Domain Decomposition Preconditioners for Spectral Nédélec Elements in Two and Three Dimensions

Bernhard Hientzsch

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, U.S.A., Bernhard.Hientzsch@na-net.ornl.gov, http://www.math.nyu.edu/~hientzsc. This work was supported in part by the U.S. Department of Energy under Contract DE-FC02-01ER25482.

Summary. In this paper, we present several domain decomposition preconditioners for high-order Spectral Nédélec element discretizations for a Maxwell model problem in $H(\text{CURL})$, in particular overlapping Schwarz preconditioners and Balancing Neumann-Neumann preconditioners. For an efficient and fast implementation of these preconditioners, fast matrix-vector products and direct solvers for problems posed on one element or a small array of elements are needed. In previous work, we have presented such algorithms for the two-dimensional case; here, we will present a new fast solver that works both in the two- and three-dimensional case. Next, we define the preconditioners considered in this paper, present numerical results for overlapping methods in three dimensions and Balancing Neumann-Neumann methods in two dimensions. We will also give a condition number estimate for the overlapping Schwarz method.

The model problem is: Find $\mathbf{u} \in H_0(\text{CURL}, \Omega)$ such that for all $\mathbf{v} \in H_0(\text{CURL}, \Omega)$

$$a(\mathbf{u}, \mathbf{v}) := (\alpha \mathbf{u}, \mathbf{v}) + (\beta \text{ CURL } \mathbf{u},\ \text{CURL } \mathbf{v}) = (\mathbf{f}, \mathbf{v}). \tag{1}$$

Here, Ω is a bounded, open, connected polyhedron in $\mathbb{R}^3$ or a polygon in $\mathbb{R}^2$, $H(\text{CURL}, \Omega)$ is the space of vectors in $(L^2(\Omega))^2$ or $(L^2(\Omega))^3$ with CURL in $L^2(\Omega)$ or $(L^2(\Omega))^3$, respectively; $H_0(\text{CURL}, \Omega)$ is its subspace of vectors with vanishing tangential components on $\partial\Omega$; $\mathbf{f} \in (L^2(\Omega))^d$ for $d = 2, 3$, and $(\cdot, \cdot)$ denotes the inner product in $L^2(\Omega)$ of functions or vector fields. For simplicity, we will assume that α and β are piecewise constant.

1 Discretization

We have previously presented the discretization for the two-dimensional case and some fast solvers for it in Hientzsch [2001] and Hientzsch [2003], and we will here concentrate on the three-dimensional case. As in the two-dimensional case, we use a hN-extension of Nédélec elements, parametrized by the values of the vector field on Gauss-Lobatto-Legendre grids inside the elements, with only the appropriate tangential continuity between elements (Nédélec [1980, 1986], Monk [1994], Belgacem and Bernardi [1999], Hientzsch [2001, 2003]). The integrals in the bilinear form and the right hand side are all evaluated by Gauss-Lobatto-Legendre quadrature of arbitrary order. On the reference element, the system reads:

$$\mathbf{E}\mathbf{u} = \tilde{\mathbf{f}} \qquad \text{or} \qquad \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{12}^T & E_{22} & E_{23} \\ E_{13}^T & E_{23}^T & E_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{pmatrix}$$

with

$$\begin{aligned}
E_{11} &= \alpha M_x^{1,1} \otimes M_y^{1,1} \otimes M_z^{1,1} + \beta M_x^{1,1} \otimes K_y^1 \otimes M_z^{1,1} + \beta M_x^{1,1} \otimes M_y^{1,1} \otimes K_z^1 \\
E_{22} &= \alpha M_x^{2,2} \otimes M_y^{2,2} \otimes M_z^{2,2} + \beta K_x^2 \otimes M_y^{2,2} \otimes M_z^{2,2} + \beta M_x^{2,2} \otimes M_y^{2,2} \otimes K_z^2 \\
E_{33} &= \alpha M_x^{3,3} \otimes M_y^{3,3} \otimes M_z^{3,3} + \beta K_x^3 \otimes M_y^{3,3} \otimes M_z^{3,3} + \beta M_x^{3,3} \otimes K_y^3 \otimes M_z^{3,3} \\
E_{12} &= -\beta (M_x^{1,2'} D_{m_2^x}) \otimes (D_{m_1^y}^T M_y^{1',2}) \otimes M_z^{1,2} \\
E_{13} &= -\beta (M_x^{1,3'} D_{m_3^x}) \otimes M_y^{1,3} \otimes (D_{m_1^z}^T M_z^{1',3}) \\
E_{23} &= -\beta M_x^{2,3} \otimes (M_y^{2,3'} D_{m_3^y}) \otimes (D_{m_2^z}^T M_z^{2',3})
\end{aligned}$$

$M_{dir}^{i,j}$ is a mass-matrix for direction dir, integrating products between components i and j. Primes indicate differentiated components. $K_{dir}^i = D_{m_i^{dir}}^T M_{dir}^{i',i'} D_{m_i^{dir}}$ is the weak 1D Laplacian, D_n is the differentiation matrix of order n, and u_i is of order $m_i^x \times m_i^y \times m_i^z$.

Subassembling such elements in a rectangular array of elements results in a system of the same form, only that the different mass matrices, laplacians, and derivatives are changed into the appropriate matrices for the entire array. In particular, the matrices $M_y^{1,1}$, $M_z^{1,1}$, K_y^1, K_z^1, $M_x^{2,2}$, $M_z^{2,2}$, K_x^2, K_z^2, $M_x^{3,3}$, $M_y^{3,3}$, K_x^3, and K_y^3 are subassembled; all other mass-matrices are block-diagonal, and the differentiation matrices in the cross-terms are mixed, mapping from continuous to discontinuous spaces.

The element-by-element computation of the matrix-vector product with the stiffness matrix can be implemented by dense matrix-matrix multiplications of the factors of the tensor products with the vector field laid out in array form. These multiplications use an optimized BLAS3 kernel and run at close to maximal efficiency on modern computer architectures.

For general geometries, mapped elements are used. The matrix-vector product associated to the discretization can again be implemented by tensor products and entry-by-entry (Hadamard) matrix products, and therefore also has a fast implementation. For some more details in the 2D case, see [Hientzsch, 2002, Section 4].

2 Fast direct solvers in 2D and 3D

We have previously developed a fast direct solver in Hientzsch [2001] and Hientzsch [2003] for the two-dimensional case, where the system in the two components is reduced to a generalized Sylvester equation in one component which is then solved either by a fast diagonalization method or by more stable methods for generalized Sylvester equations. It seems that this solver cannot be extended to three dimensions in the general case.

Instead of a diagonalization technique, we will try to change bases so that a block-diagonal matrix with small blocks is obtained. In two dimensions, the general block is 2×2, in three dimensions, it is 3×3, coupling modes across components. In three dimensions, we look for a basis change matrix

$$\mathbf{V} = \begin{pmatrix} V_x^2 \otimes V_y^1 \otimes V_z^1 & 0 & 0 \\ 0 & V_x^1 \otimes V_y^2 \otimes V_z^1 & 0 \\ 0 & 0 & V_x^1 \otimes V_y^1 \otimes V_z^2 \end{pmatrix}$$

so that in $\mathbf{V}^T\mathbf{E}\mathbf{V}$ all 3×3 blocks in the block tensor product matrix are diagonal (or if they are rectangular, diagonal with an extra block of zeros). Then, in the new basis, the system splits into many 3×3 or smaller systems.

We will only treat the x-direction, the same construction can be repeated for all three directions. Looking at the entries of $\mathbf{E}$, we realize that we can diagonalize all blocks, if we can diagonalize

$$\begin{array}{ll} V_x^{2,T} M_x^{1,1} V_x^2, & V_x^{2,T} M_x^{1,2'} D_{m_2^x} V_x^1, \\ V_x^{1,T} D_{m_2^x}^T M_x^{2',2'} D_{m_2^x} V_x^1, & V_x^{1,T} M_x^{2,2} V_x^1; \end{array}$$

if the second and third component have the same size in x, they are discretized in the same way, and the other discretization parameters are chosen so that mass matrices match. These conditions are not overly restrictive; a large class of generalized Nédélec elements and some newly proposed elements are of that form. In two dimensions, no such degree conditions appear, and the block diagonalization works in the general case.

The question is now if we can find V_x^1 and V_x^2 such that these four matrices are diagonal. If we first consider the terms only in V_x^1, we see that one reasonable choice would be to take the eigenbasis of the following generalized eigenvalue problem:

$$D_{m_2^x}^T M_x^{2',2'} D_{m_2^x} u = \lambda M_x^{2,2} u.$$

Now if $D_{m_2^x} V_x^1$ could be chosen as V_x^2 and had the right size, we would be done. Because the two components do not necessarily have the same size in x, we start with $V_x^2 = I_x^{1,2} D_{m_2^x} V_x^1$, chosing an appropriate $I_x^{1,2}$ as interpolation. For diagonalization to succeed, we need

$$D_{m_2^x}^T M_x^{2',2'} D_{m_2^x} = D_{m_2^x}^T I_x^{1,2,T} M_x^{1,1} I_x^{1,2} D_{m_2^x} = D_{m_2^x}^T I_x^{1,2,T} M_x^{1,2'} D_{m_2^x}$$

which can be satisfied by appropriate choice of discretization parameters and mass matrices, if $m_1^x \geq m_2^x - 1$. (If V_x^1 contains a constant vector, we need to remove this vector before differentiating.) Still, the V_x^2 so constructed is not yet a basis in general, since there may be not enough vectors in it. We construct a full basis for the complement of the range and complement V_x^2 with it. In special cases, we can give a basis for the complement explicitly, otherwise we start with carefully choosing vectors which we orthogonalize to $I_x^{1;2} D_{m_2^x} V_x^1$ and each other. The same method works for subassembled problems, and also for essential boundary value problems (also for mixed problems if each face of the box has only one type of boundary condition).

Using the block diagonalization just derived, all factor matrices in the tensor products only have non-zero entries on their diagonals (some factor matrices are rectangular). Therefore, in this basis, the solution of the system decouples into the solution of arrays of 3×3, 2×2 and 1×1 problems. The coefficients in the Gaussian elimination for these symmetric small systems can be precomputed, and the solution reduces to element-wise multiplication and addition.

For instance, a natural boundary value problem on one generalized Nédélec element can be solved in MATLAB like fashion:

```
[fev1,fev2,fev3]=applBasChgT(baschg,fm1,fm2,fm3);
[uev1,uev2,uev3]=nedtwoblslv(blslv,fev1,fev2,fev3);
[u1,u2,u3]=applBasChg(baschg,uev1,uev2,uev3);
```

where applBasChg and applBasChgT apply the basis change and its transpose and the resulting array of 3×3 problems is solve in nedtwoblslv:

```
function [uev1,uev2,uev3]=nedtwoblslv(blslv,fev1,fev2,fev3);
  rhs=fev3+blslv.t34.*fev2+blslv.t35.*fev1;
  uev3=rhs./blslv.t33;
  rhs=fev2+blslv.t24.*fev1-blslv.t23.*uev3;
  uev2=rhs./blslv.t22;
  rhs=fev1-blslv.g12.*uev2-blslv.g13.*uev3;
  uev1=rhs./blslv.g11;
return
```

These element and block solvers run very efficiently; see figure 1.

3 Overlapping Schwarz Methods

To define Schwarz preconditioners (see Smith et al. [1996]), we have to specify subspaces and solvers on them. For the two-dimensional set-up, see Hientzsch [2001] and Hientzsch [2003]. Here we will concentrate on the three-dimensional case. First, a collection of subdomains Ω_i is defined, each subdomain being either one spectral element or a union of several spectral elements. The typical size of a subdomain is denoted H, and each spectral element has a uniform degree N in all components. (The analysis goes through and the methods are implemented for more complicated settings, we chose this case here for simplicity and ease of presentation.) Now, overlapping subregions $\Omega'_{j,\delta} \subset \Omega$ are defined, with an overlap of δ. These subregions can be constructed in several

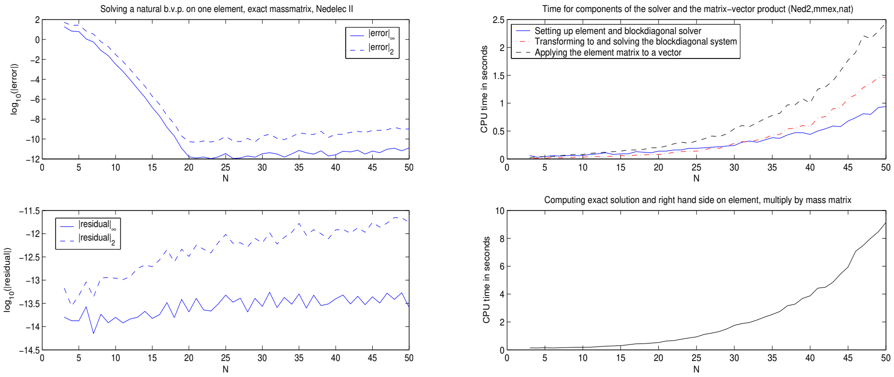

Fig. 1. Fast direct solution of a natural boundary value problem on one Nédélec 2 element of degree N by block tensor block diagonalization in 3D.

ways, e.g., by extending subdomains by a fixed overlap δ in all directions, or by finding vertex centered subdomains that overlap by δ. The theory does not require the $\Omega'_{j,\delta}$ to be unions of spectral elements, they can also just contain rectangular subsets of spectral elements. Most of our early computations (and the numerical results that we show in this paper) were performed on $2 \times 2 \times 2$ vertex centered assemblies of subdomains (taken as single spectral elements).

The local spaces V_j are the linear span of the basis functions associated to Gauss-Lobatto-Legendre points in $\Omega'_{j,\delta}$. In general, the support of functions in V_j will be larger than $\Omega'_{j,\delta}$, but if one only considers the Gauss-Lobatto-Legendre grid, they vanish on grid points outside $\Omega'_{j,\delta}$. On the local spaces we use exact solvers which corresponds to inversion of a submatrix of K. In the $2 \times 2 \times 2$ case, or in all cases where the overlapping regions only contain entire spectral elements, the local solver corresponds to the solution of a standard tangential value problem on the box made from these elements. In any case, the local solver can be implemented using the direct fast solvers introduced in the previous section.

The coarse space V_0 is a low-order Nédélec spectral element space of uniform degree N_0 defined on the coarse (subdomain level) mesh. We use the direct solvers of the last section as exact solvers. In the standard way, the local and the coarse solvers define local projections T_i and T_0 that can be used to implement different overlapping Schwarz methods. In this paper, we only consider the additive operator: a two-level additive Schwarz method T_{as2} defined by

$$T_{as2} = T_0 + \sum_{i \geq 1} T_i$$

We recall that this preconditioner gives optimal results in two dimensions; both iteration numbers and condition numbers are bounded by small constants for an increasing number of subdomains and degree, if there is a generous overlap that does not cut through spectral elements. It is also very robust against changes in α and β over a wide range of magnitudes. For minimal overlap that cuts through elements, iteration and condition numbers increase with increasing degree consistent with a linear growth, the dependence on the relative overlap could be both consistent with linear or quadratic growth (Hientzsch [2003, 2002]).

Table 1 and figure 2 suggest similar behavior in three dimensions.

We refer to Hientzsch [2002] for a proof of a condition number estimate which improves on the one given in Hientzsch [2003], in that it does not depend on the coefficients α and β, to wit for the case of overlapping regions made out of entire spectral elements

$$\kappa(T_{as2}) \leq C\left(1 + \left(\frac{H}{\delta}\right)^2\right)$$

and for the general case, with $\gamma \leq 1$ in two dimensions, and with $\gamma \leq 2$ in three dimensions,

$$\kappa(T_{as2}) \leq CN^{\gamma}\left(1 + \left(\frac{H}{\delta}\right)^2\right)$$

In both cases, C is independent of N, H, δ, α and β.

Table 1. Comparison of different methods for $\alpha = \beta = 1$, $5 \times 5 \times 5$ subdomains, Nédélec 2 elements of degree 10, $2 \times 2 \times 2$ overlapping subdomains, reduction of residual norm by 10^{-6}.

# of levels	iter	$\kappa_{est}(K)$	$\|\|\text{error}\|\|_\infty$	t_{CPU} in s
one	25	17.39	4.23e-06	187.26
two ($N_0 = 2$)	22	8.89	3.62e-06	165.07
two ($N_0 = 3$)	21	8.94	8.10e-06	155.62
two ($N_0 = 4$)	21	8.46	9.79e-06	157.18
two ($N_0 = 5$)	20	8.16	1.28e-05	154.51

4 Balancing Neumann-Neumann

Balancing Neumann-Neumann preconditioners are examples of iterative substructuring methods. Here, one iterates on the Schur complement system $Su_S = f_S$ with respect to the shared degrees of freedom u_S on the subdomain interfaces. To apply the matrix-vector product Su_S, one adds up the local contributions from the local Schur complements $S^{(i)}$ from each subdomain. This requires the solution of an essential (Dirichlet) boundary value problem per element. Balancing Neumann-Neumann preconditioners are hybrid methods, with alternating balancing and Neumann-Neumann steps (see, e.g., [Smith

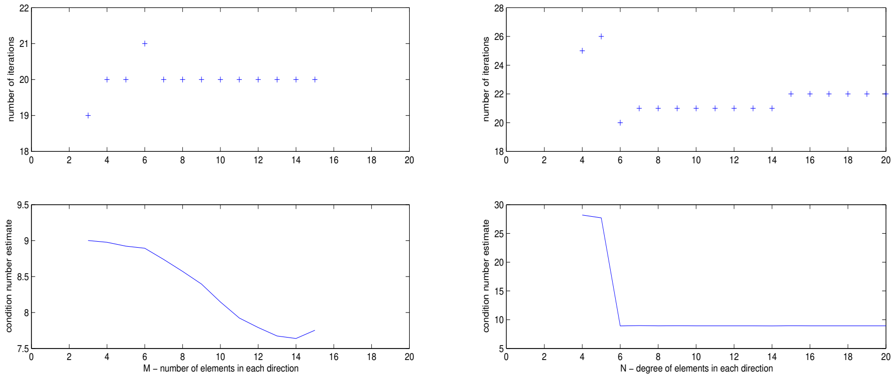

Fig. 2. Two-Level additive overlapping Schwarz method, $\alpha = \beta = 1$, $2 \times 2 \times 2$ overlapping subdomains, reduction of residual norm by 10^{-6}, $M \times M \times M$ subdomains of degree N. On the left, $N = 6$. On the right, $M = 5$.

et al., 1996, Section 4.3.3]). The Neumann-Neumann step requires the application of the inverse of the local Schur complement which can be implemented by the solution of a local Neumann problem, and also requires some diagonal scaling. The balancing step constitutes the coarse level correction, and directly inverts the Schur complement restricted to well-chosen coarse basis functions. We use the fast direct solvers from the second section to solve the Dirichlet and Neumann problems; for the coarse grid correction, in general, a general purpose factorization routine has to be used. We use the standard partition-of-unity diagonal scaling. Experiments have been run with different coarse basis functions and variable damping of the coarse grid corrections. We have been able to develop seemingly optimal and efficient damped coarse grid corrections for the two-dimensional case, and made some progress for some lower-order cases in the three-dimensional case, but we do not yet have general optimal balancing steps for the three-dimensional case.

Finally, we present in figure 3 some numerical experiments for two cases in two dimensions for several different balancing steps. We intend to present more complete experiments for two and three dimensions and theory for the two-dimensional case in future work.

As coarse grid functions we have chosen the standard partition of unity, either one function for the whole subdomain (BNN1), one for each component (BNN2), or one for each edge/face (BNN3). γ is the damping factor for the coarse grid correction. In figure 3, we see that in both cases, one coarse grid function per subdomain is not enough, even if we allow damping, but that one coarse grid function per edge gives an efficient method.

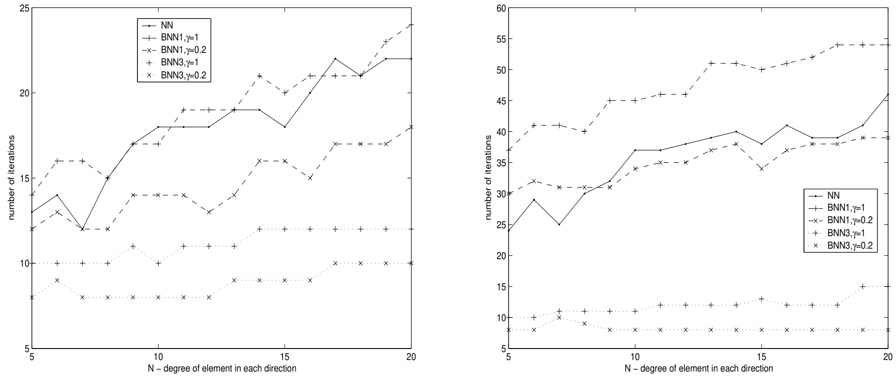

Fig. 3. Balancing Neummann-Neumann methods in two dimensions. Comparison of different coarse bases in the balancing step. $\alpha = \beta = 1$, reduction of residual norm by 10^{-6}, $M \times M \times M$ subdomains of degree N. On the left, $M = 5$. On the right, $M = 10$.

References

Faker Ben Belgacem and Christine Bernardi. Spectral element discretization of the Maxwell equations. *Math. Comp.*, 68(228):1497–1520, 1999.

Bernhard Hientzsch. *Fast Solvers and Domain Decomposition Preconditioners for Spectral Element Discretizations of Problems in H(curl)*. PhD thesis, Courant Institute of Mathematical Sciences, September 2001. Technical Report TR2001-823, Department of Computer Science, Courant Institute.

Bernhard Hientzsch. Overlapping Schwarz preconditioners for spectral Nédélec elements for a model problem in H(curl). Technical Report TR2002-834, Department of Computer Science, Courant Institute of Matical Sciences, November 2002.

Bernhard Hientzsch. Fast solvers and preconditioners for spectral Nédélec element discretizations of a model problem in H(curl). In Ismael Herrera, David E. Keyes, Olof B. Widlund, and Robert Yates, editors, *Domain Decomposition Methods in Science and Engineering*, pages 427–433. National Autonomous University of Mexico (UNAM), Mexico City, Mexico, 2003. Proceedings of the 14th International Conference on Domain Decomposition Methods in Cocoyoc, Mexico, January 6-11, 2002.

Peter Monk. On the *p*- and *hp*-extension of Nédélec's curl-conforming elements. *J. Comput. Appl. Math.*, 53(1):117–137, 1994. ISSN 0377-0427.

J.-C. Nédélec. Mixed finite elements in R^3. *Numer. Math.*, 35:315–341, 1980.

J.-C. Nédélec. A new family of mixed finite elements in R^3. *Numer. Math.*, 50:57–81, 1986.

Barry F. Smith, Petter E. Bjørstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.