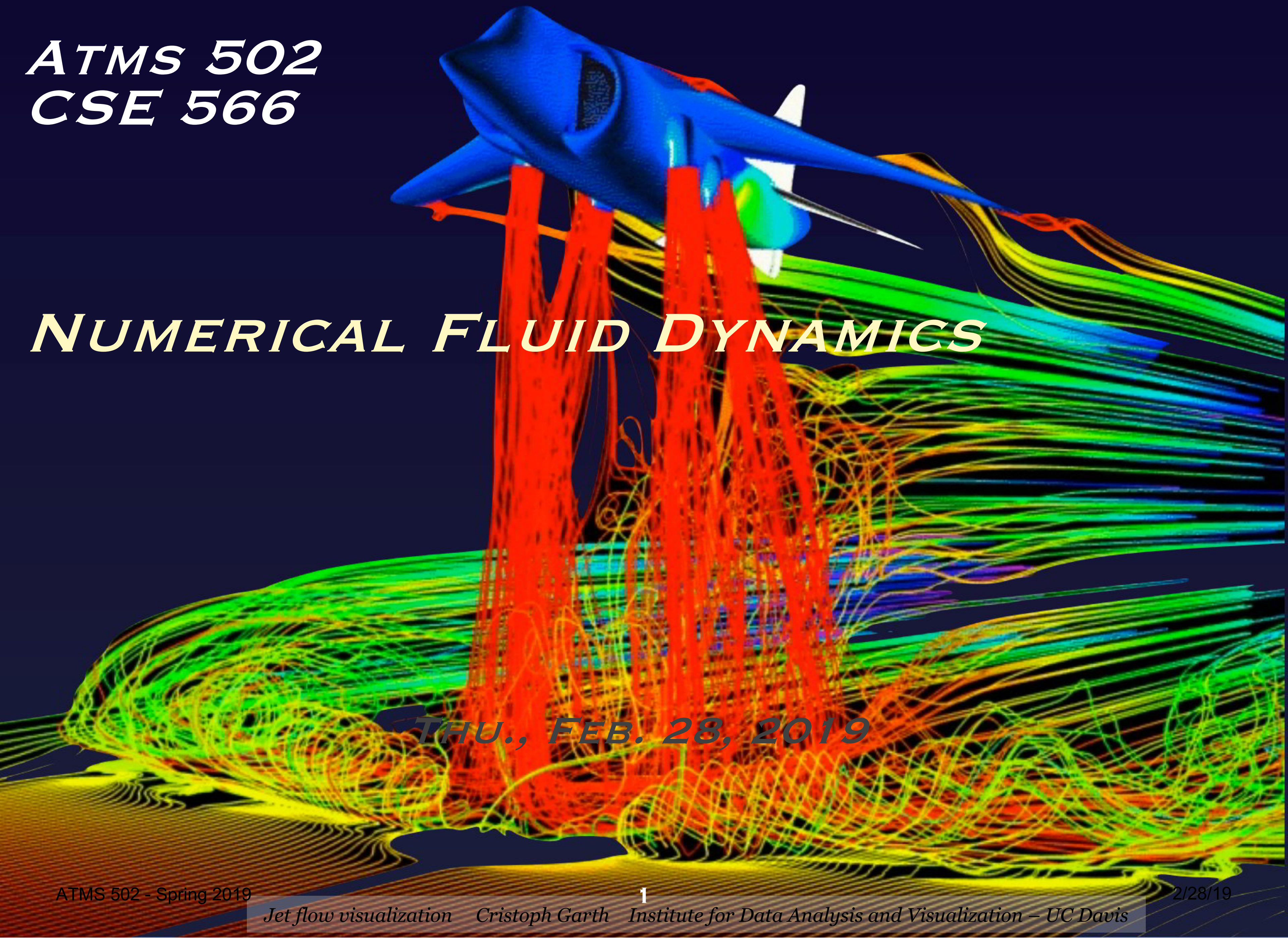
ATMS 502
CSE 566

NUMERICAL FLUID DYNAMICS

THU., FEB. 28, 2019



Jet flow visualization   Cristoph Garth   Institute for Data Analysis and Visualization – UC Davis

ATMS 502
CSE 566

Thursday,
28 February 2019

Class #14

- Pgm3 due Wed Mar. ***6***

Plan for Today

- 1) Time filtering
  - Damping leapfrog's computational mode
- 2) Grid refinement & clustering
  - Skamarock dissertation (notes: last class)
- 3) Program 4: *provided codes*
  - Placing/moving nest, & feedback
- 4) Resolution
  - Resolved/*permitted;* KE spectra method
- 5) Nesting, continued
  - Some questions & answers

Leapfrog stability - review

- We rewrote the 3-level scheme as 2-level:

$$\left.\begin{aligned}\tilde{u}^{n+1} &= \tilde{v}^{n} - \mu\tilde{u}^{n}(2i\sin\beta) \\ \tilde{v}^{n+1} &= \tilde{u}^{n}\end{aligned}\right\} \text{ so } \begin{pmatrix}\tilde{u}^{n+1} \\ \tilde{v}^{n+1}\end{pmatrix} = \begin{pmatrix}-2i\mu\sin\beta & 1 \\ 1 & 0\end{pmatrix}\begin{pmatrix}\tilde{u}^{n} \\ \tilde{v}^{n}\end{pmatrix}$$

- Write above as matrix, subtract l from diagonal, set determinant to zero. Characteristic equation:

$$\begin{vmatrix}-2i\mu\sin\beta - \lambda & 1 \\ 1 & 0-\lambda\end{vmatrix} = 0$$

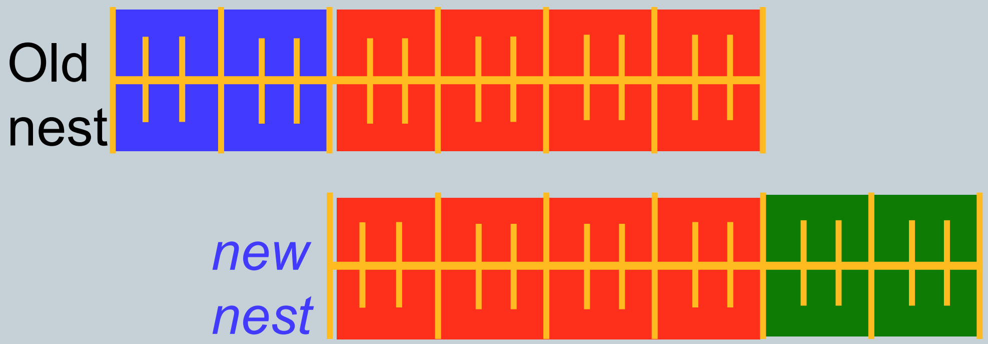
- Solve; 2 roots; physical and computational modes

$$\lambda = -i\mu\sin\beta \pm \sqrt{1-\mu^2\sin^2\beta} = -ip \pm \sqrt{1-p^2}$$

- As Δt and p ⇒0: ”+” root approaches 1, “-” root: -1
  - |λ|=-1 means amplitude varies as $(-1)^n$

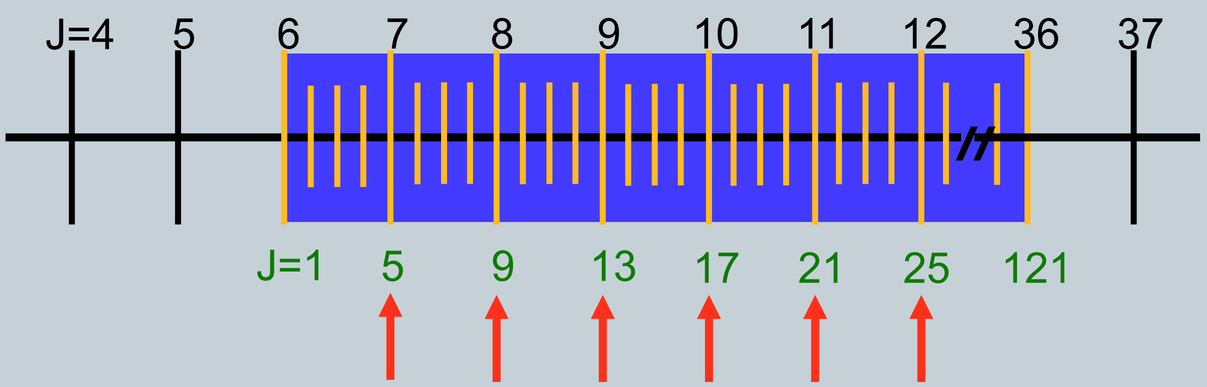
Review: 1-D Nesting

- *Interpolation: old vs. new nested grids*



THIS EXAMPLE:
3:1 NESTING

- *Feedback: copy nested ⇒ coarse*



THIS EXAMPLE:
4:1 NESTING

Time filtering

LEAPFROG TIME+SPACE DIFFERENCING

PHYSICAL AND COMPUTATIONAL MODES

UNDAMPED **COMPUTATIONAL MODES**

TIME FILTERING: *WHY*, AND *HOW TO?*

Time-filtered Leapfrog

- Advantages of the Leapfrog method:
  - Stable, 2nd order
  - Simple, & thus computationally cheap
    - but little computation for amount of communication
    - this is true for other schemes we have examined, too.
  - *No* amplitude error (if stable)
- Disadvantages:
  - Undamped computational mode
    - How to find the physical vs. computational mode
    - What is an *undamped* computational mode?
    - Odd/even solutions; may diverge
  - Dispersion, etc *(not unique to leapfrog)*

Time-filtered Leapfrog

- How to control the computational mode?
  - Periodically discard (n-1) time level data
  - Restart integration with a 2-level scheme
    - Common practice: **FTCS** scheme *(forward time, centered space)*
    - … but **FTCS** is unstable, and
    - … **FTCS** is 1st order (degrades accuracy)
    - Or: use Upstream or Lax-Wendroff
- Time smoothing
  - Remember computational mode: $\lambda \sim (-1)^n$
  - Smooth across (n-1, n, n+1) time levels

Time-filtered Leapfrog

- Time smoother for Leapfrog *(Asselin 1972)*
  - Instead of:

$$u_j^{n+1} = u_j^{n-1} - \mu\left(u_{j+1}^n - u_{j-1}^n\right)$$

  - Time smoothing:

$$u_j^{n+1} = \overline{u_j^{n-1}} - \mu\left(u_{j+1}^n - u_{j-1}^n\right) \quad \text{(Leapfrog step)}$$

$$\overline{u_j^n} = u_j^n + \varepsilon\left(u_j^{n+1} - 2u_j^n + \overline{u_j^{n-1}}\right) \quad \text{(Smoothing step)}$$

  - Stable if $\mu<(1-\varepsilon)$
    - So there is a more restrictive stability condition.

Time-filtered Leapfrog

- Time smoother for Leapfrog *(Asselin 1972)*

$$u_j^{n+1} = \overline{u_j^{n-1}} - \mu\left(u_{j+1}^n - u_{j-1}^n\right)$$ (Leapfrog step)

$$\overline{u_j^n} = u_j^n + \varepsilon\left(u_j^{n+1} - 2u_j^n + \overline{u_j^{n-1}}\right)$$ (Smoothing step)

- Sequence:
  - Have: (n-1, smoothed) and (n, unsmoothed)
  - Take leapfrog step to get (n+1, unsmoothed)
  - Use new (n+1, unsmoothed) to smooth u(n)
  - Ready for next step [smoothed u => u(n-1)]

NOTES – HANDED OUT LAST CLASS!

Grid refinement & Clustering

ADAPTIVE MESH REFINEMENT

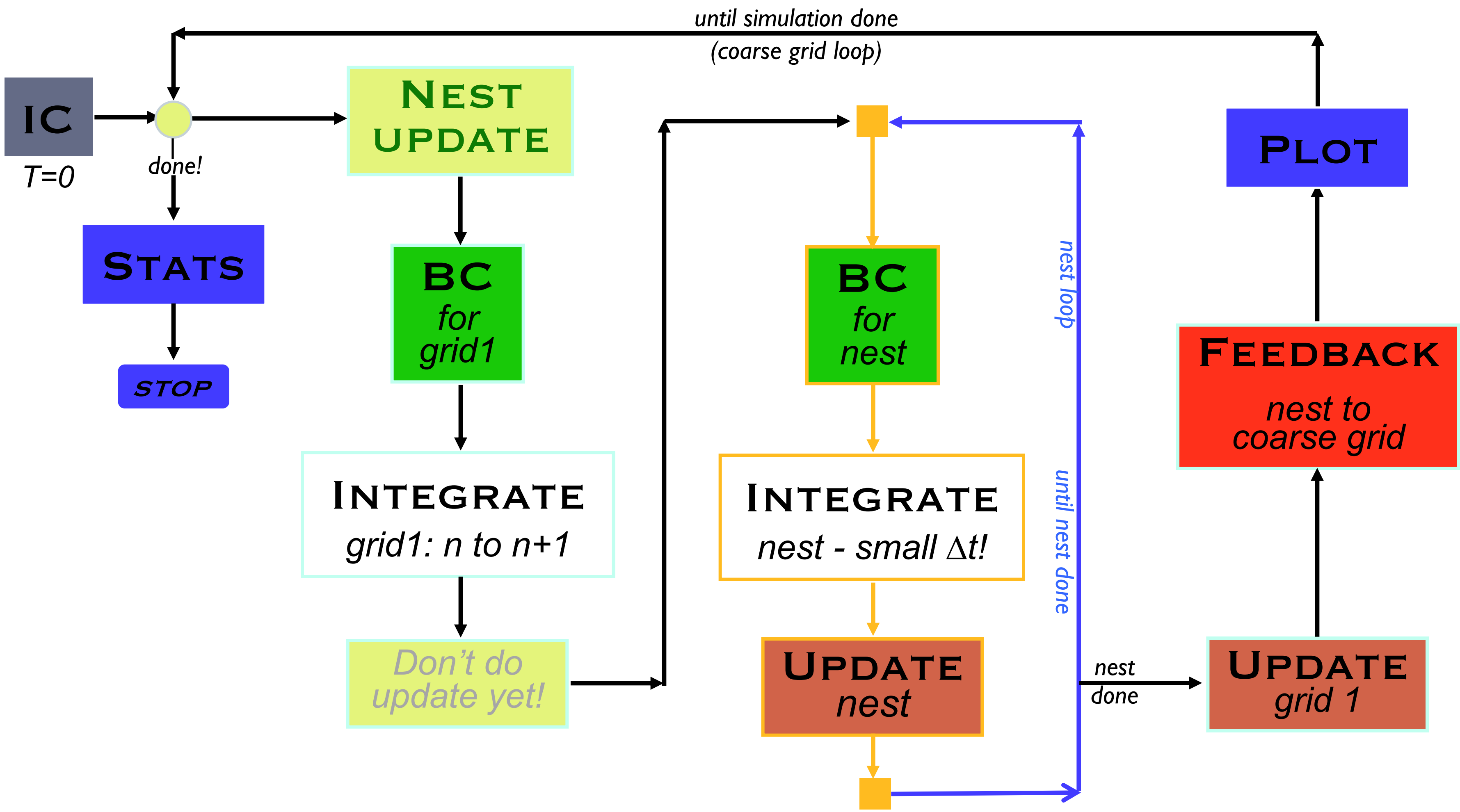
Reference pages for this section:

- C008 – Truncation error
- C009 – Resolution
- C010 – AMR / nesting
- C051 – Nesting: grid placement, movement

Program 4

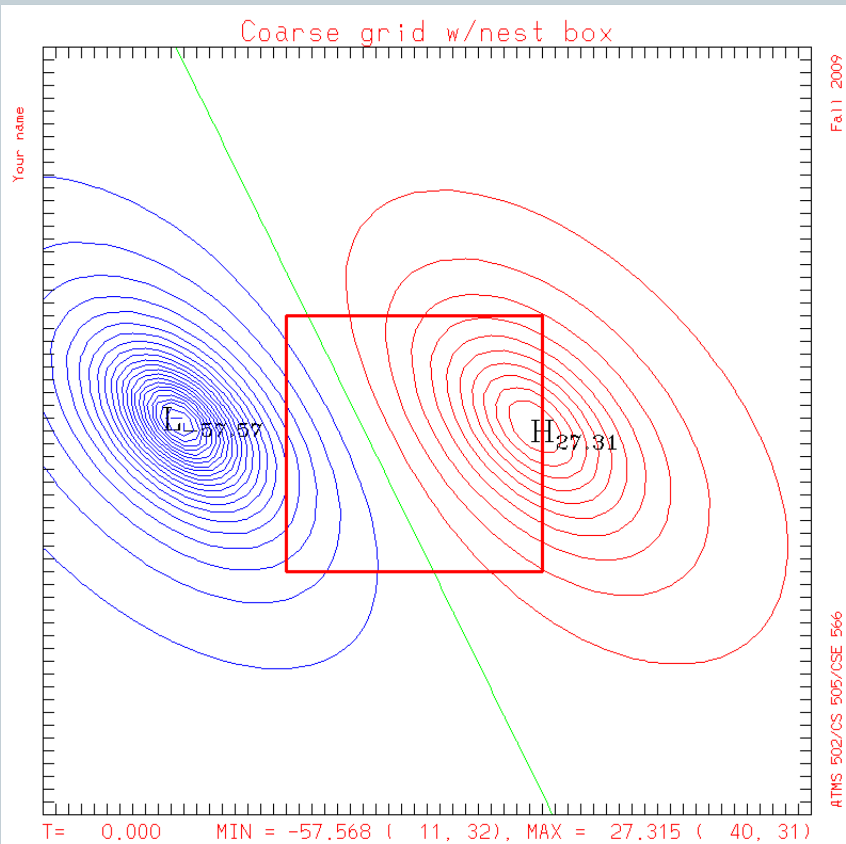
NESTING TOOLS PROVIDED TO YOU

Program #4: main routine

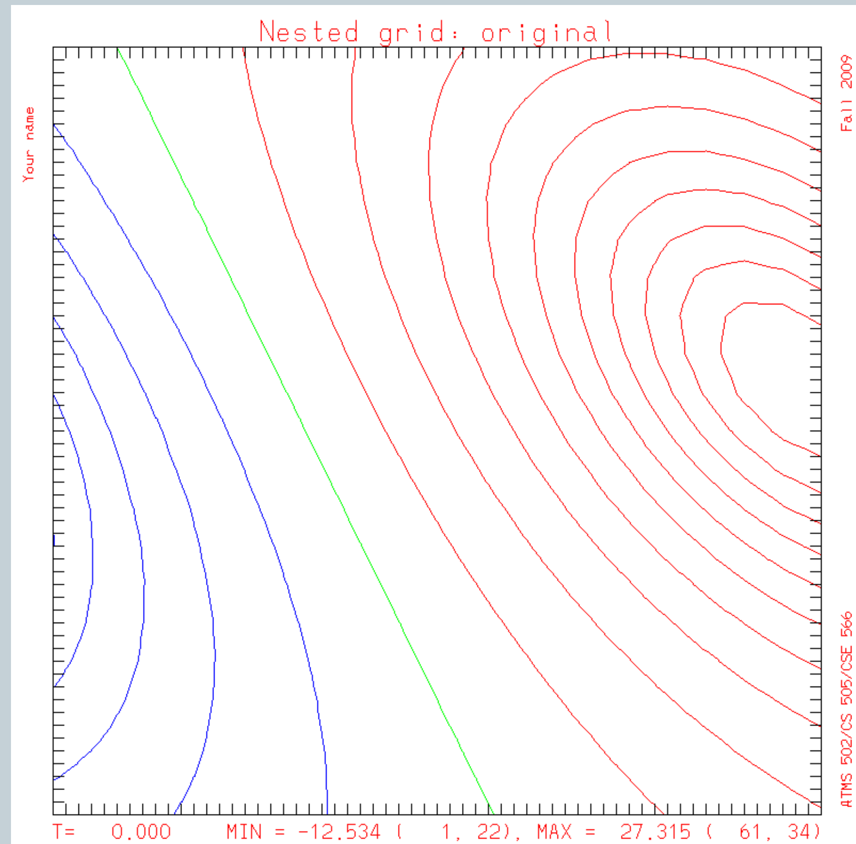


Nest BCs

- "Placing" the nest



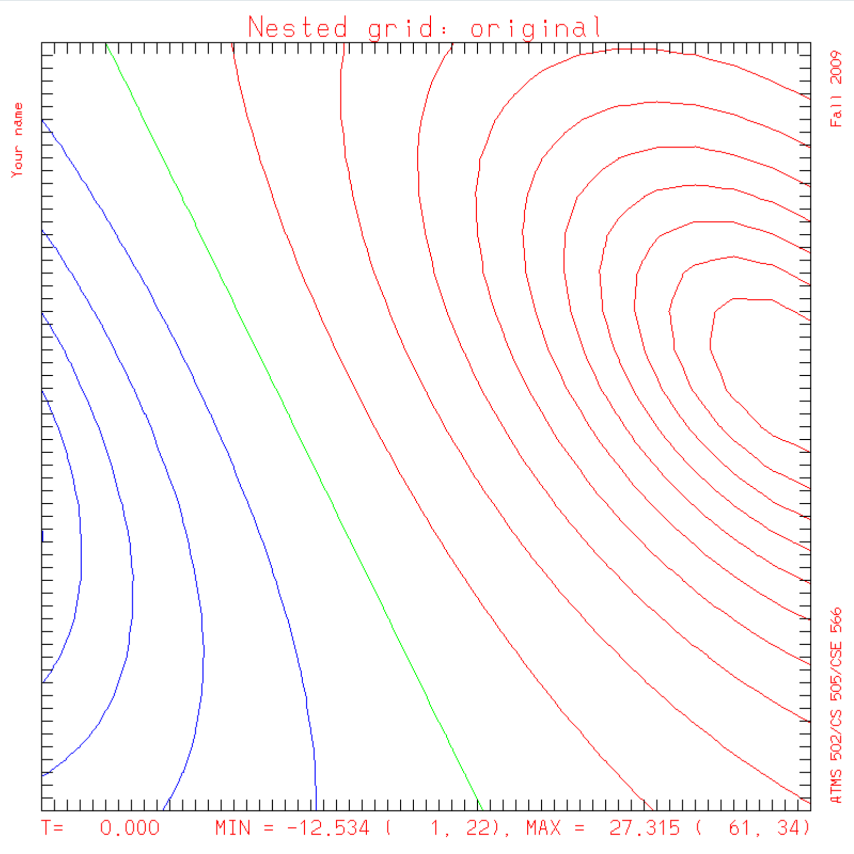
Coarse grid

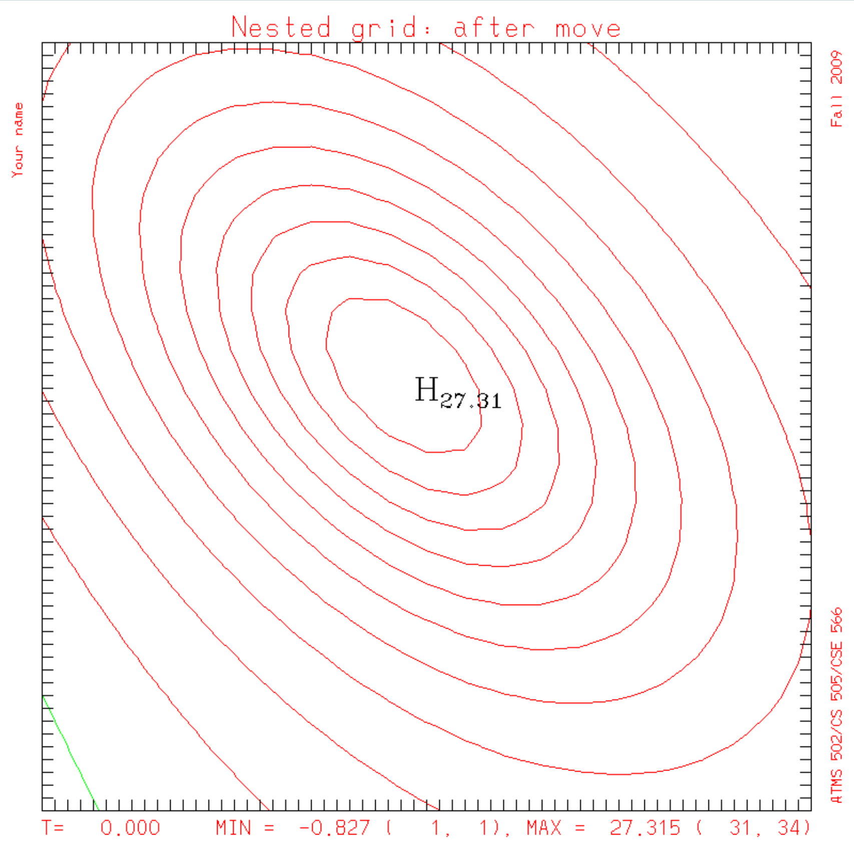


Nested grid

Nest BCs

- Moving the nest



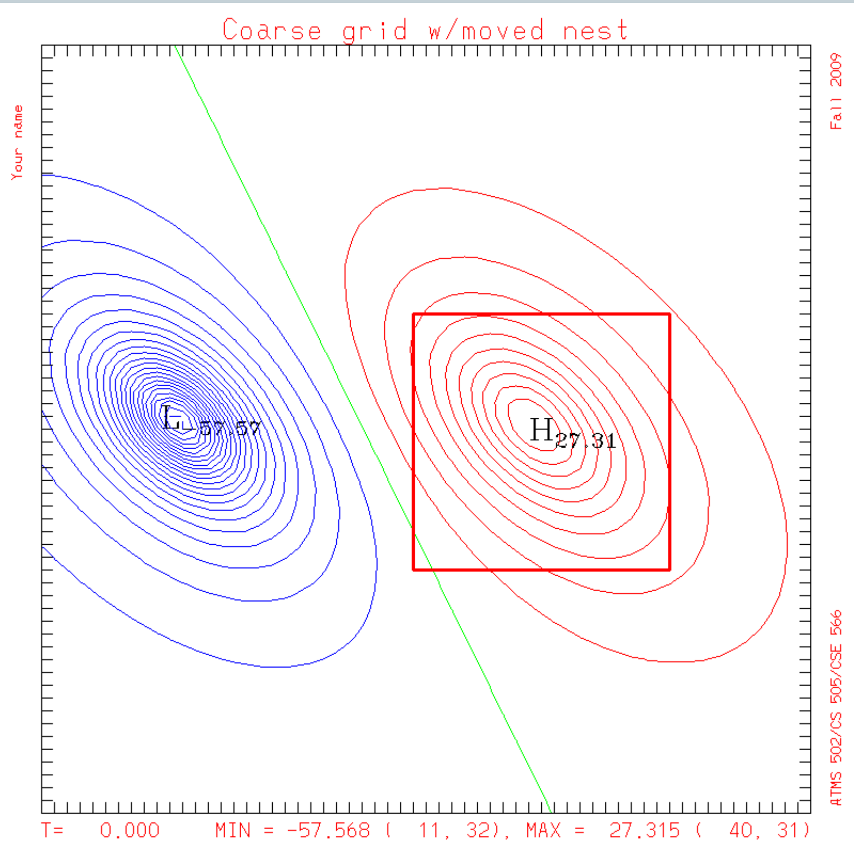


Nest

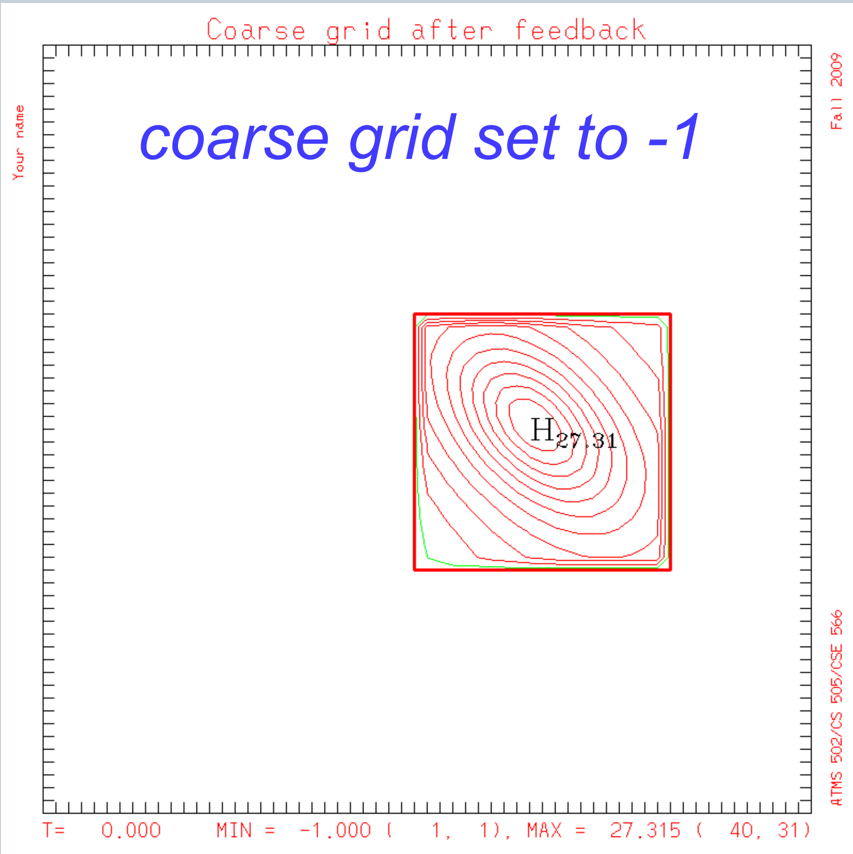
Nest moved

Nest BCs

- Testing feedback



Coarse w/moved nest



Test feedback to zero’d coarse

Resolution

Resolving vs. *permitting*

- Feature-*resolving* means what it says.
  - A model may instead have feature-*permitting* resolution
    - Means: those phenomena are "in" the model, e.g. via parameterizations and only in a very broad sense
  - Some models contain both *explicit* and *parameterized* physics
    - Explicit actually describes ~correct behavior – if really resolved
    - Parameterized reproduces bulk properties of the phenomena even though it is not resolved
    - Things get interesting in the *in-between* resolutions *("gray scales")*
- Liu and Moncrieff *(2007 Mon. Wea. Rev., p. 2866)*
  - Cloud-*permitting* runs "underperform" and exhibit greater sensitivity to parameterizations than the cloud-*resolving* models exhibit from their explicit physics.

Effective resolution

- Skamarock (2004):
  - Why move to higher resolution?
    - Typically this is done to resolve phenomena that are now marginally resolved, or unresolved (i.e. parameterized)
  - "Effective" resolution
    - There are known *kinetic energy spectra* profiles (see Skamarock Fig. 10 at right).
    - Models fail to reproduce these spectra at smaller scales. Note the dropoff at higher wavenumber (lower wavelengths)
    - He defines effective model resolution to be where the model spectra "decays"
      - WRF atmospheric model: 7Δx *(p. 3027)*

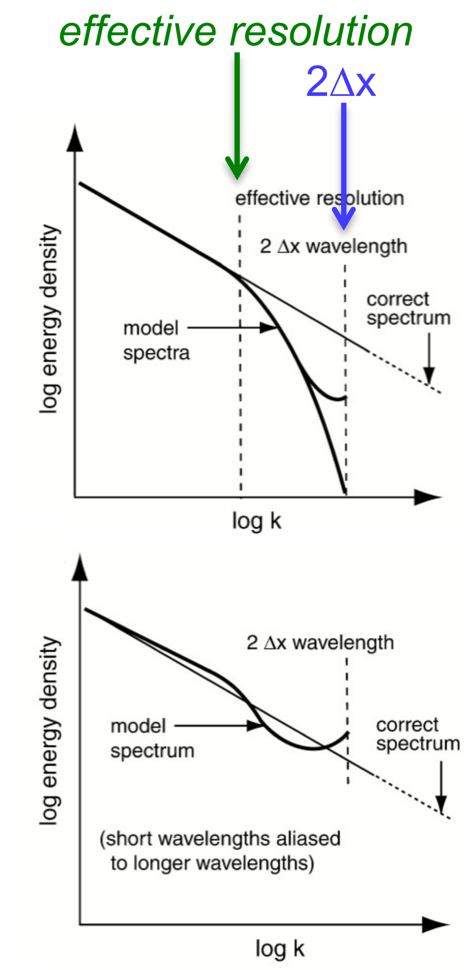
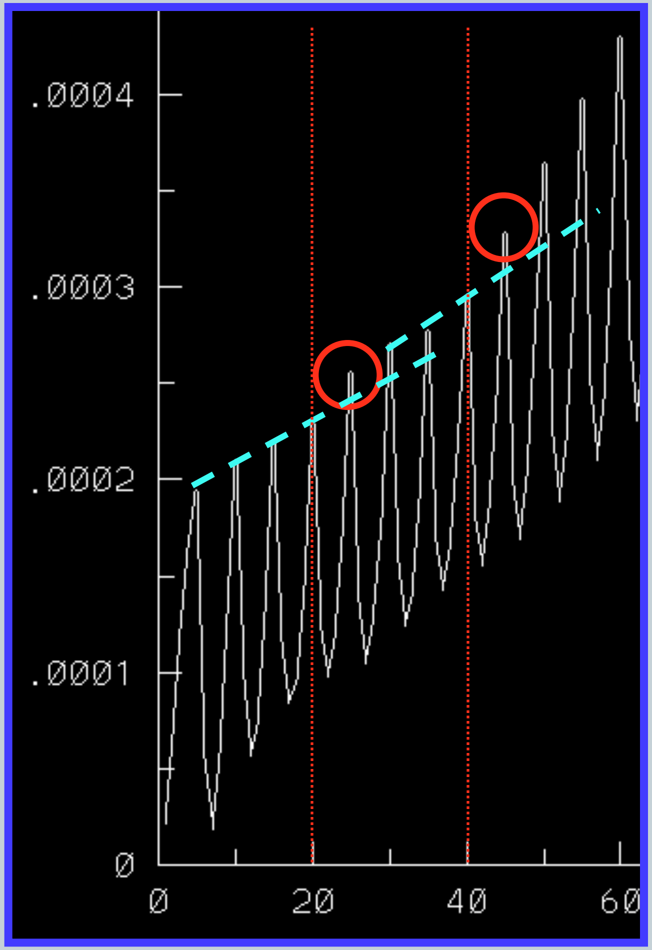


FIG. 10. Schematic depicting the possible behavior of spectral tails derived from model forecasts. Using the methodology outlined in the appendix to compute the spectra, limited-area models (including WRF) usually produce the slightly upturned tail shown at left.

Nesting

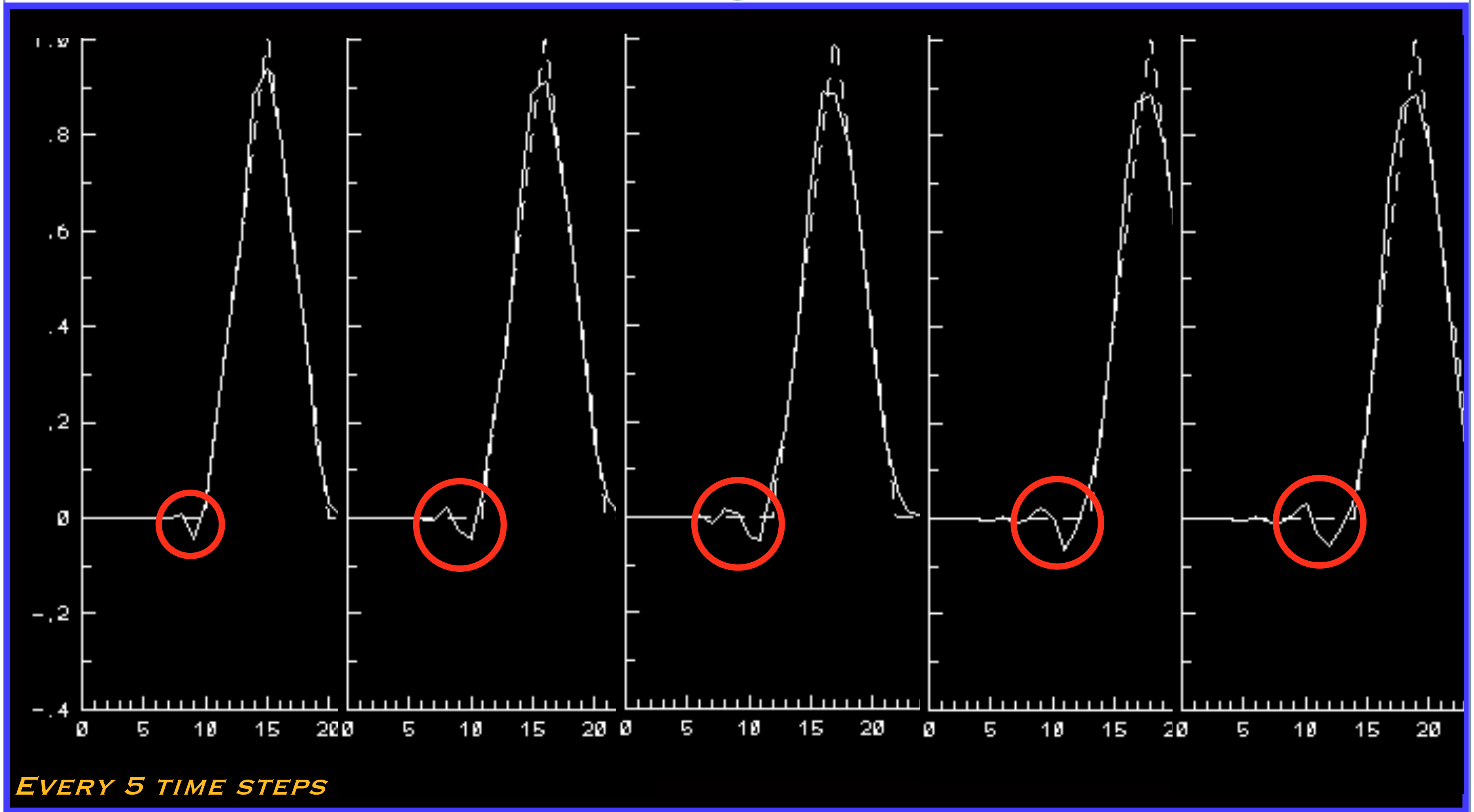
Question 1: oscillations

- Nest shock is superimposed on the high frequency oscillations seen here
- But:
  What causes oscillations?



- Total grid-1 error is plotted at each time step

No, oscillations aren't due to *this*.



Question 1: oscillations

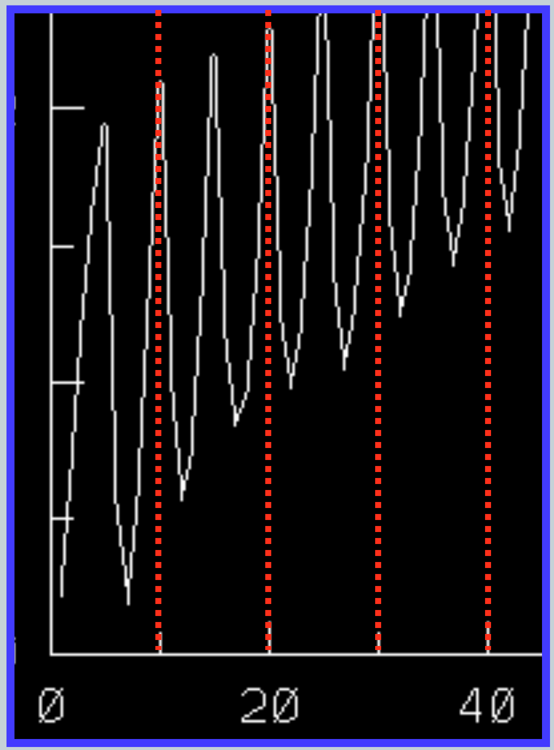
- How long does it take for features to move 1 Δx ???
  - *s=vt; time = distance/speed;* t=s/v
  - Total time to go distance Δx = $\Delta x/c$
  - # time steps to go Δx = $1/\nu$ *(why?)*

- So, every 5 time steps … *(for ν=0.2)*
  - The cone peak moves one grid distance.
  - What does this say about our "true" solution?

Question: oscillations

(1) What *are these oscillations*?

- *Every 5 time steps the (exact) solution moves one grid length.*
- *At right: 2 periods in 10 steps*

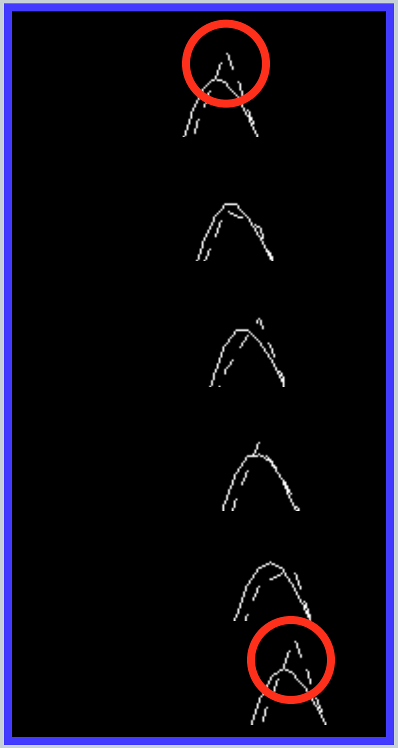


Question: oscillations

(1) What *are these oscillations*?

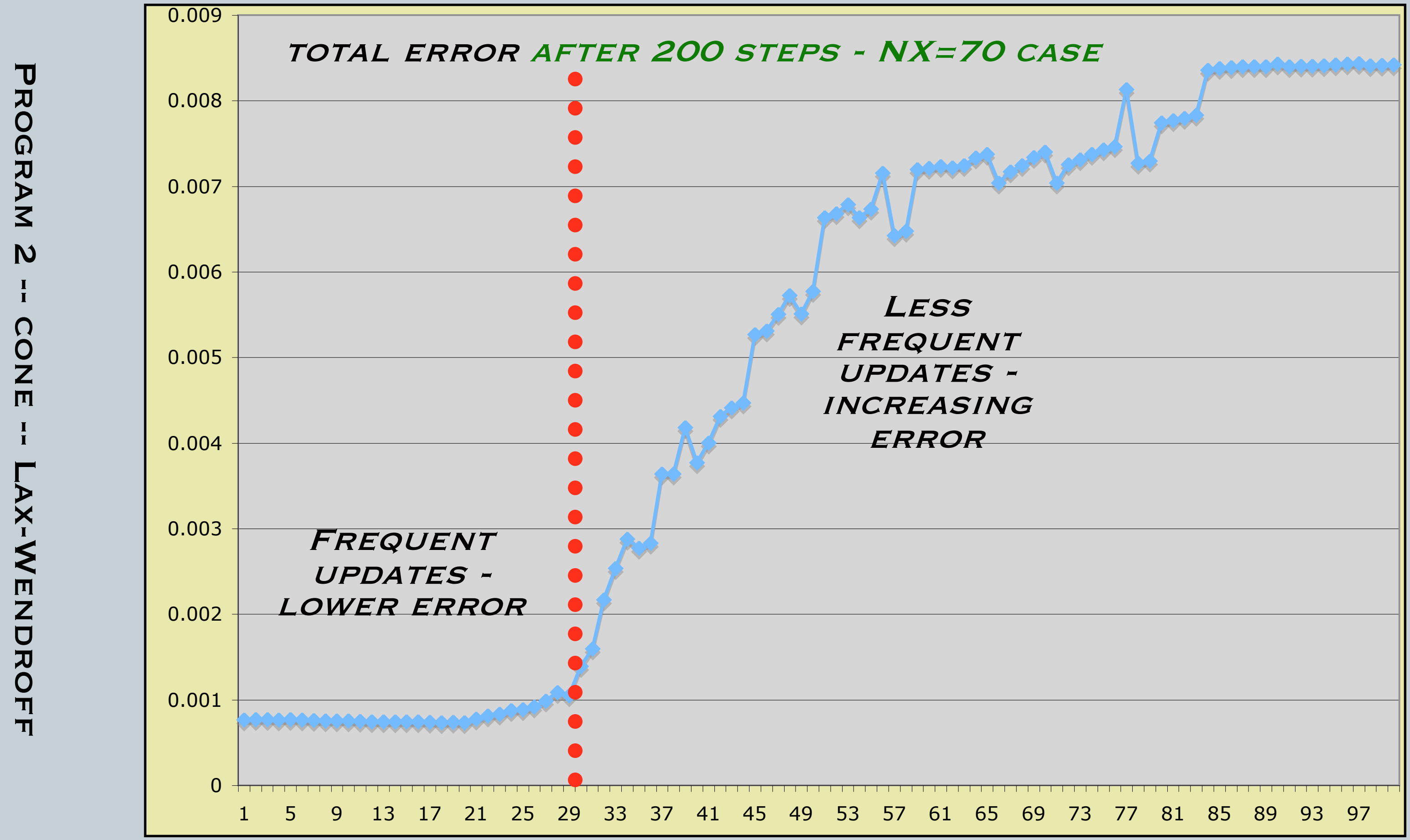
- *Every 5 time steps the (exact) solution moves one grid length.*
- *At right: 2 periods in 10 steps*

- *Projecting an exact solution on a finite grid results in errors - in the "true" solution !!*

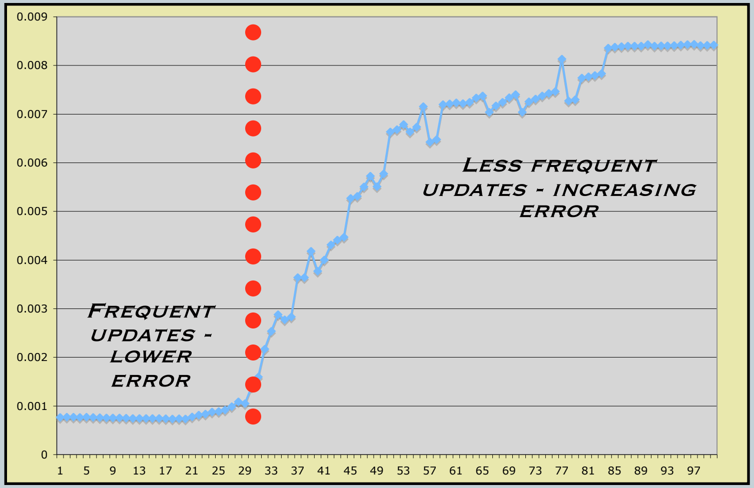


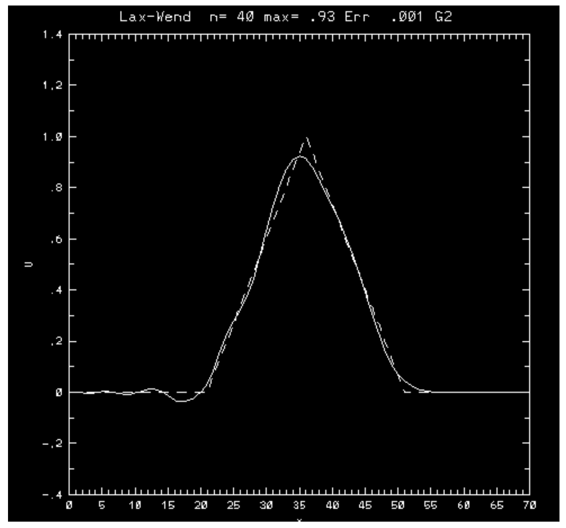
Question 2: update frequency

Below: error scores vs update frequency



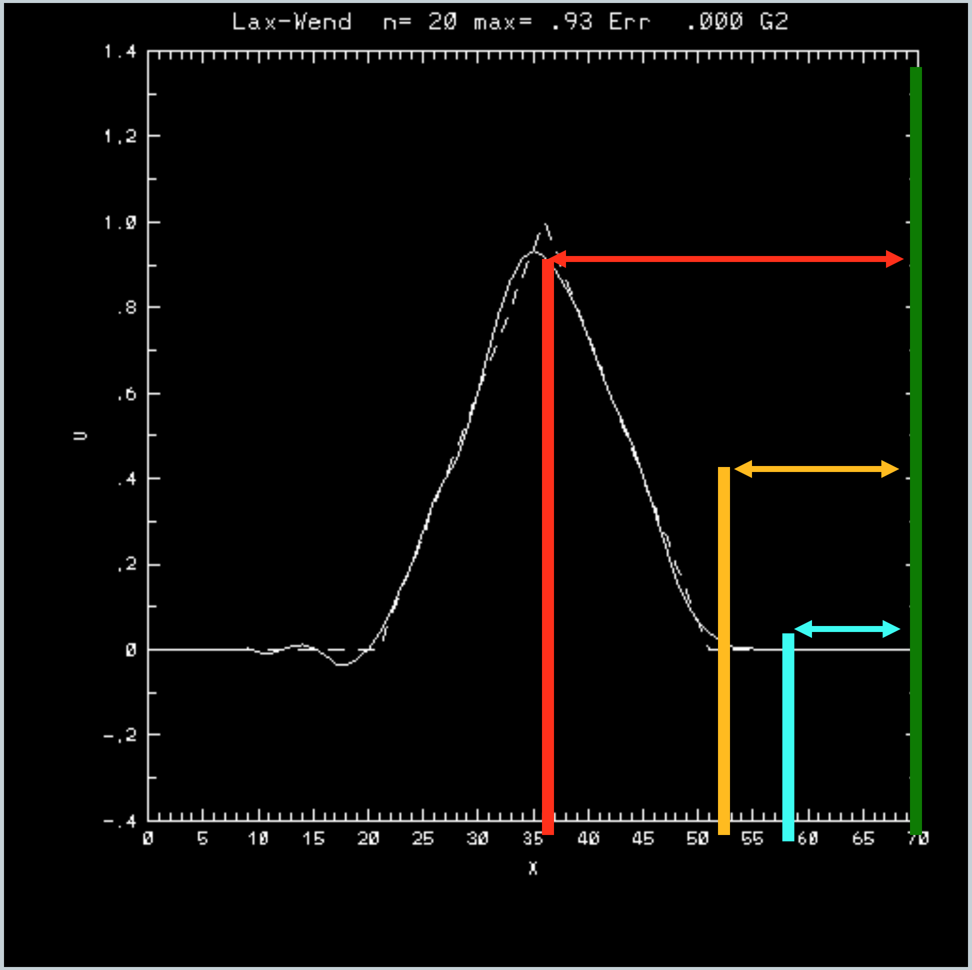
Question 2: update frequency





- Red dotted line is at ~30 (grid relocated every 30 steps).
- # steps for cone to move from center of nest to "center" of edge: *5*(24/2)=60*
- *What does this tell us?*

Question 2: update frequency



- In *60 steps* the middle of the cone is on the right nest domain edge.
- In *30 steps*, when the error starts rapidly rising, the *right edge* of the cone (and of trunc. error) hits the nest edge.
- In *20 steps*, the nest border is still ahead of the cone's leading edge.

The CFL condition

STABILITY - CONTINUED

PDE classification: Two kinds

$$Au_{xx} + Bu_{xy} + Cu_{yy} = F(x,y,u,u_x,u_y)$$

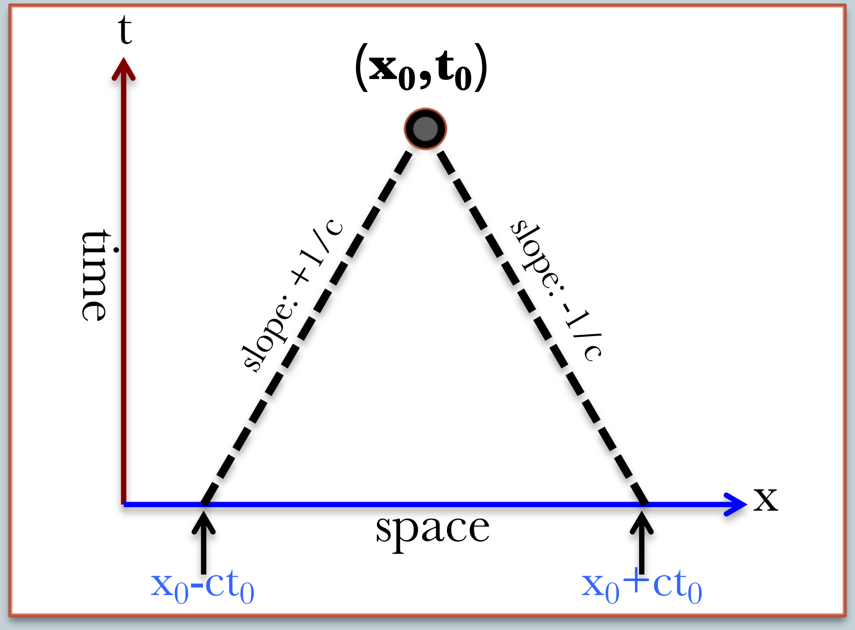
Details: ***F*** is a *linear function* of u, u_x, u_y; coefficients may only depend on x, y. Subscripts are derivatives!

- Is *elliptic* if $B^2-4AC < 0$ — $u_{xx} + u_{yy} = 0$ — Laplace's equation
- Is *parabolic* if $B^2-4AC = 0$ — $u_t = c^2 u_{xx}$ — Heat equation
- Is *hyperbolic* if $B^2-4AC > 0$ — $u_{tt} = c^2 u_{xx}$ — Wave equation
- B^2-4AC came from *characteristics*, or characteristic curves - curves of *information propagation*. The 2nd-order wave equation has the characteristics: $x \pm ct = \text{constant}$

- Initial value problems - the principal *computational* concern is the *stability* of the numerical algorithm/scheme.
  - PDE includes a time derivative!
- Boundary value problems – Involves solving solutions of large numbers of equations; *efficiency* is key concern.

Characteristics

- Hyperbolic case; *characteristic curves*
  - Example: 2-D wave equation
  - Two wave speeds: ±*c*
  - An initial value problem



Adapted from Figure 2-6 of Anderson et al., p. 23

- Anderson et al. textbook: *"A fundamental property of hyperbolic PDEs is the limited domain of dependence" [shown in figure at left]*
- Analytical solution at (x_0,t_0) depends *only* on data between the characteristic curves including *initial data* between (x_0-ct_0), (x_0+ct_0)

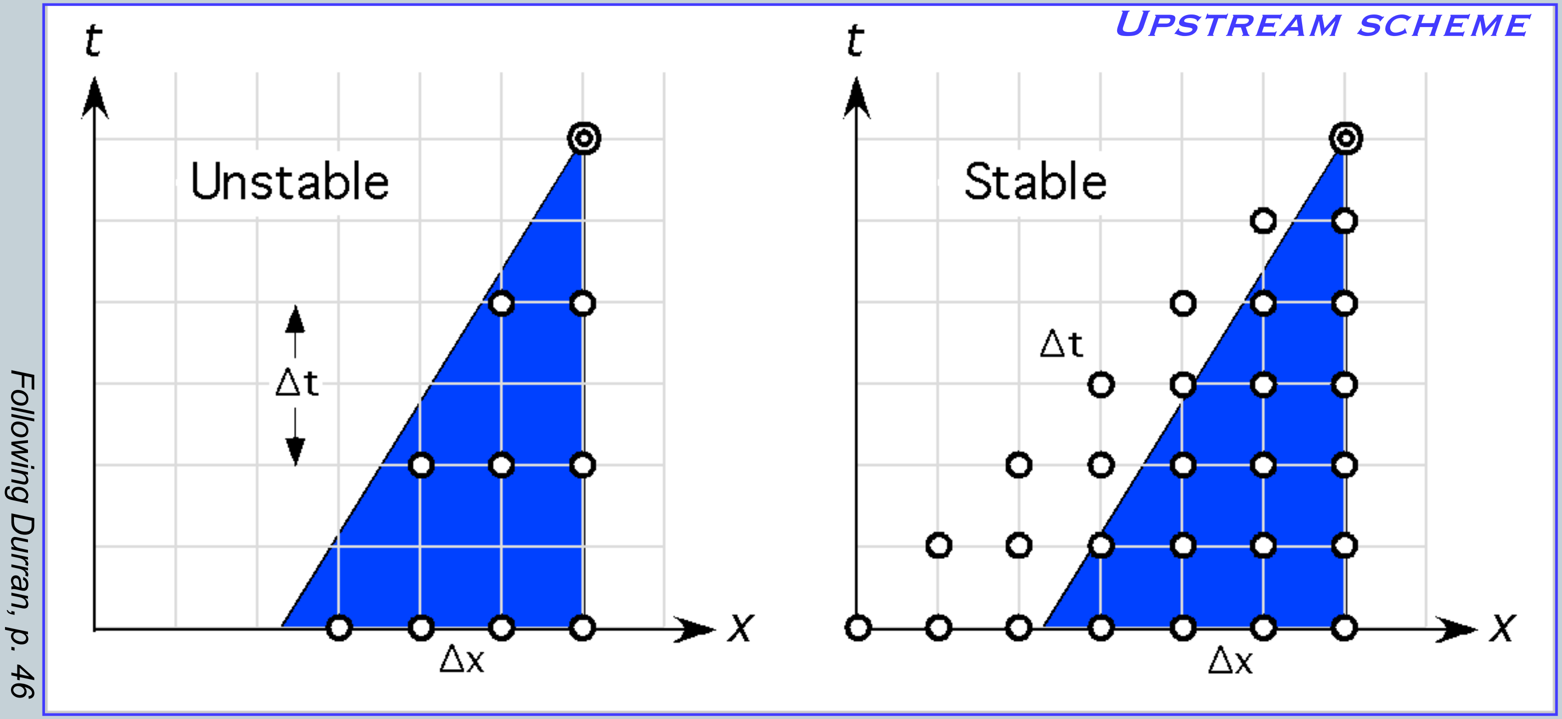
CFL

- The CFL (Courant-Freidrichs-Lewy) condition
  - "On the partial difference equations of mathematical physics" *
  - Based on the *data* that
    - determines the solution to the actual PDE
    - determines the solution in the numerical scheme
  - Requires that
    - the numerical domain of dependence must *include* *the* PDE domain of dependence.
- A *necessary but not sufficient* condition for stability.
  - The CFL criteria need not agree with the results of the Von Neumann analysis.

Courant, R., K. O. Fredrichs, and H. Lewy (1928), "Uber die Differenzengleichungen der Mathematischen Physik", Math. Ann, vol.100, p.32, 1928.

Domain of Dependence

- The CFL condition requires that the numerical domain of dependence include the PDE domain of dependence.



Unstable case: *true* domain of dependence extends outside of numerical one.
Stable case: Δt *halved*; numerical domain contains *true* domain of dependence.