 \frac{1}{n_2} c_2^k \mathfrak{m}_2 \tag{66}$$

$$\sim_{\text{GLT}} \frac{1}{n_1 n_2} \sum_k \left(c_1^k c_2^k\right) \mathfrak{m}_1 \mathfrak{m}_2 \tag{67}$$

$$\sim_{\text{GLT}} \frac{1}{n_1 n_2} c \; \mathfrak{m}_1 \mathfrak{m}_2 \tag{68}$$

On the other hand, the first term in the equation (Eq. 59) writes

$$\int_\Omega \nabla \Lambda_{\boldsymbol{j}} A \nabla \Lambda_{\boldsymbol{i}} \, d\Omega = \int_{[0,1]^2} \left(a_{11} \Lambda'_{j_1} \Lambda_{j_2} \Lambda'_{i_1} \Lambda_{i_2} + a_{12} \Lambda'_{j_1} \Lambda_{j_2} \Lambda_{i_1} \Lambda'_{i_2} + a_{21} \Lambda_{j_1} \Lambda'_{j_2} \Lambda'_{i_1} \Lambda_{i_1} + a_{22} \Lambda_{j_1} \Lambda'_{j_2} \Lambda_{i_1} \Lambda'_{i_1}\right) \, d\Omega \tag{69}$$

Let's gather now terms with respect to the first or second variable. The last expression writes

$$\begin{aligned}\int_\Omega \nabla \Lambda_{\boldsymbol{i}} A \nabla \Lambda_{\boldsymbol{j}} \, d\Omega = &\int_{[0,1]^2} a_{11} \left(\Lambda'_{i_1} \Lambda'_{j_1}\right) \left(\Lambda_{i_2} \Lambda_{j_2}\right) \, d\Omega + \int_{[0,1]^2} a_{12} \left(\Lambda'_{i_1} \Lambda_{j_1}\right) \left(\Lambda_{i_2} \Lambda'_{j_2}\right) \, d\Omega \\ &+ \int_{[0,1]^2} a_{21} \left(\Lambda_{i_1} \Lambda'_{j_1}\right) \left(\Lambda'_{i_2} \Lambda_{j_2}\right) \, d\Omega + \int_{[0,1]^2} a_{22} \left(\Lambda_{i_1} \Lambda_{j_1}\right) \left(\Lambda'_{i_2} \Lambda'_{j_2}\right) \, d\Omega\end{aligned}$$

Following the same idea as described for the *mass* term, we get the following symbol

$$\left[\int_\Omega \nabla \Lambda_{\boldsymbol{i}} A \nabla \Lambda_{\boldsymbol{j}} \, d\Omega\right]_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{n}+\boldsymbol{p}} \sim_{\text{GLT}} \frac{n_1}{n_2} a_{11} \mathfrak{s}_1 \mathfrak{m}_2 + a_{12} \mathfrak{a}_1 \mathfrak{a}_2^\star + a_{21} \mathfrak{a}_1^\star \mathfrak{a}_2 + \frac{n_2}{n_1} a_{22} \mathfrak{m}_1 \mathfrak{s}_2 \tag{70}$$

where we introduced the symbol $\mathfrak{a}$ of the advection operator

$$\left[\int_0^1 \Lambda_{i_1}^p(t) \; \left(\Lambda_{j_1}^p(t)\right)' \, dt\right]_{i_1,j_1=1}^{n+p} \sim_{\text{GLT}} \mathfrak{a} \tag{71}$$

Combining (70) and (68), we get the symbol of the sequence of matrices (59)

$$M \sim_{\text{GLT}} \frac{n_1}{n_2} a_{11} \mathfrak{s}_1 \mathfrak{m}_2 + a_{12} \mathfrak{a}_1 \mathfrak{a}_2^\star + a_{21} \mathfrak{a}_1^\star \mathfrak{a}_2 + \frac{n_2}{n_1} a_{22} \mathfrak{m}_1 \mathfrak{s}_2 + \frac{1}{n_1 n_2} c \; \mathfrak{m}_1 \mathfrak{m}_2 \tag{72}$$

Moreover, one can prove that the symbol $\kappa = \frac{n_1}{n_2}\mathfrak{s}_1\mathfrak{m}_2 + \frac{n_2}{n_1}\mathfrak{m}_1\mathfrak{s}_2$, allows for a good clustering of the eigenvalues as long as the input functions A and c are independent from the mesh size. In fact, κ is the symbol of the $2D$ Laplacian. Therefor, we can use the Laplacian (solved with the Multigrid and the presented post smoother) as preconditinoer for PGMRES or PCG (if $a_{12} = a_{21} = 0$). More details can be found in [1].
The computations presented in this section can be generalized to a large class of differential operators, even including different spaces (like $H(curl), H(div)$ for instance.