Acceleration of Time Integration

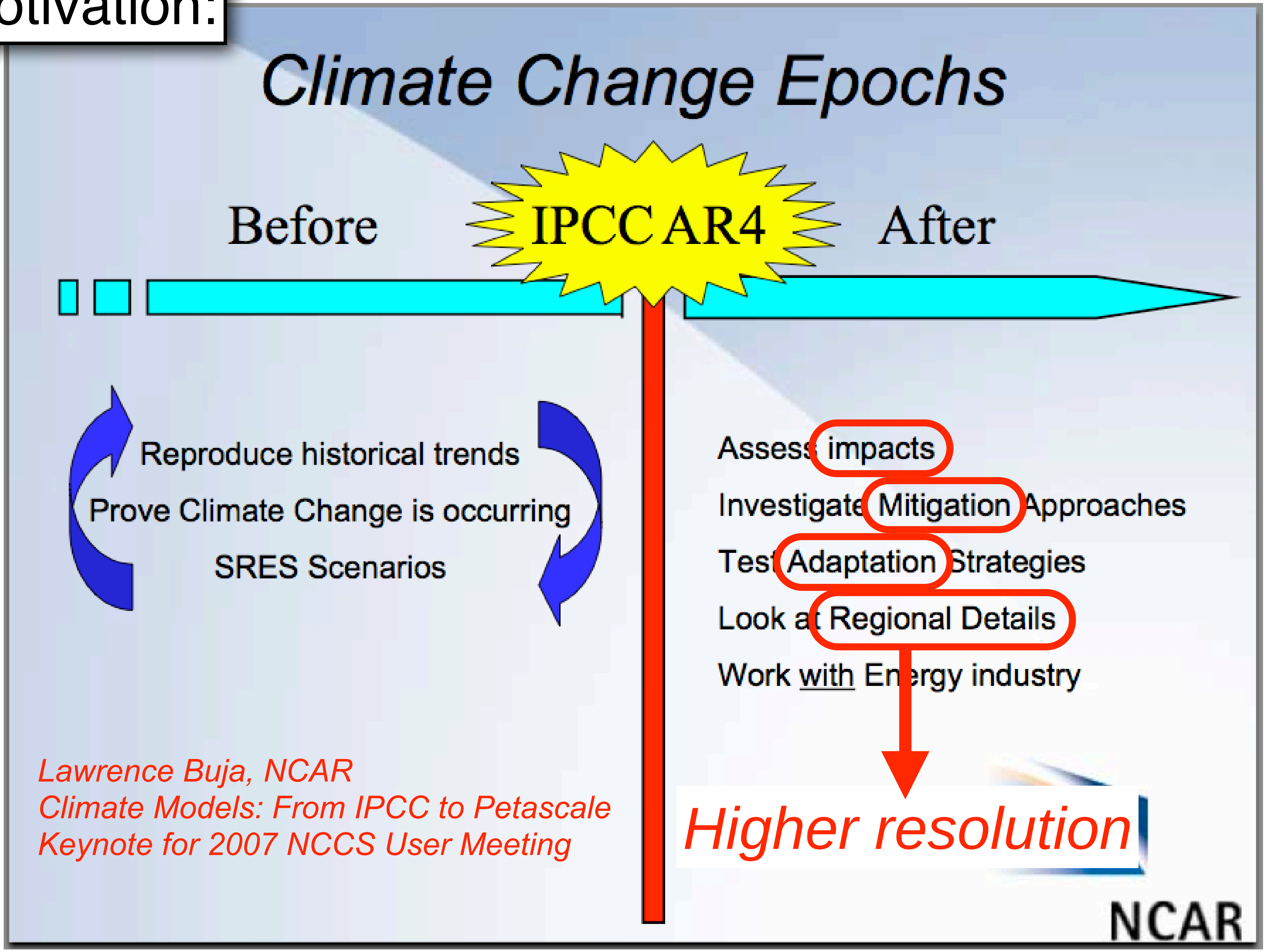
edited version, with extra images removed

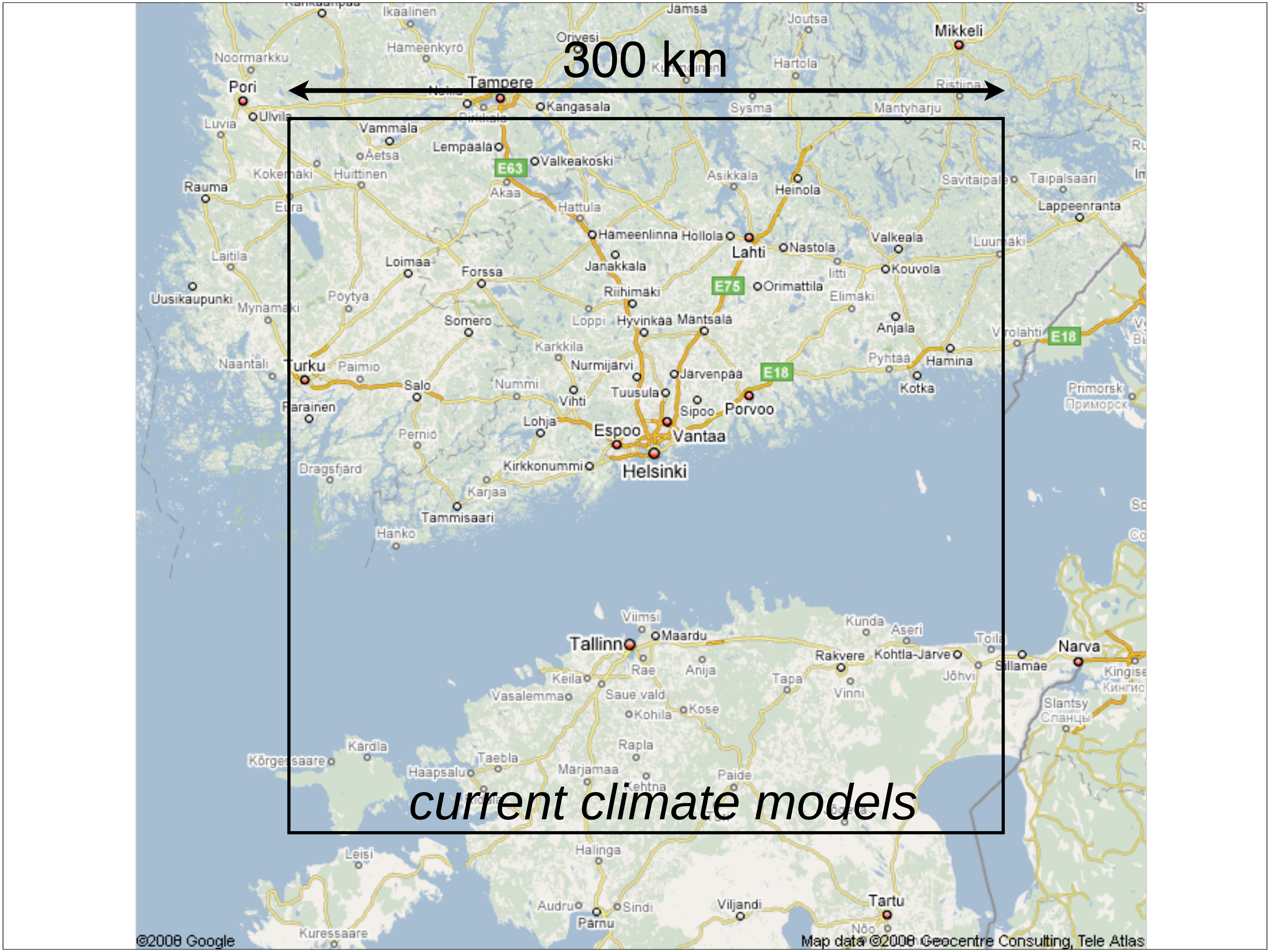
Rick Archibald, John Drake, Kate Evans, Doug Kothe, Trey White, Pat Worley

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725.

This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

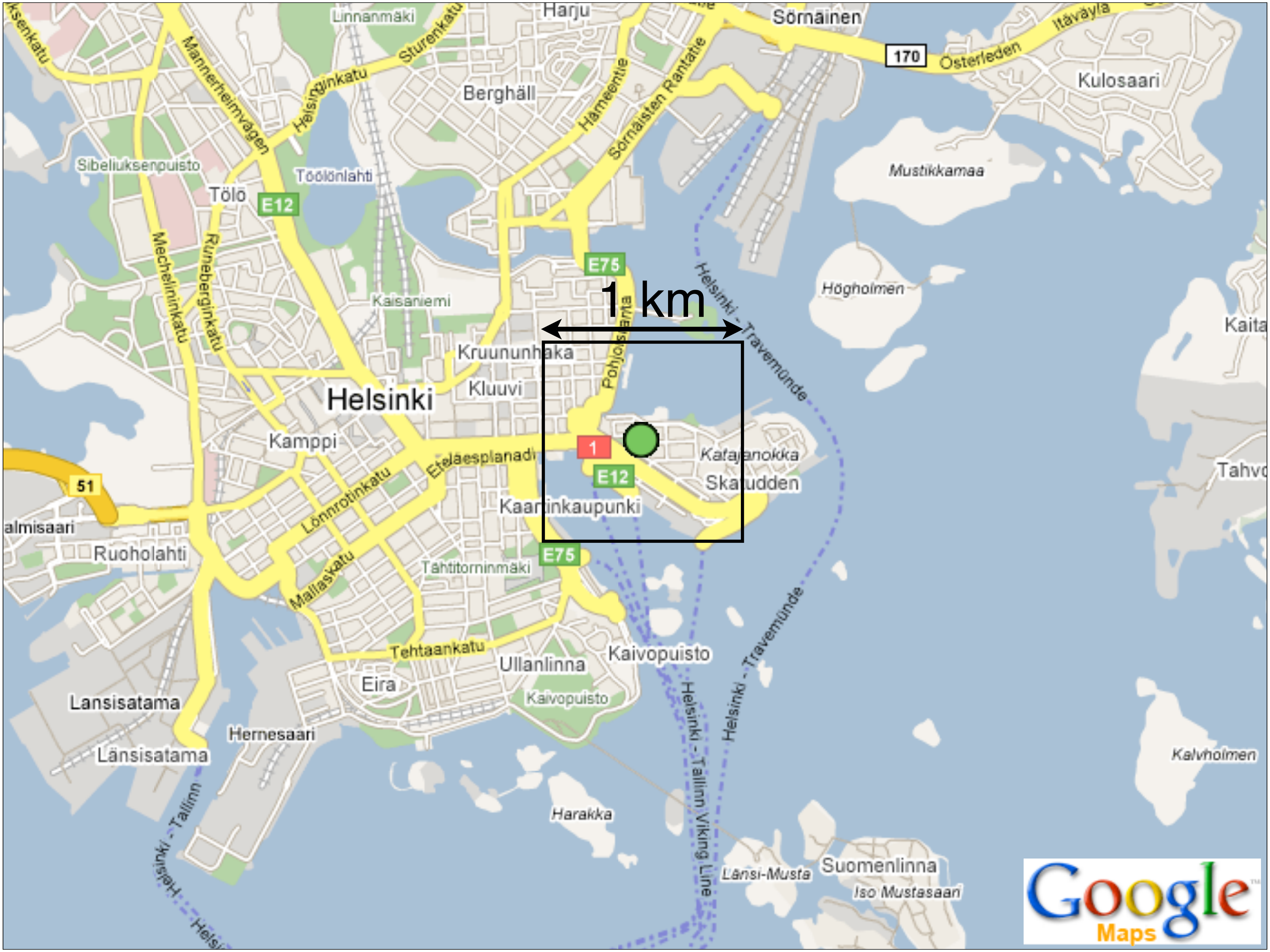
Motivation:


Climate Change Epochs
Before
IPCC AR4
After
Reproduce historical trends
Prove Climate Change is occurring
SRES Scenarios
Assess impacts
Investigate Mitigation Approaches
Test Adaptation Strategies
Look at Regional Details
Work with Energy industry
Higher resolution
Lawrence Buja, NCAR
Climate Models: From IPCC to Petascale
Keynote for 2007 NCCS User Meeting
NCAR


300 km
current climate models
©2008 Google
Map data ©2008 Geocentre Consulting, Tele Atlas

"More importantly, because the assumptions that are made in the development of parameterizations of convective clouds and the planetary boundary layer are seldom satisfied, the atmospheric component model must have sufficient resolution to dispense with these parameterizations. This would require a horizontal resolution of 1 km."

http://www.geo-prose.com/projects/pdfs/petascale_science.pdf

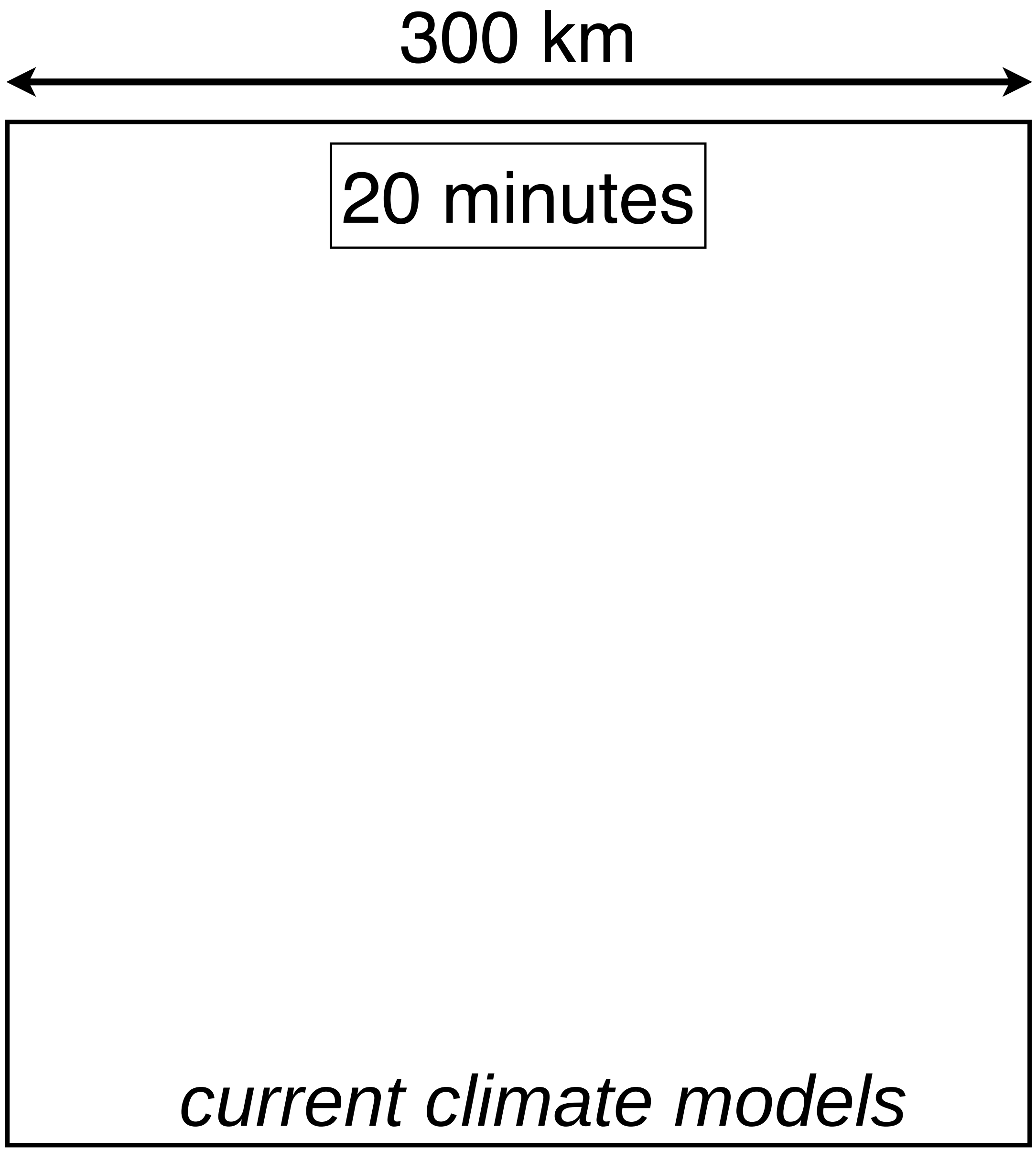

1 km
Helsinki
Harju
Sörnainen
170
Österleden
Itäväylä
Kulosaari
Berghäll
Linnanmäki
Sturenkatu
Helsinginkatu
Mannerheimvägen
Hämeentie
Sörnaisten Rantatie
Sibeliuksenpuisto
Töölönlahti
Tölö
E12
Mustikkamaa
E75
Högholmen
Kaisaniemi
Kruununhaka
Kluuvi
Pohjoisranta
Mechelininkatu
Runeberginkatu
Kamppi
Eteläesplanadi
1
Katajanokka
Skatudden
Lönnrotinkatu
Kaartinkaupunki
51
Ruoholahti
Mallaskatu
Tähtitorninmäki
Helsinki - Travemünde
Tehtaankatu
Ullanlinna
Kaivopuisto
Eira
Lansisatama
Länsisatama
Hernesaari
Harakka
Helsinki - Tallinn Viking Line
Länsi-Musta
Suomenlinna
Iso Mustasaari
Kalvholmen
Google
Maps

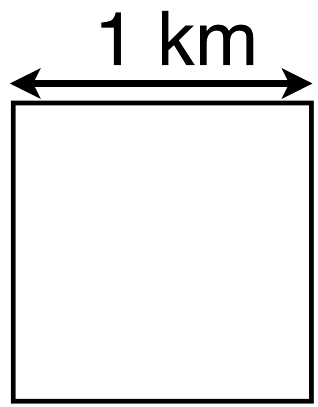
TIME BARRIER

Current climate models use ***explicit*** time integration

If resolution goes up

the time step **must** go down!


300 km
20 minutes
current climate models



4 seconds!

Extreme-scale systems will provide unprecedented parallelism!

But

performance of individual processes has stagnated

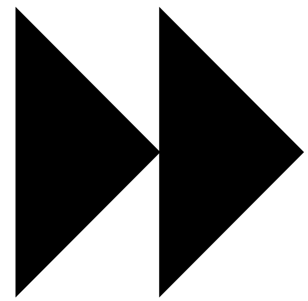
4-second
time step...

multi-century simulation?

Fast Forward

Overcoming the time barrier

- Fully implicit time integration
  - Stable for big time steps
- Parallel in time
  - Time is the biggest dimension
- New discretizations
  - Better time accuracy



How to build a new climate model

1. Start with shallow-water equations on the sphere

$$\frac{\partial h^* \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} h^* \mathbf{v}) = -f \hat{\mathbf{k}} \times h^* \mathbf{v} - g h^* \nabla h$$

$$\frac{\partial h^*}{\partial t} + \nabla \cdot (h^* \mathbf{v}) = 0$$

$$h = h^* + h_s$$

They mimic full equations for atmosphere and ocean

How to build a new climate model

2. Prove yourself on standard tests

Defined by Williamson, Drake, Hack, Jakob, and Swarztrauber in 1992 (148 citations)

How to build a new climate model

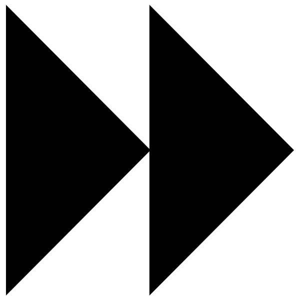
3. Proceed to 3D tests and inclusion in a full model

That’s all there is to it!

Fast Forward

Overcoming the time barrier

- Fully implicit time integration
  - Stable for big time steps
- Parallel in time
  - Time is the biggest dimension
- New discretizations
  - Better time accuracy



Explicit versus implicit

y'	y
State of simulation at next time step	State of simulation at current time
Values are unknown	Values are known

$$y' = f(y)$$

Explicit

Compute unknown directly from known

Explicit good and bad

- Good
  - Highly parallel
  - Nearest-neighbor communication
- Bad
  - Numerically unstable (blows up) for $\Delta t > O(\Delta x)$
  - Increase resolution → decrease Δx → decrease Δt

Explicit versus implicit

y'	y
State of simulation at next time step	State of simulation at current time
Values are unknown	Values are known

$$y' = ay + f(y')$$

Implicit

Solve a (nonlinear) system of equations

Implicit bad and good

- Bad
  - Must solve a (nonlinear) system of equations
- Good
  - Numerically stable for arbitrary time steps
- Ugly
  - Still need to worry about accuracy (for big time steps)

Implicit + shallow water

(Kate Evans)

- Start with HOMME shallow-water code
- Convert explicit formulation to implicit
- Solve with Trilinos

HOMME

- High-Order-Method Modeling Environment
- Principal developers
  - NCAR: John Dennis, Jim Edwards, Rory Kelly, Ram Nair, Amik St-Cyr
  - Sandia: Mark Taylor
- Cubed-sphere grid
- Spectral-element formulation (and others)
- Shallow-water equations (and others)

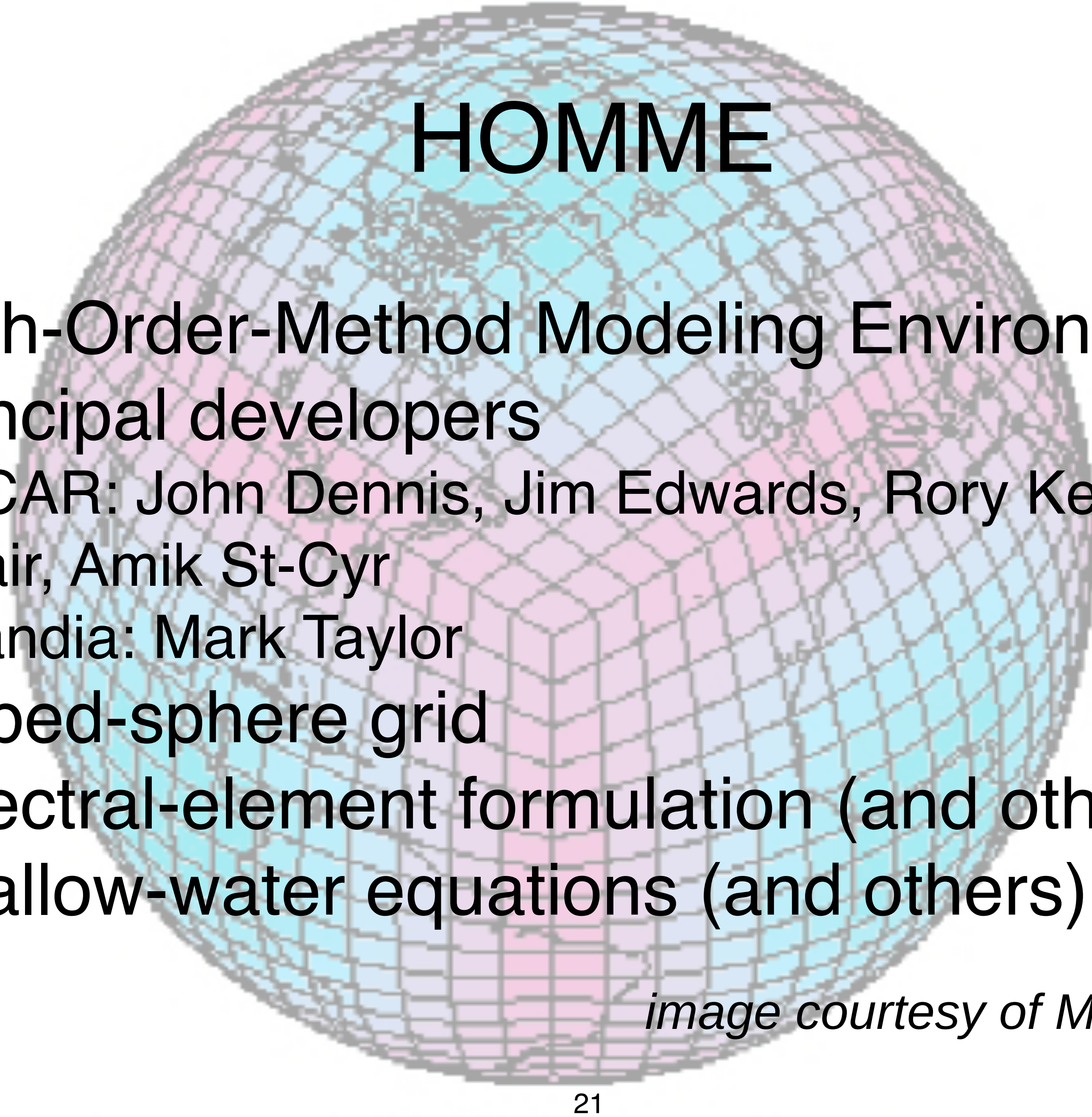


image courtesy of Mark Taylor

Jacobian-Free Newton Krylov (JFNK)

Jacobian-Free Newton Krylov

- What we want: $F(y)=0$
- What we have: $F(y)\neq 0$
- Find the change in F as y changes
  - Jacobian, J, derivative of a vector
- Approximate correction: $F(y+\Delta y)=0$
  $0=F(y+\Delta y)\approx F(y)+J\Delta y$
  $F(y)=-J\Delta y$
- Solve the *linear* system for Δy and add to y
- Repeat until $F(y)\approx 0$

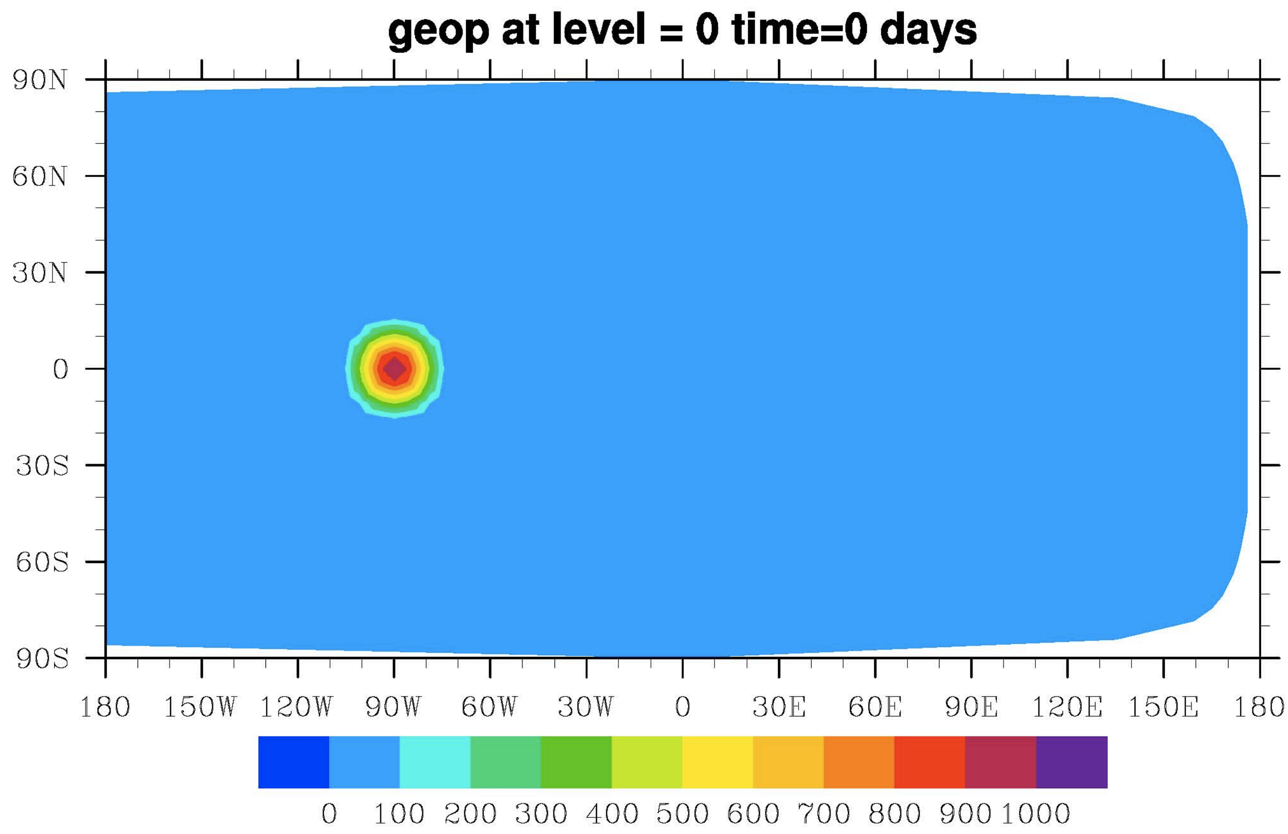
Jacobian-Free Newton Krylov

- $F(y) = -J\Delta y$
- Solve for Δy using an iterative linear solver
- Krylov subspace methods
  - Take a guess at Δy
  - Calculate how bad it is (residual)
  - Use residual to improve guess
  - Iterate, using past residuals and Russian Navy know-how to improve guess
  - Stop when residual is small (guess is good)

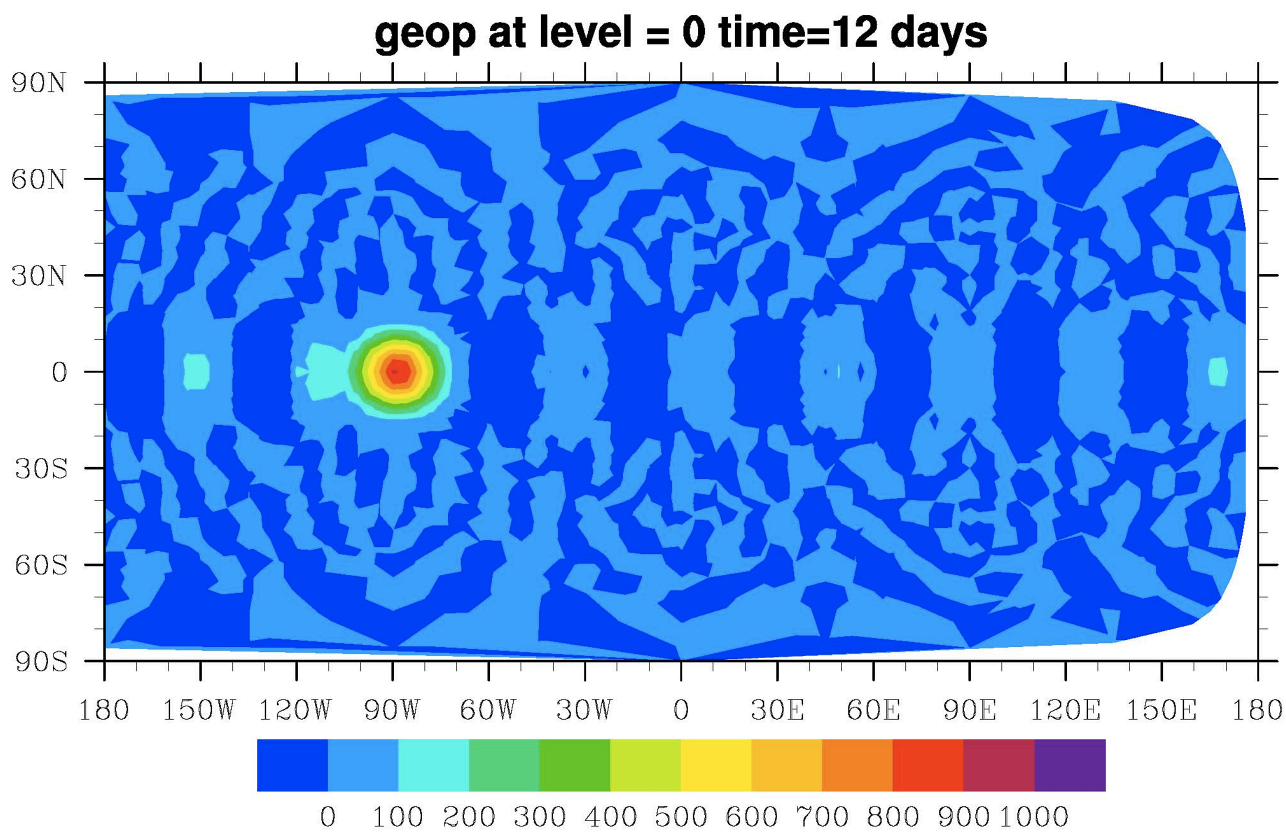
Jacobian-Free Newton Krylov

- Don't compute the Jacobian
- Approximate it using finite differences
  $J\Delta y \approx (F(y+\varepsilon\Delta y)-F(y))/\varepsilon$
  ε is a small number
- Can be much cheaper to calculate, only need F

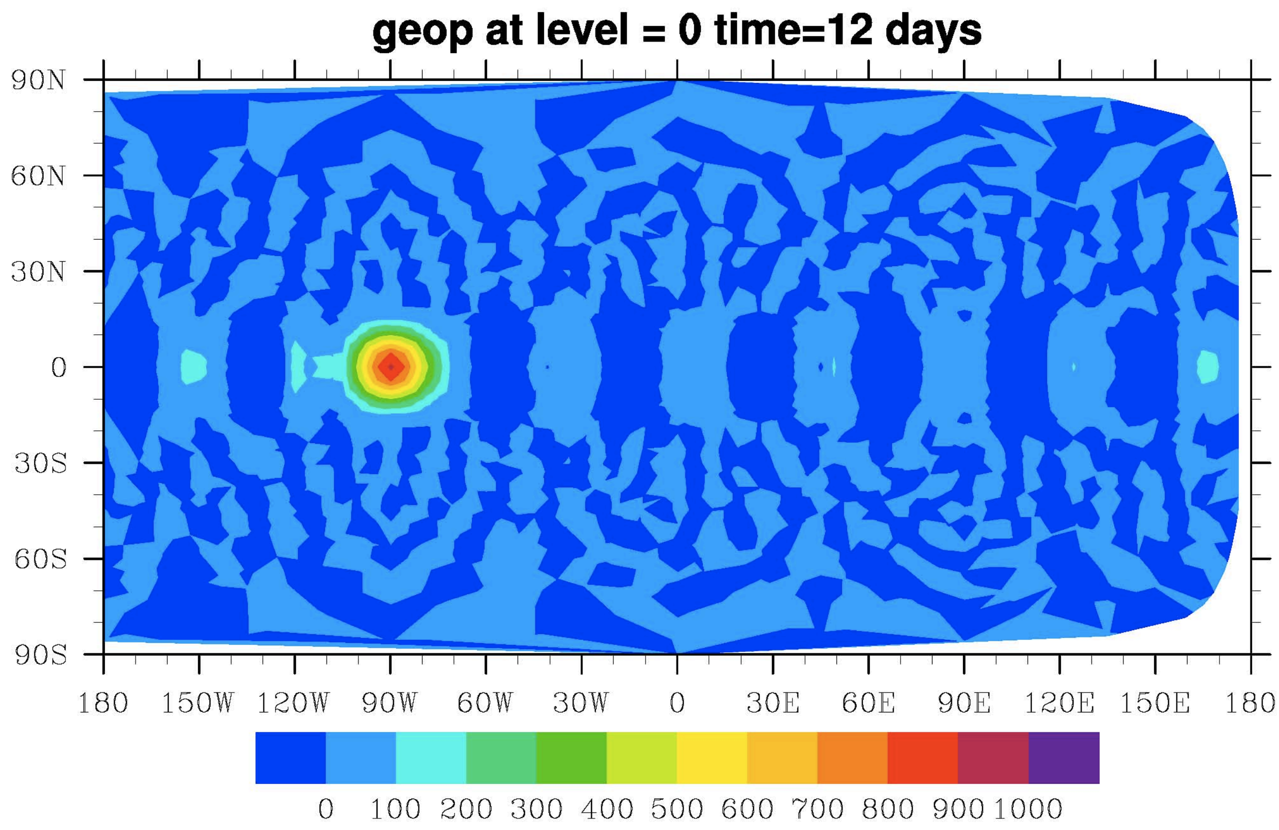
Test case 1: cosine bell initial condition



Test case 1: cosine bell explicit solver with “hyperviscosity”



Test case 1: cosine bell implicit solver, no preservatives



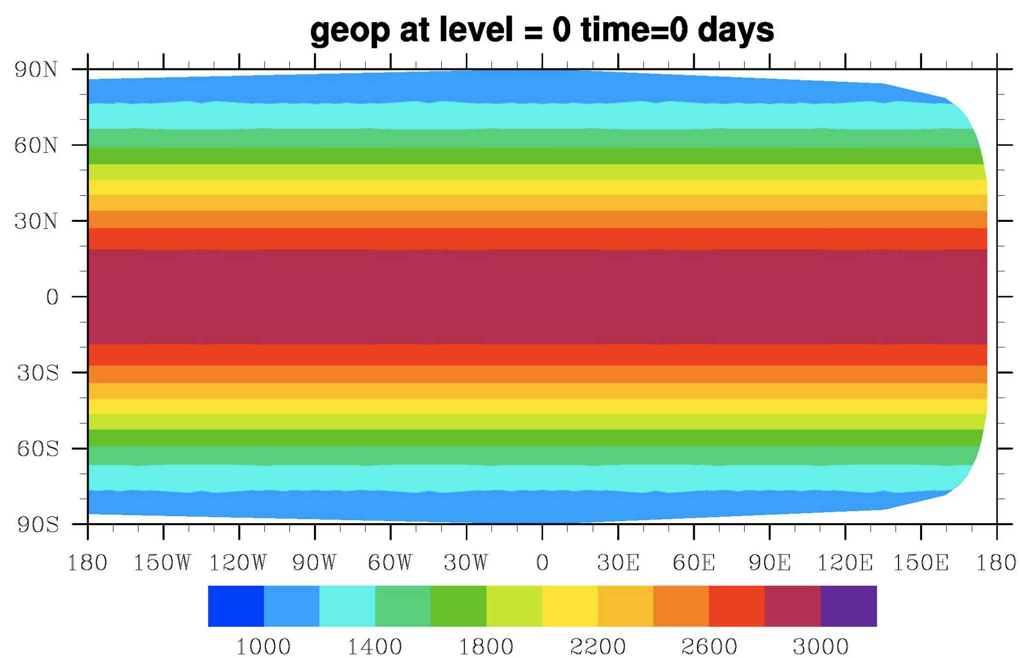
Test case 1: cosine bell implicit versus explicit

- Implicit takes many iterations per time step
- But 2-hour time step instead of 2-minute
- Similar error at the end
- 40% shorter runtime (no preconditioner)

Performance result #1!

Test case 2: steady state

- 12 simulated days
- Explicit
  - 4-minute time step
  - 28s runtime
- Implicit
  - 12-day time step
  - 3.6s runtime

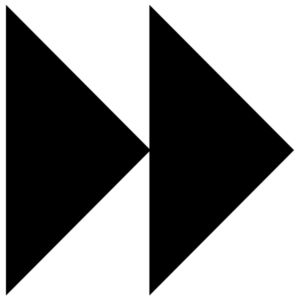


Performance result #2!

Fast Forward

Overcoming the time barrier

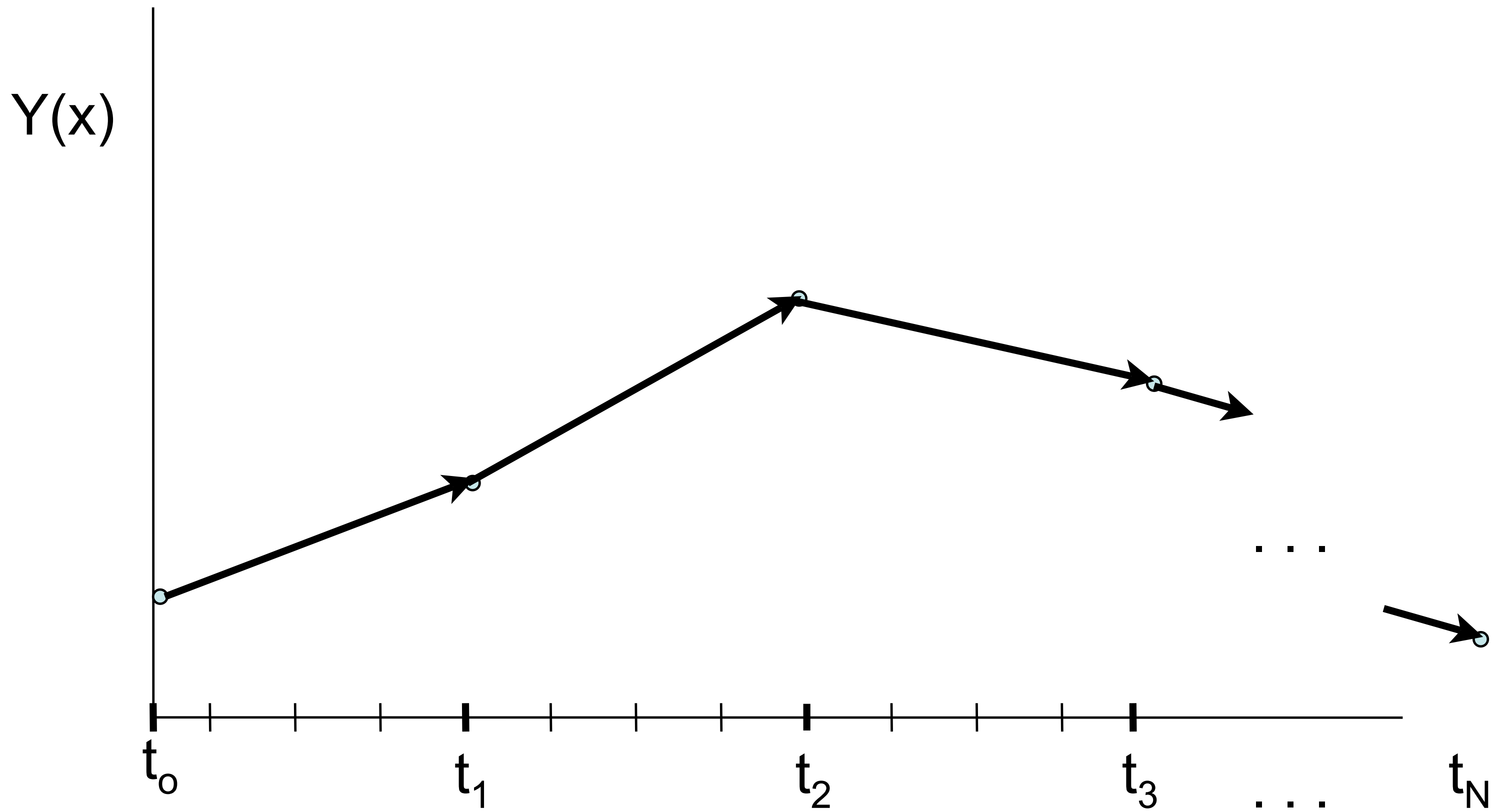
- Fully implicit time integration
  - Stable for big time steps
- Parallel in time
  - Time is the biggest dimension
- New discretizations
  - Better time accuracy



Parareal
(my interest)

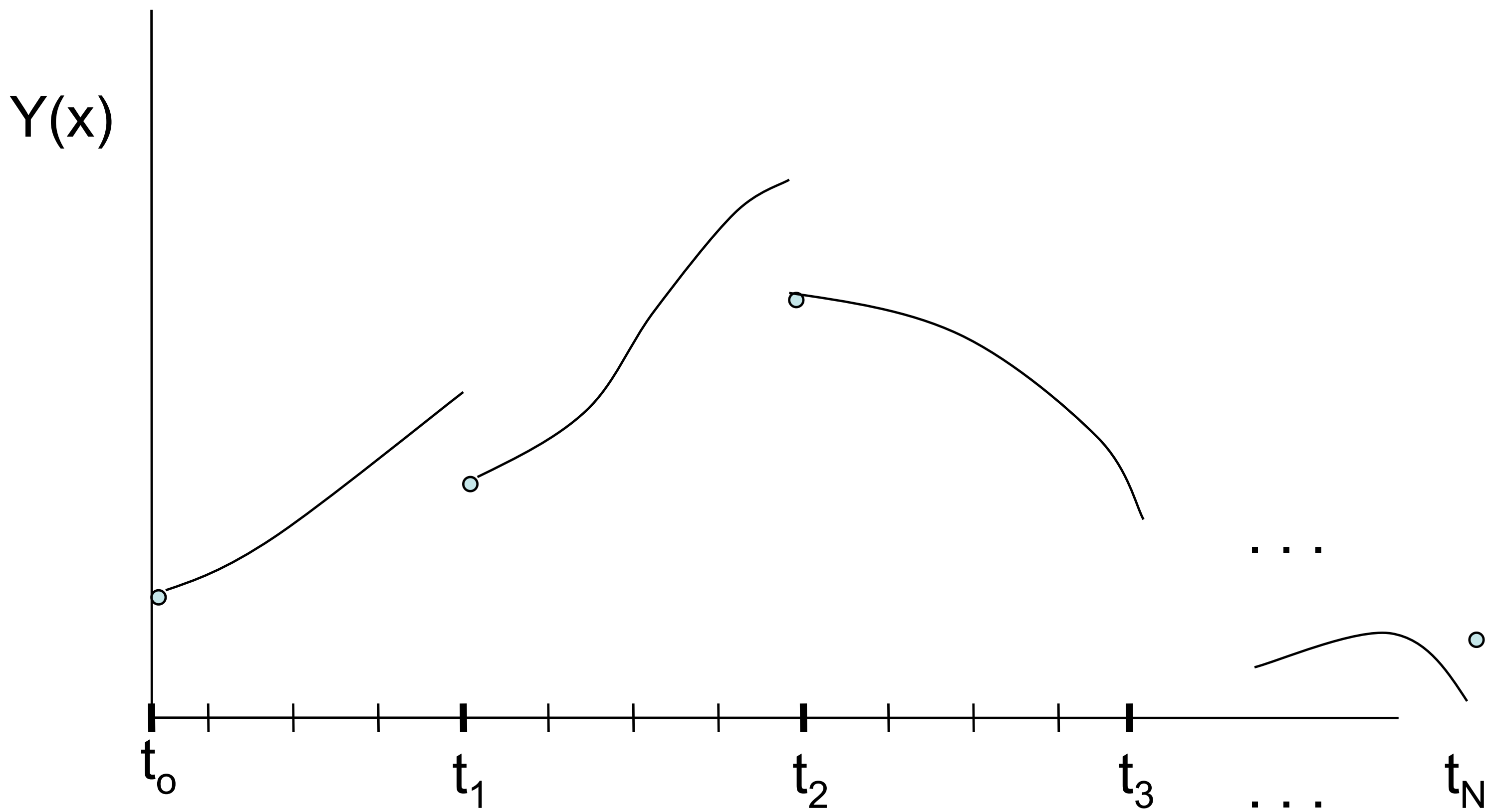
- Algorithm published in 2001 by Jacques-Louis Lions,
  Yvon Maday,
  and Gabriel Turinici
- Variants successful for range of applications
  - Navier-Stokes
  - Structural dynamics
  - Reservoir simulation

Parareal



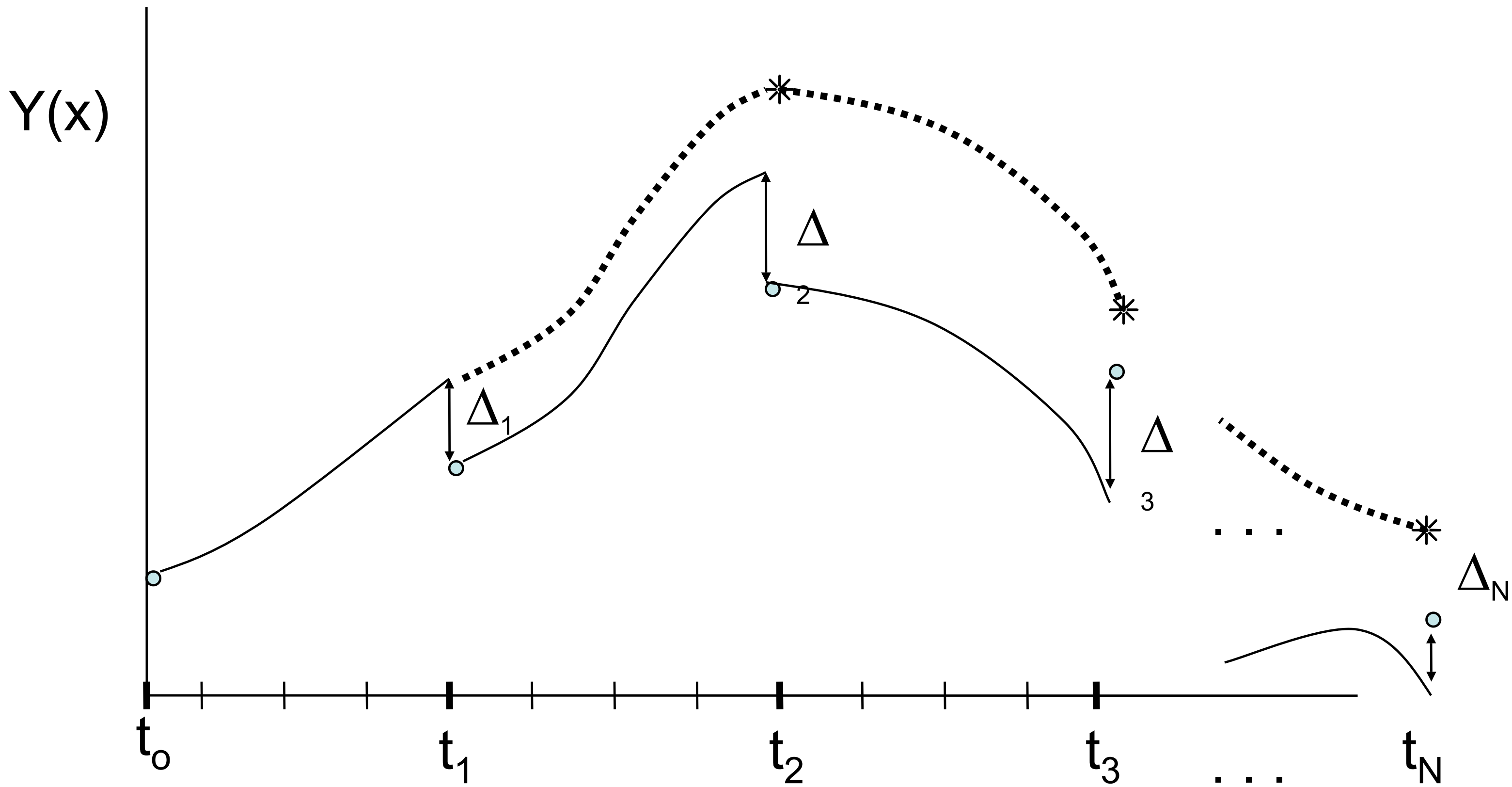
Solve serially at coarse time steps

Parareal



Compute fine time integrations between coarse steps in parallel

Parareal



Propagate and accumulate fine-time corrections at coarse scale

Parareal

- Iterate until corrections are negligible
- Published results by others: 2-3 iterations

My parareal experience

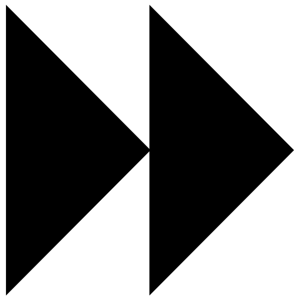
- Numerically unstable for pure advection
- Confirms theoretical result by Maday and colleagues
- Should work for Burgers' equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0 \qquad \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0$$

Fast Forward

Overcoming the time barrier

- Fully implicit time integration
  - Stable for big time steps
- Parallel in time
  - Time is the biggest dimension
- New discretizations
  - Better time accuracy



Curvelets
(Rick Archibald)

- Compact in space
  (like finite elements)
- Preserve shape
  (like Fourier waves)
- Might allow $\Delta t \sim \Delta x^{1/2}$

Curvelets

But they require a periodic domain

Multi-wavelets
(Rick Archibald)

- Adaptive
- Designed for refinement
- Strong error bounds
  - Control refinement and coarsening
- Requires integral formulation
  - Translation: more theoretical work to do
- Work just getting started

Finite differences
(my interest)

High-order single-step time integration

Consider advection:

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0$$

Time integration using a Taylor series in small Δt

$$u = u' - \Delta t\frac{\partial u'}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u'}{\partial t^2} - \frac{\Delta t^3}{6}\frac{\partial^3 u'}{\partial t^3} + O(\Delta t^4)$$

Implicit

High-order single-step time integration

- Replace time derivatives with space derivatives
- Why?
  Many grid points in space, few in time (2)
  So you can form high-order space derivatives
- How?
  Use the governing equation

$$\frac{\partial u'}{\partial t} + v\frac{\partial u'}{\partial x} = 0 \rightarrow \frac{\partial u'}{\partial t} = -v\frac{\partial u'}{\partial x}$$

High-order single-step time integration

$$u = u' + v\Delta t \frac{\partial u'}{\partial x} + \frac{v^2 \Delta t^2}{2} \frac{\partial^2 u'}{\partial x^2} + \frac{v^3 \Delta t^3}{6} \frac{\partial^3 u'}{\partial x^3} + O(\Delta t^4)$$

- Got high-order space derivatives?
- Get high accuracy in time for free*!
- Just 2 points in time: this one and next one
  - Save memory
  - Save I/O storage space and bandwidth
  - Easy startup from initial condition

* Since flops are free.

High-order single-step time integration

- Explicit and implicit work for advection
- Explicit works for Burgers' equation
- Implicit and semi-implicit for Burgers' under development
- Goal is shallow-water equations

Would you believe I cut some topics from the talk?

- High-order methods for compact stencils
- Single-cycle multi-level linear solvers