A Patch Recovery Interpolation method

David Neckels

National Center for

Atmospheric Research

Interpolating atmospheric winds

To compute the stress on the ocean surface, we require the atmospheric wind velocity on the ocean grid.

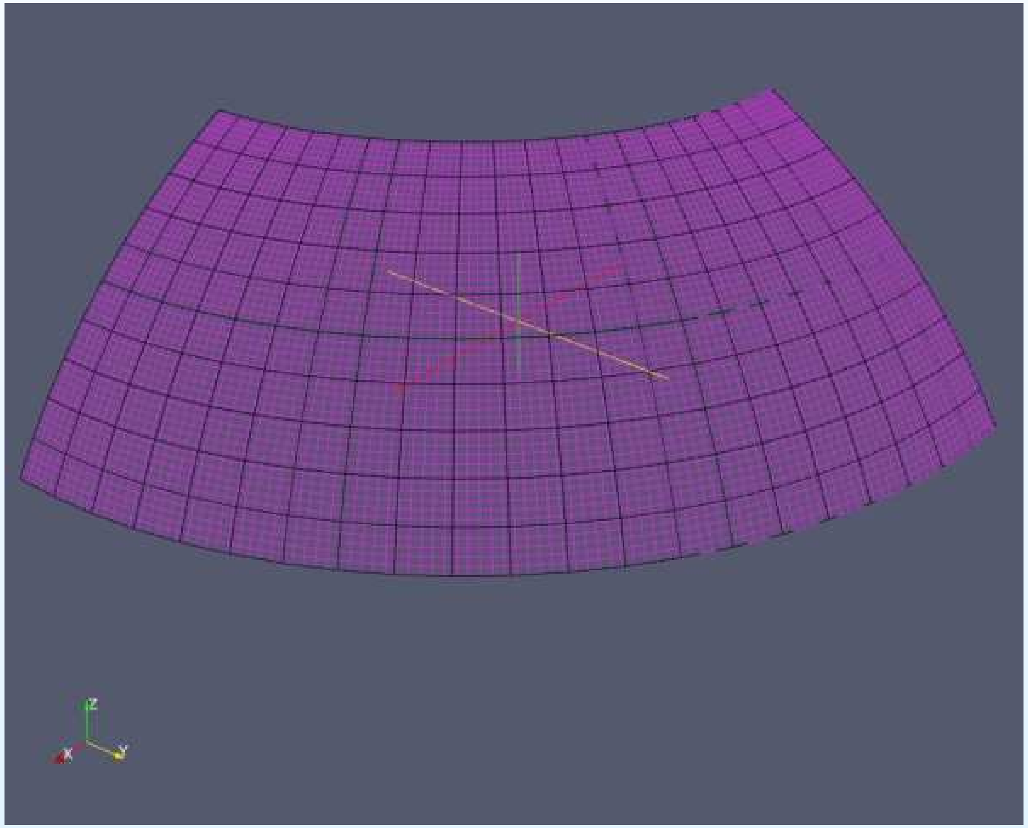
Interpolating atmospheric winds

To compute the stress on the ocean surface, we require the atmospheric wind velocity on the ocean grid.

Typically the atmospheric grid scale is much coarser than the ocean grid scale

Interpolating atmospheric winds

To compute the stress on the ocean surface, we require the atmospheric wind velocity on the ocean grid.

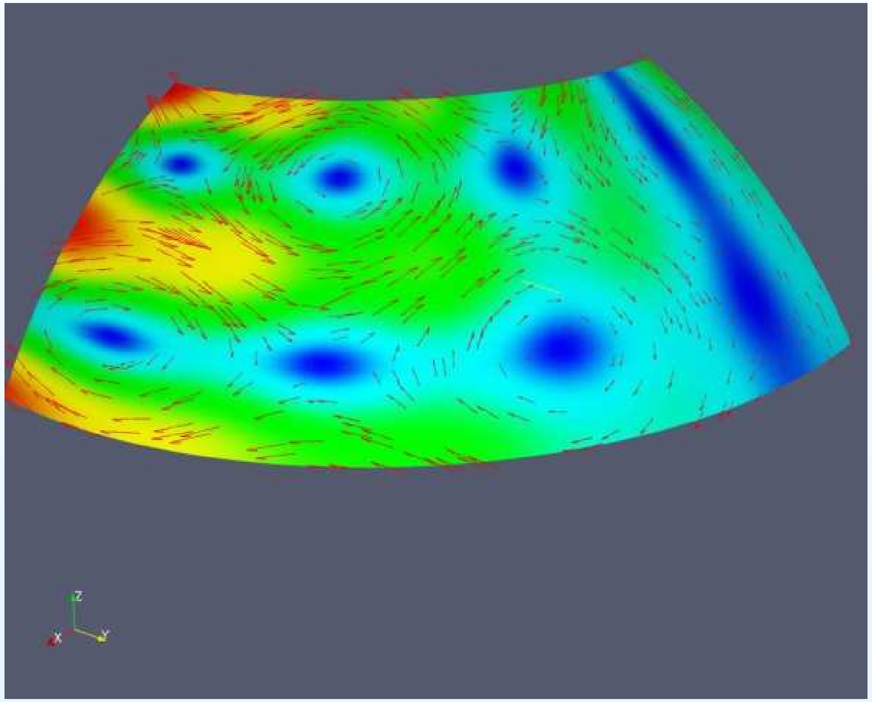


Example:Interpolating atmospheric winds, ...

Consider an analytic flow pattern

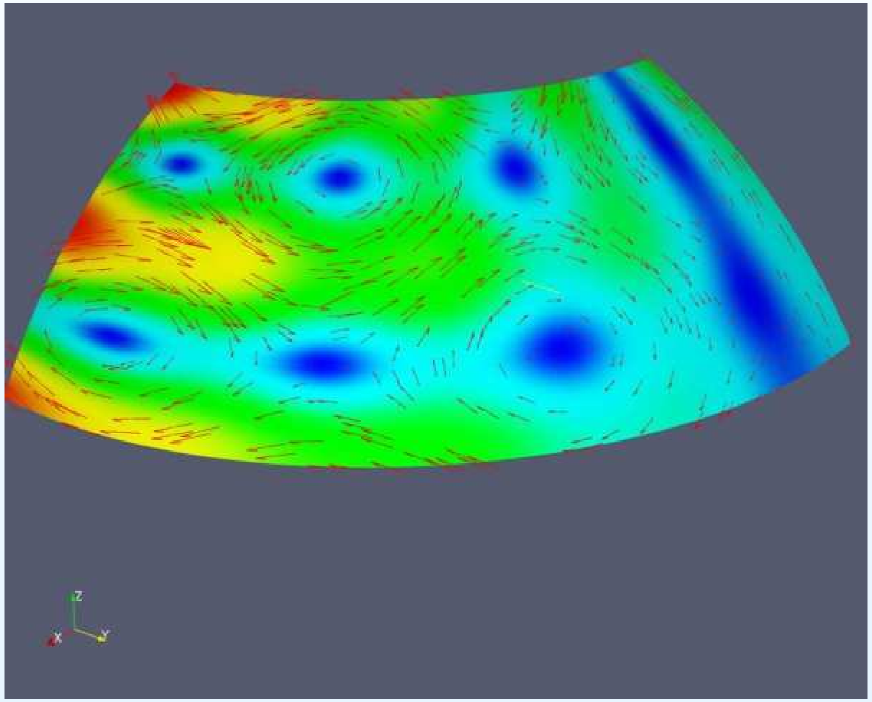
Example:Interpolating atmospheric winds, ...

Consider an analytic flow pattern



Example:Interpolating atmospheric winds, ...

Consider an analytic flow pattern



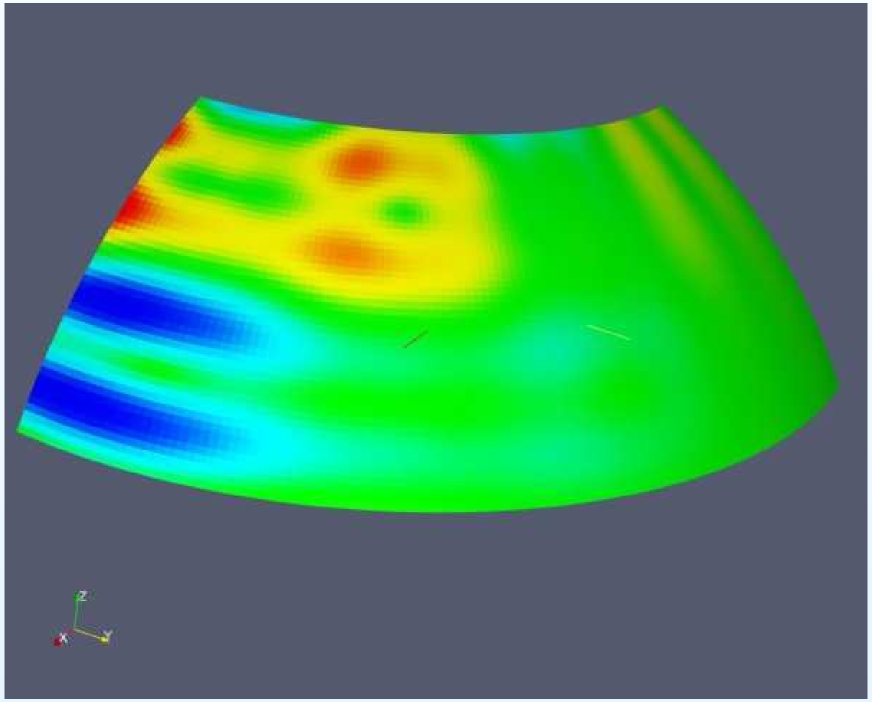
Of interest is the surface stress $\tau = C_p|U|U$, where $U = (u, v)$, especially $\nabla \times \tau$.

Example: curl of tau

Curl of the analytic flow on the ocean grid is smooth

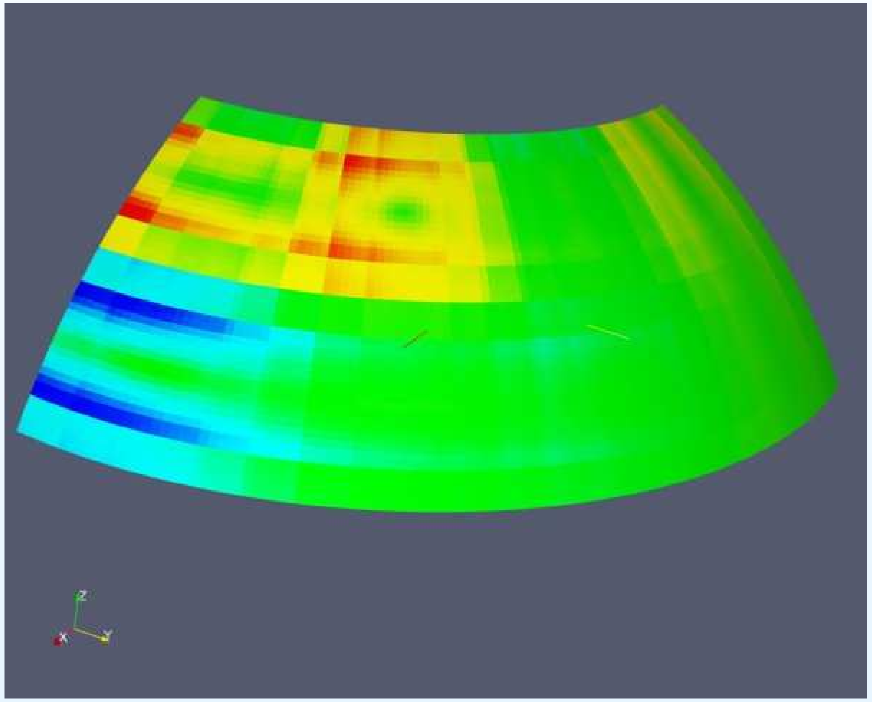
Example: curl of tau

Curl of the analytic flow on the ocean grid is smooth



Example: curl of tau

Curl of the analytic flow on the ocean grid is smooth



Curl of the standard bi-linear interpolant is not!

Computation aspects of interpolation

To compute the interpolant from two distinct grids, there are several key steps

Computation aspects of interpolation

To compute the interpolant from two distinct grids, there are several key steps

- Parallel rendezvous

Computation aspects of interpolation

To compute the interpolant from two distinct grids, there are several key steps

- Parallel rendezvous
- Search (point in box)

Computation aspects of interpolation

To compute the interpolant from two distinct grids, there are several key steps

- Parallel rendezvous
- Search (point in box)
- Interpolation method (bi-linear, conservative, patch...)

Computation aspects of interpolation

To compute the interpolant from two distinct grids, there are several key steps

- Parallel rendezvous
- Search (point in box)
- Interpolation method (bi-linear, conservative, patch...)

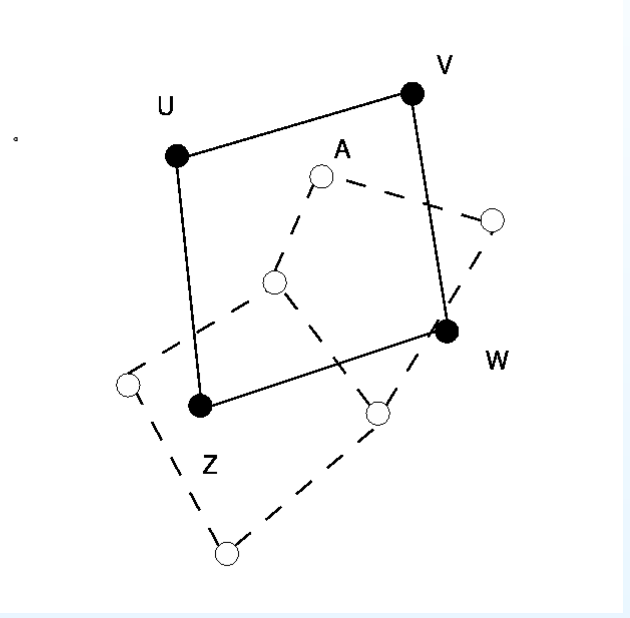
Each of these topics is its own talk. We begin with the Interpolation method.

Bi-linear interpolation

A standard interpolation scheme is the bi-linear scheme

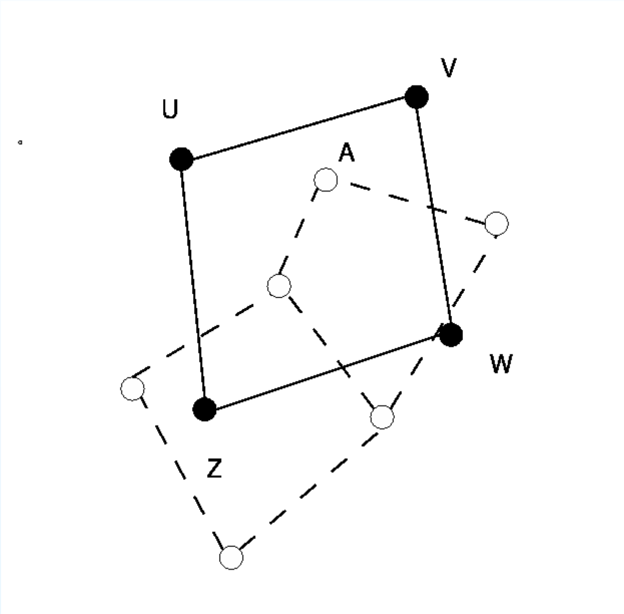
Bi-linear interpolation

A standard interpolation scheme is the bi-linear scheme



Bi-linear interpolation

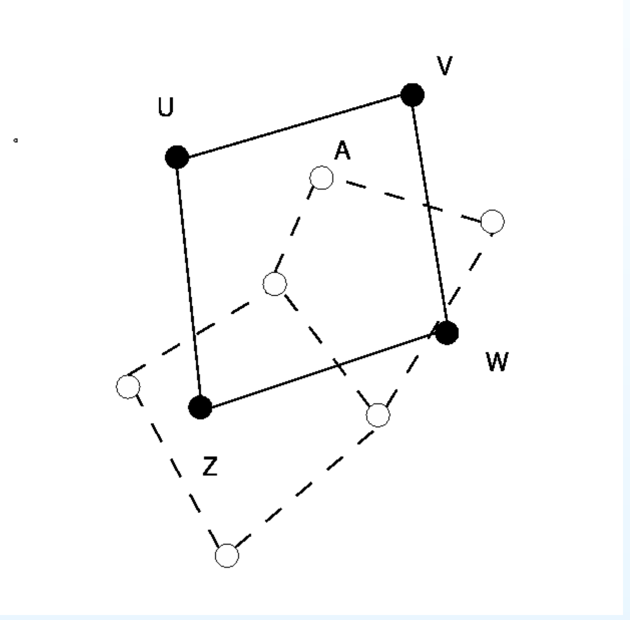
A standard interpolation scheme is the bi-linear scheme



The value at A is a weighted sum of the values at U, V, W, Z, with the bi-linear shape functions ϕ as the weights.

Bi-linear interpolation

A standard interpolation scheme is the bi-linear scheme

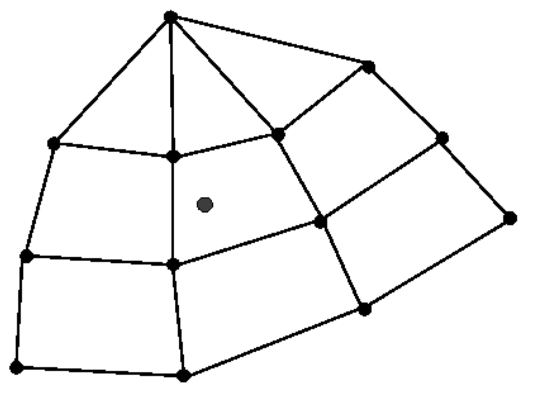


The value at A is a weighted sum of the values at U, V, W, Z, with the bi-linear shape functions ϕ as the weights.

A reasonable approximation to ∇A is $U\nabla\phi_1 + \cdots + Z\nabla\phi_Z$.

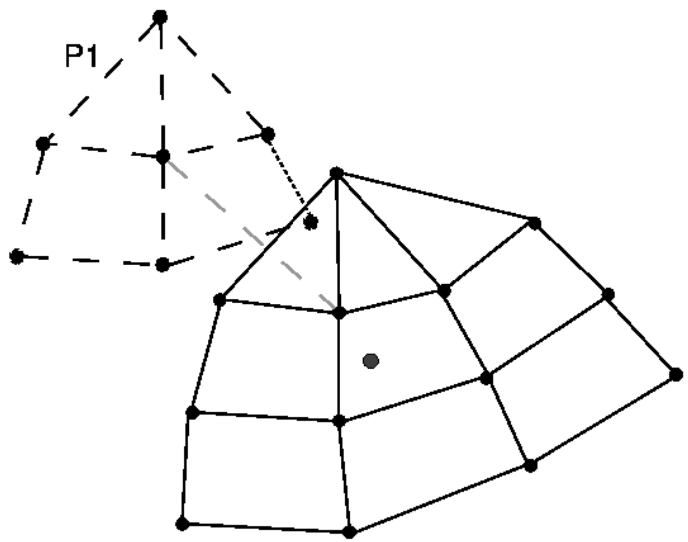
Patch based methods

We form the interpolant at • using polynomials based on the node patches of the encompassing cell:



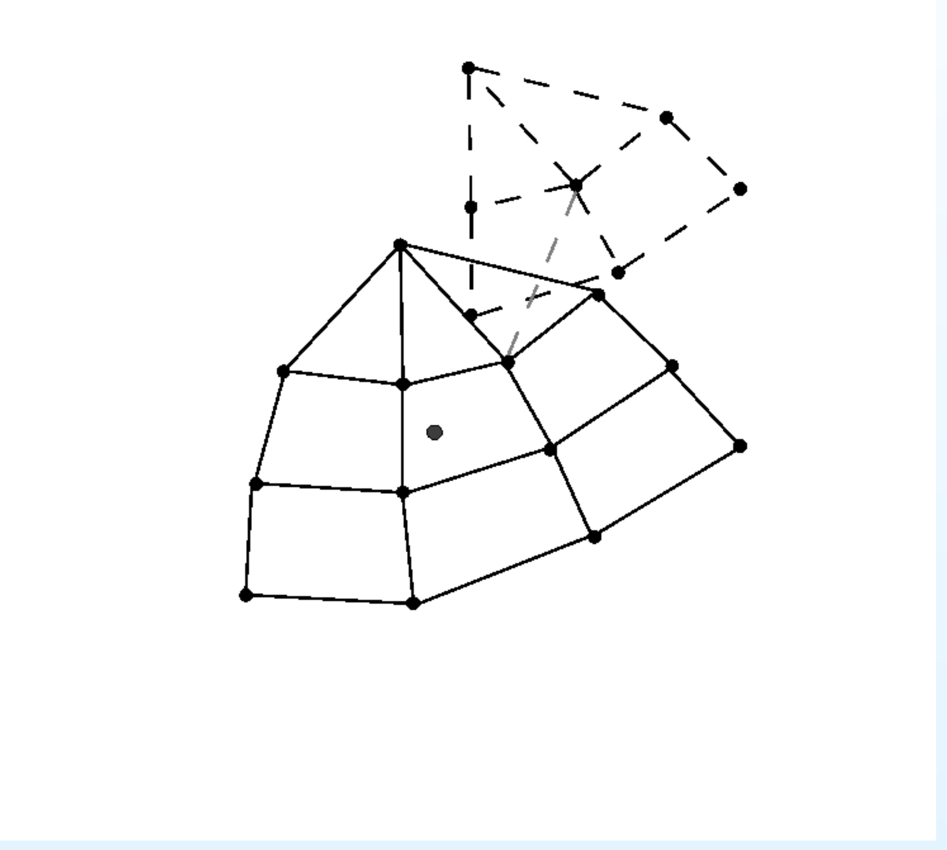
Patch based methods

We form the interpolant at • using polynomials based on the node patches of the encompassing cell:



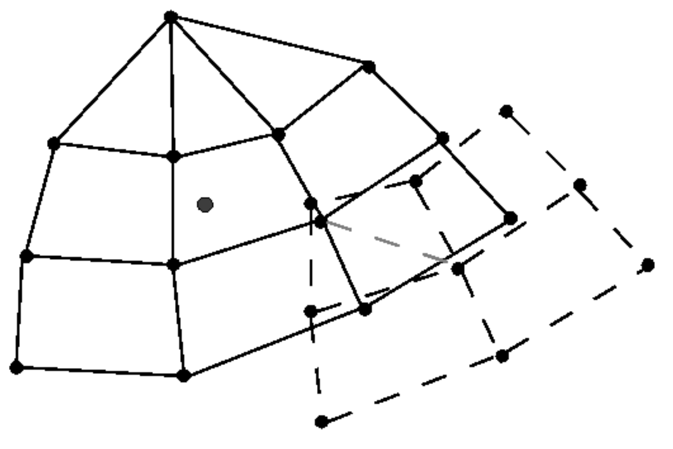
Patch based methods

We form the interpolant at ● using polynomials based on the node patches of the encompassing cell:



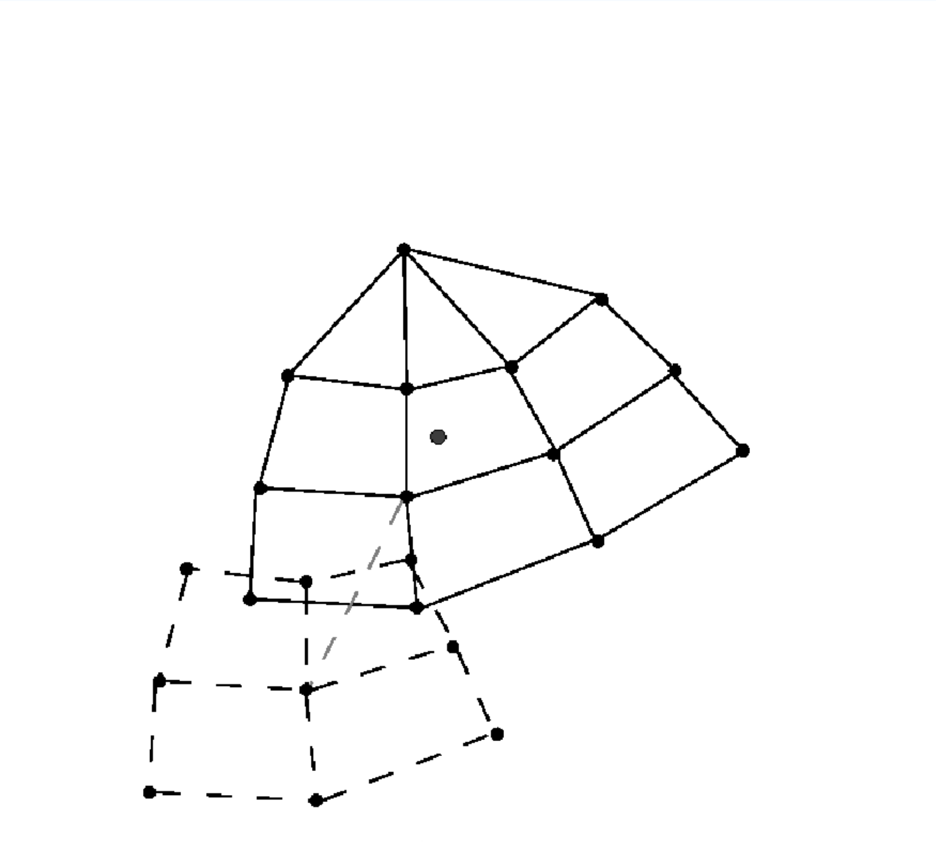
Patch based methods

We form the interpolant at • using polynomials based on the node patches of the encompassing cell:



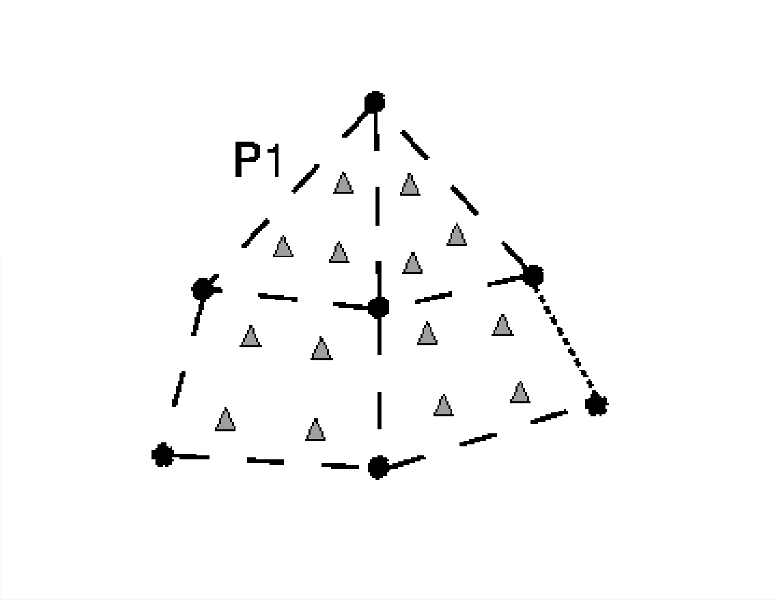
Patch based methods

We form the interpolant at • using polynomials based on the node patches of the encompassing cell:



Patch based methods,...

On each patch we sample the source function at a set of sample points (usually quadrature points) $\triangle$, using local bi-linear interpolation if necessary.



Call these samples s_i at (local 2D) coordinates p_i.

Local polynomial approximation

We fit a tensor product polynomial through these values, solving for the polynomial coefficients c

$$\min_c \sum_i (Q(c, p_i) - s_i)^2$$

ESMF

Local polynomial approximation

We fit a tensor product polynomial through these values, solving for the polynomial coefficients c

$$\min_c \sum_i (Q(c, p_i) - s_i)^2$$

Which yields the least squares system $A^\top Ac = A^\top s$ and $Q(p) = b(p)^T(A^\top A)^{-1}s$ where b is the vector of the polynomial basis functions evaluated at the sample points.

Local polynomial approximation

We fit a tensor product polynomial through these values, solving for the polynomial coefficients c

$$\min_c \sum_i (Q(c, p_i) - s_i)^2$$

Which yields the least squares system $A^\top Ac = A^\top s$ and $Q(p) = b(p)^T(A^\top A)^{-1}s$ where b is the vector of the polynomial basis functions evaluated at the sample points.

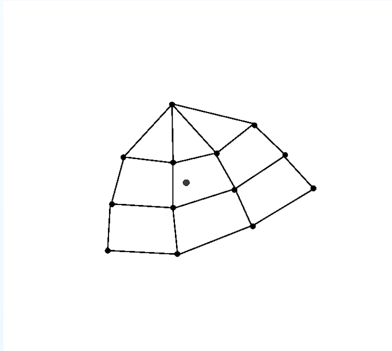
On a manifold, the local coordinates p_i may either be full 3D coordinates, or the coefficients of the co-space of a reasonable normal.
This avoids pole type singularities in the patch algorithm (i.e. don't use lat/lon).

Blending the patches

We use any partition of unity on the cell to blend the patches for a value $F(x) = \sum_j \psi_j(x) Q(x)$, for instance the bi-linear basis.

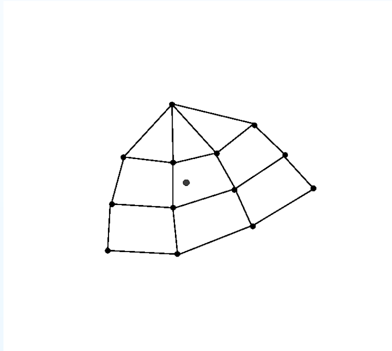
Blending the patches

We use any partition of unity on the cell to blend the patches for a value $F(x) = \sum_j \psi_j(x) Q(x)$, for instance the bi-linear basis.



Blending the patches

We use any partition of unity on the cell to blend the patches for a value $F(x) = \sum_j \psi_j(x) Q(x)$, for instance the bi-linear basis.



Explicitly, accounting for the local coordinate system $p = L(x)$ and the bi-linear interpolation to sample locations $s = \Phi f$, the interpolant is a linear function of the coefficients f on this enlarged stencil,

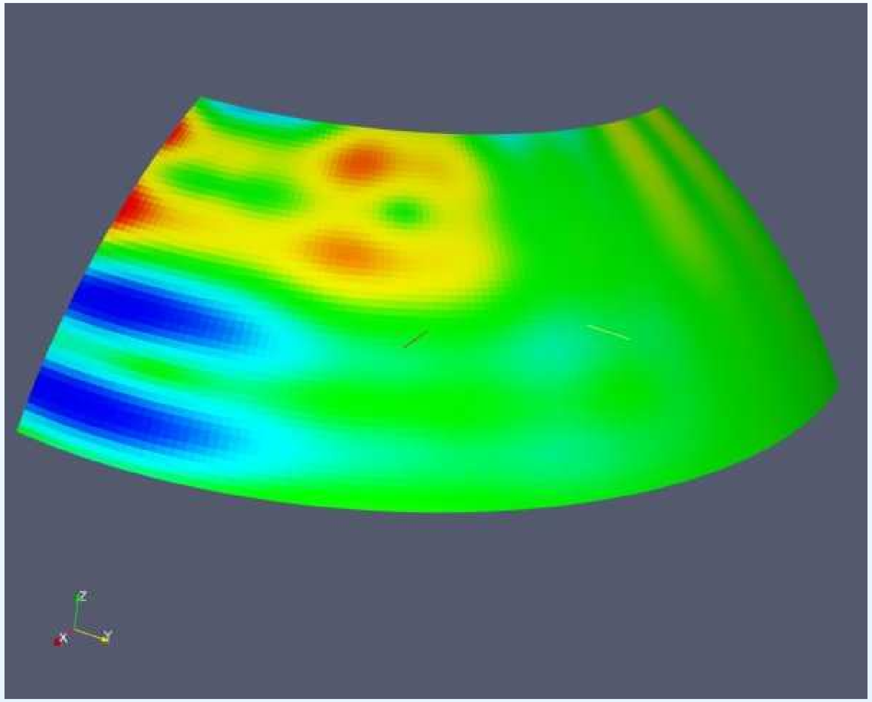
$$F(x) = \sum_j \left[\psi(x) \big(b \circ L(x) \big)^\top (A^\top A)^{-1} \Phi \right]_j f$$

Back to curl of tau

Curl of the analytic flow on the ocean grid is smooth

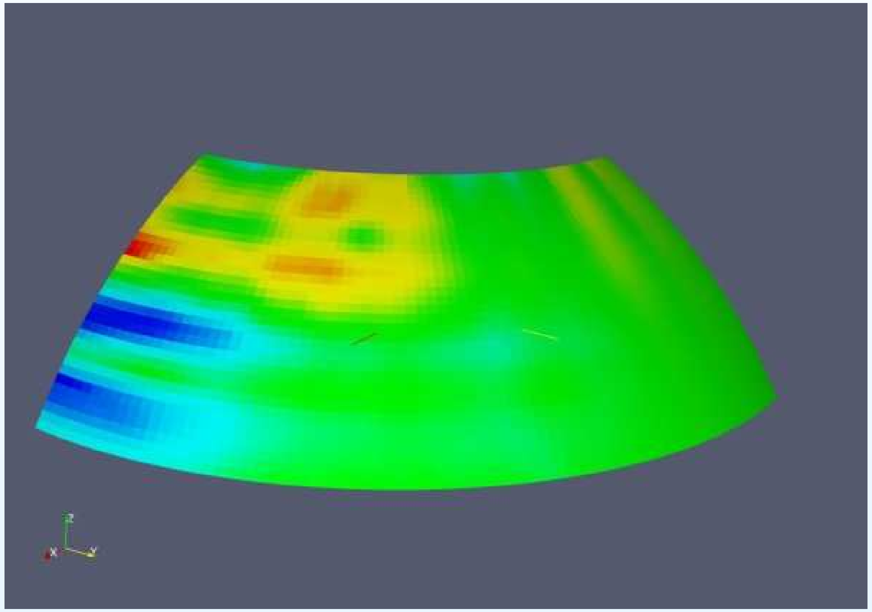
Back to curl of tau

Curl of the analytic flow on the ocean grid is smooth



Back to curl of tau

Curl of the analytic flow on the ocean grid is smooth



The patch recovery curl is far more reasonable compared to the bi-linear!

Some results from interpolation theory

Interpolating a function $f(x)$ into the space of continuous piecewise polynomial functions of order p on a discretization $\mathcal{T}_h$, with cell diameters h, using exact values of f at the nodes yields

Some results from interpolation theory

Interpolating a function $f(x)$ into the space of continuous piecewise polynomial functions of order p on a discretization $\mathcal{T}_h$, with cell diameters h, using exact values of f at the nodes yields

$$||D^m(f - \mathcal{I}f)||_{L^2} \leq Ch^{(p+1)-m}||D^{p+1}f||_{L^2}$$

Some results from interpolation theory

Interpolating a function $f(x)$ into the space of continuous piecewise polynomial functions of order p on a discretization $\mathcal{T}_h$, with cell diameters h, using exact values of f at the nodes yields

$$||D^m(f - \mathcal{I}f)||_{L^2} \leq Ch^{(p+1)-m}||D^{p+1}f||_{L^2}$$

i.e. for bi-linear interpolation

$$||f - \mathcal{I}f||_{L^2} \leq Ch^2||D^2f||_{L^2}$$

Some results from interpolation theory

Interpolating a function $f(x)$ into the space of continuous piecewise polynomial functions of order p on a discretization $\mathcal{T}_h$, with cell diameters h, using exact values of f at the nodes yields

$$||D^m(f - \mathcal{I}f)||_{L^2} \leq Ch^{(p+1)-m}||D^{p+1}f||_{L^2}$$

i.e. for bi-linear interpolation

$$||f - \mathcal{I}f||_{L^2} \leq Ch^2||D^2f||_{L^2}$$

and

$$||\nabla(f - \mathcal{I}f)||_{L^2} \leq Ch||D^2f||_{L^2}$$

Some results from interpolation theory

Interpolating a function $f(x)$ into the space of continuous piecewise polynomial functions of order p on a discretization $\mathcal{T}_h$, with cell diameters h, using exact values of f at the nodes yields

$$||D^m(f - \mathcal{I}f)||_{L^2} \leq Ch^{(p+1)-m}||D^{p+1}f||_{L^2}$$

i.e. for bi-linear interpolation

$$||f - \mathcal{I}f||_{L^2} \leq Ch^2||D^2f||_{L^2}$$

and

$$||\nabla(f - \mathcal{I}f)||_{L^2} \leq Ch||D^2f||_{L^2}$$

Smoothness is required, at least of weak derivatives $||D^2f||_{L^2}$.

An experiment

We perform a convergence study for the analytic function

$$f(x, y) = (1 - xy) \sin 3\pi x \cos 2\pi y)$$

on the unit square using patch and bi-linear interpolation.

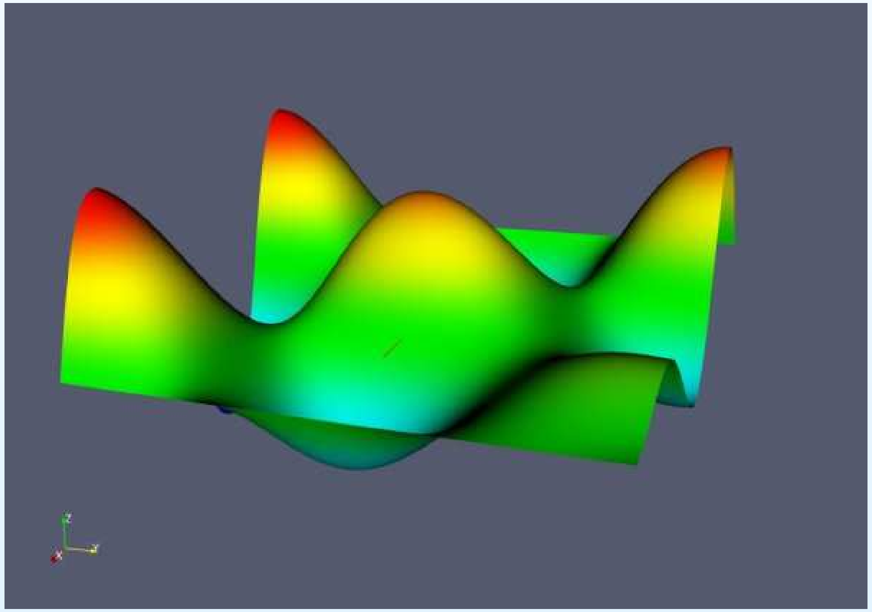
An experiment

We perform a convergence study for the analytic function

$$f(x, y) = (1 - xy)\sin 3\pi x \cos 2\pi y)$$

on the unit square using patch and bi-linear interpolation.

Exact



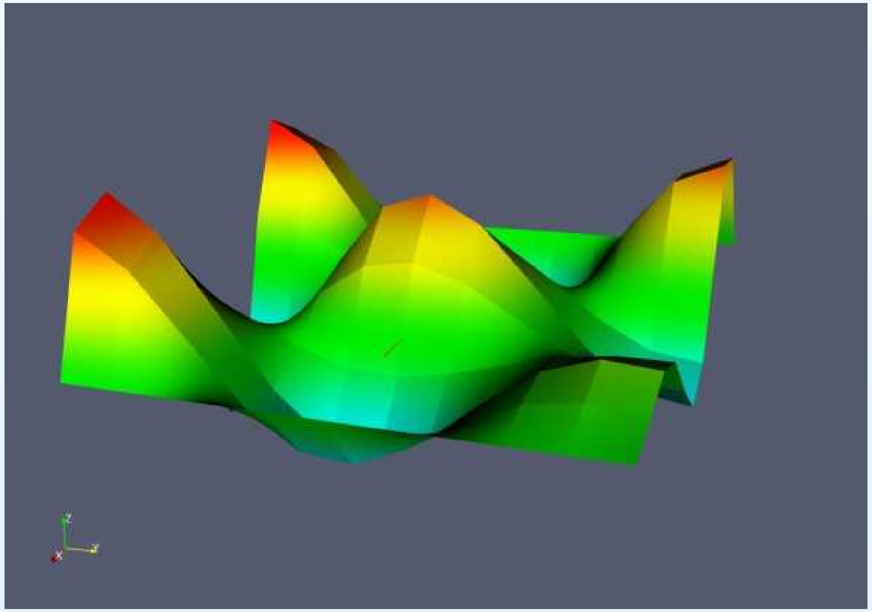
An experiment

We perform a convergence study for the analytic function

$$f(x,y) = (1 - xy)\sin 3\pi x \cos 2\pi y)$$

on the unit square using patch and bi-linear interpolation.

Bilinear



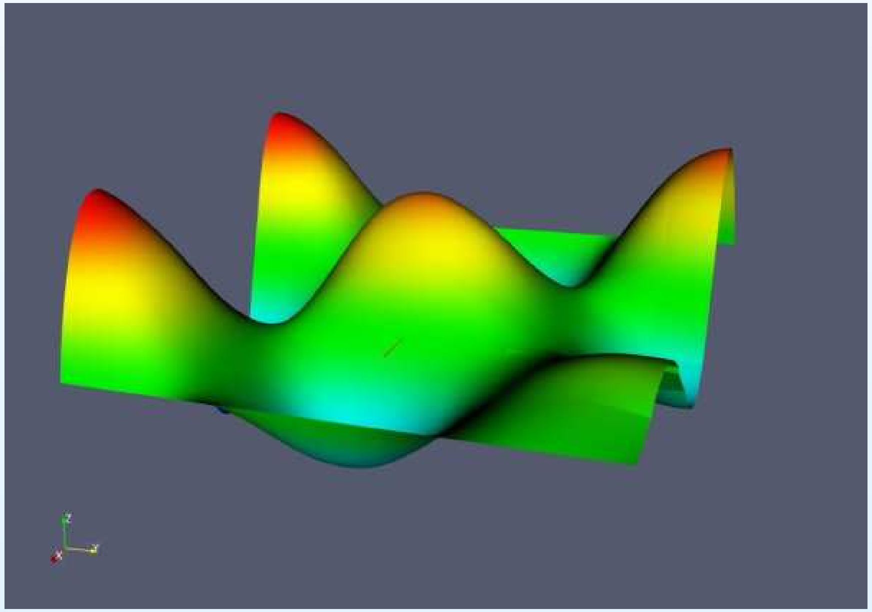
An experiment

We perform a convergence study for the analytic function

$$f(x, y) = (1 - xy) \sin 3\pi x \cos 2\pi y)$$

on the unit square using patch and bi-linear interpolation.

Patch

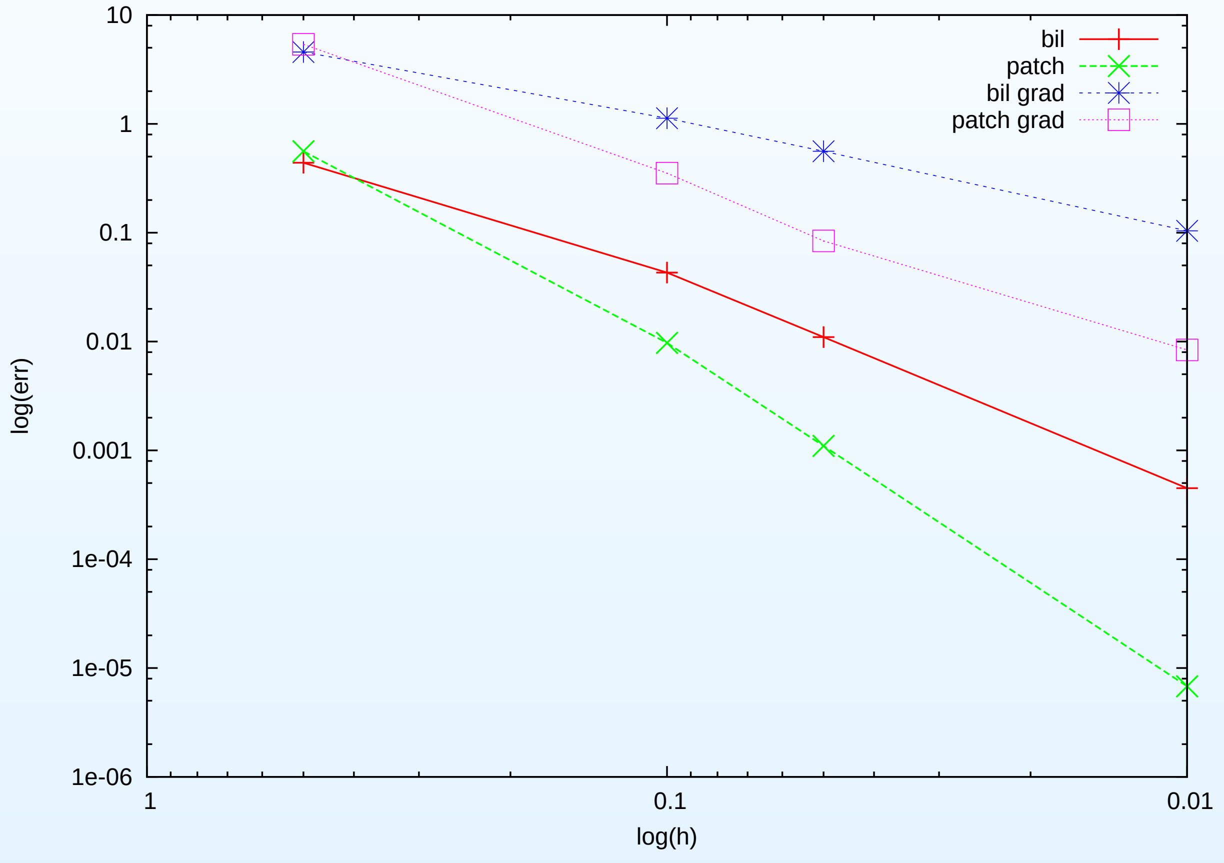


Results

We compute the L^2 error on a super fine grid.

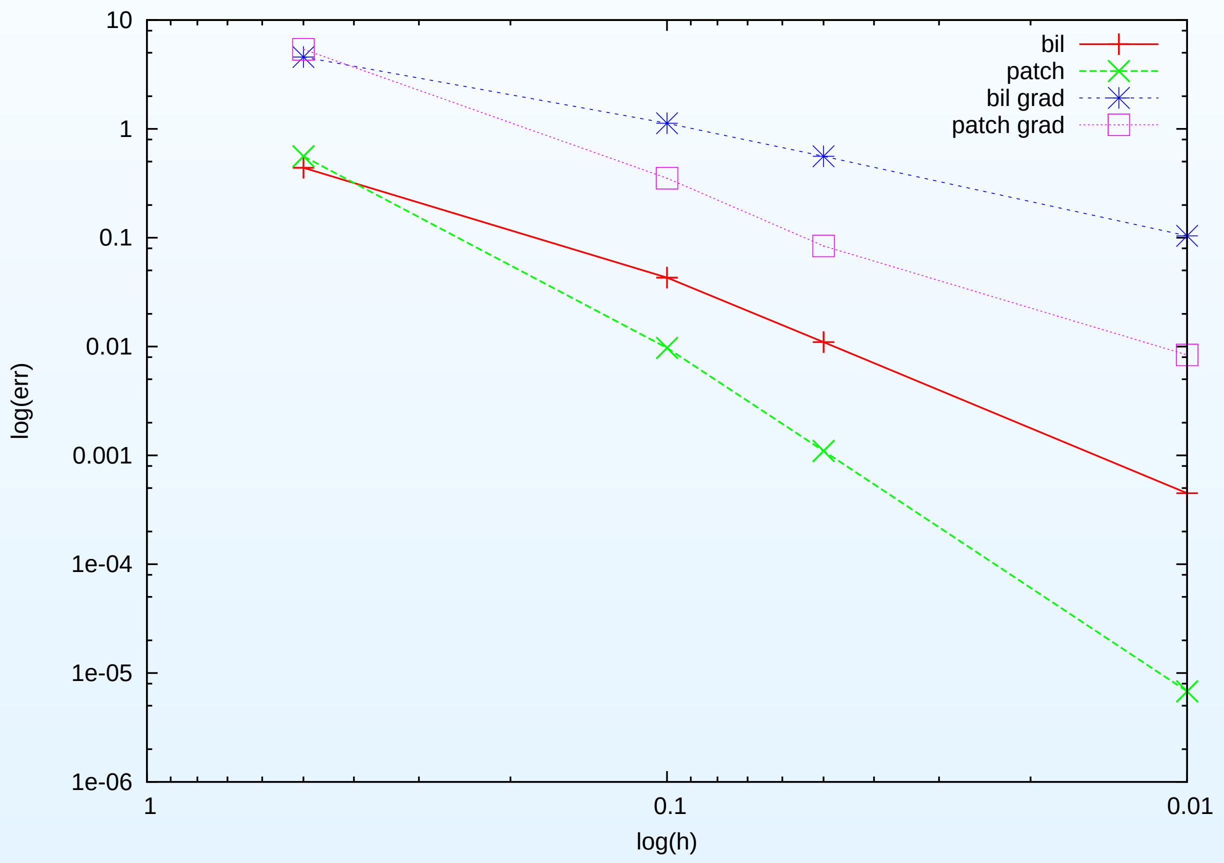
Results

We compute the L^2 error on a super fine grid.



Results

We compute the L^2 error on a super fine grid.



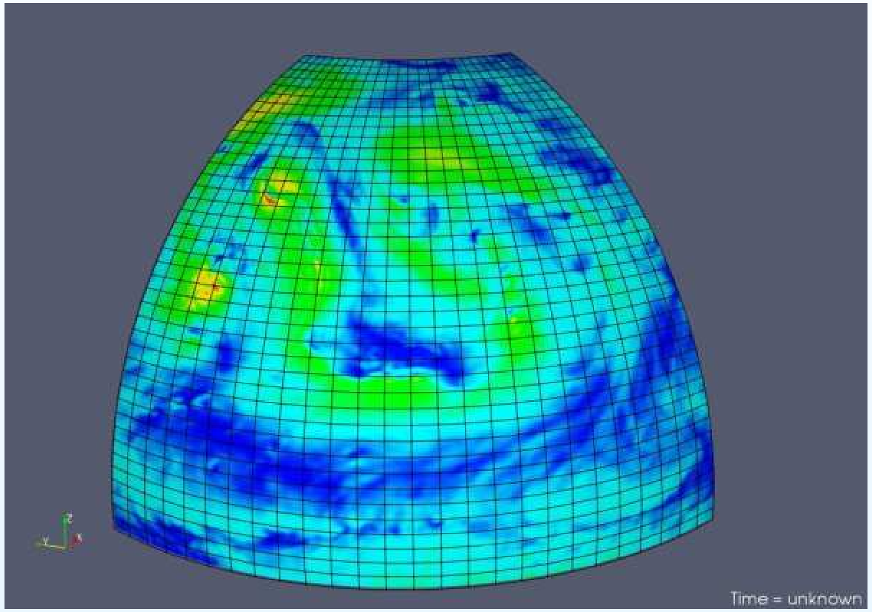
Rates are $P = 3.14, B = 1.96, \nabla P = 2.01, \nabla B = 1.01$.

A real wind field

We compare interpolation methods on realistic wind data

A real wind field

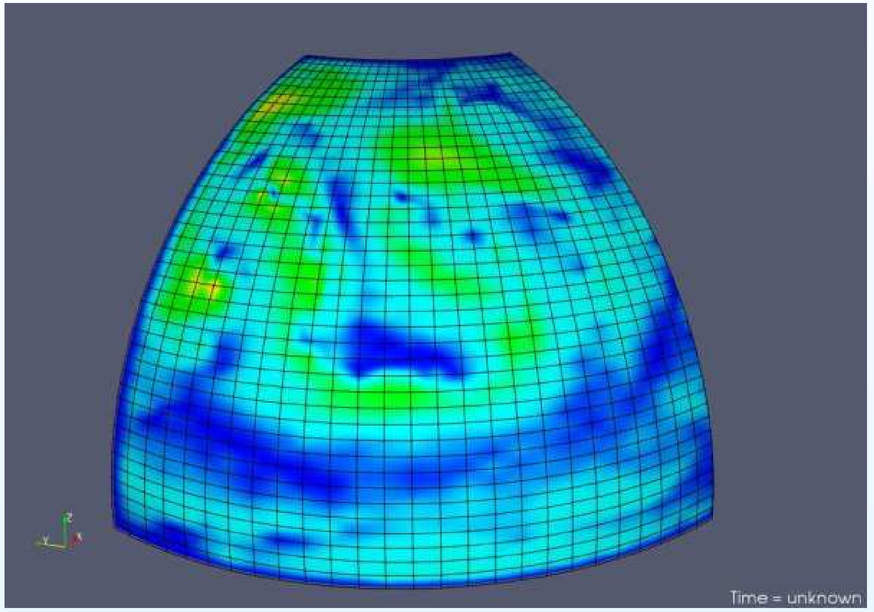
We compare interpolation methods on realistic wind data



The exact wind field ($|U|$)

A real wind field

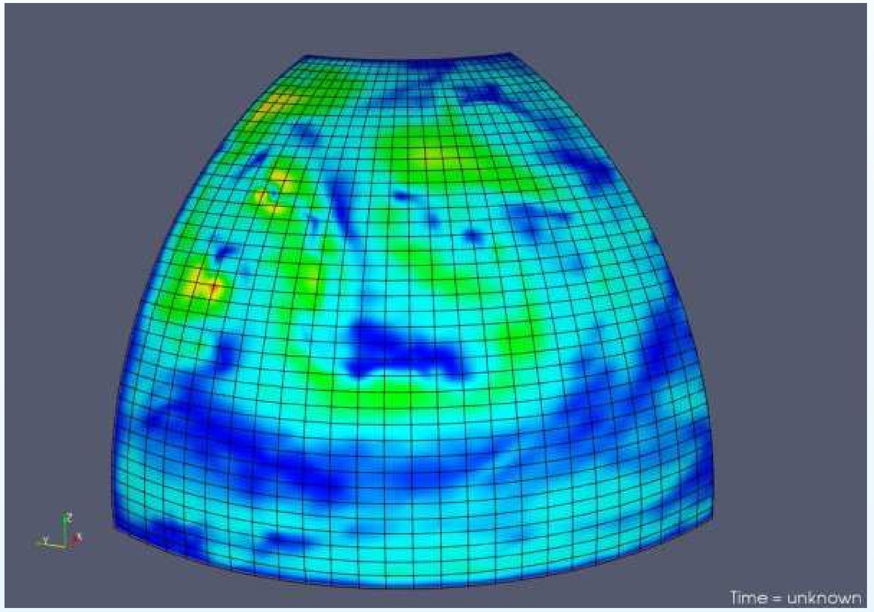
We compare interpolation methods on realistic wind data



The bi-linear interpolant

A real wind field

We compare interpolation methods on realistic wind data



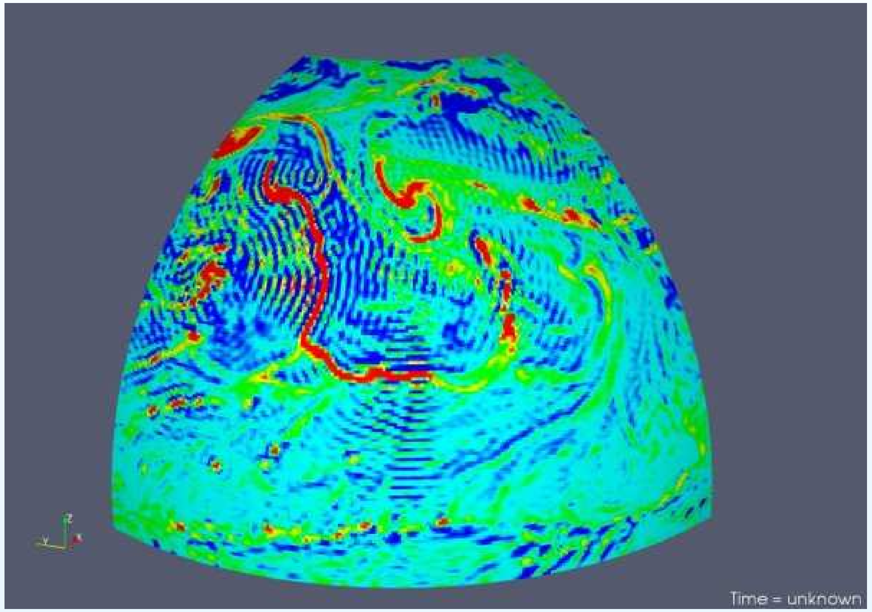
The patch interpolant

Curl of the real wind field

We compare interpolation methods on realistic wind data

Curl of the real wind field

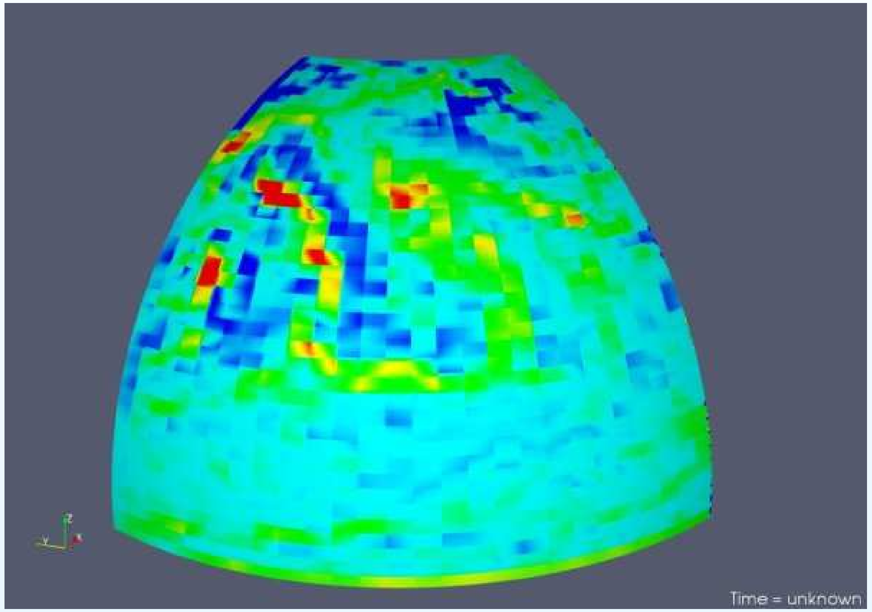
We compare interpolation methods on realistic wind data



The exact wind field ($\nabla \times U$)

Curl of the real wind field

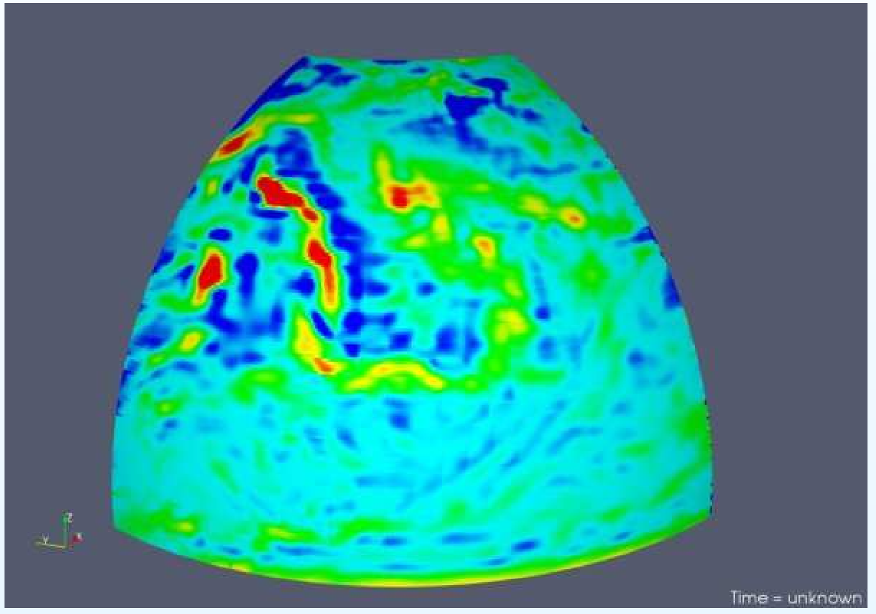
We compare interpolation methods on realistic wind data



The bi-linear interpolant

Curl of the real wind field

We compare interpolation methods on realistic wind data



The patch interpolant

The End