Generating Initial Data in General Relativity using Adaptive Finite Element Methods

B. Aksoylu[1], **D. Bernstein**[2], **S.D. Bond**[3], **M. Holst**[4]

[1] Department of Mathematics and Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA
[2] Silicon Clocks, 39141 Civic Center Dr., Suite 450 Fremont, CA 94538, USA
[3] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA
[4] Department of Mathematics, University of California at San Diego, La Jolla, CA 92093, USA

E-mail: [1] burak@cct.lsu.edu, [2] david.h.bernstein@gmail.com, [3] sdbond@uiuc.edu, [4] mholst@sobolev.ucsd.edu

Abstract. The conformal formulation of the Einstein constraint equations is first reviewed, and we then consider the design, analysis, and implementation of adaptive multilevel finite element-type numerical methods for the resulting coupled nonlinear elliptic system. We derive weak formulations of the coupled constraints, and review some new developments in the solution theory for the constraints in the cases of constant mean extrinsic curvature (CMC) data, near-CMC data, and arbitrarily prescribed mean extrinsic curvature data. We then outline some recent results on *a priori* and *a posteriori* error estimates for a broad class of Galerkin-type approximation methods for this system which includes techniques such as finite element, wavelet, and spectral methods. We then use these estimates to construct an adaptive finite element method (AFEM) for solving this system numerically, and outline some new convergence and optimality results. We then describe in some detail an implementation of the methods using the FETK software package, which is an adaptive multilevel finite element code designed to solve nonlinear elliptic and parabolic systems on Riemannian manifolds. We finish by describing a simplex mesh generation algorithm for compact binary objects, and then look at a detailed example showing the use of FETK for numerical solution of the constraints.

Contents

1. Introduction

One of the primary goals of numerical relativity is to compute solutions of the full Einstein equations and to compare the results of such calculations to the expected data from the current or next generation of interferometric gravitational wave observatories such as LIGO. Such observatories are predicted to make observations primarily of "burst" events, mainly supernovae and collisions between compact objects, e.g., black holes and neutron stars. Computer simulation of such events in three space dimensions is a challenging task; a sequence of reviews that give a fairly complete overview of the state of the field of numerical relativity at the time they appeared are [1, 2, 3]. While there is currently tremendous activity in this area of computational science, even the mathematical properties of the Einstein system are still not completely understood; in fact, even the question of which is the most appropriate mathematical formulation of the constrained evolution system for purposes of accurate longtime numerical simulation is still being hotly debated (cf. [4, 5]).

As is well known, solutions to the Einstein equations are constrained in a manner similar to Maxwell's equations, in that the initial data for a particular spacetime must satisfy a set of purely spatial equations which are then preserved throughout the evolution (cf. [6]). As in electromagnetism, these constraint equations may be put in the form of a formally elliptic system; however for the Einstein equations, the resulting system is a set of four coupled nonlinear equations which are generally nontrivial to solve numerically; moreover, the solution theory is only partially understood at the moment (cf. [7, 8, 9]). Until recently, most results were developed only in the case of constant mean extrinsic curvature (CMC) data, leading to a decoupling of the constraints that allows them to be analyzed independently. A complete characterization of CMC solutions on closed manifolds appears in [10]; see also the survey [11] and the recent work on weak solutions [12, 13]. Of the very few non-CMC results available for the coupled system are those for closed manifolds in [7]; these and all other known non-CMC results [14, 15] hold only in the case that the mean extrinsic curvature is nearly constant (referred to as near-CMC). Some very recent results on weak and strong solutions to the constaints in the setting of both CMC and near-CMC solutions on compact manifolds with boundary appear in [16]. Related results on weak and strong solutions to the constaints in the setting of CMC, near-CMC, and truly non-CMC (far-from-CMC) solutions on closed manifolds appear in [17, 18].

Numerous efforts to develop effective numerical techniques for the constraint equations, and corresponding high-performance implementations, have been undertaken over the last twenty years; the previously mentioned numerical relativity reviews [1, 2, 3] give a combined overview of the different discretization and solver technology developed to date. A recent review that focuses entirely on the constraints is [19]; this work reviews the conformal decomposition technique and its more recent incarnations, and also presents an overview of the current state of numerical techniques for the constraints in one of the conformal forms. While most previous work has involved finite difference and spectral techniques, both non-adaptive and adaptive, there have been previous applications of finite element (including adaptive) techniques to the scalar Hamiltonian constraint; these include [20, 21, 22, 23, 24]. An initial approach to the coupled system using adaptive finite element methods appears in [25]. A complete theoretical analysis of adaptive finite element methods (AFEM) for a general class of geometric PDE appears in [26, 27], and a complete analysis of AFEM for the Einstein constraints appears in [28], including proofs of convergence and optimality.

While yielding many useful numerical results and new insights into the Einstein equations, most of the approaches previously used for the constraints suffer from one or more of the following limitations:

- The 3–metric must be conformally flat;
- The extrinsic curvature tensor must be traceless or vanish altogether;
- The momentum constraint must have an exact solution;
- The problem must be spherically symmetric or axisymmetric;
- The domain must be covered by a single coordinate system which may contain singularities;
- The conformal metric must have a scalar curvature which can be easily computed analytically;
- The gauge conditions and/or source terms must be chosen so that the constraints decouple, become linear, or both;
- There is very little control of the approximation error, or even a rigorously derived *a priori* error estimate;
- One must have access to a large parallel computer in order to obtain reasonably accurate solutions on domains of physical interest.

In this paper, we describe a general approach based on adaptive finite element methods that allows one to avoid most of the limitations above.

More precisely, we develop a class of adaptive finite element methods for producing high-quality numerical approximations to solutions of the constraint equations in the setting of general domain topologies and general non-constant mean curvature coupling of the two constraint equations. Our approach is based on the adaptive approximation framework developed in [26, 27, 28], which is a theoretical framework and corresponding implementation for adaptive multilevel finite element methods for producing high-quality reliable approximations to solutions of nonlinear elliptic systems on manifolds. As in earlier work, we employ a 3+1 splitting of spacetime and use the York conformal decomposition formalism to produce a covariant nonlinear elliptic system on a 3-manifold. This elliptic system is then written in weak form, which offers various advantages for developing solution theory, approximation theory, and numerical methods. In particular, this allows us to establish *a priori* error estimates for a broad class of Galerkin-type methods which include not only finite element methods, but also wavelet-based methods as well as spectral methods. Such estimates provide a base approximation theory for establishing convergence of the underlying discretization approach. We also derive *a posteriori* error estimates which can be used to drive adaptive techniques such as local mesh refinement. We outline a class of simplex-based adaptive algorithms for weakly-formulated nonlinear elliptic PDE systems, involving error estimate-driven local mesh refinement.

We then describe in some detail a particular implementation of the adaptive techniques in the software package FETK [26], which is an adaptive multilevel finite element code based on simplex elements. This software is designed to produce provably accurate numerical solutions to a large class of nonlinear covariant elliptic systems of tensor equations on 2- and 3-manifolds in an optimal or nearly-optimal way. It employs *a posteriori* error estimation, adaptive simplex subdivision, unstructured algebraic multilevel methods, global inexact Newton methods, and numerical continuation methods for the highly accurate numerical solution of nonlinear covariant elliptic systems on 2- and 3-manifolds. The FETK implementation has several unusual features (described in Section 6) which make it ideally suited for solving problems such as the Einstein constraint equations in an adaptive way. Applications of FETK to problems in other areas such as biology and elasticity can be found in [29, 30, 31].

Outline of the paper

We review the classical York conformal decomposition in Section 2.2. In Section 2.3 we give a basic framework for deriving weak formulations. In Section 2.4 we briefly outline the notation used for the relevant function spaces. In Section 2.5 we go over a simple weak formulation example. In Section 2.6 we derive an appropriate symmetric weak formulation of the coupled constraint equations, and summarize a number of basic theoretical results. In Section 2.7 we also derive the linearized bilinear form of the nonlinear weak form for use with stability analysis or Newton-like numerical methods. A brief introduction to finite element methods for nonlinear elliptic systems is presented in Section 3.1. Adaptive methods are described in Section 3.3, and residual-type error indicators are derived in Section 3.4. A derivation of the *a posteriori* error indicator for the constraints is give in Section 3.5. We give two give an overview of *a priori* error estimates from [26, 27, 28] for general Galerkin approximations to solutions equations such as the momentum and Hamiltonian constraints in Section 3.2. The numerical methods employed by FETK are described in detail in Section 6, including the finite element discretization, the residual-based *a posteriori* error estimator, the adaptive simplex bisection strategy, the algebraic multilevel solver, and the Newton-based continuation procedure for the solution of the nonlinear algebraic equations which arise. Section 6 describes a mesh generation algorithm for modeling compact binary objects, outlines an algorithm for computing conformal Killing vectors, describes the numerical approximation of the ADM mass, and gives an example showing the use of FETK for solution of the coupled constraints in the setting of a binary compact object collision. The results are summarized in Section 7.

2. The Constraint Equations in General Relativity

2.1. Notation

Let $(\mathcal{M},\gamma_{ab})$ be a connected compact Riemannian d-manifold with boundary $(\partial\mathcal{M},\sigma_{ab})$, where the boundary metric σ_{ab} is inherited from γ_{ab}. In this paper we are interested only in the Riemannian case, where we assume γ_{ab} is strictly positive *a.e.* in $\mathcal{M}$. Later we will require some additional assumptions on γ_{ab} such as its smoothness class. To allow for general boundary conditions, we will view the boundary $(d-1)$-submanifold $\partial\mathcal{M}$ (which we assume to be oriented) as being formed from two disjoint submanifolds $\partial_0\mathcal{M}$ and $\partial_1\mathcal{M}$, i.e.,

$$\partial_0\mathcal{M} \cup \partial_1\mathcal{M} = \partial\mathcal{M}, \qquad \partial_0\mathcal{M} \cap \partial_1\mathcal{M} = \emptyset. \tag{1}$$

When convenient in the discussions below, one of the two submanifolds $\partial_0\mathcal{M}$ or $\partial_1\mathcal{M}$ may be allowed to shrink to zero measure, leaving the other to cover $\partial\mathcal{M}$. An additional technical assumption at times will be non-intersection of the closures of the boundary sets:

$$\overline{\partial_0\mathcal{M}} \cap \overline{\partial_1\mathcal{M}} = \emptyset, \tag{2}$$

This condition is trivially satisfied if either $\partial_0\mathcal{M}$ or $\partial_1\mathcal{M}$ shrinks to zero measure. It is also satisfied in practical situations such as black hole models, where $\partial_0\mathcal{M}$ represents the outer-boundary of a truncated unbounded manifold, and where $\partial_1\mathcal{M}$ represents the inner-boundary at one or more black holes. In any event, in what follows it will usually be necessary to make some minimal smoothness assumptions about the entire boundary submanifold $\partial\mathcal{M}$, such as Lipschitz continuity (for a precise definition see [32]).

We will employ the abstract index notation (cf. [33]) and summation convention for tensor expressions below, with indices running from 1 to d unless otherwise noted. Covariant partial differentiation of a tensor $t^{a_1\cdots a_p}{}_{b_1\cdots b_q}$ using the connection provided by the metric γ_{ab} will be denoted as $t^{a_1\cdots a_p}{}_{b_1\cdots b_q;c}$ or as $D_c t^{a_1\cdots a_p}{}_{b_1\cdots b_q}$. Denoting the outward unit normal to $\partial\mathcal{M}$ as n_b, recall the Divergence Theorem for a vector field w^b on $\mathcal{M}$ (cf. [34]):

$$\int_{\mathcal{M}} w^b{}_{;b}\, dx = \int_{\partial\mathcal{M}} w^b n_b\, ds, \tag{3}$$

where dx denotes the measure on $\mathcal{M}$ generated by the volume element of γ_{ab}:

$$dx = \sqrt{\det\,\gamma_{ab}}\; dx^1 \cdots dx^d, \tag{4}$$

and where ds denotes the boundary measure on $\partial\mathcal{M}$ generated by the boundary volume element of σ_{ab}. Making the choice $w^b = u_{a_1\ldots a_k} v^{a_1\cdots a_k b}$ and forming the divergence ${w^b}_{;b}$ by applying the product rule leads to a useful integration-by-parts formula for certain contractions of tensors:

$$\int_{\mathcal{M}} u_{a_1\ldots a_k} {v^{a_1\ldots a_k b}}_{;b}\, dx = \int_{\partial\mathcal{M}} u_{a_1\ldots a_k} v^{a_1\ldots a_k b} n_b\, ds - \int_{\mathcal{M}} v^{a_1\ldots a_k b} u_{a_1\ldots a_k;b}\, dx. \quad (5)$$

When $k = 0$ this reduces to the familiar case where u and v are scalars.

2.2. The York Decomposition

As discussed in the introduction, the most common form of the initial data problem for numerical work is the coupled elliptic system obtained from the York decomposition [35, 36]. We employ the standard notation whereby the spatial 3–metric is denoted γ_{ab} with indices running from 1 to 3. The Hamiltonian and momentum constraints are

$$R + (\mathrm{tr}K)^2 - K_{ab}K^{ab} = 16\pi\rho \quad (6)$$

and

$$D_b(K^{ab} - \gamma^{ab}\mathrm{tr}K) = 8\pi j^a, \quad (7)$$

where R is the scalar curvature of γ_{ab}, K_{ab} is the extrinsic curvature of the initial hypersurface, and ρ and j^a are the mass density and current.

As is well known, (6) and (7) form an under-determined system for the components of the 3–metric and extrinsic curvature. The York conformal decomposition method identifies some of these components as "freely specifiable," i.e., source terms similar to the matter terms ρ and j^a, and the remaining components as constrained. The main strength of the formalism is that it does this in such a way that the constraint system becomes a coupled set of nonlinear elliptic equations in these components. In only the most general setting do the fully coupled equations have to be solved; in many applications they can be decoupled and further simplification of the problem may eliminate one or more of the non-linearities. The primary disadvantage of the formalism is the difficulty of controlling the physics of the initial data generated; the matter terms as well as the unconstrained parts of γ_{ab} and K_{ab} are not related directly to important physical and geometrical properties of the initial data. One therefore generally relies on an evolution, i.e., a computation of the entire spacetime, to determine "what was in" the initial data to begin with. Some of the difficulties with use of the formalism have been recently overcome; cf. [19] for a recent survey.

The formalism is summarized in a number of places (see, e.g., [35, 36, 19]), in this section we give only a brief description of the most standard form. The manifold $\mathcal{M}$ is endowed with a Riemannian metric $\hat{\gamma}_{ab}$ and the solution to the initial data problem is, in part, a metric γ_{ab} related to $\hat{\gamma}_{ab}$ by

$$\gamma_{ab} = \phi^4\hat{\gamma}_{ab}. \quad (8)$$

In what follows all hatted quantities are formed out of $\hat{\gamma}_{ab}$ in the usual way, e.g., the covariant derivative $\hat{D}_a$, the Riemann tensor $\hat{R}_{abcd}$, etc., while unhatted quantities are formed out of γ_{ab}.

There are four constraint equations and 12 components of γ_{ab} and K_{ab}. The eight freely specifiable components consist of the conformal part of the 3–metric and the trace and the transverse-traceless parts of the extrinsic curvature. The remaining constrained parts are the conformal factor, ϕ, and the "longitudinal potential," W^a, of the extrinsic curvature. Specifically, we decompose the extrinsic curvature tensor as

$$K^{ab} = \phi^{-10}({}^*\hat{A}^{ab} + (\hat{L}W)^{ab}) + \frac{1}{3}\phi^{-4}\hat{\gamma}^{ab}\mathrm{tr}K, \tag{9}$$

$$A^{ab} = K^{ab} - \frac{1}{3}\gamma^{ab}\mathrm{tr}K = \phi^{-10}\hat{A}^{ab}, \tag{10}$$

$$\hat{A}^{ab} = {}^*\hat{A}^{ab} + (\hat{L}W)^{ab}, \tag{11}$$

$$(\hat{L}W)^{ab} = \hat{D}^aW^b + \hat{D}^bW^a - \frac{2}{3}\hat{\gamma}^{ab}\hat{D}_cW^c, \tag{12}$$

where $\mathrm{tr}K = \gamma^{ab}K_{ab}$. Note that $\hat{A}^{ab}$ and $(\hat{L}W)^{ab}$ are traceless by construction and ${}^*\hat{A}^{ab}$ is a freely specifiable transverse-traceless tensor. The matter terms are decomposed as

$$\rho = \hat{\rho}\phi^{-8}, \quad j^a = \hat{j}^a\phi^{-10}. \tag{13}$$

In these variables the Hamiltonian constraint (6) becomes

$$\hat{\Delta}\phi = \frac{1}{8}\hat{R}\phi + \frac{1}{12}(\mathrm{tr}K)^2\phi^5 - \frac{1}{8}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3}, \tag{14}$$

where $\hat{\Delta}\phi = \hat{D}_a\hat{D}^a\phi$ and where $(T_{ab})^2 = T^{ab}T_{ab}$, following the notation introduced in (34). In these variables the momentum constraint (7) becomes

$$\hat{D}_b(\hat{L}W)^{ab} = \frac{2}{3}\phi^6\hat{D}^a\mathrm{tr}K + 8\pi\hat{j}^a. \tag{15}$$

In this article, we are mainly interested in formulations of the constraints on manifolds with boundary. The primary motivation for this is the desire to develop techniques for producing high-quality numerical solutions to the constraints, which require the use of mathematical formulations of the constraints involving finite domains with boundary. In order to completely specify the strong (and later, the weak) forms of the constraints on manifolds with boundary, we need to specify the boundary conditions. This is very problem dependent; however, we would like to at least include the case of the vector Robin condition for asymptotically flat initial data given in [37]. This is

$$(\hat{L}W)^{bc}\hat{n}_c\left(\delta^a_b - \frac{1}{2}\hat{n}^a\hat{n}_b\right) + \frac{6}{7R}W^b\left(\delta^a_b - \frac{1}{8}\hat{n}^a\hat{n}_b\right) = O(R^{-3}) \tag{16}$$

where R is the radius of a large, spherical domain. The right hand side could be taken to be zero. Noting that $\delta^a_b + \hat{n}^a\hat{n}_b$ is the inverse of $\delta^a_b - 1/2\ \hat{n}^a\hat{n}_b$ we may compute

$$(\hat{L}W)^{ab}\hat{n}_b + \frac{6}{7R}W^b\left(\delta^a_b + \frac{3}{4}\hat{n}^a\hat{n}_b\right) = 0. \tag{17}$$

Hence we will consider the following linear Robin-like condition, which is general enough to include (17) and more recently proposed boundary conditions [38]:

$$(\hat{L}W)^{ab}\hat{n}_b + C^a{}_bW^b = Z^a \quad \text{on } \partial_1\mathcal{M}. \tag{18}$$

Similarly, we are interested in analyzing the case of a Robin-like boundary condition with the Hamiltonian constraint:

$$\hat{n}_a\hat{D}^a\phi + c\phi = z \quad \text{on } \partial_1\mathcal{M}. \tag{19}$$

Equations (14)–(15) are known to be well-posed only for restricted problem data and manifold topologies [39, 10, 40, 41, 42, 37, 43, 7, 44, 8, 9]; most of the existing results are for the case of constant mean extrinsic curvature (CMC) data on closed (compact without boundary) manifolds, with some results for near-CMC data [7]. Some very recent results on weak and strong solutions to the constaints in the setting of CMC and near-CMC solutions on compact manifolds with boundary appear in [16]. Related results on weak and strong solutions to the constaints in the setting of CMC, near-CMC, and truly non-CMC (far from-CMC) solutions on closed manifolds appear in [17, 18].

2.3. General Weak Formulations of Nonlinear Elliptic Systems

Consider now a general second-order elliptic system of tensor equations in strong divergence form over a Riemannian manifold $\mathcal{M}$ with boundary:

$$\begin{aligned} -A^{ia}(x^b, u^j, u^k{}_{;c})_{;a} + B^i(x^b, u^j, u^k{}_{;c}) &= 0 \ \text{ in } \mathcal{M}, && (20)\\ A^{ia}(x^b, u^j, u^k{}_{;c}) n_a + C^i(x^b, u^j, u^k{}_{;c}) &= 0 \ \text{ on } \partial_1\mathcal{M}, && (21)\\ u^i(x^b) &= E^i(x^b) \ \text{ on } \partial_0\mathcal{M}, && (22) \end{aligned}$$

where

$$\begin{aligned} &1 \le a,b,c \le d, \quad 1 \le i,j,k \le n, && (23)\\ &A : \mathcal{M} \times \mathbb{R}^n \times \mathbb{R}^{nd} \mapsto \mathbb{R}^{nd}, \quad B : \mathcal{M} \times \mathbb{R}^n \times \mathbb{R}^{nd} \mapsto \mathbb{R}^n, && (24)\\ &C : \partial_1\mathcal{M} \times \mathbb{R}^n \times \mathbb{R}^{nd} \mapsto \mathbb{R}^n, \quad E : \partial_0\mathcal{M} \times \mapsto \mathbb{R}^n. && (25) \end{aligned}$$

The divergence-form system (20)–(22), together with the boundary conditions, can be viewed as an operator equation

$$F(u) = 0, \qquad F : \mathcal{B}_1 \mapsto \mathcal{B}_2^*, \tag{26}$$

for some Banach spaces $\mathcal{B}_1$ and $\mathcal{B}_2$, where $\mathcal{B}_2^*$ denotes the dual space of $\mathcal{B}_2$.

Our interest here is primarily in coupled systems of one or more scalar field equations and one or more d-vector field equations. The unknown n-vector u^i then in general consists of n_s scalars and n_v d-vectors, so that $n = n_s + n_v \cdot d$. To allow the n-component system (20)–(22) to be treated notationally as if it were a single n-vector equation, it will be convenient to introduce the following notation for the unknown vector u^i and for the metric of the product space of scalar and vector components of u^i:

$$\mathcal{G}_{ij} = \begin{bmatrix} \gamma_{ab}^{(1)} & & 0 \\ & \ddots & \\ 0 & & \gamma_{ab}^{(n_e)} \end{bmatrix}, \qquad u^i = \begin{bmatrix} u^a_{(1)} \\ \vdots \\ u^a_{(n_e)} \end{bmatrix}, \qquad n_e = n_s + n_v. \tag{27}$$

If $u^a_{(k)}$ is a d-vector we take $\gamma^{(k)}_{ab} = \gamma_{ab}$; if $u^a_{(k)}$ is a scalar we take $\gamma^{(k)}_{ab} = 1$.

The weak form of (20)–(22) is obtained by taking the L^2-inner-product between a vector v^j (vanishing on $\partial_0\mathcal{M}$) lying in a product space of scalars and tensors, and the residual of the tensor system (20), yielding:

$$\int_{\mathcal{M}} \mathcal{G}_{ij} \left(B^i - A^{ia}{}_{;a} \right) v^j \ dx = 0. \tag{28}$$

Due to the definition of $\mathcal{G}_{ij}$ in (27), this is simply a sum of integrals of scalars, each of which is a contraction of the type appearing on the left side in (5). Using then (5) and (21) together in (28), and recalling that $v^i = 0$ on $\partial_0\mathcal{M}$, yields

$$\int_{\mathcal{M}} \mathcal{G}_{ij} A^{ia} v^j{}_{;a} \ dx + \int_{\mathcal{M}} \mathcal{G}_{ij} B^i v^j \ dx + \int_{\partial_1\mathcal{M}} \mathcal{G}_{ij} C^i v^j \ ds = 0. \tag{29}$$

This gives rise to a covariant weak formulation of the problem:

$$\text{Find } u \in \bar{u} + \mathcal{B}_1 \text{ s.t. } \langle F(u), v \rangle = 0, \quad \forall\, v \in \mathcal{B}_2, \tag{30}$$

for suitable Banach spaces of functions $\mathcal{B}_1$ and $\mathcal{B}_2$, where the nonlinear weak form $\langle F(\cdot), \cdot \rangle$ can be written as:

$$\langle F(u), v \rangle = \int_{\mathcal{M}} \mathcal{G}_{ij}(A^{ia} v^j{}_{;a} + B^i v^j)\ dx + \int_{\partial_1 \mathcal{M}} \mathcal{G}_{ij} C^i v^j\ ds. \tag{31}$$

The notation $\langle w, v \rangle$ will represent the duality pairing of a function v in a Banach space $\mathcal{B}$ with a bounded linear functional (or *form*) w in the dual space $\mathcal{B}^*$. Depending on the particular function spaces involved, the pairing may be thought of as coinciding with the L^2-inner-product through the Riesz Representation Theorem [45]. The affine shift tensor $\bar{u}$ in (30) represents the essential or Dirichlet part of the boundary condition if there is one; the existence of $\bar{u}$ such that $E = \bar{u}|_{\partial_0 \mathcal{M}}$ in the sense of the Trace operator is guaranteed by the Trace Theorem for Sobolev spaces on manifolds with boundary [46], as long as E^i in (22) and $\partial_0 \mathcal{M}$ are smooth enough (see §2.4 below).

The (Gateaux) linearization $\langle DF(u)w, v \rangle$ of the nonlinear form $\langle F(u), v \rangle$, necessary for both local solvability analysis and Newton-like numerical methods (cf. [26]), is defined formally as:

$$\langle DF(u)w, v \rangle = \frac{d}{d\epsilon} \langle F(u + \epsilon w), v \rangle \bigg|_{\epsilon = 0}. \tag{32}$$

This form is easily computed from most nonlinear forms $\langle F(u), v \rangle$ which arise from second order nonlinear elliptic problems, although the calculation can be tedious in some cases (as we will see shortly in the case of the constraints).

The Banach spaces which arise naturally as solution spaces for the class of nonlinear elliptic systems in (30) are product spaces of the Sobolev spaces $W^{k,p}_{0,D}(\mathcal{M})$, or the related Besov spaces $B^{k,p}_q(\mathcal{M})$. This is due to the fact that under suitable growth conditions on the nonlinearities in F, it can be shown (essentially by applying the Hölder inequality) that there exists p_k, q_k, r_k satisfying $1 < p_k, q_k, r_k < \infty$ such that the choice

$$\mathcal{B}_1 = W^{1,r_1}_{0,D}(\mathcal{M}) \times \cdots \times W^{1,r_{n_e}}_{0,D}(\mathcal{M}), \qquad \mathcal{B}_2 = W^{1,q_1}_{0,D}(\mathcal{M}) \times \cdots \times W^{1,q_{n_e}}_{0,D}(\mathcal{M}),$$

$$\frac{1}{p_k} + \frac{1}{q_k} = 1, \qquad r_k \geq \min\{p_k, q_k\}, \qquad k = 1, \ldots, n_e, \tag{33}$$

ensures $\langle F(u), v \rangle$ in (31) remains finite for all arguments [47, 26].

2.4. The Sobolev Spaces $W^{k,p}(\mathcal{M})$ and $H^k(\mathcal{M})$ on Manifolds with Boundary

For a type (r, s)-tensor $T^I{}_J = T^{a_1 a_2 \cdots a_r}{}_{b_1 b_2 \cdots b_s}$, where I and J are (tensor) multi-indices satisfying $|I| = r$, $|J| = s$, define

$$|T^I{}_J| = \left(T^I{}_J T^L{}_M \gamma_{IL} \gamma^{JM} \right)^{1/2}. \tag{34}$$

Here, γ_{IJ} and γ^{IJ} are generated from the Riemannian d-metric γ_{ab} on $\mathcal{M}$ as:

$$\gamma_{IJ} = \gamma_{ab}\gamma_{cd} \cdots \gamma_{pq}, \qquad \gamma^{IJ} = \gamma^{ab}\gamma^{cd} \cdots \gamma^{pq}, \tag{35}$$

where $|I| = |J| = m$, producing m terms in each product. This is just an extension of the Euclidean l^2-norm for vectors in $\mathbb{R}^d$. For example, in the case of a 3-manifold, taking $|I| = 1$, $|J| = 0$, $\gamma_{ab} = \delta_{ab}$, gives $|K^I{}_J| = |K^a| = \left(K^a K^b \gamma_{ab}\right)^{1/2} = \left(K^a K^b \delta_{ab}\right)^{1/2} = \|K^a\|_{l^2(\mathbb{R}^3)}$.

Employing the measure dx on $\mathcal{M}$ defined in (4), the L^p-norm of a tensor on $\mathcal{M}$, $p \in [1, \infty)$, is defined as:

$$\|T^I{}_J\|_{L^p(\mathcal{M})} = \left(\int_{\mathcal{M}} |T^I{}_J|^p \; dx\right)^{1/p}, \qquad \|T^I{}_J\|_{L^\infty(\mathcal{M})} = \text{ess} \sup_{x\in\mathcal{M}} |T^I{}_J(x)|. \tag{36}$$

The resulting L^p-spaces for $1 \le p \le \infty$ are denoted as:

$$L^p(\mathcal{M}) = \left\{ \; T^I{}_J \;\; : \;\; \|T^I{}_J\|_{L^p(\mathcal{M})} < \infty \; \right\}. \tag{37}$$

Covariant (distributional) differentiation of order $m = |L|$ (for some tensor multi-index L) using a connection generated by γ_{ab}, or generated by possibly a different metric, will be denoted as either of:

$$D^m T^I{}_J = T^I{}_{J;L}, \tag{38}$$

where m should not be confused with a tensor index. The Sobolev semi-norm of a tensor is defined through (37) as:

$$|T^I{}_J|^p_{W^{m,p}(\mathcal{M})} = \sum_{|L|=m} \|T^I{}_{J;L}\|^p_{L^p(\mathcal{M})}, \tag{39}$$

and the Sobolev norm is subsequently defined as:

$$\|T^I{}_J\|_{W^{k,p}(\mathcal{M})} = \left(\sum_{0\le m\le k} |T^I{}_J|^p_{W^{m,p}(\mathcal{M})}\right)^{1/p}. \tag{40}$$

The resulting Sobolev spaces of tensors are then defined as:

$$W^{k,p}(\mathcal{M}) = \left\{ \; T^I{}_J \;\; : \;\; \|T^I{}_J\|_{W^{k,p}(\mathcal{M})} < \infty \; \right\},$$

$$W^{k,p}_0(\mathcal{M}) = \left\{ \; \text{Completion of } C^\infty_0(\mathcal{M}) \text{ w.r.t. } \|\cdot\|_{W^{k,p}(\mathcal{M})} \; \right\}, \tag{41}$$

where $C^\infty_0(\mathcal{M})$ is the space of C^∞-tensors with compact support in $\mathcal{M}$. The space $W^{k,p}_0(\mathcal{M})$ is a special case of $W^{k,p}_{0,D}(\mathcal{M})$, which can be characterized as:

$$W^{k,p}_{0,D}(\mathcal{M}) = \left\{ \; T^I{}_J \in W^{k,p} \;\; : \;\; \text{tr } T^I{}_{J;L} = 0 \text{ on } \partial_0\mathcal{M}, \; |L| \le k-1 \; \right\}. \tag{42}$$

The spaces $W^{k,p}(\mathcal{M})$ and $W^{k,p}_{0,D}(\mathcal{M})$ are separable ($1 \le p < \infty$) and reflexive ($1 < p < \infty$) Banach spaces. The dual space of bounded linear functionals on $W^{k,p}(\mathcal{M})$ can be shown (in the sense of distributions, cf. [48]) to be $W^{-k,q}(\mathcal{M})$, $1/p + 1/q = 1$, which is itself a Banach space when equipped with the dual norm:

$$\|f\|_{W^{-k,q}(\mathcal{M})} = \sup_{0\ne u\in W^{k,p}(\mathcal{M})} \frac{|f(u)|}{\|u\|_{W^{k,p}(\mathcal{M})}}, \qquad \frac{1}{p} + \frac{1}{q} = 1. \tag{43}$$

The Hilbert space special case of $p = 2$ is given a simplified notation:

$$H^k(\mathcal{M}) = W^{k,2}(\mathcal{M}), \qquad H^{-k}(\mathcal{M}) = W^{-k,2}(\mathcal{M}), \tag{44}$$

with the same convention used for the various subspaces of $H^k(\mathcal{M})$ such as $H^k_0(\mathcal{M})$ and $H^k_{0,D}(\mathcal{M})$. The norm on $H^k(\mathcal{M})$ defined above is induced by an L^2-based inner-product as follows: $\|T^I{}_J\|_{H^k(\mathcal{M})} = (T^I{}_J, T^I{}_J)^{1/2}_{H^k(\mathcal{M})}$, where

$$(T^I{}_J, S^I{}_J)_{L^2(\mathcal{M})} = \int_{\mathcal{M}} T^I{}_J S^L{}_M \gamma_{IL}\gamma^{JM} \; dx, \tag{45}$$

and where

$$(T^I{}_J, S^I{}_J)_{H^k(\mathcal{M})} = \sum_{0\le m\le k} (D^m T^I{}_J, D^m S^I{}_J)_{L^2(\mathcal{M})}. \tag{46}$$

The Banach spaces $W^{k,p}$ and their various subspaces satisfy various relations among themselves, with the L^p-spaces, and with classical function spaces such as C^k and $C^{k,\alpha}$. These relationships are characterized by a collection of results known as embedding, compactness, density, trace, and related theorems. A number of these are fundamental to approximation theory for elliptic equations, and will be recalled when needed below.

2.5. Weak Formulation Example

Before we derive a weak formulation of the Einstein constraints, let us consider a simple example to illustrate the idea. Here we assume the 3–metric to be flat so that ∇ is the ordinary gradient operator and $\cdot$ the usual inner product. Let $\mathcal{M}$ represent the unit sphere centered at the origin (with a single chart inherited from the canonical Cartesian coordinate system in $\mathbb{R}^3$), and let $\partial\mathcal{M}$ denote the boundary, satisfying the boundary assumption in equation (1). Consider now the following semilinear equation on $\mathcal{M}$:

$$\begin{aligned} -\nabla\cdot(a(x)\nabla u(x)) + b(x,u(x)) &= 0 \ \text{ in } \mathcal{M}, && (47)\\ n(x)\cdot(a(x)\nabla u(x)) + c(x,u(x)) &= 0 \ \text{ on } \partial_1\mathcal{M}, && (48)\\ u(x) &= f(x) \ \text{ on } \partial_0\mathcal{M}, && (49) \end{aligned}$$

where $n(x) : \partial\mathcal{M} \mapsto \mathbb{R}^d$ is the unit normal to $\partial\mathcal{M}$, and where

$$\begin{aligned} a &: \mathcal{M} \mapsto \mathbb{R}^{3\times3}, \qquad b : \mathcal{M}\times\mathbb{R} \mapsto \mathbb{R}, && (50)\\ c &: \partial_1\mathcal{M}\times\mathbb{R} \mapsto \mathbb{R}, \quad f : \partial_0\mathcal{M} \mapsto \mathbb{R}. && (51) \end{aligned}$$

To produce a weak formulation, we first multiply by a test function $v \in H^1_{0,D}(\mathcal{M})$ (the subspace of $H^1(\mathcal{M})$ which vanishes on the Dirichlet portion of the boundary $\partial_0\mathcal{M}$), producing:

$$\int_{\mathcal{M}} (-\nabla\cdot(a\nabla u) + b(x,u))\, v\ dx = 0. \tag{52}$$

After applying the flat space version of the divergence theorem, this becomes:

$$\int_{\mathcal{M}} (a\nabla u)\cdot\nabla v\ dx - \int_{\partial\mathcal{M}} v(a\nabla u)\cdot n\ ds + \int_{\mathcal{M}} b(x,u)v\ dx = 0. \tag{53}$$

The boundary integral is reformulated using the boundary conditions as follows:

$$\int_{\partial\mathcal{M}} v(a\nabla u)\cdot n\ ds = -\int_{\partial_1\mathcal{M}} c(x,u)v\ ds. \tag{54}$$

If the boundary function f is regular enough so that $f \in H^{1/2}(\partial_0\mathcal{M})$, then from the Trace Theorem [32], there exists $\bar{u} \in H^1(\mathcal{M})$ such that $f = \bar{u}|_{\partial_0\mathcal{M}}$ in the sense of the Trace operator. Employing such a function $\bar{u} \in H^1(\mathcal{M})$, the weak formulation has the form:

$$\text{Find } u \in \bar{u} + H^1_{0,D}(\mathcal{M}) \ \text{ s.t. } \langle F(u), v\rangle = 0, \quad \forall\ v \in H^1_{0,D}(\mathcal{M}), \tag{55}$$

where from equations (53) and (54), the nonlinear form is defined as:

$$\langle F(u), v\rangle = \int_{\mathcal{M}} (a\nabla u\cdot\nabla v + b(x,u)v)\ dx + \int_{\partial_1\mathcal{M}} c(x,u)v\ ds. \tag{56}$$

The “weak” formulation of the problem given by equation (55) imposes only one order of differentiability on the solution u, and only in the weak or distributional sense. Under suitable growth conditions on the nonlinearities b and c, it can be shown that this weak formulation makes sense, in that the form $\langle F(\cdot), \cdot\rangle$ is finite for all arguments.

To analyze linearization stability, or to apply a numerical algorithm such as Newton's method, we will need the bilinear linearization form $\langle DF(u)w, v\rangle$, produced as the formal Gateaux derivative of the nonlinear form $\langle F(u), v\rangle$:

$$\langle DF(u)w, v\rangle = \frac{d}{d\epsilon}\langle F(u+\epsilon w), v\rangle\Big|_{\epsilon=0}$$

$$= \frac{d}{d\epsilon}\left(\int_{\mathcal{M}} (a\nabla(u+\epsilon w)\cdot\nabla v + b(x, u+\epsilon w)v)\, dx + \int_{\partial_1\mathcal{M}} c(x, u+\epsilon w)v ds\right)\Bigg|_{\epsilon=0}$$

$$= \int_{\mathcal{M}} \left(a\nabla w\cdot\nabla v + \frac{\partial b(x,u)}{\partial u} wv\right)\, dx + \int_{\partial_1\mathcal{M}} \frac{\partial c(x,u)}{\partial u} wv\ ds. \tag{57}$$

Now that the nonlinear weak form $\langle F(u), v\rangle$ and the associated bilinear linearization form $\langle DF(u)w, v\rangle$ are defined as integrals, they can be evaluated using numerical quadrature to assemble a Galerkin-type discretization; this is described in some detail below in the case of a finite element-based Galerkin method.

2.6. Weak Formulation of the Constraints

The Hamiltonian constraint (14) as well as the momentum constraint (15), taken separately or as a system, fall into the class of second-order divergence-form elliptic systems of tensor equations in (20)–(22). Therefore, we will follow the same plan as in §2.4 in order to produce the weak formulation (30)–(31). However, we now employ the conformal metric $\hat{\gamma}_{ab}$ from the preceding section to define the volume element $dx = \sqrt{\det\,\hat{\gamma}_{ab}}\ dx^1dx^2dx^3$ and the corresponding boundary volume element ds, and for use as the manifold connection for covariant differentiation. The notation for covariant differentiation using the conformal connection will be denoted $\hat{D}_a$ as in the previous section, and the various quantities from §2.4 will now be hatted to denote our use of the conformal metric. For example, the unit normal to $\partial\mathcal{M}$ will now be denoted $\hat{n}^a$.

Consider now the principle parts of the Hamiltonian and momentum constraint operators of the previous section:

$$\hat{\Delta}\phi = \hat{D}_a\hat{D}^a\phi, \qquad \hat{D}_b(\hat{L}W)^{ab} = \hat{D}_b(\hat{D}^aW^b + \hat{D}^bW^a - \frac{2}{3}\hat{\gamma}^{ab}\hat{D}_cW^c). \tag{58}$$

Employing the covariant divergence theorem in equation (5) leads to covariant versions of the Green identities

$$\int_{\mathcal{M}} \psi\hat{\Delta}\phi\ dx + \int_{\mathcal{M}} (\hat{D}_a\phi)(\hat{D}^a\psi)\ dx = \int_{\partial\mathcal{M}} \hat{n}_a\psi\hat{D}^a\phi\ ds \tag{59}$$

and

$$\int_{\mathcal{M}} V_a\hat{D}_b(\hat{L}W)^{ab}\ dx + \int_{\mathcal{M}} (\hat{L}W)^{ab}\hat{D}_bV_a\ dx = \int_{\partial\mathcal{M}} \hat{n}_bV_a(\hat{L}W)^{ab}\ ds, \tag{60}$$

for smooth functions in $C^\infty(\mathcal{M})$. These identities extend to $W^{1,p}(\mathcal{M})$ using a standard approximation argument, since $C^\infty(\mathcal{M})$ is dense in $W^{1,p}(\mathcal{M})$ (cf. Theorem 2.9 in [49]).

Due to the symmetries of $(\hat{L}W)^{ab}$ and $\hat{\gamma}^{ab}$, the second integrand in (60) can be rewritten in a completely symmetric form. To do so, we first borrow the linear strain or (symmetrized) deformation operator from elasticity:

$$(\hat{E}V)^{ab} = \frac{1}{2}\left(\hat{D}^b V^a + \hat{D}^a V^b\right). \tag{61}$$

We can then write the operator $(\hat{L}W)^{ab}$ in terms of $(\hat{E}W)^{ab}$ as follows:

$$(\hat{L}W)^{ab} = 2\mu(\hat{E}W)^{ab} + \lambda\hat{\gamma}^{ab}\hat{D}_c W^c, \qquad \text{with } \mu = 1 \text{ and } \lambda = -\frac{2}{3}. \tag{62}$$

This makes it clear that the momentum constraint operator $\hat{D}_b(\hat{L}W)^{ab}$ is a covariant version of the linear elasticity operator for a homogeneous isotropic material (cf. [50]), for a particular choice of Lamé constants. In particular, in the flat space case where $\hat{\gamma}_{ab} \equiv \delta_{ab}$, the operator becomes the usual linear elasticity operator:

$$\hat{D}_b(\hat{L}W)^{ab} \to \left(2\mu e_{ab}(W) + \lambda e_{cc}(W)\delta_{ab}\right)_{,b}, \quad e_{ab}(W) = \frac{1}{2}\left(W_{a,b} + W_{b,a}\right), \tag{63}$$

where $W_{a,b} = \partial W_a/\partial x^b$ denotes regular partial differentiation. Employing the operator $(\hat{E}W)^{ab}$ leads to a symmetric expression in the Green identity (60):

$$\begin{aligned}
(\hat{L}W)^{ab}\hat{D}_b V_a &= \frac{1}{2}\left((\hat{L}W)^{ab}\hat{D}_b V_a + (\hat{L}W)^{ab}\hat{D}_a V_b\right) && (64)\\
&= (\hat{L}W)^{ab}(\hat{E}V)_{ab} && (65)\\
&= 2\mu(\hat{E}W)^{ab}(\hat{E}V)_{ab} + \frac{1}{2}\lambda\hat{\gamma}^{ab}\hat{D}_c W^c\left(\hat{D}_b V_a + \hat{D}_a V_b\right) && (66)\\
&= 2\mu(\hat{E}W)^{ab}(\hat{E}V)_{ab} + \lambda\hat{D}_a W^a \hat{D}_b V^b. && (67)
\end{aligned}$$

While it is clear by inspection that the first operator in (58) is formally self-adjoint with respect to the covariant L^2-inner-product defined in (45), reversing the procedure in (67) implies that the same is true for the second operator in (58). In other words, the following holds (ignoring the boundary terms):

$$\begin{aligned}
(\hat{\Delta}\phi, \psi)_{L^2(\mathcal{M})} &= (\phi, \hat{\Delta}\psi)_{L^2(\mathcal{M})}, && \forall \phi, \psi, && (68)\\
(\hat{D}_b(\hat{L}W)^{ab}, V^a)_{L^2(\mathcal{M})} &= (W^a, \hat{D}_b(\hat{L}V)^{ab})_{L^2(\mathcal{M})}, && \forall W^a, V^a. && (69)
\end{aligned}$$

To make it possible to write the Hamiltonian constraint and various related equations in a concise way, we now introduce the following nonlinear function $P(\phi) = P(\phi, W^a, x^b)$, where the explicit dependence on x^b (and sometimes also the dependence on W^a) is suppressed to simplify the notation:

$$P(\phi) = \frac{1}{16}\hat{R}\phi^2 + \frac{1}{72}(\mathrm{tr}K)^2\phi^6 + \frac{1}{48}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-6} + \pi\hat{\rho}\phi^{-2}. \tag{70}$$

The first and second functional derivatives with respect to ϕ are then as follows:

$$\begin{aligned}
P'(\phi) &= \frac{1}{8}\hat{R}\phi + \frac{1}{12}(\mathrm{tr}K)^2\phi^5 - \frac{1}{8}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3}, && (71)\\
P''(\phi) &= \frac{1}{8}\hat{R} + \frac{5}{12}(\mathrm{tr}K)^2\phi^4 + \frac{7}{8}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-8} + 6\pi\hat{\rho}\phi^{-4}. && (72)
\end{aligned}$$

When working with the Hamiltonian constraint separately, we will often use these expressions involving $P(\cdot)$; when working with the coupled system, we will usually write the polynomial explicitly in order to indicate the coupling terms.

We now take the inner product of (14) with a test function ψ, which we assume to vanish on $\partial_0\mathcal{M}$. (Again, $\partial_0\mathcal{M}$ may have zero measure, or it may be all or only a piece of $\partial\mathcal{M}$.) After use of the Green identity (59) and the Robin boundary condition (19) we obtain the following form $\langle F_{\rm H}(\phi),\psi\rangle$ (nonlinear in ϕ, but linear in ψ) for use in the weak formulation in (30):

$$\langle F_{\rm H}(\phi),\psi\rangle = \int_{\partial_1\mathcal{M}} (c\phi - z)\psi \ ds + \int_{\mathcal{M}} P'(\phi)\psi \ dx + \int_{\mathcal{M}} \hat{D}_a\phi\hat{D}^a\psi \ dx. \quad (73)$$

For the momentum constraint we take the inner product of (15) with respect to a test vector field V^a (again assumed to vanish on $\partial_0\mathcal{M}$) and similarly use the Green identity (60) and the Robin condition (18) to obtain a form $\langle F_{\rm M}(W^a),V^a\rangle$ (linear in both W^a and V^a in this case) having the expression

$$\langle F_{\rm M}(W^a),V^a\rangle = \int_{\partial_1\mathcal{M}} \left(C^a{}_bW^b - Z^a\right)V_a \ ds + \int_{\mathcal{M}} \left(\frac{2}{3}\phi^6\hat{D}^a{\rm tr}K + 8\pi\hat{j}^a\right)V_a \ dx$$
$$+ \int_{\mathcal{M}} \left(2\mu(\hat{E}W)^{ab}(\hat{E}V)_{ab} + \lambda\hat{D}_aW^a\hat{D}_bV^b\right) \ dx, \quad (74)$$

where we have used (67). We will take $\mu = 1$ and $\lambda = -2/3$ in (74), but for the moment we will leave them unspecified for purposes of the discussion below.

Ordering the Hamiltonian constraint first in the system (20), and defining the product metric $\mathcal{G}_{ij}$ and the vectors u^i and v^j appearing in (27) and (31) as:

$$\mathcal{G}_{ij} = \left[\begin{array}{cc} 1 & 0 \\ 0 & \gamma_{ab} \end{array}\right], \qquad u^i = \left[\begin{array}{c} \phi \\ W^a \end{array}\right], \qquad v^j = \left[\begin{array}{c} \psi \\ V^b \end{array}\right], \quad (75)$$

produces a single nonlinear weak form for the coupled constraints in the form required in (30), where

$$\langle F(u),v\rangle = \langle F([\phi,W^a]),[\psi,V^a]\rangle = \langle F_{\rm H}(\phi),\psi\rangle + \langle F_{\rm M}(W^a),V^a\rangle$$
$$= \int_{\partial_1\mathcal{M}} \left([c\phi - z]\,\psi + \left[C^a{}_bW^b - Z^a\right]V_a\right) \ ds + \int_{\mathcal{M}} \left(\frac{2}{3}\phi^6\hat{D}^a{\rm tr}K + 8\pi\hat{j}^a\right)V_a \ dx$$
$$+ \int_{\mathcal{M}} \left(\frac{1}{8}\hat{R}\phi + \frac{1}{12}({\rm tr}K)^2\phi^5 - \frac{1}{8}({}^*\!\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3}\right)\psi \ dx$$
$$+ \int_{\mathcal{M}} \left(\hat{D}_a\phi\hat{D}^a\psi + 2\mu(\hat{E}W)^{ab}(\hat{E}V)_{ab} + \lambda\hat{D}_aW^a\hat{D}_bV^b\right) \ dx. \quad (76)$$

While we have completely specified the weak form of the separate and coupled constraints on a manifold with boundary in a formal sense, they can be shown to be well-defined (and individually well-posed) in a more precise mathematical sense; see [16, 17, 18] for an analysis and a survey of the collection of existence, uniqueness, and stability results. For a particular situation, we must specify the particular combination of the boundary conditions (20)–(21) on a splitting of the boundary $(\partial\mathcal{M})$ into Dirichlet $(\partial_0\mathcal{M})$ and Robin $(\partial_1\mathcal{M})$ parts. This is quite problem dependent; in numerical simulation one typically computes solutions to the constraints in the interior of a large box or sphere. On the surface of the sphere one employs Robin and vector Robin conditions similar to those given in [37], which fit the framework in (18) and (19). In addition, one often constructs black holes topologically by requiring the conformal metric to obey an isometry through one or more smaller non-overlapping spheres internal to the domain boundary. The isometry generates a boundary condition on the conformal factor which is well understood only when $\hat{\gamma}_{ab}$ is flat. Even in this case, the exact corresponding boundary condition on W^a is not known, but is likely to appear in the form of (18). Solvability of both constraints rests delicately on the boundary condition choices made.

2.7. Gateaux Linearization of the Weak Formulation

We will take the formal Gateaux-derivative of the nonlinear form $\langle F(\cdot), \cdot\rangle$ in equation (76) above, to produce a linearization form for use in local solvability analysis through the Implicit Function Theorem, and for use in Newton-like iterative solution methods (cf. [26]). Defining an arbitrary variation direction $w = [\xi, X^a]$, we compute the Gateaux-derivative of the nonlinear form as follows:

$$\langle DF([\phi, W^a])[\xi, X^a], [\psi, V^a]\rangle = \frac{d}{d\epsilon}\langle F([\phi + \epsilon\xi, W^a + \epsilon X^a]), [\psi, V^a]\rangle\bigg|_{\epsilon=0}$$

$$= \frac{d}{d\epsilon}\int_{\partial_1\mathcal{M}} \left([c(\phi+\epsilon\xi) - z]\,\psi + \left[C^a{}_b(W^b + \epsilon X^b) - Z^a\right] V_a\right)\, ds\bigg|_{\epsilon=0}$$

$$+ \frac{d}{d\epsilon}\int_{\mathcal{M}} \Big(\hat{D}_a(\phi+\epsilon\xi)\hat{D}^a\psi + 2\mu(\hat{E}(W+\epsilon X))^{ab}(\hat{E}V)_{ab}$$

$$+\lambda\hat{D}_a(W^a + \epsilon X^a)\hat{D}_b V^b\Big)\, dx\bigg|_{\epsilon=0}$$

$$+ \frac{d}{d\epsilon}\int_{\mathcal{M}} \left(\frac{2}{3}(\phi+\epsilon\xi)^6\hat{D}^a\mathrm{tr}K + 8\pi\hat{j}^a\right) V_a\, dx\bigg|_{\epsilon=0}$$

$$+ \frac{d}{d\epsilon}\int_{\mathcal{M}} \Bigg(\frac{1}{8}\hat{R}(\phi+\epsilon\xi) + \frac{1}{12}(\mathrm{tr}K)^2(\phi+\epsilon\xi)^5$$

$$-\frac{1}{8}({}^*\!\hat{A}_{ab} + (\hat{L}(W+\epsilon X))_{ab})^2(\phi+\epsilon\xi)^{-7} - 2\pi\hat{\rho}(\phi+\epsilon\xi)^{-3}\Bigg)\psi\, dx\bigg|_{\epsilon=0}. \qquad (77)$$

After some simple manipulations using the product and chain rules, we are left with the following bilinear form (for fixed $[\phi, W^a]$), linear separately in each of the variables $[\xi, X^a]$ and $[\psi, V^a]$:

$$\langle DF([\phi, W^a])[\xi, X^a], [\psi, V^a]\rangle = \int_{\partial_1\mathcal{M}} \left(c\xi\psi + C^a{}_b X^b V_a\right)\, ds$$

$$+ \int_{\mathcal{M}} \left(\hat{D}_a\xi\hat{D}^a\psi + 2\mu(\hat{E}X)^{ab}(\hat{E}V)_{ab} + \lambda\hat{D}_a X^a\hat{D}_b V^b\right)\, dx$$

$$+ \int_{\mathcal{M}} \left(\frac{1}{8}\hat{R} + \frac{5}{12}(\mathrm{tr}K)^2\phi^4 + \frac{7}{8}({}^*\!\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-8} + 6\pi\hat{\rho}\phi^{-4}\right)\xi\psi\, dx$$

$$- \int_{\mathcal{M}} \left(\frac{1}{4}({}^*\!\hat{A}_{ab} + (\hat{L}W)_{ab})\phi^{-7}\right)(\hat{L}X)^{ab}\psi\, dx + \int_{\mathcal{M}} \left(4\phi^5\hat{D}^a\mathrm{tr}K\right) V_a\xi\, dx. \qquad (78)$$

Note that the first two volume integrals and the surface integral are completely symmetric in their arguments, and represent the symmetric part of the bilinear form. The third and fourth volume integrals are nonsymmetric in their arguments; the third volume integral represents the linearized coupling of W^a into the Hamiltonian constraint, and the fourth volume integral represents the linearized coupling of the conformal factor ϕ into the momentum constraint.

2.8. Weak Formulations Arising from Energy Functionals

Due to the fact that the principle parts of the Hamiltonian and momentum operators produced by the conformal decomposition are self-adjoint, the weak formulations individually arise naturally as the Euler conditions for stationarity of associated (energy) functionals. It is straight-forward to verify that the following energy functionals

$$J_{\mathrm{H}}(\phi) = \frac{1}{2}\int_{\partial_1\mathcal{M}} c(\phi - z)\phi \; ds + \int_{\mathcal{M}} P(\phi, W^a) \; dx + \frac{1}{2}\int_{\mathcal{M}} \hat{D}_a\phi\hat{D}^a\phi \; dx, \tag{79}$$

$$J_{\mathrm{M}}(W^a) = \frac{1}{2}\int_{\partial_1\mathcal{M}} \left(C^a{}_b W^b - Z^a\right) W_a \; ds + \int_{\mathcal{M}} \left(\frac{2}{3}\phi^6\hat{D}^a\mathrm{tr}K + 8\pi\hat{j}^a\right) W_a \; dx \tag{80}$$

$$+ \int_{\mathcal{M}} \frac{1}{2}\left(2\mu(\hat{E}W)^{ab}(\hat{E}W)_{ab} + \lambda\hat{D}_a W^a\hat{D}_b W^b\right) \; dx, \tag{81}$$

each separately give rise to the individual weak forms (73) and (74), respectively. One computes the Gateaux derivative of $J_{\mathrm{H}}(\phi, W^a)$ with respect to ϕ, and the Gateaux derivative of $J_{\mathrm{M}}(\phi, W^a)$ with respect to W^a, and then sets them to zero:

$$\left.\frac{d}{d\epsilon}J_{\mathrm{H}}(\phi + \epsilon\psi)\right|_{\epsilon=0} = \langle J_{\mathrm{H}}'(\phi), \psi\rangle = \langle F_{\mathrm{H}}(\phi), \psi\rangle = 0, \tag{82}$$

$$\left.\frac{d}{d\epsilon}J_{\mathrm{M}}(W^a + \epsilon V^a)\right|_{\epsilon=0} = \langle J_{\mathrm{M}}'(W^a), V^a\rangle = \langle F_{\mathrm{M}}(W^a), V^a\rangle = 0. \tag{83}$$

This discussion is only formal, but can be made rigorous.

Unfortunately, while the individual constraints each arise as Euler conditions for stationarity of the separate energy functionals above, the coupled constraints do not arise in this way from any coupled energy. This follows easily from the fact that the combined linearization bilinear form in (78) is not symmetric. This can also be verified directly by considering the most general possible expression for the total energy:

$$J_{total}(\phi, W^a) = J_{\mathrm{H}}(\phi, W^a) + J_{\mathrm{M}}(\phi, W^a) + J_{\mathrm{R}}(\phi, W^a), \tag{84}$$

where $J_{\mathrm{H}}(\phi, W^a)$ and $J_{\mathrm{M}}(\phi, W^a)$ are as defined above, and where $J_{\mathrm{R}}(\phi, W^a)$ is the remainder term in the energy which must account for the coupling terms in the combined weak form (76). It is easy to verify that the Euler condition for stationarity places separate conditions on the Gateaux derivative of $J_{\mathrm{R}}(\phi, W^a)$ which are impossible to meet simultaneously.

This lack of a variational principle for the coupled constraints limits the number of techniques available for analyzing solvability of the coupled system; the existing near-CMC results [7, 16] and the non-CMC (far-from-CMC) results [17, 18] are actually based on fixed-point arguments; however, variational arguments are used to solve the individual constaint equations in [16, 17, 18] as part of the overall fixed-point argument.

3. Adaptive Finite Element Methods (AFEM)

In this section we give a brief description of Galerkin methods, finite element methods, and adaptive techniques for covariant nonlinear elliptic systems. We also derive *a posteriori* error indicators for driving adaptivity, and finish the section with some *a priori* error estimates for general Galerkin approximations to the Hamiltonian and momentum constraints. Expanded versions of this material, including proofs of all results, can be found in [26, 27, 28].

3.1. Petrov-Galerkin Methods, Galerkin Methods, and Finite Element Methods

A *Petrov-Galerkin* approximation of the solution to (30) is the solution to the following subspace problem:

$$\text{Find } (u_h - \bar{u}_h) \in U_h \subset \mathcal{B}_1 \text{ s.t. } \langle F(u_h), v \rangle = 0, \qquad \forall\ v \in V_h \subset \mathcal{B}_2, \tag{85}$$

for some chosen subspaces U_h and V_h, where $\dim(U_h) = \dim(V_h) = n$, and where the discrete Dirichlet function $\bar{u}_h$ approximates $\bar{u}$ (e.g. an interpolant). A *Galerkin* approximation refers to the case that $U_h = V_h$.

A *finite element* method is simply a Petrov-Galerkin or Galerkin method in which the subspaces U_h and V_h are chosen to have the extremely simple form of continuous piecewise polynomials with local support, defined over a disjoint covering of the domain manifold $\mathcal{M}$ by *elements*. For example, in the case of continuous piecewise linear polynomials defined over a disjoint covering with 2- or 3-simplices (cf. Figure 3.1), the basis functions are easily defined element-wise using the unit 2-simplex (triangle) and unit 3-simplex (tetrahedron) as follows:

$$\begin{array}{rcl} \tilde{\phi}_0(\tilde{x},\tilde{y}) & = & 1-\tilde{x}-\tilde{y} \\ \tilde{\phi}_1(\tilde{x},\tilde{y}) & = & \tilde{x} \\ \tilde{\phi}_2(\tilde{x},\tilde{y}) & = & \tilde{y} \end{array} \qquad \begin{array}{rcl} \tilde{\phi}_0(\tilde{x},\tilde{y},\tilde{z}) & = & 1-\tilde{x}-\tilde{y}-\tilde{z} \\ \tilde{\phi}_1(\tilde{x},\tilde{y},\tilde{z}) & = & \tilde{y} \\ \tilde{\phi}_2(\tilde{x},\tilde{y},\tilde{z}) & = & \tilde{x} \\ \tilde{\phi}_3(\tilde{x},\tilde{y},\tilde{z}) & = & \tilde{z} \end{array} \quad .$$

Global basis functions are then defined, as in the right-most picture in Figure 3.1, by combining the support regions around a given vertex, and extending the unit simplex basis functions to each arbitrary simplex using coordinate transformations. If the manifold domain can be triangulated exactly with simplex elements, then the coordinate transformations are simply affine transformations. Note that in this sense, finite element methods are by their very nature defined in a chart-wise manner. Quadratic and high-order basis functions are defined analogously.

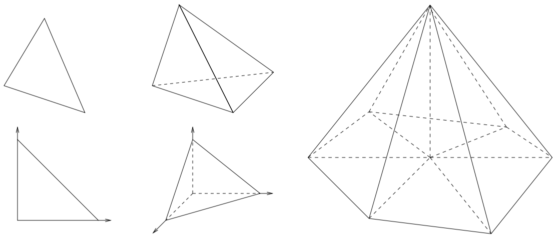


Figure 1. Reference and arbitrary 2- and 3-simplex elements, and a global (2D) basis function.

The above basis functions clearly do not form any subspace of $\mathcal{C}^2(\mathcal{M})$, the space of twice continuously differentiable functions on $\mathcal{M}$, which is the natural function space in which to look for the solutions to second order elliptic equations. This is due to the fact that they are discontinuous along simplex faces and simplex vertices in the disjoint simplex covering of $\mathcal{M}$. However, one can show [51] that in fact:

$$V_h = \text{span}\{\phi_1, \ldots, \phi_n\} \subset W^{1,p}_{0,D}(\mathcal{M}), \quad \mathcal{M} \subset \mathbb{R}^d, \tag{86}$$

so that these continuous, piecewise defined, low-order polynomial spaces do in fact form a subspace of the solution space to the weak formulation of the class of second order elliptic equations of interest. Making then the choice $U_h = \text{span}\{\phi_1, \phi_2, \ldots, \phi_n\}$, $V_h = \text{span}\{\psi_1, \psi_2, \ldots, \psi_n\}$, equation (85) reduces to a set of n nonlinear algebraic relations (implicitly defined) for the n coefficients $\{\alpha_j\}$ in the expansion

$$u_h = \bar{u}_h + \sum_{j=1}^{n} \alpha_j \phi_j. \tag{87}$$

In particular, regardless of the complexity of the form $\langle F(u), v\rangle$, as long as we can evaluate it for given arguments u and v, then we can evaluate the nonlinear discrete residual of the finite element approximation u_h as:

$$r_i = \langle F(\bar{u}_h + \sum_{j=1}^{n} \alpha_j \phi_j), \psi_i\rangle, \qquad i = 1, \ldots, n. \tag{88}$$

Since the form $\langle F(u), v\rangle$ involves an integral in this setting, if we employ quadrature then we can simply sample the integrand at quadrature points; this is a standard technique in finite element technology. Given the local support nature of the functions ϕ_j and ψ_i, all but a small constant number of terms in the sum $\sum_{j=1}^{n} \alpha_j \phi_j$ are zero at a particular spatial point in the domain, so that the residual r_i is inexpensive to evaluate when quadrature is employed.

The two primary issues in applying this approximation method are then:

(i) The approximation error $\|u - u_h\|_X$, for various norms X, and

(ii) The computational complexity of solving the n algebraic equations.

The first of these issues represents the core of finite element approximation theory, which itself rests on the results of classical approximation theory. Classical references to both topics include [51, 52, 53]. The second issue is addressed by the complexity theory of direct and iterative solution methods for sparse systems of linear and nonlinear algebraic equations, cf. [54, 55], and by the use of adaptive techniques to minimize the size n of the discrete space that must be constructed to reach a specific approximation quality.

3.2. A Priori *Error Estimates for the Constraint Equations*

We first outline an approximation result for general Galerkin approximations to solutions of the momentum constraint. It is referred to as a quasi-optimal error estimate, in that it establishes a bound on the error $\|u - u_h\|_X$ that is within a constant of being the error in the best possible approximation.

To understand this result, we begin with the two (Hilbert vector) spaces H and V, where in our setting of the momentum constraint, we have $H = L^2(\mathcal{M})$ and $V = H^1_{0,D}(\mathcal{M})$. We will stay with the abstract notation involving H and V for clarity. The weak form of the momentum constraint can be shown to have the following form:

$$\text{Find } u \in V \text{ s.t. } A(u, v) = F(v), \quad \forall v \in V, \tag{89}$$

where the bilinear form $A(u, v) : V \times V \mapsto \mathbb{R}$ is bounded

$$A(u, v) \leq M\|u\|_V \|v\|_V, \quad \forall u, v \in V, \tag{90}$$

and V-coercive (satisfying a Gårding inequality):

$$m\|u\|_V^2 \leq K\|u\|_H^2 + A(u, u), \quad \forall u \in V, \quad \text{where } \; m > 0, \tag{91}$$

and where the linear functional $F(v) : V \mapsto \mathbb{R}$ is bounded and thus lies in the dual space V^*:

$$F(v) \leq L\|v\|_V, \quad \forall v \in V.$$

It can be shown that the weak formulation of the momentum constraint fits into this framework; to simplify the discussion, we have assumed that any Dirichlet function $\bar{u}$ has been absorbed into the linear functional $F(v)$ in the obvious way. Our discussion can be easily modified to include approximation of $\bar{u}$ by $\bar{u}_h$.

Now, we are interested in the quality of a Galerkin approximation:

$$\text{Find } u_h \in V_h \subset V \text{ s.t. } A(u_h, v) = A(u, v) = F(v), \quad \forall v \in V_h \subset V. \tag{92}$$

We will assume that there exists a sequence of approximation subspaces $V_h \subset V$ parameterized by h, with $V_{h_1} \subset V_{h_2}$ when $h_2 < h_1$, and that there exists a sequence $\{a_h\}$, with $\lim_{h \to 0} a_h = 0$, such that

$$\|u - u_h\|_H \leq a_h \|u - u_h\|_V, \text{ when } A(u - u_h, v) = 0, \ \forall v \in V_h \subset V. \tag{93}$$

The assumption (93) is very natural; in our setting, it is the assumption that the error in the approximation converges to zero more quickly in the L^2-norm than in the H^1-norm. This is easily verified in the setting of piecewise polynomial approximation spaces, under very mild smoothness requirements on the solution u. Under these assumptions, we have the following *a priori* error estimate.

Theorem 3.1 *For h sufficiently small, there exists a unique approximation u_h satisfying (92), for which the following quasi-optimal* a priori *error bounds hold:*

$$\|u - u_h\|_V \leq C \inf_{v \in V_h} \|u - v\|_V, \tag{94}$$

$$\|u - u_h\|_H \leq C a_h \inf_{v \in V_h} \|u - v\|_V, \tag{95}$$

where C is a constant independent of h. If $K \leq 0$ in (91), then the above holds for all h.

Proof. See [26, 27, 28]; also [56]. □

As we did previously for the momentum constraint, we now outline an approximation result for general Galerkin approximations to solutions of the Hamiltonian constraint. Again, it is referred to as a quasi-optimal error estimate, in that it establishes a bound on the error $\|u - u_h\|_X$ that is within a constant of being the error in the best possible approximation.

We begin again with the two Hilbert spaces H and V, where again we have $H = L^2(\mathcal{M})$ and $V = H^1_{0,D}(\mathcal{M})$. We are given the following nonlinear variational problem:

$$\text{Find } u \in V \text{ s.t. } A(u, v) + \langle B(u), v \rangle = F(v), \quad \forall v \in V, \tag{96}$$

where the bilinear form $A(u, v) : V \times V \mapsto \mathbb{R}$ is bounded

$$A(u, v) \leq M\|u\|_V \|v\|_V, \quad \forall u, v \in V, \tag{97}$$

and V-elliptic:

$$m\|u\|_V^2 \leq A(u, u), \quad \forall u \in V, \quad \text{where} \quad m > 0, \tag{98}$$

where the linear functional $F(v) : V \mapsto \mathbb{R}$ is bounded and thus lies in the dual space V^*:

$$F(v) \leq L\|v\|_V, \quad \forall v \in V,$$

and where the nonlinear form $\langle B(u), v\rangle : V \times V \mapsto \mathbb{R}$ is assumed to be monotonic:

$$0 \leq \langle B(u) - B(v), u - v\rangle, \quad \forall u, v \in V, \tag{99}$$

where we have used the notation:

$$\langle B(u) - B(v), w\rangle = \langle B(u), w\rangle - \langle B(v), w\rangle. \tag{100}$$

We are interested in the quality of a Galerkin approximation:

$$\text{Find } u_h \in V_h \text{ s.t. } A(u_h, v) + \langle B(u_h), v\rangle = F(v), \quad \forall v \in V_h, \tag{101}$$

where $V_h \subset V$. We will assume that $\langle B(u), v\rangle$ is bounded in the following weak sense: If $u \in V$ satisfies (96), if $u_h \in V_h$ satisfies (101), and if $v \in V_h$, then there exists a constant $K > 0$ such that:

$$\langle B(u) - B(u_h), u - v\rangle \leq K\|u - u_h\|_V \|u - v\|_V. \tag{102}$$

It can be shown that the weak formulation of the Hamiltonian constraint fits into this framework. We have again assumed that any Dirichlet function $\bar{u}$ has been absorbed into the various forms in the obvious way to simplify the discussion. The discussion can be modified to include approximation of $\bar{u}$ by $\bar{u}_h$.

Again, we are interested in the quality of a Galerkin approximation u_h satisfying (101), or equivalently:

$$A(u - u_h, v) + \langle B(u) - B(u_h), v\rangle = 0, \ \forall v \in V_h \subset V.$$

As before, we will assume that there exists a sequence of approximation subspaces $V_h \subset V$ parameterized by h, with $V_{h_1} \subset V_{h_2}$ when $h_2 < h_1$, and that there exists a sequence $\{a_h\}$, with $\lim_{h\to 0} a_h = 0$, such that

$$\|u - u_h\|_H \leq a_h \|u - u_h\|_V, \tag{103}$$

holds whenever u_h satisfies (101). The assumption (103) is again very natural; see the discussion above following (93). Under these assumptions, we have the following *a priori* error estimate.

Theorem 3.2 *The approximation u_h satisfying (101) obeys the following quasi-optimal* a priori *error bounds:*

$$\|u - u_h\|_V \leq C \inf_{v\in V_h} \|u - v\|_V, \tag{104}$$

$$\|u - u_h\|_H \leq C a_h \inf_{v\in V_h} \|u - v\|_V, \tag{105}$$

where C is a constant independent of h.

Proof. See [26, 27, 28]. □

3.3. Adaptive Finite Element Methods (AFEM)

A priori error analysis for the finite element method for addressing the first issue is now a well-understood subject [51, 57]. Much activity has recently been centered around *a posteriori* error estimation and its use in adaptive mesh refinement algorithms [58, 59, 60, 61, 62, 63]. These estimators include weak and strong residual-based estimators [59, 60, 61], as well as estimators based on the solution of local problems [64, 65]. The challenge for a numerical method is to be as efficient as possible, and *a posteriori* estimates are a basic tool in deciding which parts of the solution require additional attention. While the majority of the work on *a posteriori* estimates has been for linear problems, nonlinear extensions are possible through linearization theorems (cf. [61, 62] and the discussion of the error estimator employed by FETK later in this paper). The solve-estimate-refine structure in simplex-based adaptive finite element codes such as FETK [26] and PLTMG [58], exploiting these *a posteriori* estimators, is as follows:

Algorithm 1 *(Adaptive multilevel finite elements)*

```
• While ($\|u - u_h\|_X > \epsilon$) do:
      (i) Find $(u_h - \bar{u}_h) \in U_h \subset \mathcal{B}_1$ such that:
          $\langle F(u_h), v \rangle = 0, \ \forall \ v \in V_h \subset \mathcal{B}_2$.
     (ii) Estimate $\|u - u_h\|_X$ over each element.
    (iii) Initialize two temporary simplex lists as empty: $Q1 = Q2 = \emptyset$.
     (iv) Place simplices with large error on the "refinement" list $Q1$.
      (v) Bisect all simplices in $Q1$ (removing from $Q1$), and place any nonconforming
          simplices created on the list $Q2$.
     (vi) $Q1$ is now empty; set $Q1 = Q2$, $Q2 = \emptyset$.
    (vii) If $Q1$ is not empty, goto (5).
• End While.
```

The conformity loop (5)–(7), required to produce a globally "conforming" mesh (described below) at the end of a refinement step, is guaranteed to terminate in a finite number of steps (cf. [66, 67]), so that the refinements remain local. Element shape is crucial for approximation quality; the bisection procedure in step (5) is guaranteed to produce nondegenerate families if the longest edge is bisected in two dimensions [68, 69], and if marking or homogeneity methods are used in three dimensions [22, 23, 70, 71, 72, 73]. Whether longest edge bisection is nondegenerate in three dimensions apparently remains an open question. Figure 2 shows a single subdivision of a 2-simplex or a 3-simplex using either 4-section (left-most figure), 8-section (fourth figure from the left), or bisection (third figure from the left, and the right-most figure). The paired triangle in the 2-simplex case of Figure 2 illustrates the nature of conformity and its violation during refinement. A globally conforming simplex mesh is defined as a collection of simplices which meet only at vertices and faces; for example, removing the dotted bisection in the third group from the left in Figure 2 produces a non-conforming mesh. Non-conforming simplex meshes create several theoretical as well as practical implementation difficulties, and the algorithms in FETK (as well as those in PLTMG [58] and similar simplex-based adaptive codes [26, 23, 74, 75, 76]) enforce conformity using the above queue swapping strategy or a similar approach.

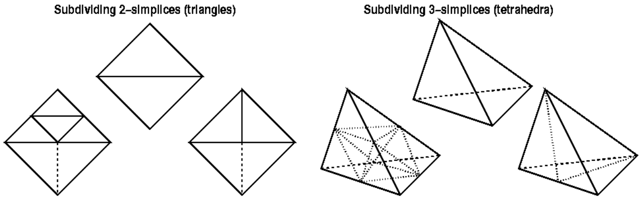

Subdividing 2–simplices (triangles)
Subdividing 3–simplices (tetrahedra)

Figure 2. Refinement of 2- and 3-simplices using 4-section, 8-section, and bisection.

Addressing the complexity of Step 1 of the algorithm above, Newton methods are often the most effective:

Algorithm 2 *(Damped-inexact-Newton)*

```
• Let an initial approximation $u$ be given.
• While $(|\langle F(u), v\rangle| > \epsilon$ for any $v)$ do:
    (i) Find $w$ such that:
        $\langle DF(u)w, v\rangle = -\langle F(u), v\rangle + r, \ \forall \ v$.
    (ii) Set $u = u + \lambda w$.
• End While.
```

The bilinear form $\langle DF(u)w, v\rangle$ in the algorithm above is simply the (Gateaux) linearization of the nonlinear form $\langle F(u), v\rangle$, defined formally as:

$$\langle DF(u)w, v\rangle = \frac{d}{d\epsilon}\langle F(u+\epsilon w), v\rangle\bigg|_{\epsilon=0}. \tag{106}$$

This form is easily computed from most nonlinear forms $\langle F(u), v\rangle$ which arise from second order nonlinear elliptic problems, although the calculation can be tedious in some cases (as in the case of the constraints in general relativity). The possibly nonzero "residual" term r is to allow for inexactness in the Jacobian solve for efficiency, which is quite effective in many cases (cf. [77, 78, 79]). The parameter λ brings robustness to the algorithm [79, 80, 81]. If folds or bifurcations are present, then the iteration is modified to incorporate path-following [82, 83].

As was the case for the nonlinear residual $\langle F(\cdot), \cdot\rangle$, the matrix representing the bilinear form in the Newton iteration is easily assembled, regardless of the complexity of the bilinear form $\langle DF(\cdot)\cdot, \cdot\rangle$. In particular, the algebraic system for $w = \sum_{j=1}^{n} \beta_j \phi_j$ has the form:

$$AU = F, \qquad U_i = \beta_i, \tag{107}$$

where

$$A_{ij} = \langle DF(\bar{u}_h + \sum_{k=1}^{n} \alpha_k \phi_k)\phi_j, \psi_i\rangle, \tag{108}$$

$$F_i \ \ = \langle F(\bar{u}_h + \sum_{j=1}^{n} \alpha_j \phi_j), \psi_i\rangle. \tag{109}$$

As long as the integral forms $\langle F(\cdot), \cdot\rangle$ and $\langle DF(\cdot)\cdot, \cdot\rangle$ can be evaluated at individual points in the domain, then quadrature can be used to build the Newton equations, regardless of the complexity of the forms. This is one of the most powerful features of the finite element method, and is exploited to an extreme in the code FETK (see Section 6 and [26]). It should be noted that there is a subtle difference between the approach outlined here (typical for a nonlinear finite element approximation) and that usually taken when applying a Newton-iteration to a nonlinear finite difference approximation. In particular, in the finite difference setting the discrete equations are linearized explicitly by computing the Jacobian of the system of nonlinear algebraic equations. In the finite element setting, the commutativity of linearization and discretization is exploited; the Newton iteration is actually performed in function space, with discretization occurring "at the last moment" in Algorithm 2 above.

It can be shown that the Newton iteration above is dominated by the computational complexity of solving the n linear algebraic equations in each iteration (cf. [77, 84]). Multilevel methods are the only known provably optimal or nearly optimal methods for solving these types of linear algebraic equations resulting from discretizations of a large class of general linear elliptic problems [84, 85, 86]. An obstacle to applying multilevel methods to the constraint equations in general relativity and to similar equations is the presence of geometrically complex domains. The need to accurately represent complicated domain features and boundaries with an adapted mesh requires the use of very fine mesh simply to describe the complexities of the domain. This may preclude the use of the solve-estimate-refine structure outlined above in some cases, which requires starting with a coarse mesh in order to build the approximation and linear algebra hierarchies as the problem is solved adaptively. In this situation, algebraic or agglomeration/aggregation-based multilevel methods can be employed [87, 88, 89, 90, 91, 92, 93, 94, 95]. A fully unstructured algebraic multilevel approach is taken in FETK, even when the refinement hierarchy is present; see Section 6 below and also [26] for a more detailed description.

3.4. Residual-Based A Posteriori *Error Indicators*

There are several approaches to adaptive error control, although the approaches based on *a posteriori* error estimation are usually the most effective and most general. While most existing work on *a posteriori* estimates has been for linear problems, extensions to the nonlinear case can be made through linearization. To describe one such result from [26], we assume that the d-manifold $\mathcal{M}$ has been exactly triangulated with a set $\mathcal{S}$ of shape-regular d-simplices (the finite dimension d is arbitrary throughout this discussion). A family of simplices will be referred to here as shape-regular if for all simplices in the family the ratio of the diameter of the circumscribing sphere to that of the inscribing sphere is bounded by an absolute fixed constant, independent of the numbers and sizes of the simplices that may be generated through refinements. (For a more careful definition of shape-regularity and related concepts, see [51].) It will be convenient to introduce the following notation:

$$\begin{array}{lcl}
\mathcal{S} & = & \text{Set of shape-regular simplices triangulating } \mathcal{M} \\
\mathcal{N}(s) & = & \text{Union of faces in simplex set } s \text{ lying on } \partial_N \mathcal{M} \\
\mathcal{I}(s) & = & \text{Union of faces in simplex set } s \text{ not in } \mathcal{N}(s) \\
\mathcal{F}(s) & = & \mathcal{N}(s) \cup \mathcal{I}(s) \\
\omega_s & = & \bigcup \; \{ \; \tilde{s} \in \mathcal{S} \mid s \bigcap \tilde{s} \neq \emptyset, \text{ where } s \in \mathcal{S} \; \} \\
\omega_f & = & \bigcup \; \{ \; \tilde{s} \in \mathcal{S} \mid f \bigcap \tilde{s} \neq \emptyset, \text{ where } f \in \mathcal{F} \; \} \\
h_s & = & \text{Diameter (inscribing sphere) of the simplex } s \\
h_f & = & \text{Diameter (inscribing sphere) of the face } f.
\end{array}$$

When the argument to one of the face set functions $\mathcal{N}$, $\mathcal{I}$, or $\mathcal{F}$ is in fact the entire set of simplices $\mathcal{S}$, we will leave off the explicit dependence on $\mathcal{S}$ without danger of confusion. Referring forward briefly to Figure 5 will be convenient. The two darkened triangles in the left picture in Figure 5 represents the set w_f for the face f shared by the two triangles. The clear triangles in the right picture in Figure 5 represents the set w_s for the darkened triangle s in the center (the set w_s also includes the darkened triangle).

Finally, we will also need some notation to represent discontinuous jumps in function values across faces interior to the triangulation. To begin, for any face $f \in \mathcal{N}$, let n_f denote the unit outward normal; for any face $f \in \mathcal{I}$, take n_f to be an arbitrary (but fixed) choice of one of the two possible face normal orientations. Now, for any $v \in L^2(\mathcal{M})$ such that $v \in C^0(s)$ $\forall s \in \mathcal{S}$, define the *jump function*:

$$[v]_f(x) = \lim_{\epsilon \to 0^+} v(x + \epsilon n_f) - \lim_{\epsilon \to 0^-} v(x - \epsilon n_f).$$

By analyzing the element-wise volume and surface integrals in (31) and using some technical results on interpolation of functions, the following fairly standard result is derived in [26]:

Theorem 3.3 *Let $u \in W^{1,r}(\mathcal{M})$ be a regular solution of (20)–(22), or equivalently of (30)–(31), where some additional minimal assumptions hold as described in [26]. Then the following* a posteriori *error estimate holds for a Petrov-Galerkin approximation u_h satisfying (85):*

$$\|u - u_h\|_{W^{1,r}(\mathcal{M})} \leq C \left(\sum_{s \in \mathcal{S}} \eta_s^p \right)^{1/p}, \tag{110}$$

where

$$C = 2 \cdot \max_{\mathcal{S},\mathcal{F}} \{C_s, C_f\} \cdot \max_{\mathcal{S},\mathcal{F}} \{D_s^{1/q}, D_f^{1/q}\} \cdot \|DF(u)^{-1}\|_{\mathcal{L}(W^{-1,q}, W^{1,p})},$$

and where the element-wise error indicator η_s is defined as:

$$\begin{aligned} \eta_s = \Bigg(& h_s^p \|B^i - A^{ia}{}_{;a}\|^p_{L^p(s)} + \frac{1}{2} \sum_{f \in \mathcal{I}(s)} h_f \| \left[A^{ia} n_a \right]_f \|^p_{L^p(f)} \\ & + \sum_{f \in \mathcal{N}(s)} h_f \|C^i + A^{ia} n_a\|^p_{L^p(f)} \Bigg)^{1/p} . \end{aligned} \tag{111}$$

Proof. See [26, 27, 28]. □

3.5. *An* A Posteriori *Error Indicator for the Constraints*

Here, we using the general indicator above, we can instantiate an estimator specifically for the constraints in general relativity. The Ph.D. thesis of Mukherjee [23] contains a residual-based error estimator for the Hamiltonian constraint that is equivalent to our estimator when the momentum constraint is not involved, in the specific case of $p = q = r = 2$. We consider first the Hamiltonian constraint, which can be thought of as an equation of the form (20)–(22). The error indicator from above now takes the form:

$$\begin{aligned} \eta_s^{\mathrm{H}} = \Bigg(& h_s^p \| \frac{1}{8} \hat{R} \phi_h + \frac{1}{12} (\mathrm{tr} K)^2 \phi_h^5 - \frac{1}{8} ({}^*\!\hat{A}_{ab} + (\hat{L} W)_{ab})^2 \phi_h^{-7} - 2\pi \hat{\rho} \phi_h^{-3} - \hat{D}_a \hat{D}^a \phi_h \|^p_{L^p(s)} \\ & + \frac{1}{2} \sum_{f \in \mathcal{I}(s)} h_f \| \left[\hat{n}_a \hat{D}^a \phi_h \right]_f \|^p_{L^p(f)} + \sum_{f \in \mathcal{N}(s)} h_f \| c \phi_h - z + \hat{n}_a \hat{D}^a \phi_h \|^p_{L^p(f)} \Bigg)^{1/p} . \end{aligned} \tag{112}$$

The momentum constraint also has the form (20)–(22), and the error indicator in [26] takes the form:

$$\begin{aligned} \eta_s^{\mathrm{M}} = \Bigg(& h_s^p \| \frac{2}{3} \phi^6 \hat{D}^a \mathrm{tr} K + 8\pi \hat{j}^a - \hat{D}_b (\hat{L} W_h)^{ab} \|^p_{L^p(s)} + \frac{1}{2} \sum_{f \in \mathcal{I}(s)} h_f \| \left[\hat{n}_b (\hat{L} W_h)^{ab} \right]_f \|^p_{L^p(f)} \\ & + \sum_{f \in \mathcal{N}(s)} h_f \| C^a{}_b W_h^b - Z^a + \hat{n}_b (\hat{L} W_h)^{ab} \|^p_{L^p(f)} \Bigg)^{1/p} . \end{aligned} \tag{113}$$

Finally, the error indicator we employ for the coupled system is the l^p-weighted average of the two estimators above:

$$\eta_s^{HM} = \left(w_H(\eta_s^H)^p + w_M(\eta_s^M)^p \right)^{1/p}, \tag{114}$$

where the weights w_H and w_M satisfying $w_H + w_M = 1$ are determined heuristically.

Note that in this special case the weighted sum estimator in (114) can be derived from the general system estimator in [26] by defining the product space metric described in [26] as follows:

$$\mathcal{G}_{ij} = \left[\begin{array}{cc} w_H & 0 \\ 0 & w_M g_{ab} \end{array} \right], u^i = \left[\begin{array}{c} \phi \\ W^a \end{array} \right], v^j = \left[\begin{array}{c} \psi \\ V^b \end{array} \right], \tag{115}$$

and employing the coupled system framework from [26, 27]. However, in general different components of a coupled system could lie in different function spaces, and the norm appearing in the estimator in [26, 27] would need to be modified. Equivalently, an estimator built from a weighted sum of estimators for individual components could be used as above.

3.6. Convergence and Optimal Complexity of AFEM for the Constraints

The following convergence result for AFEM applied the a general class of nonlinear elliptic equations that includes the Hamiltonian constraint appears in [28]. More general results which apply to the coupled system appear in [27]. Below, the energy norm is used as the natural norm for the analysis:

$$|||u||| = A(u,u)^2,$$

where the bilinear form $A(u,v)$ is defined in Section 2.6.

Theorem 3.4 (Contraction) *Let $\{P_k, S_k, U_k\}_{k\geq 0}$ be the sequence of finite element meshes, spaces, and solutions, respectively, produced by AFEM. Let the initial mesh size h_0 be sufficiently small so that a quasi-orthogonality inequality holds (see [28]) holds for $\{P_0, S_0, U_0\}_{k\geq 0}$. Then, there exist constants $\gamma > 0$ and $\alpha \in (0,1)$, depending only on some freely specifiable AFEM parameters and the shape-regularity of the initial triangulation P_0, such that*

$$|||u - U_{k+1}|||^2 + \gamma\eta_{k+1}^2 \leq \alpha \left(|||u - U_k|||^2 + \gamma\eta_k^2 \right).$$

Proof. See [28]. □

The strict contraction result above makes possible the following complexity result from [28], which guarantees optimal complexity of the AFEM iteration for the Hamiltonian constraint equation under reasonable assumptions on the problem data. The approximation class $\mathbb{A}_s$ is precisely characterized in [27, 28].

Theorem 3.5 (Optimality) *If data lies in approximation class $\mathbb{A}_s$, then there exists a constant C such that*

$$|||u - U_{k+1}|||^2 + \mathrm{osc}_k \leq C \left(\#P_k - \#P_0 \right)^{-s}.$$

Proof. See [28]. □

4. Fast Solvers and Preconditioners for AFEM

We first introduce a superscript (j) to indicate the level of the stiffness matrix (or sometimes, the discretization operator) in the algebraic system (107). At certain places, we will drop the superscript for simplicity. The FE solution is sampled on the nodes, hence the number of internal nodes is equal to the number of degrees of freedom (DOF) or unknowns in the system (107). The total number of DOF on the finest level J is denoted by $N_J = N$.

The stiffness matrix $A^{(j)}$ is ill-conditioned, with the condition number $\kappa(A)$ of the system (107) growing $\mathcal{O}(2^{2j})$ as $j \to \infty$ (in the case of second order elliptic PDE). It is imperative to improve the condition of (107) by transforming the system to an equivalent one, namely by preconditioning

$$(C^{(j)}\ A^{(j)})\ U^{(j)} = C^{(j)}\ F^{(j)},$$

where $\kappa({C^{(j)}}^{1/2}\ A^{(j)}\ {C^{(j)}}^{1/2}) \ll \kappa(A^{(j)})$. Moreover, the preconditioning matrix $C^{(j)}$ has to be positive definite, and in some sense simple. One possible way to implement the above strategy is to determine a positive definite matrix $B^{(j)}$, having the following two properties:

- ${B^{(j)}}^{-1}$ can efficiently be computed (usually, $\mathcal{O}(N_j)$ is the desirable bound for the number of arithmetical operations when solving a linear system with coefficient matrix $B^{(j)}$),
- $A^{(j)}$ and $B^{(j)}$ are "almost" spectrally equivalent, i.e.

  $$\lambda_B x^T B^{(j)} x \le x^T A^{(j)} x \le \Lambda_B x^T B^{(j)} x, \quad x \in \mathbb{R}^{N_j},$$

  with two positive constants λ_B, Λ_B with a small ratio $\frac{\Lambda_B}{\lambda_B}$.

Since $\kappa({B^{(j)}}^{-1/2}\ A^{(j)}\ {B^{(j)}}^{-1/2}) \le \frac{\Lambda_B}{\lambda_B}$, then $C^{(j)} = {B^{(j)}}^{-1}$ will be a good preconditioner choice.

Solution of the algebraic system (107) by iterative methods has been the subject of intensive research because of the enormous practical impact on a number of application areas in computational science. For quality approximation in physical simulation, one is required to use meshes containing very large numbers of simplices leading to approximation spaces $\mathcal{S}_j$ with very large dimension N_j. Only iterative methods which scale well with N_j can be used effectively, which usually leads to the use of multilevel-type iterative methods and preconditioners. Even with the use of such optimal methods for (107), which means methods which scale linearly with N_j in both memory and computational complexity, the approximation quality requirements on $\mathcal{S}_j$ often force N_j to be so large that only parallel computing techniques can be used to solve (107).

To overcome this difficulty one employs adaptive methods, which involves the use of *a posteriori* error estimation to drive *local mesh refinement* algorithms. This approach leads to approximation spaces $\mathcal{S}_j$ which are adapted to the particular target function u of interest, and as a result can achieve a desired approximation quality with much smaller approximation space dimension N_j than non-adaptive methods. One still must solve the algebraic system (107), but unfortunately most of the available multilevel methods and preconditioners are no longer optimal, in either memory or computational complexity. This is due to the fact that in the local refinement setting, the approximation spaces S_j do not increase in dimension geometrically as they do in the uniform refinement setting. As a result for example, a single multilevel V-cycle no longer has linear complexity, and the same difficulty is encountered by other multilevel methods. Moreover, storage of the discretization matrices and vectors for each approximation space, required for assembling V-cycle and similar iterations, no longer has linear memory complexity.

A partial solution to the problem with multilevel methods in the local refinement setting is provided by the hierarchical basis (HB) method [96, 97, 98]. This method is based on a direct or hierarchical decomposition of the approximation spaces $\mathcal{S}_j$ rather than the overlapping decomposition employed by the multigrid and Bramble-Pasciak-Xu (BPX) [99] method, and therefore by construction has linear memory complexity as well as linear computational complexity for a single V-cycle-like iteration. Unfortunately, the HB condition number is not uniformly bounded, leading to worse than linear overall computational complexity. While the condition number growth is slow (logarithmic) in two dimensions, it is quite rapid (geometric) in three dimensions, making it ineffective in the 3D local refinement setting. Recent alternatives to the HB method, including both BPX-like methods [100, 99] and wavelet-like stabilizations of the HB methods [101], provide a final solution to the condition number growth problem. It was shown in [102] that the BPX preconditioner has uniformly bounded condition number for certain classes of locally refined meshes in two dimensions, and more recently in [103, 104, 105, 106] it was shown that the condition number remains uniformly bounded for certain classes of locally refined meshes in three spatial dimensions. In [103, 105, 107], it was also shown that wavelet-stabilizations of the HB method give rise to uniformly bounded conditions numbers for certain classes of local mesh refinement in both the two- and three-dimensional settings.

4.1. Preliminaries on Optimal Preconditioners

In the uniform refinement setting, the parallelized or additive version of the multigrid method, also known as the BPX preconditioner, is defined as follows:

$$Xu := \sum_{j=0}^{J} 2^{j(d-2)} \sum_{i=1}^{N_j} (u, \phi_i^{(j)}) \phi_i^{(j)}, \quad u \in \mathcal{S}_J. \tag{116}$$

Only in the presence of a geometric increase in the number of DOF, the same assumption for optimality of a single classical (i.e. smoother acting on all DOF) multigrid or BPX iteration, does the cost per iteration remain optimal. In the case of local refinement, the BPX preconditioner (116) (usually known as additive multigrid) can easily be suboptimal because of the suboptimal cost per iteration. If the smoother is restricted to the space generated by *fine or newly created* basis functions, i.e. $\tilde{\mathcal{S}}_j := (I_j - I_{j-1})\ \mathcal{S}_j$, then (116) corresponds to the additive HB preconditioner in [98]:

$$X_{\text{HB}} u = \sum_{j=0}^{J} 2^{j(d-2)} \sum_{i=N_{j-1}+1}^{N_j} (u, \phi_i^{(j)}) \phi_i^{(j)}, \quad u \in \mathcal{S}_J. \tag{117}$$

However, the HB preconditioner suffers from a suboptimal iteration count. Namely, if the algebraic system (107) is preconditioned by the HB preconditioner, the arising condition number is $\mathcal{O}(J^2)$ and $\mathcal{O}(2^J)$ in 2D and 3D, respectively.

In the local refinement setting, in order to maintain optimal overall computational complexity, the remedy is to restrict the smoother to a local space $\tilde{\mathcal{S}}_j$ which is typically slightly larger than the one generated by basis functions corresponding to fine DOF:

$$(I_j - I_{j-1})\, \mathcal{S}_j \subseteq \tilde{\mathcal{S}}_j \subset \mathcal{S}_j, \tag{118}$$

where $I_j : L_2(\Omega) \rightarrow \mathcal{S}_j$ denotes the finite element interpolation operator. The subspace generated by the nodal basis functions corresponding to *fine or newly created* degrees of freedom (DOF) on level j is denoted by $(I_j - I_{j-1})\, \mathcal{S}_j$. Nodes–equivalently, DOF–corresponding to $\tilde{\mathcal{S}}_j$ and their cardinality will be denoted by $\tilde{\mathcal{N}}_j$ and $\tilde{N}_j$, respectively. The above deficiencies of the BPX and HB preconditioners can be overcome by restricting the (smoothing) operations to $\tilde{\mathcal{S}}_j$. This leads us to define the BPX preconditioner for the local refinement setting as:

$$Xu := \sum_{j=0}^{J} 2^{j(d-2)} \sum_{i\in\tilde{\mathcal{N}}_j} (u, \phi_i^{(j)})\phi_i^{(j)}, \quad u \in \mathcal{S}_J. \tag{119}$$

Remark 4.1 *In order to prove optimal results on convergence, the basic theoretical restriction on the refinement procedure is that the refinement regions from each level forms a* nested *sequence. Let Ω_j denote the refinement region, namely, the union of the supports of basis functions which are introduced at level j. Due to nested refinement $\Omega_j \subset \Omega_{j-1}$. Then the following nested hierarchy holds:*

$$\Omega_J \subset \Omega_{J-1} \subset \cdots \subset \Omega_0 = \Omega.$$

Simply, the restriction indicates that tetrahedra of level j which are not candidates for further refinement will never be touched in the future. In practice, depending on the situation, the above nestedness restriction may or may not be enforced. We enforce the restriction in Lemma 4.1. In realistic situations, it is typically not enforced.

4.2. Matrix Representations and Local Smoothing

We describe how to construct the matrix representation of the preconditioners under consideration. Let the prolongation operator from level $j-1$ to j be denoted by

$$P_{j-1}^{j} \in \mathbb{R}^{\tilde{N}_j \times \tilde{N}_{j-1}},$$

and also denote the prolongation operator from level j to J as:

$$P_j \equiv P_j^J = P_{J-1}^J \ldots P_j^{j+1} \in \mathbb{R}^{N_J \times \tilde{N}_j},$$

where P_J^J is defined to be the rectangular identity matrix $I \in \mathbb{R}^{N_J \times \tilde{N}_{J-1}}$. Then the matrix representation of (116) becomes [108]:

$$X = \sum_{j=0}^{J} 2^{j(d-2)} P_j P_j^t.$$

One can also introduce a version with an explicit smoother G_j:

$$X = \sum_{j=0}^{J} P_j G_j P_j^t.$$

Throughout this article, the smoother $G_j \in \mathbb{R}^{\tilde{N}_j \times \tilde{N}_j}$ is a symmetric Gauss-Seidel iteration. Namely, $G_j = (D_j + U_j)^{-1} D_j (D_j + L_j)^{-1}$ where $A_j = D_j + L_j + U_j$ with $\tilde{N}_0 = N_0$.

The matrix representation of (117) is formed from matrices H_j which are simply the tails of the P_j corresponding to newly introduced DOF in the fine space. In other words, $H_j \in \mathbb{R}^{N_J \times (N_j - N_{j-1})}$ is given by only keeping the fine columns (the last $N_j - N_{j-1}$ columns of P_j). Hence, the matrix representation of (117) becomes:

$$X_{\text{HB}} = \sum_{j=0}^{J} 2^{j(d-2)} H_j H_j^t.$$

If the sum over i in (116) is restricted only to those nodal basis functions with supports that intersect the refinement region [109, 100, 102, 110], then we obtain the set called as *onering* of fine DOF. Namely, the set which contains fine DOF and their immediate neighboring coarse DOF:

$$\text{ONERING}^{(j)} = \{onering(ii) : \ ii = N_{j-1} + 1, \ldots, N_j\},$$

where $onering(ii) = \{ii, fathers(ii)\}$. Now, the generic preconditioner (119) for local refinement transforms into the following preconditioner:

$$Xu = \sum_{j=0}^{J} 2^{j(d-2)} \sum_{i \in \text{ONERING}^{(j)}} (u, \phi_i^{(j)}) \phi_i^{(j)}, \quad u \in \mathcal{S}_J. \tag{120}$$

There are three popular choices for $\tilde{\mathcal{N}}_j$. We outline possible BPX choices by the following DOF corresponding to:

- **(DOF-1)** The basis functions with supports that intersect the refinement region Ω_j [109, 100, 102]. We call this set $\text{ONERING}^{(j)}$.
- **(DOF-2)** The basis functions with supports that are contained in Ω_j [110].
- **(DOF-3)** Created by red refinement and their corresponding coarse DOF.

Here, red refinement refers to quadrasection or octasection in 2D and 3D, respectively. Green refinement simply refers to bisection.

The interesting ones are DOF-1 and DOF-3 and we would like to elaborate on these. In the numerical experiments reported in [105], DOF-1 was used. For the provably optimal computational complexity result in Lemma 4.1 DOF-3 is used.

4.2.1. The Sets DOF-1, DOF-3 and Local Smoothing Computational Complexity The set DOF-1 can be directly determined by the sparsity pattern of the fine-fine subblock $A_{22}^{(j)}$ of the stiffness matrix in (123). Then, the set of DOF over which the BPX preconditioner (120) smooths is simply the union of the column locations of nonzero entries corresponding to fine DOF. Using this observation, HB smoother can easily be modified to be a BPX smoother.

DOF-3 is equivalent to the following set:

$$\tilde{\mathcal{N}}_j = \{i = N_{j-1} + 1, \ldots, N_j\} \bigcup \{i : \phi_i^{(j)} \neq \phi_i^{(j-1)}, \quad i = 1, \ldots, N_{j-1}\}, \tag{121}$$

and the corresponding space over which the smoother acts:

$$\tilde{\mathcal{S}}_j = \text{span} \left[\bigcup \{\phi_i^{(j)}\}_{i=N_{j-1}+1}^{N_j} \bigcup \{\phi_i^{(j)} \neq \phi_i^{(j-1)}\}_{i=1}^{N_{j-1}} \right].$$

This set is used in the Bornemann-Erdmann-Kornhuber (BEK) refinement [111] and we utilize this set for the estimates of 3D local refinement. Since green refinement simply bisects a simplex, the modified basis function is the same as the one before the bisection due to linear interpolation. So the set of DOF in (121) corresponds to DOF created by red refinement and corresponding coarse DOF (father DOF). The following crucial result from [111] establishes a bound for the number of nodes used for smoothing. This indicates that the BPX preconditioner has provably optimal (linear) computational complexity per iteration on the resulting 3D mesh produced by the BEK refinement procedure.

Lemma 4.1 *The total number of nodes used for smoothing satisfies the bound:*

$$\sum_{j=0}^{J} \tilde{N}_j \leq \frac{5}{3} N_J - \frac{2}{3} N_0. \tag{122}$$

Proof. See [111, Lemma 1]. □

The above lemma constitutes the first computational complexity optimality result in 3D for the BPX preconditioner as reported in [103]. A similar result for 2D red-green refinement was given by Oswald [110, page95]. In the general case of local smoothing operators which involve smoothing over newly created basis functions plus some additional set of local neighboring basis functions, one can extend the arguments from [111] and [110] using shape regularity.

4.3. Hierarchical Basis Methods and Their Stabilizations

HB methods exploit a 2-level hierarchical decomposition of the DOF. They are divided into the coarse (the ones inherited from previous levels) and the fine (the ones that are newly introduced) nodes. In fact, in the operator setting, this decomposition is a direct consequence of the direct decomposition of the finite element space as follows:

$$\mathcal{S}_j = \mathcal{S}_{j-1} \oplus \mathcal{S}_j^f.$$

Hence, A_j can be represented by a two-by-two block form:

$$A_j = \left[\begin{array}{cc} A_{j-1} & A_{12}^{(j)} \\ A_{21}^{(j)} & A_{22}^{(j)} \end{array} \right] \begin{array}{ll} \} & \mathcal{S}_{j-1} \\ \} & \mathcal{S}_j^f \end{array}, \tag{123}$$

where A_{j-1}, $A_{12}^{(j)}$, $A_{21}^{(j)}$, and $A_{22}^{(j)}$ correspond to coarse-coarse, coarse-fine, fine-coarse, and fine-fine discretization operators respectively. The same 2-level decomposition carries directly to the matrix setting.

As mentioned earlier, HB methods suffer from the condition number growth. This makes makes HB methods especially ineffective in the 3D local refinement setting. As we mentioned earlier, wavelet-like stabilizations of the HB methods [101] provide a final solution to the condition number growth problem. The motivation for the stabilization hinges on the following idea. The BPX decomposition gives rise to basis functions which are not locally supported, but they decay rapidly outside a local support region. This allows for locally supported *approximations* as illustrated in Figures 3 and 4.

The *wavelet modified hierarchical basis (WMHB) methods* [101, 112, 113] can be viewed as an approximation of the wavelet basis stemming from the BPX

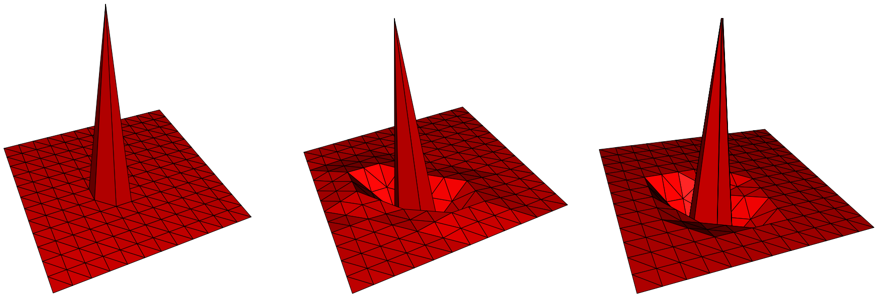


Figure 3. Left: Hierarchical basis function without modification. Wavelet modified hierarchical basis functions. Middle: One iteration of symmetric Gauss-Seidel approximation. Right: One iteration of Jacobi approximation.

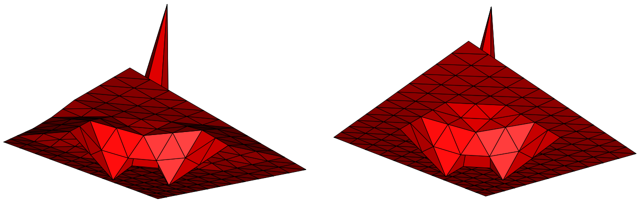


Figure 4. Lower view of middle and left basis functions in Figure 3.

decomposition [114]. A similar wavelet-like multilevel decomposition approach was taken in [115], where the orthogonal decomposition is formed by a discrete L_2-equivalent inner product. This approach utilizes the same BPX two-level decomposition [116, 115].

For local refinement setting, the other primary method of interest is the WMHB method. The WMHB methods can be described as additive or multiplicative Schwarz methods. In one of the previous papers [103], it was shown that the additive version of the WMHB method is optimal under certain types of red-green mesh refinement. Following the notational framework in [103, 105, 107, 113], this method is defined recursively as follows:

Definition 4.1 *The additive WMHB method $D^{(j)}$ is defined for $j = 1, \ldots, J$ as*

$$D^{(j)} \equiv \left[\begin{array}{ll} D^{(j-1)} & 0 \\ 0 & B_{22}^{(j)} \end{array} \right],$$

with $D^{(0)} = A^{(0)}$.

With smooth PDE coefficients, optimal results were also established for the multiplicative version of the WMHB method in [103, 107]. Our numerical experiments demonstrate such optimal results. This method can be written recursively as:

Definition 4.2 *The multiplicative WMHB method $B^{(j)}$ is defined as*

$$B^{(j)} \equiv \left[\begin{array}{ll} B^{(j-1)} & A_{12}^{(j)} \\ 0 & B_{22}^{(j)} \end{array}\right] \left[\begin{array}{ll} I & 0 \\ B_{22}^{(j)^{-1}} A_{21}^{(j)} & I \end{array}\right] = \left[\begin{array}{ll} B^{(j-1)} + A_{12}^{(j)} B_{22}^{(j)^{-1}} A_{21}^{(j)} & A_{12}^{(j)} \\ A_{21}^{(j)} & B_{22}^{(j)} \end{array}\right],$$

with $B^{(0)} = A^{(0)}$.

$B_{22}^{(j)}$ denotes an approximation of $A_{22}^{(j)}$, e.g. Gauss-Seidel or Jacobi approximation. For a more complete description of these and related algorithms, see [103, 107].

5. Practical Implementation of Fast Solvers

The overall utility of any finite element code depends strongly on efficient implementation of its core algorithms and data structures. Finite element method becomes a viable tool in addressing realistic simulations only when these critical pieces come together. Theoretical results involving complexity are of little practical importance if the methods cannot be implemented. For algorithms involving data structures, this usually means striking a balance between storage costs and computational complexity. For instance, finding a minimal representation for a data set is only useful if the information can be accessed efficiently. We elaborate on the data structure and the implementation of the methods under consideration.

5.1. Implementation of Hierarchical Basis Methods

In HB methods, nodal basis functions are transformed into hierarchical basis functions via a nonsingular change of basis matrix:

$$Y = \left[\begin{array}{cc} I & Y_{12} \\ Y_{21} & I + Y_{22} \end{array}\right],$$

where $Y \in \mathbb{R}^{N_j \times N_j}$, $Y_{12} \in \mathbb{R}^{N_{j-1} \times n_j}$, $Y_{21} \in \mathbb{R}^{n_j \times N_{j-1}}$, and $Y_{22} \in \mathbb{R}^{n_j \times n_j}$. We denote the number of fine DOF at level j as n_j. Then, $N_j = n_1 + n_2 + \cdots + n_j$ is the total number of DOF at level j. In this representation, we have assumed that the nodes are ordered with the nodes N_{j-1} inherited from the previous level listed first, and the n_j new DOF listed second. For both wavelet modified (WMHB) and unmodified hierarchical basis (HB), the Y_{21} block represents the last n_j rows of the prolongation matrix;

$$P_{j-1}^{j} = \left[\begin{array}{c} I \\ Y_{21}^{(j)} \end{array}\right].$$

In the HB case, the Y_{12} and Y_{22} blocks are zero resulting in a very simple form:

$$Y_{\rm hb} = \left[\begin{array}{cc} I & 0 \\ Y_{21} & I \end{array}\right] \tag{124}$$

Then, the original system (107) is related to the HB system through Y as follows.

$$\begin{aligned} A_{{\rm h}b}\, U_{{\rm h}b} &= F_{{\rm h}b}, \qquad (125) \\ A_{{\rm h}b} &= Y^T A Y, \\ U &= Y\, U_{{\rm h}b}, \\ F_{{\rm h}b} &= Y^T\, F. \end{aligned}$$

It will be shown later that the HBMG algorithm operates on sublocks of the HB system (125). We explicitly express the sublocks as follows.

$$\begin{bmatrix} A_{\mathrm{hb},11} & A_{\mathrm{hb},12} \\ A_{\mathrm{hb},21} & A_{\mathrm{hb},22} \end{bmatrix} = \begin{bmatrix} I & Y_{21}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ Y_{21} & I \end{bmatrix},$$

$$\begin{array}{lcccccccc} A_{\mathrm{hb},11} & = & A_{11} & + & A_{12}Y_{21} & + & Y_{21}^T A_{21} & + & Y_{21}^T A_{22} Y_{21} \\ A_{\mathrm{hb},12} & = & & & A_{12} & & & + & Y_{21}^T A_{22} \\ A_{\mathrm{hb},21} & = & & & & & A_{21} & + & A_{22}Y_{21} \\ A_{\mathrm{hb},22} & = & & & & & & & A_{22}, \end{array}$$

Next, we briefly describe the HBMG Algorithm. The HBMG routine can interpreted as an iterative process for solving the system (107) with an initial guess of U_j.

Algorithm 3 .

(i) *smooth* $A_{\mathrm{hb},22}U_{\mathrm{hb},2} = F_{\mathrm{hb},2} - (A_{\mathrm{hb},21}U_{\mathrm{hb},1})$

(ii) *form residual* $R_{\mathrm{hb},1} = F_{\mathrm{hb},1} - (A_{\mathrm{hb},11}U_{\mathrm{hb},1}) - A_{\mathrm{hb},12}U_{\mathrm{hb},2}$

(iii) *solve* $A_{\mathrm{hb},11}U_{\mathrm{hb},1} = R_{\mathrm{hb},1}$

(iv) *prolongate* $U_{\mathrm{hb}} = U_{\mathrm{hb}} + PU_{\mathrm{hb},1}$

(v) *smooth* $A_{\mathrm{hb},22}U_{\mathrm{hb},2} = F_{\mathrm{hb},2} - (A_{\mathrm{hb},21}U_{\mathrm{hb},1})$

Smoothing involves the approximate solution of the linear system by a fixed number of iterations (typically one or two) with a method such as Gauss-Seidel, or Jacobi. In order to use HBMG as a preconditioner for CG, one has to make sure the pre-smoother is the adjoint of the post-smoother. One should also note that the algorithm can be simplified by first transforming the linear system (107) into the equivalent system

$$A(U - U_j) = F - AU_j.$$

In this setting, the initial guess is zero, and the HBMG algorithm recursively iterates towards the *error* with given *residual* on the right hand side. Simplification comes from the fact that terms in the parentheses are zero in Algorithm 3.

5.1.1. The Computational and Storage Complexity of the HBMG method Utilizing the block structure of the stiffness matrix $A^{(j)}$ in (123), the storage is in the following fashion:

$$A^{(j)} = \begin{bmatrix} \begin{bmatrix} A_{11}^{(j)} & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{bmatrix}_{N_{j-1}\times N_{j-1}} & \begin{bmatrix} A_{12}^{(j)} & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}_{N_{j-1}\times n_j} \\ \begin{bmatrix} A_{21}^{(j)} & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{bmatrix}_{n_j\times N_{j-1}} & \begin{bmatrix} A_{22}^{(j)} & \rightarrow \\ \rightarrow & \rightarrow \end{bmatrix}_{n_j\times n_j} \end{bmatrix},$$

where → indicates a block stored rowwise, and × indicates a block which is not stored. By the symmetry in the bilinear form $\langle DF(u)w, v\rangle$ wrt w and v, $A_{12}^T = A_{21}$, hence A_{12} does not need to be stored.

Just like MG, the HBMG is an algebraic multilevel method as well. Namely, coarser stiffness matrices are formed algebraically through the use of variational conditions

$$A^{(j-1)} = {P^{j}_{j-1}}^{T} A^{(j)} P^{j}_{j-1}, \quad j = 1, \ldots, J; \quad A := A^{(J)}, \tag{126}$$

where P^{j}_{j-1} denotes the prolongation operator from level $j-1$ to j. Then, the only matrix to be stored for HB method is P^{j}_{j-1}, in which I is implicit, and therefore, does not have to be stored;

$$P^{j}_{j-1} = \begin{bmatrix} \begin{bmatrix} I & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}_{N_{j-1} \times N_{j-1}} \\ \begin{bmatrix} Y^{(j)}_{21} & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{bmatrix}_{n_j \times N_{j-1}} \end{bmatrix}.$$

In an adaptive scenario, new DOF are introduced in parts of the mesh where further correction or enrichment is needed. Naturally, the elements which are marked by the error estimator shrink in size by subdivision and the basis functions corresponding to fine nodes are forced to change more rapidly than the ones that correspond to the coarse nodes. Such rapidly changing components of the solution are described as the *high frequency* components. *Smoothing* is an operation which corrects the high frequency components of the solution, and is an integral part of the MG-like solvers. In MG, all DOF are exposed to smoothing which then requires to store all blocks of the stiffness matrix A. Unlike MG, the distinctive feature of the HBMG is that smoothing takes places only on basis functions corresponding to fine nodes. This feature leads us to make the crucial observation that V-cycle MG exactly becomes the HBMG method if smoothing is replaced by fine smoothing. One can describe HBMG style smoothing as *fine smoothing.* For fine smoothing, HBMG only needs to store the *fine-fine* interaction block A_{22} of A. It is exactly the fine smoothing that allows HB methods to have optimal storage complexity.

In a typical case, n_j is a small constant relative to N_j and has no relation to it. The HB method storage superiority stems from the fact that on every level j the storage cost is $\mathcal{O}(n_j)$. Fine smoothing is used for high frequency components, and this requires to store A_{22} block which is of size $n_j \times n_j$. A_{22} is stored rowwise, and the storage cost is $\mathcal{O}(n_j)$. Coarse grid correction is used recursively for low frequency components, and this requires to store A_{12} and A_{21} blocks which are of size $N_{j-1} \times n_j$ and $n_j \times N_{j-1}$ respectively. Due to symmetry of the bilinear form, A^{T}_{12} block is substituted by A_{21}, and A_{21} is stored rowwise, and the storage cost is again $\mathcal{O}(n_j)$.

Further strength of HB methods is that computational cost per cycle is $\mathcal{O}(n_j)$ on each level j. The preprocessing computational cost, namely computation of $A_{\mathrm{h}b,11}, A_{\mathrm{h}b,12}, A_{\mathrm{h}b,21}, A_{\mathrm{h}b,22}$ is $\mathcal{O}(n_j)$. Hence, in an adaptive refinement scenario, the overall computational complexity is achieved to be $\mathcal{O}(N)$.

Let us observe by a fictitious example how MG fails to maintain optimal storage complexity. Let us assume that the finest level is J, then $N = N_J, \quad N_{J-1} = N - n_J$, and count the total storage at each level. Here we take complexity constants to be 1 for simplicity.

$$\begin{array}{lll}
\text{level } J & : & N \\
\text{level } J-1 & : & N - n_J \\
\vdots & & \vdots \\
\text{level } 2 & : & N - n_J - n_{J-1} - \cdots - n_3 \\
\text{level } 1 & : & N - n_J - n_{J-1} - \cdots - n_3 - n_2.
\end{array}$$

Adding up the cost at all levels, overall cost is:

$$(J)N - (J-1)n_J - (J-2)n_{J-1} - \cdots - 2n_3 - n_2. \tag{127}$$

Since n_j's are small constants, then (127) $\leq NJ$, implying that overall storage is $\mathcal{O}(NJ)$. If numerous refinements are needed, or precisely, in an asymptotic scenario (i.e. $J \to \infty$), the storage poses a severe problem.

5.2. Sparse Matrix Products and the WMHB Implementation

Our implementation relies on a total of four distinct sparse matrix data structures: compressed column (COL), compressed row (ROW), diagonal-row-column (DRC), and orthogonal-linked list (XLN). For detailed description of the data structures, the reader can refer to [105]. Each of these storage schemes attempts to record the location and value of the nonzeros using a minimal amount of information. The schemes differ in the exact representation which effects the speed and manner with which the data can be retrieved. XLN is an orthogonal-linked list data structure format which is the only dynamically “fillable” data structure used by our methods. By using variable length linked lists, rather than a fixed length array, it is suitable for situations where the total number of nonzeros is not known *a priori*.

The key preprocessing step in the hierarchical basis methods, is converting the “nodal” matrices and vectors into the hierarchical basis. This operation involves sparse matrix-vector and matrix-matrix products for each level of refinement. To ensure that this entire operation has linear cost, with respect to the number of unknowns, the per-level change of basis operations must have a cost of $\mathcal{O}(n_j)$, where $n_j := N_j - N_{j-1}$ is the number of “new” nodes on level j. For the traditional multigrid algorithm this is not possible, since enforcing the variational conditions operates on *all* the nodes on each level, not just the newly introduced nodes.

For WMHB, the Y_{12} and Y_{22} blocks are computed using the mass matrix, which results in the following formula:

$$Y_{\text{wmhb}} = \left[\begin{array}{cc} I & -\text{inv}\left[M_{\text{hb},11}\right] M_{\text{hb},12} \\ Y_{21} & I - Y_{21}\text{inv}\left[M_{\text{hb},11}\right] M_{\text{hb},12} \end{array} \right], \tag{128}$$

where the inv $[\cdot]$ is some approximation to the inverse which preserves the complexity. For example, it could be as simple as the inverse of the diagonal, or a low-order matrix polynomial approximation. The M^{hb} blocks are taken from the mass matrix in the HB basis:

$$M_{\text{hb}} = Y_{\text{hb}}^T M_{\text{nodal}} Y_{\text{hb}}. \tag{129}$$

For the remainder of this section, we restrict our attention to the WMHB case. The HB case follows trivially with the two additional subblocks of Y set to zero.

To reformulate the nodal matrix representation of the bilinear form in terms of the hierarchical basis, we must perform a triple matrix product of the form:

$$\begin{aligned} A^{(j)}_{\mathrm{wmhb}} &= Y^{(j)^T} A^{(j)}_{\mathrm{nodal}} Y^{(j)} \\ &= \left(I + Y^{(j)^T} \right) A^{(j)}_{\mathrm{nodal}} \left(I + Y^{(j)} \right). \end{aligned}$$

In order to keep linear complexity, we can only copy A_{nodal} a fixed number of times, i.e. it cannot be copied on every level. Fixed size data structures are unsuitable for storing the product, since predicting the nonzero structure of $A^{(j)}_{\mathrm{wmhb}}$ is just as difficult as actually computing it. It is for these reasons that we have chosen the following strategy: First, copy A_{nodal} on the finest level, storing the result in an XLN which will eventually become A_{wmhb}. Second, form the product pairwise, contributing the result to the XLN. Third, the last n_j columns and rows of A_{wmhb} are stripped off, stored in fixed size blocks, and the operation is repeated on the next level, using the A_{11} block as the new A_{nodal}:

Algorithm 4 *(Wavelet Modified Hierarchical Change of Basis)*

- *Copy* $A^{(J)}_{\mathrm{nodal}} \rightarrow A_{\mathrm{wmhb}}$ *in XLN format.*
- *While* $j > 0$
    - *(i) Multiply* $A_{\mathrm{wmhb}} = A_{\mathrm{wmhb}} Y$ *as*

    $$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} += \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

    - *(ii) Multiply* $A_{\mathrm{wmhb}} = Y^T A_{\mathrm{wmhb}}$ *as*

    $$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} += \begin{bmatrix} 0 & Y^T_{21} \\ Y^T_{12} & Y^T_{22} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

    - *(iii) Remove* $A^{(j)}_{21}$, $A^{(j)}_{12}$, $A^{(j)}_{22}$ *blocks of* A_{wmhb} *storing in ROW, COL, and DRC formats respectively.*
    - *(iv) After the removal, all that remains of* A_{wmhb} *is its* $A^{(j)}_{11}$ *block.*
    - *(v) Let j = j - 1, descending to level* $j - 1$.
- *End While.*
- *Store the last* A_{wmhb} *as* A_{coarse}

We should note that in order to preserve the complexity of the overall algorithm, all of the matrix-matrix algorithms must be carefully implemented. For example, the change of basis involves computing the products of A_{11} with Y_{12} and Y^T_{12}. To preserve storage complexity, Y_{12} must be kept in compressed column format, COL. For the actual product, the loop over the columns of Y_{12} must be ordered first, then a loop over the nonzeros in each column, then a loop over the corresponding row or column in A_{11}. It is exactly for this reason, that one must be able to traverse A_{11} both by row and by column, which is why we have chosen an orthogonal-linked matrix structure for A during the change of basis (and hence A_{11}).

To derive optimal complexity algorithms for the other products, it is enough to ensure that the outer loop is always over a dimension of size n_j. Due to the limited ways in which a sparse matrix can be traversed, the ordering of the remaining loops will usually be completely determined. Further gains can be obtained in the symmetric case, since only the upper or lower portion of the matrix needs to be explicitly computed and stored.

6. The Finite Element ToolKit for the Einstein Constraints

FETK [26] (see also [29, 30, 31]) is an adaptive multilevel finite element code in ANSI C developed by one of the authors (M.H.) over several years at Caltech and UC San Diego. It is designed to produce provably accurate numerical solutions to nonlinear covariant elliptic systems of tensor equations on 2- and 3-manifolds in an optimal or nearly-optimal way. FETK employs *a posteriori* error estimation, adaptive simplex subdivision, unstructured algebraic multilevel methods, global inexact Newton methods, and numerical continuation methods for the highly accurate numerical solution of nonlinear covariant elliptic systems on (Riemannian) 2- and 3-manifolds. In this section, we describe some of the key design features of FETK. A sequence of careful numerical examples producing initial data for an evolution simulation appear in [117].

6.1. The overall design of FETK

The finite element kernel library in FETK is referred to as MC (or Manifold Code). MC is an implementation of Algorithm 1, employing Algorithm 2 for nonlinear elliptic systems that arise in Step 1 of Algorithm 1. The linear Newton equations in each iteration of Algorithm 2 are solved with algebraic multilevel methods, and the algorithm is supplemented with a continuation technique when necessary. Several of the features of FETK are somewhat unusual, allowing for the treatment of very general nonlinear elliptic systems of tensor equations on domains with the structure of 2- and 3-manifolds. In particular, 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). To use the *a posteriori* error estimator, a third function $F(u)$ must also be provided (essentially the strong form of the problem).
- *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. FETK 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, and through a `oneChart()` routine that the user provides to resolve chart boundaries.
- *Dimension independence*: The same code paths are taken for both two- and three-dimensional problems (as well as for higher-dimensional problems). To achieve this dimension independence, FETK 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 the constraints in the Einstein equations requires writing only 1000 lines of C to define the two weak forms, and to define the `oneChart()` routine. Changing to a different problem, e.g., large deformation nonlinear elasticity, involves providing only a different definition of the forms and a different domain description.

6.2. Topology and geometry representation in FETK

A datastructure called the *ringed-vertex* (cf. [26]) is used to represent meshes of d-simplices of arbitrary topology. This datastructure is illustrated in Figure 5. The ringed-vertex datastructure is somewhat similar to the winged-edge, quad-edge, and edge-facet datastructures commonly used in the computational geometry community for representing 2-manifolds [118], but it can be used more generally to represent arbitrary d-manifolds, $d = 2, 3, \ldots$. 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 discretization of an elliptic operator. The ringed-vertex datastructure also allows for dimension independent implementations of mesh refinement and mesh manipulation, with one implementation covering arbitrary dimension d. An interesting feature of this datastructure is that the C structures used for vertices, simplices, 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 [26].

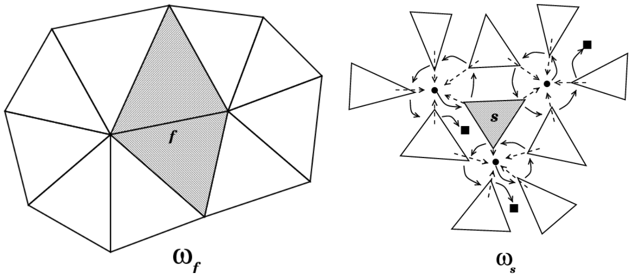


Figure 5. Polyhedral manifold representation. The figure on the left shows two overlapping polyhedral (vertex) charts consisting of the two rings of simplices around two vertices sharing an edge. The region consisting of the two darkened triangles around the face f is denoted ω_f, and represents the overlap of the two vertex charts. Polyhedral manifold topology is represented by FETK using the *ringed vertex* 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. (The set ω_s includes s itself.) Edges are temporarily created during subdivision but are then destroyed (a similar ring datastructure is used to represent the edge topology).

Since FETK is based entirely on the d-simplex, for adaptive refinement it employs simplex bisection, using one of the simplex bisection strategies outlined earlier. Bisection is first used to refine an initial subset of the simplices in the mesh (selected according to some error estimates, discussed below), and then a closure algorithm is performed in which bisection is used recursively on any non-conforming simplices, until a conforming mesh is obtained. If it is necessary to improve element shape, FETK attempts to optimize the following simplex shape measure function for a given d-simplex s, in an iterative fashion, similar to the approach taken in [119]:

$$\eta(s,d) = \frac{2^{2(1-\frac{1}{d})} 3^{\frac{d-1}{2}} |s|^{\frac{2}{d}}}{\sum_{0 \le i < j \le d} |e_{ij}|^2}. \tag{130}$$

The quantity $|s|$ represents the (possibly negative) volume of the d-simplex, and $|e_{ij}|$ represents the length of the edge that connects vertex i to vertex j in the simplex. For $d = 2$ this is the shape-measure used in [119] with a slightly different normalization. For $d = 3$ this is the shape-measure developed in [120] again with a slightly different normalization. The shape measure above can be shown to be equivalent to the sphere ratio shape measure commonly used; the sphere measure is used in the black hole mesh generation algorithms we describe below.

6.3. Discretization and adaptivity in FETK

Given a nonlinear form $\langle F(u), v\rangle$, its linearization bilinear form $\langle DF(u)w, v\rangle$, a Dirichlet function $\bar{u}$, and collection of simplices representing the domain, FETK uses a default linear element to produce and then solve the implicitly defined nonlinear algebraic equations for the basis function coefficients in the expansion (87). The user can also provide his own element, specifying the number of degrees of freedom to be located on vertices, edges, faces, and in the interior of simplices, along with a quadrature rule, and the values of the basis functions at the quadrature points on the master element. Different element types may be used for different components of a coupled elliptic system. The availability of a user-defined general element makes it possible to, for example, use quadratic elements, which in the present setting would allow for the differentiation of the resulting solutions to the constraints for use as initial data in an evolution code.

Once the equations are assembled and solved (discussed below), *a posteriori* error estimates are computed from the discrete solution to drive adaptive mesh refinement. The idea of adaptive error control in finite element methods is to estimate the behavior of the actual solution to the problem using only a previously computed numerical solution, and then use the estimate to build an improved numerical solution by upping the polynomial order (p-refinement) or refining the mesh (h-refinement) where appropriate. Note that this approach to adapting the mesh (or polynomial order) to the local solution behavior affects not only approximation quality, but also solution complexity: if a target solution accuracy can be obtained with fewer mesh points by their judicious placement in the domain, the cost of solving the discrete equations is reduced (sometimes dramatically) because the number of unknowns is reduced (again, sometimes dramatically). Generally speaking, if an elliptic equation has a solution with local singular behavior, such as would result from the presence of abrupt changes in the coefficients of the equation (e.g., the conformal metric components in the present case), then adaptive methods tend to give dramatic improvements over non-adaptive methods in terms of accuracy achieved for a given complexity price. Two examples illustrating bisection-based adaptivity patterns (driven by a completely geometrical "error" indicator simply for illustration) are shown in Figure 6.

To drive the adaptivity in FETK, we employ a residual error estimator, based on computing an upper bound on the nonlinear residual as given in (110). Reference [26] contains the detailed derivation of the indicator (110) used in FETK for a general nonlinear elliptic system of tensor equations of the form (20)–(22). The error estimator provides a bound on the error in the $W^{1,p}$-norm, $1 < p < \infty$, for an approximation of the form (85) to the solution of the weak formulation (30).

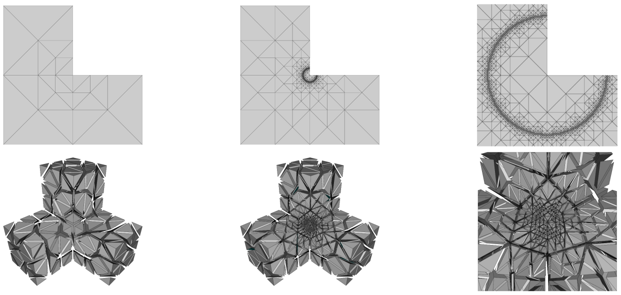


Figure 6. Examples illustrating the 2D and 3D adaptive mesh refinement algorithms in FETK. The right-most figure in each row shows a close-up of the area where most of the refinement occurred in each example.

6.4. Solution of linear and nonlinear systems with FETK

When a system of nonlinear finite element equations must be solved in FETK, the global inexact-Newton Algorithm 2 is employed, where the linearization systems are solved by linear multilevel methods. When necessary, the Newton procedure in Algorithm 2 is supplemented with a user-defined normalization equation for performing an augmented system continuation algorithm. The linear systems arising as the Newton equations in each iteration of Algorithm 2 are solved using a completely algebraic multilevel algorithm. Either refinement-generated prolongation matrices P^j_{j-1}, or user-defined prolongation matrices P^j_{j-1} in a standard YSMP-row-wise sparse matrix format, are used to define the multilevel hierarchy algebraically. In particular, once the single "fine" mesh is used to produce the discrete nonlinear problem $F(u) = 0$ along with its linearization $Au = f$ for use in the Newton iteration in Algorithm 2, a J-level hierarchy of linear problems is produced algebraically using the variational conditions (126) recursively. As a result, the underlying multilevel algorithm is provably convergent in the case of self-adjoint-positive matrices [121]. Moreover, the multilevel algorithm has provably optimal $O(N)$ convergence properties under the standard assumptions for uniform refinements [86], and is nearly-optimal $O(N \log N)$ under very weak assumptions on adaptively refined problems [97].

Coupled with the superlinear convergence properties of the outer inexact Newton iteration in Algorithm 2, this leads to an overall complexity of $O(N)$ or $O(N \log N)$ for the solution of the discrete nonlinear problems in Step 1 of Algorithm 1. Combining this low-complexity solver with the judicious placement of unknowns only where needed due to the error estimation in Step 2 and the subdivision algorithm in Steps 3-6 of Algorithm 1, leads to a very effective low-complexity approximation technique for solving a general class of nonlinear elliptic systems on 2- and 3-manifolds.

6.5. Availability of FETK

A fully-functional MATLAB version of the MC kernel of FETK for domains with the structure of Riemannian 2-manifolds, called MCLite, is available under the GNU copyleft license at the following website:

`http://www.FEtk.ORG/`

MCLite employs the ringed-vertex datastructure and implements the same adaptivity and solution algorithms that are used in MC. A number of additional tools developed for use with MC and MCLite are also available under a GNU license at the site above, including MALOC (a Minimal Abstraction Layer for Object-oriented C) and SG (an OpenGL-based X11/Win32 polygon display tool with a linear programming-based OpenGL-to-Postscript generator). SG was used to generate the pictures of finite element meshes appearing in this paper.

6.6. Tetrahedral Mesh Generation for Single or Binary Compact Objects

The general problem of generating a finite element mesh given a three dimensional domain is quite complex and has been the subject of intensive research over the past 40 years (for a relatively old comprehensive overview the reader is referred to [122]). However, for the purpose of generating a tetrahedral mesh on a domain with the geometry necessary to describe a binary collision between compact objects we do not need a completely general method. In this section we will describe a rather simple method for generating high quality coarse meshes suitable for the computation of the initial data describing binary systems. The only restriction of the method is that the geometry of the domain must have an axis of symmetry. We note that this restriction applies only to the domain geometry and not the physics (i.e., the source terms in the constraint equations do not have to have any symmetries).

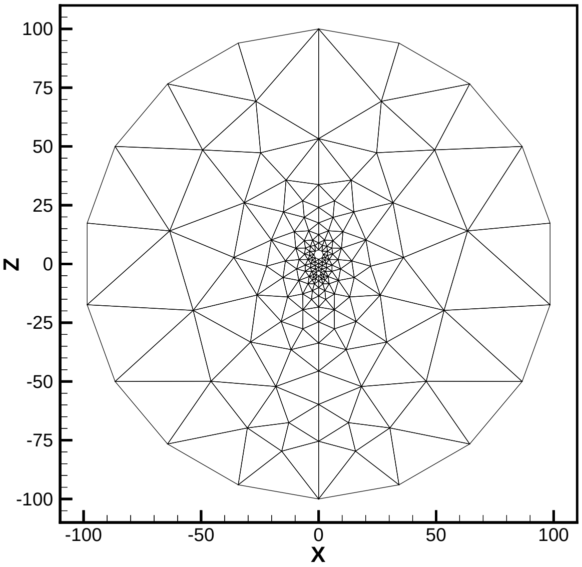


Figure 7. The planar triangulation step in the construction of a finite element mesh for a binary black hole collision. The figure shows the discretization of the entire domain; due to the mirror symmetry of the domain only the region $x \geq 0$ was meshed.

A typical domain for a binary black hole collision consists of the interior of a large sphere (the outer boundary) with two smaller spheres removed (the throats of the holes). Since the exact placement of the outer sphere relative to the throats is not important we lose no generality in making the centers of the three spheres colinear. In this case the geometry is symmetric about the line joining the three centers. Choose this line to be the z axis. Then the intersection of the domain with the x-z plane consists of a circle with two smaller circles removed from its interior. The method consists of first meshing this planar domain with triangular elements and then rotating the elements around the z-axis to form triangular tori. These tori are then subdivided into tetrahedra in such a way as to produce a conforming mesh.

We illustrate the method with an example. Let the outer boundary have radius 100 and let the two throats be centered at $z = \pm 4$ with radii 2 and 1 respectively. The geometry is chosen for illustrative purposes. We discretize the outer circle of the planar domain with 20 line segments and the inner circles with 8 and 10 line segments respectively. Two views of the triangulation are shown in figures 7 and 8.

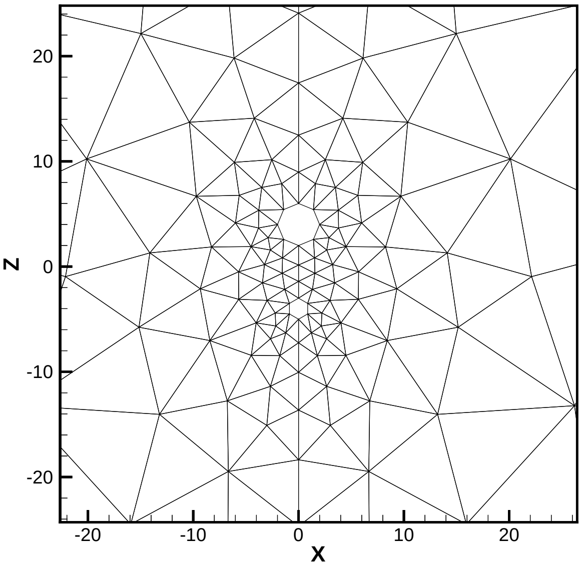


Figure 8. A closeup of the mesh from Figure 7 showing the details of the mesh around the two holes.

The triangulation algorithm one employs for this part of the mesh generation is not important. In particular, one could use any one of a number of off-the-shelf codes freely available on the internet (for example, see `www-2.cs.cmu.edu/∼quake/triangle.html`). A list of commercial and free mesh generators can be found at
`www-users.informatik.rwth-aachen.de/∼roberts/software.html`.
The triangulation code we have chosen to use is one developed by one of the authors (D.B.). We have the ability to produce highly graded meshes since the inner circles of the planar domain will in general be much small than the outer boundary circle. We would like to emphasize the distinguishing feature of the triangulation code. Although there are various codes that generate highly grades meshes with low quality, the code developed by D.B. is superior than those because it generates highly graded meshes

with high quality.

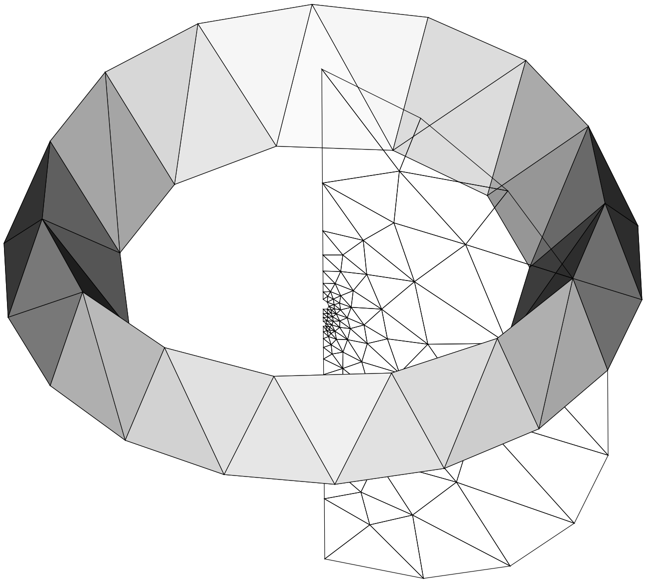


Figure 9. The second stage of mesh generation. One of the discretized tori is shown along with the planar mesh from figure 7.

Once the planar triangulation is obtained the tetrahedra are constructed as follows. Each vertex of the triangulation defines a circle in the three-dimensional space by rotation about the z-axis. These circles are discretized by computing their circumference and dividing by a measure of the size of the desired tetrahedra. The measure we have used is the average distance to the neighbors of the vertex in the triangulation. For vertices with a neighbor on the axis of symmetry the discretization of the circle is limited to between five and seven line segments, regardless of the circumference or average neighbor distance. This limits the outdegree of the vertices which lie on the z-axis and prevents elements with unusually small angles from being formed there. Once each triangular torus is discretized along each of its edges it is a relatively simple matter to subdivide each torus into tetrahedra taking care that the elements thus formed are reasonably shaped and consistent with their neighbors. In figure 9 we show one of these tori and in figure 10 we show the surface of the final mesh.

The quality of the meshes produced by this method is normally very good. One measure of the quality of a tetrahedron which is often used is the ratio of the radii of its circumscribing to inscribing spheres, $\alpha = R/3r$. The normalization factor is such that a regular tetrahedron has $\alpha = 1$ and it has been proven [120] that this is a global minimum, i.e., no tetrahedron has $\alpha < 1$. The maximum aspect ratio in the mesh shown in figure 10 is 1.91 while the average is 1.26. By comparison the aspect ratio of the reference element with vertices located at $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ is 1.37.

Finally, we note that although the example above describes the generation of a mesh for a binary black hole simulation the method is not restricted to this type of problem. We can also generate meshes suitable for any type of neutron star - black

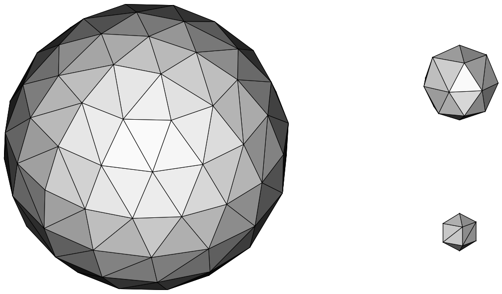


Figure 10. The final mesh generated from the planar domain shown in figure 7. On the left the surface of the outer boundary is shown, on the right the surfaces of the two holes.

hole collision by replacing one or both of the inner circles of the planar domain with a set of curves which are preserved on the mesh. The curves may represent (expected) isosurfaces of density or some other quantity. In Figures 11 and 12 we show the mesh generated for a collision between two neutron stars, one expected to be oblate and one expected to be prolate. Note that the initial data computed on such a mesh may or may not have stars of such shape, we have merely chosen a coarse mesh which we expect to reflect the nature of the solution to the constraints equations. The actual mesh which solves the numerical problem to a given tolerance may have a much different node distribution.

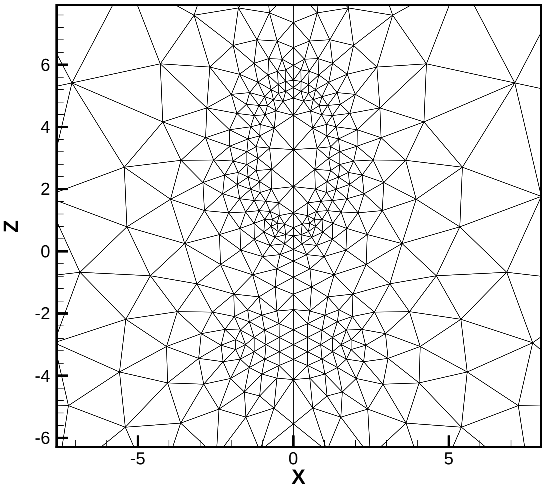


Figure 11. A mesh suitable for the calculation of the collision between two stars, one initially prolate and one initially oblate. The figure shows shows the planar triangulation about the two stars.

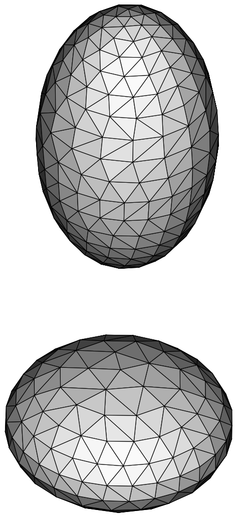


Figure 12. The “surfaces” of the two stars represented by the mesh in Figure 12.

6.7. Computing Conformal Killing Vectors

We describe a simple algorithm for computing conformal Killing vectors and their Petrov-Galerkin approximations. It can be shown that for Dirichlet and Robin boundary conditions the homogeneous version of the momentum constraint

$$\hat{D}_a(\hat{L}V)^{ab} = 0 \tag{131}$$

has only the trivial solution $V^a = 0$. However, using a pure Neumann condition removes the well-posedness of the problem and leads to a method for computing the conformal Killing vectors of $\hat{\gamma}_{ab}$. Suppose that K^a satisfies (131). This implies

$$0 = (\hat{D}_a(\hat{L}K)^{ab}, V^a)_{L^2(\mathcal{M})} = A(K^a, V^a), \tag{132}$$

$\forall\ V^a \in H^1_{0,D}$. This can be posed as the generalized eigenvalue problem

$$\text{Find}\ \ K^a \in H^1_{0,D} \text{ s. t. } F(K^a, V^a) = \epsilon G(K^a, V^a), \tag{133}$$

$\forall\ V^a \in H^1_{0,D}$, where

$$F(K^a, V^a) = \int_M 2\mu(\hat{E}K)_{ab}(\hat{E}V)^{ab}dx, \tag{134}$$

$$G(K^a, V^a) = \int_M \hat{D}_a K^a \hat{D}_b V^b dx. \tag{135}$$

If there exists a K^a such that this holds for some $\epsilon \neq 0$ then the operator $F - \epsilon G$ arising from the form through the Riesz representation theorem is singular. For a given domain with a tetrahedral decomposition, FETK can discretize (133) to produce the matrix versions of F and G which can then be given to a general eigenvalue package such as EISPACK. The result would be a set of eigenvalues and eigenvectors. The eigenvectors corresponding to the eigenvalue $\epsilon = 2/3$ form an (orthogonal) basis for the kernel of the discrete momentum operator, representing a discrete approximation to the space of conformal Killing vectors of $\hat{\gamma}_{ab}$.

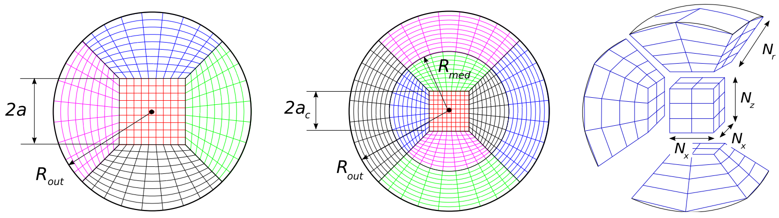
6.8. Computation of the ADM Mass on Adaptive Meshes

Here we describe an alternative expression for the ADM mass that is more appropriate for adaptive methods than that usually used in numerical relativity. The ADM mass is normally computed by the covariant surface integral

$$M = -\frac{1}{2\pi}\int_R \hat{n}_a \hat{\Delta}^a \phi \, ds \tag{136}$$

where R indicates the integral is to be taken over the outer sphere only. The mass is actually defined as the limit $R \to \infty$ and assumes that the equivalent energy associated with the conformal metric vanishes. However, if this is the case, and if the outer sphere is reasonably far away from the region where the source terms are varying rapidly then the adaptive nature of FETK will ensure that the outer sphere remains relatively coarsely meshed, even in cases where the solution is computed to a high degree of accuracy. This is because a $1/r$ variation in ϕ can be computed accurately with only a few elements. Hence the outer boundary will remain coarsely meshed and the surface integral above will not be able to be estimated with any degree of accuracy. However, by converting (136) to a volume integral using the Gauss theorem, and by then evaluating it along with any resulting surface integrals over inner boundaries, it is possible to gain from the locally adapted mesh rather than lose from it. In particular, one has:

$$M = -\frac{1}{2\pi}\left(\int_{\mathcal{M}} \hat{\Delta}\phi \, dx - \int_{\partial\mathcal{M}-R} \hat{n}_a \hat{\nabla}^a \phi \, ds\right). \tag{137}$$



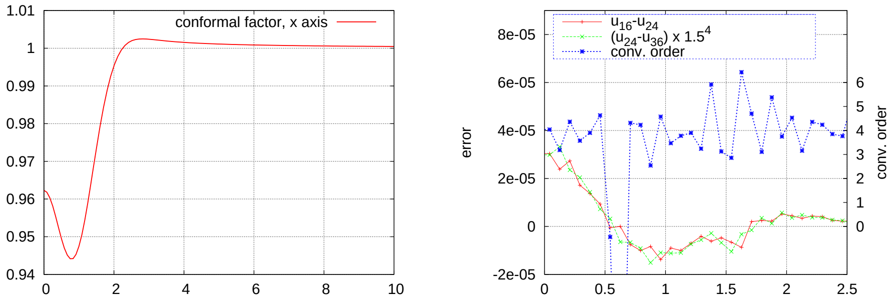
(a) (b) (c)

Figure 13. An equatorial cut of (a) seven-block and (b) thirteen-block systems. (c) Grid dimensions for the seven-block system.

Using the Hamiltonian constraint (14) this is

$$\begin{aligned} M = & -\frac{1}{2\pi}\int_{\mathcal{M}}\left(\frac{1}{8}\hat{R}\phi + \frac{1}{12}(\mathrm{tr}K)^2\phi^5\right. \qquad (138) \\ & \left. -\frac{1}{8}({}^*\hat{A}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3}\right) \, dx \qquad (139) \\ & + \frac{1}{2\pi}\int_{\partial\mathcal{M}-R} \hat{n}_a \hat{\nabla}^a \phi \, ds. \qquad (140) \end{aligned}$$

Normally the inner boundaries will be highly refined (although one can construct cases where this does not happen) and so this method will give much more accurate results than the simple use of (136).



(a)

(b)

Figure 14. (a) 1-d cut through the Brill wave conformal factor ψ_{fine} for problem (a) along the x-axis. (b) Errors $\psi_{coarse} - \psi_{medium}$, $\psi_{medium} - \psi_{fine}$, and pointwise convergence order on the 1-d cut along the x-axis.

6.9. Brill waves initial data on multi-block domains

As the final numerical experiment, we demonstrate the Brill wave initial data generated by using FETK on a multi-block spherical domain. The comprehensive treatment of this experiment can be found in [117].

In General Relativity, initial data on a spatial 3D-slice has to satisfy the Hamiltonian and momentum constraint equations [123],

$$\begin{aligned} {}^3R - K^{ij}K_{ij} + K^2 &= 0 \\ \nabla_i(K^{ij} - g^{ij}K) &= 0 \end{aligned}$$

where K_{ij} and K are the *extrinsic curvature* of the 3D-slice and its trace, respectively, and ${}^3R$ the Ricci scalar associated to the spatial metric g_{ij}.

Brill waves [124] constitute a simple yet rich example of initial data in numerical relativity. In such case the extrinsic curvature of the slice is zero, and the above equations reduce to a single one, stating that the Ricci scalar has to vanish:

$${}^3R = 0\,. \tag{141}$$

If the spatial metric is given up to one unknown function, Eq. (141) in principle allows to solve for such function and thus complete the construction of the initial data. The Brill equation is a special case of (141), where the 3-metric is expressed through the conformal transformation $g_{ij} = \psi^4 \tilde{g}_{ij}$ of an unphysical metric $\tilde{g}_{ij}$, with an unknown conformal factor ψ. Equation (141) then becomes [125]:

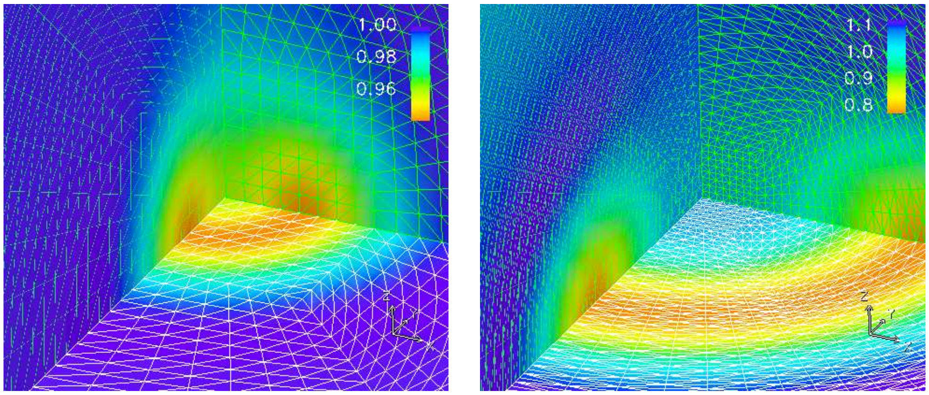
$$(-\nabla^2_{\tilde{g}} + \frac{1}{8}\tilde{R})\psi = 0 \tag{142}$$

where $\tilde{R}$ and $\nabla^2_{\tilde{g}}$ are the Ricci scalar curvature and Laplacian of the unphysical metric $\tilde{g}_{ij}$, respectively.

We work with two specific choices for $q(\rho, z)$:

(a) Holz' form [126]: $q_H(\rho, z) = a_H \rho^2 e^{-r^2}$, with amplitude $a_H = 0.5$;

(b) toroidal form: $q_t(\rho, z) = a_t \rho^2 \exp\left(-\frac{(\rho-\rho_0)^2}{\sigma_\rho^2} - \frac{z^2}{\sigma_z^2}\right)$, with amplitude $a_t = 0.05$, radius $\rho_0 = 5$, width in ρ-direction $\sigma_\rho = 3.0$ and width in z-direction $\sigma_z = 2.5$.



(a)

(b)

Figure 15. Potentials for the two types of Brill waves considered: Holz' (a) and toroidal (b) forms.

Here we will focus on the axisymmetric case with the conformal metric given in cylindrical coordinates by

$$\tilde{g}_{ij} = e^{2q(\rho,z)}(d\rho^2 + dz^2) + \rho^2 d\varphi^2\,, \tag{143}$$

where $q(\rho, z)$ is a function satisfying the following conditions:

(i) regularity at the axis: $q(\rho = 0, z) = 0$, $\frac{\partial q}{\partial \rho}|_{\rho=0} = 0$,

(ii) asymptotic flatness: $q(\rho, z)|_{r\to\infty} < O(1/r^2)$, where r is the spherical radius $r = \sqrt{\rho^2 + z^2}$.

The Hamiltonian constraint equation (141) becomes a second order elliptic PDE, which with asymptotically flat boundary conditions at $r \to \infty$ takes the form

$$-\nabla^2\psi(\rho, z) + V(\rho, z)\psi(\rho, z) = 0, \tag{144}$$

$$\psi|_{r\to\infty} = 1 + \frac{M}{2r} + O(1/r^2), \tag{145}$$

with the potential $V(\rho, z)$ given by

$$V = -\frac{1}{4}(q''_{\rho\rho} + q''_{zz}).$$

We numerically solve this equation using FETK on the 13-patch multi-block spherical domain (see figure 13). We use domain parameters $R_{out} = 30$, $R_{med} = 7$, $a_c = 1.5$, and grid dimension ratios $N : N_{r,inner} : N_{r,outer} = 2 : 3 : 12$. Our low-medium-high resolution triple is $N = 32$, $N = 36$ and $N = 40$, except for pointwise convergence tests on the x-axis (see figure 14), where we use $N = 16$, $N = 24$ and $N = 36$ (since they all differ by powers of 1.5).

7. Conclusion

In this paper we have described the use of finite element methods to construct numerical solutions to the initial value problem of general relativity. We first reviewed the classical York conformal decomposition, and gave a basic framework for deriving weak formulations. We briefly outlined the notation used for the relevant function spaces, and gave a simple weak formulation example. We then derived an appropriate symmetric weak formulation of the coupled constraint equations, and summarized a number of basic theoretical properties of the constraints. We also derived the linearization bilinear form of the weak form for use with stability analysis or Newton-like numerical methods. A brief introduction to adaptive finite element methods for nonlinear elliptic systems was then presented, and residual-type error indicators were derived. We presented several general *a priori* error estimates from [26, 27, 28] for general Galerkin approximations to solutions equations such as the momentum and Hamiltonian constraints. The numerical methods employed by MC were described in detail, including the finite element discretization, the residual-based *a posteriori* error estimator, the adaptive simplex bisection strategy, the algebraic multilevel solver, and the Newton-based continuation procedure for the solution of the nonlinear algebraic equations which arise. We described a mesh generation algorithm for modeling compact binary objects, outlined an algorithm for computing conformal Killing vectors, describes the numerical approximation of the ADM mass, and gave an example showing the use of MC for solution of the coupled constraints in the setting of a binary compact object collision.

The implementation of these methods in the ANSI C finite element code named MC was discussed in detail, including descriptions of the algorithms and data structures it employs. MC was designed specifically for solving general second-order nonlinear elliptic systems of tensor equations on Riemannian manifolds with boundary. The key feature of MC which makes it particularly useful for relativity applications is the unusually high degree of abstraction with which it can be used. The user need only supply two functions (one for a linear problem) in the form of a short C code file. These functions are generally coded exactly as the weak form of the equation and its linearization are written down (our initial data constraint specification in MC is close to a cut-and-paste of equations (76) and (78) into a C source file). The user does not have to provide the elliptic system in discrete form as is usually required in finite difference implementations, and does not normally have to supply detailed coefficient information. In particular, the user provides only the two forms $\langle F(u), v\rangle$ and $\langle DF(u)w, v\rangle$. If *a posteriori* error estimation is to be used, then the user must provide a third function $F(u)$, which is essentially the strong form of the differential equation as needed for the error estimator given in Section 3.4. MC is also able to handle systems on a manifold whose atlas has more than one chart.

Acknowledgments

M. Holst thanks K. Thorne, L. Lindblom, and H. Keller for many fruitful discussions over several years at Caltech. D. Bernstein thanks H. Keller and Caltech Applied Mathematics for their generous support during the initial phases of this research at Caltech. M. Holst was supported in part by NSF Awards 0715146 and 0511766, and DOE Awards DE-FG02-05ER25707 and DE-FG02-04ER25620.

B. Aksoylu and M. Holst thank O. Korobkin of Louisiana State University for providing the numerical results on Brill wave initial data.

References

[1] E. Seidel. New developments in numerical relativity. *Helvetica Physica Acta*, 69:454–471, 1996.

[2] G. Cook and S. Teukolsky. Numerical relativity: Challenges for computational science. In A. Iserles, editor, *Acta Numerica 8*, pages 1–44. Cambridge University Press, 1999.

[3] L. Lehner. Numerical relativity: A review. *Class. Quantum Grav.*, 18:R25–R86, 2001.

[4] L. Lindblom and M. Scheel. Dynamical gauge conditions for the Einstein evolution equations. *Phys. Rev. D*, 67:124005, 2003.

[5] G. Nagy, O. Orgiz, and O. Reula. Strongly hyperbolic second order Einstein's evolution equations. *Phys. Rev. D*, 70:044012, 2004.

[6] C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. W. H. Freeman and Company, San Francisco, CA, 1970.

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

[8] R. Bartnik and J. Isenberg. The constraint equations. In P.T. Chruściel and H. Friedrich, editors, *50 Years of the Cauchy Problem, in honour of Y. Choquet-Bruhat, 2002 Cargese Meeting*, 2003. to appear.

[9] J. Isenberg and N.O. Murchadha. Non cmc conformal data sets which do not produce solutions of the Einstein constraint equations. *Class. Quantum Grav.*, 21:S233–S242, 2004.

[10] J. Isenberg. Constant mean curvature solutions of the Einstein constraint equations on closed manifolds. *Class. Quantum Grav.*, 12:2249–2274, 1995.

[11] R. Bartnik and J. Isenberg. The constraint equations. In P. Chruściel and H. Friedrich, editors, *The Einstein equations and large scale behavior of gravitational fields*, pages 1–38. Birhäuser, Berlin, 2004.

[12] D. Maxwell. Rough solutions of the Einstein constraint equations on compact manifolds. *J. Hyp. Diff. Eqs.*, 2(2):521–546, 2005.

[13] D. Maxwell. Rough solutions of the Einstein constraint equations. *J. Reine Angew. Math.*, 590:1–29, 2006.

[14] P. Allen, A. Clausen, and J. Isenberg. Near-constant mean curvature solutions of the Einstein constraint equations with non-negative Yamabe metrics. Available as arXiv:0710.0725 [gr-qc], 2007.

[15] J. Isenberg and N. Ó Murchadha. Non CMC conformal data sets which do not produce solutions of the Einstein constraint equations. *Class. Quantum Grav.*, 21:S233–S242, 2004.

[16] M. Holst, J. Kommemi, and G. Nagy. Rough solutions of the Einstein constraint equations with nonconstant mean curvature. Preprint. Available as arXiv:0708.3410 [gr-qc].

[17] M. Holst, G. Nagy, and G. Tsogtgerel. Far-from-constant mean curvature solutions of Einstein's constraint equations with positive Yamabe metrics. *Physical Review Letters*, 100(16):161101.1–161101.4, 2008. Available as arXiv:0802.1031 [gr-qc].

[18] M. Holst, G. Nagy, and G. Tsogtgerel. Rough solutions of the Einstein constraints on closed manifolds without near-CMC conditions. Accepted for Publication in Comm. Math. Phys. Available as arXiv:0712.0798 [gr-qc].

[19] H. Pfeiffer. The initial value problem in numerical relativity. In *Proceedings of the Miami Waves 2004 Conference*, 2004.

[20] P. J. Mann. The search for stable finite element methods for simple relativistic systems. *Computer Physics Communications*, 30:127–137, 1983.

[21] P. J. Mann. A finite element method in space and time for relativistic spherical collapse. *Computer Physics Communications*, 75:10–30, 1993.

[22] D.N. Arnold, A. Mukherjee, and L. Pouly. Locally adapted tetrahedral meshes using bisection. *SIAM J. Sci. Statist. Comput.*, 22(2):431–448, 1997.

[23] A. Mukherjee. *An Adaptive Finite Element Code for Elliptic Boundary Value Problems in Three Dimensions with Applications in Numerical Relativity*. PhD thesis, Dept. of Mathematics, The Pennsylvania State University, 1996.

[24] C.-Y. Huang, F. Saied, and E. Seidel. Finite element multigrid solution of the initial value problem in numerical relativity. Technical Report TR040, National Center for Supercomputer Applications, February 1997.

[25] D. Bernstein and M. Holst. A 3D finite element solver for the initial-value problem. In A. Olinto, J. A Frieman, and D. N. Schramm, editors, *Proceedings of the Eighteenth Texas Symposium on Relativistic Astrophysics and Cosmology, December 16-20, 1996, Chicago, Illinois*, Singapore, 1998. World Scientific.

[26] M. Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Advances in*

Computational Mathematics, 15(1–4):139–191, 2001.

[27] M. Holst and G. Tsogtgerel. Adaptive finite element approximation of nonlinear geometric PDE. Preprint.

[28] M. Holst and G. Tsogtgerel. Convergent adaptive finite element approximation of the Einstein constraints. Preprint.

[29] M. 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.

[30] N. Baker, M. Holst, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems. *J. Comput. Chem.*, 21:1343–1352, 2000.

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

[32] R. A. Adams. *Sobolev Spaces.* Academic Press, San Diego, CA, 1978.

[33] R. M. Wald. *General Relativity.* University of Chicago Press, Chicago, IL, 1984.

[34] J. M. Lee. *Riemannian Manifolds.* Springer-Verlag, New York, NY, 1997.

[35] J. W. York, Jr. Kinematics and dynamics of general relativity. In L. L. Smarr, editor, *Sources of Gravitational Radiation*, pages 83–126. Cambridge University Press, Cambridge, MA, 1979.

[36] J.W. York, Jr. Conformally invariant orthogonal decomposition of symmetric tensors on Riemannian manifolds and the initial-value problem of general relativity. *J. Math. Phys.*, 14(4):456–464, April, 1973.

[37] J. W. York, Jr. and T. Piran. The initial value problem and beyond. In R. A. Matzner and L. C. Shepley, editors, *Spacetime and Geometry: The Alfred Schild Lectures*, pages 147–176, Austin, Texas, 1982. University of Texas Press.

[38] G. Cook and H. Pfeiffer. Excision boundary conditions for black hole initial data. *Phys. Rev. D*, 70:104016, 2004.

[39] A. Lichnerowicz. L'integration des equations de la gravitation relativiste et le probleme des n corps. *J. Math. Pures Appl.*, 23:37–63, 1944.

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

[41] N. O'Murchadha and J.W. York, Jr. Initial-value problem of general relativity. I. General formulation and physical interpretation. *Phys. Rev. D*, 10(2):428–436, July, 1974.

[42] N. O'Murchadha and J.W. York, Jr. Initial-value problem of general relativity. II. Stability of solutions of the initial-value equations. *Phys. Rev. D*, 10(2):437–446, July, 1974.

[43] Y. Choquet-Bruhat, J. Isenberg, and V. Moncrief. Solutions of constraints for Einstein equations. *C.R. Acad. Sci. Paris*, 315:349–355, 1992.

[44] Y. Choquet-Bruhat, J. Isenberg, and J. W. York, Jr. Einstein constraints on asymptotically euclidean manifolds. *GR-QC/9906095*, pages 1–29, 2002.

[45] K. Yosida. *Functional Analysis.* Springer-Verlag, Berlin, Germany, 1980.

[46] J. Wloka. *Partial Differential Equations.* Cambridge University Press, Cambridge, MA, 1992.

[47] S. Fucik and A. Kufner. *Nonlinear Differential Equations.* Elsevier Scientific Publishing Company, New York, NY, 1980.

[48] A. Kufner, O. John, and S. Fucik. *Function Spaces.* Noordhoff International Publishing, Leyden, The Netherlands, 1977.

[49] T. Aubin. *Nonlinear Analysis on Manifolds. Monge-Ampére Equations.* Springer-Verlag, New York, NY, 1982.

[50] P. G. Ciarlet. *Mathematical Elasticity.* North-Holland, New York, NY, 1988.

[51] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, New York, NY, 1978.

[52] R. A. DeVore and G. G. Lorentz. *Constructive Approximation.* Springer-Verlag, New York, NY, 1993.

[53] P. J. Davis. *Interpolation and Approximation.* Dover Publications, Inc., New York, NY, 1963.

[54] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, Berlin, Germany, 1994.

[55] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, NY, 1970.

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

[57] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, New York, NY, 1994.

[58] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 8.0.* Software, Environments and Tools, Vol. 5. SIAM, Philadelphia, PA, 1998.

[59] I. Babuška and W.C. Rheinboldt. A posteriori error estimates for the finite element method. *Internat. J. Numer. Methods Engrg.*, 12:1597–1615, 1978.

[60] I. Babuška and W.C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:736–754, 1978.

[61] R. Verfürth. A posteriori error estimates for nonlinear problems. Finite element discretizations of elliptic equations. *Math. Comp.*, 62(206):445–475, 1994.

[62] R. Verfrth. *A Review of* A Posteriori *Error Estimation and Adaptive Mesh-Refinement Techniques.* John Wiley & Sons Ltd, New York, NY, 1996.

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

[64] R. E. Bank and R. K. Smith. A posteriori error estimates based on hierarchical bases. *SIAM J. Numer. Anal.*, 30(4):921–935, 1993.

[65] R. E. Bank and A. Weiser. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comp.*, 44(170):283–301, 1985.

[66] M.C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Internat. J. Numer. Methods Engrg.*, 20:745–756, 1984.

[67] M.C. Rivara. Local modification of meshes for adaptive and/or multigrid finite-element methods. *J. Comput. Appl. Math.*, 36:79–89, 1991.

[68] I.G. Rosenberg and F. Stenger. A lower bound on the angles of triangles constructed by bisecting the longest side. *Math. Comp.*, 29:390–395, 1975.

[69] M. Stynes. On faster convergence of the bisection method for all triangles. *Math. Comp.*, 35:1195–1201, 1980.

[70] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *Impact of Computing in Science and Engineering*, 3:181–191, 1991.

[71] E. Bänsch. An adaptive finite-element strategy for the three-dimsional time-dependent Navier-Stokes equations. *J. Comput. Appl. Math.*, 36:3–28, 1991.

[72] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on bisection. *SIAM J. Sci. Statist. Comput.*, 16(6):1269–1291, 1995.

[73] J.M. Maubach. Local bisection refinement for N-simplicial grids generated by relection. *SIAM J. Sci. Statist. Comput.*, 16(1):210–277, 1995.

[74] R. Beck, B. Erdmann, and R. Roitzsch. KASKAD E 3.0: An ojbect-oriented adaptive finite element code. Technical Report TR95–4, Konrad-Zuse-Zentrum for Informationstechnik, Berlin, 1995.

[75] J. Bey. Adaptive grid manager: AGM3D manual. Technical Report 50, SFB 382, Math. Inst. Univ. Tubingen, 1996.

[76] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. Rentz-Reichert, and C. Wieners. *UG – A Flexible Software Toolbox for Solving Partial Differential Equations*, 1998. To appear in *Computing and Visualization in Science.*

[77] R. E. Bank and D. J. Rose. Analysis of a multilevel iterative method for nonlinear finite element equations. *Math. Comp.*, 39(160):453–465, 1982.

[78] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982.

[79] S. C. Eisenstat and H. F. Walker. Globally Convergent Inexact Newton Methods. Technical report, Dept. of Mathematics and Statistics, Utah State University, 1992.

[80] R. E. Bank and D. J. Rose. Parameter selection for Newton-like methods applicable to nonlinear partial differential equations. *SIAM J. Numer. Anal.*, 17(6):806–822, 1980.

[81] R. E. Bank and D. J. Rose. Global Approximate Newton Methods. *Numer. Math.*, 37:279–295, 1981.

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

[83] R. E. Bank and H. D. Mittelmann. Stepsize selection in continuation procedures and damped Newton's method. *J. Comput. Appl. Math.*, 26:67–77, 1989.

[84] W. Hackbusch. *Multi-grid Methods and Applications.* Springer-Verlag, Berlin, Germany, 1985.

[85] R. E. Bank and T. F. Dupont. An optimal order process for solving finite element equations. *Math. Comp.*, 36(153):35–51, 1981.

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

[87] R. E. Bank and J. Xu. The hierarchical basis multigrid method and incomplete LU decomposition. In *Seventh International Symposium on Domain Decomposition Methods for*

Partial Differential Equations (D. Keyes and J. Xu, eds.), pages 163–173. AMS, Providence, Rhode Island, 1994.

[88] R. E. Bank and J. Xu. An algorithm for coarsening unstructured meshes. *Numer. Math.*, 73:1–36, 1996.

[89] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.

[90] T. F. Chan, B. Smith, and J. Zou. Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids. Technical Report CAM 94-8, Department of Mathematics, UCLA, 1994.

[91] T.F. Chan, S. Go, and L. Zikatanov. Lecture notes on multilevel methods for elliptic problems on unstructured meshes. Technical report, Dept. of Mathematics, UCLA, 1997.

[92] Achi Brandt, Steve McCormick, and John Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and Its Applications (D.J. Evans, ed.)*. Cambridge Univ. Press, 1984.

[93] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[94] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report UCD/CCM 34, Center for Computational Mathematics, University of Colorado at Denver, 1994.

[95] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Technical Report UCD/CCM 36, Center for Computational Mathematics, University of Colorado at Denver, 1995.

[96] R. E. Bank and T. F. Dupont. Analysis of a two-level scheme for solving finite element equations. Technical Report CNA–159, Center for Numerical Analysis, University of Texas at Austin, 1980.

[97] R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.

[98] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[99] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55:1–22, 1990.

[100] J. H. Bramble and J. E. Pasciak. New estimates for multilevel algorithms including the V-cycle. *Math. Comp.*, 60(202):447–471, 1993.

[101] P. S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets, I: Theory. *Numer. Linear Alg. Appl.*, 4(2):103–126, 1997.

[102] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.

[103] B. Aksoylu and M. Holst. Optimality of multilevel preconditioners for local mesh refinement in three dimensions. *SIAM J. Numer. Anal.*, 44(3):1005–1025, 2006.

[104] B. Aksoylu and M. Holst. An odyssey into local refinement and multilevel preconditioning I: Optimality of the BPX preconditioner. Technical Report ICES 05-03, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2005.

[105] B. Aksoylu, S. Bond, and M. Holst. An odyssey into local refinement and multilevel preconditioning III: Implementation and numerical experiments. *SIAM J. Sci. Comput.*, 25(2):478–498, 2003.

[106] B. Aksoylu, S. Bond, and M. Holst. Implementation and theoretical aspects of the BPX preconditioner in the three-dimensional local mesh refinement setting. Technical Report ICES 04-50, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2004.

[107] B. Aksoylu and M. Holst. An odyssey into local refinement and multilevel preconditioning II: Stabilizing hierarchical basis methods. Technical Report ICES 05-04, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2005.

[108] J. Xu and J. Qin. Some remarks on a multigrid preconditioner. *SIAM J. Sci. Comput.*, 15:172–184, 1994.

[109] F. Bornemann and H. Yserentant. A basic norm equivalence for the theory of multilevel methods. *Numer. Math.*, 64:455–476, 1993.

[110] P. Oswald. *Multilevel Finite Element Approximation.* B. G. Teubner, Stuttgart, Germany, 1994.

[111] F. Bornemann, B. Erdmann, and R. Kornhuber. Adaptive multilevel methods in three space dimensions. *Internat. J. Numer. Methods Engrg.*, 36:3187–3203, 1993.

[112] P. S. Vassilevski and J. Wang. Wavelet-like methods in the design of efficient multilevel

preconditioners for elliptic pd es. In W. Dahmen, A. Kurdila, and P. Oswald, editors, *Multiscale Wavelet Methods for Partial Differential Equations*, chapter 1, pages 59–105. Academic Press, New York, NY, 1997.

[113] P. S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets, II: Implementation and numerical experiments. *SIAM J. Sci. Comput.*, 20(2):490–514, 1998.

[114] S. Jaffard. Wavelet methods for fast resolution of elliptic problems. *SIAM J. Numer. Anal.*, 29(4):965–986, 1992.

[115] R. Stevenson. A robust hierarchical basis preconditioner on general meshes. *Numer. Math.*, 78:269–303, 1997.

[116] R. P. Stevenson. Robustness of the additive multiplicative frequency decomposition multi-level method. *Computing*, 54:331–346, 1995.

[117] O. Korobkin, B. Aksoylu, M. Holst, E. Pazos, and M. Tiglio. Solving the Einstein constraints on multi-block triangulations using finite elements. Submitted to J. Comput. Phys. Available as arXiv:0801.1823 [gr-qc].

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

[119] R. E. Bank and R. K. Smith. Mesh smoothing using *a posteriori* error estimates. *SIAM J. Numer. Anal.*, 34:979–997, 1997.

[120] A. Liu and B. Joe. Relationship between tetrahedron shape measures. *BIT*, 34:268–287, 1994.

[121] M. Holst and S. Vandewalle. Schwarz methods: to symmetrize or not to symmetrize. *SIAM J. Numer. Anal.*, 34(2):699–722, 1997.

[122] B. K. Soni, J. F. Thompson, J. Hauser, and P. Eiseman. *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*. Mississippi State University, Mississippi State, April, 1996.

[123] Richard Arnowitt, Stanley Deser, and Charles W. Misner. The dynamics of general relativity. In L. Witten, editor, *Gravitation: An introduction to current research*, pages 227–265. John Wiley, New York, 1962.

[124] Dieter R. Brill. On the positive definite mass of the bondi-weber-wheeler time-symmetric gravitational waves. *Ann. Phys.*, 7:466–483, 1959.

[125] N. Ó Murchadha. Brill Waves. In B. L. Hu and T. A. Jacobson, editors, *Directions in General Relativity: Papers in Honor of Dieter Brill, Volume 2*, pages 210–+, 1993.

[126] Miguel Alcubierre, Gabrielle Allen, Bernd Bruegmann, Gerd Lanfermann, Edward Seidel, Wai-Mo Suen, and Malcolm Tobias. Gravitational collapse of gravitational waves in 3d numerical relativity. *Physical Review D*, 61:041501, 2000.