Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

6. Applications of Domain Decomposition and Partition of Unity Methods in Physics and Geometry

M. Holst[1]

1. Introduction. In this article we consider a class of adaptive multilevel domain decomposition-like algorithms, built from a combination of adaptive multilevel finite element, domain decomposition, and partition of unity methods. These algorithms have several interesting features such as very low communication requirements, and they inherit a simple and elegant approximation theory framework from partition of unity methods. They are also very easy to use with highly complex sequential adaptive finite element packages, requiring little or no modification of the underlying sequential finite element software. The parallel algorithm can be implemented as a simple loop which starts off a sequential local adaptive solve on a collection of processors simultaneously.

We first review the Partition of Unity Method (PUM) of Babuška and Melenk in Section 2, and outline the PUM approximation theory framework. In Section 3, we describe a variant we refer to here as the Parallel Partition of Unity Method (PPUM), which is a combination of the Partition of Unity Method with the parallel adaptive algorithm from [4]. We then derive two global error estimates for PPUM, by exploiting the PUM analysis framework it inherits, and by employing some recent local estimates of Xu and Zhou [22]. We then discuss a duality-based variant of PPUM in Section 4 which is more appropriate for certain applications, and we derive a suitable variant of the PPUM approximation theory framework. Our implementation of PPUM-type algorithms using the FEtk and MC software packages is described in Section 5. We then present a short numerical example in Section 6 involving the Einstein constraints arising in gravitational wave models.

2. The Partition of Unity Method (PUM) of Babuška and Melenk. We first briefly review the partition of unity method (PUM) of Babuška and Melenk [1]. Let $\Omega \subset \mathbb{R}^d$ be an open set and let $\{\Omega_i\}$ be an open cover of Ω with a bounded local overlap property: For all $x \in \Omega$, there exists a constant M such that

$$\sup_i \{ \ i \mid x \in \Omega_i \ \} \leq M. \tag{2.1}$$

A Lipschitz *partition of unity* $\{\phi_i\}$ subordinate to the cover $\{\Omega_i\}$ satisfies the following five conditions:

$$\sum_i \phi_i(x) \equiv 1, \quad \forall x \in \Omega, \tag{2.2}$$

$$\phi_i \in C^k(\Omega) \quad \forall i, \quad (k \geq 0), \tag{2.3}$$

$$\text{supp } \phi_i \subset \overline{\Omega}_i, \quad \forall i, \tag{2.4}$$

$$\|\phi_i\|_{L^\infty(\Omega)} \leq C_\infty, \quad \forall i, \tag{2.5}$$

$$\|\nabla\phi_i\|_{L^\infty(\Omega)} \leq \frac{C_G}{\text{diam}(\Omega_i)}, \quad \forall i. \tag{2.6}$$

[1]UC San Diego, mholst@ucsd.edu

Several explicit constructions of partitions of unity satisfying (2.2)–(2.6) exist. The simplest construction in the case of a polygon $\Omega \subset \mathbb{R}^d$ employs global C^0 piecewise linear finite element basis functions defined on a simplex mesh subdivision $\mathcal{S}$ of Ω. The $\{\Omega_i\}$ are first built by first constructing a disjoint partitioning $\{\Omega_i^\circ\}$ of $\mathcal{S}$ using e.g. spectral or inertial bisection [4]. Each of the disjoint Ω_i° are extended to define Ω_i by considering all boundary vertices of Ω_i°; all simplices of neighboring Ω_j°, $j \neq i$ which are contained in the boundary vertex 1-rings of Ω_i° are added to Ω_i° to form Ω_i. This procedure produces the smallest overlap for the $\{\Omega_i\}$, such that the properties (2.2)–(2.5) are satisfied by the resulting $\{\phi_i\}$ built from the nodal C^0 piecewise linear finite element basis functions. Property (2.6) is also satisfied, but C_G will depend on the diameter of the overlap simplices. More sophisticated constructions with superior properties are possible; see e.g. [8, 19].

The *partition of unity method (PUM)* builds an approximation $u_{ap} = \sum_i \phi_i v_i$ where the v_i are taken from the local approximation spaces:

$$V_i \subset C^k(\Omega \cap \Omega_i) \subset H^1(\Omega \cap \Omega_i), \quad \forall i, \quad (k \geq 0). \tag{2.7}$$

The following simple lemma makes possible several useful results.

Lemma 2.1 *Let $w, w_i \in H^1(\Omega)$ with supp $w_i \subseteq \overline{\Omega \cap \Omega_i}$. Then*

$$\begin{aligned} \sum_i \|w\|^2_{H^k(\Omega_i)} &\leq M\|w\|^2_{H^k(\Omega)}, \qquad k = 0, 1 \\ \|\sum_i w_i\|^2_{H^k(\Omega)} &\leq M \sum_i \|w_i\|^2_{H^k(\Omega\cap\Omega_i)}, \qquad k = 0, 1 \end{aligned}$$

Proof. The proof follows from (2.1) and (2.2)–(2.6); see [1]. ∎

The basic approximation properties of PUM following from 2.1 are as follows.

Theorem 2.1 (Babuška and Melenk [1]) *If the local spaces V_i have the following approximation properties:*

$$\begin{aligned} \|u - v_i\|_{L^2(\Omega\cap\Omega_i)} &\leq \epsilon_0(i), \quad \forall i, \\ \|\nabla(u - v_i)\|_{L^2(\Omega\cap\Omega_i)} &\leq \epsilon_1(i), \quad \forall i, \end{aligned}$$

then the following a priori *global error estimates hold:*

$$\begin{aligned} \|u - u_{ap}\|_{L^2(\Omega)} &\leq \sqrt{M} C_\infty \left(\sum_i \epsilon_0^2(i) \right)^{1/2}, \\ \|\nabla(u - u_{ap})\|_{L^2(\Omega)} &\leq \sqrt{2M} \left(\sum_i \left(\frac{C_G}{\mathit{diam}(\Omega_i)} \right)^2 \epsilon_1^2(i) + C_\infty^2 \epsilon_0^2(i) \right)^{1/2}. \end{aligned}$$

Proof. This follows from Lemma 2.1 by taking $u - u_{ap} = \sum_i \phi_i(u - v_i)$ and then by using $w_i = \phi_i(u - v_i)$ in Lemma 2.1. ∎

Consider now the following linear elliptic problem:

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{2.8}$$

where $a_{ij} \in W^{1,\infty}(\Omega)$, $f \in L^2(\Omega)$, $a_{ij}\xi_i\xi_j \geq a_0 > 0$, $\forall \xi_i \neq 0$, where $\Omega \subset \mathbb{R}^d$ is a convex polyhedral domain. A weak formulation is:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } \langle F(u), v \rangle = 0, \quad \forall v \in H_0^1(\Omega), \tag{2.9}$$

where

$$\langle F(u), v \rangle = \int_\Omega a \nabla u \cdot \nabla v \; dx - \int_\Omega f v \; dx.$$

A general Galerkin approximation is the solution to the subspace problem:

$$\text{Find } u_{ap} \in V \subset H_0^1(\Omega) \text{ s.t. } \langle F(u_{ap}), v \rangle = 0, \; \forall v \in V \subset H_0^1(\Omega). \tag{2.10}$$

With PUM, the subspace V for the Galerkin approximation is taken to be the globally coupled *PUM space* (cf. [8]):

$$V = \left\{ v \mid v = \sum_i \phi_i v_i, \; v_i \in V_i \right\} \subset H^1(\Omega),$$

If error estimates are available for the quality of the local solutions produced in the local spaces, then the PUM approximation theory framework given in Theorem 2.1 guarantees a global solution quality.

3. A Parallel Partition of Unity Method (PPUM). A new approach to the use of parallel computers with adaptive finite element methods was presented recently in [4]. The following variant of the algorithm in [4] is described in [9], which we refer to as the *Parallel Partition of Unity Method* (or PPUM). This variant replaces the final global smoothing iteration in [4] with a reconstruction based on Babuška and Melenk's original Partition of Unity Method [1], which provides some additional approximation theory structure.

Algorithm (PPUM - Parallel Partition of Unity Method [4, 9])

1. *Discretize and solve the problem using a global coarse mesh.*
2. *Compute* a posteriori *error estimates using the coarse solution, and decompose the mesh to achieve equal error using weighted spectral or inertial bisection.*
3. *Give the entire mesh to a collection of processors, where each processor will perform a completely independent multilevel adaptive solve, restricting local refinement to only an assigned portion of the domain. The portion of the domain assigned to each processor coincides with one of the domains produced by spectral bisection with some overlap (produced by conformity algorithms, or by explicitly enforcing substantial overlap). When a processor has reached an error tolerance locally, computation stops on that processor.*
4. *Combine the independently produced solutions using a partition of unity subordinate to the overlapping subdomains.*

While the PPUM algorithm seems to ignore the global coupling of the elliptic problem, recent results on local error estimation [22], as well as some not-so-recent results on interior estimates [17], support this as provably good in some sense. The principle idea underlying the results in [17, 22] is that while elliptic problems are globally coupled, this global coupling is essentially a "low-frequency" coupling, and can be handled on the initial mesh which is much coarser than that required for approximation accuracy considerations. This idea has been exploited, for example, in [21, 22], and is why the construction of a coarse problem in overlapping domain decomposition methods is the key to obtaining convergence rates which are independent of the number of subdomains (c.f. [20]). An example showing the types of local refinements that occur within each subdomain is depicted in Figure 3.1.

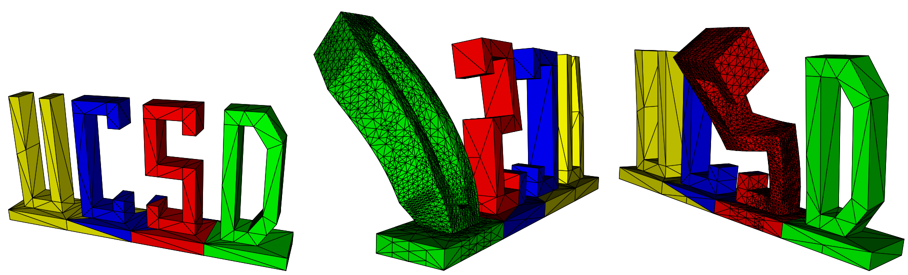

Figure 3.1: Example showing the types of local refinements created by PPUM.

To illustrate how PPUM can produce a quality global solution, we will give a global error estimate for PPUM solutions. This analysis can also be found in [9]. We can view PPUM as building a PUM approximation $u_{pp} = \sum_i \phi_i v_i$ where the v_i are taken from the local spaces:

$$V_i = \mathcal{X}_i V_i^g \subset C^k(\Omega \cap \Omega_i) \subset H^1(\Omega \cap \Omega_i), \quad \forall i, \quad (k \geq 0), \tag{3.1}$$

where $\mathcal{X}_i$ is the characteristic function for Ω_i, and where

$$V_i^g \subset C^k(\Omega) \subset H^1(\Omega), \quad \forall i, \quad (k \geq 0). \tag{3.2}$$

In PPUM, the global spaces V_i^g in (3.1)–(3.2) are built from locally enriching an initial coarse global space V_0 by locally adapting the finite element mesh on which V_0 is built. (This is in contrast to classical overlapping Schwarz methods where local spaces are often built through enrichment of V_0 by locally adapting the mesh on which V_0 is built, and then removing the portions of the mesh exterior to the adapted region.) The PUM space V is then

$$\begin{aligned} V &= \left\{ v \mid v = \sum_i \phi_i v_i, \ v_i \in V_i \right\} \\ &= \left\{ v \mid v = \sum_i \phi_i \mathcal{X}_i v_i^g = \sum_i \phi_i v_i^g, \ v_i^g \in V_i^g \right\} \subset H^1(\Omega). \end{aligned}$$

In contrast to the approach in PUM where one seeks a global Galerkin solution in the PUM space as in (2.10), the PPUM algorithm described here and in [9] builds a global approximation u_{pp} to the solution to (2.9) from decoupled *local* Galerkin solutions:

$$u_{pp} = \sum_i \phi_i u_i = \sum_i \phi_i u_i^g, \tag{3.3}$$

where each u_i^g satisfies:

$$\text{Find } u_i^g \in V_i^g \text{ such that } \langle F(u_i^g), v_i^g \rangle = 0, \quad \forall v_i^g \in V_i^g. \tag{3.4}$$

We have the following global error estimate for the approximation u_{pp} in (3.3) built from (3.4) using the local PPUM parallel algorithm.

Theorem 3.1 *Assume the solution to (2.8) satisfies $u \in H^{1+\alpha}(\Omega)$, $\alpha > 0$, that quasi-uniform meshes of sizes h and $H > h$ are used for Ω_i^0 and $\Omega \backslash \Omega_i^0$ respectively, and that diam$(\Omega_i) \geq 1/Q > 0$ $\forall i$. If the local solutions are built from C^0 piecewise linear finite elements, then the global solution u_{pp} in (3.3) produced by Algorithm PPUM satisfies the following global error bounds:*

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \end{aligned}$$

where $P =$ number of local spaces V_i. Further, if $H \leq h^{\alpha/(1+\alpha)}$ then:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \max\{C_1, C_2\} h^\alpha, \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \max\{C_1, C_2\} h^\alpha, \end{aligned}$$

so that the solution produced by Algorithm PPUM is of optimal order in the H^1-norm.

Proof. Viewing PPUM as a PUM gives access to the *a priori* estimates in Theorem 2.1; these require local estimates of the form:

$$\begin{aligned} \|u - u_i\|_{L^2(\Omega \cap \Omega_i)} = \|u - u_i^g\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_0(i), \\ \|\nabla(u - u_i)\|_{L^2(\Omega \cap \Omega_i)} = \|\nabla(u - u_i^g)\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_1(i). \end{aligned}$$

Such local *a priori* estimates are available for problems of the form (2.8) [17, 22]. They can be shown to take the following form:

$$\|u - u_i^g\|_{H^1(\Omega_i \cap \Omega)} \leq C \left(\inf_{v_i^0 \in V_i^0} \|u - v_i^0\|_{H^1(\Omega_i^0 \cap \Omega)} + \|u - u_i^g\|_{L^2(\Omega)} \right)$$

where

$$V_i^0 \subset C^k(\Omega_i^0 \cap \Omega) \subset H^1(\Omega_i \cap \Omega),$$

and where

$$\Omega_i \subset\subset \Omega_i^0, \qquad \Omega_{ij} = \Omega_i^0 \bigcap \Omega_i^0, \qquad |\Omega_{ij}| \approx |\Omega_i| \approx |\Omega_j|.$$

Since we assume $u \in H^{1+\alpha}(\Omega)$, $\alpha > 0$, and since quasi-uniform meshes of sizes h and $H > h$ are used for Ω_i^0 and $\Omega \backslash \Omega_i^0$ respectively, we have:

$$\begin{aligned} \|u - u_i^g\|_{H^1(\Omega_i \cap \Omega)} &= \left(\|u - u_i^g\|_{L^2(\Omega_i \cap \Omega)}^2 + \|\nabla(u - u_i^g)\|_{L^2(\Omega_i \cap \Omega)}^2 \right)^{1/2} \\ &\leq C_1 h^\alpha + C_2 H^{1+\alpha}. \end{aligned}$$

I.e., in this setting we can use $\epsilon_0(i) = \epsilon_1(i) = C_1 h^\alpha + C_2 H^{1+\alpha}$. The *a priori* PUM estimates in Theorem 2.1 then become:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{M} C_\infty \left(\sum_i (C_1 h^\alpha + C_2 H^{1+\alpha})^2 \right)^{1/2}, \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2M} \\ &\cdot \left(\left[\sum_i \left(\frac{C_G}{\mathrm{diam}(\Omega_i)} \right)^2 + C_\infty^2 \right] (C_1 h^\alpha + C_2 H^{1+\alpha})^2 \right)^{1/2}. \end{aligned}$$

If $P =$ number of local spaces V_i, and if $\mathrm{diam}(\Omega_i) \geq 1/Q > 0 \ \ \forall i$, this is simply:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right). \end{aligned}$$

If $H \leq h^{\alpha/(1+\alpha)}$ then u_{pp} from PPUM is asymptotically as good as a global Galerkin solution when the error is measured in the H^1-norm. ∎

Local versions of Theorem 3.1 appear in [22] for a variety of related parallel algorithms. Note that the local estimates in [22] hold more generally for nonlinear versions of (2.8), so that Theorem 3.1 can be shown to hold in a more general setting. Finally, it should be noted that improving the estimates in the L^2-norm is not generally possible; the required local estimates simply do not hold. Improving the solution quality in the L^2-norm generally requires more global information. However, for some applications one is more interested in a quality approximation of the *gradient* or the *energy* of the solution rather than to the solution itself.

4. Duality-based PPUM. We first briefly review a standard approach to the use of duality methods in error estimation. (cf. [6, 7] for a more complete discussion). Consider the weak formulation (2.9) involving a possibly nonlinear differential operator $F : H_0^1(\Omega) \mapsto H^{-1}(\Omega)$, and a Galerkin approximation u_{ap} satisfying (2.10). If $F \in C^1$, the generalized Taylor expansion exists:

$$F(u+h) = F(u) + \left\{ \int_0^1 DF(u + \xi h) d\xi \right\} h.$$

With $e = u - u_{ap}$, and with $F(u) = 0$, leads to the linearized error equation:

$$F(u_{ap}) = F(u - e) = F(u) + \mathcal{A}(u_{ap} - u) = -\mathcal{A}e,$$

where the linearization operator $\mathcal{A}$ is defined as:

$$\mathcal{A} = \int_0^1 DF(u + \xi h) d\xi.$$

Assume now we are interested in a linear functional of the error $l(e) = \langle e, \psi \rangle$, where ψ is the (assumed accessible) Riesz-representer of $l(\cdot)$. If $\phi \in H_0^1(\Omega)$ is the solution to the linearized dual problem:

$$\mathcal{A}^T \phi = \psi,$$

then we can exploit the linearization operator $\mathcal{A}$ and its adjoint $\mathcal{A}^T$ to give the following identity:

$$\langle e, \psi \rangle = \langle e, \mathcal{A}^T \phi \rangle = \langle \mathcal{A} e, \phi \rangle = -\langle F(u_{ap}), \phi \rangle. \tag{4.1}$$

If we can compute an approximation $\phi_{ap} \in \mathcal{V} \subset H_0^1(\Omega)$ to the linearized dual problem then we can estimate the error by combining this with the (computable) residual $F(u_{ap})$:

$$|\langle e, \psi \rangle| = |\langle F(u_{ap}), \phi \rangle| = |\langle F(u_{ap}), \phi - \phi_{ap} \rangle|,$$

where the last term is a result of (2.10). The term on the right is then estimated locally using assumptions on the quality of the approximation ϕ_{ap} and by various numerical techniques; cf. [6]. The local estimates are then used to drive adaptive mesh refinement. This type of duality-based error estimation has been shown to be useful for certain applications in engineering and other areas where accuracy in a linear functional of the solution is important, but accuracy in the solution itself is not (cf. [7]).

Consider now this type of error estimation in the context of domain decomposition and PPUM. Given a linear or nonlinear weak formulation as in (2.9), we are interested in the solution u as well as in the error in PPUM approximations u_{pp} as defined in (3.3)–(3.4). If a global linear functional $l(u - u_{pp})$ of the error $u - u_{pp}$ is of interest rather than the error itself, then we can formulate a variant of the PPUM parallel algorithm which has in some sense a more general approximation theory framework than that of the previous section. There are no assumptions beyond solvability of the local problems and of the global dual problems with localized data, and perhaps some minimal smoothness assumptions on the dual solution. In particular, the theory does not require local *a priori* error estimates; the local *a priori* estimates are replaced by solving global dual problem problems with localized data, and then incorporating the dual solutions explictly into the *a posteriori* error estimate. As a result, the large overlap assumption needed for the local estimates in the proof of Theorem 3.1 is unnecessary. Similarly, the large overlap assumption needed to achieve the bounded gradient property (2.6) is no longer needed.

The following result gives a global bound on a linear functional of the error based on satisfying local computable *a posteriori* bounds involving localized dual problems.

Theorem 4.1 *Let $\{\phi_i\}$ be a partition of unity subordinate to a cover $\{\Omega_i\}$. If ψ is the Riesz-representer for a linear functional $l(u)$, then the functional of the error in the PPUM approximation u_{pp} from (3.3) satisfies*

$$l(u - u_{pp}) = -\sum_{k=1}^{p} \langle F(u_i^g), \omega_i \rangle,$$

where u_i^g are the solutions to the subspace problems in (3.4), and where the ω_i are the solutions to the following global dual problems *with localized data:*

Find $\omega_i \in H_0^1(\Omega)$ such that $(A^T \omega_i, v)_{L^2(\Omega)} = (\phi_i \psi, v)_{L^2(\Omega)}, \quad \forall v \in H_0^1(\Omega).$ (4.2)

Moreover, if the local residual $F(u_i^g)$, weighted by the localized dual solution ω_i, satisfies the following error tolerance in each subspace:

$$|\langle F(u_i^g), \omega_i \rangle| < \frac{\epsilon}{p}, \qquad i = 1, \ldots, p \tag{4.3}$$

then the linear functional of the global error $u - u_{pp}$ satisfies

$$|l(u - u_{pp})| < \epsilon. \tag{4.4}$$

Proof. With $l(u - u_{pp}) = (u - u_{pp}, \psi)_{L^2(\Omega)}$, the localized representation comes from:

$$(u - u_{pp}, \psi)_{L^2(\Omega)} = (\sum_{k=1}^{p} \phi_i u - \sum_{i=1}^{p} \phi_i u_i^g, \psi)_{L^2(\Omega)} = \sum_{k=1}^{p} (\phi_i(u - u_i^g), \psi)_{L^2(\Omega \cap \Omega_i)}.$$

From (4.1) and (4.2), each term in the sum can be written in terms of the local residual $F(u_i^g)$ as follows:

$$\begin{aligned}
(\phi_i(u - u_i^g), \psi)_{L^2(\Omega \cap \Omega_i)} &= (u - u_i^g, \phi_i \psi)_{L^2(\Omega \cap \Omega_i)} \\
&= (u - u_i^g, \mathcal{A}^T \omega_i)_{L^2(\Omega)} \\
&= (\mathcal{A}(u - u_i^g), \omega_i)_{L^2(\Omega)} \\
&= -(F(u_i^g), \omega_i)_{L^2(\Omega)}.
\end{aligned}$$

This gives then

$$|(u - u_{pp}, \psi)_{L^2(\Omega)}| \leq \sum_{k=1}^{p} |\langle F(u_i^g), \psi \rangle| < \sum_{k=1}^{p} \frac{\epsilon}{p} = \epsilon.$$

■

We will make a few additional remarks about the parallel adaptive algorithm which arises naturally from Theorem 4.1. Unlike the case in Theorem 3.1, the constants C_∞ and C_G in (2.5) and (2.6) do not impact the error estimate in Theorem 4.1, removing the need for the *a priori* large overlap assumptions. Moreover, local *a priori* estimates are not required either, removing a second separate large overlap assumption that must be made to prove results such as Theorem 3.1. Using large overlap of *a priori* unknown size to satisfy the requirements for Theorem 3.1 seems unrealistic for implementations. On the other hand, no such *a priori* assumptions are required to use the result in Theorem 4.1 as the basis for a parallel adaptive algorithm. One simply solves the local dual problems (4.2) on each processor independently, adapts the mesh on each processor independently until the computable local error estimate satisfies the tolerance (4.3), which then guarantees that the functional of the global error meets the target in (4.4).

Whether such a duality-based approach will produce an efficient parallel algorithm is not at all clear; however, it is at least a mechanism for decomposing the solution to an elliptic problem over a number of subdomains. Note that ellipticity is not used in Theorem 4.1, so that the approach is also likely reasonable for other classes of PDE. These questions, together with a number of related duality-based decomposition algorithms are examined in more detail in [5]. The analysis in [5] is based on a different approach involving estimates of Green function decay rather than through partition of unity methods.

5. Implementation in FETK **and** MC**.** Our implementations are performed using FETK and MC (see [9] for a more complete discussion of MC and FETK). MC is the adaptive multilevel finite element software kernel within FETK, a large collection of collaboratively developed finite element software tools based at UC San Diego (see `www.fetk.org`). MC is written in ANSI C (as is most of FETK), and is designed to produce highly accurate numerical solutions to nonlinear covariant elliptic systems of tensor equations on 2- and 3-manifolds in an optimal or nearly-optimal way. MC employs *a posteriori* error estimation, adaptive simplex subdivision, unstructured algebraic multilevel methods, global inexact Newton methods, and numerical continuation methods. Several of the features of MC are somewhat unusual, allowing for the treatment of very general nonlinear elliptic systems of tensor equations on domains with the structure of (Riemannian) 2- and 3-manifolds. Some of these features are:

- *Abstraction of the elliptic system*: The elliptic system is defined only through a nonlinear weak form over the domain manifold, along with an associated linearization form, also defined everywhere on the domain manifold (precisely the forms $\langle F(u), v\rangle$ and $\langle DF(u)w, v\rangle$ in the discussions above).
- *Abstraction of the domain manifold*: The domain manifold is specified by giving a polyhedral representation of the topology, along with an abstract set of coordinate labels of the user's interpretation, possibly consisting of multiple charts. MC works only with the topology of the domain, the connectivity of the polyhedral representation. The geometry of the domain manifold is provided only through the form definitions, which contain the manifold metric information.
- *Dimension independence*: Exactly the same code paths in MC are taken for both two- and three-dimensional problems (as well as for higher-dimensional problems). To achieve this dimension independence, MC employs the simplex as its fundamental geometrical object for defining finite element bases.

As a consequence of the abstract weak form approach to defining the problem, the complete definition of a complex nonlinear tensor system such as large deformation nonlinear elasticity requires writing only a few hundred lines of C to define the two weak forms. Changing to a different tensor system (e.g. the example later in the paper involving the constraints in the Einstein equations) involves providing only a different definition of the forms and a different domain description.

A datastructure referred to as the *ringed-vertex* (cf. [9]) is used to represent meshes of d-simplices of arbitrary topology. This datastructure is illustrated in Figure 5.1. The ringed-vertex datastructure is similar to the winged-edge, quad-edge, and edge-facet datastructures commonly used in the computational geometry community for representing 2-manifolds [15], but it can be used more generally to represent arbitrary d-manifolds, $d \geq 2$. It maintains a mesh of d-simplices with near minimal storage, yet for shape-regular (non-degenerate) meshes, it provides $O(1)$-time access to all information necessary for refinement, un-refinement, and Petrov-Galerkin discretization of a differential operator. The ringed-vertex datastructure also allows for dimension independent implementations of mesh refinement and mesh manipulation, with one implementation (the same code path) covering arbitrary dimension d. An interesting feature of this datastructure is that the C structures used for vertices, simplices,

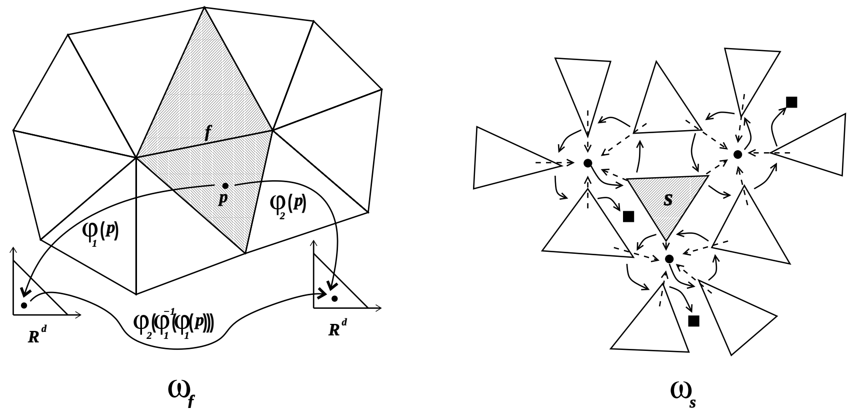

Figure 5.1: Polyhedral manifold representation. The figure on the left shows two overlapping polyhedral (vertex) charts consisting of the two rings of simplices around two vertices sharing an edge. The region consisting of the two darkened triangles around the face f is denoted ω_f, and represents the overlap of the two vertex charts. Polyhedral manifold topology is represented by MC using the *ringed-vertex* (or *RIVER*) datastructure. The datastructure is illustrated for a given simplex s in the figure on the right; the topology primitives are vertices and d-simplices. The collection of the simplices which meet the simplex s at its vertices (which then includes those simplices that share faces as well) is denoted as ω_s.

and edges are all of fixed size, so that a fast array-based implementation is possible, as opposed to a less-efficient list-based approach commonly taken for finite element implementations on unstructured meshes. A detailed description of the ringed-vertex datastructure, along with a complexity analysis of various traversal algorithms, can be found in [9].

Our modifications to MC to implement PPUM are minimal, and are described in detail in [4]. These modifications involve primarily forcing the error indicator to ignore regions outside the subdomain assigned to the particular processor. The implementation does not form an explicit partition of unity or a final global solution; the solution must be evaluated locally by locating the disjoint subdomain containing the physical region of interest, and then by using the solution produced by the processor assigned to that particular subdomain. Note that forming a global conforming mesh as needed to build a global partition of unity is possible even in a very loosely coupled parallel environment, due to the deterministic nature of the bisection-based algorithms we use for simplex subdivision (see [9]). For example, if bisection by longest edge (supplemented with tie-breaking) is used to subdivide any simplex that is refined on any processor, then the progeny types, shapes, and configurations can be predicted in a completely determinstic way. If two simplices share faces across a subdomain boundary, then they are either compatible (their triangular faces exactly match), or one of the simplices has been bisected more times than its neighbor. By exchanging only the generation numbers between subdomains, a global conforming mesh can be reached using only additional bisection.

6. Example 1: The Einstein Constraints in Gravitation. The evolution of the gravitational field was conjectured by Einstein to be governed by twelve coupled first-order hyperbolic equations for the metric of space-time and its time derivative, where the evolution is constrained for all time by a coupled four-component elliptic system. The theory basically gives what is viewed as the correct interpretation of the graviational field as a bending of space and time around matter and energy, as opposed to the classical Newtonian view of the gravitational field as analogous to the electrostatic field; cf. Figure 6.1. The four-component elliptic constraint system

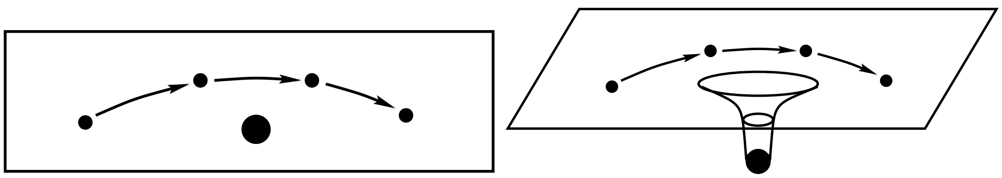

Figure 6.1: Newtonian versus general relativistic explanations of gravitation: the small mass simply follows a geodesic on the curved surface created by the large mass.

consists of a nonlinear scalar *Hamiltonian constraint*, and a linear 3-vector *momentum* constraint. The evolution and constraint equations, similar in some respects to Maxwell's equations, are collectively referred to as the *Einstein equations*. Solving the constraint equations numerically, separately or together with the evolution equations, is currently of great interest to the physics community due to the recent construction of a new generation of gravitational wave detectors (cf. [12, 11] for more detailed discussions of this application).

Allowing for both Dirichlet and Robin boundary conditions on a 3-manifold $\mathcal{M}$ with boundary $\partial\mathcal{M} = \partial_0\mathcal{M} \cup \partial_1\mathcal{M}$, as typically the case in black hole and neutron star models (cf. [12, 11]), the strong form of the constraints can be written as:

$$\begin{aligned}
\hat{\Delta}\phi &= \frac{1}{8}\hat{R}\phi + \frac{1}{12}(\mathrm{tr}K)^2\phi^5 && (6.1)\\
&\quad -\frac{1}{8}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3} \ \text{ in } \mathcal{M}, \\
\hat{n}_a\hat{D}^a\phi + c\phi &= z \text{ on } \partial_1\mathcal{M}, && (6.2)\\
\phi &= f \text{ on } \partial_0\mathcal{M}, && (6.3)\\
\hat{D}_b(\hat{L}W)^{ab} &= \frac{2}{3}\phi^6\hat{D}^a\mathrm{tr}K + 8\pi\hat{j}^a \ \text{ in } \mathcal{M}, && (6.4)\\
(\hat{L}W)^{ab}\hat{n}_b + C^a{}_bW^b &= Z^a \text{ on } \partial_1\mathcal{M}, && (6.5)\\
W^a &= F^a \ \text{ on } \partial_0\mathcal{M}, && (6.6)
\end{aligned}$$

where the following standard notation has been employed:

$$\begin{aligned}
\hat{\Delta}\phi &= \hat{D}_a\hat{D}^a\phi, \\
(\hat{L}W)^{ab} &= \hat{D}^aW^b + \hat{D}^bW^a - \frac{2}{3}\hat{\gamma}^{ab}\hat{D}_cW^c, \\
\mathrm{tr}K &= \gamma^{ab}K_{ab}, \\
(C_{ab})^2 &= C^{ab}C_{ab}.
\end{aligned}$$

In the tensor expressions above, there is an implicit sum on all repeated indices in products, and the covariant derivative with respect to the fixed background metric $\hat{\gamma}_{ab}$ is denoted as $\hat{D}_a$ The remaining symbols in the equations ($\hat{R}$, K, ${}^*\!\hat{A}_{ab}$, $\hat{\rho}$, $\hat{j}^a$, z, Z^a, f, F^a, c, and C^a_b) represent various physical parameters, and are described in detail in [12, 11] and the references therein. Stating the system as set of tensor equations comes from the need to work with domains which generally have the structure of 3-manifolds rather than single open sets in $\mathbb{R}^3$ (cf. [9]).

Equations (6.1)–(6.6) are known to be well-posed only for certain problem data and manifold topologies [16, 13]. Note that if multiple solutions in the form of folds or bifurcations are present in solutions of (6.1)–(6.6) then path-following numerical methods will be required for numerical solution [14]. For our purposes here, we select the problem data and manifold topology such that the assumptions for the two general well-posedness results in [12] hold for (6.1)–(6.6). The assumptions required for the two results in [12] are quite weak, and are, for the most part, minimal assumptions beyond those required to give a well-defined weak formulation in L^p-based Sobolev spaces.

In [9], two quasi-optimal *a priori* error estimates are established for Galerkin approximations to the solutions to (6.1)–(6.6). These take the form (see Theorems 4.3 and 4.4 in [9]):

$$\|u - u_h\|_{H^1(\mathcal{M})} \leq C \inf_{v \in V_h} \|u - v\|_{H^1(\mathcal{M})} \tag{6.7}$$

$$\|u - u_h\|_{L^2(\mathcal{M})} \leq C a_h \inf_{v \in V_h} \|u - v\|_{H^1(\mathcal{M})}, \tag{6.8}$$

where $V_h \subset H^1(\mathcal{M})$ is e.g. a finite element space. In the case of the momentum constraint, there is a restriction on the size of the elements in the underlying finite element mesh for the above results to hold, characterized above by the parameter a_h. This restriction is due to the fact that the result is established through of the Gårding inequality result due to Schatz [18]. In the case of the Hamiltonian constraint, there are no restrictions on the approximation spaces.

To use MC to calculate the initial bending of space and time around a single massive black hole by solving the above constraint equations, we place a spherical object of unit radius in space, and infinite space is truncated with an enclosing sphere of radius 100. (This outer boundary may be moved further from the object to improve the accuracy of boundary condition approximations.) Reasonable choices for the remaining functions and parameters appearing in the equations are used below to completely specify the problem for use as an illustrative numerical example. (More careful examination of the various functions and parameters appear in [12], and a number of detailed experiments with more physically meaningful data appear in [11].)

We then generate an initial (coarse) mesh of tetrahedra inside the enclosing sphere, exterior to the spherical object within the enclosing sphere. The mesh is generated by adaptively bisecting an initial mesh consisting of an icosahedron volume filled with tetrahedra. The bisection procedure simply bisects any tetrahedron which touches the surface of the small spherical object. When a reasonable approximation to the surface of the sphere is obtained, the tetrahedra completely inside the small spherical object are removed, and the points forming the surface of the small spherical object are projected to the spherical surface exactly. This projection involves solving a linear elasticity problem, together with the use of a shape-optimization-based smoothing procedure. The smoothing procedure locally optimizes the shape measure function described in [9] in an iterative fashion. A much improved binary black hole mesh generator has been developed by D. Bernstein; the new mesh generator is described in [11] along with a number of more detailed examples using MC.

The initial coarse mesh is shown in Figure 6.2, generated using the procedure described above, has approximately 30,000 tetrahedral elements and 5,000 vertices. To solve the problem on a 4-processor computing cluster using a PPUM-like algorithm, we begin by partitioning the domain into four subdomains (shown in Figure 6.3) with approximately equal error using the recursive spectral bisection algorithm described in [4]. The four subdomain problems are then solved independently by MC, starting from the complete coarse mesh and coarse mesh solution. The mesh is adaptively refined in each subdomain until a mesh with roughly 50000 vertices is obtained (yielding subdomains with about 250000 simplices each).

The refinement performed by MC is confined primarily to the given region as driven by the weighted residual error indicator described in [9], with some refinement into adjacent regions due to the closure algorithm which maintains conformity and shape regularity. The four problems are solved completely independently by the sequential adaptive software package MC. One component of the solution (the conformal factor ϕ) of the elliptic system is depicted in Figures 6.4 (the subdomain 0 and subdomain 1 solutions).

A number of more detailed examples involving the contraints, using more physically meaningful data, appear in [11]. Application of PPUM to massively parallel simulations of microtubules and other extremely large and complex biological structures can be found in [3, 2]. The results in [3, 2] demonstrate both good parallel scaling of PPUM as well as quality approximation of the gradient of electrostatic potentials (solutions to the Poisson-Boltzmann equation; cf. [10]).

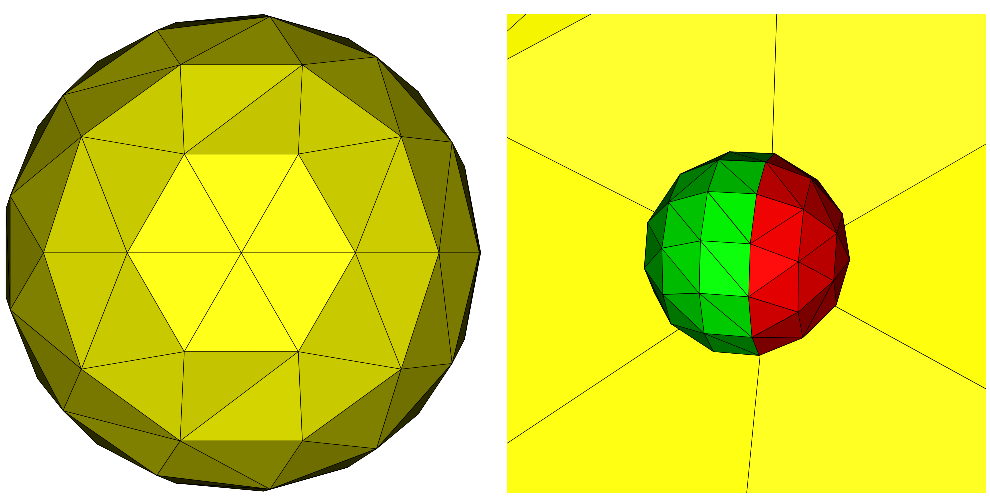

Figure 6.2: Recursize spectral bisection of the single hole domain into four subdomains (boundary surfaces of three of the four subdomains are shown).

REFERENCES

[1] I. Babuška and J. M. Melenk. The partition of unity finite element method. *Internat. J. Numer. Methods Engrg.*, 40:727–758, 1997.

[2] N. Baker, D. Sept, M. J. Holst, and J. A. McCammon. The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. *IBM Journal of Research and Development*, 45:427–438, 2001.

[3] N. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA*, 98:10037–10041, 2001.

[4] R. E. Bank and M. J. Holst. A new paradigm for parallel adaptive mesh refinement. *SISC*, 22(4):1411–1443, 2000.

[5] D. Estep, M. J. Holst, and M. Larson. *Solution Decomposition using Localized Influence Functions.* In Preparation.

[6] D. Estep, M. J. Holst, and D. Mikulencak. Accounting for stability: *a posteriori* error estimates for finite element methods based on residuals and variational analysis. *Communications in Numerical Methods in Engineering*, 18(1):15–30, 2002.

[7] M. B. Giles and E. Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 2002.

[8] M. Griebel and M. A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. *SISC*, 22(3):853–890, 2000.

[9] M. J. Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Advances in Computational Mathematics*, 15:139–191, 2001.

[10] M. J. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: algorithms and examples. *J. Comput. Chem.*, 21:1319–1342, 2000.

[11] M. J. Holst and D. Bernstein. *Adaptive Finite Element Solution of the Initial Value Problem in General Relativity I. Algorithms.* In Preparation.

[12] M. J. Holst and D. Bernstein. *Some results on non-constant mean curvature solutions to the Einstein constraint equations.* In Preparation.

[13] J. Isenberg and V. Moncrief. A set of nonconstant mean curvature solutions of the Einstein constraint equations on closed manifolds. *Classical and Quantum Gravity*, 13:1819–1847, 1996.

[14] H. B. Keller. *Numerical Methods in Bifurcation Problems.* Tata Institute of Fundamental Research, 1987.

[15] E. P. Mücke. *Shapes and Implementations in Three-Dimensional Geometry.* PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1993.

[16] N. Murchadha and J. W. York. Existence and uniqueness of solutions of the Hamiltonian constraint of general relativity on compact manifolds. *J. Math. Phys.*, 14(11):1551–1557, 1973.

[17] J. A. Nitsche and A. H. Schatz. Interior estimates for Ritz-Galerkin methods. *Math. Comp.*, 28:937–958, 1974.

[18] A. H. Schatz. An observation concerning Ritz-Galerkin methods with indefinite bilinear forms. *Math. Comp.*, 28(128):959–962, 1974.

[19] D. S. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proceedings of the 1968 ACM National Conference, New York*, pages 517–524, 1968.

[20] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, December 1992.

[21] J. Xu. Two-grid discretization techniques for linear and nonlinear pdes. *SIAM J. N. A.*, 33:1759–1777, 1996.

[22] J. Xu and A. hui Zhou. Local and parallel finite element algorithms based on two-grid discretizations. *Math. Comput.*, 69:881–909, 2000.

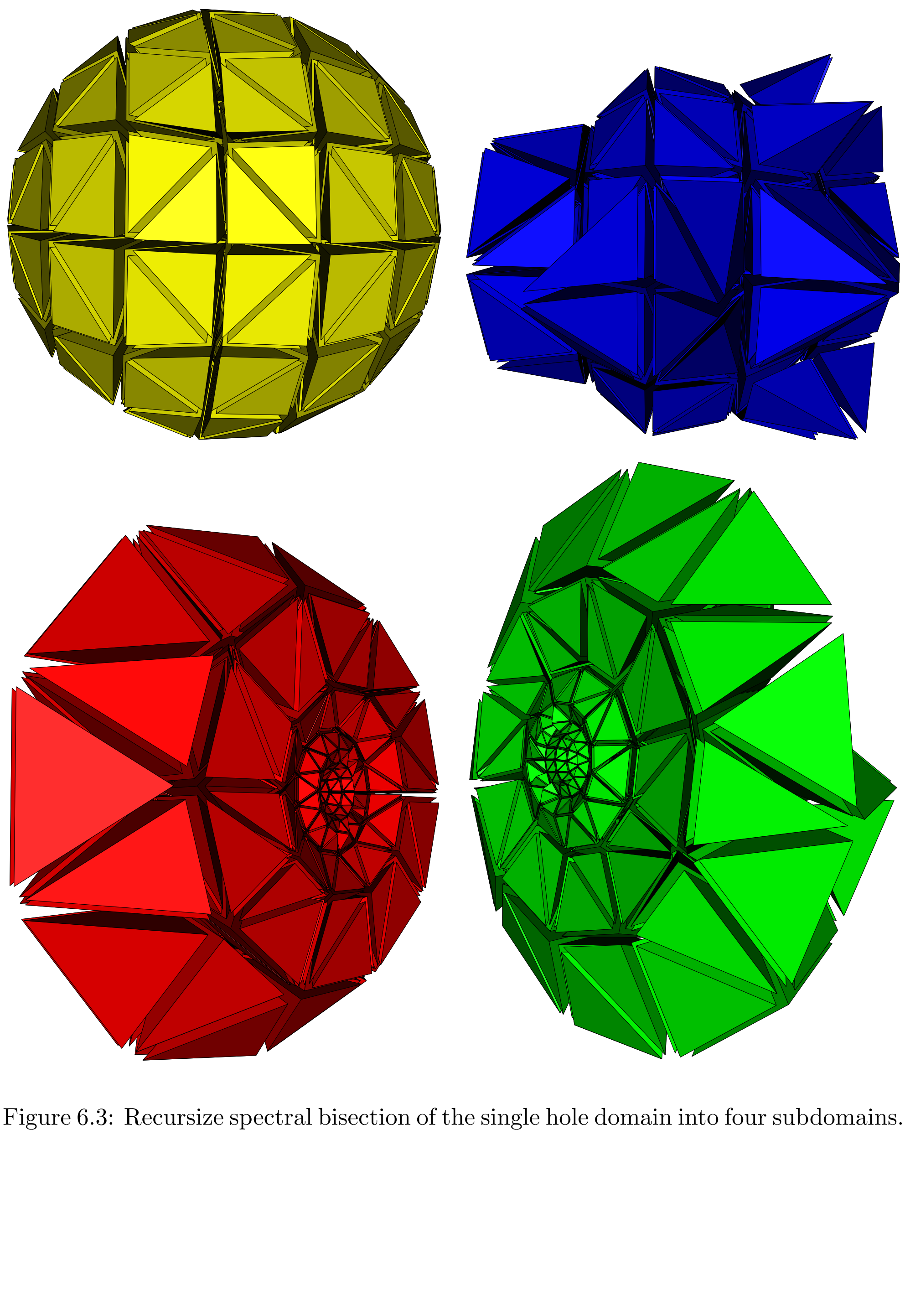

Figure 6.3: Recursize spectral bisection of the single hole domain into four subdomains.

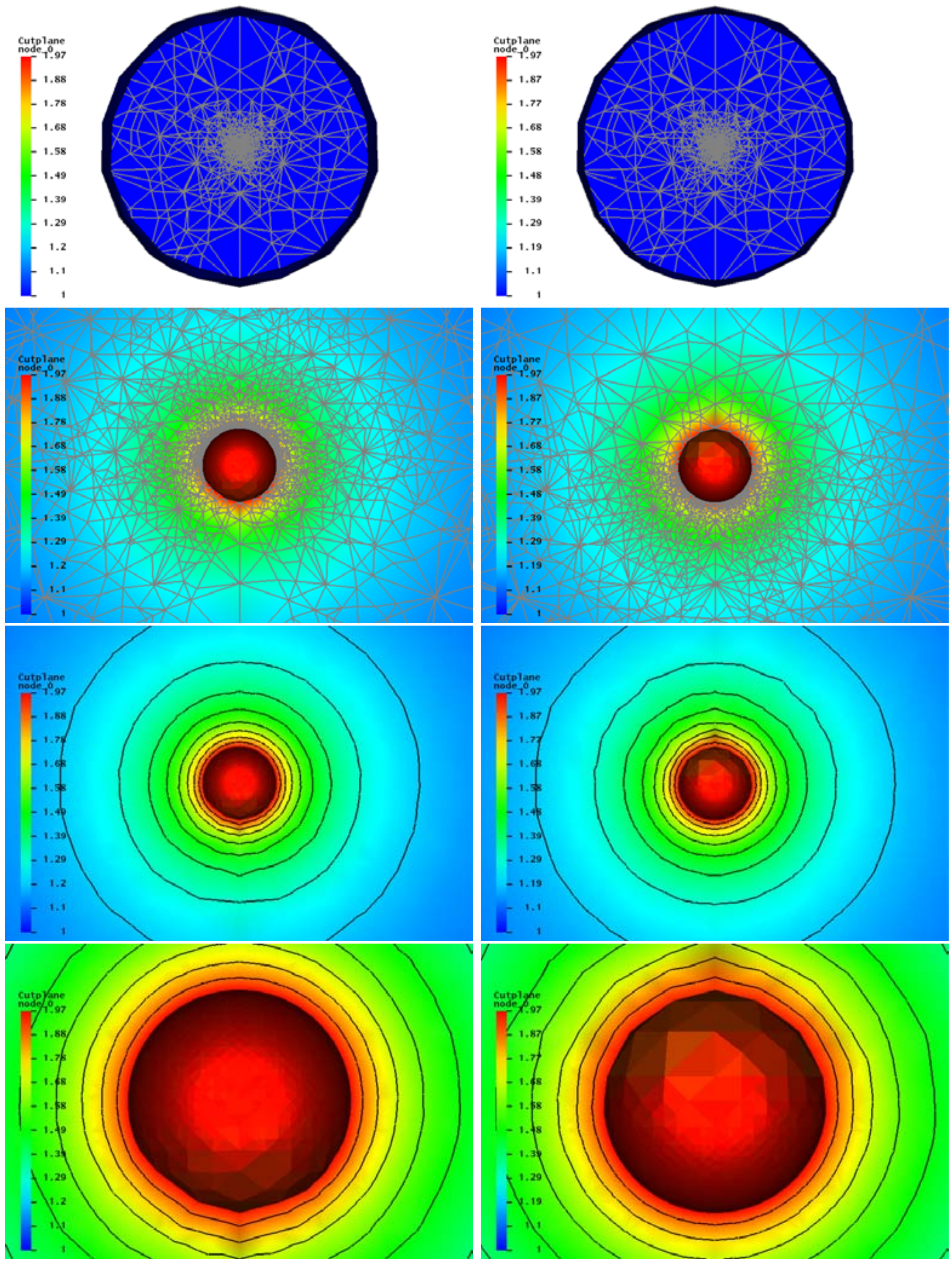

Figure 6.4: Decoupling of the scalar conformal factor in the initial data using PPUM; domain 0 in the left column, and domain 1 on the right.