SIAM J. SCI. COMPUT.
Vol. 31, No. 6, pp. 4422–4451
© 2010 Society for Industrial and Applied Mathematics

FINITE-ELEMENT PRECONDITIONING OF G-NI SPECTRAL METHODS*

CLAUDIO CANUTO†, PAOLA GERVASIO‡, AND ALFIO QUARTERONI§

Abstract. Several old and new finite-element preconditioners for nodal-based spectral discretizations of $-\Delta u = f$ in the domain $\Omega = (-1,1)^d$ ($d = 2$ or 3), with Dirichlet or Neumann boundary conditions, are considered and compared in terms of both condition number and computational efficiency. The computational domain covers the case of classical single-domain spectral approximations (see [C. Canuto et al., *Spectral Methods. Fundamentals in Single Domains*, Springer, Heidelberg, 2006]), as well as that of more general spectral-element methods in which the preconditioners are expressed in terms of local (upon every element) algebraic solvers. The primal spectral approximation is based on the Galerkin approach with numerical integration (G-NI) at the Legendre–Gauss–Lobatto (LGL) nodes in the domain. The preconditioning matrices rely on either $\mathbb{P}_1$, $\mathbb{Q}_1$, or $\mathbb{Q}_{1,NI}$ (i.e., with numerical integration) finite elements on meshes whose vertices coincide with the LGL nodes used for the spectral approximation. The analysis highlights certain preconditioners, which yield the solution at an overall cost proportional to N^{d+1}, where N denotes the polynomial degree in each direction.

Key words. spectral method, finite elements, preconditioned iterative methods, elliptic equations

AMS subject classifications. 65F10, 65N35

DOI. 10.1137/090746367

1. Introduction.

Spectral methods are currently recognized as among the fundamental successful strategies for numerically solving partial differential equations. Their distinguishing feature is the intrinsic ability to yield a high rate of convergence (even exponentially fast) for smooth solutions. Their potential drawback arises from the severe condition number (higher than those of the corresponding finite-element or finite-difference matrices, for instance) of the associated algebraic system. This fact has called, over the years, for the development of ad hoc preconditioning strategies. In this regard, a major conceptual breakthrough for preconditioning nodal-based spectral methods has been the intuition (early pursued by Orszag [22], Deville and Mund [13], and Canuto and Quarteroni [8]) of using lower-order approximation matrices (those of finite differences or finite elements) built up on the same grid involved in the spectral discretization.

Orszag considered the matrix arising from the Fourier or Chebyshev collocation approximation of the Laplace operator with or without periodic boundary conditions; he proposed to precondition it by the second-order finite-difference matrix built up on the same collocation grid. Successively, Canuto and Quarteroni extended the finite-difference preconditioner to the variable-coefficients differential operator $Lu = -\nabla\cdot(\nu(x)\nabla u)+\alpha(x)u$ with Dirichlet boundary conditions; moreover, they introduced a bilinear Lagrange finite-element preconditioner. Independently, Deville and

*Received by the editors January 12, 2009; accepted for publication (in revised form) October 27, 2009; published electronically January 15, 2010.
http://www.siam.org/journals/sisc/31-6/74636.html
†Department of Mathematics, Politecnico di Torino, 10129 Torino, Italy (ccanuto@calvino.polito.it).
‡Department of Mathematics, University of Brescia, 25133 Brescia, Italy (gervasio@ing.unibs.it).
§Laboratory of Modeling and Scientific Computing (MOX), Department of Mathematics, Politecnico di Milano, 20133 Milan, Italy, and CMCS-EPFL, CH-1015 Lausanne, Switzerland (alfio.quarteroni@epfl.ch).

Mund proposed to precondition the Chebyshev collocation matrix by either bilinear or biquadratic Lagrange finite elements, as well as by bicubic Hermite elements. They investigated the efficiency of such preconditioners and deduced that bilinear Lagrange elements produced spectral accuracy with the minimum computational work. In the successive paper [14], Deville and Mund analyzed the spectrum of the Chebyshev collocation matrix when preconditioned by finite differences, Lagrange or Hermite finite elements, versus the variation of both boundary conditions and operator coefficients. In [26], Quarteroni and Zampieri proposed and investigated a bilinear finite-element preconditioner for the matrix arising from a Galerkin discrete variational formulation of the Laplace equation with either Neumann or Dirichlet boundary conditions; the use of numerical integration based on the Legendre–Gauss–Lobatto grid yields the equivalence of the Galerkin (or *weak*) approach with the collocation (or *strong*) approach, up to a multiplication by a diagonal matrix coinciding with the spectral mass matrix. The use of Legendre expansions instead of Chebyshev ones permits the formulation of spectral methods in a weak form, an alternative to the strong one, yielding greater generality and flexibility. Indeed, the weak Legendre formulation prevailed over strong forms, and various preconditioners based on either linear ($\mathbb{P}_1$) or bilinear ($\mathbb{Q}_1$) finite elements were used also inside multidomain strategies (see, e.g., [18, 9]).

What follows is a brief account on known theoretical results for the abovementioned preconditioners on the Laplace operator. Orszag [22] proved that the condition number of the Fourier collocation matrix (for periodic boundary conditions) preconditioned by finite differences is bounded by $\pi^2/4$. The same result was established by Haldenwang et al. [20] for the Chebyshev collocation case. Canuto [6] and Parter and Rothman [25] proved the so-called *finite-element–spectral* equivalence (sometimes referred to as the FEM-SEM equivalence) in both L^2- and H^1-norms for univariate functions; in particular, the equivalence in the H^1-norm states that the linear finite-element stiffness matrix built on the Legendre–Gauss–Lobatto grid is spectrally equivalent to the nodal-based Legendre Galerkin stiffness matrix. Thanks to the tensorial structure of interpolation operators, such results are easily extended to multivariate functions for multilinear ($\mathbb{Q}_1$) finite elements. Moreover, Parter and Rothman [25] also proved the equivalence for $\mathbb{P}_1$ elements in two dimensions. Finally, Parter [23, 24] investigated the preconditioning of the Legendre collocation (or strong) spectral matrix by both bilinear finite elements and finite differences, and he proved that the eigenvalues of the preconditioned matrices are bounded in modulus independently of N.

In this paper we further elaborate on this algorithmic and theoretical pathway. First, we propose several new kinds of preconditioners and analyze their theoretical behavior. Next, we extensively investigate their numerical performance and compare it with those of already existing preconditioners. More specifically, in addition to classical $\mathbb{P}_1$ and $\mathbb{Q}_1$ finite-element preconditioners, we consider a number of preconditioners based on $\mathbb{Q}_1$ finite elements with numerical integration. We distinguish between strong and weak forms of the reference differential problem, and we adapt finite-element preconditioners to both forms. Moreover, since strong forms are not symmetric, to allow for the use of the conjugate-gradient algorithm, we also propose symmetrized-strong versions of such preconditioners.

A careful numerical investigation, supported in many cases by theoretical proofs, shows that all preconditioners considered in this paper are spectrally equivalent to the corresponding Legendre spectral matrices. The inspection of the condition numbers of the preconditioned matrices indicates that the preconditioner based on the

$\mathbb{Q}_1$ approach for the strong form gives the smallest condition number. However, if we measure preconditioner efficiency in terms of CPU-time, the best performance is obtained by the preconditioners based on the $\mathbb{Q}_1$ approach with numerical integration, for both 2D and 3D (two- and three-dimensional) geometries. Symmetrized-strong preconditioners show very good theoretical properties, i.e., their iterative condition numbers are very small, yet they are not efficient from the computational point of view due to the higher cost of each iteration. Finally, we have considered three different algebraic solvers to compute the preconditioned residual at each conjugate-gradient iteration: the classical Cholesky factorization, a multifrontal method with nested dissection ordering, and a preconditioned conjugate-gradient with inexact factorization. The computational performances of each of these solvers have been carefully measured and compared.

Our analysis will concern the case of a reference computational domain, a square in 2D, a cube in 3D. This choice has a twofold motivation. On the one hand, spectral methods are still widely used nowadays to approximate (initial-) boundary-value problems in a single domain (see [7]): the latter is either the reference hypercube $\hat{\Omega} = (-1,1)^d$ $(d = 2,3)$ or another domain Ω_s that can be mapped into $\hat{\Omega}$ by an invertible regular map $\mathbf{F}_s : \Omega_s \to \hat{\Omega}$. On the other hand, our results may also be of interest in the framework of spectral element methods (SEMs). The latter are set up on a computational domain Ω, possibly featuring a complex shape, that is split into smaller subdomains, say $\Omega_m,\ m = 1, \ldots, M$, which may or may not overlap. In this context, domain decomposition preconditioners are typically built upon an additive sum of local terms, which involve restriction and prolongation matrices and local algebraic solvers on each subdomain, say, for the sake of conciseness, $\sum_m R_m A_m^{-1} R_m$. The solution of the local systems $A_m \mathbf{w}_m = \mathbf{r}_m$ on each subdomain Ω_m must therefore be efficiently addressed by either direct factorization algorithms (in the case where the size of local matrices is moderate) or preconditioned iterative algorithms. The latter can benefit from the preconditioning strategies developed in this paper on the reference domain $\hat{\Omega}$.

We remark that, even if the focus of the paper is on using FEM low-order discretizations as preconditioners for nodal-based spectral discretizations, many alternative choices are possible in the context of modal-based methods, e.g., those based on multigrid and multilevel techniques [4, 5].

An outline of the paper is as follows. In section 2 we review both spectral and finite-element discretizations of the Laplace problem. In section 3 we introduce all the preconditioners discussed in the paper. In section 4 we theoretically analyze the iterative condition numbers of all the weak forms and of those strong forms based on both $\mathbb{Q}_1$ and $\mathbb{Q}_1$ with numerical integration approaches. We also briefly consider the case of variable diffusion coefficient and Neumann boundary conditions. In section 5 we introduce the algebraic solvers for computing the preconditioned residuals, and we report a detailed analysis of the computational costs of all possible strategies.

2. Galerkin-numerical integration and finite-element matrices. We first consider the homogeneous Dirichlet boundary-value problem

$$-\Delta u = f \quad \text{in } \Omega = (-1,1)^d, \qquad u = 0 \quad \text{on } \partial\Omega, \tag{2.1}$$

where $d = 1, 2, 3$ and $f \in C^0(\overline{\Omega})$. Other boundary conditions will be discussed later on.

The Legendre Galerkin–numerical integration (G-NI) discretization of this problem consists of finding a polynomial u_N in $\mathbb{Q}_N^0(\Omega)$ (the space of the algebraic polyno-

mials of degree $\leq N$ in each direction, vanishing on $\partial\Omega$) satisfying

$$(\nabla u_N, \nabla v_N)_N = (f, v_N)_N \qquad \text{for all } v_N \in \mathbb{Q}_N^0(\Omega), \tag{2.2}$$

where $(\cdot,\cdot)_N$ denotes the d-dimensional Legendre–Gauss–Lobatto (LGL) discrete inner product in Ω; it can be written as

$$(u, v)_N = \sum_{j=1}^{(N-1)^d} u(\mathbf{x}_j) v(\mathbf{x}_j) w_j, \qquad u, v \in \mathbb{Q}_N^0(\Omega), \tag{2.3}$$

where $\mathbf{x}_j$ (for $j = 1, \ldots, (N-1)^d$) denote the $(N-1)^d$ interior LGL nodes (numbered in lexicographical order) and w_j are the corresponding weights (see [7, sect. 2.2]).

The algebraic system corresponding to (2.2) reads

$$K_{GNI}\mathbf{u} = M_{GNI}\mathbf{f}, \tag{2.4}$$

where $\mathbf{u}$ and $\mathbf{f}$ are the vectors whose components are the values of u_N and f at $\mathbf{x}_j$. Correspondingly, ψ_j (for $j = 1, \ldots, (N-1)^d$) will denote the characteristic Lagrange polynomial at $\mathbf{x}_j$, defined by the conditions $\psi_j \in \mathbb{Q}_N^0(\Omega)$ and $\psi_j(\mathbf{x}_k) = \delta_{jk}$ for all $k = 1, \ldots, (N-1)^d$. Thus, the symmetric positive-definite (s.p.d.) stiffness and mass matrices K_{GNI} and M_{GNI} are defined as

$$(K_{GNI})_{ij} = (\nabla\psi_j, \nabla\psi_i)_N, \qquad (M_{GNI})_{ij} = (\psi_j, \psi_i)_N, \tag{2.5}$$

for $i, j = 1, \ldots, (N-1)^d$. While the algebraic system (2.4) corresponds to the discretization of the *weak* form of (2.1), the linear system

$$M_{GNI}^{-1} K_{GNI}\mathbf{u} = \mathbf{f} \tag{2.6}$$

corresponds to the discretization of (2.1) by the collocation approach (see [7, 3]), also referred to as the *strong* form. In view of an efficient iterative solution, system (2.6) can be equivalently written in symmetric form as

$$(M_{GNI}^{-1/2} K_{GNI} M_{GNI}^{-1/2})(M_{GNI}^{1/2}\mathbf{u}) = M_{GNI}^{1/2}\mathbf{f}, \tag{2.7}$$

where, given any s.p.d. matrix B, $B^{1/2}$ denotes its *square root*, i.e., the matrix such that $B^{1/2}B^{1/2} = B$, while $B^{-1/2}$ is a shorthand notation for $(B^{1/2})^{-1}$. System (2.7) will be referred to as the *symmetrized-strong* form.

We will write systems (2.4), (2.6), and (2.7) in the general form

$$L\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{2.8}$$

where, for $\mathbf{v} = \mathbf{u}$ or $\mathbf{f}$, the symbol $\tilde{\mathbf{v}}$ means $\mathbf{v}$ in both (2.4) and (2.6), while it stands for $M_{GNI}^{1/2}\mathbf{v}$ in (2.7).

The stiffness matrix K_{GNI} is structured with lower and upper bandwidth equal to $nb = (N-1)^{d-1}(N-2)$; the total number of its nonzero elements is about $nz = d \cdot N^{(d+1)}$. Thanks to the orthogonality of Lagrange basis functions ψ_j in the discrete inner product $(\cdot,\cdot)_N$, the mass matrix M_{GNI} is diagonal.

The extremal eigenvalues of K_{GNI} and M_{GNI} satisfy the following estimates [3, 21, 7],

$$\begin{aligned} &\lambda_{\min}(K_{GNI}) \asymp N^{-2}\,, \qquad &&\lambda_{\max}(K_{GNI}) \asymp N\,, \\ &\lambda_{\min}(M_{GNI}) \asymp N^{-2}\,, \qquad &&\lambda_{\max}(M_{GNI}) \asymp N^{-1}\,, \end{aligned} \tag{2.9}$$

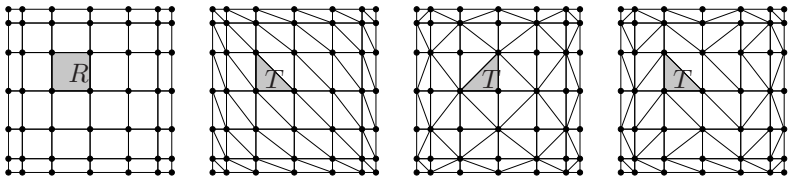

FIGURE 2.1. *Finite elements in* Ω *induced by the* 2*D LGL grid.* (a) $\mathbb{Q}_1$, (b) $\mathbb{P}_1$ *with all triangles oriented in the same way,* (c) $\mathbb{P}_1$ *with alternating orientation,* (d) $\mathbb{P}_1$ *with random orientation.*

and this yields

$$\mathcal{K}(K_{GNI}) \asymp N^3, \qquad \mathcal{K}(M_{GNI}^{-1}K_{GNI}) = \mathcal{K}(M_{GNI}^{-1/2}K_{GNI}M_{GNI}^{-1/2}) \asymp N^4, \tag{2.10}$$

where $\mathcal{K}(A) := \max_i \lambda_i(A)/\min_i \lambda_i(A)$ is the so-called *iterative condition number* of any matrix A similar to an s.p.d. matrix.

The matrix $M_{GNI}^{-1}K_{GNI}$ is similar to an s.p.d. matrix since both M_{GNI} and K_{GNI} are s.p.d. matrices. Moreover, $M_{GNI}^{-1/2}K_{GNI}M_{GNI}^{-1/2}$ is symmetric and similar to $M_{GNI}^{-1}K_{GNI}$.

It is well known (see, e.g., Figure 4.46 in [7]) that the solution of (2.8) by a direct method is efficient only for very small values of N (on the order of 10). For larger systems, preconditioned iterative techniques should be preferred. Among them, algebraic preconditioners, such as those based on the diagonal or the incomplete Cholesky factorization of the stiffness matrix, yield iterative condition numbers of the preconditioned matrix which grow linearly with respect to N (see Figures 4.44–4.45 in [7]). On the other hand, preconditioners based on the sparse matrices generated by low-order finite-element discretizations on the Gauss-Lobatto grid may yield iterative condition numbers not only independent of N but also extremely small (close to unity).

In what follows, we will carry on a thorough comparative investigation of the performances of several finite-element preconditioners; each of them is inspired by one of the weak, strong, or symmetrized-strong forms, (2.4), (2.6), or (2.7), introduced above.

The finite-element matrices we are going to consider are built on the partition (or mesh) of $\overline{\Omega} = [-1,1]^d$, made of all the rectangles in two or parallelepipeds in three dimensions (in general, d-intervals denoted by R) whose vertices are two consecutive LGL nodes in each direction (see Figure 2.1(a)). On such a mesh, piecewise d-linear shape functions are defined, yielding $\mathbb{Q}_1$ finite elements. Alternatively, one can build the finite-element preconditioners on the mesh of $\overline{\Omega}$ made of triangles or tetrahedra (in general, simplices denoted by T), still with vertices at the LGL nodes (see Figures 2.1(b)–(d) and 2.2), corresponding to $\mathbb{P}_1$ finite elements. In 2D geometries, two triangles T are obtained by splitting each rectangle R by one of its diagonals; we distinguish among uniformly oriented meshes as in Figure 2.1(b), alternating meshes as in Figure 2.1(c), and random meshes as in Figure 2.1(d). When $\Omega \subset \mathbb{R}^3$, we have considered two splittings of a hexahedron into tetrahedra, with five or six elements, as shown in the left or right portions of Figure 2.2, respectively. The latter choice allows us to put side by side hexahedra with the same internal splitting, leading to a globally uniformly oriented mesh. The former choice requires two adjacent hexahedra to have

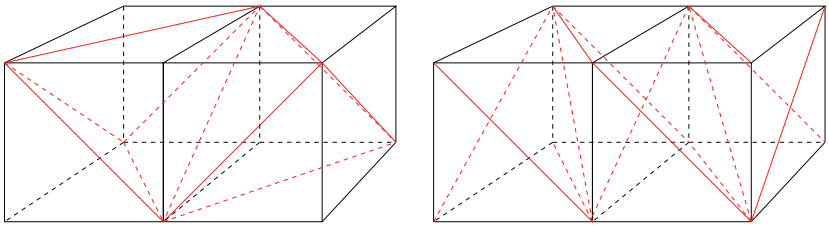

FIGURE 2.2. *At left (resp., right), two adjacent hexahedra, each of them partitioned into five (resp., six) tetrahedra, which have two consecutive LGL nodes (in each direction) as vertices.*

complementary splittings which reflect into each other across the common interface; they generate a global alternating mesh.

Let φ_j (with $j = 1, \ldots, (N-1)^d$) denote the $\mathbb{Q}_1$ finite-element characteristic Lagrange function at an interior $\mathbf{x}_j$, i.e., the globally continuous, piecewise d-linear function in each R, vanishing on $\partial\Omega$, such that $\varphi_j(\mathbf{x}_k) = \delta_{jk}$ for all $k = 1, \ldots, (N-1)^d$. The associated finite-element stiffness matrix $K_{\mathbb{Q}_1}$ is defined by

$$(K_{\mathbb{Q}_1})_{ij} = (\nabla\varphi_j, \nabla\varphi_i), \qquad i, j = 1, \ldots, (N-1)^d, \tag{2.11}$$

where $(\cdot, \cdot)$ denotes the standard L^2-inner product in Ω. We will also consider its numerical approximation $K_{\mathbb{Q}_{1,NI}}$, defined by

$$(K_{\mathbb{Q}_{1,NI}})_{ij} = \sum_R \int_R \Pi_{1,R}(\nabla\varphi_j^T \nabla\varphi_i)\, d\mathbf{x}, \qquad i, j = 1, \ldots, (N-1)^d, \tag{2.12}$$

where $\Pi_{1,R}(g)$ denotes the d-linear interpolant of a function g at the vertices of R; this corresponds to using the trapezoidal numerical integration formula in each R. The finite-element mass matrix $M_{\mathbb{Q}_1}$ is defined by

$$(M_{\mathbb{Q}_1})_{ij} = (\varphi_j, \varphi_i), \qquad i, j = 1, \ldots, (N-1)^d, \tag{2.13}$$

and its diagonal approximation is the *lumped mass matrix* $M_{\mathbb{Q}_{1,NI}}$, defined by

$$(M_{\mathbb{Q}_{1,NI}})_{ij} = \sum_R \int_R \Pi_{1,R}(\varphi_j \varphi_i)\, d\mathbf{x}, \qquad i, j = 1, \ldots, (N-1)^d. \tag{2.14}$$

We note that $K_{\mathbb{Q}_1} = K_{\mathbb{Q}_{1,NI}}$ when $d = 1$, thanks to the exactness of the trapezoidal rule for linear functions. In contrast, $M_{\mathbb{Q}_1} \neq M_{\mathbb{Q}_{1,NI}}$ for $d = 1, 2, 3$.

Finally, for $d = 2, 3$ and a simplicial mesh in $\overline{\Omega}$, let $\tilde{\varphi}_j$ denote the $\mathbb{P}_1$ finite-element characteristic Lagrange function at interior $\mathbf{x}_j$, i.e., the globally continuous, piecewise linear function in each T, vanishing on $\partial\Omega$, such that $\tilde{\varphi}_j(\mathbf{x}_k) = \delta_{jk}$ for all $k = 1, \ldots, (N-1)^d$. The resulting stiffness and mass matrices are

$$(K_{\mathbb{P}_1})_{ij} = (\nabla\tilde{\varphi}_j, \nabla\tilde{\varphi}_i), \quad (M_{\mathbb{P}_1})_{ij} = (\tilde{\varphi}_j, \tilde{\varphi}_i), \quad \text{for } i, j = 1, \ldots, (N-1)^d. \tag{2.15}$$

Remark 1. Since the computational domain $\Omega \subset \mathbb{R}^2$ is a rectangle, the stiffness matrices $K_{\mathbb{Q}_{1,NI}}$ and $K_{\mathbb{P}_1}$ coincide independently of the orientation of the triangles of the mesh, as can be checked in a straightforward manner. Moreover, denoting by L_{FD} the classic five-point centered finite-difference Laplace approximation matrix, the identity $L_{FD} = M_{\mathbb{Q}_{1,NI}}^{-1} K_{\mathbb{Q}_{1,NI}}$ holds.

The matrix K_{FE}, chosen among $K_{\mathbb{Q}_1}$, $K_{\mathbb{Q}_{1,NI}}$, and $K_{\mathbb{P}_1}$, may be used to precondition system (2.4) in *weak* form; the matrix $M_{FE}^{-1}K_{FE}$, with M_{FE} chosen among $M_{\mathbb{Q}_1}$, $M_{\mathbb{Q}_{1,NI}}$, and $M_{\mathbb{P}_1}$, may be invoked to precondition system (2.6) in *strong* form; while the matrix $M_{FE}^{-1/2}K_{FE}M_{FE}^{-1/2}$ may be useful to precondition system (2.7) in *symmetrized-strong* form.

We introduce the space $\mathbb{Q}_N(\Omega)$ of algebraic polynomials defined on Ω, of degree $\leq N$ in each direction (a possible basis for $\mathbb{Q}_N$ is given by the characteristic Lagrange functions ψ_j associated with all nodes of the LGL grid); the space V_h of continuous functions on Ω which are d-linear on each d-interval R induced by the LGL mesh of Ω (the functions φ_j associated with all nodes of the LGL grid form a basis for V_h); and the space W_h of continuous functions on Ω which are linear on each simplex T induced by the LGL mesh of Ω (the functions $\tilde{\varphi}_j$ associated with all nodes of the LGL grid form a basis for W_h). V_h^0 and W_h^0 will denote the subspaces of V_h and W_h, respectively, of vanishing functions at the boundary $\partial\Omega$.

For any $v_N \in \mathbb{Q}_N(\Omega)$ we denote by $v_h \in V_h$ the continuous piecewise d-linear interpolation of v_N at LGL nodes. It is well known [6, 25] that v_N and v_h are linked together by an algebraic interpolation isomorphism. Moreover, for $d = 2, 3$ and for any $v_N \in \mathbb{Q}_N(\Omega)$ we will denote by $w_h \in W_h$ the continuous piecewise linear interpolation of v_N at LGL nodes. Even if, for any given N, the nodes of the mesh in Ω are uniquely defined, the mesh of simplexes T is not, as we have discussed above. This implies that w_h will depend on the mesh chosen. For a fixed mesh, w_h and v_N (and then also w_h and v_h) are linked together by an algebraic interpolation isomorphism.

For any $N \geq 2$, given $v_N \in \mathbb{Q}_N^0(\Omega)$ (or equivalently either $v_h \in V_h^0$ or $w_h \in W_h^0$), $\mathbf{v} \in \mathbb{R}^{(N-1)^d}$ will be the array whose components are the values $v_N(\mathbf{x}_j) = v_h(\mathbf{x}_j) = w_h(\mathbf{x}_j)$ at the interior LGL nodes $\mathbf{x}_j$.

3. Preconditioners. The finite-element matrices introduced above can be suitably combined to produce preconditioned matrices and systems in order to solve (2.8). We will denote by H any preconditioning matrix for the spectral matrix L which appears in (2.8), so that the corresponding (left) preconditioned system will be

$$H^{-1}L\tilde{\mathbf{u}} = H^{-1}\tilde{\mathbf{f}}. \tag{3.1}$$

In what follows we will set $P = H^{-1}L$.

We have considered eleven possible expressions for P, which, for the reader's convenience, are listed in Table 3.1. (A subset of these combinations was already reported in [7].) Three preconditioned matrices, named $P_{\mathbb{Q}_1}^w$, $P_{\mathbb{Q}_{1,NI}}^w$, and $P_{\mathbb{P}_1}^w$, are based on the weak form (2.4); three others, named $P_{\mathbb{Q}_1}^s$, $P_{\mathbb{Q}_{1,NI}}^s$, and $P_{\mathbb{P}_1}^s$, are based on the strong form (2.6); finally, five preconditioners, named $P_{\mathbb{Q}_1}^{ss,rt}$, $P_{\mathbb{Q}_1}^{ss,ch}$, $P_{\mathbb{Q}_{1,NI}}^{ss,rt}$, $P_{\mathbb{P}_1}^{ss,ch}$, and $P_{\mathbb{P}_1}^{ss,rt}$, are symmetrized forms of the previous strong preconditioners. In particular, $P_{\mathbb{Q}_1}^{ss,rt}$ and $P_{\mathbb{Q}_1}^{ss,ch}$ are two symmetrized version of $P_{\mathbb{Q}_1}^s$, which differ from each other in the computation of $(M_{\mathbb{Q}_1})^{-1/2}$, as we are going to explain.

For any s.p.d. matrix B, its square root $B^{1/2}$ can be expressed as $B^{1/2} = W\Lambda^{1/2}W^T$, where Λ and W denote, respectively, the matrices of eigenvalues and eigenvectors of B. The matrix $P_{\mathbb{Q}_1}^{ss,rt}$ is defined starting from the square root of $M_{\mathbb{Q}_1}^{-1}$, computed in this way. However, when the computation of both Λ and W becomes too expensive, as an alternative to diagonalization, one can employ the Cholesky decomposition of B, namely $B = B_{Ch}B_{Ch}^T$, with B_{Ch} lower triangular; then, B_{Ch} replaces $B^{1/2}$. The matrix $P_{\mathbb{Q}_1}^{ss,ch}$ is defined accordingly. The matrix $P_{\mathbb{Q}_{1,NI}}^{ss,rt}$ is the symmetrized version of $P_{\mathbb{Q}_{1,NI}}^s$ (in this case $M_{\mathbb{Q}_{1,NI}}$ is diagonal with positive entries

TABLE 3.1
Preconditioned matrices and associated transformed linear systems $P\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$ *for* (2.4) *and* (2.6) $(with B^{-T} = (B^T)^{-1})$.

(3.2)	$P = H^{-1}L$	H (preconditioner)	L (spectral matrix)	$\tilde{\mathbf{u}}$	$\tilde{\mathbf{f}}$
(3.3)	$P^{w}_{\mathbb{Q}_1}$	$K_{\mathbb{Q}_1}$	K_{GNI}	$\mathbf{u}$	$M_{GNI}\mathbf{f}$
(3.4)	$P^{s}_{\mathbb{Q}_1}$	$(M_{\mathbb{Q}_1})^{-1}K_{\mathbb{Q}_1}$	$(M_{GNI})^{-1}K_{GNI}$	$\mathbf{u}$	$\mathbf{f}$
(3.5)	$P^{w}_{\mathbb{Q}_{1,NI}}$	$K_{\mathbb{Q}_{1,NI}}$	K_{GNI}	$\mathbf{u}$	$M_{GNI}\mathbf{f}$
(3.6)	$P^{s}_{\mathbb{Q}_{1,NI}}$	$(M_{\mathbb{Q}_{1,NI}})^{-1}K_{\mathbb{Q}_{1,NI}}$	$(M_{GNI})^{-1}K_{GNI}$	$\mathbf{u}$	$\mathbf{f}$
(3.7)	$P^{ss,rt}_{\mathbb{Q}_1}$	$(M_{\mathbb{Q}_1})^{-1/2}K_{\mathbb{Q}_1}(M_{\mathbb{Q}_1})^{-1/2}$	$(M_{GNI})^{-1/2}K_{GNI}(M_{GNI})^{-1/2}$	$(M_{GNI})^{1/2}\mathbf{u}$	$(M_{GNI})^{1/2}\mathbf{f}$
(3.8)	$P^{ss,ch}_{\mathbb{Q}_1}$	$(M_{\mathbb{Q}_{1,Ch}})^{-1}K_{\mathbb{Q}_1}(M_{\mathbb{Q}_{1,Ch}})^{-T}$	$(M_{GNI})^{-1/2}K_{GNI}(M_{GNI})^{-1/2}$	$(M_{GNI})^{1/2}\mathbf{u}$	$(M_{GNI})^{1/2}\mathbf{f}$
(3.9)	$P^{ss,rt}_{\mathbb{Q}_{1,NI}}$	$(M_{\mathbb{Q}_{1,NI}})^{-1/2}K_{\mathbb{Q}_{1,NI}}(M_{\mathbb{Q}_{1,NI}})^{-1/2}$	$(M_{GNI})^{-1/2}K_{GNI}(M_{GNI})^{-1/2}$	$(M_{GNI})^{1/2}\mathbf{u}$	$(M_{GNI})^{1/2}\mathbf{f}$
(3.10)	$P^{w}_{\mathbb{P}_1}$	$K_{\mathbb{P}_1}$	K_{GNI}	$\mathbf{u}$	$M_{GNI}\mathbf{f}$
(3.11)	$P^{s}_{\mathbb{P}_1}$	$(M_{\mathbb{P}_1})^{-1}K_{\mathbb{P}_1}$	$(M_{GNI})^{-1}K_{GNI}$	$\mathbf{u}$	$\mathbf{f}$
(3.12)	$P^{ss,rt}_{\mathbb{P}_1}$	$(M_{\mathbb{P}_1})^{-1/2}K_{\mathbb{P}_1}(M_{\mathbb{P}_1})^{-1/2}$	$(M_{GNI})^{-1/2}K_{GNI}(M_{GNI})^{-1/2}$	$(M_{GNI})^{1/2}\mathbf{u}$	$(M_{GNI})^{1/2}\mathbf{f}$
(3.13)	$P^{ss,ch}_{\mathbb{P}_1}$	$(M_{\mathbb{P}_{1,Ch}})^{-1}K_{\mathbb{P}_1}(M_{\mathbb{P}_{1,Ch}})^{-T}$	$(M_{GNI})^{-1/2}K_{GNI}(M_{GNI})^{-1/2}$	$(M_{GNI})^{1/2}\mathbf{u}$	$(M_{GNI})^{1/2}\mathbf{f}$

and $(M^{-1/2}_{\mathbb{Q}_{1,NI}})_{ii} = (M^{-1}_{\mathbb{Q}_{1,NI}})^{1/2}_{ii})$, while $P^{ss,rt}_{\mathbb{P}_1}$ and $P^{ss,ch}_{\mathbb{P}_1}$ are symmetrized versions of $P^{s}_{\mathbb{P}_1}$ (again based either on the square root or the Cholesky factor of $(M_{\mathbb{P}_1})^{-1}$, respectively).

4. Condition number analysis. We first examine the iterative condition number of all the preconditioned matrices defined in (3.3)–(3.13). In order to simplify the exposition, matrices $P^{w}_{\mathbb{Q}_1}$, $P^{w}_{\mathbb{Q}_{1,NI}}$, and $P^{w}_{\mathbb{P}_1}$ will be referred to as *weak matrices*; $P^{s}_{\mathbb{Q}_1}$, $P^{s}_{\mathbb{Q}_{1,NI}}$, and $P^{s}_{\mathbb{P}_1}$ as *strong matrices*; and $P^{ss,rt}_{\mathbb{Q}_1}$, $P^{ss,ch}_{\mathbb{Q}_1}$, $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$, $P^{ss,rt}_{\mathbb{P}_1}$, and $P^{ss,ch}_{\mathbb{P}_1}$ as *symmetrized-strong matrices*.

In the following subsections, we review the theoretical results concerning weak and strong matrices. The symmetrized-strong matrices are similar to s.p.d. matrices; hence their eigenvalues are all real positive. No other theoretical result is available so far, so we refer to section 4.3 for numerical results.

4.1. Weak matrices. We begin by considering weak matrices. $P^{w}_{\mathbb{Q}_1}$, $P^{w}_{\mathbb{Q}_{1,NI}}$, and $P^{w}_{\mathbb{P}_1}$ have real positive eigenvalues, being products of two s.p.d. matrices. We start with $P^{w}_{\mathbb{Q}_1}$ and $P^{w}_{\mathbb{Q}_{1,NI}}$.

In order to analyze their iterative condition numbers we note that, since Ω is a Cartesian product of intervals, we can express both multidimensional mass and stiffness matrices, based on either $\mathbb{Q}_1$, $\mathbb{Q}_{1,NI}$, or $\mathbb{Q}_N$, as Kronecker products of 1D matrices; the latter will be denoted by the superindex (1).

By recalling that $K^{(1)}_{\mathbb{Q}_{1,NI}} \equiv K^{(1)}_{\mathbb{Q}_1}$ and that Ω is a Cartesian product of intervals, the following identities hold for $d = 2$,

$$K_{\mathbb{Q}_1} = M^{(1)}_{\mathbb{Q}_1} \otimes K^{(1)}_{\mathbb{Q}_1} + K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1}, \tag{4.1}$$

$$K_{\mathbb{Q}_{1,NI}} = M^{(1)}_{\mathbb{Q}_{1,NI}} \otimes K^{(1)}_{\mathbb{Q}_1} + K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_{1,NI}}, \tag{4.2}$$

$$K_{GNI} = M^{(1)}_{GNI} \otimes K^{(1)}_{GNI} + K^{(1)}_{GNI} \otimes M^{(1)}_{GNI}, \tag{4.3}$$

and for $d = 3$,

$$K_{\mathbb{Q}_1} = M^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1} \otimes K^{(1)}_{\mathbb{Q}_1} + M^{(1)}_{\mathbb{Q}_1} \otimes K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1} + K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1},$$

$$\begin{aligned} K_{\mathbb{Q}_{1,NI}} = {} & M^{(1)}_{\mathbb{Q}_{1,NI}} \otimes M^{(1)}_{\mathbb{Q}_{1,NI}} \otimes K^{(1)}_{\mathbb{Q}_1} + M^{(1)}_{\mathbb{Q}_{1,NI}} \otimes K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_{1,NI}} \\ & + K^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_{1,NI}} \otimes M^{(1)}_{\mathbb{Q}_{1,NI}}, \end{aligned}$$

$$K_{GNI} = M^{(1)}_{GNI} \otimes M^{(1)}_{GNI} \otimes K^{(1)}_{GNI} + M^{(1)}_{GNI} \otimes K^{(1)}_{GNI} \otimes M^{(1)}_{GNI} + K^{(1)}_{GNI} \otimes M^{(1)}_{GNI} \otimes M^{(1)}_{GNI},$$

where $\otimes$ denotes the Kronecker product of matrices; that is, the block C_{ij} of $C = A \otimes B$ is given by $C_{ij} = a_{ij}B$.

We will use the following well-known property (see, e.g., [7, Chap. 7]): if A_i, B_i, $i = 1, 2$, are s.p.d. matrices of order n such that

$$\mathbf{v}^T A_i \mathbf{v} \le \alpha_i \mathbf{v}^T B_i \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^n$$

for suitable choice of real positive coefficients $\alpha_1,\ \alpha_2$, then one has

$$\mathbf{v}^T (A_1 \otimes A_2)\mathbf{v} \le \alpha_1 \alpha_2 \mathbf{v}^T (B_1 \otimes B_2)\mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{n^2}. \tag{4.4}$$

Henceforth, the bounds on multidimensional stiffness matrices immediately follow from bounds on 1D matrices.

By definition (2.14), the nonzero entries of $M^{(1)}_{\mathbb{Q}_{1,NI}}$ are the weights of the composite trapezoidal rule, i.e., $(M^{(1)}_{\mathbb{Q}_{1,NI}})_{ii} = \frac{x_{i+1}-x_{i-1}}{2}$ for $i = 1, \ldots, N-1$, so that the following lemma is a direct consequence of a result established in [23, (2.44), (2.49)].

LEMMA 4.1. *There exist two positive constants c_0, c_1 independent of N such that, for any $N \geq 2$, it holds that*

$$c_0 \mathbf{v}^T M^{(1)}_{\mathbb{Q}_{1,NI}} \mathbf{v} \leq \mathbf{v}^T M^{(1)}_{GNI} \mathbf{v} \leq c_1 \mathbf{v}^T M^{(1)}_{\mathbb{Q}_{1,NI}} \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{N-1}. \tag{4.5}$$

Numerical results show that $c_1/c_0 \leq 1.00245$.

LEMMA 4.2. *For any $N \geq 2$*

$$\frac{1}{3} \mathbf{v}^T M^{(1)}_{\mathbb{Q}_{1,NI}} \mathbf{v} \leq \mathbf{v}^T M^{(1)}_{\mathbb{Q}_1} \mathbf{v} \leq \mathbf{v}^T M^{(1)}_{\mathbb{Q}_{1,NI}} \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{N-1}. \tag{4.6}$$

Proof. Let $v_h \in V_h^0$. For any $j = 0, \ldots, N-1$, it holds that

$$\int_{x_j}^{x_{j+1}} v_h^2(x) \mathrm{d}x = \frac{x_{j+1} - x_j}{3} \left[v_h(x_j)^2 + v_h(x_j) v_h(x_{j+1}) + v_h(x_{j+1})^2 \right].$$

Thanks to the Young inequality we have

$$-\frac{1}{2} \left(v_h(x_j)^2 + v_h(x_{j+1})^2 \right) \leq v_h(x_j) v_h(x_{j+1}) \leq \frac{1}{2} \left(v_h(x_j)^2 + v_h(x_{j+1})^2 \right),$$

and by summing on j we have

$$\frac{1}{3} I_T \leq \int_{-1}^{1} v_h^2(x) \mathrm{d}x \leq I_T,$$

where $I_T = \sum_{j=0}^{N-1} \frac{x_{j+1}-x_j}{2} \left(v_h^2(x_j) + v_h^2(x_{j+1}) \right)$ is the approximation of $\int_{-1}^{1} v_h^2 dx$ by the trapezoidal rule. Therefore, if $v_h \in V_h^0$ is the piecewise linear function that interpolates the $(N-1)$-uple $\mathbf{v}$ at the interior LGL nodes, the thesis follows. ∎

Thanks to both Lemmas 4.1 and 4.2, the following lemma is easily proved.

LEMMA 4.3. *For any $N \geq 2$*

$$c_0 \mathbf{v}^T M^{(1)}_{\mathbb{Q}_1} \mathbf{v} \leq \mathbf{v}^T M^{(1)}_{GNI} \mathbf{v} \leq 3 c_1 \mathbf{v}^T M^{(1)}_{\mathbb{Q}_1} \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{N-1}, \tag{4.7}$$

where c_0 and c_1 are the constants introduced in Lemma 4.1.

Now we need a result for stiffness matrices $K^{(1)}_{GNI}$ and $K^{(1)}_{\mathbb{Q}_1}$. To this aim we recall the following property, stated in both [6] and [25]: there exists a $c_2 > 1$ independent of N such that

$$\|v_N'\|^2_{L^2(-1,1)} \leq \|v_h'\|^2_{L^2(-1,1)} \leq c_2 \|v_N'\|^2_{L^2(-1,1)} \tag{4.8}$$

for any $v_N \in \mathbb{Q}_N^0(-1,1)$, with $v_h \in V_h^0$ being its piecewise linear interpolation.

The following lemma is an immediate consequence of (4.8).

LEMMA 4.4. *For any $N \geq 2$*

$$\frac{1}{c_2} \mathbf{v}^T K^{(1)}_{\mathbb{Q}_1} \mathbf{v} \leq \mathbf{v}^T K^{(1)}_{GNI} \mathbf{v} \leq \mathbf{v}^T K^{(1)}_{\mathbb{Q}_1} \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{N-1}, \tag{4.9}$$

where c_2 is the constant introduced in (4.8).

Numerical results shown in the first column of Table 4.1 give $c_2 \leq \pi^2/4 < 2.5$.

TABLE 4.1

1D case. Iterative condition numbers of some of the preconditioned matrices defined in Table 3.1 *for* $d = 1$. *Note that* $K^{(1)}_{\mathbb{Q}_1} = K^{(1)}_{\mathbb{P}_1} = K^{(1)}_{\mathbb{Q}_{1,NI}}$ *and* $M^{(1)}_{\mathbb{Q}_1} = M^{(1)}_{\mathbb{P}_1}$, *so that* $P^{w}_{\mathbb{Q}_1} = P^{w}_{\mathbb{P}_1} = P^{w}_{\mathbb{Q}_{1,NI}}$, $P^{s}_{\mathbb{Q}_1} = P^{s}_{\mathbb{P}_1}$, *and* $P^{ss,rt}_{\mathbb{Q}_1} = P^{ss,rt}_{\mathbb{P}_1}$.

N	$P^{w}_{\mathbb{Q}_1}$	$P^{s}_{\mathbb{Q}_1}$	$P^{s}_{\mathbb{Q}_{1,NI}}$	$P^{ss,rt}_{\mathbb{Q}_1}$	$P^{ss,rt}_{\mathbb{Q}_{1,NI}}$
16	2.18516	1.35975	2.18512	1.60205	2.18512
32	2.32011	1.38172	2.32010	1.59526	2.32010
48	2.36773	1.40196	2.36772	1.59491	2.36772
64	2.39207	1.41180	2.39207	1.59483	2.39207
80	2.40686	1.41813	2.40686	1.59479	2.40686
96	2.41680	1.42170	2.41680	1.59477	2.41680
112	2.42393	1.42507	2.42393	1.59476	2.42393
128	2.42930	1.42703	2.42930	1.59475	2.42930

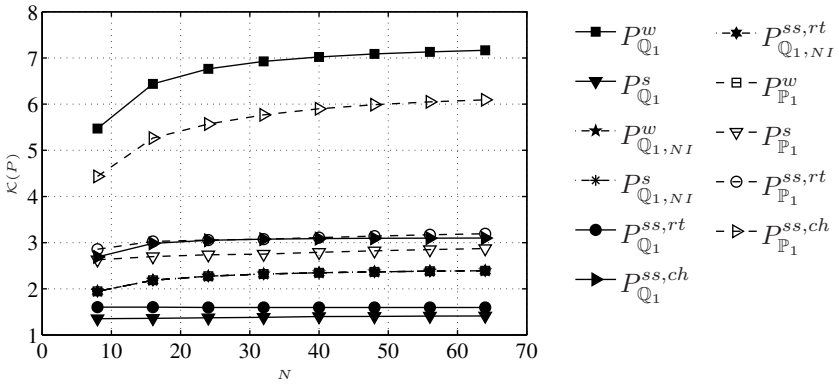

FIGURE 4.1. *2D case. Iterative condition numbers of the preconditioned matrices* (3.3)–(3.13). *Curves relative to* $P^{w}_{\mathbb{Q}_{1,NI}}$, $P^{s}_{\mathbb{Q}_{1,NI}}$, *and* $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$ *are very close to each other. Those relative to* $P^{w}_{\mathbb{Q}_{1,NI}}$ *and* $P^{w}_{\mathbb{P}_1}$ *coincide. For non s.p.d. matrices,* $\mathcal{K}$ *has been replaced by* $\mathcal{K}^*$. *The triangles of the* $\mathbb{P}_1$ *mesh are all oriented in the same way as in Figure* 2.1(b).

We are now able to state the following result, whose proof is a consequence of the previous lemmas and the property stated in (4.4).

THEOREM 4.5. *For any* $N \geq 2$

$$\mathcal{K}(P^{w}_{\mathbb{Q}_1}) = \mathcal{K}(K^{-1}_{\mathbb{Q}_1} K_{GNI}) \leq c_2 \left(\frac{3c_1}{c_0}\right)^{d-1}, \qquad d = 1, 2, 3, \tag{4.10}$$

$$\mathcal{K}(P^{w}_{\mathbb{Q}_{1,NI}}) = \mathcal{K}(K^{-1}_{\mathbb{Q}_{1,NI}} K_{GNI}) \leq c_2 \left(\frac{c_1}{c_0}\right)^{d-1}, \qquad d = 1, 2, 3, \tag{4.11}$$

where c_0 *and* c_1 *are the constants introduced in Lemma* 4.1, *while* c_2 *is the constant introduced in* (4.8).

Remark 2. Both estimates (4.10) and (4.11) are corroborated by the numerical results shown in Table 4.1 and in Figures 4.1 and 4.2. Estimates (4.10) and (4.11)

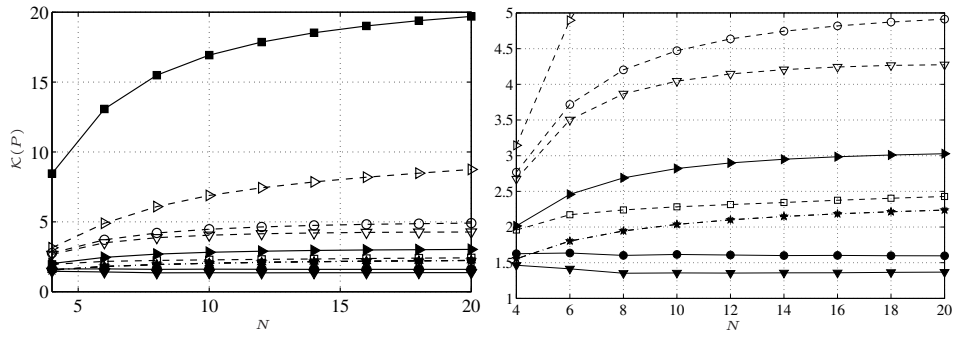

FIGURE 4.2. *3D case. Iterative condition numbers of the preconditioned matrices* (3.3)–(3.13). *The right picture is a zoom of the left one. Curves relative to* $P^{w}_{\mathbb{Q}_{1,NI}}$, $P^{s}_{\mathbb{Q}_{1,NI}}$, *and* $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$ *are very close to each other. The 6-tetrahedra mesh has been considered for those preconditioners based on* $\mathbb{P}_1$ *approximation. The symbols used in these pictures follow the legend of Figure* 4.1.

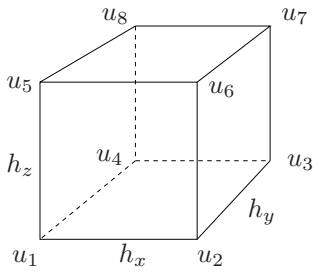

FIGURE 4.3. *Numbering of vertices in the reference hexahedron R.*

predict that the weak preconditioned matrix $P^{w}_{\mathbb{Q}_{1,NI}}$ based on the $\mathbb{Q}_{1,NI}$ approach is more efficient than that based on $\mathbb{Q}_1$ finite elements in terms of preconditioned conjugate gradient (PCG) iterations. Moreover, recalling that $c_1/c_0 \simeq 1$ as mentioned above, we expect $\mathcal{K}(P^{w}_{\mathbb{Q}_{1,NI}})$ to be basically independent of the dimension d (for values of d of practical interest), as opposed to $\mathcal{K}(P^{w}_{\mathbb{Q}_1})$. This is confirmed by the numerical results of both Figures 5.1 and 5.8 below.

Finally, let us analyze the condition number of $P^{w}_{\mathbb{P}_1}$. The following result will be useful.

LEMMA 4.6. *Let* $d = 3$, *and let each hexahedron R be split into six tetrahedra as in Figure* 2.2(right). *Then, for any* $N \geq 2$

$$\frac{3}{4}\mathbf{v}^T K_{\mathbb{P}_1}\mathbf{v} \leq \mathbf{v}^T K_{\mathbb{Q}_{1,NI}}\mathbf{v} \leq \frac{3}{2}\mathbf{v}^T K_{\mathbb{P}_1}\mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^{(N-1)^3}. \tag{4.12}$$

Proof. First, we observe that

$$\mathbf{v}^T K_{\mathbb{P}_1}\mathbf{v} = \|\nabla w_h\|^2_{L^2(\Omega)}, \qquad \mathbf{v}^T K_{\mathbb{Q}_{1,NI}}\mathbf{v} = \|\nabla v_h\|^2_{\Omega,T},$$

where $\|\cdot\|_{\Omega,T}$ denotes the approximation of the $L^2(\Omega)$-norm obtained by using the (tensorial) trapezoidal rule in each hexahedron R. Exploiting the additive property of the (squared) norms, it is enough to establish the analogue of (4.12) in each element R, i.e.,

$$\frac{3}{4}\mathbf{u}^T K^R_{\mathbb{P}_1}\mathbf{u} \leq \mathbf{u}^T K^R_{\mathbb{Q}_{1,NI}}\mathbf{u} \leq \frac{3}{2}\mathbf{u}^T K^R_{\mathbb{P}_1}\mathbf{u},$$

where $\mathbf{u} \in \mathbb{R}^8$ is the vector collecting the values of $\mathbf{v}$ associated with the eight vertices of R ordered as in Figure 4.3, whereas $K^R_{\mathbb{P}_1}$ and $K^R_{\mathbb{Q}_{1,NI}}$ are the local stiffness matrices. A lengthy but straightforward computation yields

$$\begin{aligned}
\mathbf{u}^T K^R_{\mathbb{P}_1}\mathbf{u} &= \|\nabla w_h\|^2_{L^2(R)} \\
&= \frac{h_y h_z}{6h_x}\left[2((u_2-u_1)^2+(u_7-u_8)^2)+(u_6-u_5)^2+(u_3-u_4)^2\right] \\
&\quad + \frac{h_x h_z}{6h_y}\left[2((u_8-u_5)^2+(u_3-u_2)^2)+(u_4-u_1)^2+(u_7-u_6)^2\right] \\
&\quad + \frac{h_x h_y}{6h_z}\left[2((u_8-u_4)^2+(u_6-u_5)^2)+(u_5-u_1)^2+(u_7-u_3)^2\right]
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{u}^T K^R_{\mathbb{Q}_{1,NI}}\mathbf{u} &= \|\nabla v_h\|^2_{R,T} \\
&= \frac{h_y h_z}{4h_x}\left[(u_2-u_1)^2+(u_7-u_8)^2+(u_6-u_5)^2+(u_3-u_4)^2\right] \\
&\quad + \frac{h_x h_z}{4h_y}\left[(u_8-u_5)^2+(u_3-u_2)^2+(u_4-u_1)^2+(u_7-u_6)^2\right] \\
&\quad + \frac{h_x h_y}{4h_z}\left[(u_8-u_4)^2+(u_6-u_5)^2+(u_5-u_1)^2+(u_7-u_3)^2\right].
\end{aligned}$$

Then the result follows from a repeated application of the inequalities $A^2 + B^2 \le 2A^2 + B^2 \le 2(A^2 + B^2)$. ∎

THEOREM 4.7. *For any $N \ge 2$,*

$$\mathcal{K}(P^w_{\mathbb{P}_1}) \le \sigma_d c_2 \left(\frac{c_1}{c_0}\right)^{d-1}, \qquad d = 1, 2, 3, \tag{4.13}$$

where $\sigma_1 = \sigma_2 = 1$, $\sigma_3 = 2$, c_0 and c_1 are the constants introduced in Lemma 4.1, *while c_2 is the constant introduced in* (4.8).

Proof. When $d = 1$, $K_{\mathbb{P}_1} = K^{(1)}_{\mathbb{P}_1} = K^{(1)}_{\mathbb{Q}_1}$; hence the result follows from Lemma 4.4.

When $d = 2$, $K_{\mathbb{P}_1} = K_{\mathbb{Q}_{1,NI}}$ holds (see Remark 1) so that, thanks to Theorem 4.5, we have

$$\mathcal{K}(P^w_{\mathbb{P}_1}) \le c_2 \frac{c_1}{c_0}. \tag{4.14}$$

When $d = 3$, Theorem 4.5 and Lemma 4.6 ensure that $\mathcal{K}(P^w_{\mathbb{P}_1})$ is bounded independently of N also for the 3D geometry; precisely, we have

$$\mathcal{K}(P^w_{\mathbb{P}_1}) \le \mathcal{K}(K^{-1}_{\mathbb{P}_1} K_{\mathbb{Q}_{1,NI}})\mathcal{K}(K^{-1}_{\mathbb{Q}_{1,NI}} K_{GNI}) \le 2c_2 \left(\frac{c_1}{c_0}\right)^2. \tag{4.15}$$ ∎

Note that the bound (4.15) is not sharp, as shown by the numerical results of Figure 4.2.

4.2. Strong matrices. We now consider the strong matrices $P^s_{\mathbb{Q}_1}$, $P^s_{\mathbb{Q}_{1,NI}}$, and $P^s_{\mathbb{P}_1}$. They are no longer similar to s.p.d. matrices; nevertheless, numerical evidence indicates that $P^s_{\mathbb{Q}_{1,NI}}$ has real eigenvalues, while $P^s_{\mathbb{Q}_1}$ and $P^s_{\mathbb{P}_1}$ have complex eigenvalues with imaginary parts hardly larger than one-tenth of the corresponding moduli.

For a matrix with this type of eigenstructure, the parameter

$$\mathcal{K}^* = \mathcal{K}^*(A) = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|} \simeq \mathcal{K}(A_S), \tag{4.16}$$

where A_S denotes the symmetric part of A, is an effective surrogate for $\mathcal{K}(A)$ as an indicator of the convergence properties of gradient-like methods. (In what follows, we will not usually comment on our use of this surrogate for $\mathcal{K}$ for those matrices for which the surrogate is more appropriate; however, the relevant figure labels and captions will reflect the use of the surrogate in those cases.)

THEOREM 4.8. *There exist two positive constants C_1 and C_2 independent of both N and $d(=1,2,3)$ such that*

$$\mathcal{K}^*(P^s_{\mathbb{Q}_1}) \le C_1, \qquad \mathcal{K}^*(P^s_{\mathbb{Q}_{1,NI}}) \le C_2. \tag{4.17}$$

Proof. Let us consider the system $P^s_{\mathbb{Q}_1}\tilde{\mathbf{u}} = K^{-1}_{\mathbb{Q}_1} M_{\mathbb{Q}_1}\tilde{\mathbf{f}}$, where we have $P^s_{\mathbb{Q}_1} = K^{-1}_{\mathbb{Q}_1} M_{\mathbb{Q}_1} M^{-1}_{GNI} K_{GNI}$. We begin to analyze the case $d = 1$. The eigenvalues $\lambda_i(P^s_{\mathbb{Q}_1})$ belong to the set

$$\begin{aligned} A_1 &= \left\{ z = \frac{\mathbf{u}^* K^{(1)}_{GNI}\mathbf{u}}{\mathbf{u}^* M^{(1)}_{GNI}(M^{(1)}_{\mathbb{Q}_1})^{-1} K^{(1)}_{\mathbb{Q}_1}\mathbf{u}} \quad \forall \mathbf{u} \in \mathbb{C}^n \right\} \\ &= \left\{ z = \frac{\mathbf{u}^* K^{(1)}_{GNI}\mathbf{u}}{\mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u}} \quad \forall \mathbf{u} \in \mathbb{C}^n, \mathbf{v} = (M^{(1)}_{\mathbb{Q}_1})^{-1} M^{(1)}_{GNI}\mathbf{u} \right\}, \end{aligned} \tag{4.18}$$

where $n = (N-1)$ is the dimension of 1D matrices. In order to estimate $\inf_{z\in A_1}|z|$ and $\sup_{z\in A_1}|z|$, we take into account the bound (4.9) and the following results proved in [24, Thm. 3.1 and Lem. 3.4, 3.5]: there exist positive constants $c_i,\ i = 3,\dots,7$, independent of N, such that

$$\begin{aligned} &c_3 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v} \le \mathbf{u}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u} \le c_4 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v}, \\ &c_5 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v} \le Re(\mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u}) \le c_6 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v}, \qquad |\mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u}| \le c_7 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v} \end{aligned} \tag{4.19}$$

for any $\mathbf{u} \in \mathbb{C}^n$ and $\mathbf{v} = (M^{(1)}_{\mathbb{Q}_1})^{-1} M^{(1)}_{GNI}\mathbf{u}$. By (4.9) and (4.19) we have

$$\frac{c_3 c_5}{c_7^2 c_2} \le \frac{\frac{c_5}{c_2}\mathbf{u}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u}}{c_7^2 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v}} \le Re z \le |z| \le \frac{\mathbf{u}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{u}}{c_5 \mathbf{v}^* K^{(1)}_{\mathbb{Q}_1}\mathbf{v}} \le \frac{c_4}{c_5} \qquad \text{for all } z \in A_1, \tag{4.20}$$

and then

$$\mathcal{K}^*(P^s_{\mathbb{Q}_1}) \le \frac{c_2 c_4 c_7^2}{c_3 c_5^2} \qquad \text{for } d = 1 \text{ and for all } N \ge 2. \tag{4.21}$$

Let us consider now the case $d = 2$. By recalling definitions (4.1) and (4.3) and by writing $M_{\mathbb{Q}_1} = M^{(1)}_{\mathbb{Q}_1} \otimes M^{(1)}_{\mathbb{Q}_1}$ and $M_{GNI} = M^{(1)}_{GNI} \otimes M^{(1)}_{GNI}$, the eigenvalues of $P^s_{\mathbb{Q}_1}$ belong to the set

$$\begin{aligned} A_2 &= \left\{ z = \frac{\mathbf{u}^* K_{GNI}\mathbf{u}}{\mathbf{u}^* M_{GNI} M^{-1}_{\mathbb{Q}_1} K_{\mathbb{Q}_1}\mathbf{u}} \quad \forall \mathbf{u} \in \mathbb{C}^{n^2} \right\} \\ &= \left\{ z = \frac{\mathbf{v}^*(D\otimes B + B\otimes D)\mathbf{v}}{\mathbf{v}^*(C\otimes B + B\otimes C)\mathbf{v}} \quad \forall \mathbf{v} \in \mathbb{C}^{n^2} \right\}, \end{aligned} \tag{4.22}$$

where $B = M^{(1)}_{\mathbb{Q}_1}(M^{(1)}_{GNI})^{-1}M^{(1)}_{\mathbb{Q}_1}$, $C = K^{(1)}_{\mathbb{Q}_1}(M^{(1)}_{GNI})^{-1}M^{(1)}_{\mathbb{Q}_1}$, and

$$D = M^{(1)}_{\mathbb{Q}_1}(M^{(1)}_{GNI})^{-1}K^{(1)}_{GNI}(M^{(1)}_{GNI})^{-1}M^{(1)}_{\mathbb{Q}_1}.$$

By setting

$$E = M^{(1)}_{\mathbb{Q}_1}(M^{(1)}_{GNI})^{-1}K^{(1)}_{\mathbb{Q}_1}(M^{(1)}_{GNI})^{-1}M^{(1)}_{\mathbb{Q}_1},$$

estimates (4.19) read also

$$\begin{aligned} &c_3\mathbf{v}^*K^{(1)}_{\mathbb{Q}_1}\mathbf{v} \le \mathbf{v}^*E\mathbf{v} \le c_4\mathbf{v}^*K^{(1)}_{\mathbb{Q}_1}\mathbf{v},\\ &c_5\mathbf{v}^*K^{(1)}_{\mathbb{Q}_1}\mathbf{v} \le Re(\mathbf{v}^*C\mathbf{v}) \le c_6\mathbf{v}^*K^{(1)}_{\mathbb{Q}_1}\mathbf{v}, \quad |\mathbf{v}^*C\mathbf{v}| \le c_7\mathbf{v}^*K^{(1)}_{\mathbb{Q}_1}\mathbf{v}\\ &\text{for all } \mathbf{v}\in\mathbb{C}^n. \end{aligned} \tag{4.23}$$

From (4.9), (4.4), and $(4.23)_1$, the numerator of any $z \in A_2$ satisfies the bounds

$$\begin{aligned} &\frac{c_3}{c_2}\mathbf{v}^*(K^{(1)}_{\mathbb{Q}_1}\otimes B + B\otimes K^{(1)}_{\mathbb{Q}_1})\mathbf{v} \le \frac{1}{c_2}\mathbf{v}^*(E\otimes B + B\otimes E)\mathbf{v}\\ &\quad\le \mathbf{v}^*(D\otimes B + B\otimes D)\mathbf{v} \le \mathbf{v}^*(E\otimes B + B\otimes E)\mathbf{v} \le c_4\mathbf{v}^*(K^{(1)}_{\mathbb{Q}_1}\otimes B + B\otimes K^{(1)}_{\mathbb{Q}_1})\mathbf{v} \end{aligned}$$

for any $\mathbf{v}\in\mathbb{C}^{n^2}$. About the denominator, we observe that if A, B, and C are square matrices of size n, with A and B s.p.d., and if there exist positive constants α_i such that $\alpha_1\mathbf{v}^*A\mathbf{v} \le Re(\mathbf{v}^*C\mathbf{v}) \le \alpha_2\mathbf{v}^*A\mathbf{v}$ and $|\mathbf{v}^*C\mathbf{v}| \le \alpha_3\mathbf{v}^*A\mathbf{v}$, then

$$\begin{aligned} &\alpha_1\mathbf{v}^*(A\otimes B)\mathbf{v} \le Re(\mathbf{v}^*(C\otimes B)\mathbf{v}) \le \alpha_2\mathbf{v}^*(A\otimes B)\mathbf{v} \ \text{ for all } \mathbf{v}\in\mathbb{C}^{n^2},\\ &|\mathbf{v}^*(C\otimes B)\mathbf{v}| \le \alpha_3\mathbf{v}^*(A\otimes B)\mathbf{v} \qquad\qquad \text{for all } \mathbf{v}\in\mathbb{C}^{n^2}. \end{aligned} \tag{4.24}$$

The previous bounds may be proved by exploiting the fact that the eigenvectors of B form a basis for the space $\mathbb{C}^n$. Therefore, by (4.24) it follows that

$$\begin{aligned} c_5\mathbf{v}^*(K^{(1)}_{\mathbb{Q}_1}\otimes B + B\otimes K^{(1)}_{\mathbb{Q}_1})\mathbf{v} &\le Re(\mathbf{v}^*(C\otimes B + B\otimes C)\mathbf{v})\\ &\le c_6\mathbf{v}^*(K^{(1)}_{\mathbb{Q}_1}\otimes B + B\otimes K^{(1)}_{\mathbb{Q}_1})\mathbf{v},\\ |\mathbf{v}^*(C\otimes B + B\otimes C)\mathbf{v}| &\le c_7\mathbf{v}^*(K^{(1)}_{\mathbb{Q}_1}\otimes B + B\otimes K^{(1)}_{\mathbb{Q}_1})\mathbf{v}, \end{aligned}$$

and it holds that

$$\frac{c_3c_5}{c_7^2c_2} \le Rez \le |z| \le \frac{c_4}{c_5} \qquad \text{for all } z\in A_2, \tag{4.25}$$

so that (4.21) is true also for $d = 2$. The extension to the case $d = 3$ can be carried out by the same technique.

If we consider now the matrix $P^s_{\mathbb{Q}_{1,NI}}$, we can follow the same steps explained above, thanks to formulas (3.37a) and (3.37b) in [23], which are the analogues of the second and third estimates in (4.19). Note that a bound like the first one in (4.19) immediately follows from Lemma 4.1 and the fact that both M_{GNI} and $M_{\mathbb{Q}_{1,NI}}$ are diagonal matrices. In particular, the constants c_3 and c_4 in $(4.19)_1$ are replaced now by $1/c_1^2$ and $1/c_0^2$, respectively, where c_0 and c_1 are introduced in Lemma 4.1. ∎

Remark 3. About the matrix $P^s_{\mathbb{P}_1}$, we recall that it coincides with $P^s_{\mathbb{Q}_1}$ when $d = 1$. On the other hand, for $d = 2, 3$, both $K_{\mathbb{P}_1}$ and $M_{\mathbb{P}_1}$ do not feature a tensorial structure, so that we can no longer exploit the same arguments used for $\mathbb{Q}_1$ finite elements. Numerical results shown in Figures 4.1–4.2 highlight that $\mathcal{K}^*(P^s_{\mathbb{P}_1}) < C$, with C independent of N, also for $d = 2, 3$, but now C slightly grows with d.

4.3. Numerical results. In Table 4.1 we report the iterative condition numbers, for $d = 1$, of some preconditioned matrices $P = H^{-1}L$ defined in Table 3.1, while in Figures 4.1 and 4.2 we report the iterative condition numbers of all the preconditioned matrices $P = H^{-1}L$ given in Table 3.1 for $d = 2$ and $d = 3$, respectively. We specify that numerical results in Figure 4.1 (resp., in Figure 4.2) for $P^{w}_{\mathbb{P}_1}$, $P^{s}_{\mathbb{P}_1}$, $P^{ss,rt}_{\mathbb{P}_1}$, and $P^{ss,ch}_{\mathbb{P}_1}$ refer to an oriented mesh such as that shown in Figure 2.1(b) (resp., in Figure 2.2(right)).

If $d = 2$, the stiffness matrix $K_{\mathbb{P}_1}$ is invariant with respect to the orientation of the triangles; however, this property does not hold true for the mass matrix $M_{\mathbb{P}_1}$. Consequently, the iterative condition number of the strong preconditioned matrices depends on the mesh, even though it remains bounded independently of N. In Table 4.2 (specifically, in the left column of each column block of the table) we show the iterative condition number $\mathcal{K}^*(P^{s}_{\mathbb{P}_1})$ for three different meshes of triangles induced by LGL nodes. The best performance is achieved when the mesh has all triangles oriented in the same way, while the worst one is obtained when the mesh has alternating triangles. This phenomenon can be ascribed to the presence of Lagrange basis functions with support of different size in the nonuniformly oriented cases.

The iterative condition number behaves in a similar manner also when $d = 3$ and when each hexahedron induced by the LGL mesh is split into five instead of six tetrahedra (see Figure 2.2). We have observed that the 6-tetrahedra mesh induces the same effects as the oriented 2D mesh does, while the 5-tetrahedra mesh induces the same effects as an alternating 2D mesh.

For any $d = 1, 2, 3$, all the condition numbers of both weak and strong matrices are uniformly bounded with respect to N. The smallest one is obtained for $P^{s}_{\mathbb{Q}_1}$, for any $d = 1, 2, 3$. The condition number of $P^{w}_{\mathbb{Q}_1}$ is significantly larger than the others, for $d \geq 2$, and it noticeably depends on the space dimension d, according to the theoretical results presented in the previous section (see Theorem 4.5 and the subsequent Remark 2).

Concerning the symmetrized-strong matrices, numerical results (see Figure 4.1) show that the iterative condition number of $P^{ss,rt}_{\mathbb{Q}_1}$, $P^{ss,ch}_{\mathbb{Q}_1}$, $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$ is bounded independently of N. In contrast, when simplicial $\mathbb{P}_1$ finite elements are used, two situations

TABLE 4.2

2D case. Iterative condition number $\mathcal{K}^*$ of the preconditioned matrices $P^{s}_{\mathbb{P}_1}, \ldots, P^{ss,ch}_{\mathbb{P}_1}$ associated with problem (2.1)*. Oriented mesh, alternating mesh, and random mesh are considered. The stiffness matrix is invariant with respect to mesh orientation; then $\mathcal{K}(P^{w}_{\mathbb{P}_1})$ is the same for all considered meshes.*

	Oriented mesh			Alternating mesh			Random mesh		
N	$P^{s}_{\mathbb{P}_1}$	$P^{ss,rt}_{\mathbb{P}_1}$	$P^{ss,ch}_{\mathbb{P}_1}$	$P^{s}_{\mathbb{P}_1}$	$P^{ss,rt}_{\mathbb{P}_1}$	$P^{ss,ch}_{\mathbb{P}_1}$	$P^{s}_{\mathbb{P}_1}$	$P^{ss,rt}_{\mathbb{P}_1}$	$P^{ss,ch}_{\mathbb{P}_1}$
8	2.630	2.857	4.434	3.802	15.693	13.441	3.052	5.731	6.282
16	2.698	3.027	5.265	3.943	108.238	73.647	3.199	11.311	11.468
24	2.737	3.056	5.569	4.034	439.980	275.359	3.321	25.420	23.230
32	2.751	3.075	5.769	4.106	1277.766	771.505	3.380	79.004	69.486
40	2.790	3.109	5.900	4.160	2995.229	1776.044	3.330	90.404	94.588
48	2.823	3.142	5.988	4.199	6072.810	3564.090	3.386	199.216	190.145
56	2.850	3.170	6.050	4.226	11097.759	6471.917	3.422	302.417	304.913
64	2.872	3.193	6.094	4.247	18764.135	10896.959	3.447	482.504	501.969

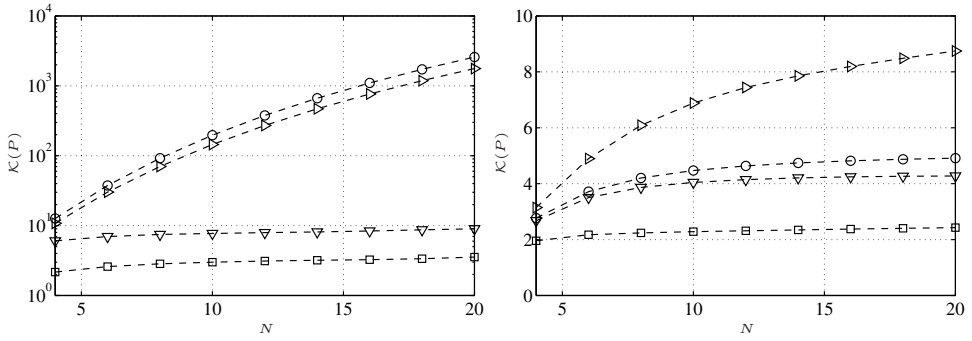

FIGURE 4.4. *3D case. Iterative condition numbers of the preconditioned matrices* $P^w_{\mathbb{P}_1}$, $P^s_{\mathbb{P}_1}$, $P^{ss,rt}_{\mathbb{P}_1}$, *and* $P^{ss,ch}_{\mathbb{P}_1}$ *for both* 5*-tetrahedra (left) and* 6*-tetrahedra (right) meshes. The symbols used in these pictures follow the legend of Figure* 4.1.

are faced. When $d = 2$, Table 4.2 shows that if all triangles are oriented in the same way, the iterative condition number of both $P^{ss,rt}_{\mathbb{P}_1}$ and $P^{ss,ch}_{\mathbb{P}_1}$ is uniformly bounded with respect to N; in contrast, if the rectangles are split either randomly or with an alternating orientation, then both $\mathcal{K}(P^{ss,rt}_{\mathbb{P}_1})$ and $\mathcal{K}(P^{ss,ch}_{\mathbb{P}_1})$ grow like N^p, for some $p \in [3, 4]$. The latter growth has also been observed for the 5-tetrahedra mesh when $d = 3$, as we can see in Figure 4.4. For this reason, in what follows we will only consider the 6-tetrahedra mesh for the 3D case.

4.4. Neumann boundary conditions, nonconstant viscosity, and skewness of the domain. Let us confine our analysis to the case $d = 2$. When Neumann boundary conditions are imposed on two consecutive edges of the boundary $\partial\Omega$ and homogeneous Dirichlet boundary conditions are assigned on the remaining edges, the iterative condition numbers of the strong matrices $P^s_{\mathbb{Q}_1}$, $P^s_{\mathbb{Q}_{1,NI}}$, and $P^s_{\mathbb{P}_1}$ and of the weak matrices $P^w_{\mathbb{Q}_1}$, $P^w_{\mathbb{Q}_{1,NI}}$, and $P^w_{\mathbb{P}_1}$ are again independent of the polynomial degree N. In contrast, the iterative condition numbers associated with all the symmetrized-strong matrices now depend on N, precisely as N^3. These results are shown in Figure 4.5 (top left).

Now we ask whether the preconditioners introduced in the previous sections are still efficient when a variable viscosity shows up in (2.1). In particular we consider the problem

$$-\nabla \cdot (\nu \nabla u) = f \quad \text{in } \Omega = (-1, 1)^d\,, \qquad u = 0 \quad \text{on } \partial\Omega, \tag{4.26}$$

where the viscosity $\nu = \nu(\mathbf{x})$ satisfies $\nu \in L^\infty(\Omega)$ and $\nu(\mathbf{x}) \geq \nu_0$ for all $\mathbf{x} \in \Omega$, for some constant $\nu_0 > 0$.

In Figure 4.5 we report the iterative condition numbers for both weak and strong matrices relative to three different choices of the viscosity function. As in the constant-coefficient case, the condition numbers $\mathcal{K}(P)$ are always uniformly bounded with respect to N, although the bounds are slightly larger. The specific dependence on N becomes more apparent as the variation of ν in the domain increases.

A situation similar to that of variable coefficients occurs when we consider constant coefficients on a deformed domain and we map the deformed demain onto the reference domain by an affine invertible transformation. More precisely, we obtain an elliptic problem with mixed second-order derivatives and variable coefficients depending on the skewness of the domain. The condition numbers $\mathcal{K}(P)$ are again bounded

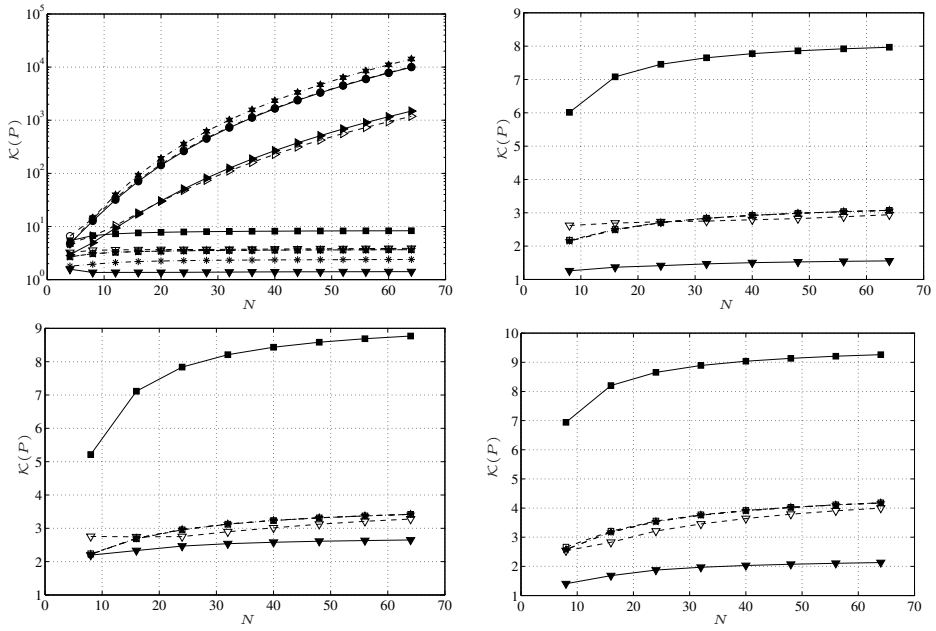

FIGURE 4.5. *2D case. At top left, iterative condition numbers of the preconditioned matrices* (3.3)–(3.13). *Homogeneous Dirichlet boundary conditions are imposed on two consecutive sides of* $\partial\Omega$, *and Neumann boundary conditions on the two others. At top right and at bottom, iterative condition numbers of some of the preconditioned matrices* (3.3)–(3.13) *referred to in problem* (4.26). *Only weak and strong matrices have been considered. At top right,* $\nu(x,y) = 1 + x^2y^2$; *at bottom left,* $\nu(x,y) = 3 + sin(\pi x) + cos(\pi y)$; *at bottom right,* $\nu(x,y) = 1 + 3x^2y^2$. *In all cases, the condition numbers relative to* $P^w_{\mathbb{Q}_{1,NI}}$, $P^s_{\mathbb{Q}_{1,NI}}$, *and* $P^w_{\mathbb{P}_1}$ *are very close to each other. The symbols used here follow the legend of Figure* 4.1.

independently of N, even if now they are larger and depend on the deformation of the domain. For instance, we have considered a quadrilateral with angles of 45, 117, 90, and 108 degrees, and we have computed the condition number for both weak and strong preconditioners, with the exception of skew-symmetric ones. The worst upper bound has been obtained for $P^s_{\mathbb{P}_1}$ and is about 12, while the best one is about 6 for $P^s_{\mathbb{Q}_1}$. Both preconditioners $P^w_{\mathbb{Q}_{1,NI}}$ and $P^s_{\mathbb{Q}_{1,NI}}$ are less sensitive to the skewing; for them we have $\mathcal{K}(P) \leq 6.2$.

5. Performances of the preconditioners and the solution strategies. The preconditioned systems associated with either (2.4) or (2.7) can be solved by a PCG algorithm, whereas those associated with (2.6) need a nonsymmetric solver. For the latter, the preconditioned Bi-CGStab (PBi-CGStab) method [29] is our choice; however, the preconditioned GMRES method would represent a viable alternative.

Each PCG (resp., PBi-CGStab) iteration applied to (3.1) requires the solution of one (resp., two) systems of the form

$$H\mathbf{z} = \mathbf{r}, \tag{5.1}$$

where $\mathbf{r} = \mathbf{r}^{(k)} = \tilde{\mathbf{f}} - L\tilde{\mathbf{u}}^{(k)}$ is the residual of (2.8) (corresponding to the kth iteration) and H is the preconditioning matrix. Taking into account the definitions of H given in Table 3.1, it is readily seen that solving system (5.1) turns into solving an equivalent system whose matrix is one of the finite-element stiffness matrices $K_{\mathbb{Q}_1}$, $K_{\mathbb{Q}_{1,NI}}$, $K_{\mathbb{P}_1}$,

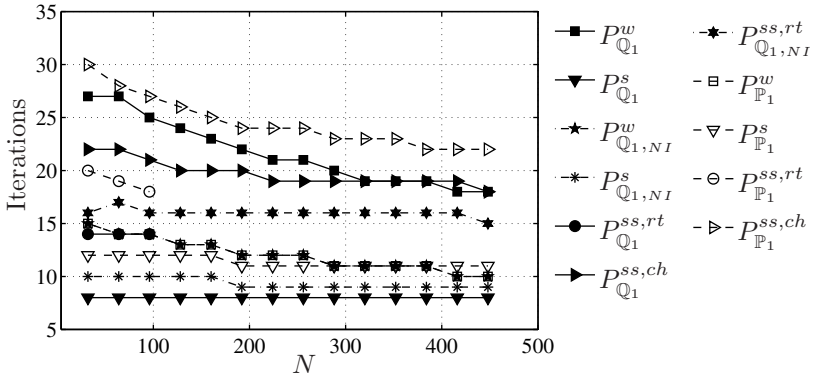

FIGURE 5.1. *2D case. Number of PCG and PBi-CGStab iterations to solve problem* (2.1) *with* $f \equiv 1$ *and* $\mathbf{u}^{(0)} = \mathbf{0}$*, for the preconditioners given by* (3.3)–(3.13).

while the mass matrices $M_{\mathbb{Q}_1}$, $M_{\mathbb{Q}_{1,NI}}$, $M_{\mathbb{P}_1}$ (or either their square roots or Cholesky factors when the symmetrized-strong preconditioners are invoked) are involved only in matrix-vector products. Therefore, we are invariably left with the task of solving a system with an s.p.d. banded matrix K_{FE}.

From now on we prefer to treat the 2D and 3D cases separately.

5.1. 2D case. The first element in comparing preconditioners is the number of iterations required by either PCG or PBi-CGstab to converge or, more precisely, to meet the stopping criterion $||\mathbf{r}^{(k)}||_{H^{-1}}/||\mathbf{r}^{(0)}||_{H^{-1}} < 10^{-14}$. The number of iterations will affect the iterative process cost. Figure 5.1 reports the number of iterations needed to solve problem (2.1) with $f \equiv 1$, with the initial guess $\mathbf{u}^{(0)} = \mathbf{0}$, by either PCG or PBi-CGStab algorithm. It is interesting to note that this number decreases for increasing N for almost all preconditioners. This is because of the good choice of the initial guess $\mathbf{u}^{(0)} = \mathbf{0}$, which is compatible with the Dirichlet boundary conditions imposed in problem (2.1). In fact, by inspecting the coefficients of the Legendre expansion of the initial residual $\mathbf{r}^{(0)}$, one observes that they decay very quickly for increasing wave-numbers; furthermore, the larger modal components are associated with the lower wave-numbers. In contrast, if the initial guess $\mathbf{u}^{(0)}$ for CG-iterations does not satisfy the Dirichlet boundary conditions for u, which is the case if, e.g., $\mathbf{u}^{(0)} = \mathbf{1}$, then the larger modal components of $\mathbf{r}^{(0)}$ are associated with both low and high wave-numbers. In this case the number of PCG iterations needed to converge to a given tolerance remains nearly constant for increasing N. A behavior similar to that reported in Figure 5.1 is observed if f is such that the solution u of problem (2.1) is infinitely smooth.

The previous results indicate that the smallest number of iterations is given by $P^{s}_{\mathbb{Q}_1}$. However, the number of iterations is just one aspect of the evaluation of the performance of an iterative method, the cost of the preprocessing and of the single iteration being equally important factors of analysis. We will report below numerical results concerning CPU-times for several iterative solution schemes applied to the preconditioned systems (3.3)–(3.13).

Three different algebraic solvers have been considered for solving system (5.1):

1. the Lapack Cholesky factorization for banded matrices and subsequent forward/backward substitutions (CHOL, for short);

2. the PCG with relaxed incomplete Cholesky factorization with zero fill-in [2, 10] (RICCG(0), for short);
3. the HSL_MA57 [17, 15, 16, 27] multifrontal algorithm with *nested dissection* ordering produced by MeTiS [19] (ND-MF, for short).

The relaxed incomplete Cholesky (RIC) factorization is an interpolated version (by a relaxation parameter $\omega \in [0,1]$) of the incomplete Cholesky (IC) factorization with the modified incomplete Cholesky factorization fulfilling the row-sum equivalence condition (RS-MIC, for short). When $\omega = 0$, RIC corresponds to IC, while when $\omega = 1$, it corresponds to RS-MIC. Note that the matrix K_{FE} is an M-matrix, which is a sufficient condition for the existence of the RIC factorization with $\omega < 1$. In contrast, existence of RS-MIC factorization is not guaranteed for general M-matrices and is highly dependent on the ordering of the unknowns [10]. About the choice of the relaxation parameter, van der Vorst [28] suggested using $\omega = 0.95$ in practice. Our experiments show that, for 2D test cases, the choice $\omega = 0.95$ performs better than $\omega = 0$ on both stiffness matrices $K_{\mathbb{Q}_1}$ and $K_{\mathbb{Q}_{1,NI}} = K_{\mathbb{P}_1}$. On the other hand, for 3D problems, the choice $\omega = 0$ guarantees more robustness to RICCG(0) than $\omega > 0$, when applied to the stiffness matrix $K_{\mathbb{Q}_1}$. From now on, the abbreviation RICCG(0) will imply the choice $\omega = 0.95$ for $K_{\mathbb{Q}_{1,NI}}$ $(d = 2, 3)$, $K_{\mathbb{P}_1}$ $(d = 2, 3)$, and $K_{\mathbb{Q}_1}$ $(d = 2)$, and $\omega = 0$ for $K_{\mathbb{Q}_1}$ $(d = 3)$.

Concerning the multifrontal algorithm, the indicated choice has been made after a comparison with both HSL_MA57 with *approximate minimum degree* (AMD) ordering and UMFPACK [12, 11] with AMD ordering, for its better performance in the examined situations, particularly in the 3D case.

In order to implement each of these algebraic solvers, a preprocessing step is needed, which includes the assembly of both K_{FE} and M_{FE}, the factorization of K_{FE}, and, if required, the computation of either the square root of M_{FE}^{-1} or its Cholesky factor. Additionally, at each PCG iteration one matrix-vector product plus one solution of the linear system (5.1) on the preconditioner are required, whereas at each PBiCGStab iteration two matrix-vector products plus two solutions of the linear system (5.1) are required.

We have measured CPU-times in seconds on an HP xw4400 Workstation with an Intel Core$^{\text{TM}}$ 2 Duo processor E6700, 2.67GHz. Both solvers CHOL and ND-MF have been applied to all of the preconditioners (3.3)–(3.13), whereas RICCG(0) has been applied only to (3.3)–(3.6) and (3.9)–(3.11).

The total CPU-times are shown in Figures 5.2–5.4. We note that, for any choice of the algebraic solver among CHOL, RICCG(0), and ND-MF, the fastest solution was obtained from the preconditioned matrix $P^w_{\mathbb{Q}_{1,NI}}$, which coincides with $P^w_{\mathbb{P}_1}$ (although the CPU-times are slightly different, due to different assembly operations). Remarkably, the corresponding preconditioning matrices produce the best results without even involving the mass matrix.

The slowest solutions are those obtained using the preconditioned matrices $P^{ss,rt}_{\mathbb{Q}_1}$ and $P^{ss,rt}_{\mathbb{P}_1}$. In such cases the (soon prohibitive) major cost is due to the evaluation of the square roots of $M^{-1}_{\mathbb{Q}_1}$ and $M^{-1}_{\mathbb{P}_1}$, respectively; due to their inefficiency, we have reported CPU-time for these two preconditioners only for $N \leq 96$, and we will not consider them in the subsequent analysis. The use of the Cholesky factor of M^{-1}_{FE} inside the symmetrized-strong forms $P^{ss,ch}_{\mathbb{Q}_1}$ and $P^{ss,ch}_{\mathbb{P}_1}$ is not as expensive as the computation of the square root of the matrix; thus, the total CPU-times are comparable to those of the weak and strong forms of the preconditioned system. Nevertheless, the latter choices require a wider memory storage. Note, however, that $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$, which

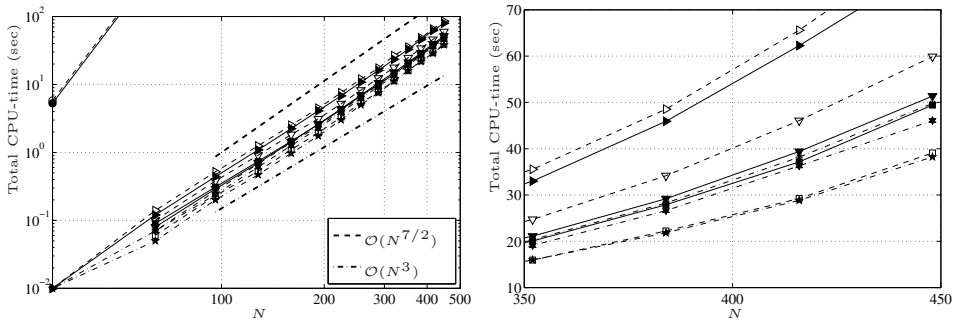

FIGURE 5.2. 2*D case. Total CPU-time (sec.) for solving* (3.1) *for the different preconditioners defined in* (3.3)–(3.13). *CHOL is used for solving system* (5.1). $f \equiv 1$ *is chosen in* (2.1), *and* $\mathbf{u}^{(0)} = \mathbf{0}$ *as initial guess for either PCG or PBi-CGStab. The right picture is a zoom of the left one. The symbols used in these pictures follow the legend of Figure* 5.1.

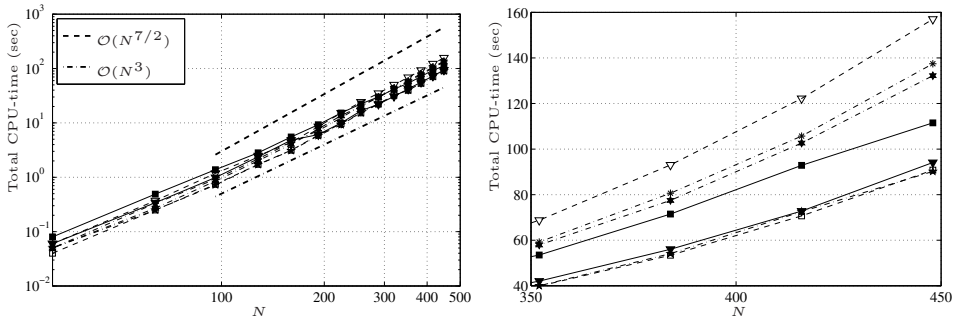

FIGURE 5.3. 2*D case. Total CPU-time (sec.) for solving* (3.1) *for the different preconditioners defined in* (3.3)–(3.6) *and* (3.9)–(3.11); *RICCG*(0) *is used for solving system* (5.1). $f \equiv 1$ *is chosen in* (2.1), *and* $\mathbf{u}^{(0)} = \mathbf{0}$ *as initial guess for either PCG or PBi-CGStab. The right picture is a zoom of the left one. The symbols used in these pictures follow the legend of Figure* 5.1.

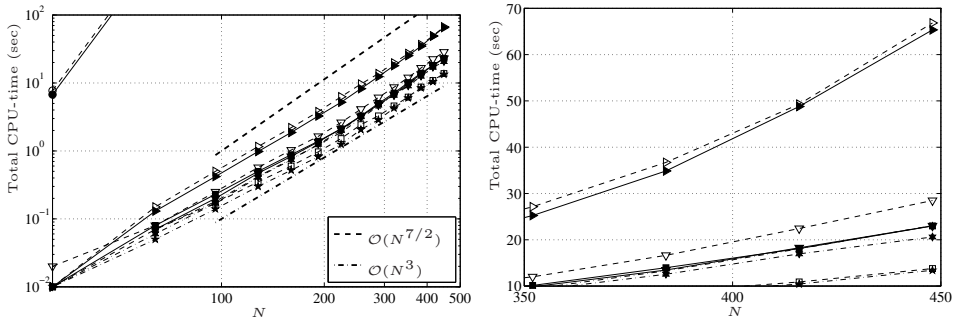

FIGURE 5.4. 2*D case. Total CPU-time (sec.) for solving* (3.1) *for the different preconditioners defined in* (3.3)–(3.13). *ND-MF is used for solving system* (5.1). $f \equiv 1$ *is chosen in* (2.1), *and* $\mathbf{u}^{(0)} = \mathbf{0}$ *as initial guess for either PCG or PBi-CGStab. The right picture is a zoom of the left one. The symbols used in these pictures follow the legend of Figure* 5.1.

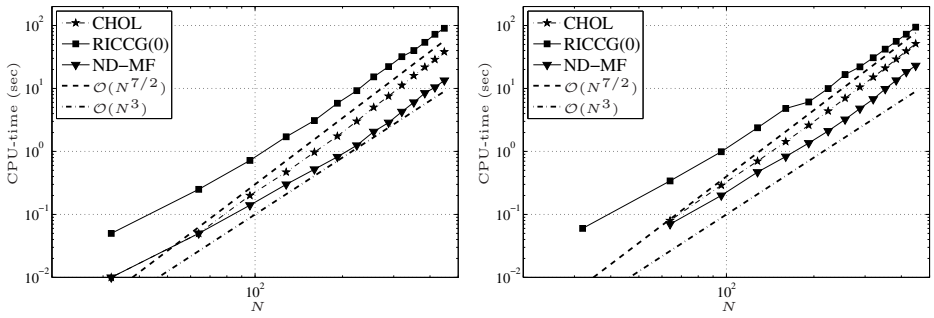

FIGURE 5.5. *2D case. Total CPU-time (sec.) for solving* (3.1) *for both choices* (3.5) *(left) and* (3.4) *(right). Either CHOL, RICCG*(0) *with* $\omega = 0.95$*, or ND-MF are used for solving system* (5.1)*.* $f \equiv 1$ *is chosen in* (2.1)*, while* $\mathbf{u}^{(0)} = \mathbf{0}$ *is the initial guess for both PCG and PBi-CGStab.*

makes use of the diagonal mass matrix $M_{\mathbb{Q}_{1,NI}}$ inside the symmetrized-strong form, produces good results too.

Among the strong matrices, the best performing one (if we disregard the runs invoking RICCG(0)) is $P^s_{\mathbb{Q}_{1,NI}}$, thanks to the diagonal structure of the mass matrix $M_{\mathbb{Q}_{1,NI}}$, in spite of the fact that $P^s_{\mathbb{Q}_1}$ has the minimum iterative condition number.

Remark 4. Within the option of using an incomplete factorization, say C, of the finite-element stiffness matrix K_{FE}, one could think of taking a "shortcut," i.e., applying the inverse of such factorization directly to the spectral stiffness matrix K_{GNI}. However, this strategy would be inefficient, since it would require the evaluation of many spectral residuals, a significant burden in terms of computational cost. Indeed, the iterative condition number of the preconditioned matrix $C^{-1}K_{GNI}$ is reported to satisfy $\mathcal{K}(C^{-1}K_{GNI}) = \mathcal{O}(N^2)$, implying an $\mathcal{O}(N)$ number of CG iterations needed to solve system (2.4) in this way. One should then perform an equivalent number of evaluations of the spectral residual, as opposed to the $\mathcal{O}(1)$ number for all the strategies investigated in what follows.

Our next aim is to compare the efficiency of the three algebraic solvers (CHOL, RICCG(0), and ND-MF) when applied to solving the system (5.1). To this aim we limit our analysis to only two preconditioners, $P^w_{\mathbb{Q}_{1,NI}}$ and $P^s_{\mathbb{Q}_1}$, which are among the most efficient ones in terms of computational time; the former does not need the mass matrix, while the latter does require such a matrix within an extra matrix-vector product. In view of the fact that $P^w_{\mathbb{P}_1} = P^w_{\mathbb{Q}_{1,NI}}$ (in two dimensions), the same analysis done for $P^w_{\mathbb{Q}_{1,NI}}$ can be extended to $P^w_{\mathbb{P}_1}$.

In Figure 5.5 we directly compare the total CPU-times for $P^w_{\mathbb{Q}_{1,NI}}$ (left) and $P^s_{\mathbb{Q}_1}$ (right) measured when we use CHOL, RICCG(0), and ND-MF. It is not surprising that the most efficient algebraic method is the multifrontal one. In general, we can observe that the total CPU-time required by RICCG(0) is about six to ten times that required by ND-MF, for any choice of the preconditioners defined in (3.3)–(3.6), (3.9)–(3.13); furthermore, the total CPU-time measured by using CHOL grows more quickly than these two as N tends to infinity. CPU-times exhibit a growth proportional to N^3 when either RICCG(0) or ND-MF is used, and to $N^{7/2}$ when CHOL is used. A comparison between the plots on the left-hand side and on the right-hand side of Figure 5.5 indicates that the weak-$\mathbb{Q}_{1,NI}$ preconditioner invariably outperforms the strong-$\mathbb{Q}_1$ one by a factor of about 2.

It is worthwhile analyzing in more detail the cost of both the preprocessing step, say C_{PRE}, and the iterative process, say C_{LOOP}, in terms of elementary floating point

operations versus either the polynomial degree N or the global number of degrees of freedom $n = (N-1)^2$. We confine ourselves to the cases of weak and strong preconditioners (thus we do not address the symmetrized-strong versions). C_{LOOP} is given by the product of the number of iterations it and the cost of a single iteration C_{ITER}. We thus have for the total cost C_{TOT}:

$$C_{\mathrm{TOT}} = C_{\mathrm{PRE}} + C_{\mathrm{LOOP}} = C_{\mathrm{PRE}} + it * C_{\mathrm{ITER}}.$$

On the other hand, we have

$$C_{\mathrm{PRE}} = C_{\mathrm{ASS}} + C_{\mathrm{FACT}}, \qquad C_{\mathrm{ITER}} = C_{\mathrm{RHS}} + C_{\mathrm{SOL}},$$

where we have used the following notation:

C_{ASS}:	cost of assembling the matrices needed by the FEM preconditioner,
C_{FACT}:	cost of factorizing the stiffness matrix K_{FE},
C_{RHS}:	cost of forming the right-hand side of the finite-element system (5.1),
C_{SOL}:	cost of solving the finite-element system (5.1).

(We deliberately ignore the cost of assembling the stiffness matrix K_{GNI}, which scales as $N^{d+1} = N^3$, since it is common to all solution strategies and because it could be avoided by exploiting the tensorial structure of the matrix in the computation of the spectral residual.)

The cost of each stage can be related to the number of required floating point operations, for which we now provide theoretical estimates. Extra time is spent during memory access operations, whose analysis, being strictly related to the knowledge of the specific hardware in use, will be omitted.

The assembly time C_{ASS} depends on the assembly of the stiffness matrix K_{FE}, which requires $\mathcal{O}(N^d)$ flops since the number of nonzero entries per row is bounded independently of N. In addition, for the strong matrices, one has to assemble the mass matrix M_{FE}, in $\mathcal{O}(N^d)$ flops, and form the spectral matrix $L = M_{GNI}^{-1} K_{GNI}$ in $\mathcal{O}(N^{d+1})$ flops.

The factorization time C_{FACT} depends on the iterative solver. CHOL requires $\mathcal{O}(np^2)$ flops, where $n \simeq N^d$, while $p \simeq N^{d-1}$ is the bandwidth of K_{FE}; thus $C_{\mathrm{FACT}} = \mathcal{O}(N^{3d-2})$. RICCG(0) requires $\mathcal{O}(1)$ flops per row, yielding $C_{\mathrm{FACT}} = \mathcal{O}(N^d)$. Finally, the factorization time of ND-MF is given by $C_{\mathrm{FACT}} = \mathcal{O}(N^3)$ for $d = 2$ or $C_{\mathrm{FACT}} = \mathcal{O}(N^6)$ for $d = 3$ [1].

The cost C_{RHS} depends on the cost of forming the spectral residual $\mathbf{r^{(k)}} = \mathbf{f} - L\mathbf{u}^{(k)}$, which requires $\mathcal{O}(N^{d+1})$ flops. In addition, for the strong preconditioners, we have to account for a matrix-vector product by M_{FE}, which costs $\mathcal{O}(N^d)$ flops. In any case, this stage requires $\mathcal{O}(N^{d+1})$ flops.

At last, let us investigate C_{SOL}. CHOL costs $2n(2p+1)$ flops, where n and p have the same meaning as above, yielding $C_{\mathrm{SOL}} = \mathcal{O}(N^{2d-1})$. Concerning RICCG(0), each inner iteration requires $\mathcal{O}(N^d)$ flops; on the other hand, the condition number of the preconditioned matrix $C^{-1}K_{FE}$, where C^{-1} stands for the RICCG(0)-preconditioning, satisfies (experimentally) $\mathcal{K}(C^{-1}K_{FE}) = \mathcal{O}(N)$ for small to moderate values of N and $\mathcal{K}(C^{-1}K_{FE}) = \mathcal{O}(N^2)$ in the asymptotic regime, as opposed to $\mathcal{K}(K_{FE}) = \mathcal{O}(N^3) = \mathcal{K}(K_{GNI})$. Recalling the convergence rate of the CG method, which is proportional to the inverse of $\sqrt{\mathcal{K}(C^{-1}K_{FE})}$, we obtain that the number of RICCG(0)-iterations needed to reduce the residual to machine accuracy scales with $\sqrt{N}$ in the first case and with N in the second one; therefore in the asymptotic regime,

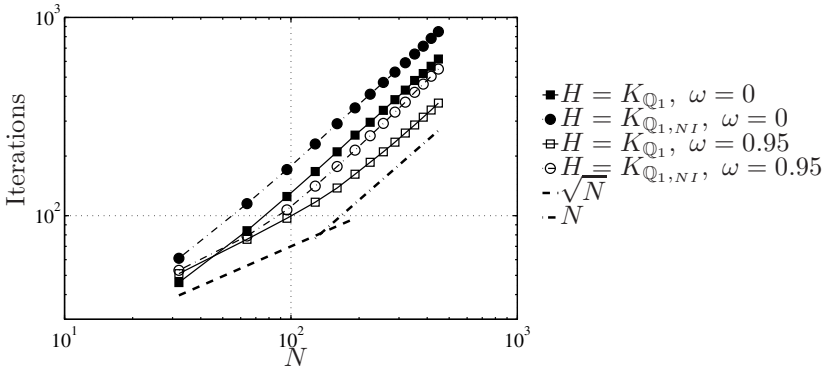

FIGURE 5.6. *2D case. Number of RICCG*(0) *iterations to solve the system* (5.1) *with either* $H = K_{\mathbb{Q}_1}$ *or* $H = K_{\mathbb{Q}_{1,NI}}$. ω *is the relaxation parameter of RICCG*(0).

TABLE 5.1

2D case. Theoretical (upper rows of each case) and measured (lower rows) costs of the various stages of the iterative solution scheme, versus N.

		C_{PRE} $= C_{\text{ASS}} + C_{\text{FACT}}$	C_{LOOP} $= it * (C_{\text{RHS}} + C_{\text{SOL}})$	C_{TOT}
CHOL	Weak	$c_A N^2 + c_F N^4$	$c_R N^3 + c_S N^3$	$c_T N^4$
		$3 \cdot 10^{-9} N^{3.72}$	$5 \cdot 10^{-8} N^{3.16}$	$2 \cdot 10^{-8} N^{3.46}$
	Strong	$c_A N^3 + c_F N^4$	$c_R N^3 + c_S N^3$	$c_T N^4$
		$8 \cdot 10^{-9} N^{3.59}$	$7 \cdot 10^{-8} N^{3.19}$	$5 \cdot 10^{-8} N^{3.38}$
RICCG(0)	Weak	$c_A N^2 + c_F N^2$	$c_R N^3 + c_S N^3$	$c_T N^3$
		$3 \cdot 10^{-6} N^{1.93}$	$2 \cdot 10^{-7} N^{3.25}$	$2 \cdot 10^{-7} N^{3.24}$
	Strong	$c_A N^3 + c_F N^2$	$c_R N^3 + c_S N^3$	$c_T N^3$
		$2 \cdot 10^{-7} N^{2.67}$	$3 \cdot 10^{-7} N^{3.21}$	$3 \cdot 10^{-7} N^{3.19}$
ND-MF	Weak	$c_A N^2 + c_F N^3$	$c_R N^3 + c_S N^2 \log N$	$c_T N^3$
		$1 \cdot 10^{-6} N^{2.43}$	$5 \cdot 10^{-8} N^{3.11}$	$2 \cdot 10^{-7} N^{2.90}$
	Strong	$c_A N^3 + c_F N^3$	$c_R N^3 + c_S N^3 \log N$	$c_T N^3$
		$8 \cdot 10^{-7} N^{2.59}$	$3 \cdot 10^{-8} N^{3.29}$	$2 \cdot 10^{-7} N^{2.99}$

$C_{\text{SOL}} = \mathcal{O}(N^{d+1})$ for RICCG(0). This is confirmed by Figure 5.6. The same figure also displays a comparison between the choices $\omega = 0.95$ and $\omega = 0$ inside RICCG(0); we can deduce that the former is 1/3 less expensive than the latter. Concerning ND-MF, the cost of backward/forward solution is proportional to the fill-in and therefore given by $C_{\text{SOL}} = \mathcal{O}(N^2 \log N)$ for $d = 2$, or $C_{\text{SOL}} = \mathcal{O}(N^4)$ for $d = 3$ [1].

Finally, as seen above, for both preconditioners here considered, the number of iterations needed to solve (5.1) to machine accuracy is $it = \mathcal{O}(1)$, precisely between 7 and 15, and actually it is a decreasing function of N for certain initial guesses $\mathbf{u}^{(0)}$.

The individual theoretical bounds presented so far can be combined to produce bounds for the intermediate costs C_{PRE} and C_{LOOP} and for the total cost C_{TOT}. Table 5.1 collects all these results for the strategies under investigation. (The terms *weak* and *strong* refer to the $P^w_{\mathbb{Q}_{1,NI}}$ and $P^s_{\mathbb{Q}_1}$ preconditioned matrices, respectively.) The cost of each stage is described as cN^α, where α is drawn from the previous discussion;

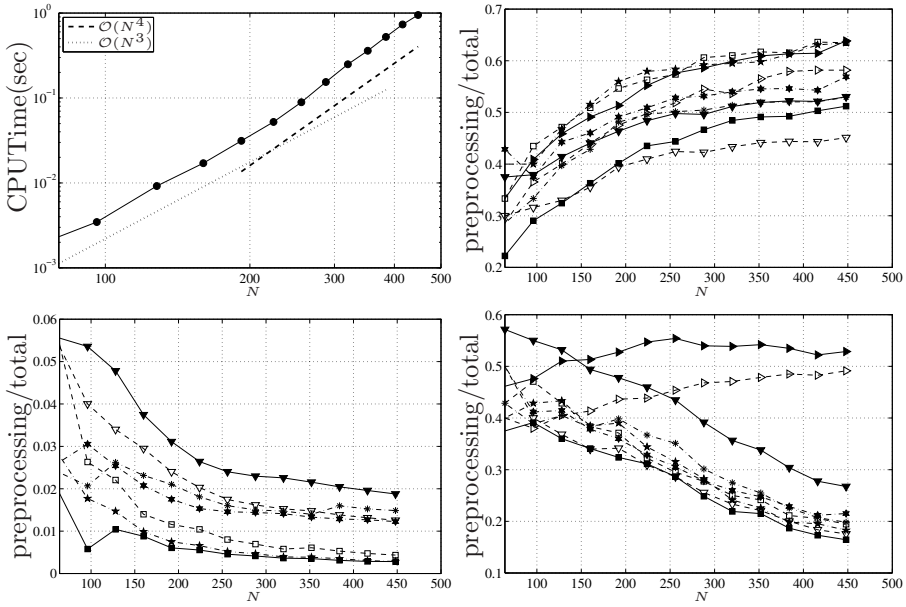

FIGURE 5.7. *2D case. At top left, CPU-times needed for evaluating one spectral residual,* $\mathbf{r}^{(k)} = \tilde{\mathbf{f}} - L\tilde{\mathbf{u}}^{(k)}$. *At top right and at bottom, the part of the total CPU-time required by the preprocessing step when either CHOL (top right), RICCG*(0) *(bottom left), or ND-MF (bottom right) is used to solve* (5.1). *CPU-times shown in Figures* 5.2 *and* 5.3 *have been considered in computing the percentages shown here. The symbols used in these pictures follow the legend of Figure* 5.1.

obviously, this is the leading term in the expansion of each cost with respect to N; i.e., it represents the expected asymptotic behavior as $N \to \infty$. The theoretical results are compared, in the same Table 5.1, to the actual results of our experiments, which are reported below them. For each stage, we have computed a least-squares fit of a law like $\overline{c}N^{\overline{\alpha}}$, for N in the range $[32, 448]$, of the values indicated in Figures 5.2–5.4 for the total CPU-times, as well as of the measured intermediate CPU-times of the partial steps.

The results indicate a good agreement between theory and experiments. They also confirm and provide better evidence for the ranking among the methods expressed by Figure 5.5. It is worth noticing that the measured exponent of C_{LOOP} is higher than the one predicted by the theory; this phenomenon has to be ascribed to the growth of the CPU-time needed for the spectral residual evaluation. Indeed, for N large enough, memory access costs become predominant over floating point operation costs, yielding an overall $\mathcal{O}(N^4)$ cost for this stage, as opposed to the $\mathcal{O}(N^3)$ estimate based only on consideration of flops. Figure 5.7 (top left) clearly documents this behavior.

Another useful piece of information which can be drawn from Table 5.1 concerns the ratio between preprocessing cost and total cost for the different strategies. A complementary picture is provided by Figure 5.7, where the results for all preconditioners are shown. Both theory and experiments indicate that this ratio tends to 1 for CHOL (with values between 0.4 and 0.6 in the explored range of N), whereas it tends to 0 for RICCG(0) (with values between 0.06 down to 0.01 and below). There is evidence of the decay of this ratio also for ND-MF, although less pronounced than for RICCG(0).

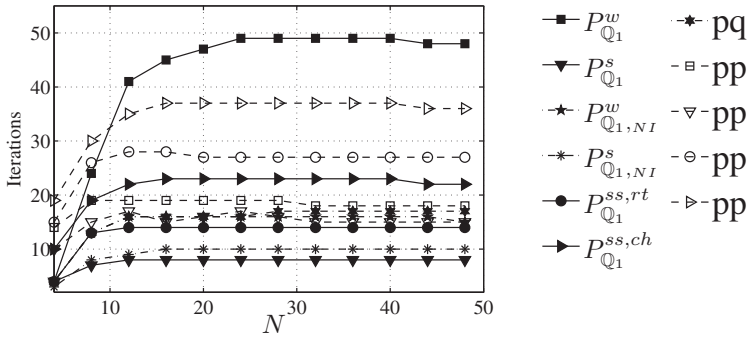

FIGURE 5.8. *3D case. Number of PCG and PBi-CGStab iterations to solve problem* (2.1) *with* $f \equiv 1$ *and* $\mathbf{u}^{(0)} = \mathbf{0}$*, for the preconditioners defined in* (3.3)–(3.13).

The conclusion of the 2D investigation is that the weak $\mathbb{Q}_{1,NI}$ preconditioning approach coupled with the multifrontal solver for the FEM system allows one to compute the solution of the spectral system (2.4) with a total cost which scales as n^β in the number $n \simeq N^2$ of degrees of freedom, β being slightly less than 3/2; this result holds in the range $2 \leq N \leq 448$ at least.

5.2. 3D case. We consider again the model problem (2.1) with $f \equiv 1$. In Figure 5.8 we report the number of iterations needed to meet the stopping criterion $||\mathbf{r}^{(k)}||_{H^{-1}}/||\mathbf{r}^{(0)}||_{H^{-1}} < 10^{-14}$ with the initial guess $\mathbf{u}^{(0)} = \mathbf{0}$, whose behavior agrees with that of the iterative condition numbers reported in Figure 4.2. In particular, the number of iterations is independent of N for all preconditioners.

The high sparsity of both mass and stiffness finite-element matrices for 3D computational domains has induced us to solve system (5.1) by either RICCG(0) or ND-MF, with the exclusion of CHOL.

As done for the 2D case, we first compare the total CPU-times needed to solve system (3.1); see Figures 5.9 and 5.10. For both cases, the fastest solution was obtained from the preconditioned matrix $P^w_{\mathbb{Q}_{1,NI}}$, although also $P^s_{\mathbb{Q}_{1,NI}}$ and $P^{ss,rt}_{\mathbb{Q}_{1,NI}}$ are very competitive. These results reflect what happens in the 2D case. In contrast, if we compare RICCG(0) and ND-MF for the best performing preconditioned matrix $P^w_{\mathbb{Q}_{1,NI}}$, we find that RICCG(0) performs better than ND-MF. In particular for $N = 48$, the total CPU-time needed to solve (5.1) with $P^w_{\mathbb{Q}_{1,NI}}$ is about 11 seconds when RICCG(0) is used, while it is about 42 seconds when ND-MF is used, reversing what happens in the 2D case.

By using the notation introduced in the previous section and by recalling the theoretical flop count of the various stages, expressed as a function of both the polynomial degree N and the geometric dimension d, we can estimate the computational cost of our preconditioning approaches also for the 3D cases. In Table 5.2 we exhibit the theoretical flop counts (upper rows) and the actual results of our experiments (lower rows) for both the weak preconditioned matrix $P^w_{\mathbb{Q}_{1,NI}}$ and the strong one $P^s_{\mathbb{Q}_1}$. The least-squares fits have been performed with N in the range $[4, 48]$. Again, a fairly good agreement between prediction and observation is obtained.

We note that if the solution of the reference differential problem falls within the case of time-dependent partial differential equations and a significant number of solve calls are made, the preprocessing cost can be ignored, and the C_{LOOP} time

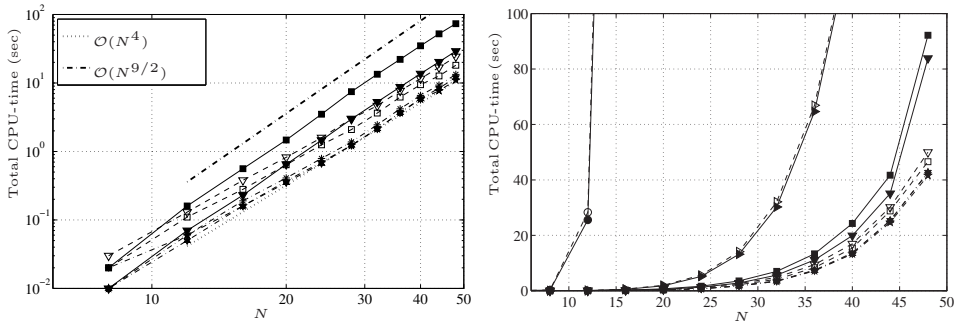

FIGURE 5.9. 3*D case. Total CPU-time (sec.) for solving* (3.1) *for the different preconditioners defined in* (3.3)–(3.6) *and* (3.9)–(3.11). *RICCG*(0) *(with* $\omega = 0.95$ *when either* $H = K_{\mathbb{Q}_1,NI}$ *or* $H = K_{\mathbb{P}_1}$, *and* $\omega = 0$ *when* $H = K_{\mathbb{Q}_1}$*) is used for solving system* (5.1). $f \equiv 1$ *is chosen in* (2.1), *and* $\mathbf{u}^{(0)} = \mathbf{0}$ *as the initial guess for either PCG or PBi-CGStab. The right picture is a zoom of the left one. The symbols used in these pictures follow the legend of Figure* 5.8.

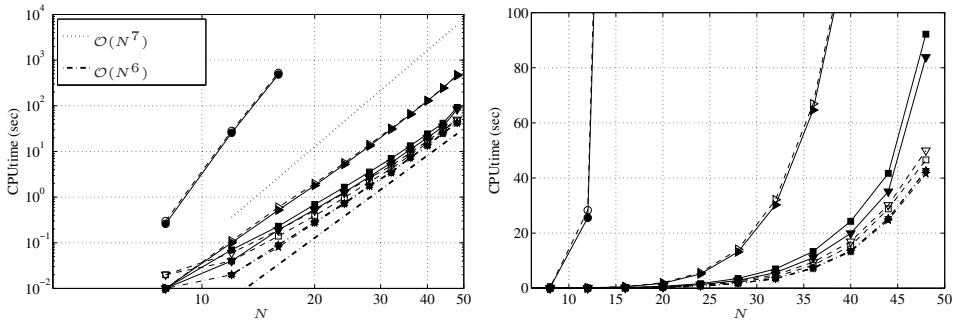

FIGURE 5.10. 3*D case. Total CPU-time (sec.) for solving* (3.1) *for the different preconditioners defined in* (3.3)–(3.13). *ND-MF is used for solving system* (5.1). $f \equiv 1$ *is chosen in* (2.1), *and* $\mathbf{u}^{(0)} = \mathbf{0}$ *as the initial guess for either PCG or PBi-CGStab. The right picture is a zoom of the left one. The symbols used in these pictures follow the legend of Figure* 5.8.

TABLE 5.2

3*D case. Theoretical (upper rows of each case) and measured (lower rows) costs of the various stages of the iterative solution scheme, versus* N.

		C_{PRE} $= C_{\mathrm{ASS}} + C_{\mathrm{FACT}}$	C_{LOOP} $= it * (C_{\mathrm{RHS}} + C_{\mathrm{SOL}})$	C_{TOT}
RICCG(0)	Weak	$c_A N^3 + c_F N^3$	$c_R N^4 + c_S N^4$	$c_T N^4$
		$5 \cdot 10^{-6} N^{3.04}$	$2 \cdot 10^{-6} N^{4.13}$	$2 \cdot 10^{-6} N^{3.95}$
	Strong	$c_A N^4 + c_F N^3$	$c_R N^4 + c_S N^4$	$c_T N^4$
		$1 \cdot 10^{-5} N^{3.08}$	$7 \cdot 10^{-7} N^{4.53}$	$1 \cdot 10^{-6} N^{4.36}$
ND-MF	Weak	$c_A N^3 + c_F N^6$	$c_R N^4 + c_S N^4$	$c_T N^6$
		$4 \cdot 10^{-9} N^{5.90}$	$2 \cdot 10^{-7} N^{4.30}$	$2 \cdot 10^{-8} N^{5.53}$
	Strong	$c_A N^4 + c_F N^6$	$c_R N^4 + c_S N^4$	$c_T N^6$
		$6 \cdot 10^{-8} N^{5.31}$	$2 \cdot 10^{-7} N^{4.42}$	$7 \cdot 10^{-8} N^{5.30}$

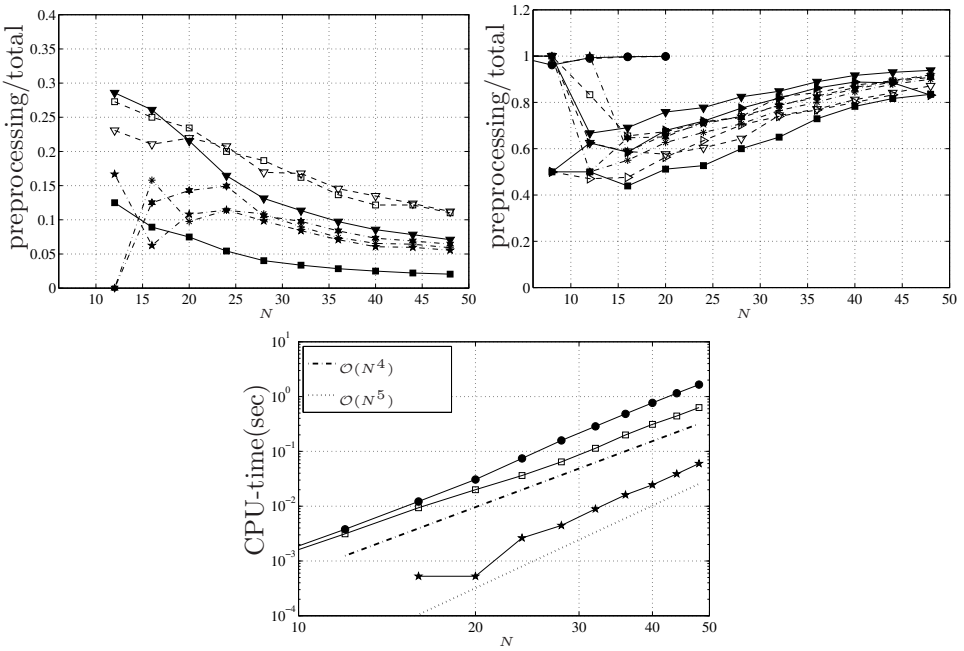

FIGURE 5.11. *3D case. At top, the part of the total CPU-time required by the preprocessing step when either the RICCG*(0) *(left) or ND-MF (right) solver is used to solve* (5.1)*. CPU-times shown in Figure* 5.9 *and* 5.10 *have been considered. The symbols used in these pictures follow the legend of Figure* 5.8*. At bottom, the CPU-times needed for evaluating one spectral residual* $\mathbf{r}^{(k)} = \tilde{\mathbf{f}} - L\tilde{\mathbf{u}}^{(k)}$ *(⋆) and for implementing the iterative RICCG*(0) *solution of* (5.1)*; the symbol • refers to the choice* $H = M_{\mathbb{Q}_1}^{-1} K_{\mathbb{Q}_1}$*, while □ refers to* $H = K_{\mathbb{Q}_{1,NI}}$.

becomes the unique term of comparison. In this scenario ND-MF and RICCG(0) have a comparable computational cost, behaving like N^4.

In Figure 5.11 the cost of the preprocessing step over the total CPU-time is shown. When ND-MF is used (see Figure 5.11(top right)), the preprocessing step increasingly dominates the total computational time. Numerical results indicate that the overall cost of the preprocessing step almost invariably takes more than 50 percent of the total solution cost, and it grows with N up to 90 percent in the range of N under consideration. About the preconditioned matrices $P_{\mathbb{Q}_1}^{ss,rt}$ and $P_{\mathbb{P}_1}^{ss,rt}$, we note that the computation of the square root of both mass matrices $M_{\mathbb{Q}_1}^{-1}$ and $M_{\mathbb{P}_1}^{-1}$ is very expensive, so that the resulting strategies are greatly inefficient. In contrast, when RICCG(0) is used (see Figure 5.11(top left)), the iterative stage is the most expensive. The preprocessing step takes at most 30 percent of the total CPU-time, and its cost decreases for increasing N. This is in agreement with the results in the first two columns of Table 5.2.

Finally, the bottom panel of Figure 5.11 shows the CPU-times needed to evaluate the spectral residual and to solve system (5.1) with RICCG(0) for both weak $P_{\mathbb{Q}_{1,NI}}^{w}$ and strong $P_{\mathbb{Q}_1}^{s}$ approaches. As for the 2D case, the measured CPU-time needed to evaluate one spectral residual grows more rapidly than the theoretical estimate (it is $\mathcal{O}(N^{d+1})$ rather than $\mathcal{O}(N^d)$), due to memory access overhead, and this sensibly affects the cost of both the iteration step and the global solution stage.

6. Conclusions. We have considered the approximation by spectral methods of the Laplace equation $-\Delta u = f$ with Dirichlet boundary conditions in $\Omega \subset \mathbb{R}^d$ with $d = 2,\ 3$. We have also addressed the case of an elliptic operator with variable coefficients, as well as the case of Neumann boundary conditions. Both strong (i.e., collocation based on Legendre–Gauss–Lobatto nodes) and weak (i.e., Galerkin with Legendre–Gauss–Lobatto numerical integration) approaches have been taken into account in building up spectral matrices. We have also considered symmetrized-strong preconditioners in order to take advantage of algebraic solvers for s.p.d. matrices. Eleven different kind of finite-element preconditioners have been considered, based on either $\mathbb{P}_1$, $\mathbb{Q}_1$, or $\mathbb{Q}_{1,NI}$ (i.e., $\mathbb{Q}_1$ with numerical integration) shape functions. Vertices of finite-element meshes coincide with the Legendre–Gauss–Lobatto quadrature nodes used for the primal spectral approximation.

The preconditioner based on the $\mathbb{Q}_1$-FEM approach for the strong form of the primal spectral approximation gives the smallest condition number. Nevertheless, if we measure preconditioner efficiency in terms of memory storage and CPU-time, the best performance is obtained for weak and strong preconditioners based on a $\mathbb{Q}_{1,NI}$-FEM approach, for both 2D and 3D geometries. The efficiency of $\mathbb{P}_1$ preconditioners depends on the kind of mesh on which they are built, or, more precisely, on grid orientation. Our analysis highlights iterative strategies for solving (2.4) or (2.6) whose overall cost scales as n^β, with β slightly less than $3/2$ (in 2D) and $4/3$ (in 3D), in the total number n of d.o.f.'s (explored up to some $\mathcal{O}(10^5)$).

Acknowledgment. We warmly thank Dr. Mario Arioli for his advice and help on the use of multifrontal direct solvers.

REFERENCES

[1] M. Arioli, *Personal communication*, STFC Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, UK, 2008.
[2] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.
[3] C. Bernardi and Y. Maday., *Approximations Spectrales de Problèmes aux Limites Elliptiques*, Springer-Verlag, Paris, 1992.
[4] S. Beuchler, *Multigrid solver for the inner problem in domain decomposition methods for P-FEM*, SIAM J. Numer. Anal., 40 (2002), pp. 928–944.
[5] S. Beuchler, R. Schneider, and C. Schwab, *Multiresolution weighted norm equivalences and applications*, Numer. Math., 98 (2004), pp. 67–97.
[6] C. Canuto, *Stabilization of spectral methods by finite element bubble functions*, Comput. Methods Appl. Mech. Engrg., 116 (1994), pp. 13–26.
[7] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods. Fundamentals in Single Domains*, Springer, Heidelberg, 2006.
[8] C. Canuto and A. Quarteroni, *Preconditioned minimal residual methods for Chebyshev spectral calculations*, J. Comput. Phys., 60 (1985), pp. 315–337.
[9] M. A. Casarin, *Quasi-optimal Schwarz methods for the conforming spectral element discretization*, SIAM J. Numer. Anal., 34 (1997), pp. 2482–2502.
[10] T. F. Chan and H. A. van der Vorst, *Approximate and incomplete factorizations*, in Parallel Numerical Algorithms (Hampton, VA, 1994), Vol. 4, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 167–202.
[11] T. A. Davis, *UMFPACK Version* 5.0.1, Technical report, University of Florida, Gainesville, FL, 2006.
[12] T. A. Davis and I. S. Duff, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Trans. Math. Software, 25 (1999), pp. 1–20.
[13] M. O. Deville and E. H. Mund, *Chebyshev pseudospectral solution of second-order elliptic equations with finite element preconditioning*, J. Comput. Phys., 60 (1985), pp. 517–533.
[14] M. O. Deville and E. H. Mund, *Finite-element preconditioning for pseudospectral solutions of elliptic problems*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 311–342.

[15] I. S. Duff, *MA57—A code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Software, 30 (2004), pp. 118–144.
[16] I. S. Duff and S. Pralet, *Strategies for scaling and pivoting for sparse symmetric indefinite problems*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 313–340.
[17] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.
[18] P. F. Fischer, *An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations*, J. Comput. Phys., 133 (1997), pp. 84–101.
[19] G. Karypis lab, *MeTiS, A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, available online at http://glaros.dtc.umn.edu/gkhome/views/metis.
[20] P. Haldenwang, G. Labrosse, S. Abboudi, and M. Deville, *An element-by-element solution algorithm for problems of structural and solid mechanics*, Comput. Methods Appl. Mech. Engrg., 36 (1983), pp. 241–254.
[21] J. M. Melenk, *On condition numbers in hp-FEM with Gauss–Lobatto-based shape functions*, J. Comput. Appl. Math., 139 (2002), pp. 21–48.
[22] S. A. Orszag, *Spectral methods for problem in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92.
[23] S. V. Parter, *Preconditioning Legendre spectral collocation methods for elliptic problems* I*: Finite difference operators*, SIAM J. Numer. Anal., 39 (2001), pp. 330–347.
[24] S. V. Parter, *Preconditioning Legendre spectral collocation methods for elliptic problems* II*: Finite element operators*, SIAM J. Numer. Anal., 39 (2001), pp. 348–362.
[25] S. V. Parter and E. E. Rothman, *Preconditioning Legendre spectral collocation approximations to elliptic problems*, SIAM J. Numer. Anal., 32 (1995), pp. 333–385.
[26] A. Quarteroni and E. Zampieri, *Finite element preconditioning for Legendre spectral collocation approximations to elliptic equations and systems*, SIAM J. Numer. Anal., 29 (1992), pp. 917–936.
[27] STFC, Numerical Analysis Group, *Hsl, A collection of FORTRAN codes for large-scale scientific computation*, available online at Science and Technology Facilities Council, http://hsl.rl.ac.uk/hsl2007.
[28] H. A. van der Vorst, *High performance preconditioning*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 1174–1185.
[29] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monogr. Appl. Comput. Math. 13, Cambridge University Press, Cambridge, UK, 2003.