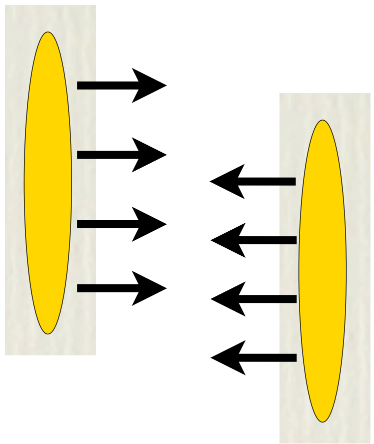
Initial data for characteristic evolution

Laurence Yaffe
University of Washington

Numerical Holography workshop, CERN, December 10, 2014

Motivation

- holographic collisions as models of heavy ion collisions
  - scattering ➡ Poincaré patch AdS
  - previous work:
    - homogeneous isotropization: 1+1D PDEs
      no transverse or longitudinal dynamics
    - planar shocks: 2+1D PDEs
      no transverse dynamics
    - boost invariant: 2+1D PDEs w. transverse radial dynamics
      unrealistic longitudinal dynamics
  - in progress (w. Paul Chesler):
    - finite "nuclei": 4+1D PDEs
      transverse and longitudinal dynamics



Initial projectiles

- Exact, analytic solution for stable, null "projectile" with arbitrary boundary energy density:

$$ds^2 = \frac{L^2}{s^2}\left[-d\tilde{t}^2 + d\tilde{x}^2 + d\tilde{y}^2 + d\tilde{z}^2 + ds^2 + h(\tilde{x},\tilde{y},\tilde{z}-\tilde{t},s)\,(d\tilde{z}-d\tilde{t})^2\right]$$

Fefferman-Graham (FG) coordinates $\tilde{X} \equiv (\tilde{t},\tilde{x},\tilde{y},\tilde{z},s)$

- metric deformation function *h:*

$$\left(\partial_s^2 - \tfrac{3}{s}\,\partial_s + \tilde{\nabla}_\perp^2\right) h = 0$$

arbitrary

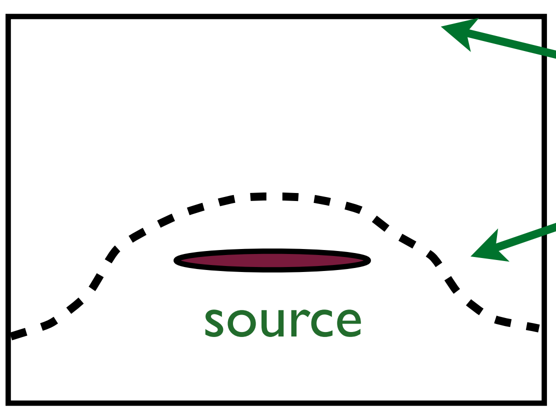
$$h(\tilde{x},\tilde{y},\tilde{z}-\tilde{t},s) = \int \frac{d^2k_\perp}{(2\pi)^2}\, e^{ik_\perp\cdot\tilde{x}_\perp}\,\tilde{h}_\perp(k_\perp)\,h_\parallel(\tilde{z}-\tilde{t})\,8s^2k_\perp^{-2}\,I_2(k_\perp s)$$

Projectile solution

- sourced or sourceless?
  - sourced: non-analytic at source, well behaved below source

    c.f. Gubser, Pufu, Yarom, arXiv:0805.1551

    - Ex: Gaussian smeared point source



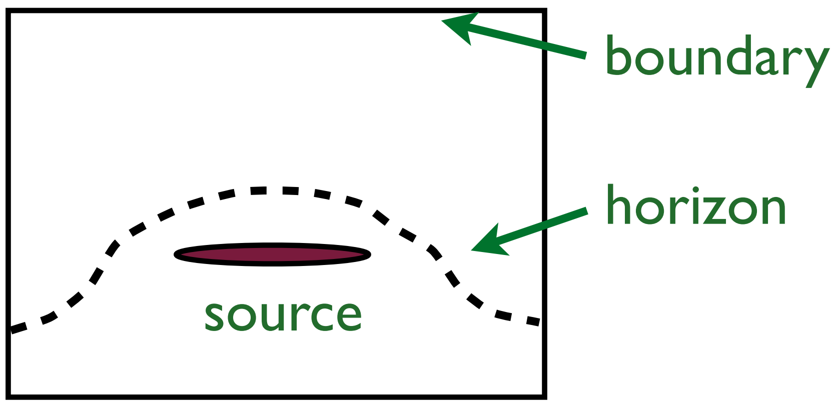
  - sourceless: analytic, exponential growth with depth
    - Ex: Gaussian (longitudinal & transverse) energy density

Projectile solution

- sourced or sourceless?
  - sourced: non-analytic at source, well behaved below source

    c.f. Gubser, Pufu, Yarom, arXiv:0805.1551

    - Ex: Gaussian smeared point source



  - sourceless: analytic, exponential growth with depth
    - ☞ Ex: Gaussian (longitudinal & transverse) energy density

Infalling coordinates

$$ds^2 = 2dt\left[dr - A(X)\,dt - F_i(X)\,dx^i\right] + \Sigma(X)^2\,\hat{g}_{ij}(X)\,dx^i dx^j$$

Eddington-Finkelstein (EF) coordinates

$$X \equiv (t, x, y, z, r) = (x^\mu_{\rm bdry}, r)$$

required initial data = rescaled (unit determinant) spatial metric

- Must transform FG → EF: $g_{MN}(X) = \dfrac{\partial \tilde{X}^A}{\partial X^M}\,\dfrac{\partial \tilde{X}^B}{\partial X^N}\,\tilde{g}_{AB}(\tilde{X}(X))$
- Strategy:
  - Demand $g_{rt}(X) = 1\,, \quad g_{r\mu}(X) = 0$

    ugly coupled equations
  - Compute infalling geodesic congruence:

$$\frac{d^2\tilde{X}^M}{dr^2} + \Gamma^M_{PQ}(\tilde{X})\,\frac{d\tilde{X}^P}{dr}\,\frac{d\tilde{X}^Q}{dr} = 0$$

"pretty" coupled equations

$$\lim_{r\to\infty} \tilde{X}^\mu(r) = x_{\rm bdry}\,, \quad \lim_{r\to\infty} d\tilde{X}^\mu/dr = 0$$

radial infall from boundary

Infalling coordinates

$$ds^2 = 2dt\left[dr - A(X)\,dt - F_i(X)\,dx^i\right] + \Sigma(X)^2\,\hat{g}_{ij}(X)\,dx^i dx^j$$

Eddington-Finkelstein (EF) coordinates

$$X \equiv (t,x,y,z,r) = (x^\mu_{\rm bdry}, r)$$

required initial data = rescaled (unit determinant) spatial metric

- Must transform FG → EF: $g_{MN}(X) = \dfrac{\partial \tilde{X}^A}{\partial X^M}\,\dfrac{\partial \tilde{X}^B}{\partial X^N}\,\tilde{g}_{AB}(\tilde{X}(X))$
- Strategy:
  - Demand $g_{rt}(X) = 1\,, \quad g_{r\mu}(X) = 0$

    ugly coupled equations
  - ☞ Compute infalling geodesic congruence:

$$\frac{d^2\tilde{X}^M}{dr^2} + \Gamma^M_{PQ}(\tilde{X})\,\frac{d\tilde{X}^P}{dr}\,\frac{d\tilde{X}^Q}{dr} = 0$$

"pretty" coupled equations

$$\lim_{r\to\infty} \tilde{X}^\mu(r) = x_{\rm bdry}\,, \quad \lim_{r\to\infty} d\tilde{X}^\mu/dr = 0$$

radial infall from boundary

Choices (1)

- Strategy
  - adaptive ODE solver
  - fixed radial grid, spectral methods, Newton iteration
- Language
  - C, C++
  - Matlab
  - Mathematica

Choices (1)

- Strategy
  - adaptive ODE solver
  - fixed radial grid, spectral methods, Newton iteration
- Language
  - C, C++
  - Matlab
  - ☞ Mathematica

    familiarity, high level, quick prototyping,
    NDSolve, Interpolation, SparseArray, Parallelize, ...

Choices (1)

- Strategy
  - ☞ adaptive ODE solver
  - fixed radial grid, spectral methods, Newton iteration
- Language
  - C, C++
  - Matlab
  - ☞ Mathematica

    familiarity, high level, quick prototyping,

    NDSolve, Interpolation, SparseArray, Parallelize, ...

Choices (1)

- Strategy
  - adaptive ODE solver
  - fixed radial grid, spectral methods, Newton iteration
- Language
  - C, C++
  - Matlab
  - ☞ Mathematica

    familiarity, high level, quick prototyping,

    NDSolve, Interpolation, SparseArray, Parallelize, ...

Choices (1)

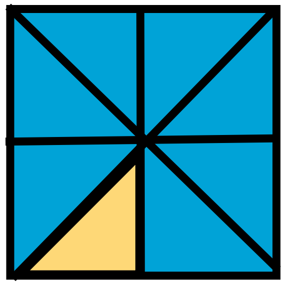
- Strategy
  - adaptive ODE solver
  - ☞ fixed radial grid, spectral methods, Newton iteration
- Language
  - C, C++
  - Matlab
  - ☞ Mathematica

    familiarity, high level, quick prototyping,

    NDSolve, Interpolation, SparseArray, Parallelize, ...

Choices (II)

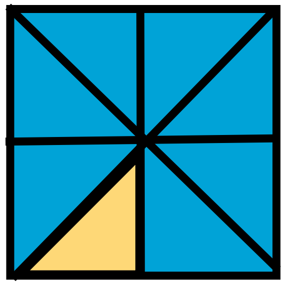
- Spatial boundary coordinates
  - cylindrical
    - dimensional reduction, take advantage of axial symmetry
    - coordinate singularity at $\rho=0$
  - Cartesian w. periodic spatial box
    - use Fourier spatial grids
    - cubic symmetry in transverse plane:



- Radial grid
  - multiple Chebyshev domains
    - sparse differentiation matrices, limits error propagation

Choices (II)

- Spatial boundary coordinates
    - cylindrical
        - dimensional reduction, take advantage of axial symmetry
        - coordinate singularity at $\rho=0$
    - ☞ Cartesian w. periodic spatial box
        - use Fourier spatial grids
        - cubic symmetry in transverse plane:



- Radial grid
    - ☞ multiple Chebyshev domains
        - sparse differentiation matrices, limits error propagation

Issues (I)

- Scale:
  - Ex: 64 x 64 x 256 x 96 (x,y,z,u) grid
    - 1 M boundary points, ÷ 8 ➡ 144K distinct geodesics
      - 1 sec/geo ➡ 40 hours
      - 10 sec/geo ➡ 17 days
      - 100 sec/geo ➡ 5.5 months
    - multi-Gb data files
- Accuracy
  - iterative solution error
  - interpolation error
  - arithmetic error

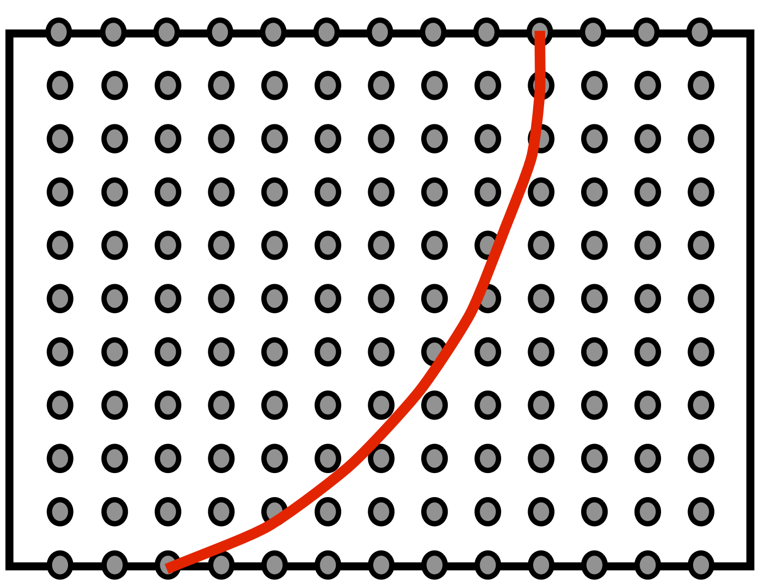
Issues (I)

- Scale:
    - Ex: 64 x 64 x 256 x 96 (x,y,z,u) grid
        - 1 M boundary points, ÷ 8 ➡ 144K distinct geodesics
            - 1 sec/geo ➡ 40 hours
            - 10 sec/geo ➡ 17 days
            - 100 sec/geo ➡ 5.5 months
        - multi-Gb data files
- Accuracy
    - iterative solution error
    - interpolation error
    - arithmetic error

☞ MMA, spectral/Newton, arbitrary precision (40 digits)

Issues (II)

- radial shift residual reparameterization freedom
  - near boundary $s \sim [r + \lambda(x, y, z)]^{-1}$
    arbitrary
  - use $u \equiv 1/r$, require $u_{\max} = s_{\max}$
  - turns initial value prob into boundary value prob
- off-grid interpolation required:



Issues (III)

- parallelization (multicore desktop computer)
  - initial efforts: horrible memory contention
  - eventual success: > 95% cpu utilization (8 cores)
- checkpointing
  - network glitches w. remote backend
  - random MMA freezes/crashes

  ➡ must be restartable
- MMA ➡ Matlab data transport
  - 2Gb size limits in some file formats

Lessons

- feasibility of 5D asymptotically AdS GR on a desktop?
  - so far, so good
- choice of Mathematica for initial data calculation?
  - initial hopes (e.g., NDSolve, Interpolate) too naive
  - development time likely no less than in lower-level language
  - must guess/infer what MMA does behind the scenes
  - but very convenient high-level constructs (Map, MapThread,...)
  - using extended precision definitely helpful