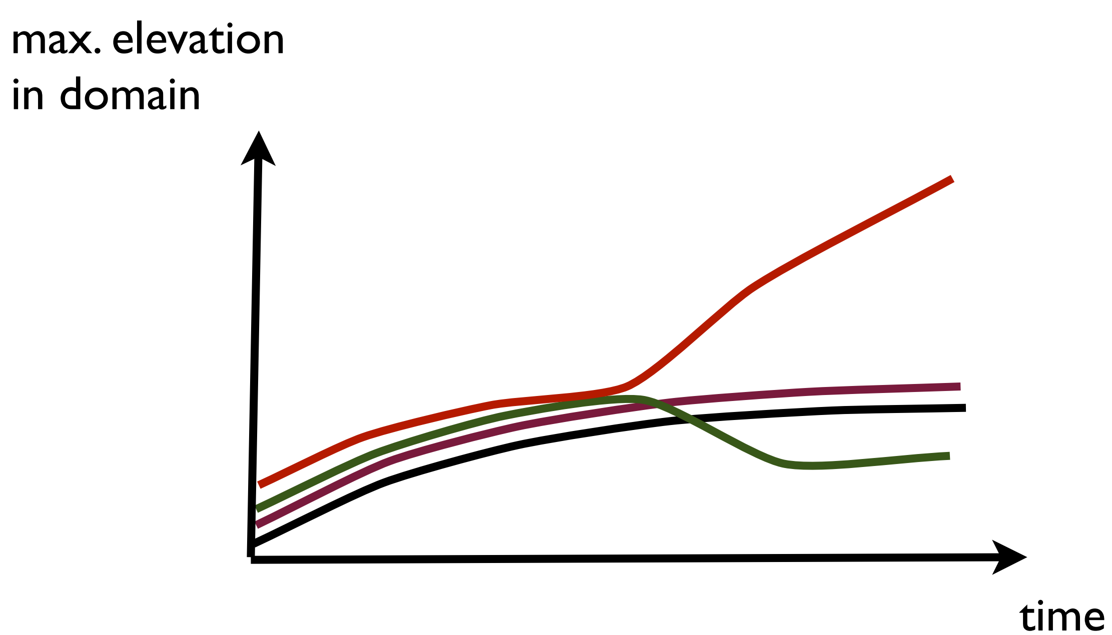
Surge sensitivity to physical parameters through forward and adjoint modelling with MITgcm

Chris Wilson and Kevin Horsburgh
National Oceanography Centre, Liverpool (Formerly POL)

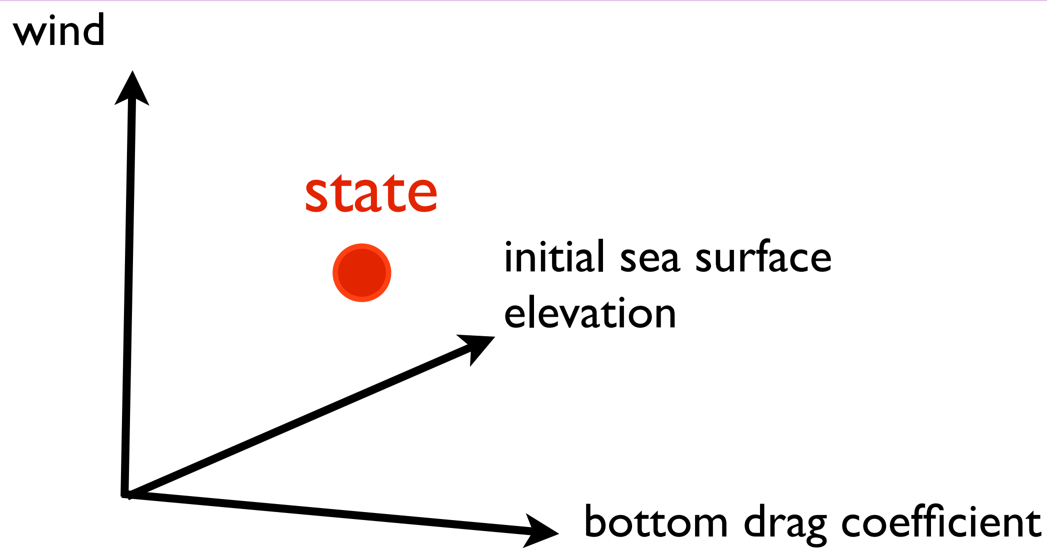
- What controls the maximum elevation within a given region during a given time period?

- Boundary conditions (e.g. wind forcing)
- Physical parameters (e.g. bottom drag coefficient)
- Initial conditions (e.g. tidal phase errors)



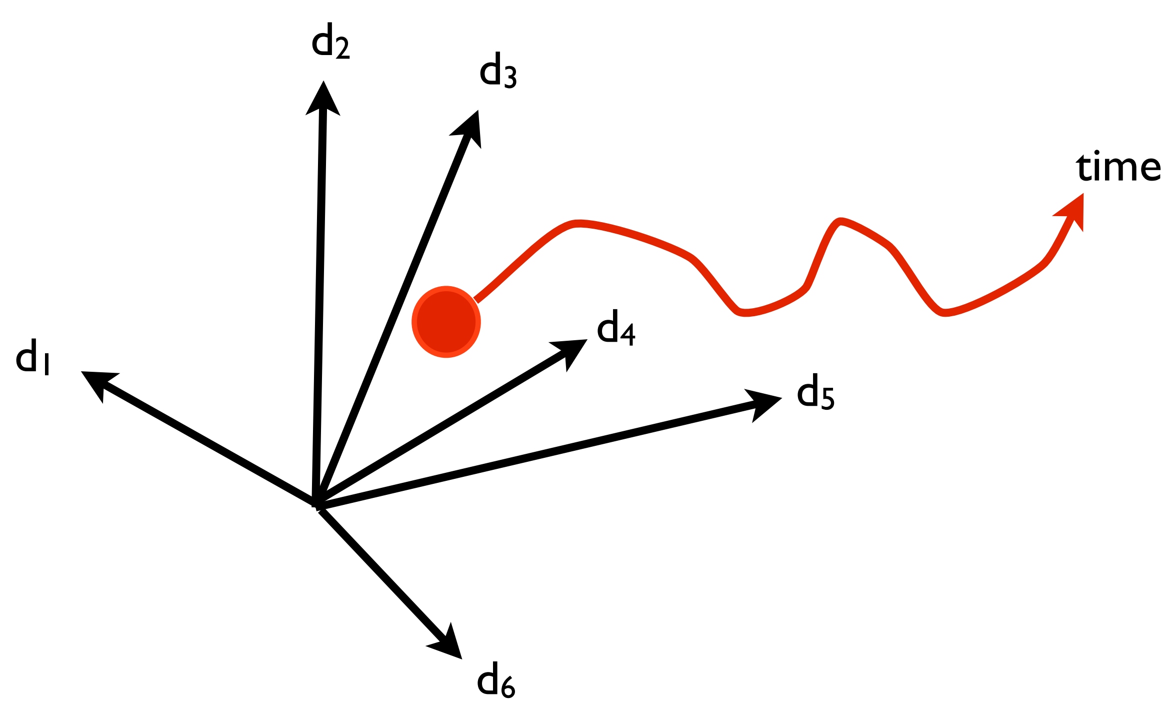
Q1. What are the **relative** roles of boundary conditions, physical parameterizations and initial conditions on the predicted tide+surge?

Q2. How does the sensitivity of maximum tide+surge to wind depend on the ocean initial condition, etc?



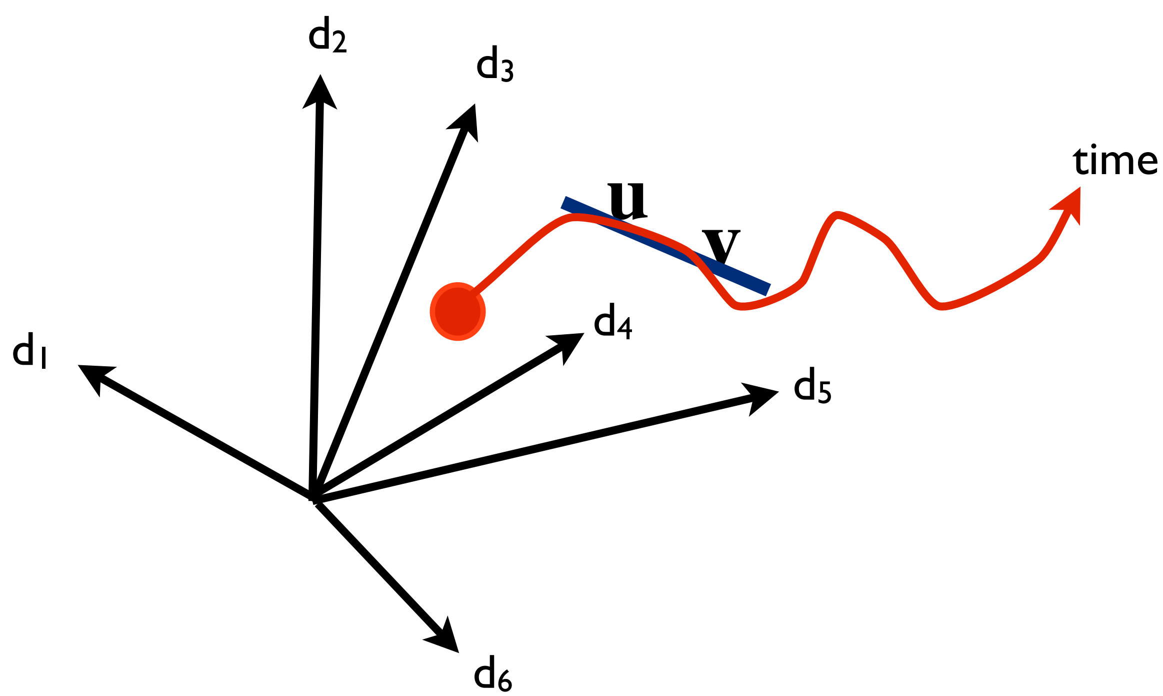
huge ensemble!

model state consists of millions of degrees of freedom



trajectory describes time-evolution of model state

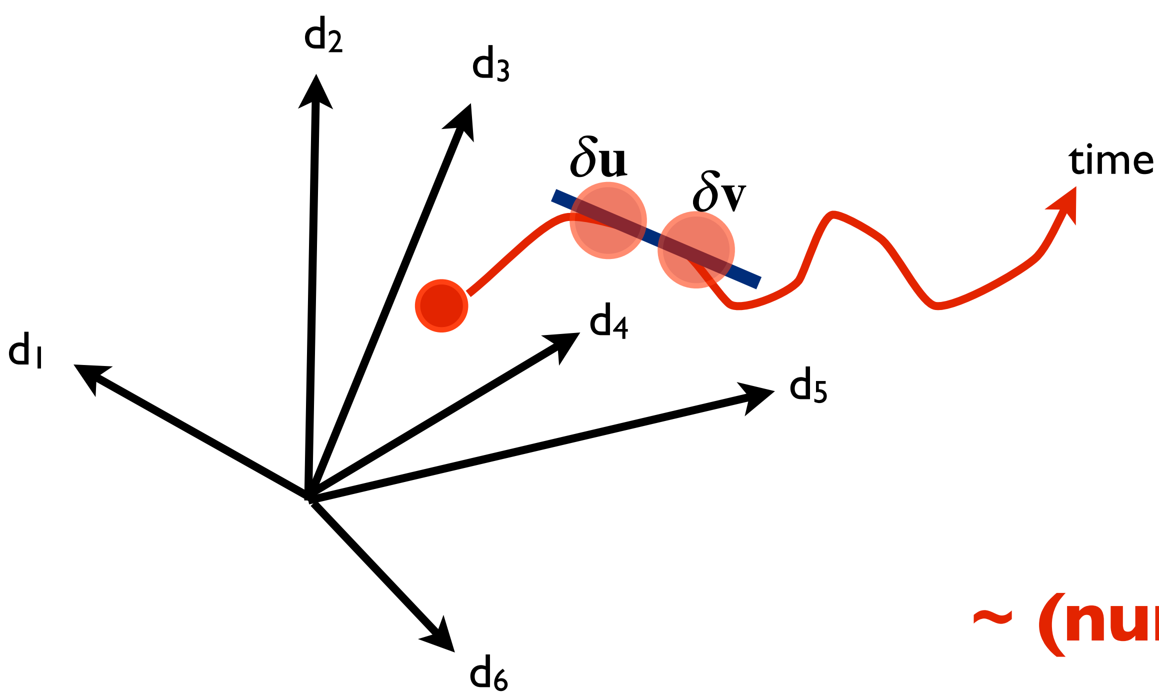
Model state space and sensitivity



$$\mathbf{u} \mapsto \mathbf{v} = \mathcal{M}(\mathbf{u})$$

perturbations to $\mathbf{v}$
can be approximated by the
Tangent Linear Model,
the Taylor Series expansion of $\mathcal{M}$

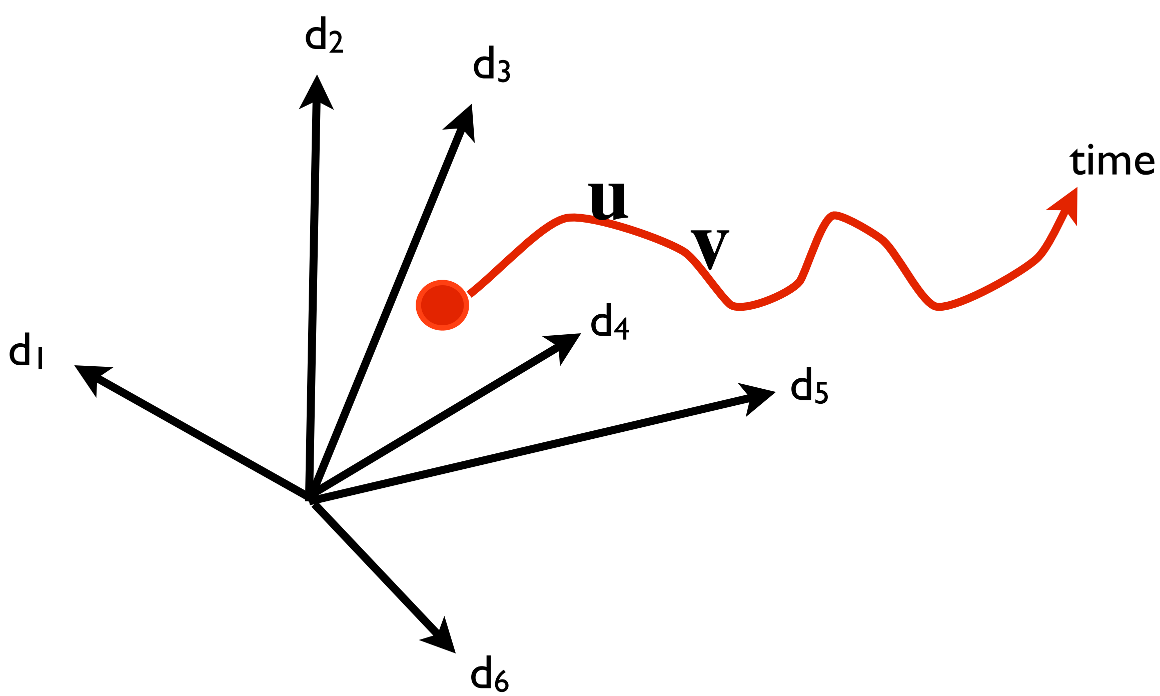
Forward sensitivity



BUT every component of $\delta\mathbf{v}$
requires an individual row of
the Jacobian matrix of $\mathcal{M}$
multiplied by $\delta\mathbf{u}$

~ (number of degrees of freedom)^2

Adjoint sensitivity



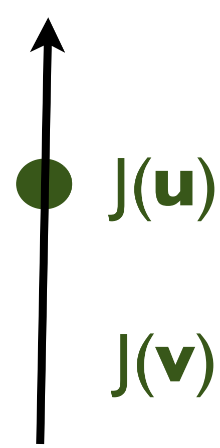
simplify the problem using an adjoint

reduce the number of degrees of freedom in the sensitivity by defining a scalar cost function (tailored to the question in hand)

calculate sensitivity of cost function to perturbations of model state

$$\frac{\partial J}{\partial \mathbf{u}}$$

e.g. J=maximum sea surface elevation in domain



sensitivities calculated within the model via **'automatic differentiation'**, where the effect of each line of code on the model operator, $\mathcal{M}$, is considered

~ (number of degrees of freedom)^1

Example from Losch and Heimbach, 2007

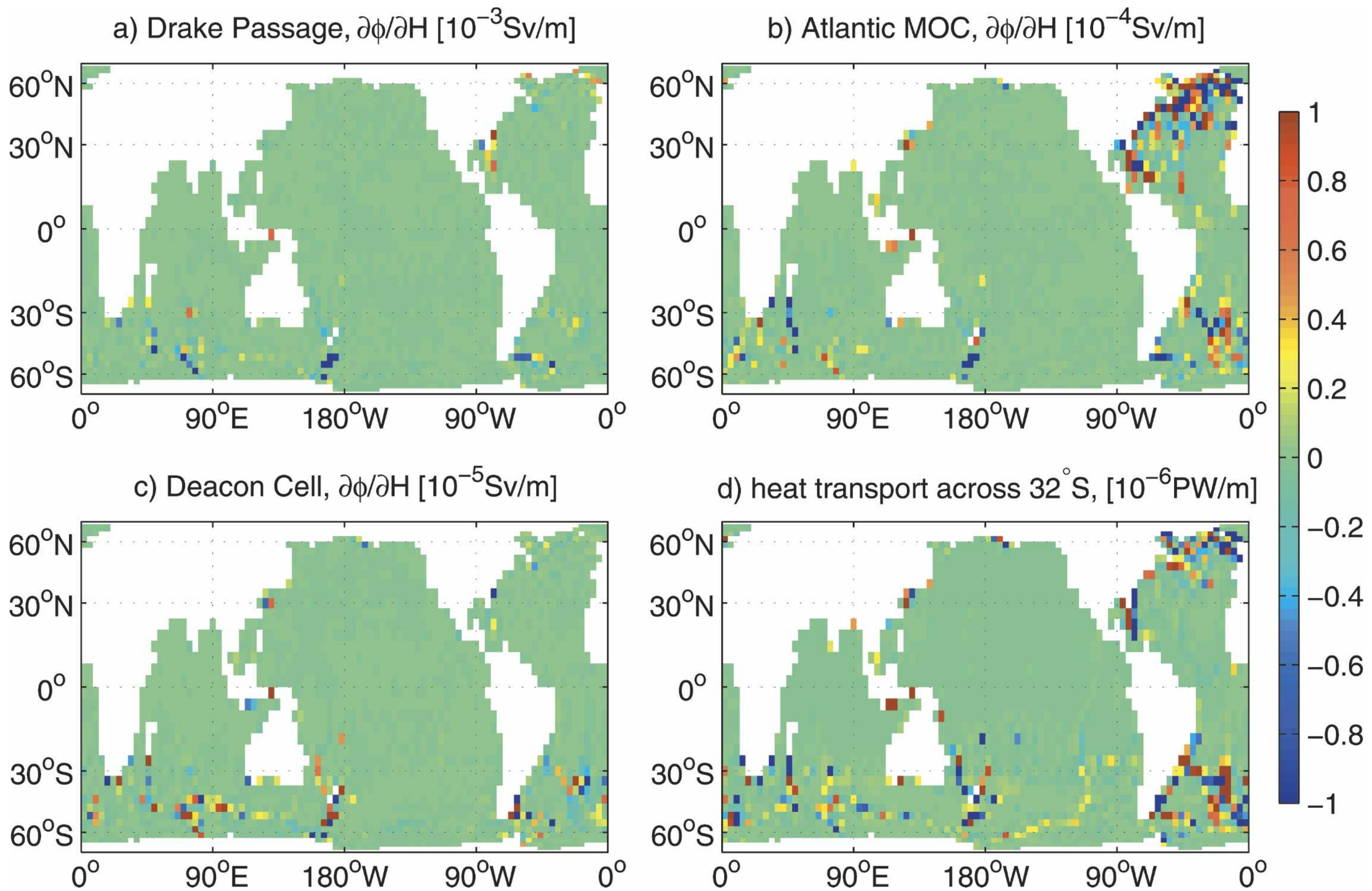


FIG. 2. Adjoint sensitivity with respect to the topography of (a) the volume transport through Drake Passage, (b) the Atlantic MOC, (c) the Deacon cell, and (d) the heat transport across 32°S. In all cases, the integration period is 100 yr.

MITgcm designed be run in both forward and adjoint modes

- Develop shelf seas storm surge forward model
- Run perturbed parameter ensemble experiments
- Run adjoint model and compare surge sensitivity with ensembles

1. Develop shelf seas storm surge forward model

MITgcm tide model established (J. Xing; S. Legg), but for 2D x-z

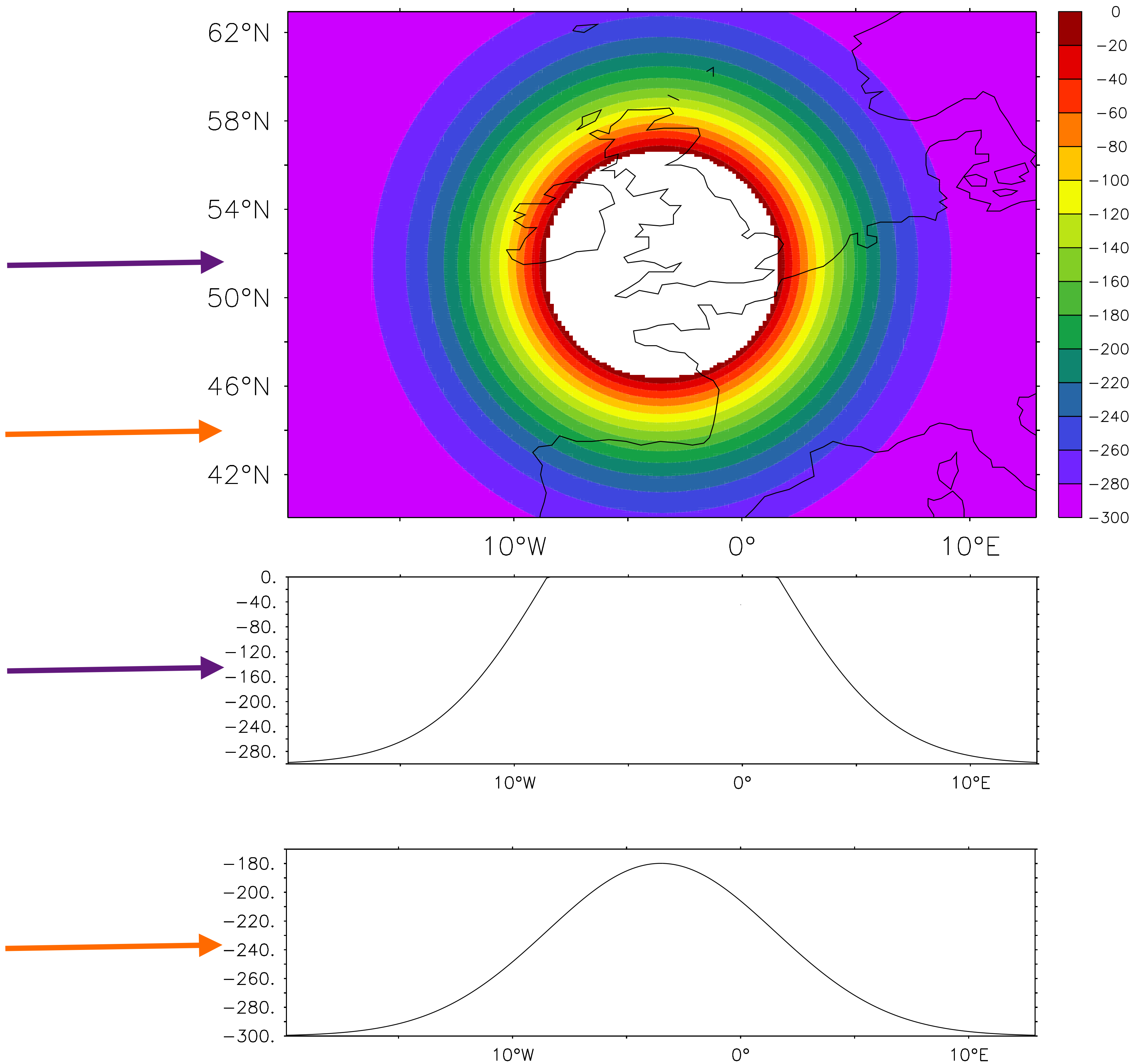
Starting from example experiments, closest variants ‘natl_box’, ‘barotropic gyre’

•Modify domain to match that of POL CS3X model

•1/6 X 1/9 degree lon X lat
•Idealised (Gaussian island) and realistic topography (from ETOPO5)
•6 level PE, 1 level Shallow Water Equations

•Prescribe tidal boundary conditions for velocity

Idealised Gaussian topographic elevation (m)



dz:
10 m
25 m
25 m
70 m
70 m
100 m

Unstratified
T=S=constant

Boundary conditions and tidal forcing

Can't use tide from realistic models, tidal atlases for idealised domain

Excite Kelvin wave around island topography

Plane wave not in geostrophic balance

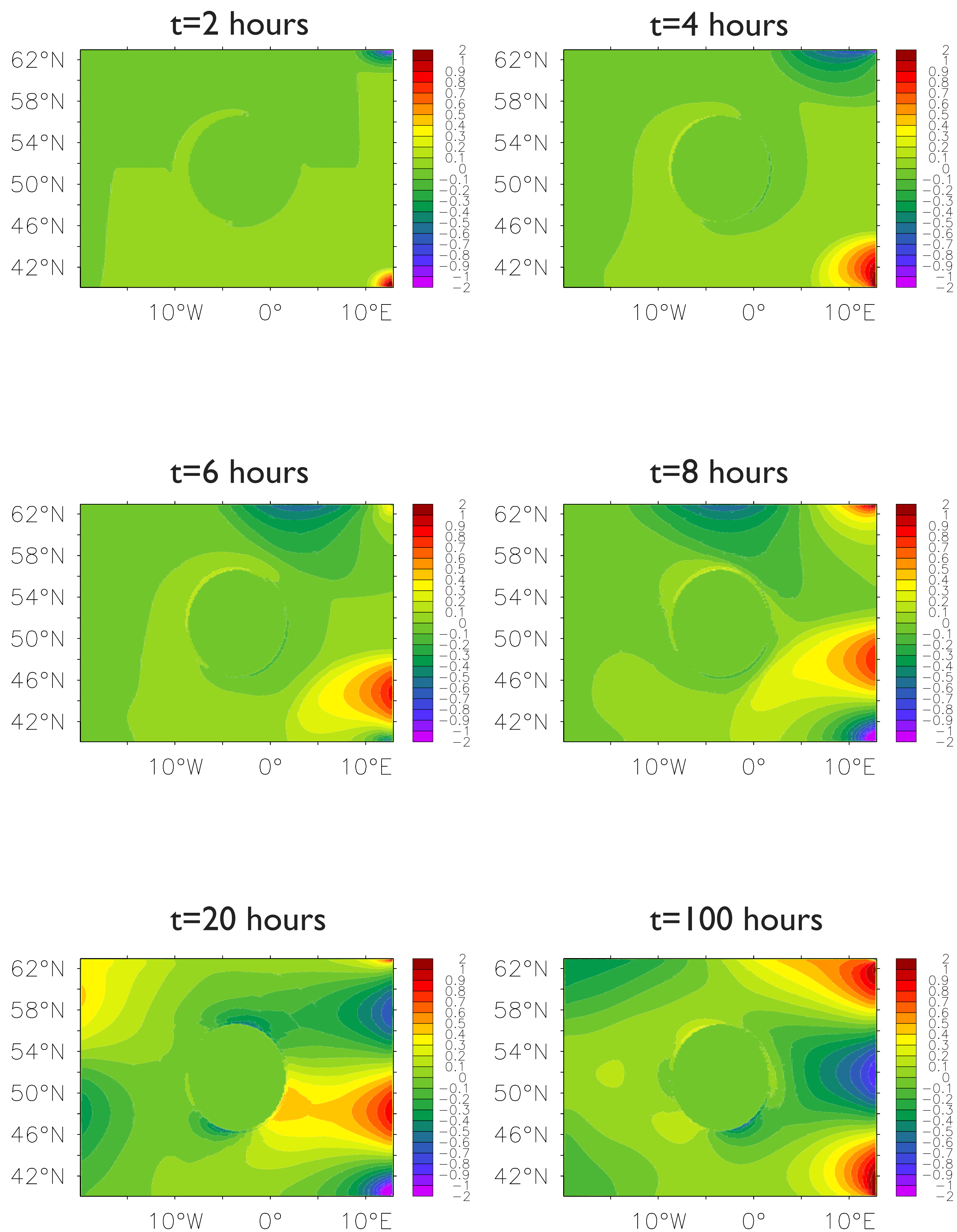
Insert Kelvin wave as meridional barotropic velocity at northern and southern periodic boundaries

$$v = V_0 \exp(X - x/L) \sin(\omega t)$$

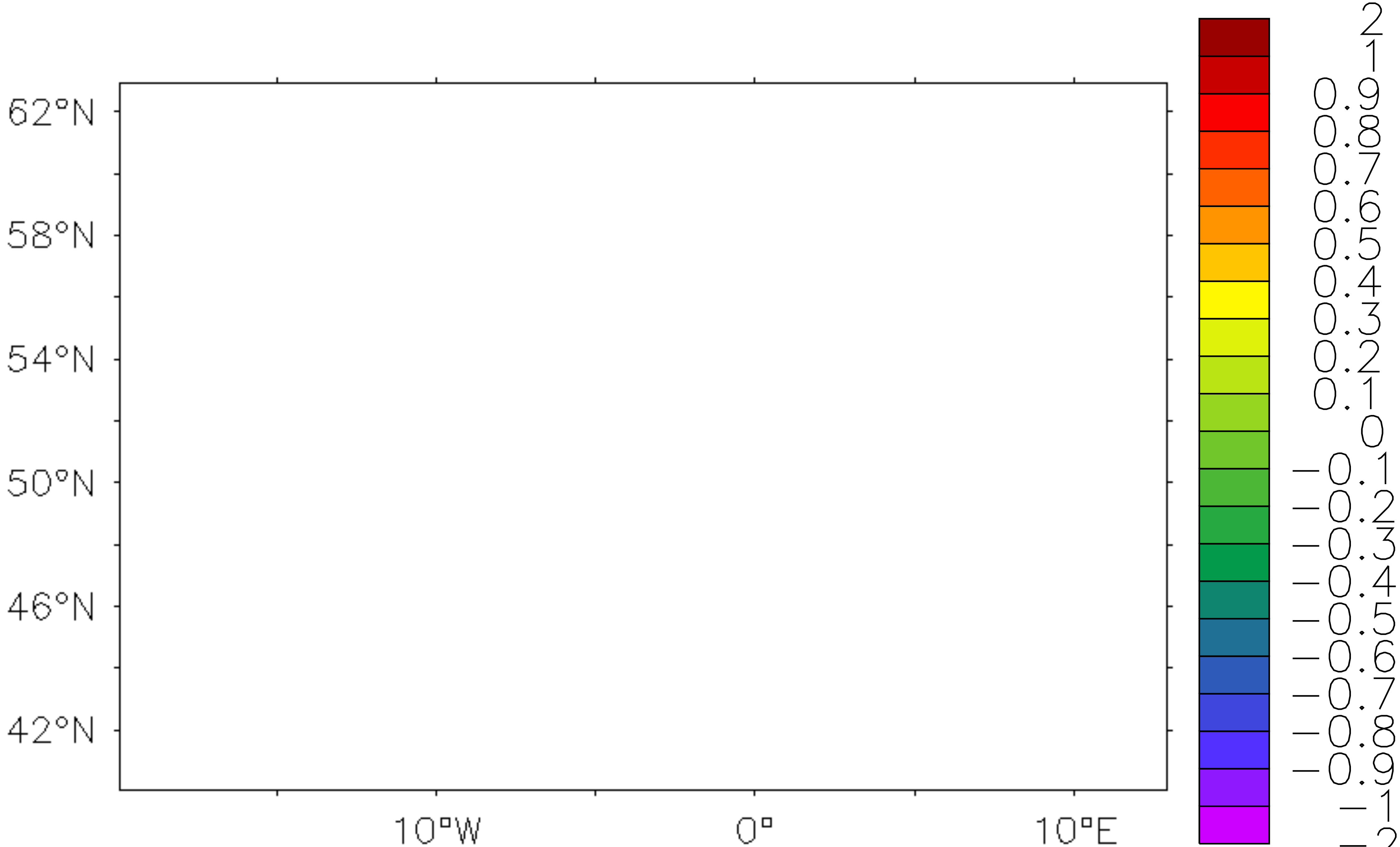
$$V_0 = 1.5\ m/s \qquad L = 30\ km \qquad \omega = 2\pi / 12\ hours$$

Impose zero normal velocity at eastern and western boundaries

Sequence of snapshots of free surface elevation (m) showing spin-up

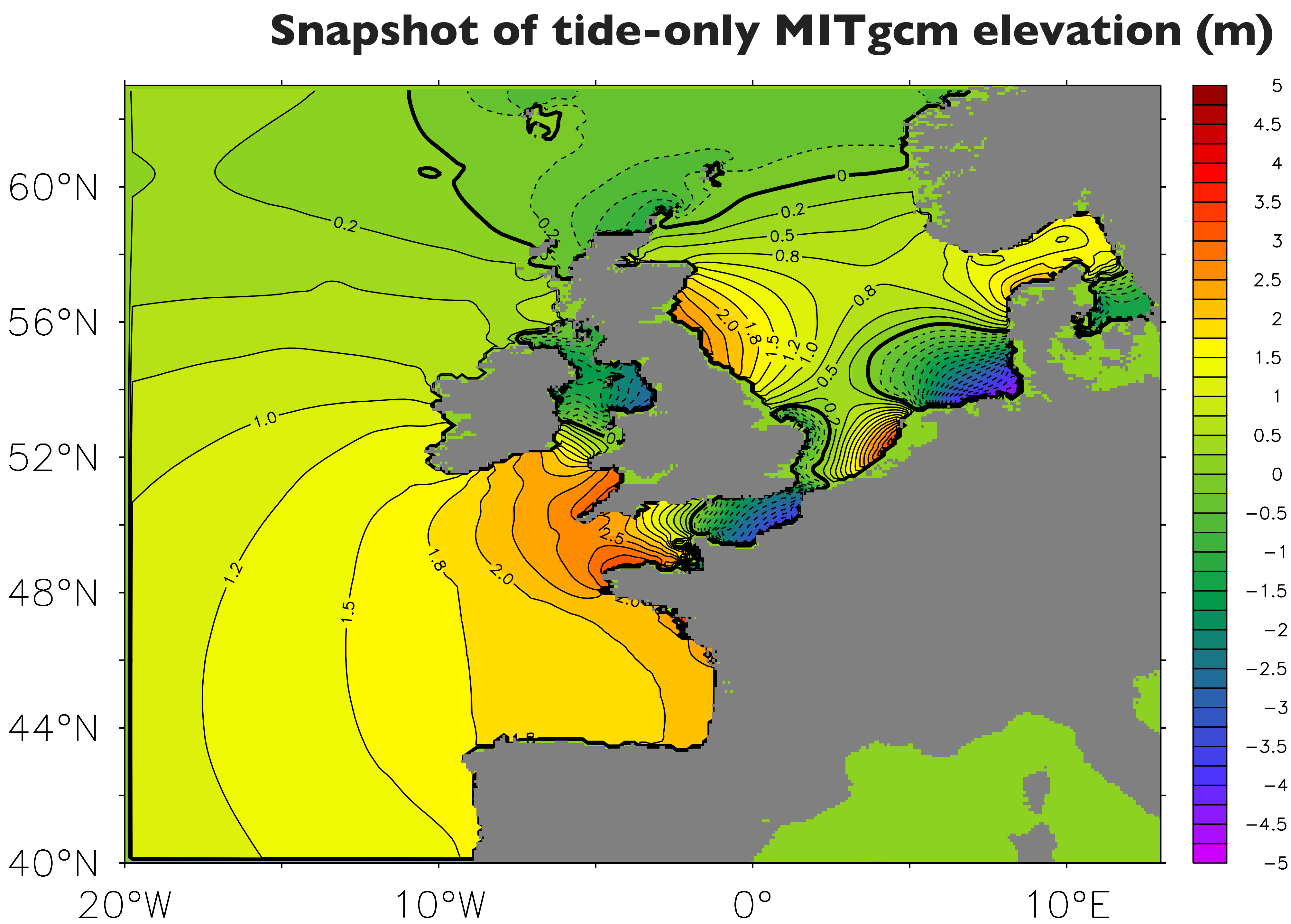


Movie of free surface elevation (m) over first 10 days
1 frame==1 hour; frame rate = 30 fps
Tide-only, no wind forcing



Realistic domain - like CS3X model

Shallow water equations, shaved cell topography
Forced by boundary currents from tide-only run of CS3X



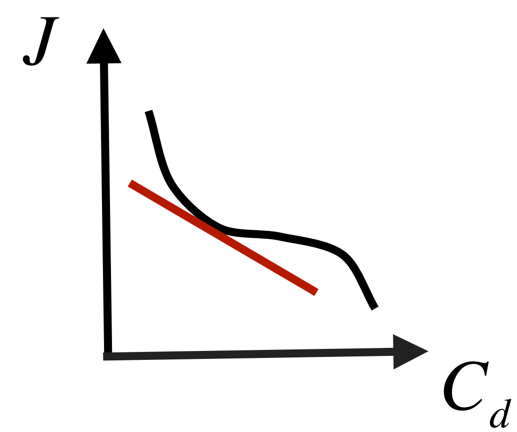
2. Basic forward model experiments

$$\tau_{13}^{bottom-drag} = \left(2A_v \frac{1}{\Delta r_c} + r_b + C_d \sqrt{2\overline{KE}^i}\right) u \quad (2.138)$$

$$\tau_{23}^{bottom-drag} = \left(2A_v \frac{1}{\Delta r_c} + r_b + C_d \sqrt{2\overline{KE}^j}\right) v \quad (2.139)$$

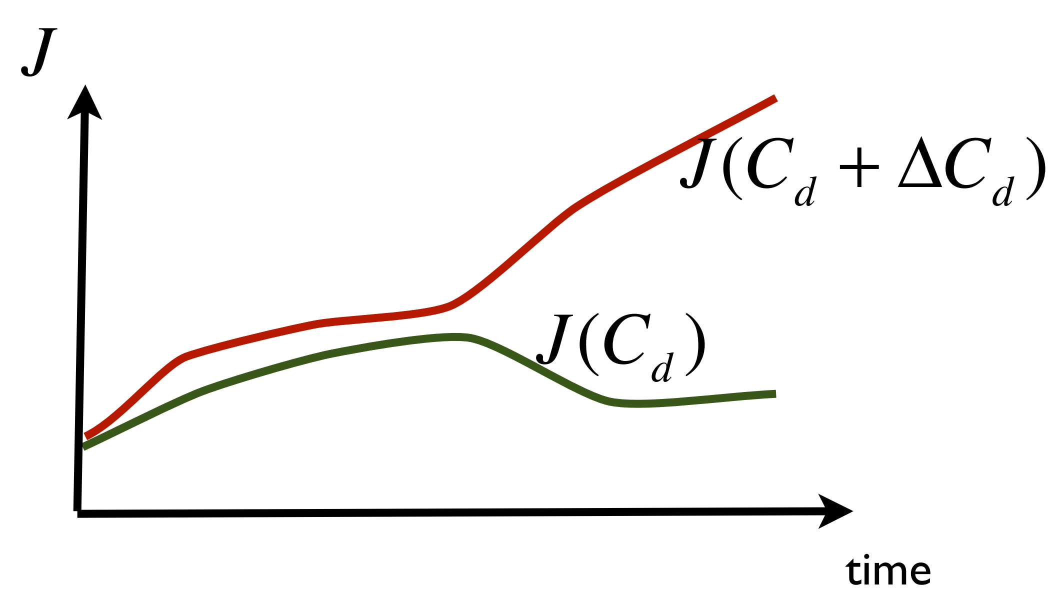
where these terms are only evaluated immediately above topography. r_b (**bottomDragLinear**) has units of ms^{-1} and a typical value of the order 0.0002 ms^{-1}. C_d (**bottomDragQuadratic**) is dimensionless with typical values in the range 0.001–0.003.

$$J = \max_{x,y}(Z)$$



sensitivity $\dfrac{\partial J}{\partial C_d}$

$$\frac{\partial J}{\partial C_d} \approx \frac{J(C_d + \Delta C_d) - J(C_d)}{\Delta C_d}$$



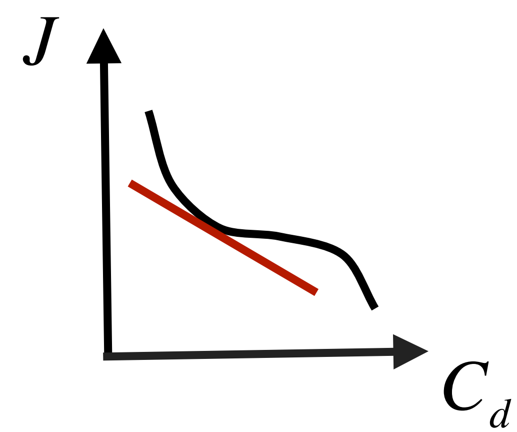
2. Basic forward model experiments

$$\tau_{13}^{bottom-drag} = \left(2A_v \frac{1}{\Delta r_c} + r_b + C_d \sqrt{2\overline{KE}^i}\right) u \qquad (2.138)$$

$$\tau_{23}^{bottom-drag} = \left(2A_v \frac{1}{\Delta r_c} + r_b + C_d \sqrt{2\overline{KE}^j}\right) v \qquad (2.139)$$

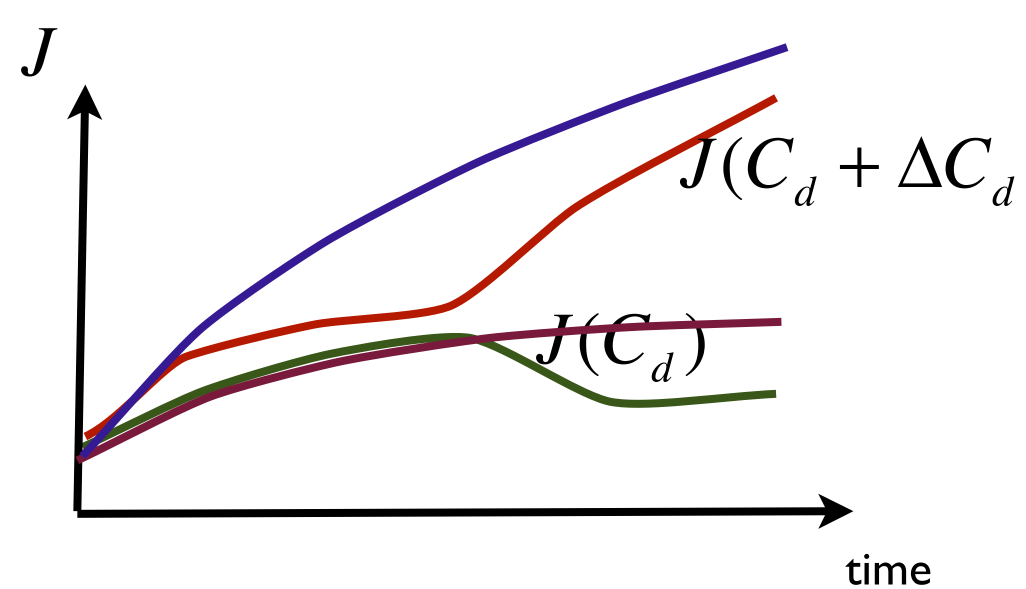
where these terms are only evaluated immediately above topography. r_b (**bottomDragLinear**) has units of ms^{-1} and a typical value of the order 0.0002 ms^{-1}. C_d (**bottomDragQuadratic**) is dimensionless with typical values in the range 0.001–0.003.

$$J = \max_{x,y}(Z)$$



sensitivity $\dfrac{\partial J}{\partial C_d}$

$$\frac{\partial J}{\partial C_d} \approx \frac{J(C_d + \Delta C_d) - J(C_d)}{\Delta C_d}$$

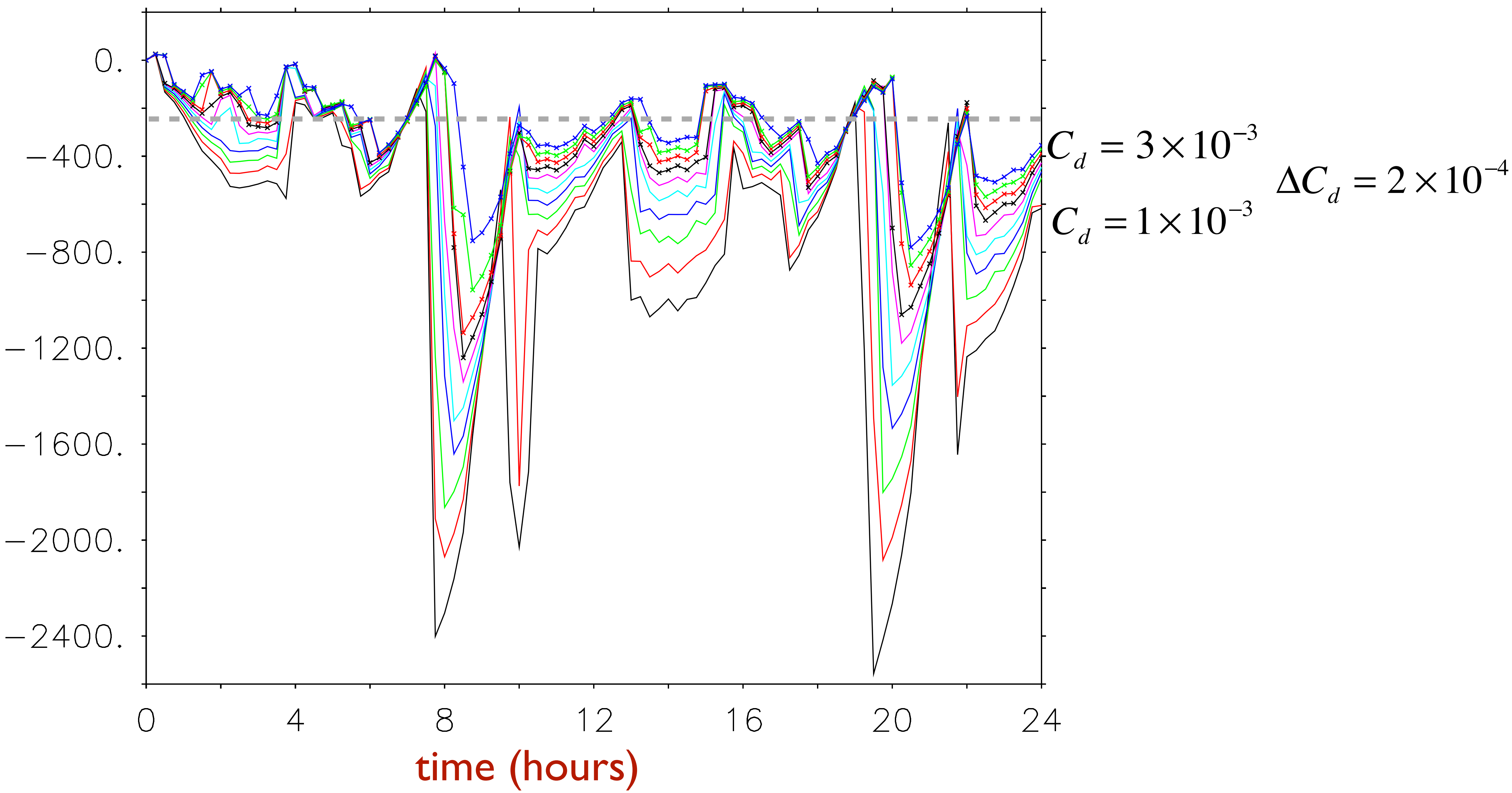


Sensitivity estimate from 11 member ensemble of forward model

$$\frac{J(C_d + \Delta C_d) - J(C_d)}{\Delta C_d} \approx \frac{\partial J}{\partial C_d}$$ (units=metres)

time dependence

larger ensemble needed



3. Build adjoint model **Trickier than I thought**

Automatic differentiation performed by

- TAMC, TAF (Ralf Gierling) ~ not free
- OpenAD ~ free

Routines parse the code line by line, evaluating the Jacobian of $\mathcal{M}$

Patrick Heimbach (MIT), Jean Utke (Argonne National Lab.) leading development of OpenAD with MITgcm

Still deciding on a standard Fortran compiler for OS X
Jean previously used Open64, but that has its pitfalls

Looking likely that Intel/TAF will be necessary - also recommended by Oxford group

Summary

- Adjoint models are useful for answering questions about forecast sensitivity

- MITgcm is carefully constructed to allow Auto Differentiation tools to build adjoint
- needs to be setup for tide-surge modelling on NW European shelf

- Promising results
- OpenAD problems need to be overcome

cwi@noc.ac.uk

Two routes may be followed to determine the sensitivity of the output variable $\vec{v}$ to its input $\vec{u}$.

5.1.1 Forward or direct sensitivity

Consider a perturbation to the input variables $\delta\vec{u}$ (typically a single component $\delta\vec{u} = \delta u_i\,\vec{e}_i$). Their effect on the output may be obtained via the linear approximation of the model $\mathcal{M}$ in terms of its Jacobian matrix M, evaluated in the point $u^{(0)}$ according to

$$\delta\vec{v} \;=\; M|_{\vec{u}^{(0)}}\,\delta\vec{u} \tag{5.2}$$

with resulting output perturbation $\delta\vec{v}$. In components $M_{ji} \;=\; \partial\mathcal{M}_j/\partial u_i$, it reads

$$\delta v_j \;=\; \sum_i \left.\frac{\partial\mathcal{M}_j}{\partial u_i}\right|_{u^{(0)}} \delta u_i \tag{5.3}$$

Eq. (5.2) is the tangent linear model (TLM). In contrast to the full nonlinear model $\mathcal{M}$, the operator M is just a matrix which can readily be used to find the forward sensitivity of $\vec{v}$ to perturbations in u, but if there are very many input variables ($\gg O(10^6)$ for large-scale oceanographic application), it quickly becomes prohibitive to proceed directly as in (5.2), if the impact of each component $\mathbf{e_i}$ is to be assessed.

5.1.2 Reverse or adjoint sensitivity

Let us consider the special case of a scalar objective function $\mathcal{J}(\vec{v})$ of the model output (e.g. the total meridional heat transport, the total uptake of CO_2 in the Southern Ocean over a time interval, or a measure of some model-to-data misfit)

$$\begin{array}{cccccc} \mathcal{J}\,: & U & \longrightarrow & V & \longrightarrow & I\!R \\ & \vec{u} & \longmapsto & \vec{v} = \mathcal{M}(\vec{u}) & \longmapsto & \mathcal{J}(\vec{u}) = \mathcal{J}(\mathcal{M}(\vec{u})) \end{array} \tag{5.4}$$

The perturbation of $\mathcal{J}$ around a fixed point $\mathcal{J}_0$,

$$\mathcal{J} \;=\; \mathcal{J}_0 \,+\, \delta\mathcal{J}$$

can be expressed in both bases of $\vec{u}$ and $\vec{v}$ w.r.t. their corresponding inner product $\langle\;,\;\rangle$

$$\begin{aligned} \mathcal{J} &= \mathcal{J}|_{\vec{u}^{(0)}} \,+\, \left\langle \nabla_u\mathcal{J}^T|_{\vec{u}^{(0)}} \,,\, \delta\vec{u} \right\rangle \,+\, O(\delta\vec{u}^2) \\ &= \mathcal{J}|_{\vec{v}^{(0)}} \,+\, \left\langle \nabla_v\mathcal{J}^T|_{\vec{v}^{(0)}} \,,\, \delta\vec{v} \right\rangle \,+\, O(\delta\vec{v}^2) \end{aligned} \tag{5.5}$$

(note, that the gradient ∇f is a co-vector, therefore its transpose is required in the above inner product). Then, using the representation of $\delta\mathcal{J} = \left\langle \nabla_v\mathcal{J}^T \,,\, \delta\vec{v} \right\rangle$, the definition of an adjoint operator A^* of a given operator A,

$$\langle\, A^*\vec{x} \,,\, \vec{y} \,\rangle = \langle\, \vec{x} \,,\, A\vec{y} \,\rangle$$

which for finite-dimensional vector spaces is just the transpose of A,

$$A^* \;=\; A^T$$

and from eq. (5.2), (5.5), we note that (omitting |'s):

$$\delta\mathcal{J} \;=\; \left\langle \nabla_v\mathcal{J}^T \,,\, \delta\vec{v} \right\rangle \;=\; \left\langle \nabla_v\mathcal{J}^T \,,\, M\,\delta\vec{u} \right\rangle \;=\; \left\langle M^T\,\nabla_v\mathcal{J}^T \,,\, \delta\vec{u} \right\rangle \tag{5.6}$$

With the identity (5.5), we then find that the gradient $\nabla_u\mathcal{J}$ can be readily inferred by invoking the adjoint M^* of the tangent linear model M

$$\begin{aligned} \nabla_u\mathcal{J}^T|_{\vec{u}} &= M^T|_{\vec{u}} \cdot \nabla_v\mathcal{J}^T|_{\vec{v}} \\ &= M^T|_{\vec{u}} \cdot \delta\vec{v}^* \\ &= \delta\vec{u}^* \end{aligned} \tag{5.7}$$

Eq. (5.7) is the adjoint model (ADM), in which M^T is the adjoint (here, the transpose) of the tangent linear operator M, $\delta\vec{v}^*$ the adjoint variable of the model state $\vec{v}$, and $\delta\vec{u}^*$ the adjoint variable of the control variable $\vec{u}$.