Solving PDEs Numerically on Manifolds with Arbitrary Spatial Topology

Lee Lindblom and Béla Szilágyi

Theoretical Astrophysics, Caltech

Workshop on Geometrical Numerical Methods for PDEs
Center for Computational Mathematics, UCSD
13 November 2011

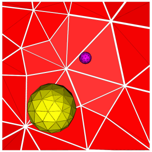
- Representations of arbitrary 3-manifolds.
- Boundary conditions for elliptic and hyperbolic PDEs.
- Mapping tensor fields across computational boundaries.
- Numerical methods.
- Numerical tests for solutions of simple PDEs.

Representations of Arbitrary 3-Manifolds

- Goal: Develop numerical methods that are easily adapted to solving elliptic PDEs on 3-manifolds Σ with arbitrary topology, and hyperbolic PDEs on manifolds with topology $R \times \Sigma$.

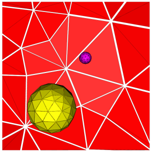
Representations of Arbitrary 3-Manifolds

- Goal: Develop numerical methods that are easily adapted to solving elliptic PDEs on 3-manifolds Σ with arbitrary topology, and hyperbolic PDEs on manifolds with topology $R \times \Sigma$.
- Every 3-manifold admits a triangulation (Moire 1952), i.e. can be represented as a set of tetrahedrons, plus a list of rules for gluing their faces together.



Representations of Arbitrary 3-Manifolds

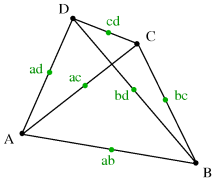
- Goal: Develop numerical methods that are easily adapted to solving elliptic PDEs on 3-manifolds Σ with arbitrary topology, and hyperbolic PDEs on manifolds with topology $R \times \Sigma$.
- Every 3-manifold admits a triangulation (Moire 1952), i.e. can be represented as a set of tetrahedrons, plus a list of rules for gluing their faces together.



- Cubes make more convenient computational domains for finite difference and spectral numerical methods.
- Can arbitrary 3-manifolds be "cubed", i.e. represented as a set of cubes plus a list of rules for gluing their faces together?

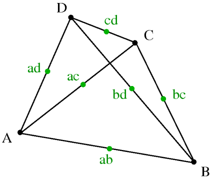
“Cubed” Representations of Arbitrary 3-Manifolds

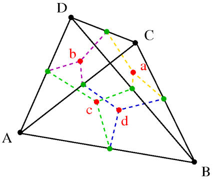
- Every triangulation can be refined to a ”cubed” representation: divide each tetrahedron into four “distorted” cubes.



“Cubed” Representations of Arbitrary 3-Manifolds

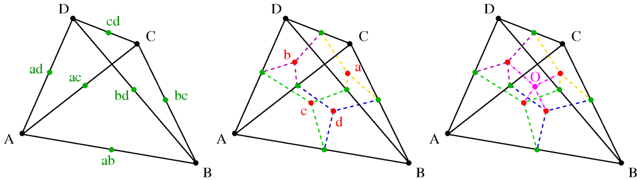
- Every triangulation can be refined to a ”cubed” representation: divide each tetrahedron into four “distorted” cubes.





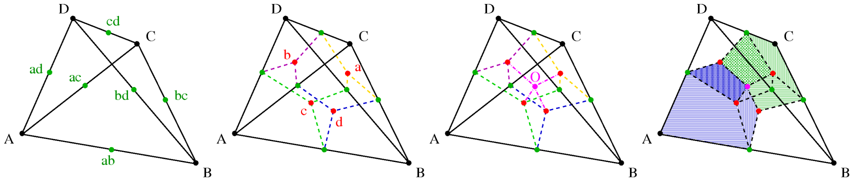
"Cubed" Representations of Arbitrary 3-Manifolds

- Every triangulation can be refined to a "cubed" representation: divide each tetrahedron into four "distorted" cubes.



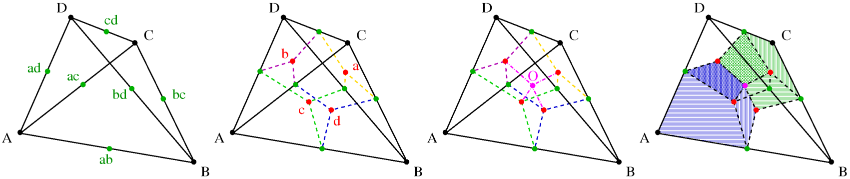
"Cubed" Representations of Arbitrary 3-Manifolds

- Every triangulation can be refined to a "cubed" representation: divide each tetrahedron into four "distorted" cubes.

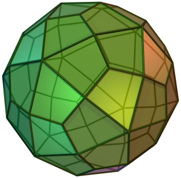


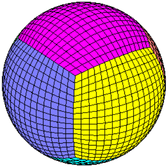
“Cubed” Representations of Arbitrary 3-Manifolds

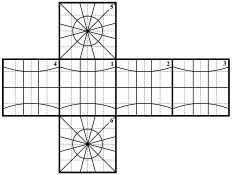
- Every triangulation can be refined to a ”cubed” representation: divide each tetrahedron into four “distorted” cubes.



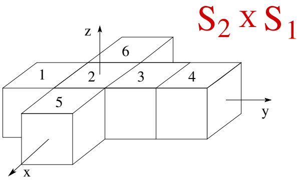
- Every 3-manifold can therefore be represented as a set of cubes, plus maps that identify their faces in the appropriate way.

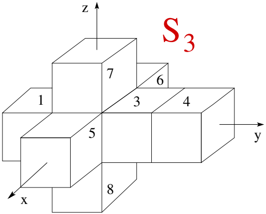






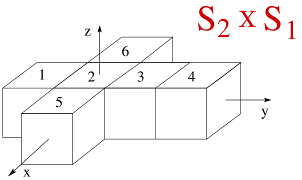
Solving PDEs on Cubed Manifolds

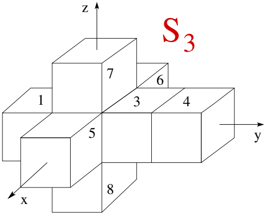




- Solve PDEs in each cubic block region separately.
- Use boundary conditions on cube faces to select the correct smooth global solution.

Solving PDEs on Cubed Manifolds

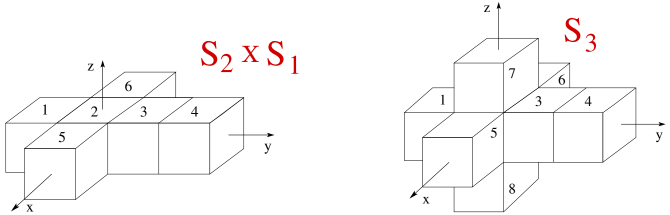




- Solve PDEs in each cubic block region separately.
- Use boundary conditions on cube faces to select the correct smooth global solution.
- For second-order strongly elliptic systems: enforce continuity on one face and continuity of normal derivatives on neighboring face,

$$u_A = u_B \qquad\qquad \nabla_{n_B} u_B = -\nabla_{n_A} u_A.$$

Solving PDEs on Cubed Manifolds



- Solve PDEs in each cubic block region separately.
- Use boundary conditions on cube faces to select the correct smooth global solution.
- For second-order strongly elliptic systems: enforce continuity on one face and continuity of normal derivatives on neighboring face,

$$u_A = u_B \qquad\qquad \nabla_{n_B} u_B = -\nabla_{n_A} u_A.$$

- For first-order symmetric hyperbolic systems: set incoming characteristic fields with outgoing characteristics from neighbor,

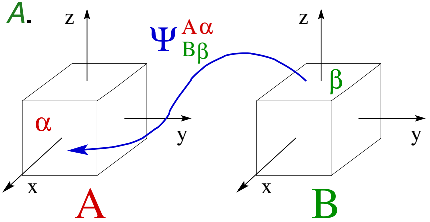
$$\tilde{u}_A^- = \tilde{u}_B^+ \qquad\qquad \tilde{u}_B^- = \tilde{u}_A^+.$$

Mapping Boundary Data: Scalars

- Choose the cubic-block coordinate patches to have uniform (coordinate) size and orientation.
- Maps $\Psi^{A\alpha}_{B\beta}$ between boundary faces are linear:

$$x^i_A = c^i_{A\,\alpha} + C^{A\,\alpha\,i}_{B\,\beta\,k}(x^k_B - c^k_{B\,\beta}),$$

where $C^{A\,\alpha\,i}_{B\,\beta\,k}$ is a rotation-reflection matrix, and $c^i_{A\,\alpha}$ is the center of the α face of block A.

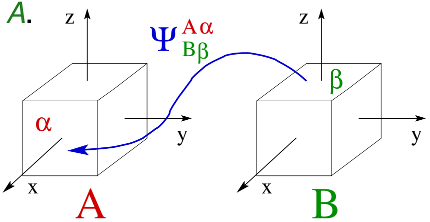


Mapping Boundary Data: Scalars

- Choose the cubic-block coordinate patches to have uniform (coordinate) size and orientation.
- Maps $\Psi^{A\alpha}_{B\beta}$ between boundary faces are linear:

$$x^i_A = c^i_{A\,\alpha} + C^{A\,\alpha\,i}_{B\,\beta\,k}(x^k_B - c^k_{B\,\beta}),$$

where $C^{A\,\alpha\,i}_{B\,\beta\,k}$ is a rotation-reflection matrix, and $c^i_{A\,\alpha}$ is the center of the α face of block A.



- This map provides the needed boundary transformation law for scalar fields: $\bar{u}_A(x^i_A) \equiv u_B(x^k_B)$, where x^i_A and x^k_B are related by the coordinate boundary map.

Mapping Boundary Data: Tensors

- Jacobian of the boundary coordinate map gives the appropriate transformation law for vectors tangent to the boundary surface:

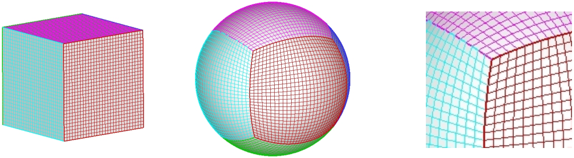
$$\bar{v}^{p}_{A}(x^{i}_{A}) \equiv C^{A\,\alpha\,p}_{B\,\beta\,q} v^{q}_{B}(x^{k}_{B}).$$

Mapping Boundary Data: Tensors

- Jacobian of the boundary coordinate map gives the appropriate transformation law for vectors tangent to the boundary surface:

$$\bar{v}^p_A(x^i_A) \equiv C^{A\,\alpha\,p}_{B\,\beta\,q} v^q_B(x^k_B).$$

- In general the normal coordinate basis vector $\partial_{A\,\sigma}$ is not the smooth extension of $\partial_{B\,\sigma}$, so a more complicated transformation law is needed for generic vectors.

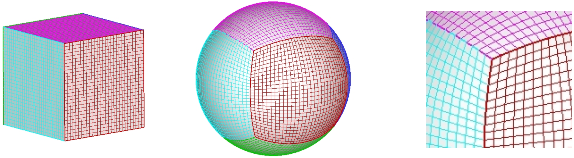


Mapping Boundary Data: Tensors

- Jacobian of the boundary coordinate map gives the appropriate transformation law for vectors tangent to the boundary surface:

$$\bar{v}^p_A(x^i_A) \equiv C^{A\,\alpha\,p}_{B\,\beta\,q} v^q_B(x^k_B).$$

- In general the normal coordinate basis vector $\partial_{A\,\sigma}$ is not the smooth extension of $\partial_{B\,\sigma}$, so a more complicated transformation law is needed for generic vectors.



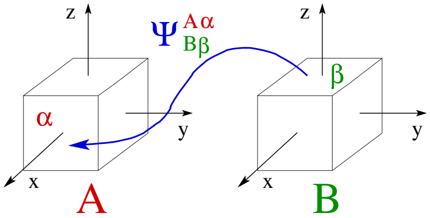
- Additional information must be specified to fix the relationship between the normal coordinate basis vectors, to ensure that smooth functions have smooth derivatives across the block boundaries.

Mapping Boundary Data: Tensors II

- One way to specify the required differentiable structure at the boundaries is to require a global smooth metric g_{ab} be provided.

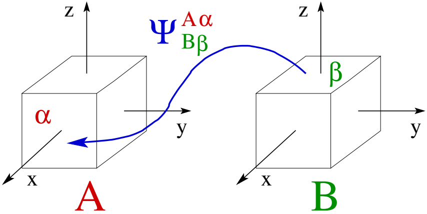
Mapping Boundary Data: Tensors II

- One way to specify the required differentiable structure at the boundaries is to require a global smooth metric g_{ab} be provided.
- Use the metric g_{ab} to construct the outward directed unit normals n_{Aa} and n_{Ba} on each boundary face.



Mapping Boundary Data: Tensors II

- One way to specify the required differentiable structure at the boundaries is to require a global smooth metric g_{ab} be provided.
- Use the metric g_{ab} to construct the outward directed unit normals n_{Aa} and n_{Ba} on each boundary face.



- These outward directed geometrical normals, $n^a_A = g^{ab}_A n_{Ab}$ and $n^a_B = g^{ab}_B n_{Bb}$, can be used to define the natural transformation law for smooth vectors across the boundaries:

$$\bar{v}^a_A(x^i_A) \equiv J^{A\alpha\,a}_{B\beta\,b} v^b_B(x^k_B),$$

with $J^{A\alpha\,a}_{B\beta\,b} = C^{A\alpha\,a}_{B\beta\,c}(\delta^c_b - n^c_B n_{Bb}) - n^a_A n_{Bb}$.

Numerical Methods

- Represent each component of each tensor function as a (finite) sum of spectral basis functions, $\mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, T_i(x)\, T_j(y)\, T_k(z)$, in each cubic block region.

Numerical Methods

- Represent each component of each tensor function as a (finite) sum of spectral basis functions, $\mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, T_i(x)\, T_j(y)\, T_k(z)$, in each cubic block region.
- Evaluate derivatives of the functions using the known derivatives of the basis functions: $\partial_x \mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, \partial_x T_i(x)\, T_j(y)\, T_k(z)$.

Numerical Methods

- Represent each component of each tensor function as a (finite) sum of spectral basis functions, $\mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, T_i(x)\, T_j(y)\, T_k(z)$, in each cubic block region.
- Evaluate derivatives of the functions using the known derivatives of the basis functions: $\partial_x \mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, \partial_x T_i(x)\, T_j(y)\, T_k(z)$.
- Evaluate the PDEs and BCs on a set of collocation points, $\{x_i, y_j, z_k\}$, chosen so that $\mathbf{u}(x_i, y_j, z_k)$ can be mapped efficiently onto the spectral coefficients $\mathbf{u}_{ijk}$. Derivatives become linear combinations of the fields: $\partial_x \mathbf{u}(x_i, y_j, z_k) = \sum_\ell D_i{}^\ell\, \mathbf{u}(x_\ell, y_j, z_k)$.

Numerical Methods

- Represent each component of each tensor function as a (finite) sum of spectral basis functions, $\mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, T_i(x)\, T_j(y)\, T_k(z)$, in each cubic block region.
- Evaluate derivatives of the functions using the known derivatives of the basis functions: $\partial_x \mathbf{u} = \sum_{ijk} \mathbf{u}_{ijk}\, \partial_x T_i(x)\, T_j(y)\, T_k(z)$.
- Evaluate the PDEs and BCs on a set of collocation points, $\{x_i, y_j, z_k\}$, chosen so that $\mathbf{u}(x_i, y_j, z_k)$ can be mapped efficiently onto the spectral coefficients $\mathbf{u}_{ijk}$. Derivatives become linear combinations of the fields: $\partial_x \mathbf{u}(x_i, y_j, z_k) = \sum_\ell D_i{}^\ell\, \mathbf{u}(x_\ell, y_j, z_k)$.
- For elliptic systems, these pseudo-spectral equations become a system of algebraic equations for $\mathbf{u}(x_i, y_j, z_k)$. Solve these algebraic equations using standard numerical methods.
- For hyperbolic systems these equations become a system of ordinary differential equations for $\mathbf{u}(x_i, y_j, z_k, t)$. Solve these equations by the method of lines using standard ode integrators.

Testing the Elliptic PDE Solver

- Solve the elliptic PDE, $\nabla^i\nabla_i\psi - c^2\psi = f$ where c^2 is a constant, and f is a given function.

Testing the Elliptic PDE Solver

- Solve the elliptic PDE, $\nabla^i\nabla_i\psi - c^2\psi = f$ where c^2 is a constant, and f is a given function.
- Use the co-variant derivative ∇_i for the round metric on $S^2 \times S^1$:

$$
\begin{aligned}
ds^2 &= R_1^2 d\chi^2 + R_2^2\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right),\\
&= \left(\frac{2\pi R_1}{L}\right)^2 dz^2 + \left(\frac{\pi R_2}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)}{(1+X_A^2+Y_A^2)^2}\\
&\quad\times\left[(1+X_A^2)\,dx^2 - 2X_AY_A\,dx\,dy + (1+Y_A^2)\,dy^2\right].
\end{aligned}
$$

where $X_A = \tan\left[\pi(x-c_A^x)/2L\right]$ and $Y_A = \tan\left[\pi(y-c_A^y)/2L\right]$ are “local” Cartesian coordinates in each cubic-block.

Testing the Elliptic PDE Solver

- Solve the elliptic PDE, $\nabla^i\nabla_i\psi - c^2\psi = f$ where c^2 is a constant, and f is a given function.
- Use the co-variant derivative ∇_i for the round metric on $S^2\times S^1$:

$$
\begin{aligned}
ds^2 &= R_1^2 d\chi^2 + R_2^2\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right),\\
&= \left(\frac{2\pi R_1}{L}\right)^2 dz^2 + \left(\frac{\pi R_2}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)}{(1+X_A^2+Y_A^2)^2}\\
&\quad\times\left[(1+X_A^2)\,dx^2 - 2X_A Y_A\,dx\,dy + (1+Y_A^2)\,dy^2\right].
\end{aligned}
$$

where $X_A = \tan\left[\pi(x-c_A^x)/2L\right]$ and $Y_A = \tan\left[\pi(y-c_A^y)/2L\right]$ are “local” Cartesian coordinates in each cubic-block.

- Let $f = -(\omega^2 + c^2)\psi_A$, where $\psi_A = \Re\left[e^{ik\chi}Y_{\ell m}(\theta,\varphi)\right]$. The angles χ, θ and φ are functions of the coordinates x, y and z.

Testing the Elliptic PDE Solver

- Solve the elliptic PDE, $\nabla^i\nabla_i\psi - c^2\psi = f$ where c^2 is a constant, and f is a given function.
- Use the co-variant derivative ∇_i for the round metric on $S^2\times S^1$:

$$
\begin{aligned}
ds^2 &= R_1^2 d\chi^2 + R_2^2\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right),\\
&= \left(\frac{2\pi R_1}{L}\right)^2 dz^2 + \left(\frac{\pi R_2}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)}{(1+X_A^2+Y_A^2)^2}\\
&\quad\times\left[(1+X_A^2)\,dx^2 - 2X_AY_A\,dx\,dy + (1+Y_A^2)\,dy^2\right].
\end{aligned}
$$

where $X_A = \tan\left[\pi(x-c_A^x)/2L\right]$ and $Y_A = \tan\left[\pi(y-c_A^y)/2L\right]$ are "local" Cartesian coordinates in each cubic-block.

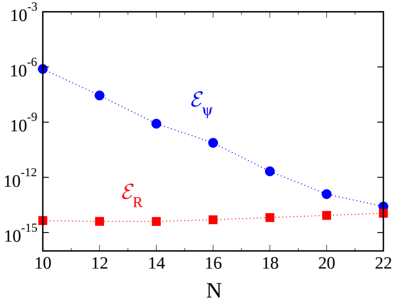
- Let $f = -(\omega^2 + c^2)\psi_A$, where $\psi_A = \Re\left[e^{ik\chi}Y_{\ell m}(\theta,\varphi)\right]$. The angles χ, θ and φ are functions of the coordinates x, y and z.
- The unique, exact, analytical solution to this problem is $\psi = \psi_A$, when $\omega^2 = \ell(\ell+1)/R_2^2 + k^2/R_1^2$.

Testing the Elliptic PDE Solver II

- Measure the accuracy of the numerical solution ψ_N as a function of numerical resolution N (grid points per dimension) in two ways:
  - First, with the residual $R_N \equiv \nabla^i\nabla_i\psi_N - c^2\psi_N - f$, and its norm:
$$\mathcal{E}_R = \sqrt{\frac{\int R_N^2\sqrt{g}d^3x}{\int f^2\sqrt{g}d^3x}}.$$
  - Second, with the solution error, $\Delta\psi = \psi_N - \psi_A$, and its norm:
$$\mathcal{E}_\psi = \sqrt{\frac{\int \Delta\psi^2\sqrt{g}d^3x}{\int \psi_A^2\sqrt{g}d^3x}}.$$

Testing the Elliptic PDE Solver II

- Measure the accuracy of the numerical solution ψ_N as a function of numerical resolution N (grid points per dimension) in two ways:
  - First, with the residual $R_N \equiv \nabla^i\nabla_i\psi_N - c^2\psi_N - f$, and its norm:
$$\mathcal{E}_R = \sqrt{\frac{\int R_N^2\sqrt{g}d^3x}{\int f^2\sqrt{g}d^3x}}.$$
  - Second, with the solution error, $\Delta\psi = \psi_N - \psi_A$, and its norm:
$$\mathcal{E}_\psi = \sqrt{\frac{\int \Delta\psi^2\sqrt{g}d^3x}{\int \psi_A^2\sqrt{g}d^3x}}.$$



- All these numerical tests were performed by implementing the ideas described here into the Spectral Einstein Code (SpEC) developed originally by the Caltech/Cornell numerical relativity collaboration.

Testing the Hyperbolic PDE Solver

- Solve the equation $\partial_t^2\psi = \nabla_i\nabla^i\psi$ with given initial data.
- Convert the second-order equation into an equivalent first-order system: $\partial_t\psi = -\Pi$, $\partial_t\Pi = -\nabla^i\Phi_i$ and $\partial_t\Phi_i = -\nabla_i\Pi$ with constraint $\mathcal{C}_i = \nabla_i\psi - \Phi_i$.

Testing the Hyperbolic PDE Solver

- Solve the equation $\partial_t^2\psi = \nabla_i\nabla^i\psi$ with given initial data.
- Convert the second-order equation into an equivalent first-order system: $\partial_t\psi = -\Pi$, $\partial_t\Pi = -\nabla^i\Phi_i$ and $\partial_t\Phi_i = -\nabla_i\Pi$ with constraint $\mathcal{C}_i = \nabla_i\psi - \Phi_i$.
- Use the co-variant derivative ∇_i for the round metric on S^3:

$$
\begin{aligned}
ds^2 &= R_3^2\left[d\chi^2 + \sin^2\chi\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right)\right], \\
&= \left(\frac{\pi R_3}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)(1+Z_A^2)}{(1+X_A^2+Y_A^2+Z_A^2)^2}\left[\frac{(1+X_A^2)(1+Y_A^2+Z_A^2)}{(1+Y_A^2)(1+Z_A^2)}dx^2 + \frac{(1+Y_A^2)(1+X_A^2+Z_A^2)}{(1+X_A^2)(1+Z_A^2)}dy^2\right. \\
&\qquad \left. + \frac{(1+Z_A^2)(1+X_A^2+Y_A^2)}{(1+X_A^2)(1+Y_A^2)}dz^2 - \frac{2X_AY_A}{1+Z_A^2}dx\,dy - \frac{2X_AZ_A}{1+Y_A^2}dx\,dz - \frac{2Y_AZ_A}{1+X_A^2}dy\,dz\right].
\end{aligned}
$$

Testing the Hyperbolic PDE Solver

- Solve the equation $\partial_t^2\psi = \nabla_i\nabla^i\psi$ with given initial data.
- Convert the second-order equation into an equivalent first-order system: $\partial_t\psi = -\Pi$, $\partial_t\Pi = -\nabla^i\Phi_i$ and $\partial_t\Phi_i = -\nabla_i\Pi$ with constraint $\mathcal{C}_i = \nabla_i\psi - \Phi_i$.
- Use the co-variant derivative ∇_i for the round metric on S^3:

$$
\begin{aligned}
ds^2 &= R_3^2\left[d\chi^2 + \sin^2\chi\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right)\right],\\
&= \left(\frac{\pi R_3}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)(1+Z_A^2)}{(1+X_A^2+Y_A^2+Z_A^2)^2}\left[\frac{(1+X_A^2)(1+Y_A^2+Z_A^2)}{(1+Y_A^2)(1+Z_A^2)}dx^2 + \frac{(1+Y_A^2)(1+X_A^2+Z_A^2)}{(1+X_A^2)(1+Z_A^2)}dy^2\right.\\
&\qquad\left. + \frac{(1+Z_A^2)(1+X_A^2+Y_A^2)}{(1+X_A^2)(1+Y_A^2)}dz^2 - \frac{2X_AY_A}{1+Z_A^2}dx\,dy - \frac{2X_AZ_A}{1+Y_A^2}dx\,dz - \frac{2Y_AZ_A}{1+X_A^2}dy\,dz\right].
\end{aligned}
$$

- Choose initial data with $\psi_{t=0} = \Re[Y_{k\ell m}(\chi,\theta,\varphi)]$, $\Pi_{t=0} = -\Re[i\omega Y_{k\ell m}(\chi,\theta,\varphi)]$ and $\Phi_{i\,t=0} = \Re[\nabla_i Y_{k\ell m}(\chi,\theta,\varphi)]$ where $\omega^2 = k(k+2)/R_3^2$.

Testing the Hyperbolic PDE Solver

- Solve the equation $\partial_t^2\psi = \nabla_i\nabla^i\psi$ with given initial data.
- Convert the second-order equation into an equivalent first-order system: $\partial_t\psi = -\Pi$, $\partial_t\Pi = -\nabla^i\Phi_i$ and $\partial_t\Phi_i = -\nabla_i\Pi$ with constraint $\mathcal{C}_i = \nabla_i\psi - \Phi_i$.
- Use the co-variant derivative ∇_i for the round metric on S^3:

$$
\begin{aligned}
ds^2 &= R_3^2\left[d\chi^2 + \sin^2\chi\left(d\theta^2 + \sin^2\theta\, d\varphi^2\right)\right], \\
&= \left(\frac{\pi R_3}{2L}\right)^2 \frac{(1+X_A^2)(1+Y_A^2)(1+Z_A^2)}{(1+X_A^2+Y_A^2+Z_A^2)^2}\left[\frac{(1+X_A^2)(1+Y_A^2+Z_A^2)}{(1+Y_A^2)(1+Z_A^2)}dx^2 + \frac{(1+Y_A^2)(1+X_A^2+Z_A^2)}{(1+X_A^2)(1+Z_A^2)}dy^2\right. \\
&\qquad \left. + \frac{(1+Z_A^2)(1+X_A^2+Y_A^2)}{(1+X_A^2)(1+Y_A^2)}dz^2 - \frac{2X_AY_A}{1+Z_A^2}dx\,dy - \frac{2X_AZ_A}{1+Y_A^2}dx\,dz - \frac{2Y_AZ_A}{1+X_A^2}dy\,dz\right].
\end{aligned}
$$

- Choose initial data with $\psi_{t=0} = \Re[Y_{k\ell m}(\chi,\theta,\varphi)]$, $\Pi_{t=0} = -\Re[i\omega Y_{k\ell m}(\chi,\theta,\varphi)]$ and $\Phi_{i\,t=0} = \Re[\nabla_i Y_{k\ell m}(\chi,\theta,\varphi)]$ where $\omega^2 = k(k+2)/R_3^2$.
- The unique, exact, analytical solution to this problem is $\psi = \psi_A = \Re[e^{i\omega t}Y_{k\ell m}(\chi,\theta,\varphi)]$, $\Pi = -\partial_t\psi_A$, and $\Phi_i = \nabla_i\psi_A$.

Testing the Hyperbolic PDE Solver II

- Measure the accuracy of the numerical solution ψ_N as a function of numerical resolution N (grid points per dimension) in two ways:
  - First, with the solution error, $\Delta\psi = \psi_N - \psi_A$, and its norm:

$$\mathcal{E}_\psi = \sqrt{\frac{\int \Delta\psi^2 \sqrt{g} d^3x}{\int \psi^2 \sqrt{g} d^3x}}.$$

  - Second, with the constraint error, $\mathcal{C}_i = \Phi_i - \nabla_i\psi$, and its norm:

$$\mathcal{E}_\mathcal{C} = \sqrt{\frac{\int g^{ij}\mathcal{C}_i\mathcal{C}_j\sqrt{g} d^3x}{\int g^{ij}(\Phi_i\Phi_j + \nabla_i\psi\nabla_j\psi)\sqrt{g} d^3x}}.$$

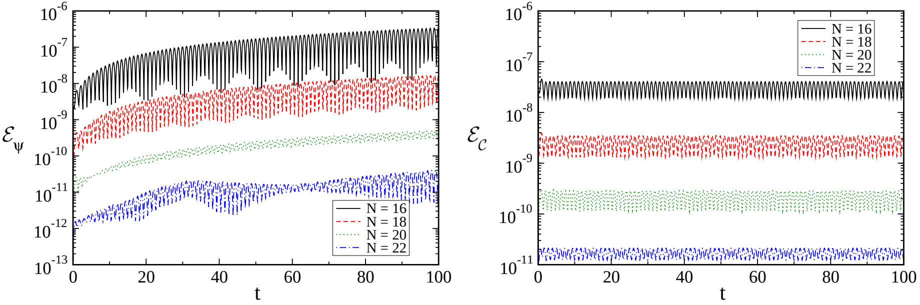
Testing the Hyperbolic PDE Solver II

- Measure the accuracy of the numerical solution ψ_N as a function of numerical resolution N (grid points per dimension) in two ways:
  - First, with the solution error, $\Delta\psi = \psi_N - \psi_A$, and its norm:

$$\mathcal{E}_\psi = \sqrt{\frac{\int \Delta\psi^2 \sqrt{g} d^3x}{\int \psi^2 \sqrt{g} d^3x}}.$$

  - Second, with the constraint error, $\mathcal{C}_i = \Phi_i - \nabla_i\psi$, and its norm:

$$\mathcal{E}_\mathcal{C} = \sqrt{\frac{\int g^{ij}\mathcal{C}_i\mathcal{C}_j \sqrt{g} d^3x}{\int g^{ij}(\Phi_i\Phi_j + \nabla_i\psi\nabla_j\psi)\sqrt{g} d^3x}}.$$



Summary

- We have developed a simple and flexible multi-block numerical method for solving partial differential equations on manifolds with arbitrary spatitial topology.

Summary

- We have developed a simple and flexible multi-block numerical method for solving partial differential equations on manifolds with arbitrary spatitial topology.
- Each new spatial topology requires:
  - A cubic-block representation of the topology, i.e. a list of cubic-block regions and a list of boundary identification maps.
  - A smooth reference metric g_{ab} to define the global differential structure on this cubic-block representation of the manifold.

Summary

- We have developed a simple and flexible multi-block numerical method for solving partial differential equations on manifolds with arbitrary spatitial topology.
- Each new spatial topology requires:
  - A cubic-block representation of the topology, i.e. a list of cubic-block regions and a list of boundary identification maps.
  - A smooth reference metric g_{ab} to define the global differential structure on this cubic-block representation of the manifold.
- These methods have been tested by solving simple elliptic and hyperbolic equations on several compact manifolds.
- These methods have also been tested by finding simple solutions to Einstein's equation on several compact manifolds.