On the Mortar Element Method: Generalizations and Implementation*

George Anagnostou†
Yvon Maday**
Catherine Mavriplis‡
Anthony T. Patera†

Abstract. The mortar element method is an optimal nonconforming domain decomposition method for the discretization of partial differential equations which provides for a maximum of mesh, refinement, and resolution flexibility while simultaneously preserving locality and elemental structure. In this paper, we present several enhancements to the spectral element version of the mortar method: the mortar treatment of arbitrary two-dimensional element topologies; the application of the mortar method to moving-geometry sliding-mesh problems; and the development of new data structures based on composite data objects and topology trees which allow for simple implementation of complex discretizations.

1 Introduction

Spectral element methods [1,2] are weighted residual techniques for the approximation of partial differential equations that combine the rapid convergence rate of spectral methods [3,4] with the generality of finite element techniques [5,6,7]. The spectral element discretization, coupled to fast iterative solvers [8,9,10], has proven computationally efficient on high performance serial and parallel processors [11,12]. Although the spectral element method is, by construction, applicable in complex geometries [13,14], the conforming, geometric matching required between neighboring, high-order elements leads to complications and inefficiencies in mesh generation, dynamic mesh refinement, and the treatment of moving boundaries.

Non-conforming discretizations greatly increase the flexibility of the spectral element domain decomposition. During mesh generation, this additional degree of functionality

*This work was supported by Avions Marcek Dassault-Bréguet Aviation, the ONR and DARPA under contract N00014-89-J-1610, the ONR under contract N00014-88-K-0188, and the NSF under Grant ASC-8806925.

†Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139.

**Laboratoire d'Analyse Numérique d l'Université Pierre et Marie Curie, Paris, France.

‡Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544.

reduces the complexity associated with element generation and localized mesh refinement by greatly decreasing the coupling between subdomains. In addition, such discretizations can be used to effectively resolve solution intricacies in particular subdomains of the problem without expending computational resources in regions where a small number of high-order elements, or an assembly of low-order finite elements [15], is adequate for capturing the essential features of the solution.

The mortar element method [15,16] is a nonconforming discretization based on the *explicit* construction of an optimal approximation space. The method preserves element-based locality, distinguishing it from other, more global, techniques such as functional minimization with Lagrange multiplier constraints [17], or the "global element method" [18]. The mortar method lends itself well to efficient implementation on parallel supercomputers: it allows for the classification and decoupling of the computational work into loosely coupled subdomains; it lays the foundation for sparse inter-domain communication; it preserves local structure for fast evaluation procedures.

In Section 2 of this paper we start with a brief introduction to the variational form of the partial differential equations of interest, derive the mortar method discrete equations, and describe the application of existing iterative approaches to solution of the resulting matrix systems. In Section 3 we describe a mortar-based method for the sliding-mesh treatment of moving boundaries, and present numerical verification of the accuracy of the method. Finally, in Section 4, we describe a unified approach to the implementation of the general non-conforming spectral element method, and demonstrate the significant simplifications in computational evaluation and algorithmic complexity provided by appropriate data structures.

2 Problem Formulation

Discretization. The discussion here is restricted to second-order linear partial differential equations, as this will demonstrate the essential features of the discretization; extensions to Stokes and Navier-Stokes problems are given in [16]. We consider the solution of the Poisson equation on a domain Ω of $\mathbf{R}^2$: Find $u(x,y)$ such that

$$\begin{aligned} -\nabla^2 u &= f \quad \text{in } \Omega, & (1)\\ u &= 0 \quad \text{on } \partial\Omega, & (2)\end{aligned}$$

where $\partial\Omega$ is the boundary of Ω, and f is the prescribed force. We suppose that Ω is rectangularly decomposable, that is, that there exist rectangular subdomains Ω^k, $k = 1, ..., K$ such that

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}^k, \quad \forall k, l,\ k \neq l,\ \Omega^k \cap \Omega^l = \emptyset. \tag{3}$$

The weak formulation of the problem (1,2) takes the following form: Find $u \in X = H_0^1(\Omega)$ such that

$$(\nabla u, \nabla v) = (f, v), \quad \forall v \in X. \tag{4}$$

Here $(\cdot,\cdot)$ represents the L^2 inner product over the domain Ω, and L^2 and H_0^1 are the usual Sobolev spaces [19].

The Galerkin discretization of the problem (1, 2) consists of testing the variational form (4) with respect to a family of discrete finite dimensional spaces X_h. The discrete problem becomes: Find $u_h \in X_h$ such that

$$(\nabla u_h, \nabla v_h) = (f, v_h), \quad \forall v_h \in X_h. \tag{5}$$

In the conforming case $X_h \subset H_0^1(\Omega)$; in nonconforming approximations, X_h is not a subspace of H_0^1, thereby introducing additional "consistency" errors. We now describe the spectral element form of X_h for the nonconforming mortar discretization.

To begin, we identify the K rectangular subdomains of (3) as spectral elements, with edges denoted $\Gamma^{k,l}$, $l = 1, ..., 4$,

$$\partial\Omega^k = \bigcup_{l=1}^{4} \overline{\Gamma}^{k,l} .$$

We then define the skeleton S of the "mortar" system as

$$S = \bigcup_{k=1}^{K} \partial\Omega^k . \tag{6}$$

We next introduce a non-unique set of M non-empty mortars, γ_p, as follows;

$$\gamma^p = \Gamma^{k,l} \text{ for some } k, l \text{ such that } \begin{cases} S = \bigcup_{p=1}^{M} \overline{\gamma}^p \\ \gamma^p \cap \gamma^q = \emptyset \quad p \neq q \end{cases} , \tag{7}$$

where p is an arbitrary enumeration $p = 1, ..., M$. The intersection of the closures of the γ^p defines a set of vertices, $\mathcal{V}$, composed of all non-empty elements

$$v^q = (\overline{\gamma}^m \cap \overline{\gamma}^n) \quad \text{such that} \quad v^q \notin \gamma^p \ \forall\, p = 1, ..., M \quad , \tag{8}$$

where q is an arbitrary enumeration $q = 1, ..., V$. Finally, we define the set of *virtual* vertices, $\tilde{\mathcal{V}}$, composed of all non-empty elements

$$\tilde{v}^q = \overline{\gamma}^m \cap \gamma^n \ , \tag{9}$$

for $q = 1, ..., \tilde{\mathcal{V}}$. The geometry of the nonconforming decomposition for a general $K = 4$ element case is illustrated in Figure 1. Note that the *conforming* spectral element discretization corresponds to the case where we (can) choose a mortar set in which $\Gamma^{k,l} = \gamma^p$ for some p for all k, l; the earlier *"refinement"* mortar method [16] corresponds to the case where there are no virtual vertices.

The nonconforming spectral element discretization space, X_h is then given by

$$\begin{aligned} X_h = \quad & \{ v \in L^2(\Omega),\ \forall k = 1, ..., K,\ v_{|_{\Omega^k}} \in P_N(\Omega^k) \ \text{ such that } \exists \phi \in W_h \text{ for which:} \\ & \forall q = 1, ..., V,\ \forall k = 1, ..., K, \text{ such that } v^q \text{ is a vertex of } \Omega^k,\ v_{|_{\Omega^k}}(v^q) = \phi(v^q); (10) \\ & \forall q = 1, ..., \tilde{\mathcal{V}},\ \forall k = 1, ..., K, \text{ such that } \tilde{v}^q \text{ is a vertex of } \Omega^k,\ v_{|_{\Omega^k}}(\tilde{v}^q) = \phi(\tilde{v}^q); (11) \\ \text{and} \quad & \forall l = 1, ..., 4,\ \forall k = 1, ..., K, \quad \forall \psi \in P_{N-2}(\Gamma^{k,l}), \quad \textstyle\int_{\Gamma^{k,l}} (v_{|_{\Omega^k}} - \phi)\psi ds = 0 \quad \}. (12) \end{aligned}$$

Here, $P_N(\Omega^k)$ denotes the space of all polynomials on Ω^k of degree $\leq N$ in each spatial direction, and the auxiliary mortar space W_h is defined as

$$W_h = \{ \phi \in C^0(S),\ \forall p = 1, ..., M,\ \phi_{|_{\gamma^p}} \in P_N(\gamma^p),\ \phi_{|_{\partial\Omega}} = 0 \} . \tag{13}$$

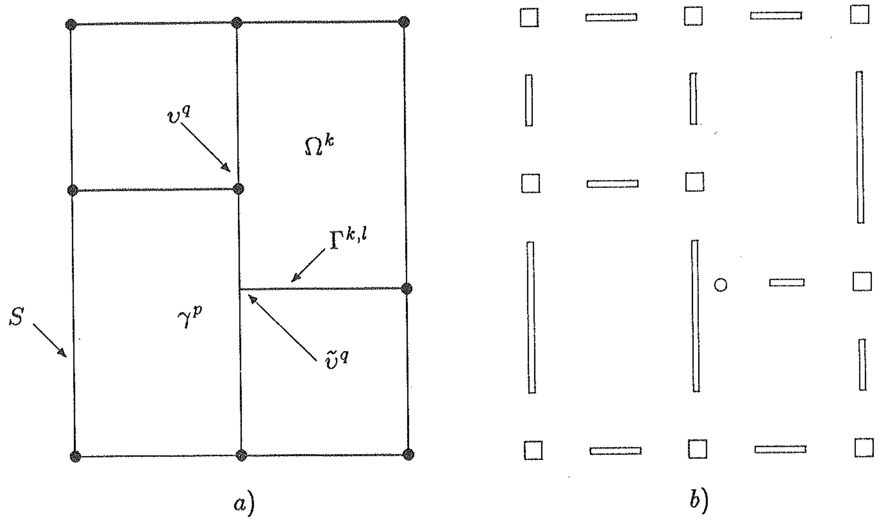

Figure 1: Nonconforming subdomain decomposition (a) and associated skeleton structure (b), showing vertices, v^q, (□), virtual vertices, $\tilde{v}^q$, (○), and mortars, γ^p (▭).

Equation (10), referred to as the vertex condition, ensures exact continuity at cross points, while Equation (12), the integral matching condition, represents an L^2 minimization of the jump in functions at internal boundaries. The properties of uniqueness of the solution and optimality of the approximation and consistency errors have been investigated in previous publications [15,16,20].

Bases and Discrete Equations. To complete the implementation of the method a choice of basis is required. Although the spaces W_h and X_h appear quite complicated, they have simple basis representations which are easily evaluated to yield an efficient domain decomposition algorithm. To begin, we write for the auxiliary space W_h,

$$\phi_{|\gamma^p} = \sum_{j=0}^{N} \phi_j^p h_j^N(\tilde{s}), \quad \forall p \in \{1, ..., M\} \ , \tag{14}$$

where $\tilde{s}$ is a mortar-local variable, defined on $[-1,1]$, and the h_j^N are the N^{th} order Gauss-Lobatto Lagrangian interpolants defined as

$$h_i \in P_N(]-1,1[), \ h_i(\xi_j) = \delta_{ij}, \forall i,j \in \{0, ..., N\}^2. \tag{15}$$

The ξ_j are the Gauss-Lobatto Legendre points [21]. The representation for $v \in X_h$ is given by

$$v_{|\Omega^k} = \sum_{i=0}^{N} \sum_{j=0}^{N} v_{ij}^k h_i^N(\hat{x}) h_j^N(\hat{y}), \quad \forall k \in \{1, ..., K\} \ , \tag{16}$$

where $\hat{x}, \hat{y}$ are element-local variables on $]-1,1[^2$. It is clear that the internal degrees-of-freedom, v_{ij}^k, $i,j \in \{1, ..., N-1\}^2$, are free, however the boundary degrees-of-freedom are constrained through (10-12).

In order to implement the matching condition (12), we require further notation. In Figure 2 we illustrate an arbitrary mortar/edge configuration, and introduce the notions

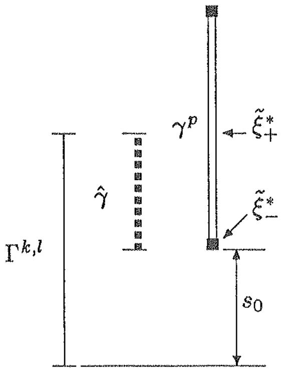

Figure 2: Illustration of a mortar offset s_0, edge $\Gamma^{k,l}$, mortar γ^p, and integration strip $\hat{\gamma}$ and intersection points $\tilde{\xi}^*_+$ and $\tilde{\xi}^*_-$ for a particular edge-mortar combination.

of mortar offset s_0, mortar γ_p of length $\mid \gamma^p \mid$, elemental edge $\Gamma^{k,l}$ of length $\mid \Gamma^{k,l} \mid$, and integration strip $\hat{\gamma} = \Gamma^{k,l} \cap \gamma^p$ of length $\mid \hat{\gamma} \mid$. We also show the intersection points $\tilde{\xi}^*_+$ and $\tilde{\xi}^*_-$ of the mortar γ^p with the top and bottom corners of the corresponding integration strip $\hat{\gamma}$, given in *mortar-local* coordinates. Note that all lengths are in absolute (x, y) coordinates. We also introduce the notation

$$r_{+_i} = \begin{cases} h_i(\tilde{\xi}^*_+) & \text{if } \mid \hat{\gamma} \mid + \mid \min(0, -s_0) \mid = \mid \Gamma^{k,l} \mid \\ 0 & \text{otherwise} \end{cases}$$

and

$$r_{-_i} = \begin{cases} h_i(\tilde{\xi}^*_-) & \text{if } s_0 \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

for some particular edge-mortar combination. In the actual implementation of r_{+_i} and r_{-_i} the two stringent equality conditions need only be met to within a pre-specified tolerance level [22], ϵ; stability ensures a no-greater-than-commensurate increase in the discretization error.

In order to express the integral matching condition (12) in matrix form (for some particular choice of $\Gamma^{k,l}$ and γ^p) we also require a basis for ψ, which we choose as

$$\psi|_{\Gamma^{k,l}} = \sum_{q=1}^{N-1} \beta_q \eta_q^{N-2}(\hat{s}) \tag{17}$$

where

$$\eta_q^{N-2}(z) = (-1)^{N-q} \frac{L'_N(z)}{\xi_q - z} \qquad z \in]-1, 1[, \quad q \in \{1, \ldots, N-1\}, \tag{18}$$

and $\hat{s} = \hat{x}$ or $\hat{y}$ for a horizontal or vertical edge $\Gamma^{k,l}$, respectively. We then perform (here exact) piecewise Gauss-Lobatto quadrature on $N+1$ points on a particular elemental edge $\Gamma^{k,l}$ and integration strip $\hat{\gamma}$, giving

$$\sum_{j=1}^{N-1} \overline{B}_{ij} v_j = \sum_{j=0}^{N} \overline{P}_{ij} \phi_j, \quad \forall i \in \{1, \ldots, N-1\} \tag{19}$$

where for the destination edge $\Gamma^{k,l}$ we have

$$\int_{\Gamma^{k,l}} v\psi \longrightarrow \overline{B}_{ij} = \frac{|\Gamma^{k,l}|}{2}(-1)^{N-i}(-L''_N(\xi_i))\rho_i\delta_{ij}, \quad \forall i,j \in \{1,...,N-1\}^2. \tag{20}$$

For $\int_{\hat{\gamma}} \phi\psi ds \longrightarrow \overline{P}_{ij}$ we have, for $s_0 \geq 0$,

$$\overline{P}_{ij} = \frac{|\hat{\gamma}|}{2}\sum_{q=0}^{N}\rho_q\eta_i^{N-2}\left(\frac{2s_0}{|\Gamma^{k,l}|} - 1 + (1+\xi_q)\frac{|\hat{\gamma}|}{|\Gamma^{k,l}|}\right) h_j\left(-1 + (1+\xi_q)\frac{|\hat{\gamma}|}{|\gamma^p|}\right)$$
$$- \frac{|\Gamma^{k,l}|}{2}\eta_i^{N-2}(-1)r_{-j}\rho_0 - \frac{|\Gamma^{k,l}|}{2}\eta_i^{N-2}(+1)r_{+j}\rho_N \tag{21}$$

whereas for $s_0 < 0$,

$$\overline{P}_{ij} = \frac{|\hat{\gamma}|}{2}\sum_{q=0}^{N}\rho_q\eta_i^{N-2}\left(-1 + (1+\xi_q)\frac{|\hat{\gamma}|}{|\Gamma^{k,l}|}\right) h_j\left(\frac{2\,|s_0|}{|\gamma^p|} - 1 + (1+\xi_q)\frac{|\hat{\gamma}|}{|\gamma^p|}\right)$$
$$- \frac{|\Gamma^{k,l}|}{2}\eta_i^{N-2}(-1)r_{-j}\rho_0 - \frac{|\Gamma^{k,l}|}{2}\eta_i^{N-2}(+1)r_{+j}\rho_N\ . \tag{22}$$

The integral matching condition (12) may then be expressed in matrix form via the transformation matrix $\overline{Q}$, given as

$$\overline{Q}_{ij} = [\overline{Q}] = [\overline{B}]^{-1}[\overline{P}], \quad \forall i \in \{1,...,N-1\},\ \forall j \in \{0,...,N\}. \tag{23}$$

It is worthwhile to note that the formulation of the $\overline{Q}$ matrix is stable as regards virtual vertices: their contribution vanishes uniformly as they approach true vertices from any direction.

Figure 3 is a diagrammatic representation of the integral matching conditions, and, in fact, the entire X_h basis. The arrows represent the descendance of mortar data to elements with the assumed priority that, in any location, all incoming contributions will be completed before any outgoing contributions can be activated. The vertex condition is illustrated by the arrows emanating from the large open boxes or circles which represent real and virtual vertices, respectively; vertex data is assigned equally to the local mortar endpoints (■=) as well as the elemental data vertices (●). The mortar projection (multiplication by the $\overline{Q}$ transformation matrix) is represented by arrows between mortar strips and elemental edges. Note that more than one arrows leading to the same edge location indicates a sum of the contributions of all intersections of γ^p with the particular $\Gamma^{k,l}$, whereas equations (21) and (22) refer to just one such intersection (i.e., just one integration strip). The role of the virtual vertex is primarily that of a temporary buffer for holding the value of the function ϕ and descending it to elemental corners.

It is now a simple matter to construct the discrete equations. In particular, we note that our basis construction allows us to express admissible *elemental* degrees-of-freedom in terms of their images via the Q transformation. (We henceforth refer to the Q operator as the "sum" of the vertex assignment and the $\overline{Q}$ matrix operations.) This, in turn, permits us to construct the global discrete equations directly from local structure-preserving elemental

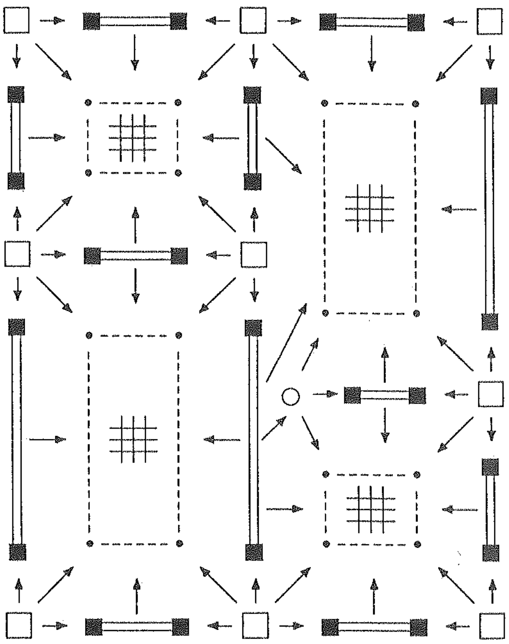

Figure 3: Diagrammatic representation of the basis for X_h on the nonconforming mesh Fig. 1a

equations, which is at the heart of the discretization-driven domain decomposition approach. We thus arrive, rather simply, at the fully discrete equations for the Poisson problem (1, 2):

$$Q^T blk(A^k) Q \underline{u} = Q^T f \ , \tag{24}$$

where $blk(A^k)$ is the matrix composed of the decoupled elemental (Neumann) problems. Equation (24) illustrates that the global Laplace operator can be thought of as a local operator "mortared" together by the Q^T, Q operations; indeed, the Q^T operator is the algebraic form of the standard direct stiffness procedure (here extended to nonconforming elements). In the implementation of iterative procedures the Q, Q^T are, of course, never explicitly formed, but rather are evaluated; diagrammatic evaluation of Q^T (direct stiffness summation) is shown in Figure 4.

Although the emphasis here is on the mortar discretization, the bases and evaluation procedure have been tailored to admit efficient iterative solution, and it is therefore appropriate to briefly indicate how the method is used in conjunction with (for example) conjugate gradient iteration [23]. To solve (24) we write

$$\begin{gathered}
\underline{u}_0; \qquad \underline{r}_0 = Q^T f - Q^T blk(A^k) Q \underline{u}_0; \qquad \underline{q}_0 = \underline{r}_0 \\
a_m = (\underline{r}_m, \underline{r}_m)/(\underline{q}_m, Q^T blk(A^k) Q \underline{q}_m) \\
\underline{u}_{m+1} = \underline{u}_m + a_m \underline{q}_m \\
\underline{r}_{m+1} = \underline{r}_m - a_m Q^T blk(A^k) Q \underline{q}_m \\
b_m = (\underline{r}_{m+1}, \underline{r}_{m+1})/(\underline{r}_m, \underline{r}_m) \\
\underline{q}_{m+1} = \underline{r}_{m+1} + b_m \underline{q}_m ,
\end{gathered} \tag{25}$$

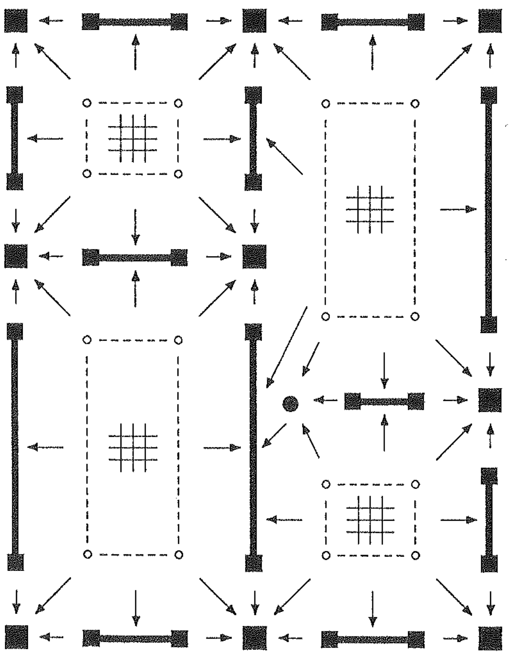

Figure 4: Direct stiffness summation Q^T of residuals on the nonconforming mesh of Figure 1a.

where m refers to iteration number, $\underline{r}_m$ is the residual, $\underline{q}_m$ the search direction and $(.,.)$ is the usual discrete inner product. All evaluations are performed through the diagrams of Figures 3 and 4. The Laplacian operations are entirely local at the elemental level, with all transmission and coupling occurring through Q. The local Laplacian calculations are the standard conforming spectral element tensor product evaluations, as the mortar decoupling allows all local structure to remain intact despite global irregularity.

3 Mortar Methods Applied to Sliding Meshes

The non-conforming approach is particularly well suited to the discretization of problems involving moving boundaries, either prescribed or free. These kinds of problems render the task of numerical approximation of partial differential equations even more difficult by introducing new geometric and discretization problems associated with the deforming domain. A step toward alleviating this additional complexity is to allow for part of the computational domain to translate (slide) [24] in a fashion complying with the continuously evolving geometry. The advantages of sliding meshes are that they eliminate laborious remeshing and interpolation procedures between consecutive time steps, and allow for $O(1)$ element translations without encountering ill-conditioned mappings. Under the current approach, the problem can be viewed as a time series of non-conforming discretizations treated in a manner consistent with the previous section.

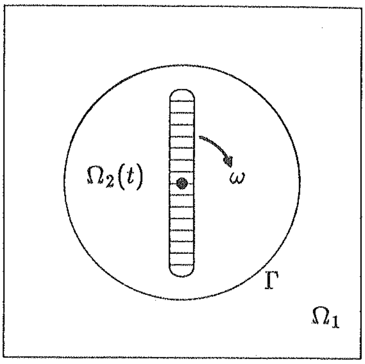

Figure 5: Model of a mixing process. The internal subdomain $\Omega_2(t)$ follows the mixer plate rotating with angular velocity ω, while sub-domain Ω_1 remains fixed for all time.

Weak form. To illustrate the sliding mesh concept we consider a model parabolic equation

$$\frac{\partial u}{\partial t} = \mathcal{L}u, \quad x \in \Omega(t) \tag{26}$$

with Dirichlet boundary conditions on $\partial\Omega(t)$. Here $\mathcal{L}$ is some second order elliptic spatial operator, such as the Laplacian. Writing (26) in weak form, we arrive at the problem: Find $u \in H_0^1(\Omega(t))$ such that

$$(v, \frac{\partial u}{\partial t}) = (v, \mathcal{L}u), \quad \forall v \in H_0^1(\Omega(t)). \tag{27}$$

Although no complications exist in the continuous formulation (27), for the discrete in-time case where the mesh is sliding or the geometry is changing (with time-step Δt), it is clear that the test function v, the new solution $u^{n+1}(\mathrm{x}) = u((n+1)\Delta t, \mathrm{x})$ and the old solution $u^n(\mathrm{x}) = u(n\Delta t, \mathrm{x})$ cannot all be defined on the same domain. We now address this issue.

For the sake of simplicity, let us focus on the simple case where $\Omega(t)$ is composed of two rigid bodies with one sliding with respect to the other along an interface Γ

$$\begin{cases} \Omega(t) = \Omega_1 \cup \Omega_2(t) \\ \Gamma = \overline{\Omega}_1 \cap (\overline{\Omega_2(t)}) \end{cases},$$

as shown in Figure 5 for a mixing process. By introducing a strip $\Gamma_\epsilon \in \Omega(t)$ which encloses the actual interface Γ by an arbitrary small amount ϵ, we can decompose any test function $v \in H_0^1$ into a (non-unique) sum of two functions v_1 and v_2 such that

$$\begin{aligned} v_1 &\in H_0^1(\Omega_1 \cup \Gamma_\epsilon) \\ v_2 &\in H_0^1(\Omega_2(t)) \\ \text{and} \quad v &= v_1 + v_2 \end{aligned} \quad , \tag{28}$$

where the v_i's are extended by zero in the regions of $\Omega(t)$ outside their particular range of definition. Problem (27) can then be written as

$$(v_1, \frac{\partial u}{\partial t}) + (v_2, \frac{\partial u}{\partial t}) = (v_1, \mathcal{L}u) + (v_2, \mathcal{L}u), \quad \forall v \in H_0^1(\Omega(t)), \tag{29}$$

or, by restricting the integrals in the left-hand side to the support of the v_i's, as

$$\int_{\Omega_1\cup\Gamma_\epsilon} v_1\frac{\partial u}{\partial t} + \int_{\Omega_2(t)} v_2\frac{\partial u}{\partial t} = (v_1,\mathcal{L}u) + (v_2,\mathcal{L}u), \quad \forall v \in H_0^1(\Omega(t)). \tag{30}$$

By considering v_2 as a Lagrangian test function satisfying a relation of the form

$$v_{2,t} + \mathbf{V}\cdot\nabla v_2 = 0 \ , \tag{31}$$

where $\mathbf{V}$ is the rigid-body motion of $\Omega_2(t)$, we can rewrite equation (30) as

$$\begin{aligned} \frac{d}{dt}(v_1,u)_{\Omega_1\cup\Gamma_\epsilon} + \frac{d}{dt}(v_2,u)_{\Omega_2(t)} + (u,\mathbf{V}\cdot\nabla v_2)_{\Omega_2(t)} &= (v_1,\mathcal{L}u) + (v_2,\mathcal{L}u) \\ \forall v = v_1 + v_2 \in H_0^1(\Omega(t)) &\text{ that satisfy (31)} \ , \end{aligned} \tag{32}$$

or

$$\frac{d}{dt}(v,u)_\Omega + (u,\mathbf{V}\cdot\nabla v_2)_{\Omega_2(t)} = (v,\mathcal{L}u) \quad \forall v = v_1 + v_2 \in H_0^1(\Omega(t)) \text{ that satisfy (31).} \tag{33}$$

Since the time derivative now appears outside the inner product in (33), we no longer have the potential inconsistency present in (27). We can now turn to the discretization.

Mortar Method. The mortars are chosen on Γ to be the set of edges of those elements that compose Ω_1. The discrete functions over $\Omega_2(t)$ are then spanned by polynomials vanishing over $\partial\Omega_2(t)$, and by the extension of traces derived from Γ mortar functions that vanish at any internal Gauss-Lobatto point of $\overline{\Omega_2(t)}/\Gamma$. The essential aspect of the mortar method is to allow the decomposition (28), that is, the mesh can slide via (31), yet maintain sufficient continuity of the test function through mortar projection.

We have chosen an implicit, Euler backward time integration scheme and an explicit, third-order Adams-Bashforth method for treatment of the diffusion and convective terms of equation (33), respectively. The corresponding fully-discrete equations can then be written as:

$$\begin{aligned} {Q^{n+1}}^T blk(B_i) Q^{n+1}\, u^{n+1} \ - \ {Q^n}^T blk(B_i) Q^n\, u^n \ = \\ \Delta t \left[\sum_{q=0}^{2} \gamma_q C^{n-q} u^{n-q} \ + \ {Q^{n+1}}^T blk(L_i) Q^{n+1} u^{n+1} \right] \end{aligned} \tag{34}$$

where

$$C^n \ = \ mask_\Gamma \ * \ {Q^n}^T \begin{bmatrix} 0 & 0 \\ 0 & (v,\mathbf{V}\cdot\nabla) \end{bmatrix} Q^n \ .$$

Here $mask_\Gamma$ is the vector $(...)^T$ representing the algebraic basis of that test function v which is unity except at mortars and vertices located on Γ, where it vanishes. There is a direct analogy between equations (34) and (33): $blk(B_i)$ contains the numerical quadrature weights corresponding to the inner product evaluations $(\cdot,\cdot)$ of (33); $blk(L_i)$ is the discrete form of the second-order elliptic operator $(v,\mathcal{L}u)$; $\sum\gamma_q C^{n-q}$ is the Adams-Bashforth explicit treatment of the discrete convection operator C^n which, after an integration by parts, is equivalent to the term $(u,\mathbf{V}\cdot\nabla v_2)$; and Q, Q^T represent the familiar transformation matrices for the projection and direct stiffness summation operations related to the non-conforming

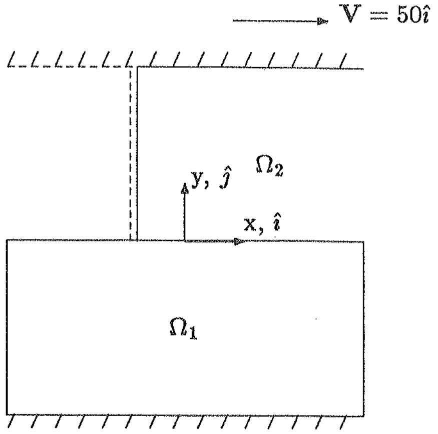

Figure 6: Sliding-mesh example problem

discretization of sliding mesh problems. We note that the distinguishing feature of our sliding-mesh approach as compared to previous work [24,25] is the *high-order* convergence, and the treatment of *second-order* spatial operators.

An Example. Having defined the variational form for the model parabolic equation in (30) and the bases for the non-conforming spaces W_h and X_h in the previous section, we will proceed with an example: Find $u(x, y, t)$ defined over the domain $\Omega = [-1, 1] \times [-1, 1]$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla^2 u \quad \text{in } \Omega \\ u(x, y, 0) &= \sin(\pi x)\sin(\pi y) \\ u(x, \pm 1, t) &= 0 \\ u(-1, y, t) &= u(1, y, t) \quad . \end{aligned} \tag{35}$$

We initiate the discretization by breaking the domain into two elements Ω_1 and Ω_2, as shown in Figure 6 , with Ω_2 translating with a velocity of $\mathbf{V} = 50\hat{\imath}$.

We plot in Figure 7 the L^∞-error as a function of N (polynomial order), for a $\Delta t = 10^{-5}$ and a final time $T_f = .0005$. The error decreases exponentially fast, as would be expected for spectral approximation of a smooth solution. In Figure 8 we show the L^∞-error as a function of Δt for a fixed spatial discretization ($N = 11$) and the same T_f. As predicted from the accuracy of the Euler backward scheme, the error follows a linear dependence on the size of time-step. In addition to these tests, we have carried out very long time integrations, $T_f = .25$, corresponding to 6.25 "wrap-arounds" of the domain, without encountering any stability problems.

As a final comment, we note that the accuracy of the temporal scheme, possibly viewed as inadequate when compared with the spatial accuracy of our discretizations, can be easily improved by chosing any appropriate higher-order integration technique, such as backward-differentiation.

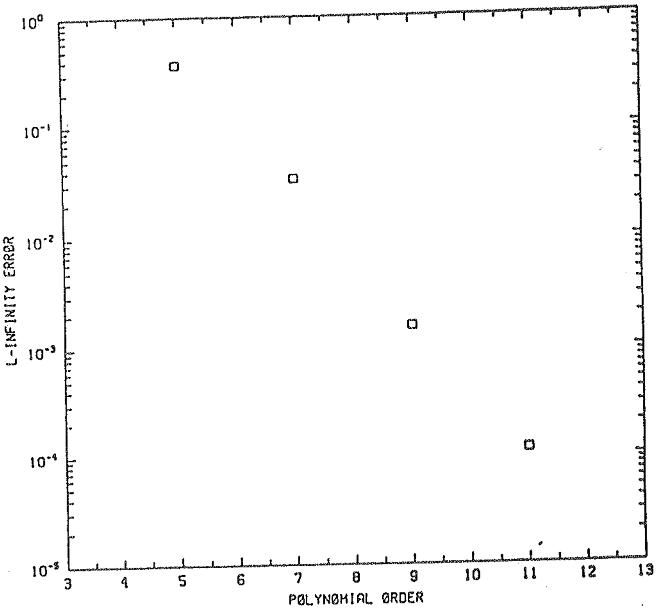

Figure 7: Convergence in space of the solution for the diffusion equation (34) for a time step of $\Delta t = .00001$, and final time of $T_f = .0005$.

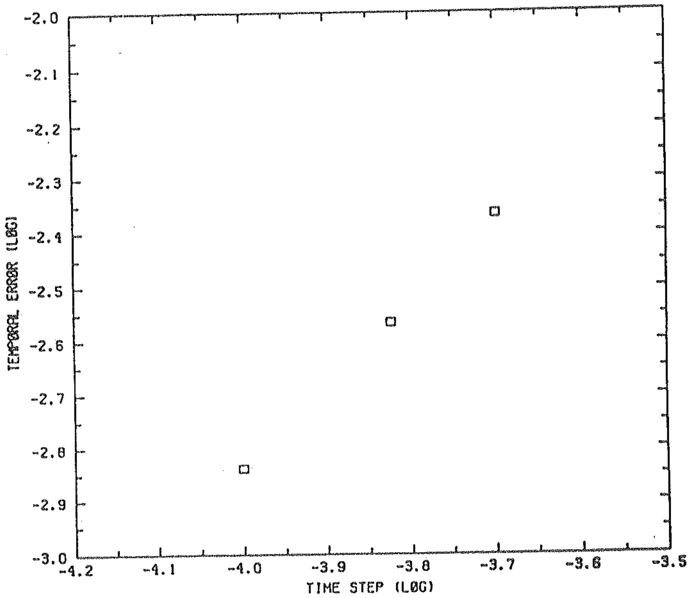

Figure 8: Convergence in time of the solution for the diffusion equation (34) for a polynomial order $N = 11$ approximation in each spatial direction.

4 Implementation

The final issue of concern to us regarding the successful development of non-conforming discretizations involves the extent to which the method can be easily implemented in a fashion that keeps intact the locality of the formulation.It has become imperative that, for simulations of ever more complex physical systems, minimizing the complexity of the implementation is of equal importance to raw computational speed. In this section, we will investigate the underlying structure of the required computational tasks for the mortar element method, differentiate among its fundamental building blocks, and describe methods that ensure a smooth operation among the differing modes of the decomposition.

During the formulation of the method in Sections 2 and 3, we have made certain to preserve locality and control at the elemental level, to ensure conceptual and actual algorithmic simplification, and high performance on parallel computer architectures. Locality condenses the development efforts to the optimal treatment of a simple, easily-replicated, functional unit, and allows for the breakdown of the computational load into largely independent sub-processes which can conceivably execute concurrently [12]. We have identified spectral elements as the foundation of our computational model, with sparse inter-element communication through mortars providing the necessary links for neighbor-neighbor interaction. By its nature, this partitioning of the required work into locally-structured / globally-unstructured operations impose widely differing needs on the underlying algorithmic design, and these issues will thus be treated separately in what follows.

Locality. Non-conforming discretizations introduce dissimilarities among elements. Across-the-board uniformity, a distinct property of classical conforming discretizations, is no longer present in the mortar element method, where objects vary considerably in size, topology, and, hence, complexity. The potentially differing polynomial order N is an obvious example of heterogeneity, as is the potentially different number of neighboring element-objects. These observations lead us to the development of element-based, irregular, composite data structures that preserve and reflect this heterogeneity by treating elements as fully-independent, autonomous, computational and information-holding entities. An illustration of this concept is shown in Figure 9, where an element object is depicted along with all of its pertinent information, including identification, type parameters, field data, and geometry data. As regards the implementation of this approach in a typical high-level computer language, say C, the element-object becomes a user-defined structure, with the associated arrows of Figure 9 directly translated into memory-address pointers providing immediate access to corresponding data blocks. Intra-element, numerical processing required by, say, the Laplacian operator of algorithm (25), is readily supported by such data structures; field data is retrievable locally in vectorizable form.

Subdomain Integration. Apart from the intensive floating-point computations, inter-element communication is the other feature of algorithm (25) that we must address. The projection of mortar data to element edges, its transpose operation of edge to mortar transformation, and the inner product vector reduction of (25) are the main events where elemental information is exchanged and summed. As illustrated in Figures 3 and 4, such communication can not only be tedious, but is also of varying complexity for different elements.

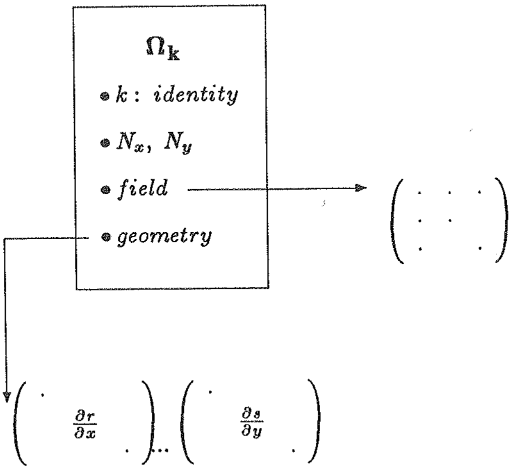
Ω_k
$\bullet\, k:$ *identity*
$\bullet\, N_x,\ N_y$
$\bullet$ *field*
$\bullet$ *geometry*
$\frac{\partial r}{\partial x}$
$\frac{\partial s}{\partial y}$

Figure 9: Illustration of an element-object. This model of a data structure optimizes storage and accessibilty of relevant information, such as field data and geometric factors.

We have developed a hierarchical connectivity pattern linking the elemental data structures of Figures 3 and 4. This tree-like topology network is naturally organized from the geometry of the problem in layers, with connections present both in an intra- and inter-layer fashion. An illustration of this global data structure is shown in Figure 11 for the sliding mesh model problem of Figure 6 for the mortar/vertex decomposition in Figure 10. The objects, i.e. elements, occupying the top of our hierarchy, lead the way to the local edges and corners. The latter are data structures in their own right, containing their own information and pointing to other memory locations representing the skeleton of the non-conforming discretization: mortars, vertices, and virtual vertices. Commonalities across elemental boundaries in the computational domain, such as coincident edges and corners, are reflected by the multiplicity of arrows leading to a particular mortar or vertex.

Such a direct mapping between geometrical relationships and data inter-connections can be likened to the unrolling of the graph of, say, Figure 1 into the underlying computer architecture. Once this data model is constructed, communication operations such as Q, Q^T are initiated from, and completed by, traversing the nodes in the hierarchy. For instance, the direct stiffness summation for the sliding-mesh problem is completed by elemental edges $(1', ..., 4')$ and corners $(1, ...4)$ projecting the local data to the *common* memory locations of the mortars (e,..., i) and vertices (a,..., d), respectively; by virtue of multiple pointers to a common memory location, the elemental contributions are automatically summed. Communication primitives that drive these operations are simple but at the same time general-purpose; assuming a consistent configuration, in the form of Figure 11, they complete the transformations of Figures 3 and 4 by a one-time walk through the elemental/mortar spaces. The symbolic manipulation required by traditional data structure models to decipher the complex geometric relationships of the computational domain are, in our case, replaced by simple "guides" which lead the way through the mortar/edge and corner/vertex interrelations.

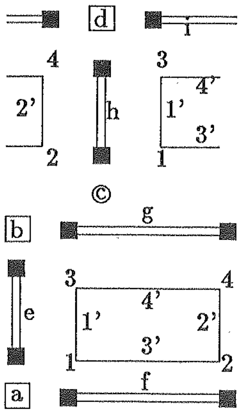

Figure 10: Skeleton structure of the sliding mesh problem as seen at a particular instant in time.

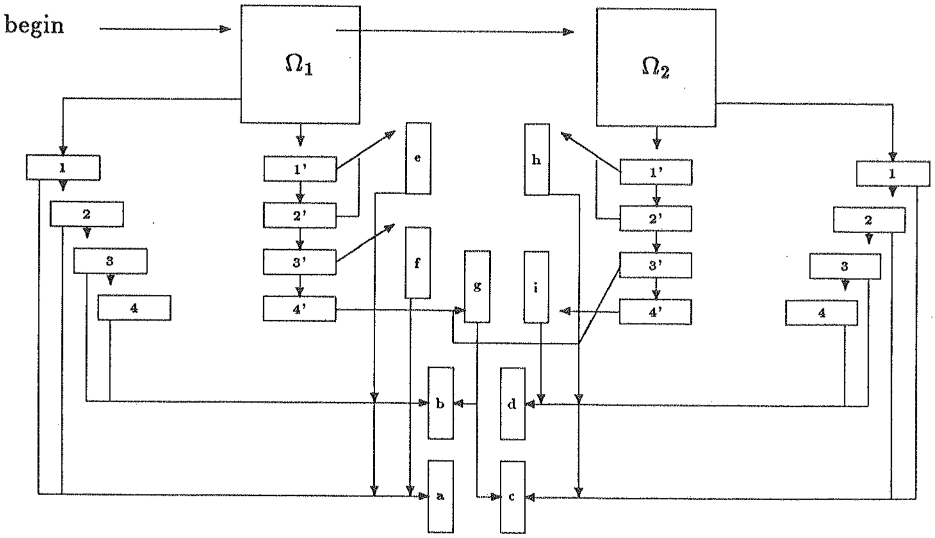

Figure 11: Global Data Structure for the sliding mesh problem of Figure 8.

The general properties of domain-decomposition methods, that is, numerically-intensive computations at the local level, and irregular memory accesses and data-exchange operations at the inter-element level, are reflected in our data structures in a simple and intuitive manner. The simplicity of this structured programming approach is translated into the rapid vertical integration of procedures into very large process entities, typical of the next generation simulation environments.

Acknowledgements

This work was supported by Avions Marcel Dassault-Bréguet Aviation, the ONR and DARPA under contract N00014-89-J-1610, the ONR under contract N00014-88-K-0188, and the NSF under Grant ASC-8806925.

References

[1] A. T. Patera (1984). A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, **54**, p. 468.

[2] Y. Maday, and A. T. Patera (1989). Spectral Element Methods for the Incompressible Navier-Stokes Equations. In A. Noor (ed.) *State-of-the-Art Surveys in Computational Mechanics*, pp. 71-143.

[3] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang (1987). *Spectral Methods in Fluid Dynamics*. Springer-Verlag.

[4] D. O. Gottlieb, and S. A. Orszag (1977). *Numerical Analysis of Spectral Methods.* SIAM, Philadelphia.

[5] P. Ciarlet (1978). *The Finite Element Method*, North Holland.

[6] V. Girault, P. A. Raviart (1986). *Finite Element Methods for the Navier-Stokes Equations*, Springer.

[7] G. Strang, and G. Fix (1973). *An Analysis of the Finite Element Method*, Prentice-Hall.

[8] Y. Maday, and R. Munoz (1988). Spectral Element Multigrid. II. Theoretical Justification, *Journal of Scientific Computing*, to appear.

[9] E. M. Ronquist, and A. T. Patera (1987). Spectral Element Multigrid. I. Formulation and Numerical results. *Journal of Scientific Computing*, **2**(4), p. 389.

[10] T. A. Zang, Y. S. Wong, and M. Y. Hussaini (1982). Spectral Multigrid Methods for elliptic equations. *Journal of Computational Physics*, **48**, p. 485.

[11] P. F. Fischer, L. W. Ho, G. E. Karniadakis, E. M. Ronquist, and A. T. Patera (1988). Recent Advances in parallel spectral element simulation of unsteady incompressible flows. *Proc. Symp. on Advances and Trends in Computational Structural Mechanics and Computational Fluid Dynamics*, Washington D.C.

[12] P. F. Fischer, and A. T. Patera (1989). Parallel spectral element solution of the Stokes Problem. *Journal of Computational Physics*, to appear.

[13] G. E. Karniadakis (1988). Numerical simulation of heat transfer from a cylinder in crossflow. *Intl. Journal of Heat and Mass Transfer*, **31**(1), p. 107.

[14] G. E. Karniadakis, B. B. Mikic, and A. T. Patera (1988). Minimum-dissipation transport enhancement by flow destabilization: Reynolds' analogy revisited. *Journal of Fluid Dynamics*, **192**, p. 365.

[15] C. Bernardi, N. Débit, and Y. Maday (1989). Coupling spectral and finite element methods for the Laplace Equation. *Mathematics of Computation*, to appear.

[16] Y. Maday, C. Mavriplis, and A.T. Patera (1988). Nonconforming Mortar Element Methods: Application to Spectral Discretizations. In T. Chan (ed) *Intl. Symp. on Domain Decomposition Methods*, SIAM pp. 392-418.

[17] M. Dorr (1989). Domain Decomposition via Lagrange multipliers. *Numerische Mathematik*, to appear.

[18] L. M. Delves, and C. A. Hall (1979). An implicit matching procedure for global element calculations. *J. Inst. Math. Appl.* **23**, p. 223.

[19] R. Adams (1975). *Sobolev Spaces.* Academic Press.

[20] C. Bernardi, Y. Maday, and A. T. Patera (1989). *J. College de France*, to appear.

[21] A. H. Stroud, and D. Secrest (1966). *Gaussian Quadrature Formulas.* Prentice-Hall.

[22] George Anagnostou (1991). *Ph.D. Thesis*, Massachusetts Institute of Technology, in progress.

[23] G. H. Golub, and C. F. Van Loan (1983). *Matrix Computations*, Johns Hopkins University Press.

[24] Karen L. Gundy-Burlet, and Man Mohan Rai (1989). Two-dimensional computations of multi-stage compressor flows using a zonal approach. *AIAA-89-2452.*

[25] J. T. Oden, S. J. Robertson, T. Strouboulis, P. Devloo, L. W. Spradley, H. V. Mc Connaughey (1986). Adaptive and moving mesh finite element methods for flow interaction problems. In M. O. Bristeau, R. Glowinski, A. Hauguel, J. Periaux (eds), *Sixth Intl. Symp. Finite Element Methods in Flow Problems*, p. 339.