Multigrid for Finite Elements using Splines.

Trach-Minh Tran, Stephan Brunner v0.1, January 2013

A multigrid formulation for finite elements is derived, using variational principles. More specifically the grid transfer operators will be derived and tested in 2D Cartesian and cylindrical geometry for arbitrary order B-Splines.

Contents

1 The Model Problems

1.1 Cartesian Geometry

The following second-order boundary value problem is considered

$$
\begin{gathered}
-\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right]u(x,y)=f(x,y) \qquad 0\le x\le 1, \quad 0\le y\le 1\\
u(0,y)=u(1,y)=u(x,0)=u(x,1)=0.
\end{gathered}
\tag{1}
$$

By choosing

$$f(x,y)=\sin(\pi k_x x+\pi k_y y),$$

where k_x and k_y are integers, the solution of the BVP is simply

$$u(x,y)=\frac{\sin(\pi k_x x+\pi k_y y)}{\pi^2(k_x^2+k_y^2)}.$$

Using a weak formulation on Eq.(1) and a grid of $N_x\times N_y$ intervals, one obtains the following discretized linear system

$$\sum_{i'=1}^{N_x+p}\sum_{j'=1}^{N_y+p}A_{iji'j'}u_{i'j'}=b_{ij}, \qquad i=1,\dots,N_x+p, \quad j=1,\dots,N_y+p, \tag{2}$$

where the unknowns u_{ij} are the Spline (of order p) expansion coefficients of the solution

$$u(x,y) = \sum_{i=1}^{N_x+p} \sum_{j=1}^{N_y+p} u_{ij} \Lambda_i(x) \Lambda_j(y), \tag{3}$$

and the matrix A and right hand side b are determined from

$$A_{iji'j'} = \int_0^1 \int_0^1 dxdy \left[\Lambda'_{i'}(x)\Lambda_{j'}(y)\Lambda'_i(x)\Lambda_j(y) + \Lambda_{i'}(x)\Lambda'_{j'}(y)\Lambda_i(x)\Lambda'_j(y) \right], \tag{4}$$

$$b_{ij} = \int_0^1 \int_0^1 dxdy \Lambda_i(x)\Lambda_j(y) f(x,y). \tag{5}$$

Note that using a Gauss quadrature with $\lceil (2p+1)/2 \rceil$ points per interval to calculate the matrix A would yield an exact integration.

1.2 Cylindrical Geometry

The following second-order boundary value problem is considered:

$$\begin{aligned} -\left[\frac{1}{r}\frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} \right] u(r,\theta) = f(r,\theta) \qquad 0 \le r \le 1, \quad 0 \le \theta < 2\pi \\ u(1,\theta) = 0, \end{aligned} \tag{6}$$

By choosing

$$f(r,\theta) = j_{ms}^2 J_m(j_{ms} r) \cos(m\theta),$$

where m is an integer and j_{ms}, the s^{th} zero of J_m, the solution of this BVP is

$$u(r,\theta) = J_m(j_{ms} r) \cos(m\theta).$$

Using a weak formulation on Eq.(6)and a grid of $N_r \times N_\theta$ intervals, one obtains the following discretized linear system

$$\sum_{i'=1}^{N_r+p} \sum_{j'=1}^{N_\theta} A_{iji'j'} u_{i'j'} = b_{ij}, \qquad i = 1, \dots, N_r + p, \quad j = 1, \dots, N_\theta, \tag{7}$$

where the unknowns u_{ij} are the Spline (of order p) expansion coefficients of the solution

$$u(r,\theta) = \sum_{i=1}^{N_r+p} \sum_{j=1}^{N_\theta} u_{ij} \Lambda_i(r) \Lambda_j(\theta), \tag{8}$$

and the matrix A and right hand side b are determined from

$$A_{iji'j'} = \int_0^1 \int_0^{2\pi} r dr d\theta \left[\Lambda'_{i'}(r)\Lambda_{j'}(\theta)\Lambda'_i(r)\Lambda_j(\theta) + \frac{1}{r^2}\Lambda_{i'}(r)\Lambda'_{j'}(\theta)\Lambda_i(r)\Lambda'_j(\theta) \right], \tag{9}$$

$$b_{ij} = \int_0^1 \int_0^{2\pi} r dr d\theta \Lambda_i(r)\Lambda_j(\theta) f(r,\theta). \tag{10}$$

Note that A has an $1/r$ singularity in the integrand. For $m \neq 0$, this should not be problematic since the converged solution behaves as $\sim r^m$ near $r = 0$. The case $m = 0$ will be investigated numerically latter in this report, together withe the $m \neq 0$ case.

2 Restriction Operator

In the following, let us use the superscripts h and $2h$ to denote quantities defined respectively on a *fine* ($N_x \times N_y$ or $N_r \times N_\theta$) and a *coarser* ($N_x/2 \times N_y/2$ or $N_r/2 \times N_\theta/2$) grid.

The two grid transfers required in the standard *multigrid* [1, 2] are:

1. the *restriction* of the right hand side: $\mathbf{b}^h \longrightarrow \mathbf{b}^{2h}$ and
2. the *prolongation* of the solution: $\mathbf{u}^{2h} \longrightarrow \mathbf{u}^h$.

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

$$\Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} c_{ii'} \Lambda_{i'}^h(x), \quad i = 1, \dots, N/2+p. \tag{11}$$

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

$$b_i^{2h} = \int_0^1 dx f(x) \Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} c_{ii'}\, b_{i'}^h = \sum_{i'=1}^{N+p} R_{ii'}\, b_{i'}^h.$$

It can be computed by simply projecting Eq.(11) on the fine grid basis function $\Lambda_j^h(x)$ [1]:

$$\sum_{i'=1}^{N+p} R_{ii'} \underbrace{\int_0^1 dx \Lambda_{i'}^h(x) \Lambda_j^h(x)}_{M_{i'j}^h} = \underbrace{\int_0^1 dx \Lambda_i^{2h}(x) \Lambda_j^h(x)}_{M_{i'j}^{2h,h}} \Longrightarrow \mathbf{R} = \mathbf{M}^{2h,h} \cdot (\mathbf{M}^h)^{-1}. \tag{12}$$

It should be stressed that the representation for $\Lambda_i^{2h}(x)$ in Eq.(11) is *unique*. This is checked by verifying that the same matrix $R_{ii'}$ is obtained using for example the *collocation* methods. One such method, which is used for this check is detailed in Appendix A. The calculated grid transfer matrices for linear, quadratic and cubic periodic and non-periodic Splines are given in [1].

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

$$b_{ij}^{2h} = \int_0^1 \int_0^1 dx dy f(x,y) \Lambda_i^{2h}(x) \Lambda_j^{2h}(y) = \sum_{i'=1}^{N+p} \sum_{j'=1}^{N+p} R_{ii'}^x R_{jj'}^y b_{i'j'}^h,$$

and thus

$$\boxed{\mathbf{b}^{2h} = \mathbf{R}^x \cdot \mathbf{b}^h \cdot (\mathbf{R}^y)^T.} \tag{13}$$

3 Prolongation Operator

Using Eq.(11) (with $c_{ii'} = R_{ii'}$), the solution at the coarse grid can be expressed as

$$u^{2h}(x) = \sum_{i=1}^{N/2+p} u_i^{2h} \Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} \left[\sum_{i=1}^{N/2+p} R_{ii'} u_i^{2h} \right] \Lambda_{i'}^h(x) = \sum_{i'=1}^{N+p} \underbrace{\left[\sum_{i=1}^{N/2+p} (R)_{i'i}^T u_i^{2h} \right]}_{\tilde{u}_{i'}^h} \Lambda_{i'}^h(x),$$

from which one obvious choice for the *prolongation* operator would be

$$\mathbf{P} = \mathbf{R}^T = (\mathbf{M}^h)^{-1} \cdot \mathbf{M}^{h,2h}. \tag{14}$$

Generalization to a two-dimensional prolongation is obtained as follows, where summation over repeated indices is assumed:

$$u^{2h}(x,y) = u_{ij}^{2h} \Lambda_i^{2h}(x) \Lambda_j^{2h}(y) = \left[R_{ii'}^x u_{ij}^{2h} R_{jj'}^y \right] \Lambda_{i'}^h(x) \Lambda_{j'}^h(y)$$

which leads to the prolonged solution $\tilde{\mathbf{u}}^h$ given by

$$\boxed{\tilde{\mathbf{u}}^h = \mathbf{P}^x \cdot \mathbf{u}^{2h} \cdot (\mathbf{P}^y)^T.} \tag{15}$$

It should be noted here that, while the restricted right hand side $\mathbf{b}^{2h}$ as defined in Eq.(13) is *exactly identical* to the assembled right hand side, the prolonged solution $\tilde{\mathbf{u}}^h$ defined in Eq.(15) is just a representation of $u^{2h}(x,y)$ on the fine mesh and *not* the solution $u^h(x,y)$ which can only be obtained by solving the problem on the fine mesh!

4 Numerical Experiments

The multigrid performance can be characterized by looking at the convergence of the residual Euclidean norm for the linear system $\mathbf{Au} = \mathbf{b}$:

$$\|\mathbf{r}\|_2 = \|\mathbf{b} - \mathbf{Au}\|_2. \tag{16}$$

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

$$\|e\|_2 = \sqrt{\int dV \left[\sum_{ij} u_{ij}\Lambda_{ij}(x,y) - u(x,y)\right]^2} \tag{17}$$

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

4.1 Cartesian Geometry

The multigrid performances for varying problem sizes are displayed in Fig.(1) for linear Splines and Fig.(2) for cubic Splines. They show that the number of iterations required for convergence (abount 3 for both linear and cubic Splines) is insensitive to the problem sizes. Compared to direct methods, the multigrid should scale much better for large problem sizes, as indicated in Table 1. For this model problem, using cubic Splines seems to converge slightly faster than linear Splines!

	Linear Splines		Cubic Splines	
N	$V(2,1)$	Direct	$V(2,1)$	Direct
16	8.844E-04	2.051E-03	2.653E-03	3.970E-03
32	1.661E-03	5.345E-03	4.983E-03	1.540E-02
64	5.766E-03	2.054E-02	1.730E-02	7.492E-02
128	2.347E-02	3.288E-01	7.042E-02	1.060E+00

Table 1: Times (in seconds) used by a the *direct sparse* solver MUMPS-4.10.0 for different problem sizes versus the times used by *three* multigrid $V(2,1)$ cycles. The Intel Fortran-13.0 compiler is used on an Intel i7 platform.

The effects of the relaxation parameters ν_1, ν_2 on the multigrid performnace (Fig.(3)) indicates that only a few relaxations are sufficient to achieve a good multigrid performance. Further analysis of the computational cost is required however to determine the *optimal* ν_1, ν_2.

Finally, the effects of the number of grid levels are analyzed in Fig.(4). In addition to the computational cost (see Table 2), the memory required for the *direct solver* at the coarsest grid level should be taken into account for the choice of the optimal number of grid levels, especially for very large problems.

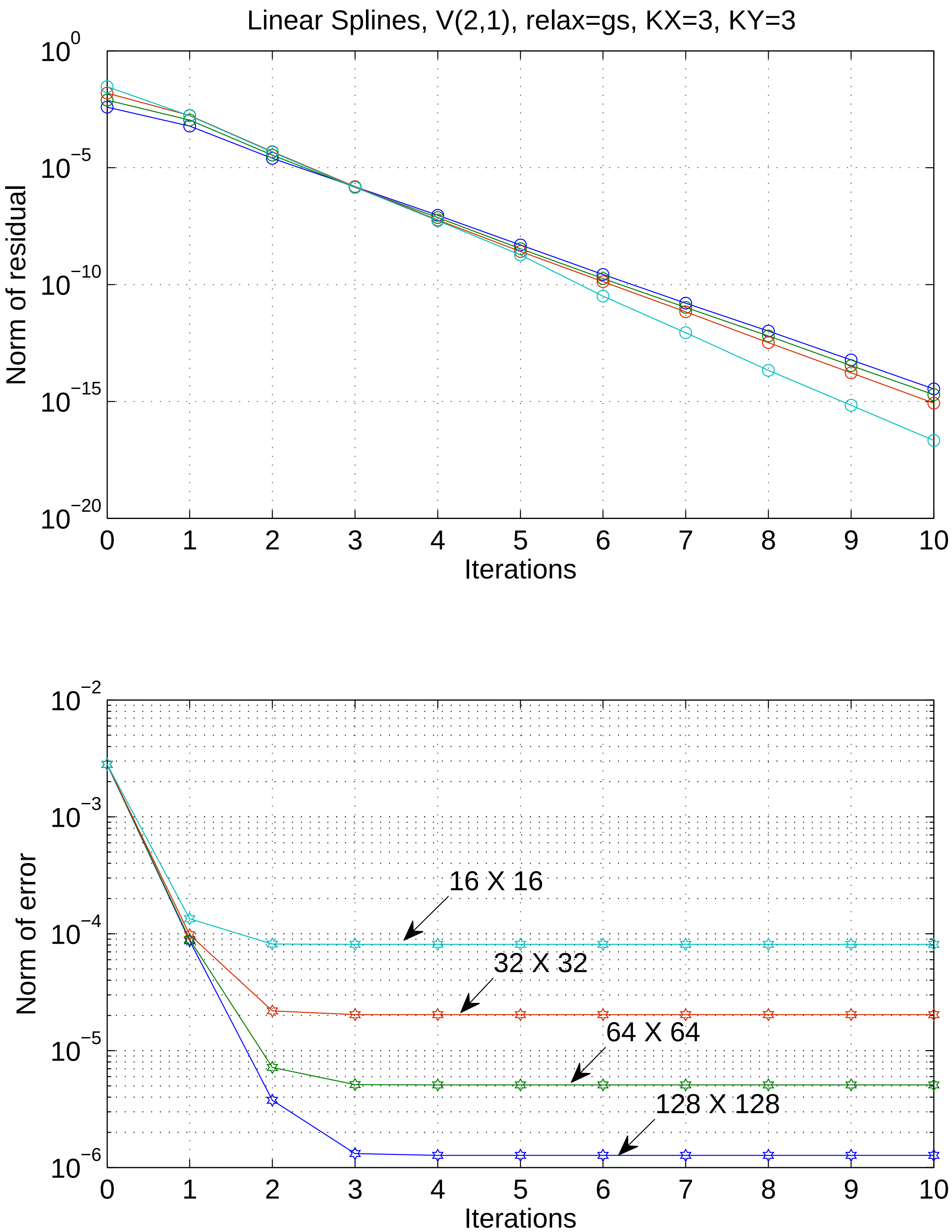

Figure 1: Performance of the multigrid $V(2,1)$ scheme using a Gauss-Seidel relaxation and *linear Splines* for different problem sizes. The size of the *coarsest* grid is 2×2.

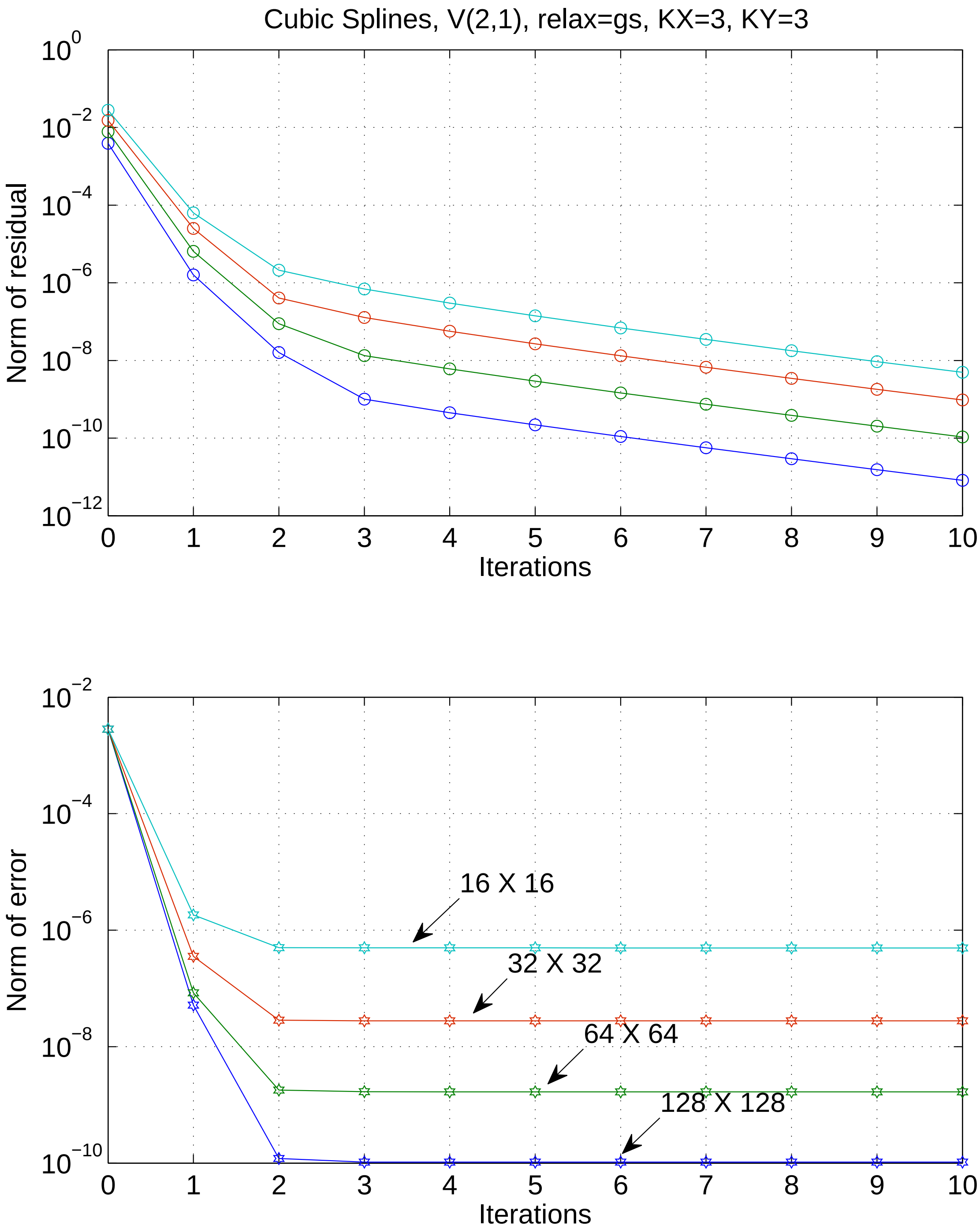

Figure 2: Performance of the multigrid $V(2,1)$ scheme using a Gauss-Seidel relaxation and *cubic Splines* for different problem sizes. The size of the *coarsest* grid is 2×2.

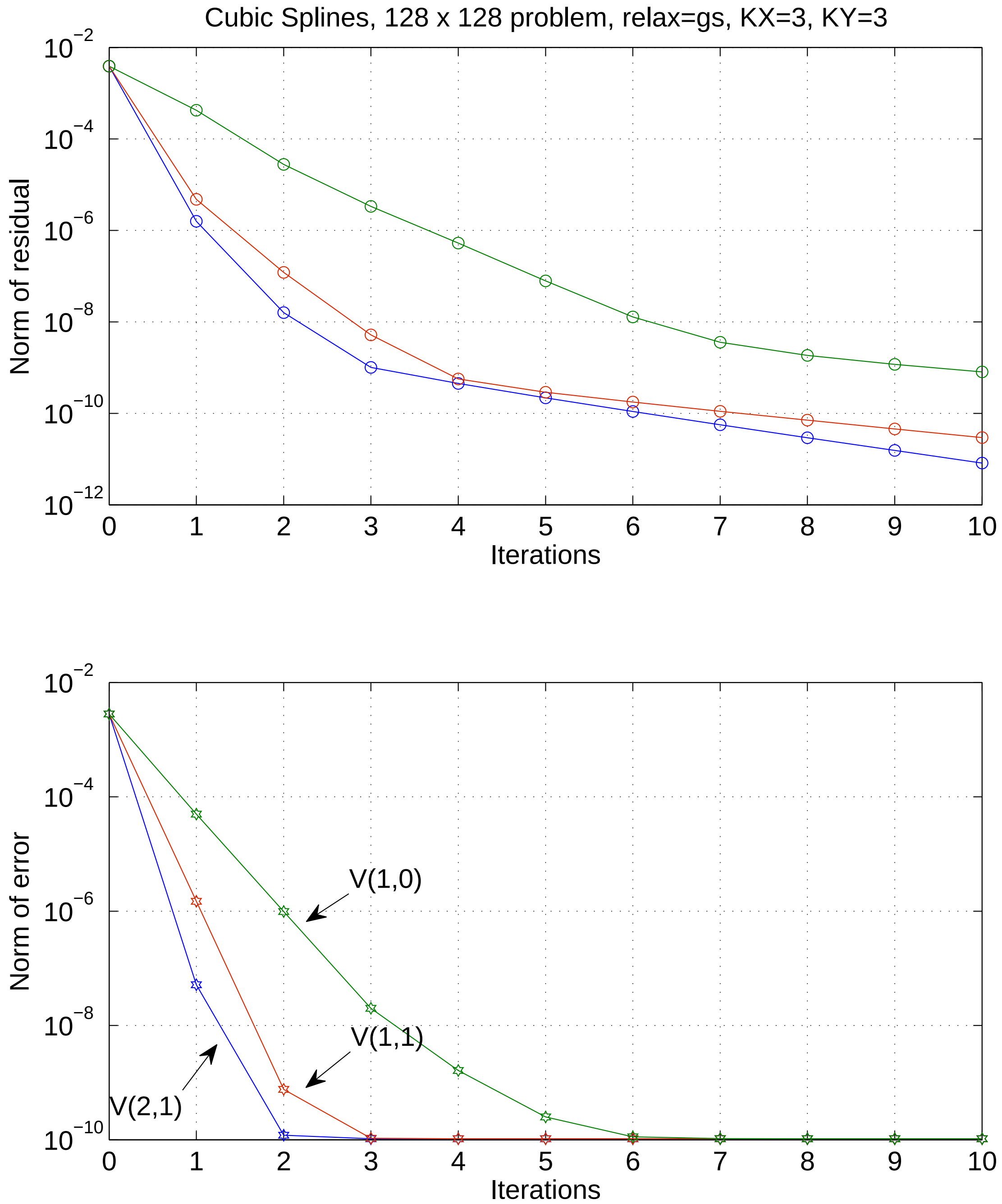

Figure 3: Effect of the number of the relaxation sweeps ν_1, ν_2 on the performance of the multigrid $V(\nu_1, \nu_2)$-cycle for *Cubic Splines*. The finest grid has 128×128 intervals.

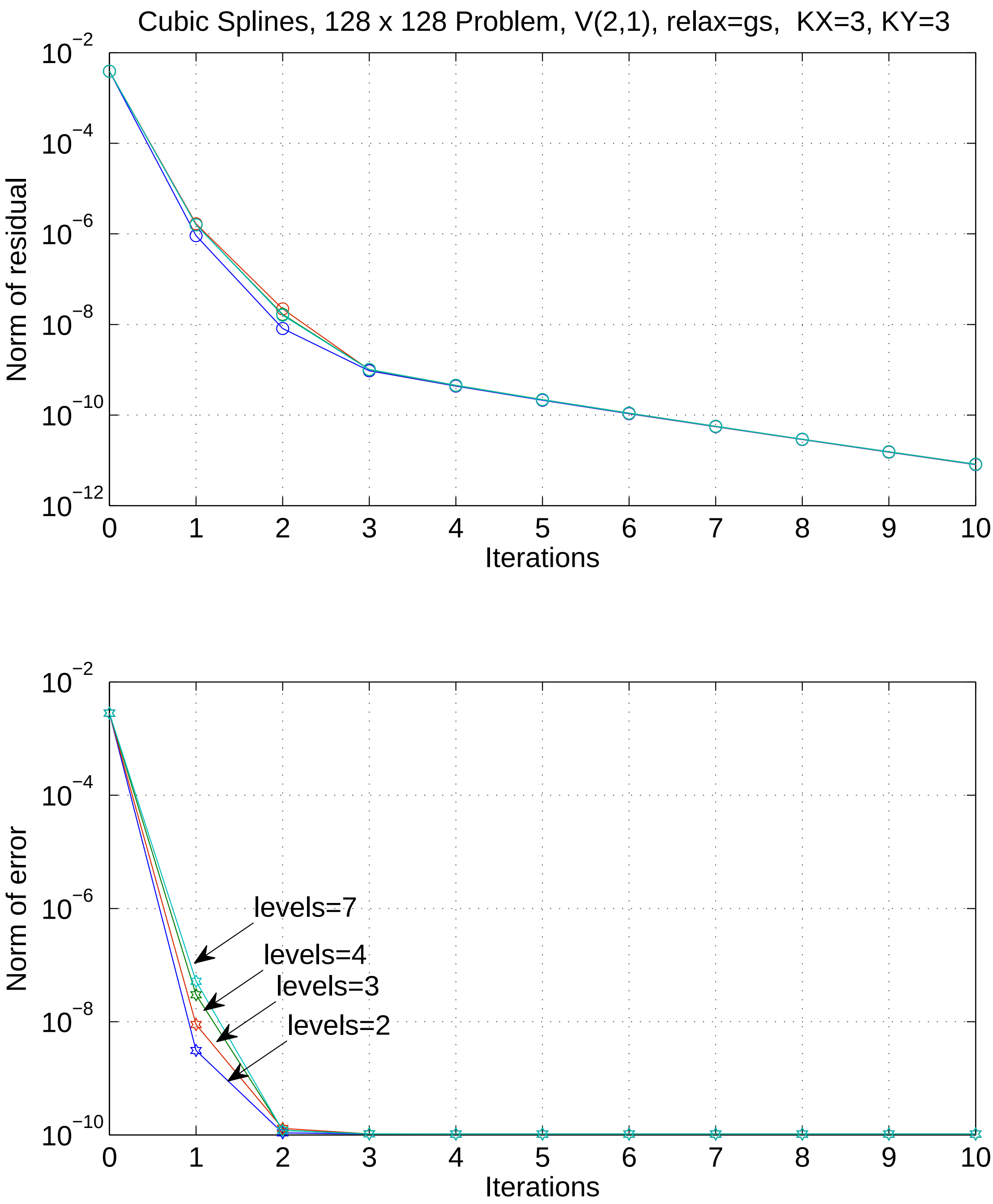

Figure 4: Effect of the number grid levels on the performance of the multigrid $V(2,1)$-cycle for *Cubic Splines*. The finest grid has 128×128 intervals.

Number of levels	$V(1,0)$	$V(1,1)$	$V(2,1)$
2	3.386E-02	3.881E-02	4.031E-02
3	2.923E-02	3.398E-02	3.605E-02
4	2.880E-02	3.275E-02	3.595E-02
7	2.912E-02	3.236E-02	3.566E-02

Table 2: Effects of the times in seconds used per V-cyclefor different number of grid levels and relaxation paramters for a 128×128 problem. The Intel Fortran-13.0 compiler is used on an Intel i7 platform.

4.2 Cylindrical Geometry

A Grid transfer matrix by collocation

Let first consider the *periodic case*. Denoting N as the number of intervals of the fine grid, the *periodic* Spline basis functions on the *coarse* grid Λ_i^{2h} can be expressed as linear combinations of the *fine* grid Spline basis functions as:

$$\Lambda_i^{2h}(x) = \sum_{i'=1}^{N} R_{ii'} \Lambda_{i'}^h(x), \quad i = 1, \dots, N/2. \tag{18}$$

For any given i, the coefficients $R_{ii'}$ can be calculated by expressing the relation above on exactly N points on the x-grid. For *odd* Spline order p, these *collocation* (or interpolating) points can be chosen as the *break* points of the fine grid $x_k^h, \quad k = 0, \dots, N-1$. For *even* values of p, the collocation points should be $x_{k+1/2}^h = (x_k^h + x_{k+1}^h)/2$ in order to obtain a non-singular linear system of equations [3]. The resulting system of equations to solve for $R_{ii'}$ are given below:

$$\begin{aligned}
p \text{ odd}: &\quad \sum_{i'=1}^{N} \Lambda_{i'}^h(x_k^h)\, R_{ii'} = \Lambda_i^{2h}(x_k^h), \qquad k = 0, \dots, N-1, \quad i = 1, \dots, N/2, \\
p \text{ even}: &\quad \sum_{i'=1}^{N} \Lambda_{i'}^h(x_{k+1/2}^h)\, R_{ii'} = \Lambda_i^{2h}(x_{k+1/2}^h), \qquad k = 0, \dots, N-1, \quad i = 1, \dots, N/2.
\end{aligned} \tag{19}$$

For *non-periodic* Splines, there are $N + p$ and $N/2 + p$ basis functions respectively on the fine and coarse grid:

$$\Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} R_{ii'} \Lambda_{i'}^h(x), \quad i = 1, \dots, N/2 + p. \tag{20}$$

This implies that for any given Λ_i^{2h}, $N + p$ conditions are required to determined the $N + p$ terms of row i of the matrix $R_{ii'}$. For odd p, $N + 1$ collocation points $x_k, \quad k = 0, \dots, N$ can be used with the missing $p - 1$ equations obtained by expressing all the $(p-1)/2$ derivatives of $\Lambda_i^{2h}(x)$ at the end points x_0 and x_N:

$$\frac{d^\alpha}{dx^\alpha} \Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} R_{ii'} \frac{d^\alpha}{dx^\alpha} \Lambda_{i'}^h(x), \quad \alpha = 1, \dots, \frac{p-1}{2} \quad (p \text{ odd}). \tag{21}$$

For *even* p, in addition to the N relations obtained with the collocation points $x_{k+1/2}$ (as in the *periodic* case), the missing p conditions can be obtained by expressing Λ_i^{2h} and its derivatives up to $p/2 - 1$ at the end points x_0 and x_N:

$$\frac{d^\alpha}{dx^\alpha} \Lambda_i^{2h}(x) = \sum_{i'=1}^{N+p} R_{ii'} \frac{d^\alpha}{dx^\alpha} \Lambda_{i'}^h(x), \quad \alpha = 0, \dots, \frac{p}{2} - 1 \quad (p \text{ even}). \tag{22}$$

References

[1] `Multigrid Formulation for Finite Elements`, `https://crppsvn.epfl.ch/repos/bsplines/trunk/multigrid/docs/multigrid.pdf`

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

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

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