

Experimental design  
○○○○

Frames of reference  
○

Lagrangian method  
○

Semi-Lagrangian method  
○○○

Semi-Implicit Methods  
○○

Some real world a  
○○○○○

# Numerical modelling of atmosphere and oceans

## Lecture 7

P.L. Vidale

NCAS Climate, Department of Meteorology  
University of Reading, UK  
[p.l.vidale@reading.ac.uk](mailto:p.l.vidale@reading.ac.uk)

2 March 2023

## 1 Experimental design

Choosing the type of model

Numerical analysis tied to phenomena

## 2 Frames of reference

Eulerian and Lagrangian

## 3 Lagrangian method

Trajectory calculations

## 4 Semi-Lagrangian method

Semi Lagrangian in practice

## 5 Semi-Implicit Methods

## 6 Some real world applications

## 7 Extras

Eulerian and Lagrangian rates of change

Contour dynamics

Hybrid models

# Grid point methods

Let us say that we are working with a periodic function  $f$  in the interval  $0..2\pi$ , as shown in the figure below.

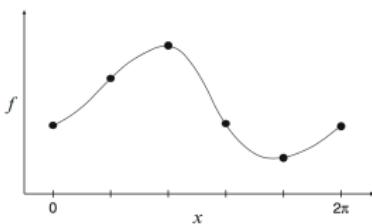


Fig. 1.1 Grid-point approximation of a periodic function on the interval  $[0, 2\pi]$ . Individual points show the function values at intervals of  $2\pi/5$

In grid-point methods, each function is approximated by a value at a set of discrete grid points. Using a finite difference method, we can approximate the derivative this way:

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \quad (1)$$

# Finite volumes

Using a finite volume method, we do not really need to compute derivatives, rather the fluxes between cells, but we will need to approximate the structure of the solution inside the grid cell, for instance with a piecewise constant function, or a piecewise linear function, like this:

$$f(x) \approx f_j + \sigma_j(x - j\Delta x) \text{ for all } x \in (j - \frac{1}{2}\Delta x, j + \frac{1}{2}\Delta x) \quad (2)$$

where  $f_j$  is the average of the approximate solution over the grid cell centered at  $j\Delta x$  and  $\sigma_j = \frac{(f_{j+1} - f_j)}{\Delta x}$

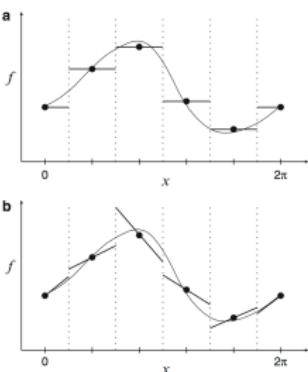


Fig. 1.2 Finite-volume approximation of a periodic function on the interval  $[0, 2\pi]$  using a piecewise-constant functions, and b piecewise-linear functions

# Series expansion methods: spectral methods

In series expansion methods, the unknown function is approximated by a linear combination of a finite set of continuous expansion functions, and the data set describing the approximated function is the finite set of expansion coefficients.

When the expansion functions form an **orthogonal set**, the series-expansion approach is a spectral method. For instance, we have seen already the use of Fourier series:

$$a_1 + a_2 \cos x + a_3 \sin x + a_4 \cos 2x + a_5 \sin 2x \quad (3)$$

The goal is always to find coefficients that minimise the error. The five coefficients above need not be chosen such that the value of the Fourier series exactly matches the value of  $f(x)$  at any specific point in the interval  $0 \leq x \leq \pi$ . However, we could do it at specific grid points by using the grid method we discussed in the spectral methods lecture.

An alternative method is to try to minimise the integral of the square of the error (residual) in the  $x$  domain.

# Series expansion methods: Finite Elements

If the expansion functions are nonzero in only a small part of the domain, the series expansion technique is a finite-element method.

$$f(x) \approx b_0 s_0(x) + b_1 s_1(x) + \dots + b_5 s_5(x) \quad (4)$$

similar to the spectral method, but now the functions  $s_n$  differ from trigonometric functions, because each function  $s$  is zero over most of the domain. The simplest FE expansion functions  $s$  are piecewise-linear functions defined over a grid: each function is equal to 1 at one grid point (or node) and zero everywhere else. The values of the expansion function between nodes are determined by linear interpolation.

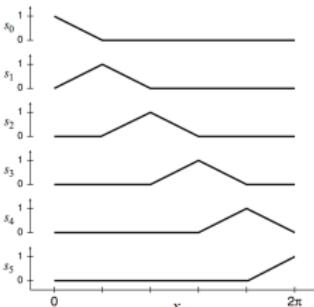
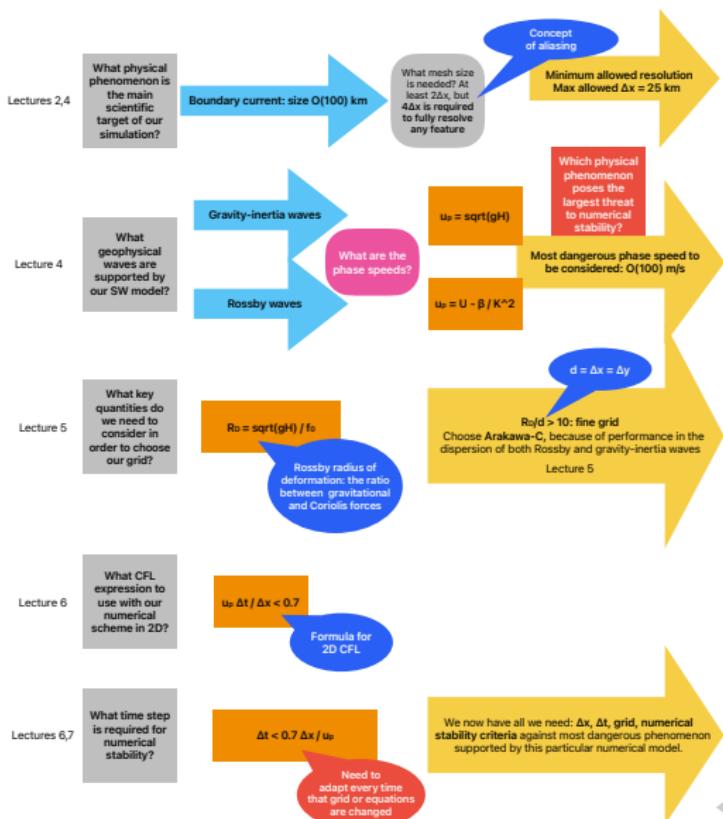


Fig. 1.3 Six finite-element expansion functions,  $s_0(x), s_1(x), \dots, s_5(x)$



# What we learned in designing Project 2



This is all to say that there is a logical sequence to be followed when making key decisions on how to set up our experiment. We start with our scientific objectives, but we must consider several worst case scenarios, in order to make sure that our model will not become unstable, else return poor quality solutions.

# A reminder of the differences between Eulerian and Lagrangian frames of reference

We want to describe the evolution of a chemical tracer  $\Psi(x, t)$  in a one dimensional flow field, with sources/sinks  $S(x, t)$ . We can do so within two frameworks:

- Eulerian:  $\frac{\partial \Psi}{\partial t} + u \frac{\partial \Psi}{\partial x} = S$



- Lagrangian:  $\frac{d\Psi}{dt} = S \quad (1)$



The two are tied by the definition of the total derivative:  $\frac{d}{dt} = \frac{\partial}{\partial t} + \frac{dx}{dt} \frac{\partial}{\partial x}$  and the definition of velocity:  $\frac{dx}{dt} = u \quad (2)$

We could solve (1) as an initial value problem, by choosing a number of regularly spaced fluid particles at  $t = 0$ , assigning a  $\Psi$  value to each and then following them around the flow by integrating the two ODEs (1) and (2) in time.

Problem: the parcels would very likely spread out in a non-homogeneous fashion, so that any numerical approximation of  $\Psi(x, t)$  will become highly inaccurate wherever they are sparse (see the extra slides at the end for more). What can we do?

# Lagrangian models in practice

First calculate fluid parcel trajectories by integrating:

$$\frac{dx}{dt} = u(x, t) \quad (3)$$

and then calculate the evolution of air parcel properties,  $\Psi$ , following fluid parcels by modelling  $S$  and integrating (3).

**Advantages:** No underlying grid, so can represent air masses accurately as they thin to arbitrarily fine-scales. Also, no CFL criterion to limit time-step (i.e., parcels can travel far in one time-step).

**Disadvantages:** Velocity stirs air parcels so they are irregularly spaced and often far apart, making estimation of gradients difficult. In *kinematic models*, velocity is given but mixing between air-masses typically depends on concentration gradients.

# A compromise

We learned that:

- Eulerian frameworks are limited by CFL criteria, thus requiring short time steps (e.g. I would need to use  $\Delta t = 0.3\text{s}$  in my 10km GCM, instead of 4 minutes).
- Lagrangian frameworks are most often impractical<sup>1</sup> and/or locally inaccurate.

A better scheme could be chosen, in which we re-define the number and distribution of the fluid parcels at every time step. We choose parcels in this set as being those arriving at each node on a regularly spaced grid at the end of each time step. This will automatically regulate the number and distribution of the fluid parcels. **This is known as the semi-Lagrangian method (Wiin-Nielsen, 1959).**

In practice, choose  $t^n = n\Delta t$  and  $x_j = n\Delta x$ , then (1) can be approximated as:

$$\frac{\Phi(x_j, t^{n+1}) - \Phi(\tilde{x}_j^n, t^n)}{\Delta t} = \frac{S(x_j, t^{n+1}) + S(\tilde{x}_j^n, t^n)}{2} \quad (4)$$

where  $\Phi$  is the numerical approximation of  $\Psi$  and  $\tilde{x}_j^n$  is the estimated x coordinate of the departure point of the trajectory originating at time  $t$  and arriving at grid point and time:  $(x_j, t^{n+1})$ . The value of  $\tilde{x}_j^n$  can be found by integrating (3) backwards over a time interval  $\Delta t$ , with initial condition:  $x(t^{n+1}) = x_j$ . We shall need interpolation...

---

<sup>1</sup>we could remove parcels from regions where they are too abundant, add them where they are sparse ▶

## A simple 1D example: temperature evolution. I

We want to predict the evolution of temperature for a fluid. We postulate that there are no sources/sinks of energy, so that temperature is conserved and we can say that, from the *Lagrangian* perspective:

$$\frac{dT}{dt} = 0 \quad (5)$$

If we now take the *Eulerian* view and we decide to predict the evolution of temperature at a particular location, the local change of temperature will be governed by temperature advection alone:

$$\frac{\partial T}{\partial t} = -U \frac{\partial T}{\partial x} \quad (6)$$

where  $T$  is temperature [ $K$ ], the variable we are predicting;  $t$  is time [ $s$ ],  $x$  is the zonal distance [ $m$ ] and  $U$  is the zonal velocity [ $ms^{-1}$ ].

## A simple 1D example: temperature evolution. II

We can also solve 5 by using the semi-Lagrangian (SL) method, which also enables a longer time step:

$$\Delta t_s = n \Delta t_e \quad (7)$$

where  $\Delta t_s$  is the SL time step and  $\Delta t_e$  is the Eulerian time step. We will start from our original grid at time  $t^0 = 0s$ . **Remember that, with SL, you will be using the total derivative**, because the problem, seen from a Lagrangian perspective, is simply:

$$\frac{dT}{dt} = 0 \quad (8)$$

A semi-Lagrangian approximation to 8 can be written in this form:

$$\frac{T(x_j, t^{n+1}) - T(\tilde{x}_j^n, t^n)}{\Delta t} = 0 \quad (9)$$

where  $\tilde{x}_j^n$  denotes the point of origin of a trajectory originating at time  $t^n$  and arriving at point  $(x_j, t^{n+1})$ .

Since velocity  $U$  is constant, it is quite easy to show that:

$$\tilde{x}_j^n = x_j - U\Delta t \quad (10)$$

# Semi-Lagrangian Methods in 2D: how to find $\tilde{x}_{i,j}^n$

Calculate short trajectories backward-in-time from a fixed grid and use them to evaluate the Lagrangian rates of change at every grid-point. Trajectory calculation can be cheap, because quite short. For example, the two-stage mid-point method:

$$\begin{aligned} x_* &= x_{i,j}^{n+1} - u(x_{i,j}^{n+1}, t^n) \Delta t / 2 \\ \tilde{x}_{i,j}^n &= x_{i,j}^{n+1} - u(x_*, t^{n+\frac{1}{2}}) \Delta t \end{aligned}$$

gives the *departure points*,  $\tilde{x}_{i,j}^n$ . The advection equation is then solved using simple methods such a trapezoidal scheme (see Durran's book, Chap. 6):

$$\frac{\Phi_{i,j}^{n+1} - \Phi(\tilde{x}_{i,j}^n, t^n)}{\Delta t} \approx \frac{1}{2} \left\{ S_{i,j}^{n+1} + \tilde{S}_{i,j}^n \right\} \quad (11)$$

**Advantages:** avoids CFL criterion for numerical stability (especially nonlinear  $\mathbf{u} \cdot \nabla \mathbf{u}$  term), allowing longer time-step. For same accuracy Ritchie *et al*, found that the time-step of the ECMWF forecast model could be increased from 3 to 15 minutes.

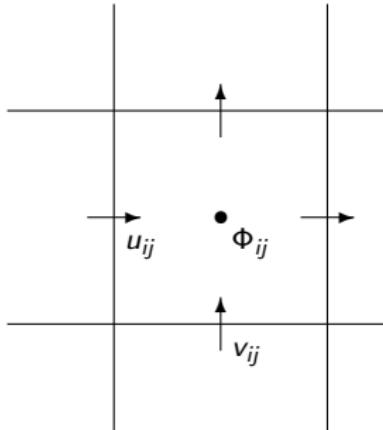
**Disadvantages:** the schemes are not positive definite. Interpolation is necessary from the grid to the departure points, which is equivalent to strong *numerical dissipation*.

# Semi-Lagrangian Methods: in practice

We aim to predict the value of  $\Phi$  at time  $t = n + 1$ , so  $\Phi(x^{n+1})$ , or  $\Phi_{i,j}^{n+1}$ .

Because we are projecting a Lagrangian calculation onto an Eulerian grid, we always know where we are going to be at time  $t=n+1$ : the **arrival point**  $x^{n+1} = x_{i,j}$ .

Unlike in purely Eulerian frameworks, we do not know a-priori where information is coming from; we must compute the **departure point**  $\tilde{x}_{i,j}^n$ .



**Target is the future (time = n+1):**  $\Phi_{i,j}^{n+1}$

**Available to us now (time = n):**  $\Phi_{i,j}^n, u_{i,j}^n, S_{i,j}^n$

We integrate  $u_{i,j}^n$  to find  $\tilde{x}_{i,j}^n$ , thus  $\Phi(\tilde{x}_{i,j}^n, t^n)$  and  $S(\tilde{x}_{i,j}^n, t^n)$ .

Finally, we compute the future state like this:

$$\Phi_{i,j}^{n+1} \approx \Phi(\tilde{x}_{i,j}^n, t^n) + \Delta t \left\{ \frac{S(x_{i,j}, t^{n+1}) + S(\tilde{x}_{i,j}^n, t^n)}{2} \right\} \quad (12)$$

# Semi-Lagrangian Methods: from first order to second order

It is perfectly possible to only use a single time level to compute departure points.

However, if we were to simply make use of  $u_{i,j}^n$  to find  $\tilde{x}_{i,j}^n$ , we would inevitably end up with a first order scheme:

$$\tilde{x}_{i,j}^n = x_{i,j}^{n+1} - u(x_{i,j}^{n+1}, t^n) \Delta t$$

Instead, we saw previously that we can use a two-stage mid-point method:

$$x_* = x_{i,j}^{n+1} - u(x_{i,j}^{n+1}, t^n) \Delta t / 2$$

$$\tilde{x}_{i,j}^n = x_{i,j}^{n+1} - u(x_*, t^{n+\frac{1}{2}}) \Delta t$$

Ideally we would want to compute the mid-point,  $x_*$ , based on current and future velocities, but this would result in an implicit scheme (and expensive iteration). It is possible, instead, to use interpolation and *forward extrapolation* in time to keep the scheme explicit and yet second-order. When computing this term:  $u(x_*, t^{n+\frac{1}{2}})$ , we can write:

$$u(t^{n+\frac{1}{2}}) = \frac{3}{2}u(t^n) - \frac{1}{2}u(t^{n-1}) \quad (13)$$

Note that we now need to carry two time levels.

# Semi-Implicit Methods (see also the last slides in lecture 5)

Explicit methods tend to have a time-step restriction for numerical stability at a given spatial resolution, summarised by the CFL criterion:

$$\alpha = \frac{c\Delta t}{\Delta x} < 1$$

where  $c$  is the fastest speed of information propagation. Generally, flows are *unbalanced* meaning that there are fast waves involving fluid movement that is not related to PV. Such fast waves can limit the time-step. Then, the semi-Lagrangian method does not help, because it only ensures stability with respect to advection.

- Sound waves are fastest but filtered out by the anelastic approximation.
- Gravity waves are not filtered out unless more severe balance approximations are made - therefore  $c_{GW}(> U)$  limits  $\Delta t$ .
- Semi-implicit methods lift this restriction by treating the **gravity wave terms** implicitly while the rest of the terms in the equations are explicit.

All global NWP models currently use semi-implicit, semi-Lagrangian methods attempting to achieve stability for long time-steps ( $1 < \alpha < 10$ ).

# From Lecture 5: gravity waves and Semi Implicit

In atmospheric models, the fastest gravity waves, i.e., the external-gravity or “Lamb” waves, have speeds on the order of  $300 \text{ ms}^{-1}$ , which is also the speed of sound, requiring a very short time step. This is unfortunate, because the external gravity modes are believed to play only a minor role in weather and climate dynamics.

The SWEs are discretised below using a leapfrog scheme for Coriolis terms but a trapezoidal scheme (mixed implicit-explicit) for the gravity wave terms:

$$\begin{aligned}\frac{u^{n+1} - u^{n-1}}{2\Delta t} - fv^n + \frac{g}{2} \left( \frac{\partial h^{n+1}}{\partial x} + \frac{\partial h^{n-1}}{\partial x} \right) &= 0 \\ \frac{v^{n+1} - v^{n-1}}{2\Delta t} + fu^n + \frac{g}{2} \left( \frac{\partial h^{n+1}}{\partial y} + \frac{\partial h^{n-1}}{\partial y} \right) &= 0 \\ \frac{h^{n+1} - h^{n-1}}{2\Delta t} + \frac{H}{2} \left( \frac{\partial u^{n+1}}{\partial x} + \frac{\partial u^{n-1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} + \frac{\partial v^{n-1}}{\partial y} \right) &= 0.\end{aligned}$$

Re-arranging with future values on the left:

$$\begin{aligned}u^{n+1} + \Delta t g \frac{\partial h^{n+1}}{\partial x} &= A \\ v^{n+1} + \Delta t g \frac{\partial h^{n+1}}{\partial y} &= B \\ h^{n+1} + \Delta t H \left( \frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} \right) &= C\end{aligned}$$

Experimental design  
○○○○

Frames of reference  
○○

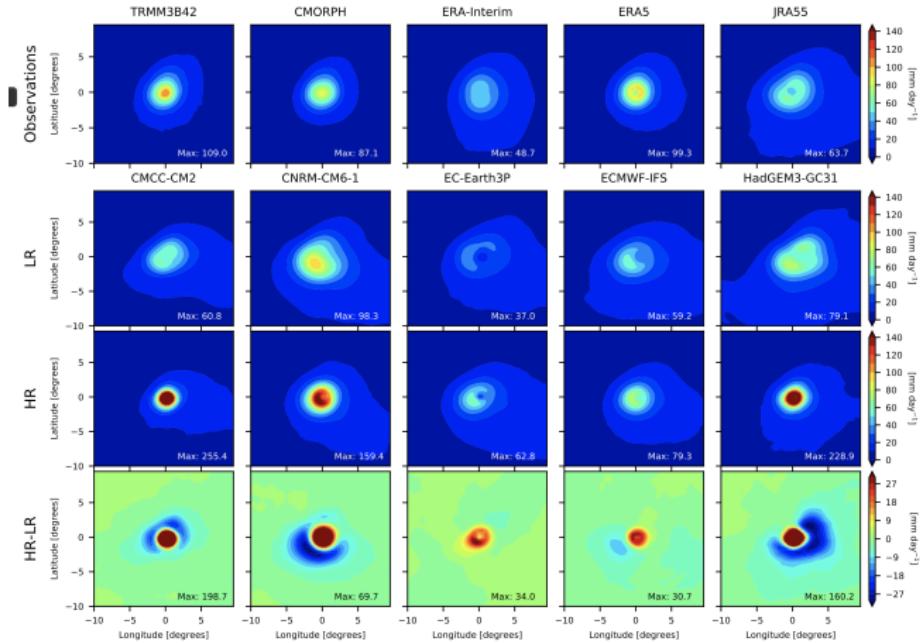
Lagrangian method  
○○○○

Semi-Lagrangian method  
○○○○

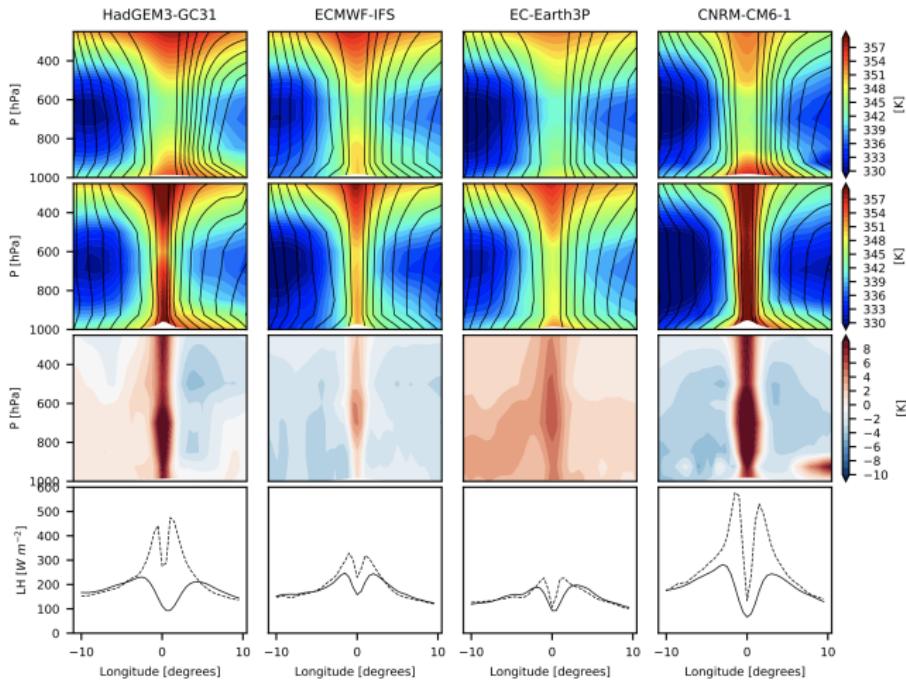
Semi-Implicit Methods  
○○○○

Some real world  
●○○○○

# Hurricane and Typhoon simulation in climate GCMs: precipitation composite



# Hurricane and Typhoon simulation in climate GCMs: vertical structure composite



Experimental design  
○○○○

Frames of reference  
○○

Lagrangian method  
○○

Semi-Lagrangian method  
○○○○

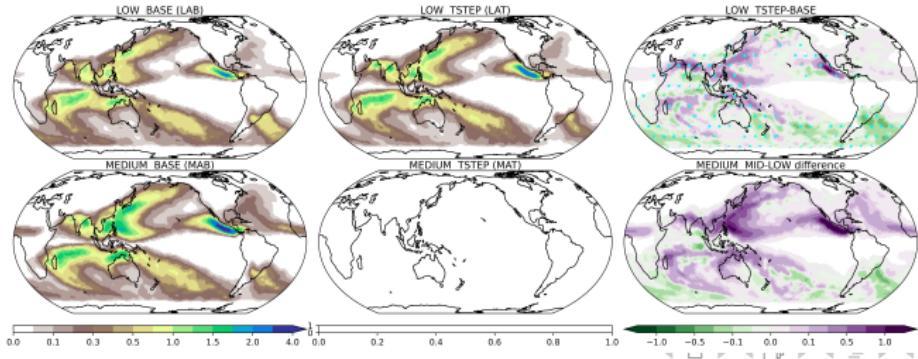
Semi-Implicit Methods  
○○

Some real world a  
○○●○○

## Why the differences

These GCMs are very different in terms of dynamical core, parameterizations etc., but they are also quite different in terms of the use of the time step. I have been experimenting with our UK model (HadGEM3), and Colin Zarzycki has been experimenting with the NCAR model, managing to double the number of hurricanes with a time step of 1/4, but I have also tested the same ideas in the ECMWF model, since it is the one that uses very long time steps.

PRESENT : annual climatologies



# Hurricane and Typhoon simulation in climate GCMs: frequency

czarzycki@psu.edu - How physics timestep controls tropical cyclone frequency in high resolution CESM

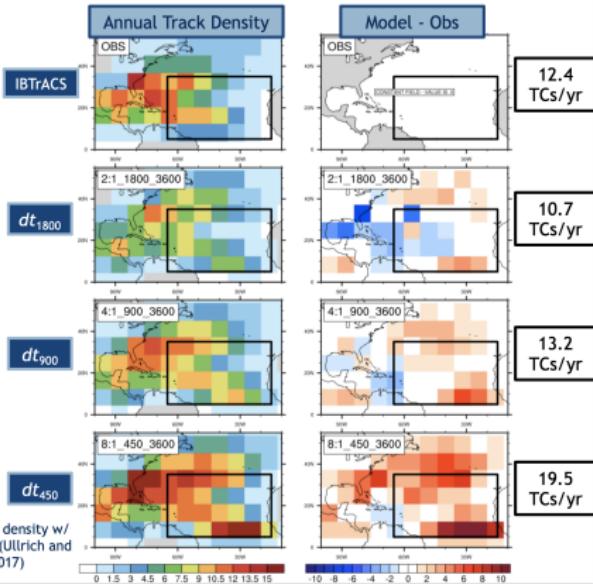
@weatherczar



PennState

## High-res TC sensitivity to timestep

- NCAR CAM5.4  
0.25° tropical  
cyclones (TCs)  
→ marked  
sensitivity to  
physics  
timestep ( $dt$ )
- $dt = 450s$   
produces ~2x  
number of TCs  
than  $dt = 1800s$
- Bit-for-bit  
otherwise!



# Hurricane and Typhoon simulation in climate GCMs: dynamics and physics

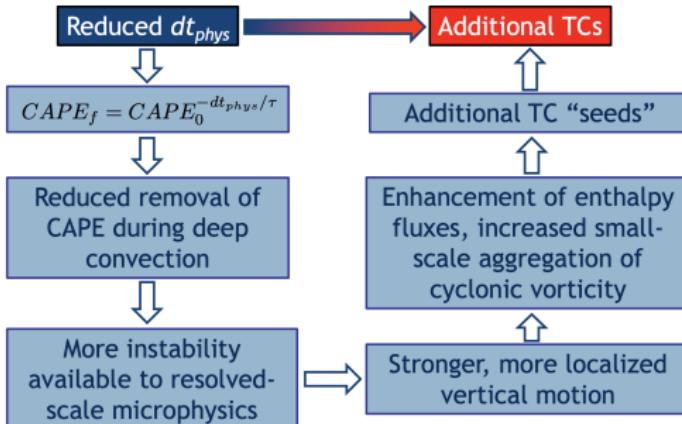
czarzycki@psu.edu - How physics timestep controls tropical cyclone frequency in high resolution CESM

@weatherczar



PennState

## Mechanism and take home thought



- Simulated TCs can be tremendously sensitive to “innate” model design choices!



# Monotonicity and Positive Definiteness

- Many properties are carried with the flow . A statement of **material conservation** is:

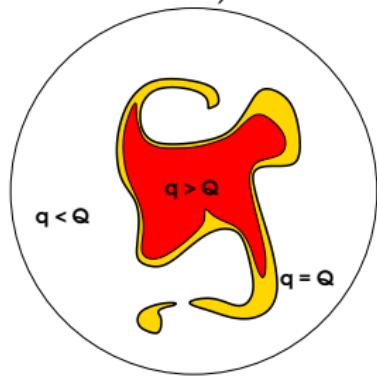
$$\frac{Dq}{Dt} = S \quad (14)$$

where the *Lagrangian derivative* is the rate of change following a fluid parcel:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q \quad (15)$$

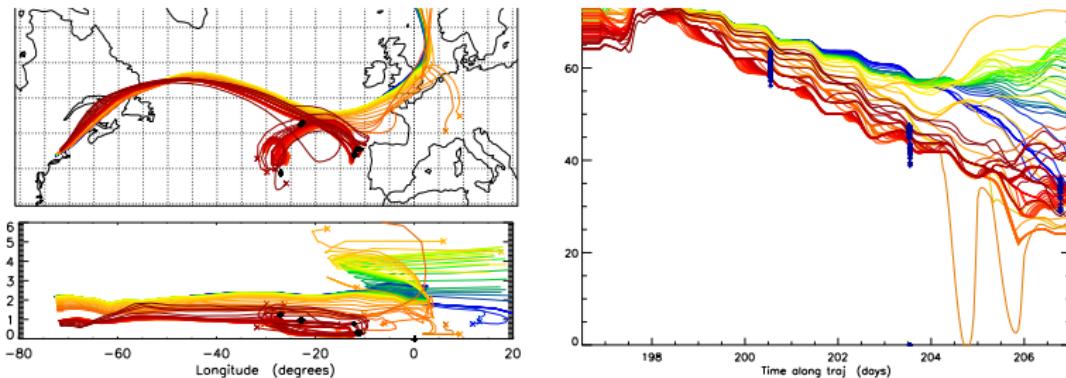
Some implications for solutions (for conserved properties where  $S = 0$ ) are:

- monotonicity preservation:** no new maxima or minima in  $q$  can appear
- positive definiteness:** if tracer has  $q \geq 0$  everywhere at initial time, no negative values can appear (e.g., humidity mixing ratio).



## Kinematic Lagrangian model example

This example shows trajectories calculated from analysed winds (gridded in space and time). Initiated from the coordinates of the flight of a research aircraft<sup>3</sup>, July 2004. Ozone concentrations have been integrated forward in time along the trajectories using a Lagrangian model for photochemistry and mixing, initialised with the aircraft measurements of constituent concentrations.



The crosses show the coordinates and ozone measurements made during three further flights intercepting the air-mass downstream.

<sup>3</sup>

Methven, J. et al. (2006) *J. Geophys. Res.*, **111**, D23S62, doi:10.1029/2006JD007540.

## Balanced dynamics

*Dynamics* involves solution for velocity (and pressure) as well as tracer equations.  
Neither of these are conserved variables.

However, in *balanced dynamics*, potential vorticity (PV) is carried as a *tracer* and all other variables are obtained by *inverting PV*.

Simplest example is the barotropic vorticity equation where

$$q = f + \nabla^2 \psi$$

This is inverted to obtain the streamfunction and geostrophic velocity:

$$\psi = \nabla^{-2}(q - f) \quad u_g = -\frac{\partial \psi}{\partial y} \quad v_g = \frac{\partial \psi}{\partial x}$$

The inversion operation typically does not depend on time, but is a difficult *boundary value problem* akin to integration.

# Models for balanced dynamics

A family of models is obtained by treating the *material conservation* and *PV inversion* operations in an Eulerian or Lagrangian framework.

## Contour dynamics<sup>4</sup> - a fully Lagrangian method

- Nodes are placed around PV contours (more nodes where curvature is higher).
- The inversion step to obtain velocity at the nodes is done by integration around PV contours (without the use of a grid).
- Nodes are stepped forward by calculating trajectories (4th order Runge-Kutta method), giving updated PV contours.
- Repeat the inversion and trajectory steps.

**Advantages:** No underlying grid, so accurate in principle and no CFL  $\Delta t$  restriction.

**Disadvantages:** As the contours stir into convoluted shapes, the number of nodes increases exponentially to retain accuracy. Inversion by contour integration cost scales with *nodes*  $\times$  *contours* and becomes too expensive.

Also implicit assumption is that PV values are discretised changes wave dynamics.

<sup>4</sup>Zabusky et al (1979) *J. Comput. Phys.*, **30**, 96-106.

# Contour-advective semi-Lagrangian (CASL)

Potential vorticity is treated in a Lagrangian way (by calculating trajectories of nodes on tracer contours).

Example here shows a single contour advected by an unsteady polar vortex. Note the finescale filamentary structure.



But velocity and other *wave-like* variables are stored on a fixed grid<sup>5</sup>.

**Advantages:** Avoids numerical dissipation of conserved variables and the CFL criterion. Potential vorticity inversion to obtain *balanced velocity* is cheap because contour-to-grid conversion estimates *coarse-grained PV* on the velocity grid and then inversion can be done by fast Fourier transform methods.

**Disadvantages:** Allowing for non-conservation (e.g., sources/sinks) is difficult because such processes violate monotonicity preservation and must introduce new *q*-contours.

<sup>5</sup>

Dritschel, D.G. and Ambaum, M.H.P. (1997) *Quart. J. Roy. Meteor. Soc.*, **123**, 1097-1130.