An update on Cauchy-Characteristic Matching in General Relativity

Béla Szilágyi

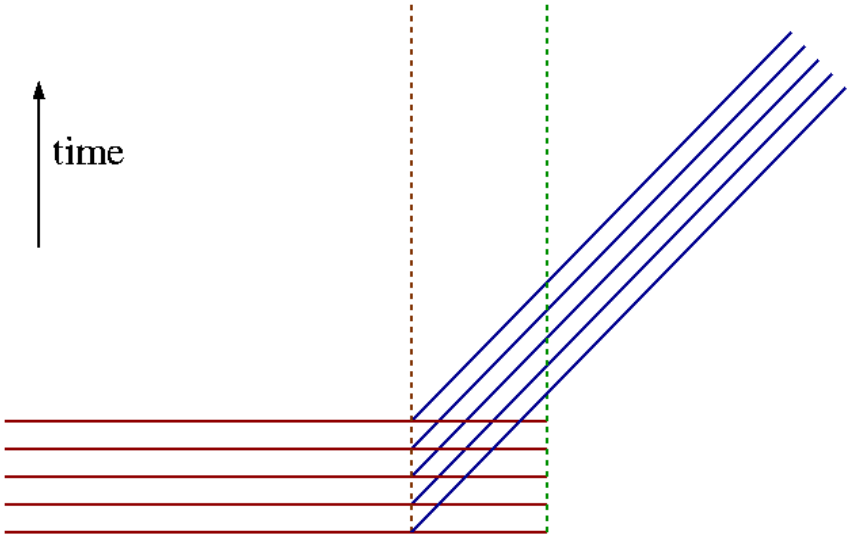
Jeffrey Winicour
Yosef Zlochower

Department of Physics and Astronomy
University of Pittsburgh, Pittsburgh, PA 15260, USA

Cauchy Characteristic Matching (CCM)

What is it?

In this approach one covers space-time by a Cauchy and a Characteristic coordinate-patch.



Well-posedness of CCM

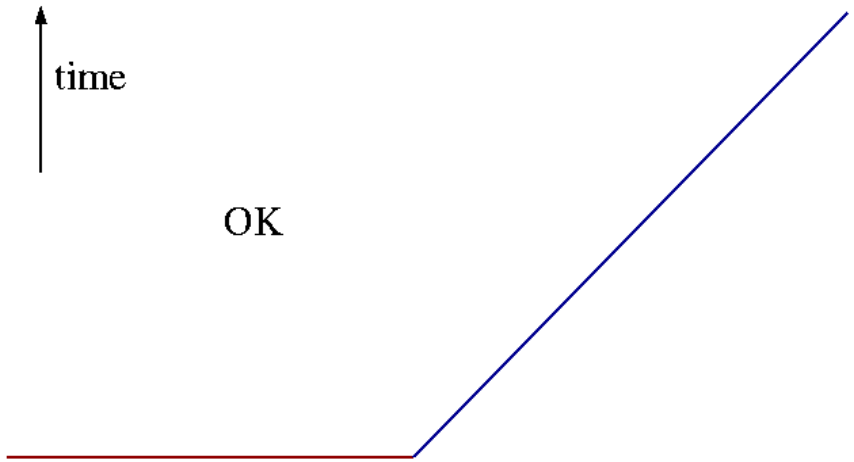
Does it make sense?

Hopefully.

Well-posedness of CCM

Does it make sense?

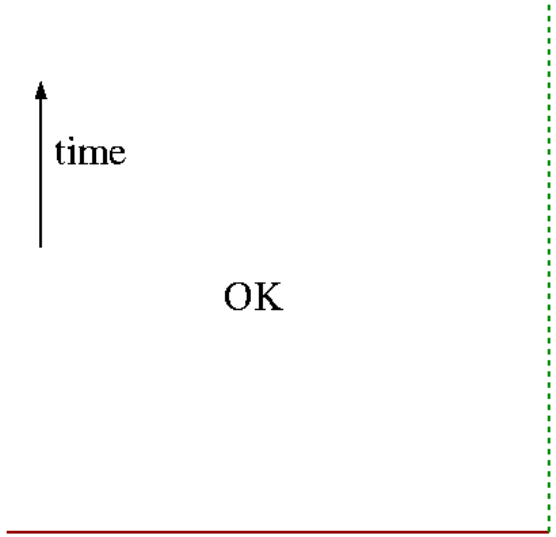
Hopefully.



Well-posedness of CCM

Does it make sense?

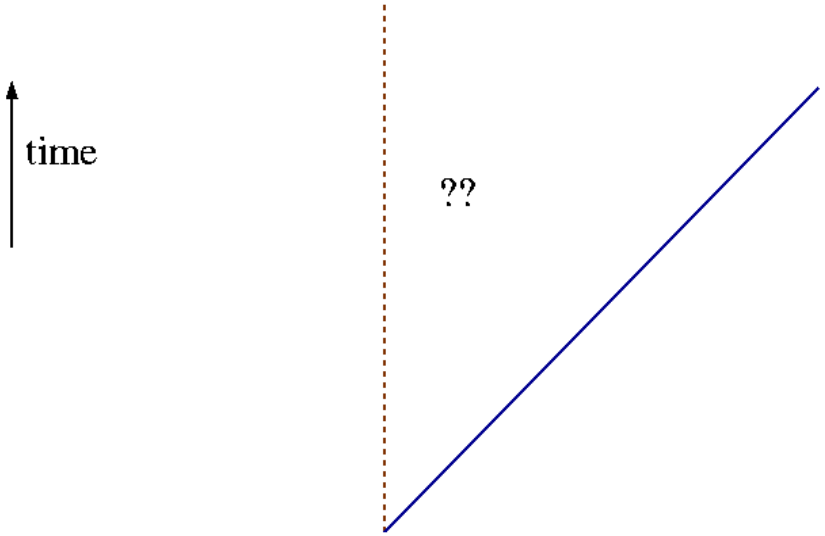
Hopefully.



Well-posedness of CCM

Does it make sense?

Hopefully.

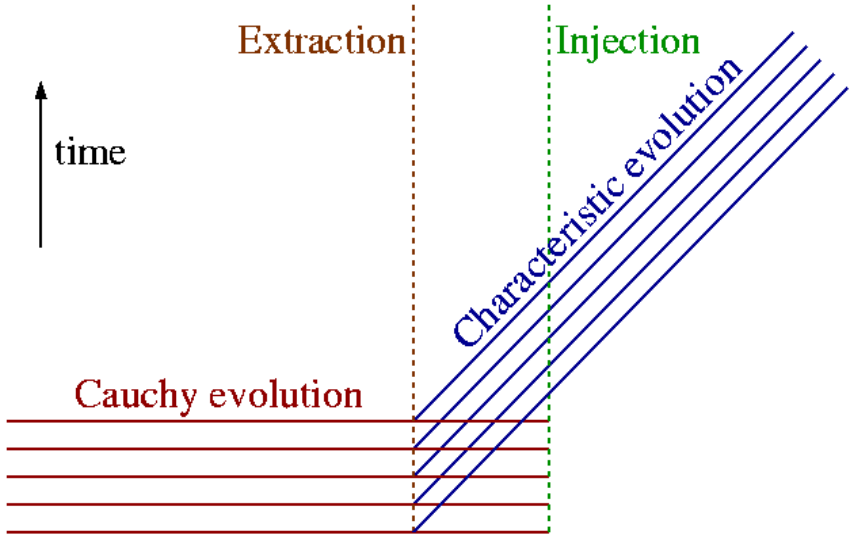


Well-posedness of CCM

Does it make sense?

Hopefully.

One of the issues is the definition of the matching boundaries.

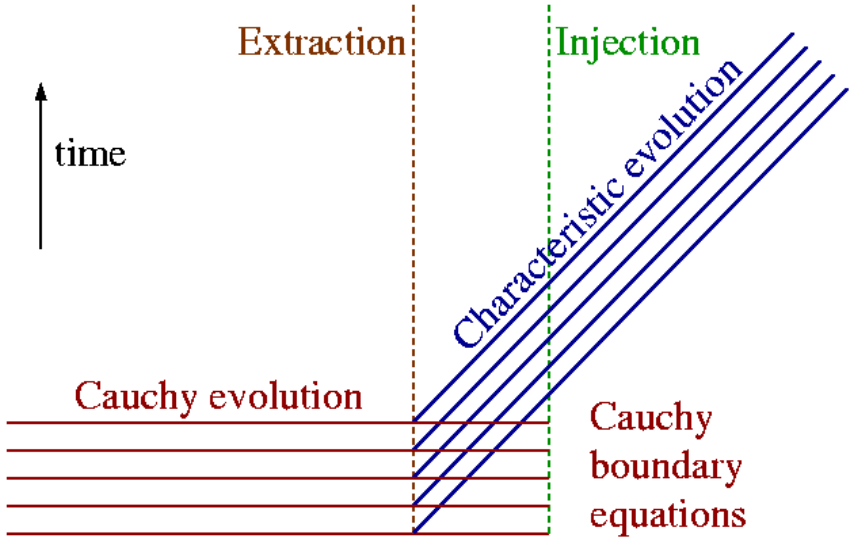


Well-posedness of CCM

Does it make sense?

Hopefully.

Another issue is that of the constraints of the two formulations.

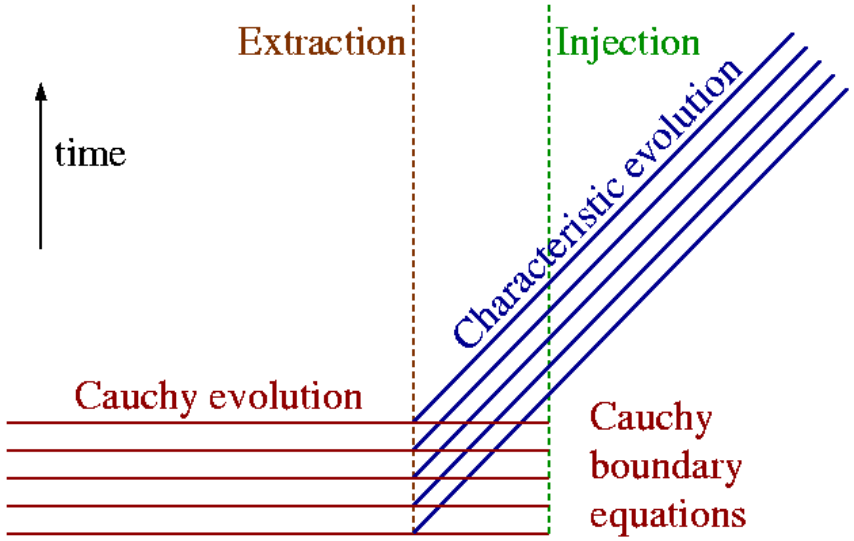


Well-posedness of CCM

Does it make sense?

Hopefully.

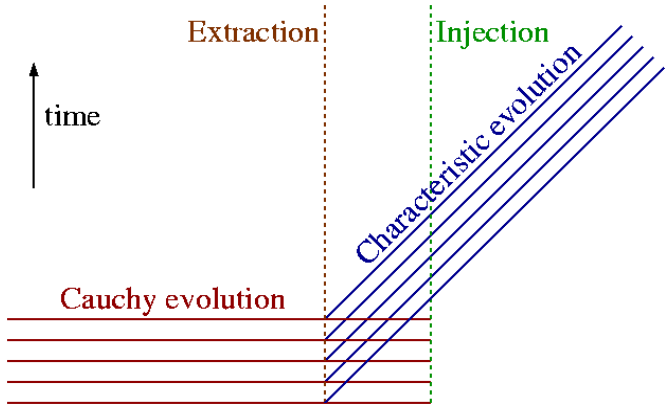
Yet another issue is that of consistent initialization.



Successful CCM schemes

Can it work?

Apparently.



Examples:

- 3D non-linear scalar evolution on a fixed Euclidean background

  Bishop et al, J. Comp. Phys. **136**, 140 (1997)

- 1D (non-linear) GR evolution, with Black Hole excision

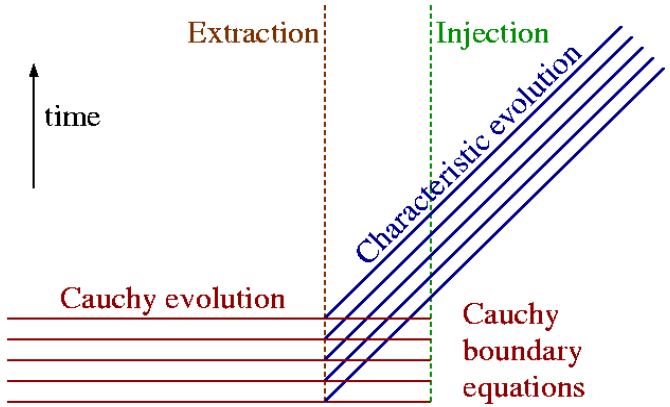
  Gomez et al, PRD **56**, 6310 (1997)

- *[2D (non-linear) GR evolution – Southampton group]*

CCM in linearized GR

Can it work?

It also works for 3D linearized GR.



CCM in linearized GR

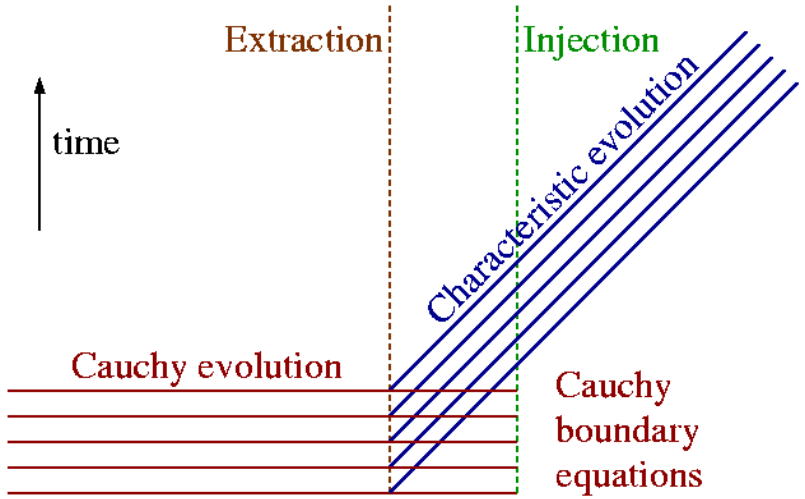
Details:

Cauchy evol: linearized harmonic

Cauchy bdry: enforce 4 constraints

Characteristic evol:

- linearized Bondi
- Gauge evolution – this allows knowledge of the Bondi -> Cartesian Jacobian

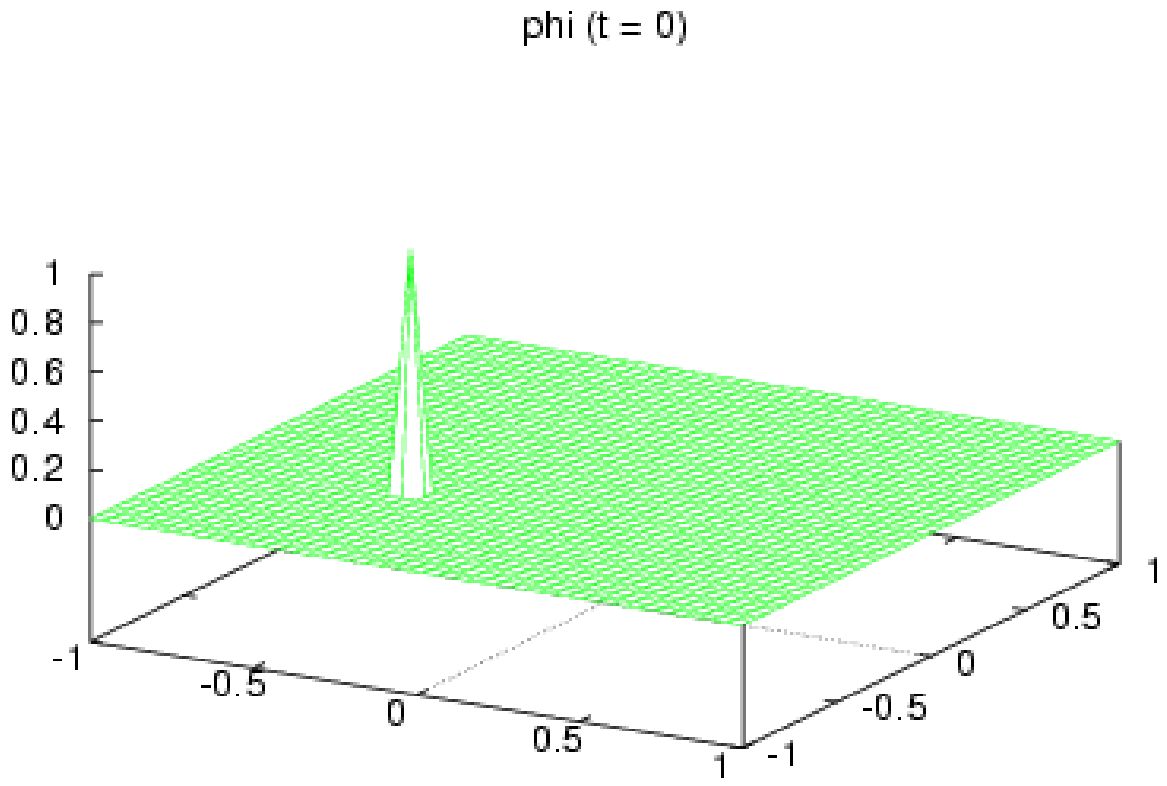


Extraction: polynomial interp & lots of algebra

Injection: “Sommerfeld”-style, normal to faces of the cube

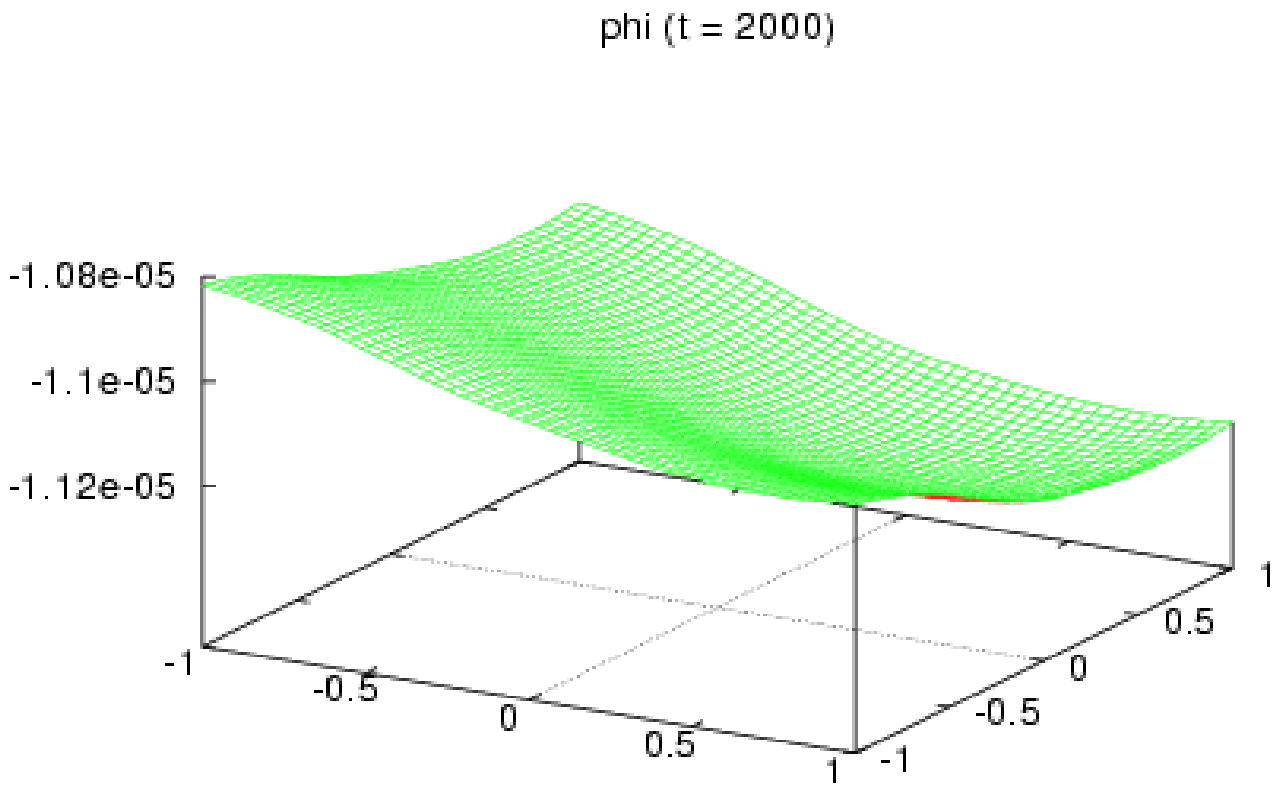
CCM in linearized GR

Result:



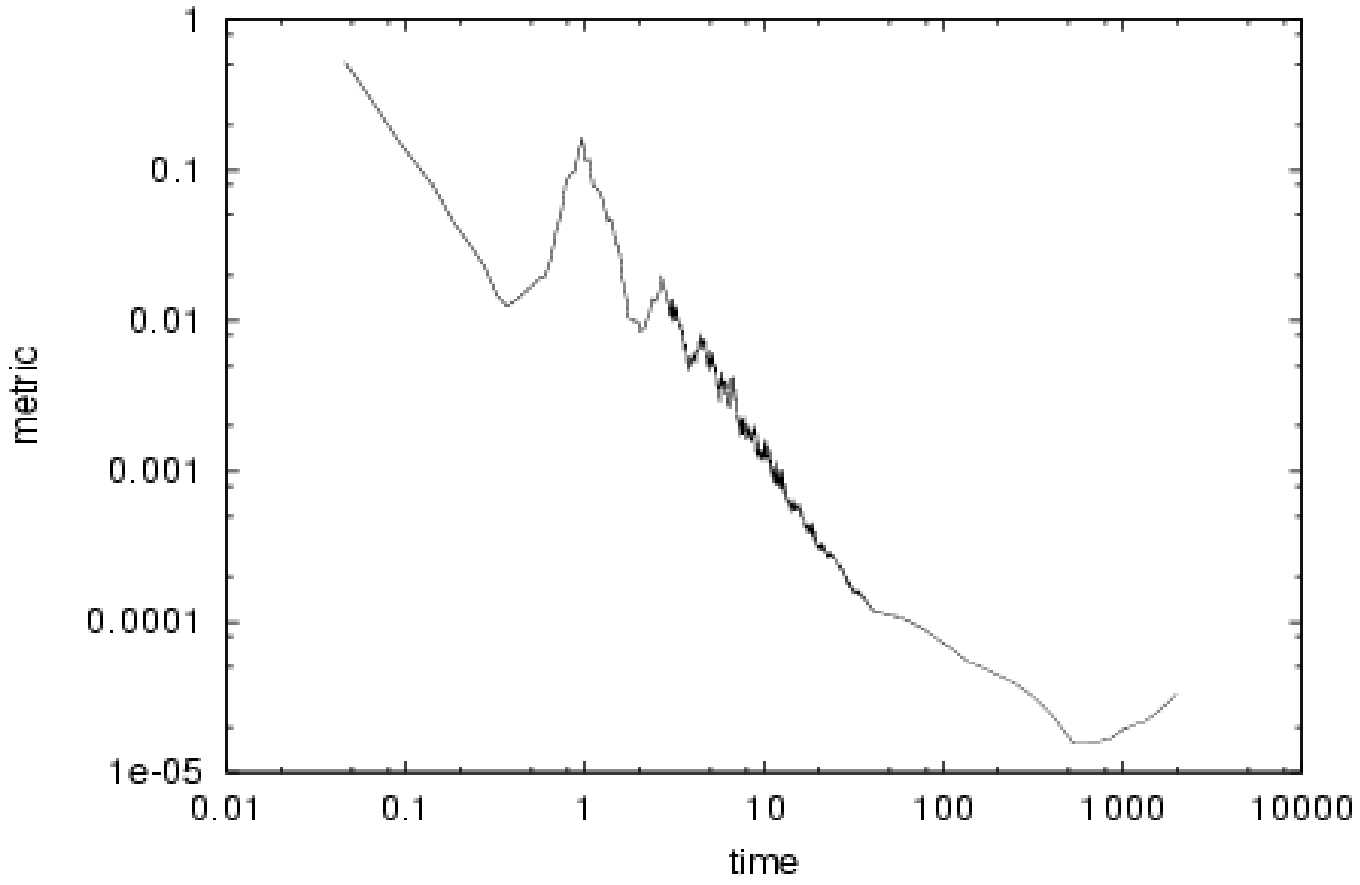
CCM in linearized GR

Result:



CCM in linearized GR

Result:



CCM in linearized GR

Lessons learned so far:

- Need working Cauchy (and Characteristic) initial-boundary code(s)
- Need constraint preserving boundary conditions (Cauchy boundary equations)
- Need finite distance between Injection and Extraction (?)
- Need Sommerfeld style Injection
- Need parallelization (Yosef is getting there)

Status of the non-linear Cauchy code

Numerical evolution scheme:

- 1^{st} differential order in time, 2^{nd} differential order in space
  (Does anybody know how to keep it fully 2^{nd} differential order?)

- Discretization: finite differencing (2^{nd} or 4^{th} order accurate)

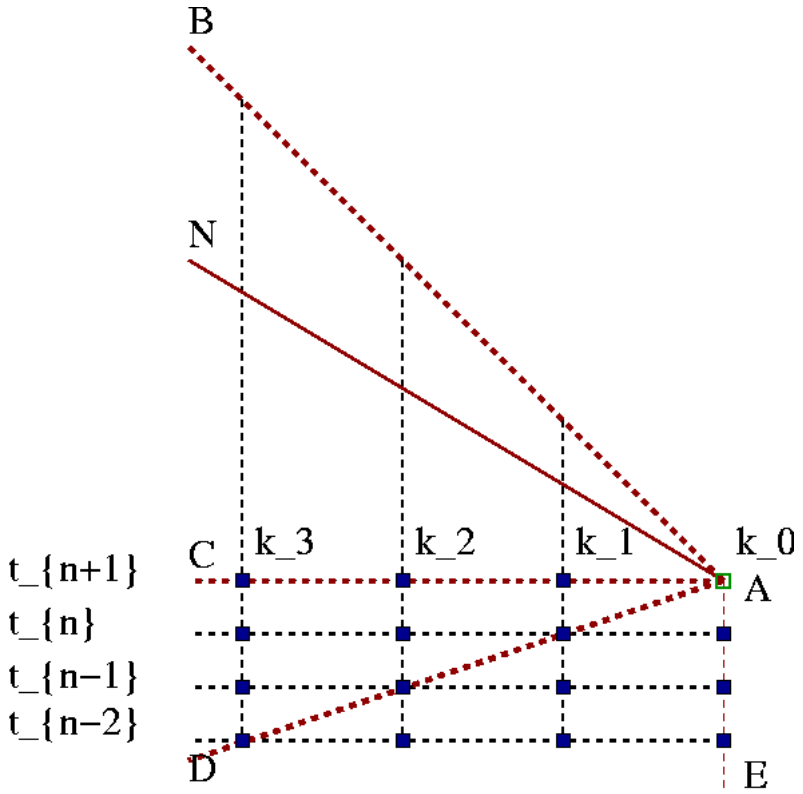
- Time-integration:
  - Iterative Crank-Nicholson
  - 3^{rd} and 4^{th} order accurate predictor-corrector schemes
  - 4^{th} order Runge Kutta

Status of the non-linear Cauchy code

Numerical boundary scheme:

- A first major issue is the need of a good Neumann algorithm

1) Sideways algorithm:

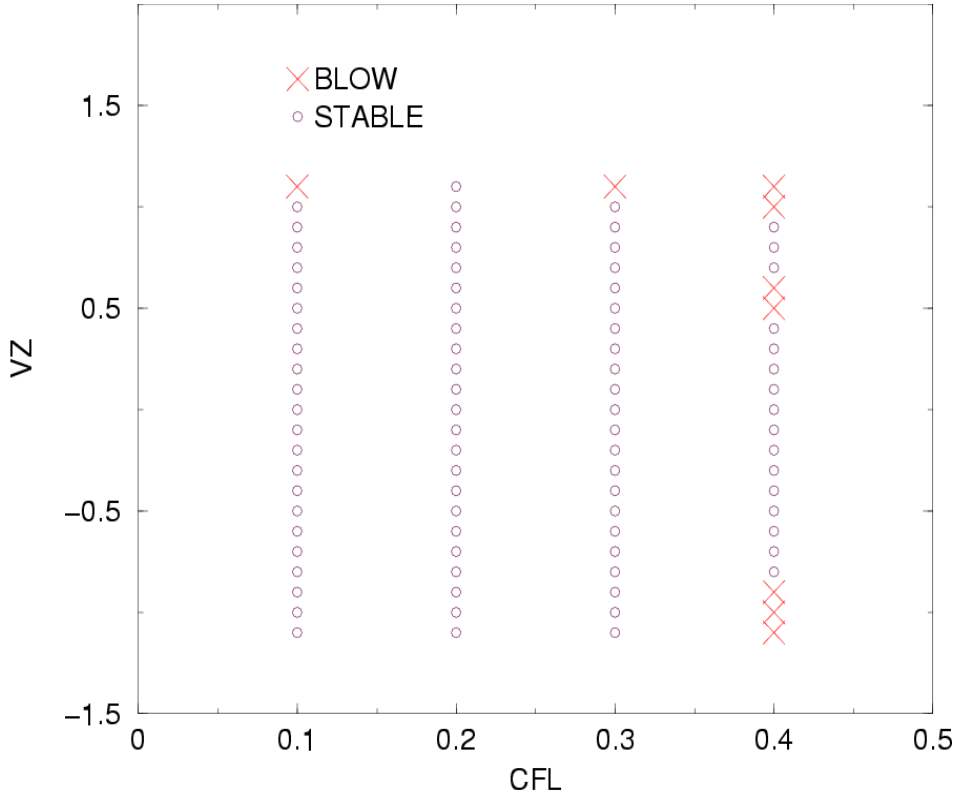


Status of the non-linear Cauchy code

Numerical boundary scheme:

- A first major issue is the need of a good Neumann algorithm

1) Sideways algorithm:
   - works (too) well in 1D
   - uses future time-levels
   - no obvious (smooth) extension to edges



Status of the non-linear Cauchy code

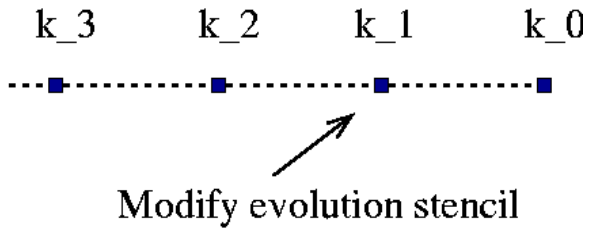
Numerical boundary scheme:

- A first major issue is the need of a good Neumann algorithm

  1) Sideways algorithm

  2) "Evolution-based" algorithm:

  - simple stencil
  - good for edges
  - provides no boundary value for evolution variables

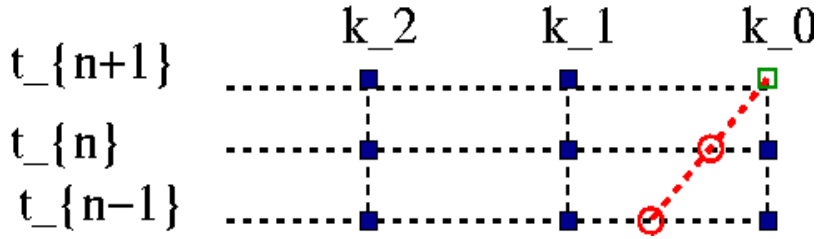


Status of the non-linear Cauchy code

Numerical boundary scheme:

- A first major issue is the need of a good Neumann algorithm

1) Sideways algorithm
2) "Evolution-based" algorithm
3) Characteristic algorithm:
   - no future levels
   - Generalization of 1st diff. order schemes
   - no good edge algorithm



Status of the non-linear Cauchy code

Numerical boundary scheme:

- A first major issue is the need of a good Neumann algorithm

  1) Sideways algorithm

  2) "Evolution-based" algorithm

  3) Characteristic algorithm

- The boundary equations ask for both the metric components and their Neumann derivatives at the boundary point. As a result, we plan to use a combination of the 2nd and 3rd algorithms.

Conclusion

- CCM works in linearized (harmonic) GR
- Implementation of a non-linear (harmonic) Cauchy evolution-boundary algorithm is well under way
- We already have:
  - A non-linear characteristic code
  - A non-linear extraction module (on 1 CPU)
- Still need to work out a non-linear injection module (in it's current setup)