Efficient Parallel Algorithms for Climate Data Analysis

Kara Peterson Pavel Bochev

Sandia National Laboratories

2013 AGU Fall Meeting

December 13, 2013

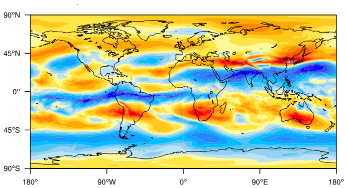
Collaborators: Robert Jacob, Tim Tautges, Jayesh Krishna, Iulian Grindeanu (ANL)

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation

- Climate models are generating huge amounts of data making even simple analysis tasks time consuming

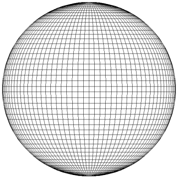


Monthly output file size:

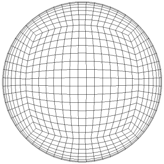
1 degree CAM-FV - 223 MB
1/2 degree CAM-FV - 821 MB
1/4 degree CAM-FV - 2.9 GB

Motivation

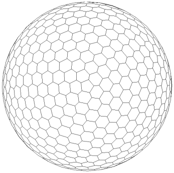
- Climate models are generating huge amounts of data making even simple analysis tasks time consuming
- Models are moving away from traditional structured lat/lon grids



structured lat/lon



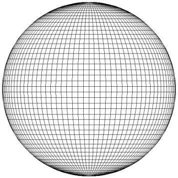
cubed sphere



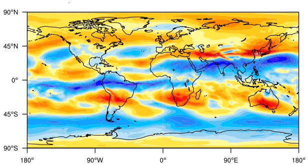
Voronoi

Motivation

- Climate models are generating huge amounts of data making even simple analysis tasks time consuming
- Models are moving away from traditional structured lat/lon grids
- Widely used analysis tools like NCAR Command Language (NCL) were designed to run in serial on structured lat/lon grids

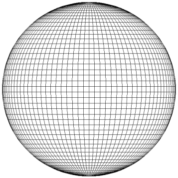


structured lat/lon

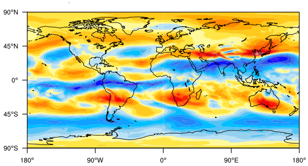


Motivation

- Climate models are generating huge amounts of data making even simple analysis tasks time consuming
- Models are moving away from traditional structured lat/lon grids
- Widely used analysis tools like NCAR Command Language (NCL) were designed to run in serial on structured lat/lon grids



structured lat/lon



We need efficient algorithms and methods for postprocessing large volumes of climate data on structured and unstructured meshes.

ParGAL

Parallel Gridded Analysis Library

- Contains parallel versions of standard post-processing functions
- Extends analysis to grids other than rectilinear lat/lon
- Component of the Parallel NCAR Command Language (ParNCL)

parVis

Parallel Analysis Tools and New Visualization Techniques for Ultra-Large Climate Data Sets

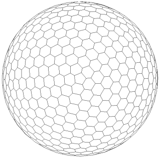
https://trac.mcs.anl.gov/projects/parvis

ParVis Team: Robert Jacob, Jayesh Krishna, Xiabing Xu, Sheri Mickelson, Tim Tautges, Iulian Grindeanu, Mike Wilde, Rob Latham, Ian Foster, Rob Ross, Mark Hereld, (ANL); Pavel Bochev, Kara Peterson, Mark Taylor (SNL); Jeff Daily, Karen Schuchardt, Jian Yin (PNNL); Don Middleton, Mary Haley, David Brown, Richard Brownrigg, Wei Huang, Dennis Shea, Mariana Vertenstein (NCAR), Kwan-Liu Ma, Jinrong Xie (UC-Davis)

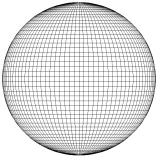
ParGAL Software Components

Mesh-Oriented datABase (MOAB)

- Library for representing and interfacing with mesh data
- Uses array-based storage for efficient access to fields associated with mesh entities
- Performs domain decomposition and supports parallel communication
- Provides mesh connectivity information for analysis algorithms



Supports polygonal meshes...



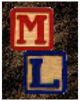
as well as structured meshes

https://trac.mcs.anl.gov/projects/ITAPS/wiki/MOAB

ParGAL Software Components

Trilinos: an object-oriented implementation of algorithms and enabling technologies for solving large-scale scientific and engineering problems

- Intrepid
    - Library for compatible discretizations of PDEs
    - FEM basis functions and integration rules on standard cell topologies
    - Reference to physical cell mappings
- ML
    - Multigrid preconditioner and solver library
    - Highly scalable ML preconditioners have been used on thousands of processors
- Epetra
    - Maps, multivectors, and sparse matrix containers for parallel linear algebra
    - Common matrix and vector operations



http://trilinos.org

Functions of Interest

$\zeta = \nabla \times \mathbf{v}$	Vorticity (ζ) from velocity field*
$\delta = \nabla \cdot \mathbf{v}$	Divergence (δ) from velocity field*
$\nabla \times \mathbf{v} = \nabla \cdot \nabla\psi$	Streamfunction (ψ) from velocity field
$\zeta = \nabla \cdot \nabla\psi$	Streamfunction (ψ) from vorticity
$\nabla \cdot \mathbf{v} = \nabla \cdot \nabla\chi$	Velocity potential (χ) from velocity field
$\delta = \nabla \cdot \nabla\chi$	Velocity potential (χ) from divergence

* In ParNCL 1.0.0b2

Functions of Interest

$\zeta = \nabla \times \mathbf{v}$	Vorticity (ζ) from velocity field*
$\delta = \nabla \cdot \mathbf{v}$	Divergence (δ) from velocity field*
$\nabla \times \mathbf{v} = \nabla \cdot \nabla\psi$	Streamfunction (ψ) from velocity field
$\zeta = \nabla \cdot \nabla\psi$	Streamfunction (ψ) from vorticity
$\nabla \cdot \mathbf{v} = \nabla \cdot \nabla\chi$	Velocity potential (χ) from velocity field
$\delta = \nabla \cdot \nabla\chi$	Velocity potential (χ) from divergence

NCL (spherical harmonics)

- Requires global data
- Requires regular lat/lon grid
- Serial

ParGAL/ParNCL (FEM or FVM)

- Applicable to subdomains
- Extensible to unstructured grids
- Parallel

* In ParNCL 1.0.0b2

Computing Vorticity and Divergence

with the Finite Element Method

L^2 projection:

Find $\zeta, \delta \in L^2$ s.t. $\int_\Omega \phi\zeta \mathrm{d}\Omega = \int_\Omega \phi(\nabla \times \mathbf{v})\mathrm{d}\Omega \qquad \int_\Omega \phi\delta \mathrm{d}\Omega = \int_\Omega \phi(\nabla \cdot \mathbf{v})\mathrm{d}\Omega$

Algorithm

- Given velocity field $\mathbf{v}$ with zonal and meridional components (u, v)
- Discretize using bilinear $H^1(\Omega)$-conforming finite elements
- Obtain discrete linear systems of the form $Mx = b$
- Solve with ML

Computing Vorticity and Divergence
with the Finite Element Method

L^2 projection:

Find $\zeta, \delta \in L^2$ s.t. $\int_\Omega \phi\zeta d\Omega = \int_\Omega \phi(\nabla \times \mathbf{v}) d\Omega \qquad \int_\Omega \phi\delta d\Omega = \int_\Omega \phi(\nabla \cdot \mathbf{v}) d\Omega$

Algorithm

- Given velocity field $\mathbf{v}$ with zonal and meridional components (u, v)
- Discretize using bilinear $H^1(\Omega)$-conforming finite elements
- Obtain discrete linear systems of the form $Mx = b$
- Solve with ML

For structured lat/lon grids:

$$\nabla \times \mathbf{v} = \frac{1}{r\cos\theta}\left(\frac{\partial v}{\partial \lambda} - \frac{\partial}{\partial \theta}(u\cos\theta)\right) \quad \nabla \cdot \mathbf{v} = \frac{1}{r\cos\theta}\left(\frac{\partial u}{\partial \lambda} + \frac{\partial}{\partial \theta}(v\cos\theta)\right)$$

Computing Vorticity and Divergence

with the Finite Element Method

L^2 projection:

Find $\zeta, \delta \in L^2$ s.t. $\int_\Omega \phi\zeta \mathrm{d}\Omega = \int_\Omega \phi(\nabla \times \mathbf{v})\mathrm{d}\Omega \qquad \int_\Omega \phi\delta \mathrm{d}\Omega = \int_\Omega \phi(\nabla \cdot \mathbf{v})\mathrm{d}\Omega$

Algorithm

- Given velocity field $\mathbf{v}$ with zonal and meridional components (u, v)
- Discretize using bilinear $H^1(\Omega)$-conforming finite elements
- Obtain discrete linear systems of the form $Mx = b$
- Solve with ML

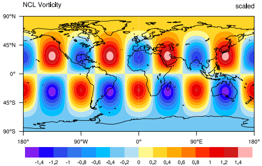
For cubed-sphere grids:

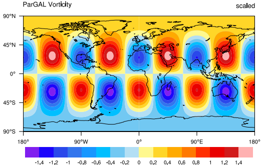
$$\nabla \times \mathbf{v} = \frac{1}{\sqrt{G}}\left(\frac{\partial u_2}{\partial x^1} - \frac{\partial u_1}{\partial x^2}\right) \quad \nabla \cdot \mathbf{v} = \frac{1}{\sqrt{G}}\left(\frac{\partial \sqrt{G}u^1}{\partial x^1} + \frac{\partial \sqrt{G}u^2}{\partial x^2}\right)$$

where G is the determinant of the metric tensor, (u_1, v_1) are covariant velocities, and (u^1, v^1) are contravariant velocities.

ParGAL Algorithm Convergence
Vorticity

$$u = \sin\theta\cos\lambda\cos\theta + 4\cos^3\theta\sin^2\theta\cos 4\lambda - \cos^5\theta\cos 4\lambda$$
$$v = \cos^2\theta\sin\lambda - \sin^2\theta\sin\lambda - 4\cos^3\theta\sin\theta\sin 4\lambda$$





$$\zeta = \frac{1}{r}\left(2\sin\theta + 30\cos^4\theta\sin\theta\cos 4\lambda\right)$$

ParGAL Vorticity Errors

Mesh	Size	l_2 error	rate	l_1 error	rate
4° FV	48×72	4.45×10^{-3}		2.18×10^{-3}	
2° FV	96× 144	1.35×10^{-3}	1.72	5.51×10^{-4}	1.98
1° FV	192× 288	4.47×10^{-4}	1.66	1.42×10^{-4}	1.97
T31	48×96	1.58×10^{-3}		1.26×10^{-3}	
T42	64×128	9.34×10^{-4}	1.83	7.36×10^{-4}	1.87
T85	128×256	2.61×10^{-4}	1.84	1.98×10^{-4}	1.89

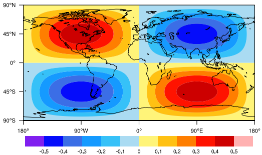
ParGAL Algorithm Convergence

Divergence

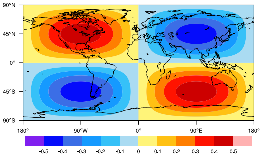
$$u = \sin\theta\cos\lambda\cos\theta + 4\cos^3\theta\sin^2\theta\cos 4\lambda - \cos^5\theta\cos 4\lambda$$

$$v = \cos^2\theta\sin\lambda - \sin^2\theta\sin\lambda - 4\cos^3\theta\sin\theta\sin 4\lambda$$

NCL Divergence



ParGAL Divergence



$$\delta = \frac{-6}{r}\left(\cos\theta\sin\theta\sin\lambda\right)$$

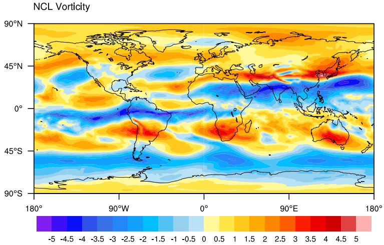
ParGAL Divergence Errors

Mesh	Size	l_2 error	rate	l_1 error	rate
4° FV	48×72	1.11×10^{-2}		5.37×10^{-3}	
2° FV	96× 144	3.37×10^{-3}	1.72	1.33×10^{-4}	1.99
1° FV	192× 288	1.12×10^{-3}	1.65	3.45×10^{-4}	1.97
T31	48×96	7.77×10^{-3}		4.30×10^{-3}	
T42	64×128	4.87×10^{-3}	1.62	2.51×10^{-3}	1.87
T85	128×256	1.71×10^{-3}	1.54	6.67×10^{-4}	1.90

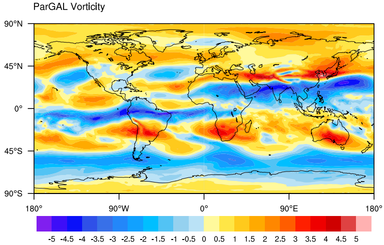
ParGAL versus Native NCL

For Structured Grids

NCL



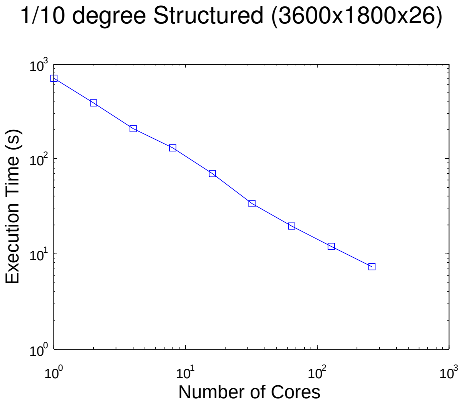
ParGAL



1 degree output file from CAM-FV

Timing: NCL - 4.02 s, ParGAL - 1p 8.59 s, 2p 4.36 s, 4p 2.58 s, 8p 1.88 s
Intel Xeon X5450 3GHz processors

Vorticity Computation Timing

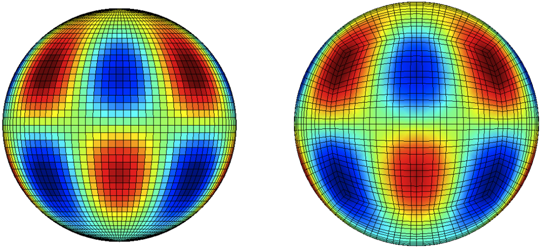


Execution time for computing vorticity on the fusion cluster at ANL

NCL 1 core - 6568 s

Vorticity on Unstructured Grid

$$\zeta = \frac{1}{r}\left(2\sin\theta + 30\cos^4\theta\sin\theta\cos 4\lambda\right)$$



Directly operating on native grid more efficient than interpolating to structured grid performing analyses on the interpolated data

Computing Streamfunction and Vel. Potential

with the Finite Element Method

Weak equations:

$$\text{Given } \mathbf{v}, \text{ find } \psi \in H^1 \text{ s.t.} \quad \int_\Omega \nabla\phi \cdot \nabla\psi \mathrm{d}\Omega = \int_\Omega \phi\nabla \times \mathbf{v}\mathrm{d}\Omega \quad \forall\phi \in H^1$$

$$\text{Given } \mathbf{v}, \text{ find } \chi \in H^1 \text{ s.t.} \quad \int_\Omega \nabla\phi \cdot \nabla\chi \mathrm{d}\Omega = \int_\Omega \phi\nabla \cdot \mathbf{v}\mathrm{d}\Omega \quad \forall\phi \in H^1$$

Algorithm

- Discretize using bilinear $H^1(\Omega)$-conforming finite elements
- Obtain discrete linear systems of the form $Kx = b$
- Solve with ML

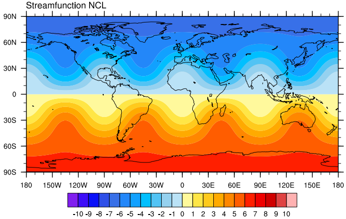
ParGAL versus Native NCL

Streamfunction

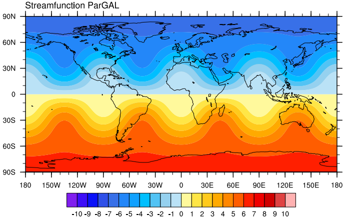
$$u = \sin\theta\cos\lambda\cos\theta + 4\cos^3\theta\sin^2\theta\cos 4\lambda - \cos^5\theta\cos 4\lambda$$
$$v = \cos^2\theta\sin\lambda - \sin^2\theta\sin\lambda - 4\cos^3\theta\sin\theta\sin 4\lambda$$

$$\psi = r\sin\theta + r\cos^4\theta\sin\theta\cos 4\lambda$$

NCL



ParGAL



1 degree CAM-FV grid

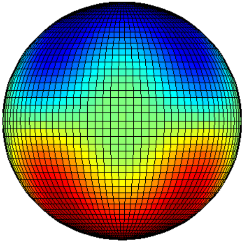
ParGAL versus Native NCL

Streamfunction

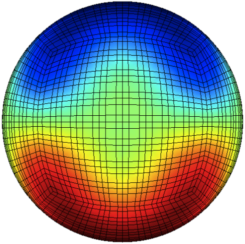
$$u = \sin\theta\cos\lambda\cos\theta + 4\cos^3\theta\sin^2\theta\cos 4\lambda - \cos^5\theta\cos 4\lambda$$
$$v = \cos^2\theta\sin\lambda - \sin^2\theta\sin\lambda - 4\cos^3\theta\sin\theta\sin 4\lambda$$

$$\psi = r\sin\theta + r\cos^4\theta\sin\theta\cos 4\lambda$$

Structured lat/lon



Cubed sphere



ne8np4 cubed sphere grid

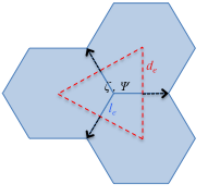
Computing Streamfunction

with the Finite Volume Method

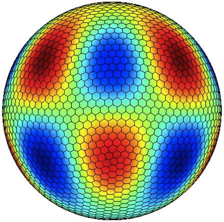
Given ζ, find ψ s.t.

$$\int_{\kappa_{tri}} \zeta \mathrm{dA} = -\int_{S_{tri}} \nabla\psi \cdot \mathbf{n}\mathrm{dS}$$

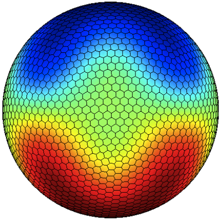
Integrate over dual mesh (triangles)



Vorticity (ζ)



Streamfunction (ψ)



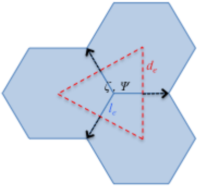
Computing Streamfunction

with the Finite Volume Method

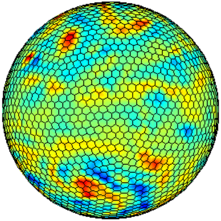
Given ζ, find ψ s.t.

$$\int_{\kappa_{tri}} \zeta \mathrm{dA} = -\int_{S_{tri}} \nabla\psi \cdot \mathbf{n}\mathrm{dS}$$

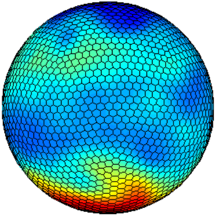
Integrate over dual mesh (triangles)



Vorticity (ζ)

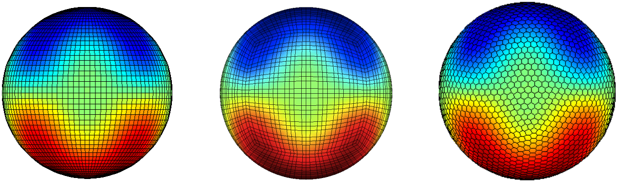


Streamfunction (ψ)



Conclusions

- We have implemented parallel algorithms for postprocessing climate model data in ParGAL
  - Scalable and extensible to unstructured grids
  - Use finite element and finite volume methods
  - MOAB for mesh connectivity
  - Trilinos packages for discretization, solvers, distributed linear algebra



For more information see - https://trac.mcs.anl.gov/projects/parvis