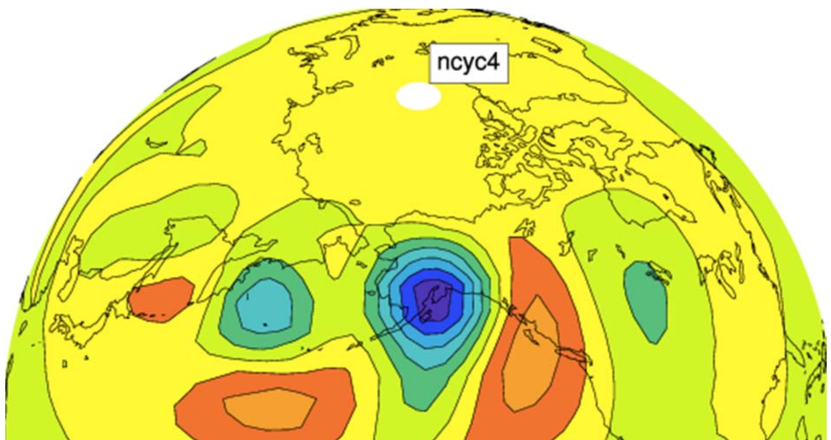
Reduction of Temporal Discretization Error in an Atmospheric General Circulation Model (AGCM)

Final Presentation



Apr. 30, 2013

Daisuke Hotta
hotta@umd.edu

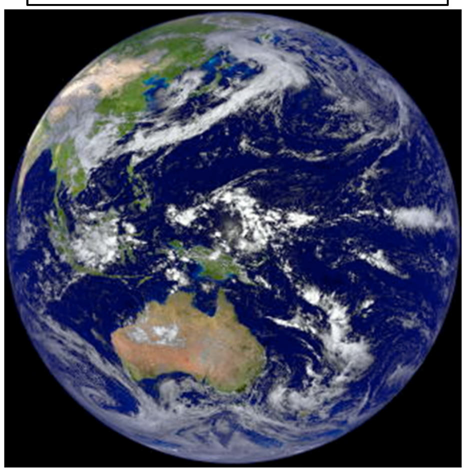
Advisor: Prof. Eugenia Kalnay
Dept. of Atmospheric and Oceanic Science,
University of Maryland, College Park
ekalnay@atmos.umd.edu

Outline

1. Introduction
2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. Code Validation
    7. Verification: Dynamical-core test
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual
4. Outcome/Delivarable
5. References

Numerical Weather Prediction (NWP): = Initial Value Problem of PDE

Atmospheric Phenomena → Governing Equations → Numerical Discretization → Solve ! (Simulate)



(Real Atmosphere)

(from JMA website)

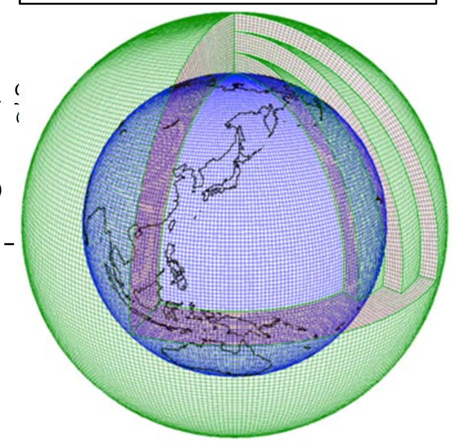
$$\frac{\partial \zeta}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial F_V}{\partial \lambda} - \frac{1}{a}\frac{\partial F_U}{\partial \mu}$$
$$- K_R(\sigma)(\zeta - [\zeta]) - K_v\left(\nabla_\sigma^4 - \frac{2^2}{a^2}\right)\zeta + \cdots$$

$$\frac{\partial D}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial F_U}{\partial \lambda} + \frac{1}{a}\frac{\partial F_V}{\partial \mu} - \nabla_\sigma^2(\Phi + R\bar{T}\pi + KE)$$
$$- K_R(\sigma)(D - [D]) - K_v\left(\nabla_\sigma^4 - \cdots\right.$$

$$\frac{\partial T}{\partial t} = - \frac{1}{a(1-\mu^2)}\frac{\partial UT'}{\partial \lambda} - \frac{1}{a}\frac{\partial VT'}{\partial \mu} + T'D$$
$$- \dot{\sigma}\frac{\partial T}{\partial \sigma} + \kappa T\left(\frac{\partial \pi}{\partial t} + v_H \cdot \nabla_\sigma \pi + \frac{\dot{\sigma}}{\sigma}\right)$$
$$- K_N(\sigma)(T - [T]) - K_h\left(\nabla_\sigma^4 - \frac{2^2}{a^2}\right)T + \left.\frac{dT}{dt}\right|_{forcing}$$

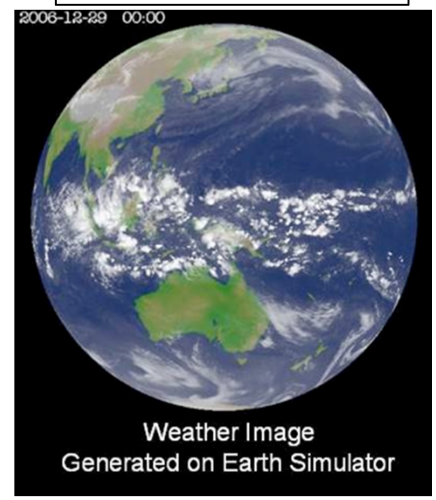
$$\frac{\partial \pi}{\partial t} + v_H \cdot \nabla_\sigma \pi = -D - \frac{\partial \dot{\sigma}}{\partial \sigma}$$

Hydrodynamic PDE



from http://www.jma.go.jp/jma/jma-eng/jma-center/nwp/nwp-top.htm

$O(10^9)$-dimensional ODE



Simulated Atmosphere

from http://iprc.soest.hawaii.edu/news/news_2009.php

AGCM: Atmospheric General Circulation Model
= a computer program which simulates the flow of global atmosphere by numerically integrating the governing fluid dynamical PDEs

Introduction: Motivation

Due to computational restrictions …

- most AGCMs adopt low-order time-integration schemes, such as

- Leap-frog with Robert-Asselin filter (1st order)
- Explicit Backward Euler (aka. Matsuno; 1st order)

- Often, Δt is taken as the largest value for which computational instability is suppressed,

- under the premise that temporal discretization errors are negligible compared to those associated with spatial discretization or Physical Parameterizations.

Introduction: Motivation

However …

- Spatial resolutions become finer and finer as the supercomputers become faster.
- Is the premise that time truncation errors are negligible justified ?
- If not, how can we alleviate such errors ?

→ Goal of the Project: Reduction of such model errors

Approaches :

Phase 1: Use a more accurate scheme with the same computational cost

Phase 2: Identify and parameterize the error, and reduce it using data assimilation

Phase I:
Implementation of Lorenz *N*-cycle
to SPEEDY model

Phase 1 : A Better integration scheme (Lorenz *N*-cycle)

Lorenz (1971) proposed an incredibly smart time-integration scheme which:

- requires only 1 function evaluation per step
- but yet (every *N* steps) it is of
  - (up to) 4^{th}-order accuracy (for nonlinear systems)
  - arbitrary order of accuracy (for linear systems)

However, this scheme seems to have remained forgotten.
No applications have been made to AGCMs.

➔ Apply Lorenz *N*-cycle to an AGCM (Phase 1)

Phase 1: Approach

- Implement Lorenz *N*-cycle to an existing AGCM
- Implement 4^{th} order Runge-Kutta (RK4) as well as a reference

- Compare the accuracy and efficiency of the newly introduced schemes with the original scheme
- Perform verification by the Jablonowski-Williamson (2006) Dynamical-core Tests

Phase 1: Algorithms

ODE to be solved:

$$\frac{du}{dt} = F(u)$$

Lorenz N-cycle	**(existing)** Leap-frog with Robert-Asselin Filter	4^{th} order Runge-Kutta
$G=0, w^1=1, w^k=\frac{N}{N-k},$ do $k=1,\ldots,N-1$ $G \to w^k F(u)+(1-w^k)G$ $u \to u+G\Delta t$ end do	do $k=1,\ldots,$ $u^{+1} \to u^{-1}+2\Delta t F(u^0)$ $u^0 \to u^0+\alpha(u^{+1}-2u^0+u^{-1})$ $u^{-1} \to u^0$ $u^0 \to u^{+1}$ end do	do $k=1,\ldots,$ $h^1 \to F(u)$ $h^2 \to F(u+\frac{\Delta t}{2}h^1)$ $h^3 \to F(u+\frac{\Delta t}{2}h^2)$ $h^4 \to F(u+\Delta t h^3)$ $u += \frac{\Delta t}{6}(h^1+2h^2+2h^3+h^4)$ end do
Memory consumption: **2 x** dim{model state}	Memory consumption: **2 x** dim{model state}	Memory consumption: **4 x** dim{model state}
F-evaluation: **1** per time step	*F*-evaluation: **1** per time step	*F*-evaluation: **4** per time step
accuracy: (N <= 4) ***O((NΔt)^N)*** (every N steps) ***O(NΔt)*** (in between)	accuracy: ***O(Δt)***	accuracy: ***O(Δt^4)***

Outline

1. Introduction
2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. **Model description**
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. Code Validation
    7. Verification: Dynamical-core test
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual
4. Outcome/Delivarable
5. References

AGCM: SPEEDY model

- A fast AGCM with simplified physical parameterizations
- Developed in ICTP (Italy) by Drs. F. Molteni and F. Kucharski
- Horizontal Discretization:

  Spectral Representation with Spherical Harmonics

  truncated at total wavenumber 30 (T30) ≈ 400km mesh
- Vertical Discretization:

  8-layers Finite Difference on σ-coordinate
- Temporal Discretization:

  Leap-Frog scheme with Robert-Asselin Filter

  (1st order Forward Euler for the physical parameterizations)

The equations solved: the "primitive" equation system (PDEs) on a spherical geometry + parametrized processes

Dynamical Core

$$\frac{\partial \zeta}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial F_V}{\partial \lambda} - \frac{1}{a}\frac{\partial F_U}{\partial \mu} - K_R(\sigma)(\zeta - [\zeta]) - K_v\left(\nabla_\sigma^4 - \frac{2^2}{a^2}\right)\zeta + \left.\frac{d\zeta}{dt}\right|_{forcing}$$

$$\frac{\partial D}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial F_U}{\partial \lambda} + \frac{1}{a}\frac{\partial F_V}{\partial \mu} - \nabla_\sigma^2(\Phi + R\bar{T}\pi + KE) - K_R(\sigma)(D - [D]) - K_v\left(\nabla_\sigma^4 - \frac{2^2}{a^2}\right)D$$

$$\frac{\partial T}{\partial t} = -\frac{1}{a(1-\mu^2)}\frac{\partial UT'}{\partial \lambda} - \frac{1}{a}\frac{\partial VT'}{\partial \mu} + T'D - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \kappa T\left(\frac{\partial \pi}{\partial t} + v_H \cdot \nabla_\sigma \pi + \frac{\dot{\sigma}}{\sigma}\right) - K_N(\sigma)(T - [T]) - K_h\left(\nabla_\sigma^4 - \frac{2^2}{a^2}\right)T + \left.\frac{dT}{dt}\right|_{forcing}$$

$$\frac{\partial \pi}{\partial t} + v_H \cdot \nabla_\sigma \pi = -D - \frac{\partial \dot{\sigma}}{\partial \sigma}$$

$$\zeta \equiv \frac{1}{a(1-\mu^2)}\frac{\partial V}{\partial \lambda} - \frac{1}{a}\frac{\partial U}{\partial \mu}$$

$$D \equiv \frac{1}{a(1-\mu^2)}\frac{\partial U}{\partial \lambda} + \frac{1}{a}\frac{V}{\mu}$$

$$F_U \equiv (\zeta + f)V - \dot{\sigma}\frac{\partial U}{\partial \sigma} - \frac{RT'}{a}\frac{\partial \pi}{\partial \lambda}$$

$$F_V \equiv -(\zeta + f)U - \dot{\sigma}\frac{\partial V}{\partial \sigma} - \frac{RT'}{a}(1-\mu^2)\frac{\partial \pi}{\partial \mu}$$

$$KE \equiv \frac{U^2 + V^2}{2(1-\mu^2)}$$

$$v_H \cdot \nabla \equiv \frac{u}{a\cos\phi}\left(\frac{\partial}{\partial \lambda}\right)_\sigma + \frac{v}{a}\left(\frac{\partial}{\partial \phi}\right)_\sigma = \frac{U}{a(1-\mu^2)}\left(\frac{\partial}{\partial \lambda}\right)_\sigma + \frac{V}{a}\left(\frac{\partial}{\partial \mu}\right)_\sigma$$

$$\nabla_\sigma^2 \equiv \frac{1}{a^2(1-\mu^2)}\frac{\partial^2}{\partial \lambda^2} + \frac{1}{a^2}\frac{\partial}{\partial \mu}\left[(1-\mu^2)\frac{\partial}{\partial \mu}\right].$$

$$\theta \equiv T(p/p_{00})^{-\kappa}$$

$$\kappa \equiv R/C_p$$

$$\Phi \equiv gz = -\int_1^\sigma \frac{RT}{\sigma}d\sigma$$

$$\pi \equiv \ln p_s$$

$$\dot{\sigma} \equiv \frac{d\sigma}{dt}$$

$$\mu \equiv \sin\phi$$

$$U \equiv u\cos\phi$$

$$V \equiv v\cos\phi$$

$$T \equiv \bar{T}(\sigma) + T'$$

$$\dot{\sigma} = -\sigma\frac{\partial \pi}{\partial t} - \int_0^\sigma D\,d\sigma - \int_0^\sigma v_H \cdot \nabla_\sigma \pi d\sigma,$$

Sub-grid Parametrizations

$${}^uF^m_{N-h} = {}^dF^m_{N-h} = F^*$$

$${}^uF^Q_{N-h} = F^* \cdot Q^{sat}_N \;;\; {}^dF^Q_{N-h} = F^* \cdot Q_{N-h}$$

$${}^uF^{SE}_{N-h} = F^* \cdot SE_N \;;\; {}^dF^{SE}_{N-h} = F^* \cdot SE_{N-h}$$

$$\left(\frac{\partial Q_N}{\partial t}\right)_{cnv} = -\frac{g\,F^*\,(Q^{sat}_N - Q_{N-h})}{\Delta p_N} = -\frac{Q_N - Q_{thr}}{\tau_{cnv}}$$

$$F^* = \frac{\Delta p_N}{g\,\tau_{cnv}}\frac{Q_N - Q_{thr}}{Q^{sat}_N - Q_{N-h}}$$

$$E^m_k = \epsilon(\sigma_k)\,F^m_{k+h}$$

$$F^m_{k-h} = F^m_{k+h} + E^m_k$$

$${}^uF^Q_{k-h} = {}^uF^Q_{k+h} + E^m_k\,Q_k \;;\; {}^dF^Q_{k-h} = F^m_{k-h}$$

$${}^uF^{SE}_{k-h} = {}^uF^{SE}_{k+h} + E^m_k\,SE_k \;;\; {}^dF^{SE}_{k-h} = F^m_{k-h}$$

$$P_{cnv} = {}^uF^Q_{k0+h} - F^m_{k0+h}\,Q^{sat}_{k0+h}$$

$$\Delta F^Q_{k0} = {}^uF^Q_{k0+h} - {}^dF^Q_{k0+h} - P_{cnv}$$

$$\Delta F^{SE}_{k0} = {}^uF^{SE}_{k0+h} - {}^dF^{SE}_{k0+h} + L_c\,P_{cnv}$$

$$\left(\frac{\partial Q_k}{\partial t}\right)_{lsc} = -\frac{Q_k - RH(\sigma_k)_{lsc}\,Q^{sat}_k}{\tau_{lsc}}$$

$$\left(\frac{\partial T_k}{\partial t}\right)_{lsc} = -\frac{L_c}{c_p}\left(\frac{\partial Q_k}{\partial t}\right)_{lsc}$$

$$P_{lsc} = -\frac{1}{g}\sum_{k=2}^{N}\Delta p_k\left(\frac{\partial Q_k}{\partial t}\right)_{lsc}$$

$$\tau^{LR}_{k,1} = exp\,[-(\alpha^{LR}_{win} + \alpha^{LR}_{cl}\,CLC)\,\Delta p'_k]$$

$$\tau^{LR}_{k,2} = exp\,(-\alpha^{LR}_{CO_2}\,\Delta p'_k)$$

$$\tau^{LR}_{k,3} = exp\,(-\alpha^{LR}_{wv1}\,Q_k\,\Delta p'_k)$$

$$\tau^{LR}_{k,4} = exp\,(-\alpha^{LR}_{wv2}\,Q_k\,\Delta p'_k)$$

$${}^dF^{LR}_{k+h} = {}^dF^{LR}_{k-h}\,\tau^{LR}_k + (1 - \tau^{LR}_k)\,{}^dB_k$$

$${}^uF^{LR}_{k-h} = {}^uF^{LR}_{k+h}\,\tau^{LR}_k + (1 - \tau^{LR}_k)\,{}^uB_k$$

$${}^uB_k = f_b(T_k)\,\sigma_{SB}\,[T^4_k + w^{LR}_k\,(T^4_{k-h} - T^4_k$$

$$U_{sa} = f_{wind}\,U_N \;;\; V_{sa} = f$$

$$|V_0| = (U^2_{sa} + V^2_{sa} + V^2_{gust}$$

$$\alpha_{sw} = \frac{D_{top}\,W_{top} + f_{veg}\,D_{r}}{D_{top}\,W_{cap} + }$$

$${}^uF^U_{ls} = \tau^U_{ls} = -$$

$${}^uF^V_{ls} = \tau^V_{ls} = -$$

$${}^uF^{SE}_{ls} = SHF_{ls} = \rho_{sa}$$

$${}^uF^Q_{ls} = E_{ls} = \rho_{sa}\,C^H_l\,|V_0|\;max\,[$$

$${}^uF^{SE}_{ss} = SHF_{ss} = \rho_{sa}$$

$${}^uF^Q_{ss} = E_{ss} = \rho_{sa}\,C^H_s\,|V_0|\;ma$$

$$F^{SR}_s + F^{LR}_s + SHF_{ls} + E_l$$

$$d_s c_s \frac{\partial T_s}{\partial t} = GHF_{ls} - d_s c_s \tau_s^{-1}$$

$$\Delta T(t+1) = \frac{\tau}{\tau + \delta t}\Delta T(t)$$

⋮

Outline

Stiffness Problem

- The “Primitive” Equations = Stiff system :
- Fast but insignificant modes (= gravity waves) are superposed on the slow but meteorologically meaningful modes (= Rossby waves)
- Due to the CFL condition for the fast modes, we need very small timestepping Δt

→ Semi-implicit method (Robert, 1969)

Semi-Implicit for Leapfrog (1) Formulation

ODE to be solved: $$\frac{du}{dt} = \underbrace{F^E(u)}_{\text{Slow modes}} + \underbrace{L^I u}_{\text{Fast modes assumed Linear}}$$

Solve this discretized equation

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = \underbrace{F^E(u^n)}_{\text{Explicit}} + \underbrace{L^I \left(\alpha u^{n+1} + (1-\alpha)u^{n-1}\right)}_{\text{Implicit}}$$

Semi-Implicit for Leapfrog (2)
How to solve

Define: $$\delta u = \frac{u^{n+1} - u^{n-1}}{2\Delta t}$$

Substitute $u^{n+1} = u^{n-1} + 2\Delta t \delta u$
to the discretized equation,

$$\begin{aligned} \delta u &= F^E(u^n) + L^I u^{n-1} + 2\alpha \Delta t L^I \delta u \\ \Leftrightarrow \delta u &= (I - 2\alpha \Delta t L^I)^{-1} (F^E(u^n) + L^I u^{n-1}) \end{aligned}$$

Once you get δu, you can integrate
the equation by

$$u^{n+1} = u^{n-1} + 2\Delta t \delta u$$

Outline

1. Introduction
2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. **Semi-implicit Lorenz *N*-cycle**
    6. Code Validation
    7. Verification: Dynamical-core test
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual
4. Outcome/Delivarable
5. References

Semi-implicit Lorenz *N*-cycle

Explicit (original)

$$
\begin{aligned}
&\textbf{do } k = 0, \ldots \\
&\quad w \leftarrow w^{\mathrm{mod}(k,N)} \\
&\quad G \leftarrow wF(u) + (1-w)G \\
&\quad u \leftarrow u + G\Delta t \\
&\textbf{end do}
\end{aligned}
$$

Semi-implicit

$$
\begin{aligned}
&\textbf{do } k = 0, \ldots \\
&\quad w \leftarrow w^{\mathrm{mod}(k,N)} \\
&\quad G \leftarrow wF^{E}(u) + (1-w)G \\
&\quad \delta u = (I - \alpha \Delta t L^{I})^{-1}(G + L^{I}u) \\
&\quad u \leftarrow u + \Delta t \delta u \\
&\textbf{end do}
\end{aligned}
$$

Accuracy(Consistency) & Stability Analysis: Method

- Following Durran (1991, 1999; *MWR*) and Williamson (2011; *MWR*), apply semi-implicit modification to the second term of the equation:

$$\frac{du}{dt} = i\omega_L u + i\omega_H u$$

- Examine the modulus of Amplification factor *A*.
- If |*A*| < 1, the scheme is stable.
- Range of interest:

$$\omega_L \Delta t < 0.5, \quad \omega_H \Delta t > 2$$

i.e., CFL condition is met for low-frequency part but is violated for high-frequency part.

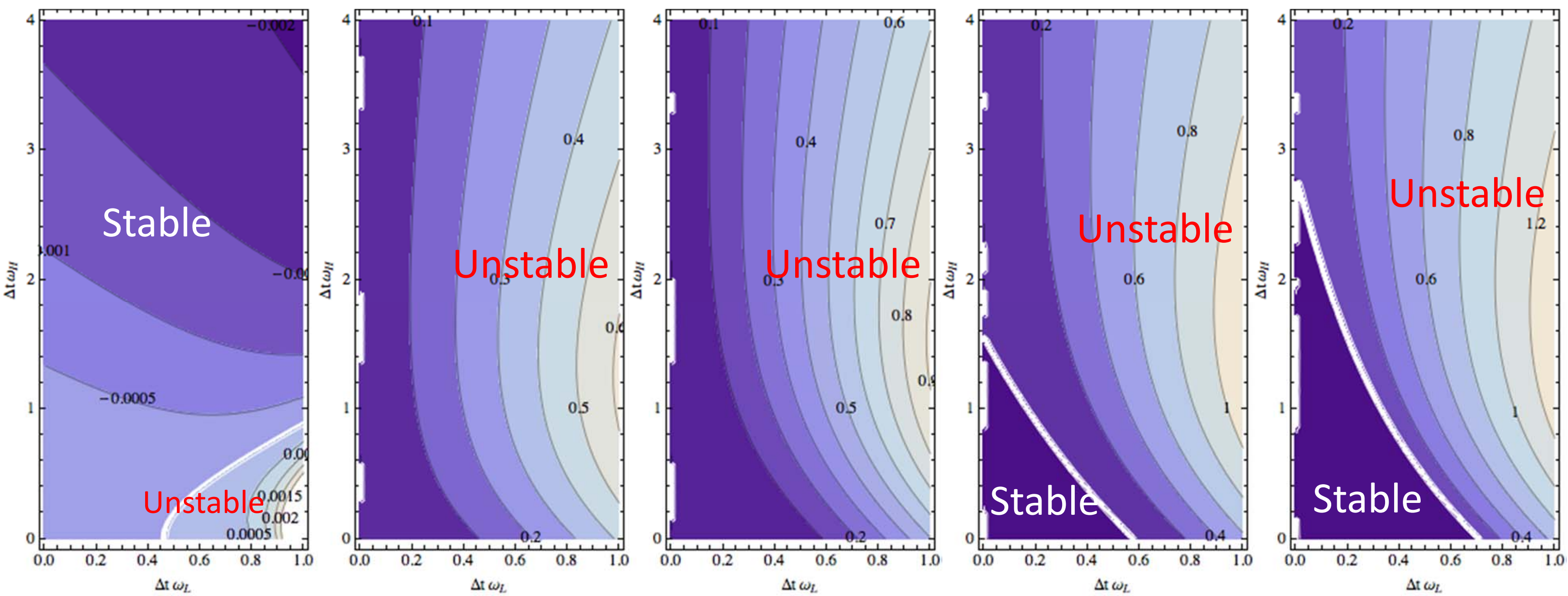
Accuracy(Consistency)

- Truncation Error:

$$\frac{u^N - u^{\mathrm{Exact}}}{u^0} = \frac{1}{2N}(1-2\alpha)\omega_H(\omega_H+\omega_L)(N\Delta t)^2 + O(\Delta t^3)$$

- By taking $\alpha=1/2$ (Crank-Nicolson), the scheme becomes 2nd-order

Plots of |*A*|-1 (α=1/2: Crank-Nicolson)



Leapfrog with R/A filter: Stable almost everywhere

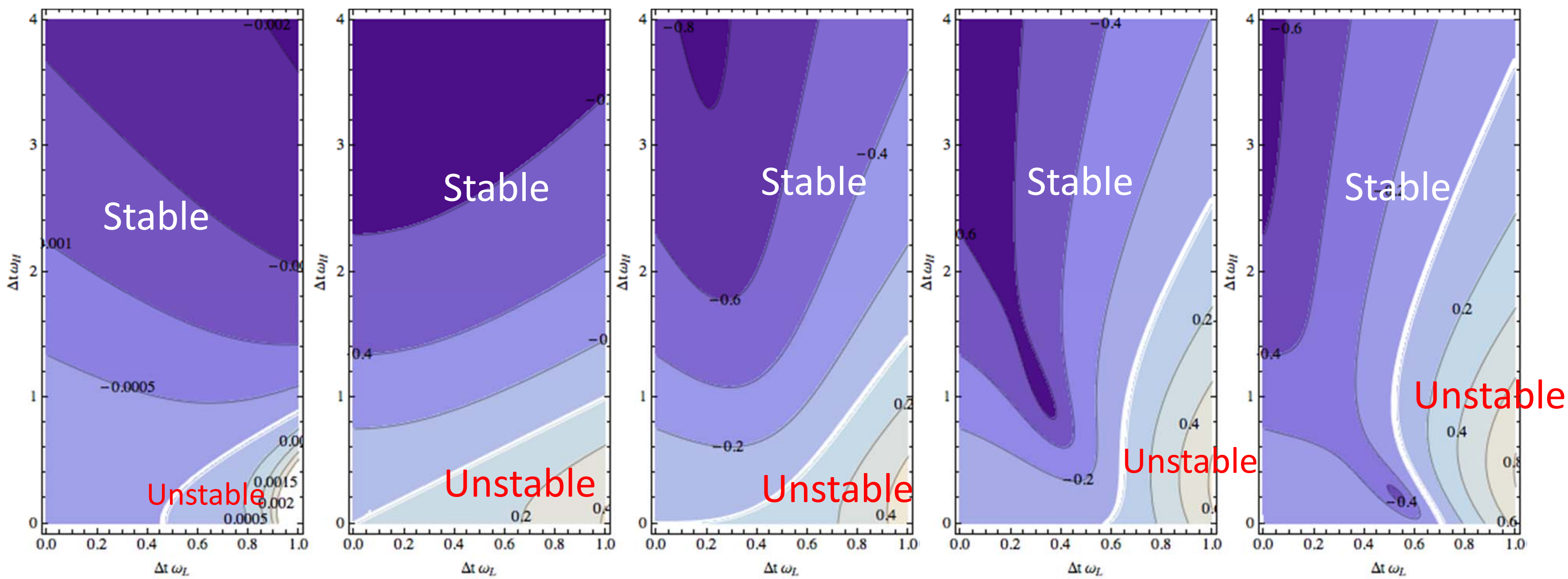
Lorenz 1-cycle (=Forward Euler): Unstable everywhere

Lorenz 2-cycle: Unstable everywhere

Lorenz 3-cycle: Absolutely Unstable for > 1.5

Lorenz 4-cycle: Absolutely Unstable for > 2.7

Plots of |*A*|-1 (α=1: Backward Euler)



Ref:
Leapfrog+R/A filter

Lorenz 1-cycle
(=Forward Euler):

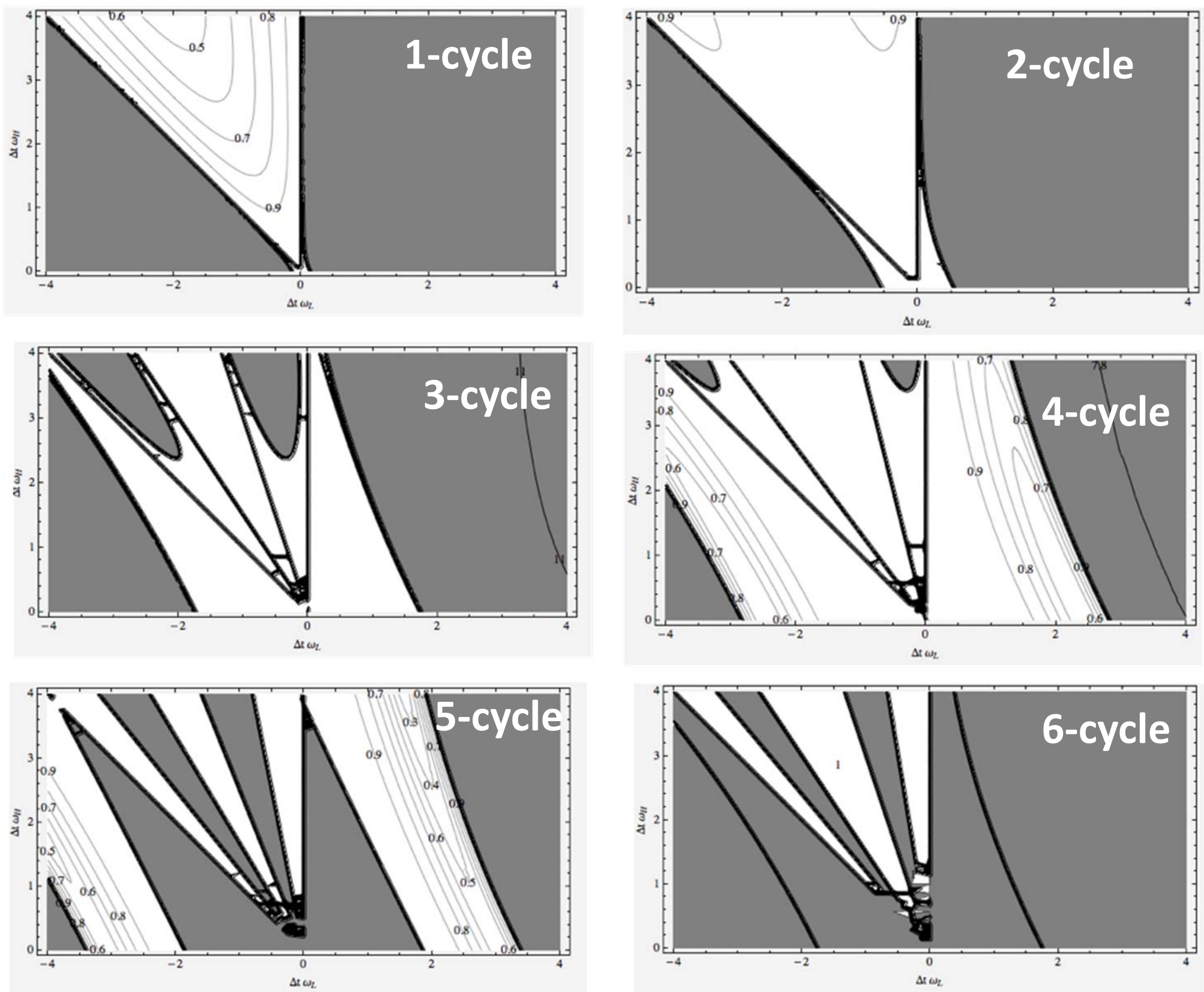
Lorenz 2-cycle:

Lorenz 3-cycle:

Lorenz 4-cycle:

Stability is good, but damping is too strong (~50%)

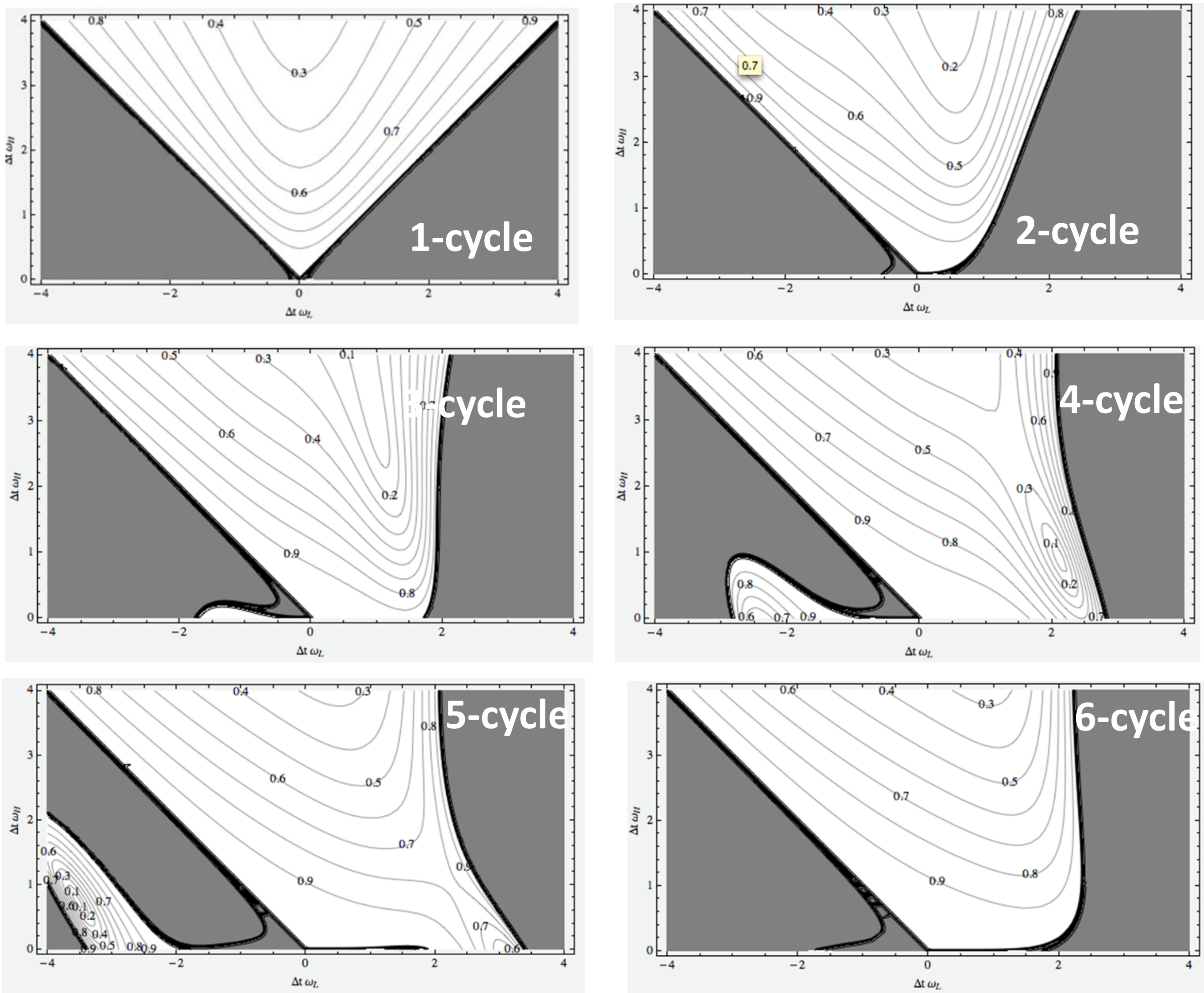
Stability for α=1/2: Crank-Nicolson
N=1,…,6



White: stable |A|< 1

Gray: Unstable |A| > 1

Stability for α=1: Backward Euler
N=1,…,6



White: stable |A|< 1

Gray: Unstable |A| > 1

Stability: Summary

- Crank-Nicolson semi-implicit *N*-cycle is unstable for *N*=1,2
- Stability region is largest for *N*=4
- More unstable than Leapfrog.

- Backward Euler semi-implicit *N*-cycle is more stable, but damping is too strong

Outline

1. Introduction
2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. **Code Validation**
    7. Verification: Dynamical-core test
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual
4. Outcome/Delivarable
5. References

Code Validation: Idea

- Lorenz *N*-cycle:
  - Lorenz 1-cycle is equivalent to Forward Euler, which is built-in in the SPEEDY model.
  - → Compare Lorenz 1-cycle with Forward Euler.
- RK4 :
  - For a linear system, RK4 with $4\Delta t$ is equivalent to Lorenz 4-cycle with Δt.
  - → Remove all nonlinear terms from SPEEDY and Compare RK4 ($4\Delta t$) with Lorenz 4-cycle (Δt).

Code Validation: Results

- Outputs of single step integrations of Lorenz 1-cycle and Forward Euler from the same initial condition are compared using UNIX diff command.
  - Result → no difference (Success)!
- Similarly, outputs of single step integration of RK4 and four-step integration of Lorenz 4-cycle from the same initial condition are compared using UNIX diff command.
  - Result → no difference (Success)!

Outline

1. Introduction

2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. Code Validation
    7. **Verification: Dynamical-core test**
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual

3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual

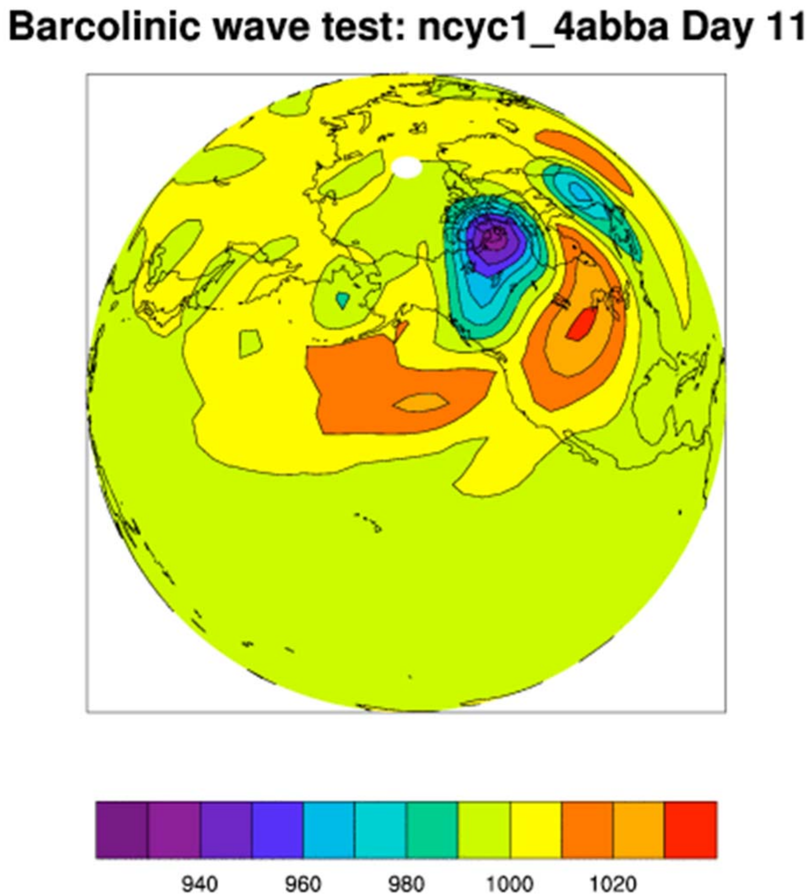
4. Outcome/Delivarable
5. References

Verification: Dynamical-Core test cases

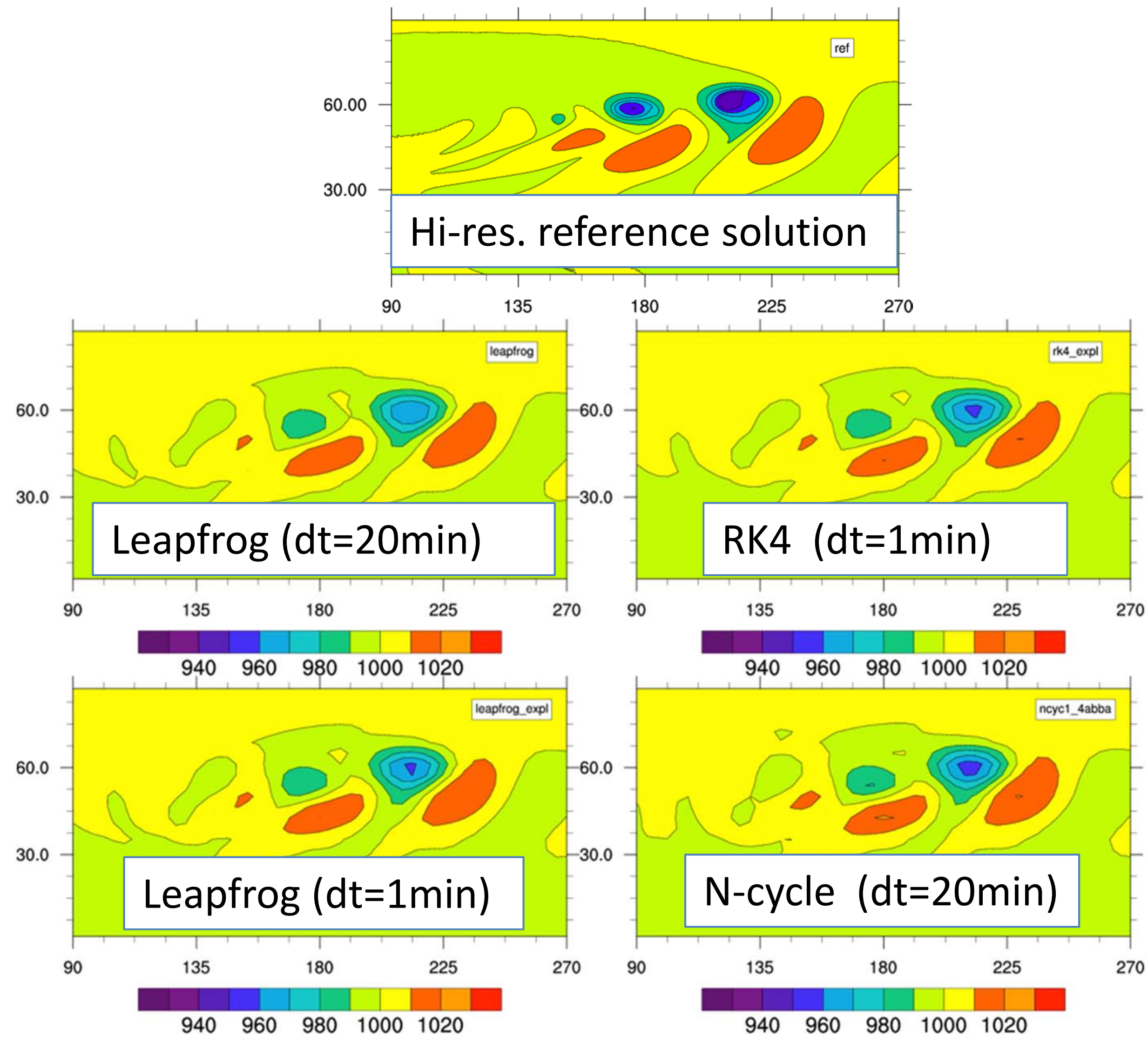
- Standardized tests for dynamical cores of AGCM proposed by Jablonowski and Williamson (2006) which consists of two tests:

1. Steady-State test:
   - A model is integrated from an analytical steady-state solution of the primitive equation
   - The model is evaluated by how well it can keep the steady-state intact.
2. Baroclinic-Wave test:
   - The model is integrated with initial and boundary conditions which is designed to produce an idealized baroclinic wave (= extratropic cyclones and highs)

Baroclinic-wave Test: Result (Lorenz 4-cycle)



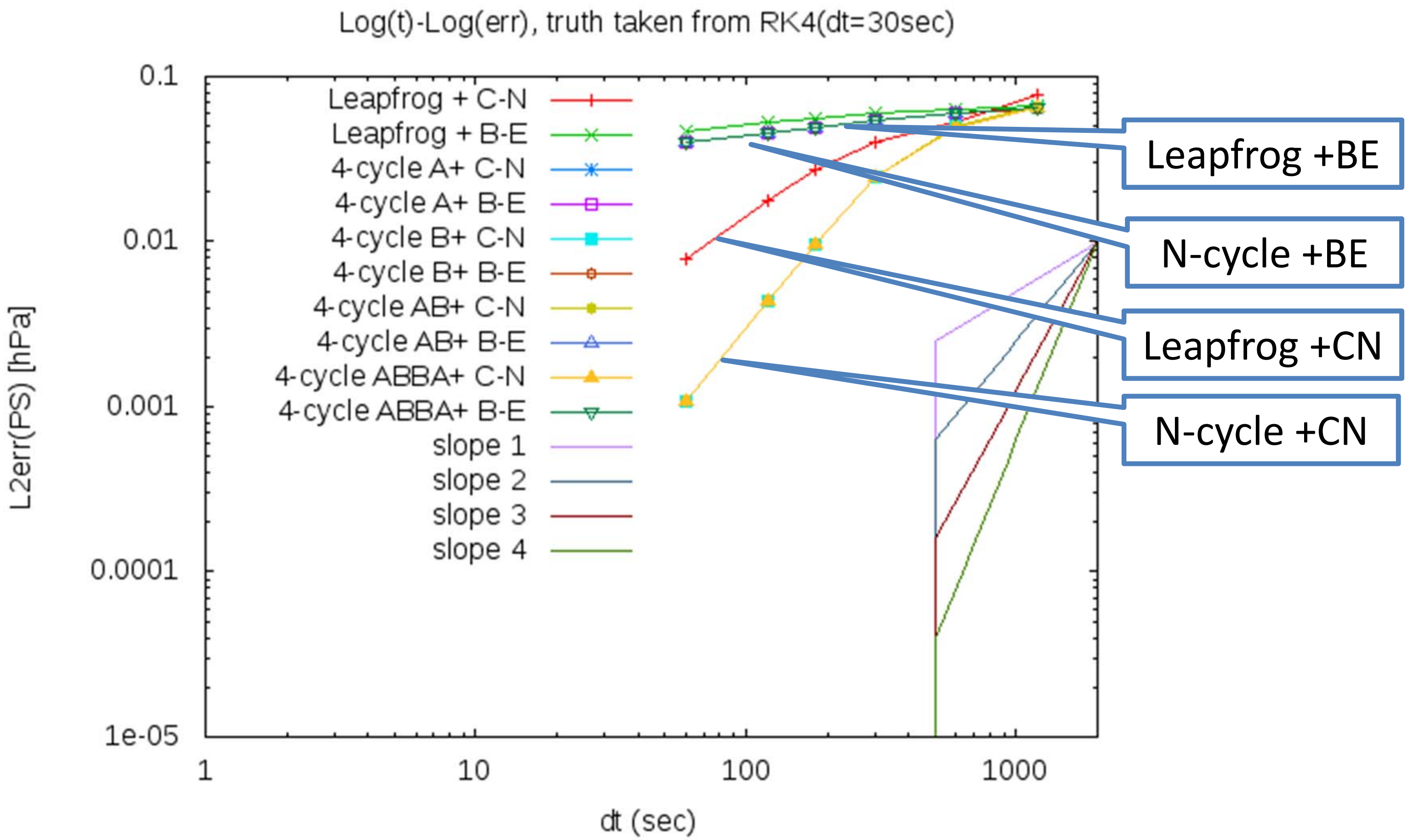
Snapshots of surface-pressure at Day 9



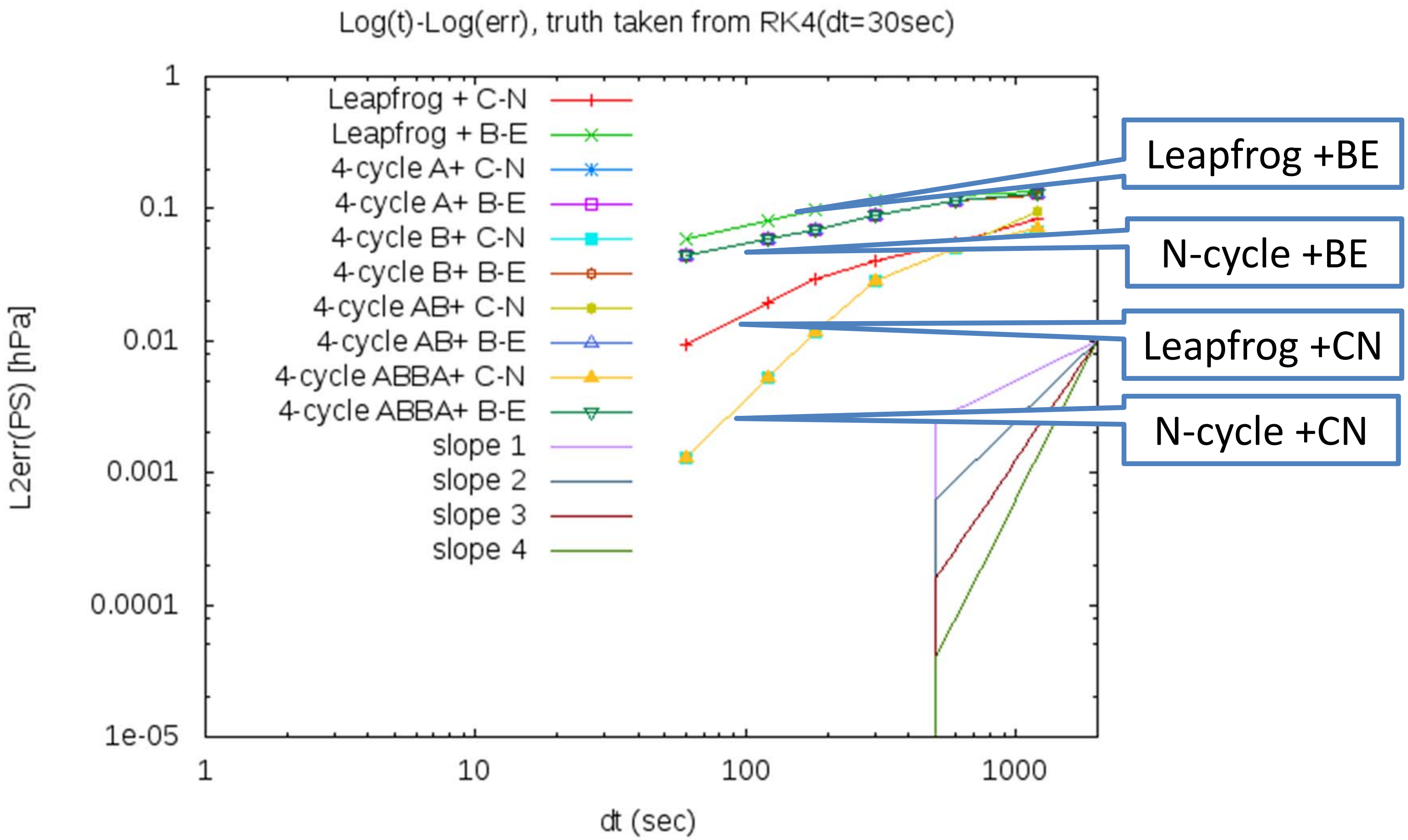
Order estimation

- Lorenz 4-cycle is supposedly of 4th-order, while Leapfrog (with R/A filter) is only of 1st-order
- Confirm this by plotting L^2 error vs. Δt on a log-log plane.
- Experimental set-up: Jablonowski-Williamson baroclinic-wave test
- Measure of the error: difference in surface pressure with respect to the reference solution produced by RK4 with Δt=0.5min in L^2 -norm

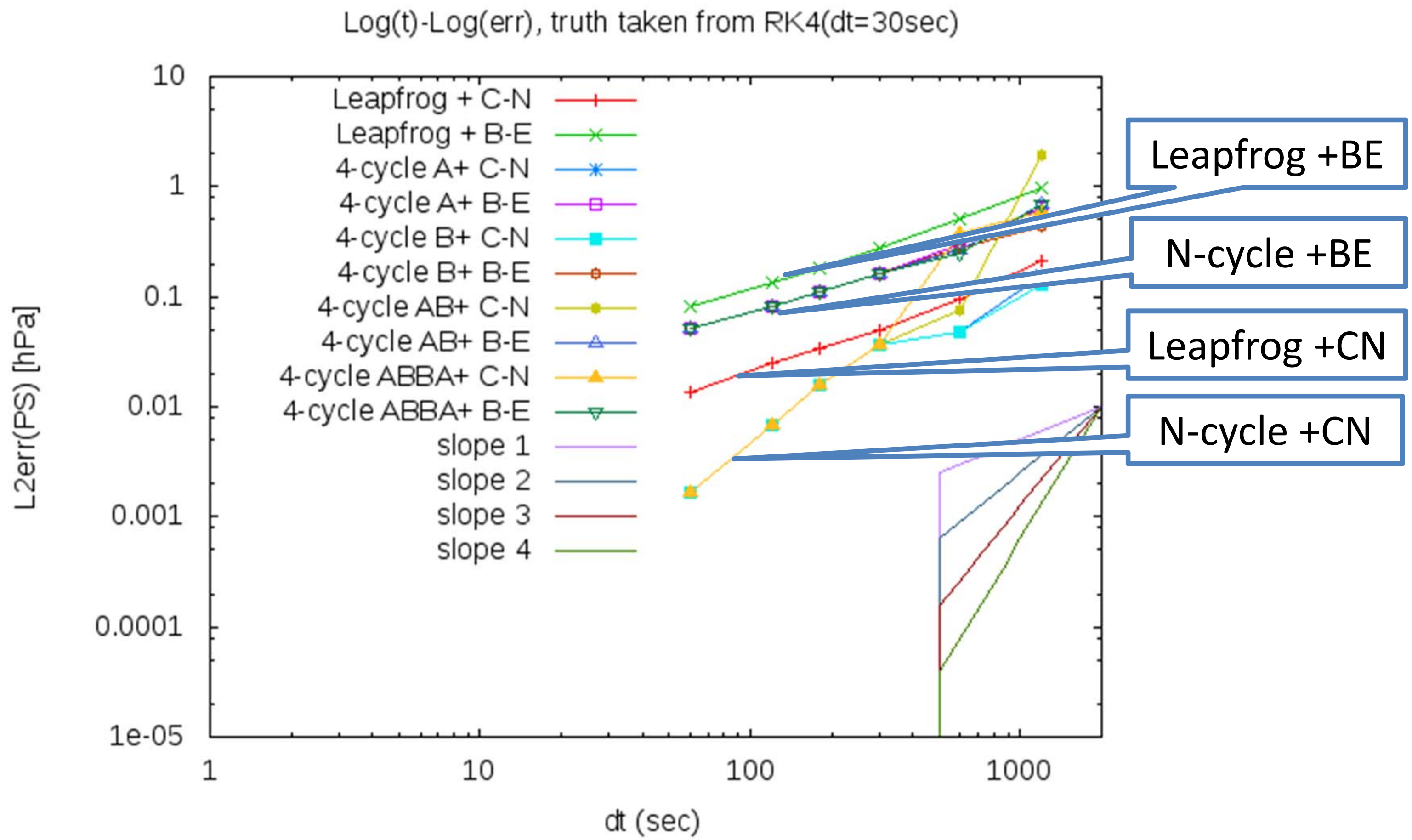
Result : L^2(Ps) at t=3days



Result : L^2(Ps) at t=5days



Result : L^2(Ps) at t=10days



Dynamical-core test
Summary of the results

- For $\Delta t \leq$ 10min, the order of Lorenz 4-cycle is $3^{rd} \sim 4^{th}$
- For a large Δt, A-B-B-A cycle is inferior to version A or version B, which contradicts with Lorenz (1971)'s claim.
- Possible reasons:
  - Cancellation of truncation errors of version A and B does may hold because of the introduction of semi-implicit method.
  - Cancellation between A and B itself is not attained for the AGCM.
  - A-B-B-A cycle is more unstable for nonlinear models than A-only or B-only.

➔ To be examined

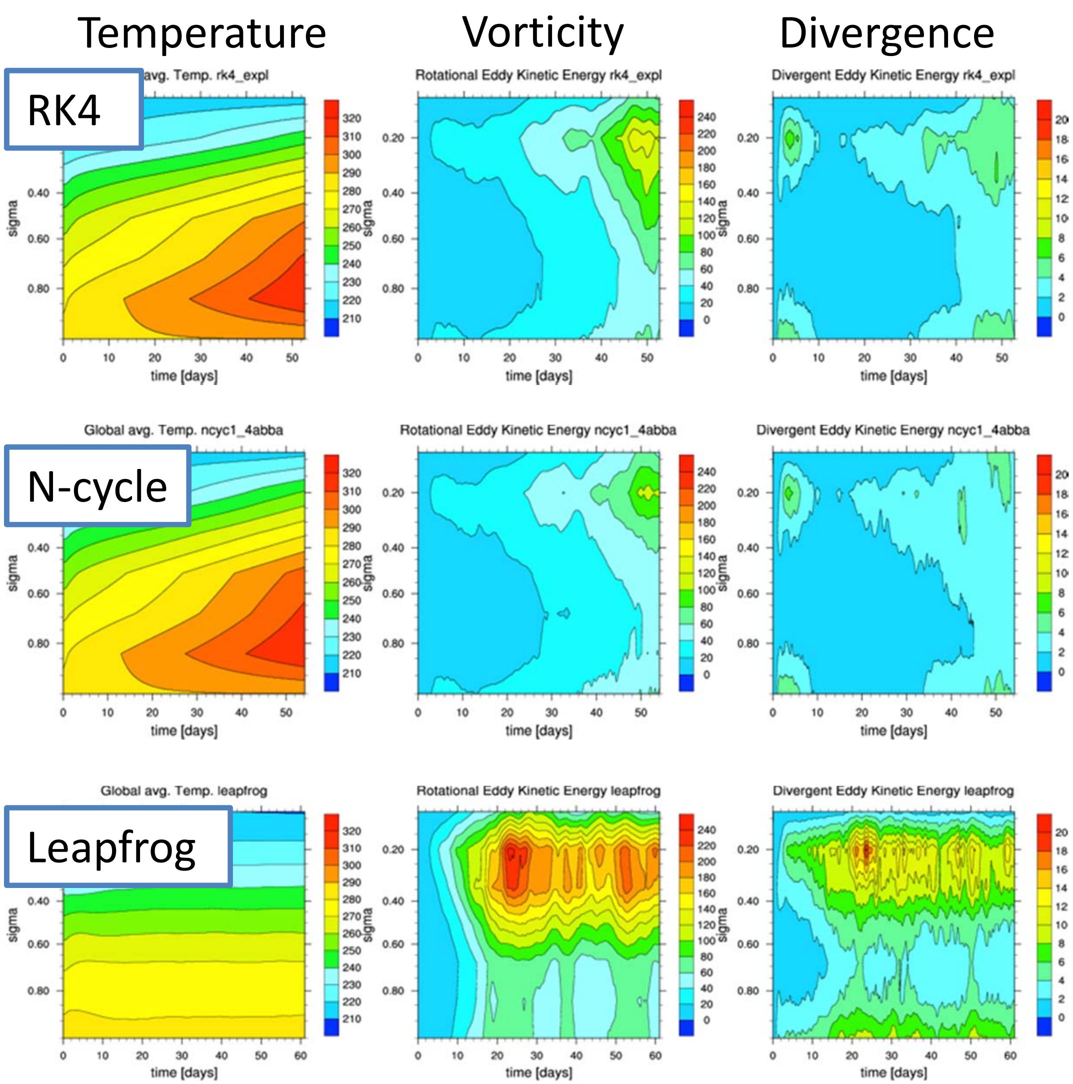
Outline

1. Introduction
2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. Code Validation
    7. Verification: Dynamical-core test
    8. **Inclusion of Physics & Comparison of Climatoligies**
    9. Conclusion
    10. Schedule, planned and actual
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. Model Error Bias: results
    5. Schedule, planned and actual
4. Outcome/Delivarable
5. References

Inclusion of Physics

- Having proved that *N*-cycle works without physical parameterizations, I next tried to include physics in the *N*-cycle SPEEDY model.

Problem encountered in the Last Semester:
N-cycle and RK4 blow up when run with physics on



➔ Resolved:
It was a Stupid Bug !

Stability

- After fixing the bug, I tried to find the largest Δt with which the N-cycle can integrated stably.

Bad news: the largest stable Δt

- for N-cycle : 15 mins, whereas
- for Leapfrog: 40mins.

Possible remedy (with some accuracy degradation) :

- the largest stable Δt can be made 30mins
- by using Backward Euler for the gravity-wave part (instead of the default Crank-Nicolson)

- N-cycle can be integrated with Δt=15mins. for at least 100 years (=3,504,000 steps)

Comparison of Climatology (= long-term mean)

- In climate application, it is important that the long-term mean of the atmospheric state (called *climatology*) does not change.

- ➔ Statistically compared the climatologies using Welch's *t*-test.

- Examined Climatologies:

  two seasons (DJF: 12-2 and JJA: 6-8), each from 3 models:
  - Leapfrog with dt=40mins (default)
  - 4-cycle (Crank-Nicolson for gravity-wave) dt=15mins
  - 4-cycle (Backward Euler for gravity-wave) dt=30mins

t-test for the difference of climatologies

- Experimental design:
  - Run all the models from the same initial condition (state of rest)
  - Integrate for 30 years.
  - Discard the first 10 years as a spin-up.
  - Use the remaining 20 years as the samples.
- Null-hypothesis:

climatologies (= sample means) are taken from the same population.

- The probability of the differences between two sampled climatologies being larger than the observed difference under the null-hypothesis is computed.
- If this probability is larger than 95%, then the null-hypothesis is not rejected.

Results for T, Z, U and V, MSLP and OLR: **no significant difference !**

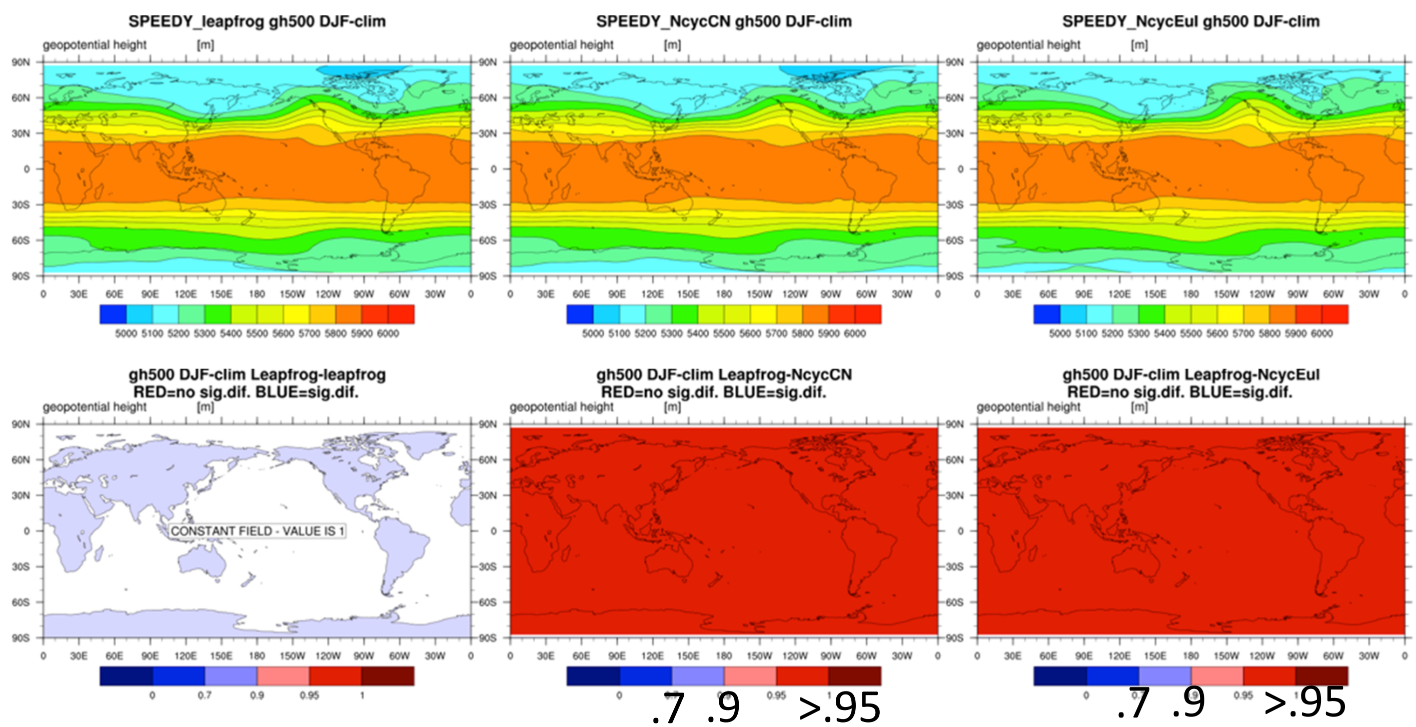
- all results are uploaded on

http://www.atmos.umd.edu/~dhotta/speedy_clim2/clim.html

Example: Z500	Leapfrog + CrankNicolson	4-cycle + CrankNicolson	4-cycle + Backward Euler

climatology

Probability of null-hypothesis not being rejected



Conclusion for Phase I

- Designed semi-implicit version of *N*-cycle
  - Analyzed stability using a toy-model
  - 4-cycle is the most stable
  - For semi-implicit method, Crank-Nicolson is more accurate than Backward-Euler but is more unstable
- Verification through Dynamical-Core test:
  - 4-cycle with Crank-Nicolson exhibits order-of-accuracy which is higher ($2^{nd} \sim 3^{rd}$) than filtered Leapfrog (1^{st}-order)

Schedule for Phase 1: Planned and Actual

Planned

- Implement RK4 and *N*-cycle, Nov.
- Write the mid-year report, prepare the oral presentation, Dec.
- Switch-off physical parameterizations, prepare flat topography, Jan.
- perform the dynamical core tests. Feb.

Actual

- Formulate semi-implicit N-cycle, Oct.
- Implement RK4 and *N*-cycle, Nov.
- Switch-off physical parameterizations, prepare flat topography, Dec.
- perform the dynamical core tests, Dec.
- Write the mid-year report, prepare the oral presentation, Dec.
- Coded a bug, and fixed the bug, Jan. (winter break)
- Compare climatologies, performed statistical test, Feb.

Phase II:
Empirical characterization of Model Errors

Outline

1. Introduction

2. Phase I: Implementation of Lorenz *N*-cycle to SPEEDY model
   1. Approach
   2. Algorithms
   3. Model description
   4. Stiffness-problem and semi-implicit method
   5. Semi-implicit Lorenz *N*-cycle
   6. Code Validation
   7. Verification: Dynamical-core test
   8. Inclusion of Physics & Comparison of Climatoligies
   9. Conclusion
   10. Schedule, planned and actual

3. Phase II: Empirical Characterization of Model Errors
   1. **Approach**
   2. Algorithm
   3. Interpolation
   4. Model Error Bias: results
   5. Schedule, planned and actual

4. Outcome/Delivarable
5. References

Phase 2: Approach

- Objective: Characterize the model errors due to temporal discretizations

- Take the Truth from NCEP/NCAR reanalysis (Kalnay et al. 1996)

NCEP=National Centers for Environmental Prediction

NCAR=National Center for Atmospheric Research

- Extract model errors by applying the method of Danforth et al. (2007) to the models with:

1. the original scheme (Leap-Frog; M^{LF})
2. Lorenz N-cycle scheme (M^{NCYC})

- (time permitting) Correct the model errors on-line during the course of model integration

  (➔ Phase 3&4)

Phase 2: Algorithm

1. Generate initial values from the Truth (NCEP/NCAR reanalysis)
2. Perform short-range forecasts using the 2 models (*M*^{LF}, *M*^{NCYC4}) from the initial conditions
3. find the bias of the model errors for each model
4. Build the covariance matrix

$$\left\langle (x(t) - \bar{x}) \left(M^{true}(x(t)) - M^{LF}(x(t)) \right)^T \right\rangle$$

5. Extract the dominant modes by conducting SVD

Interpolation from NCEP/NCAR grid to SPEEDY grid

- Obtained the original code (used in Danforth et.al (2007)) and wrote a code that does exactly the same operation
- Original code:
  - written in MatLab script
  - performs Simple linear interpolation in 3D
- My code:
  - wrote in NCL (NCAR Command Language)
  - Basically a line-by-line translation from MatLab to NCL
- Validation:
  - Method:
    - produce data on SPEEDY grid from NCEP/NCAR data using the original code and my code
    - compare the two outputs, one from the original, the other from my code
  - Result: the two outputs agreed within single-precision rounding error

Outline

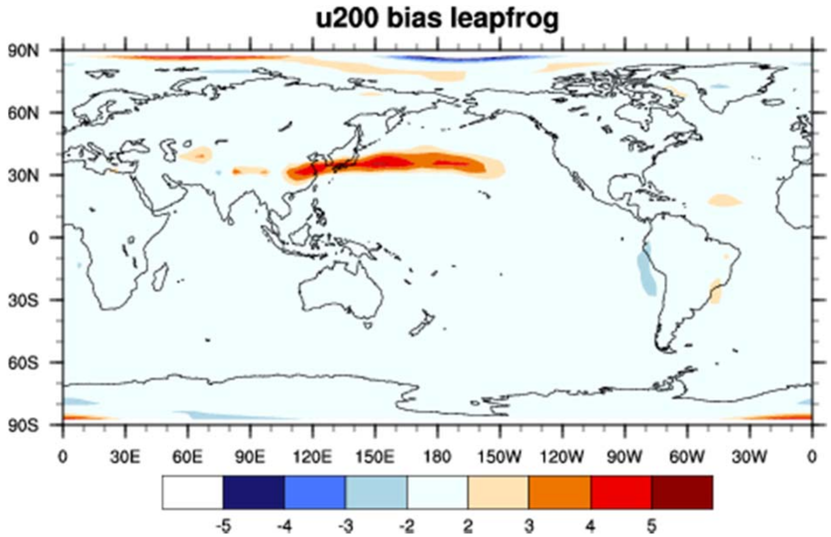
1. Introduction

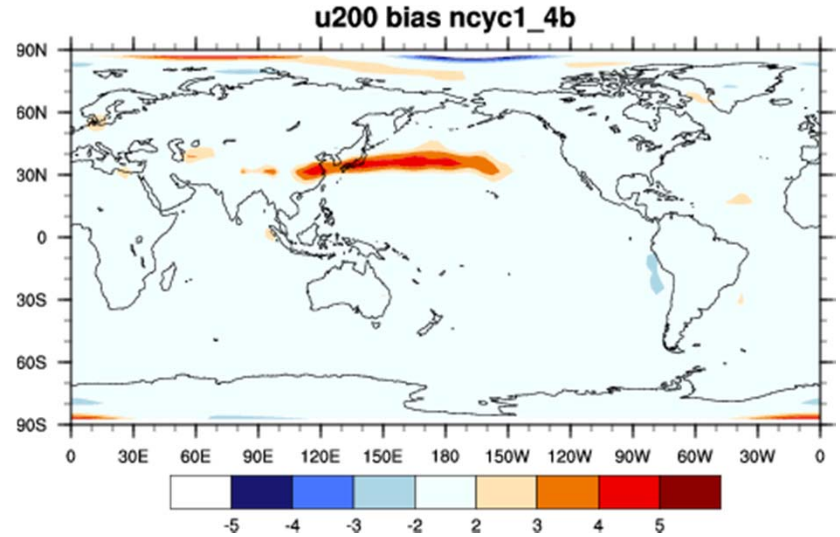
2. Phase I: Implementation of Lorenz *N*-cycle
 to SPEEDY model
    1. Approach
    2. Algorithms
    3. Model description
    4. Stiffness-problem and semi-implicit method
    5. Semi-implicit Lorenz *N*-cycle
    6. Code Validation
    7. Verification: Dynamical-core test
    8. Inclusion of Physics & Comparison of Climatoligies
    9. Conclusion
    10. Schedule, planned and actual

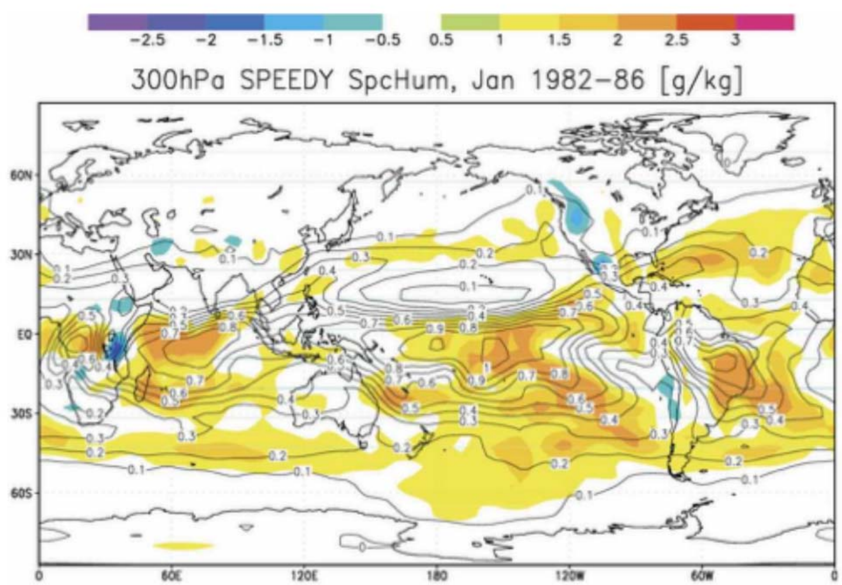
3. Phase II: Empirical Characterization of Model Errors
    1. Approach
    2. Algorithm
    3. Interpolation
    4. **Model Error Bias: results**
    5. Schedule, planned and actual

4. Outcome/Delivarable
5. References

Model Error Bias : Results
Zonal wind at 200hPa

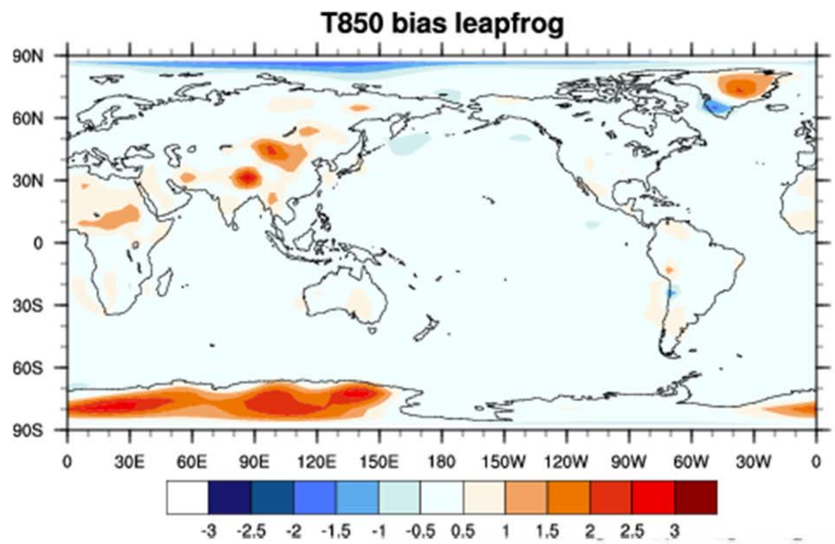


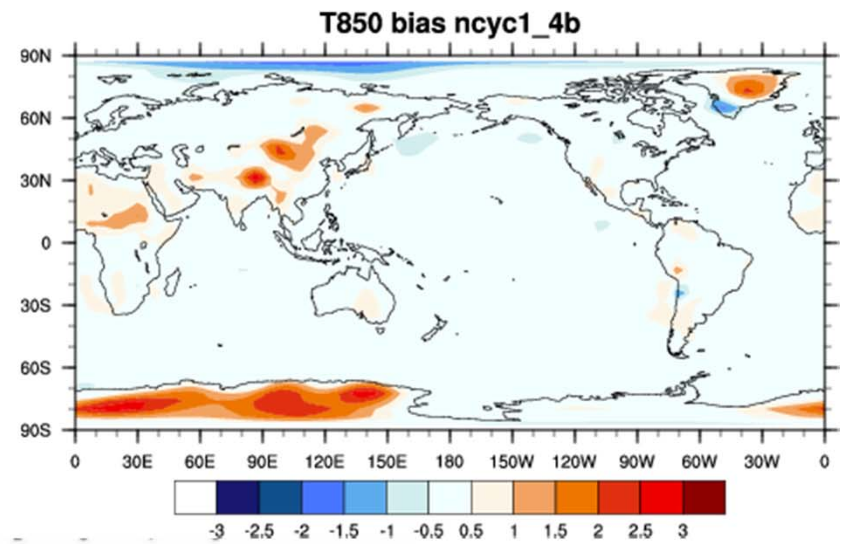


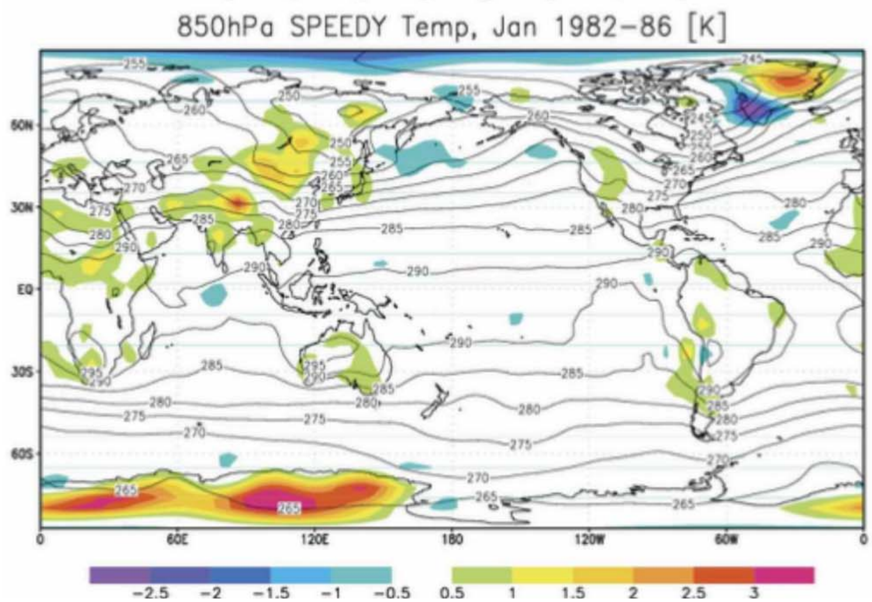


From Danforth et.al (2007)

Model Error Bias : Results
Temperature at 850hPa

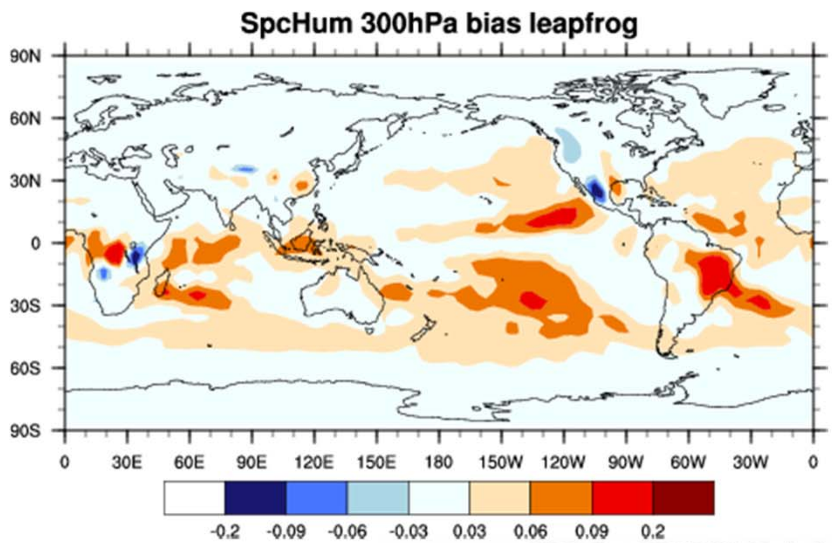


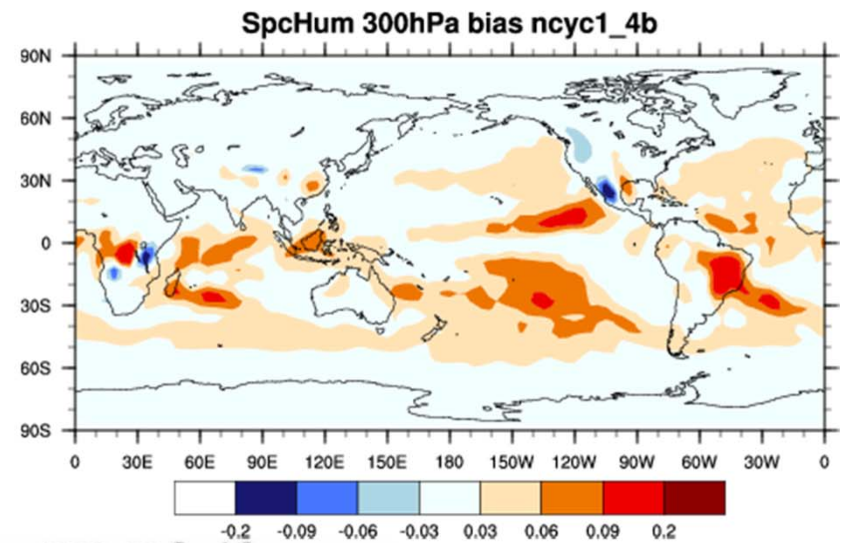


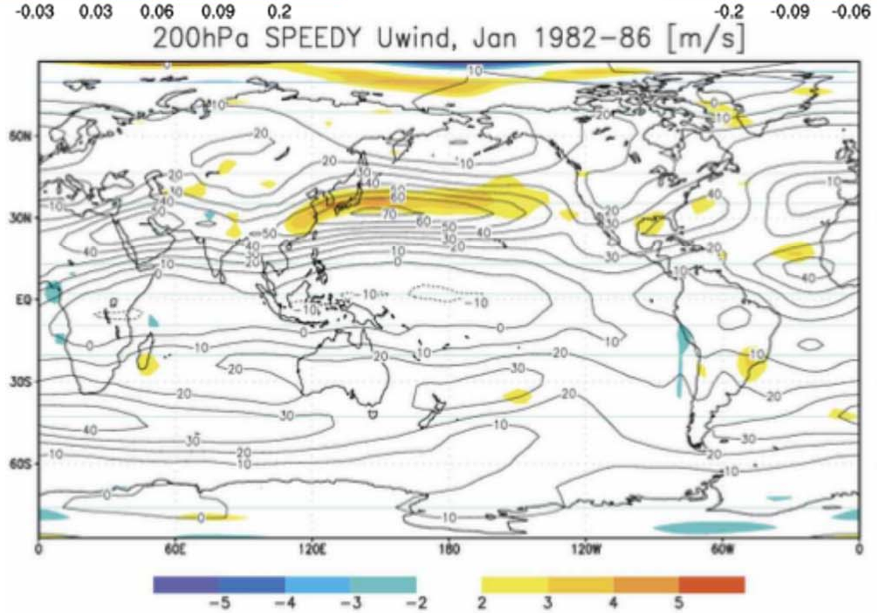


From Danforth et.al (2007)

Model Error Bias : Results
Specific Humidity at 300hPa

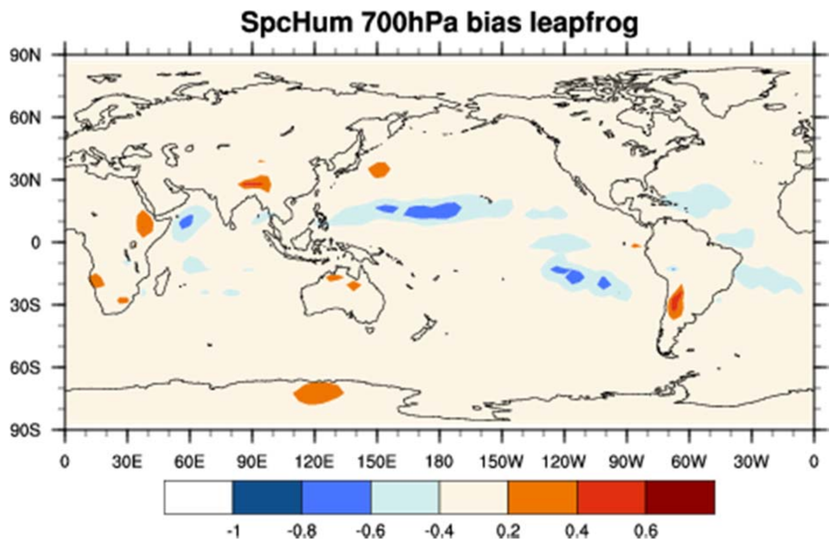


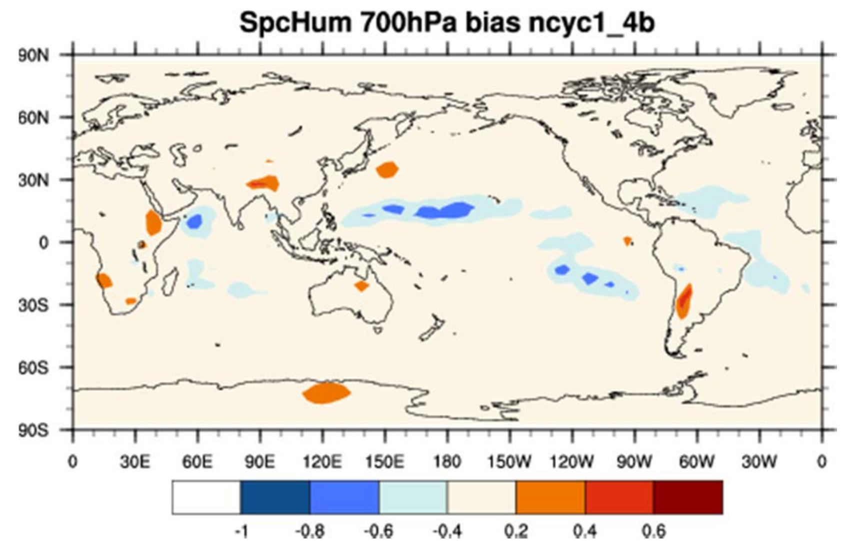


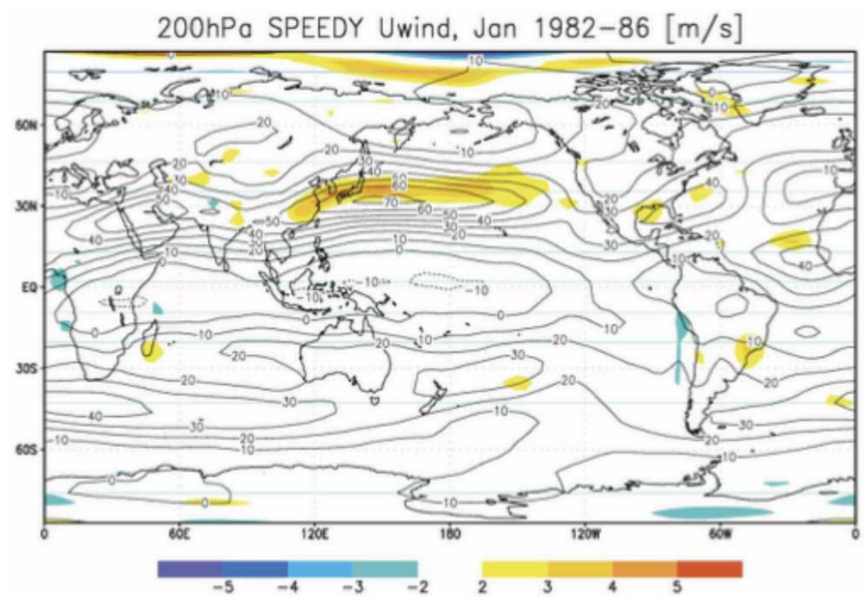


From Danforth et.al (2007)

Model Error Bias : Results
Specific Humidity at 700hPa







From Danforth et.al (2007)

Model Error Bias : Summary

- For all variables, Leapfrog and *N*-cycle produce almost identical bias pattern
  - Interpretation:

  bias is dominated by the model's deficiencies associated with physical parameterizations

- They are consistent with Danforth et.al (2007)

Plan for Phase II

In Danforth et.al (2007),

1. Error samples {δx} are produced with the original model *M*
2. Bias <δx> is computed
3. The model is modified by incorporating nudging to -<δx> to yield M^+
4. Error samples {δx^+} are resampled using the de-biased model M^+
5. Diurnal errors are extracted from {δx^+} by performing EOF analysis and retaining the first two dominant modes
6. The de-biased model M^+ is again modified by incorporating nudging to negative of diurnal biases to yield M^{++}
7. Error samples {δx^{++}} are again resampled using M^{++}
8. Perform SVD analysis to extract dominant co-variation of model error δx^{++} and the anomaly of the model states

while, in my original plan,

1. Error samples {δx} are produced with the original model *M*
2. Bias <δx> is computed
3. The model is modified by incorporating nudging to -<δx> to yield M^+
4. Error samples {δx^+} are resampled using the de-biased model M^+
5. Diurnal errors are extracted from {δx^+} by performing EOF analysis and retaining the first two dominant modes
6. The de-biased model M^+ is again modified by incorporating nudging to negative of diurnal biases to yield M^{++}
7. Error samples {δx^{++}} are again resampled using M^{++}

3. Perform SVD analysis to extract dominant co-variation of model error δx^{++} and the anomaly of the model states
4. Validation: Results are compared with Danforth et.al (2007)

Plan for Phase II

- Danforth et.al (2007) involves much more tricks than I originally planned.
- Reproducing all the procedures in Danforth et.al (2007) is impossible given that I have only 2-weeks left.

→

- I continue the original plan, but modify the Validation part.
- New Validation:
  - Check orthogonality between the extracted modes

Conclusion for Phase II

- Generated samples of model error
- Computed biases for Leapfrog and Lorenz 4-cycle, compared them with Danforth et.al (2007)

→Result:
  - No significant bias improvements by using a better temporal scheme. Perhaps dominated by physics errors
  - Consistent with Danforth et.al (2007)

- TODO (in two weeks):
  - SVD analysis to identify the dominant co-varying modes between model state anomaly and model error
  - Comparison with Danforth et.al (2007)

Schedule for Phase 2: Planned and Actual

Planned

- Generate initial values from the NCEP/NCAR reanalysis, end of Feb.
- build the bias and covariance matrix, Mar.
- Code and test a program for SVD, Apr.
- Compare the model errors for the new and the original schemes, May.
- Write the final report (paper draft), May.

Actual

- Generate initial values from the NCEP/NCAR reanalysis, end of Feb.
- Compute the bias, Mar.
- Plot the bias and compare It with Danforth et.al, Apr. (← Now I'm here)
- Code and test a program for SVD, May.
- Compare the model errors for the new and the original schemes, May.
- Write the final report (paper draft), May.

Outcome/Deliverables

Phase 1:

- Upgraded code for SPEEDY model ✔

  - subroutines for Lorenz N-cycle and 4th order Runge-Kutta
- Test-case results for the SPEEDY model (both for the original scheme and the new schemes) ✔

Available at https://code.google.com/p/speedy-lorenz-ncycle/

Phase 2:

- Archive of the model errors ✔
- Bias of model errors ✔
- Pairs of Singular Vectors for the model state and the model error
- Code for performing SVD

To be completed in two weeks

Bibliography

Lorenz *N*-cycle

- Lorenz, Edward N., 1971: An N-cycle time-differencing scheme for stepwise numerical integration. *Mon. Wea. Rev.*, **99**, 644–648.

SPEEDY model

- Molteni, Franco, 2003: Atmospheric simulations using a GCM with simplified physical parameterizations. I. Model climatology and variability in multi-decadal experiments. *Clim. Dyn.,* **20**, 175-191.
- Kucharski F, Molteni F, and Bracco A, 2006: Decadal interactions between the western tropical Pacific and the North Atlantic Oscillation. *Clim. Dyn*., **26**, 79-91

SPEEDY-LETKF

- Miyoshi, T**.**, 2005: *Ensemble Kalman filter experiments with a primitive-equation global model*. Ph.D. dissertation, University of Maryland, College Park, 197pp.

Atmospheric GCM Dynamical Core test cases

- Jablonowski, C. and D. L. Williamson 2006: A baroclinic instability test case for atmospheric model dynamical cores, *Q. J. R. Metorol. Soc*., **132**, 2943-2975

NCEP/NCAR reanalysis

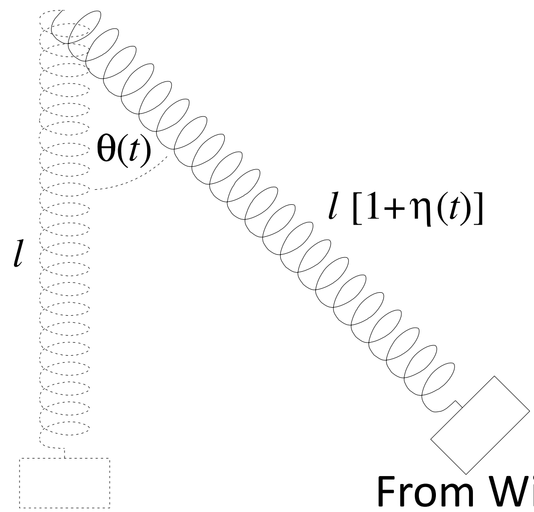
- Kalnay, E., and Coauthors, 1996: The NCEP/NCAR 40-Year Reanalysis Project. *Bull. Amer. Meteor. Soc.*, **77**, 437–471.

Model Error Correction

- Danforth, Christopher M., Eugenia Kalnay, Takemasa Miyoshi, 2007: Estimating and Correcting Global Weather Model Error. *Mon. Wea. Rev.*, **135**, 281–299.
- Danforth, Christopher M., Eugenia Kalnay, 2008: Using Singular Value Decomposition to Parameterize State-Dependent Model Errors. *J. Atmos. Sci.*, **65**, 1467–1478.

back-up slides

Swinging-pendulum problem



From Williams (2011)

FIG. 7. Schematic diagram of the elastic pendulum, or swinging spring, showing the equilibrium position (dotted) and a non-equilibrium position defined by the values of η and θ (solid).

$$\dot{\eta} = \underline{\upsilon_\eta},$$

$$\dot{\upsilon}_\eta = -\omega_{\rm low}^2(1-\cos\theta) - \underline{\omega_{\rm high}^2\eta} + (1+\eta)\upsilon_\theta^2,$$

$$\dot{\theta} = \upsilon_\theta, \quad \text{and}$$

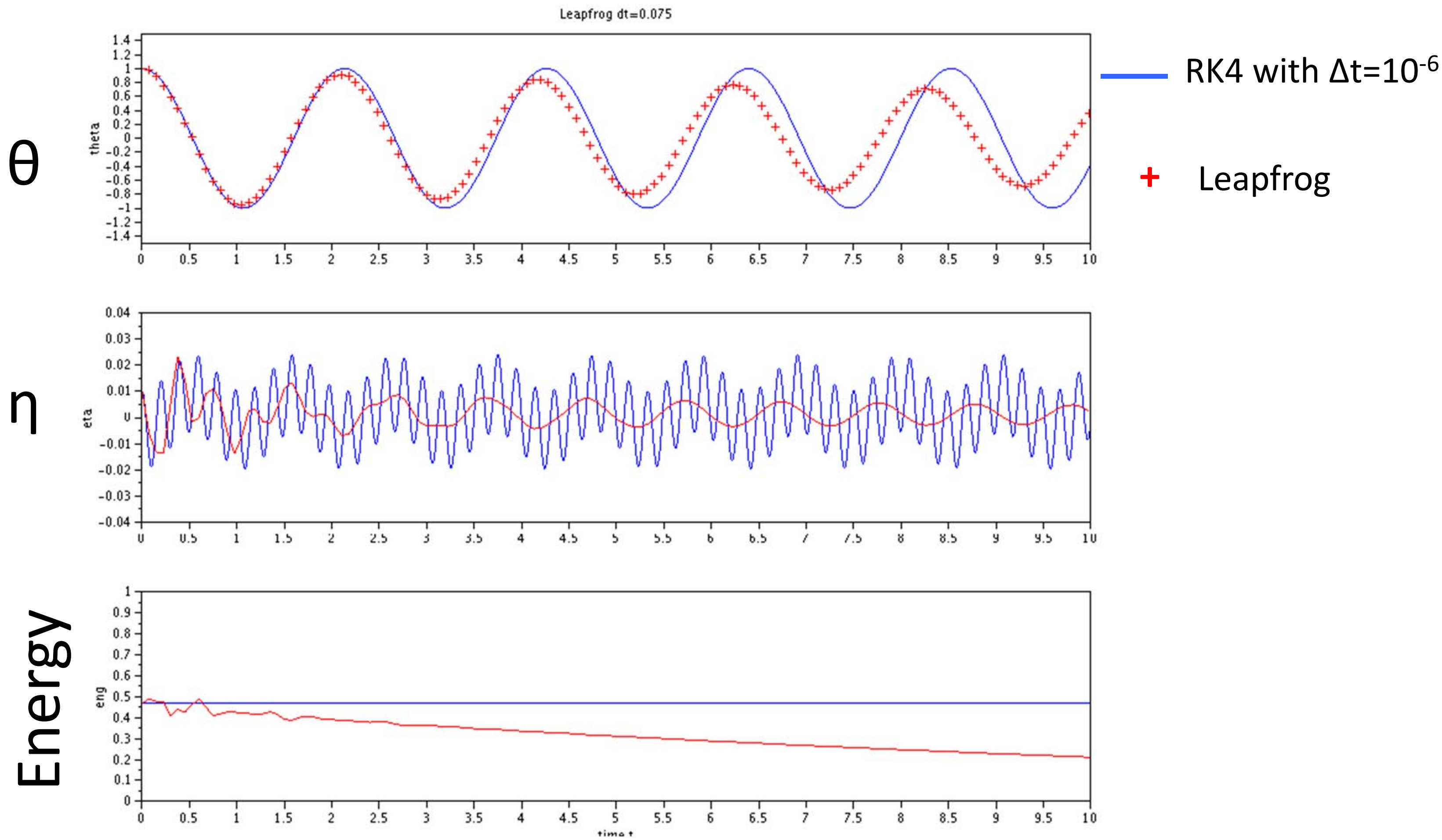
$$\dot{\upsilon}_\theta = \frac{-\omega_{\rm low}^2\sin\theta - 2\upsilon_\eta\upsilon_\theta}{1+\eta}.$$

- A simple nonlinear test-bed for semi-implicit schemes.
- Fast oscillation = elastic spring
- Slow oscillation = pendulum
- Fast mode is treated implicitly, slow mode explicitly.

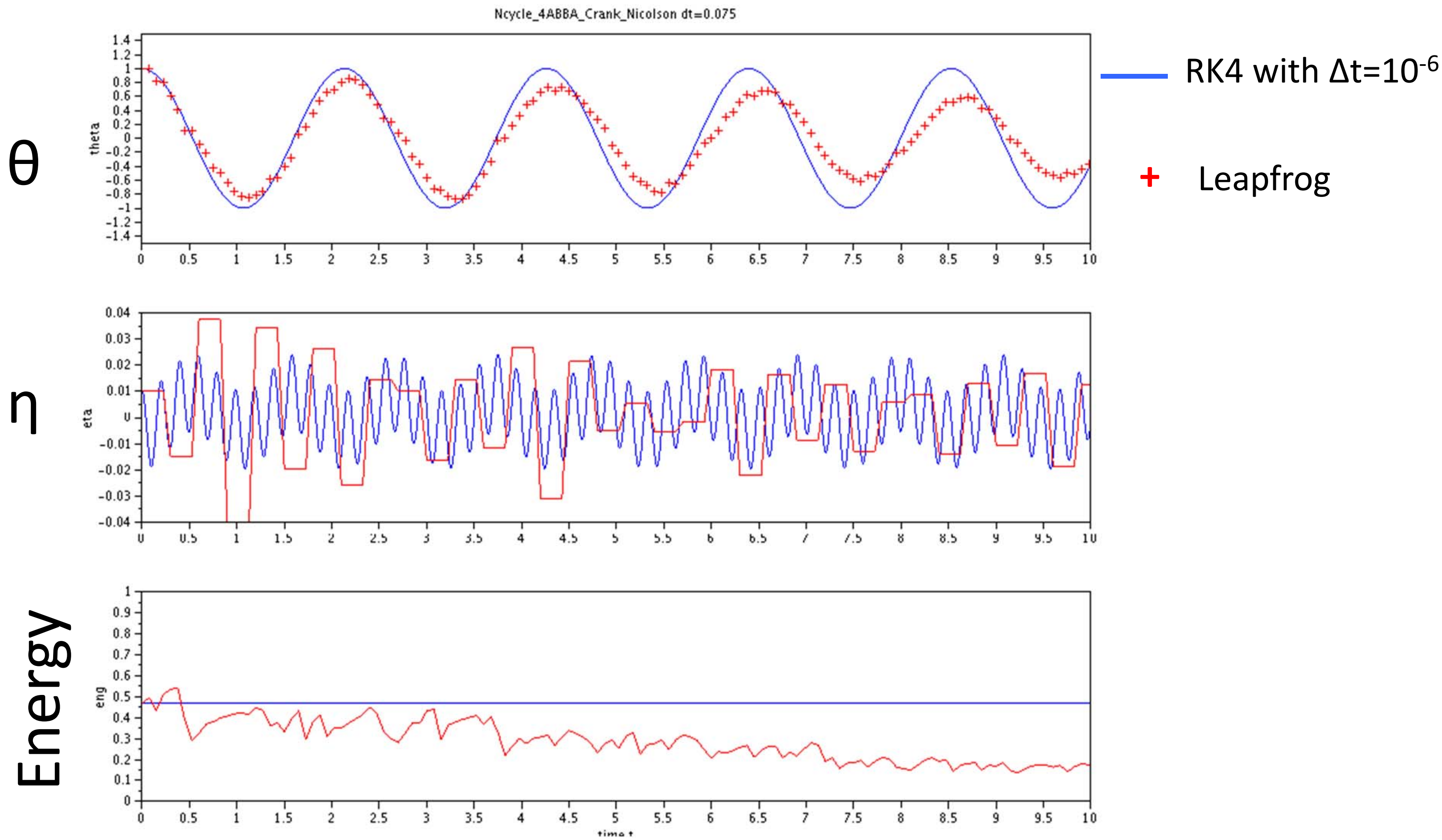
- For this test, I use $\Delta t=0.075$ which gives

$\omega_{LOW}\Delta t=0.225$, $\omega_{HIGH}\Delta t=2.25$,

Leapfrog with RA filter (α=0.01)

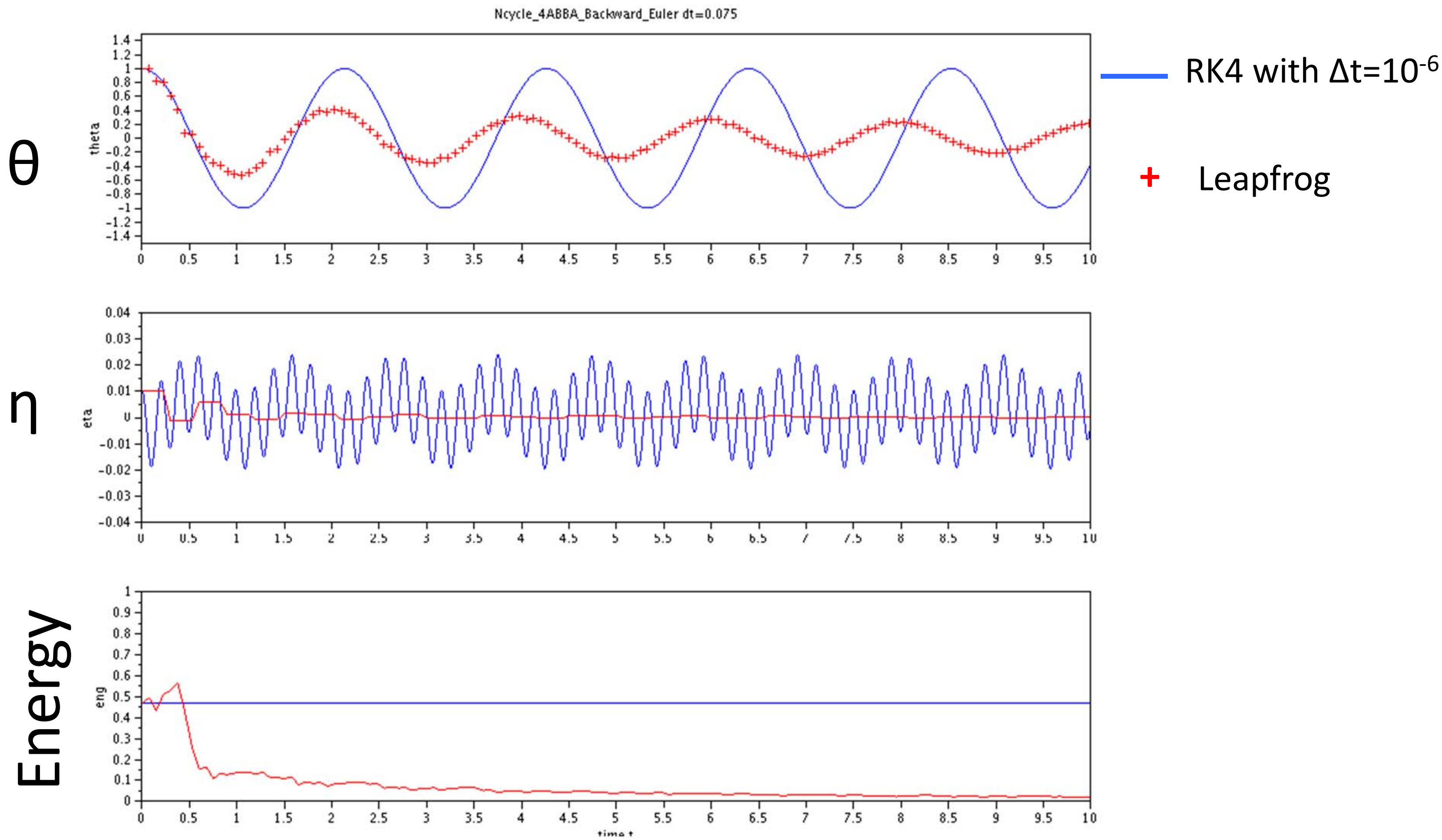


4-cycle with semi-implicit correction on every 4 steps (Crank-Nicolson)

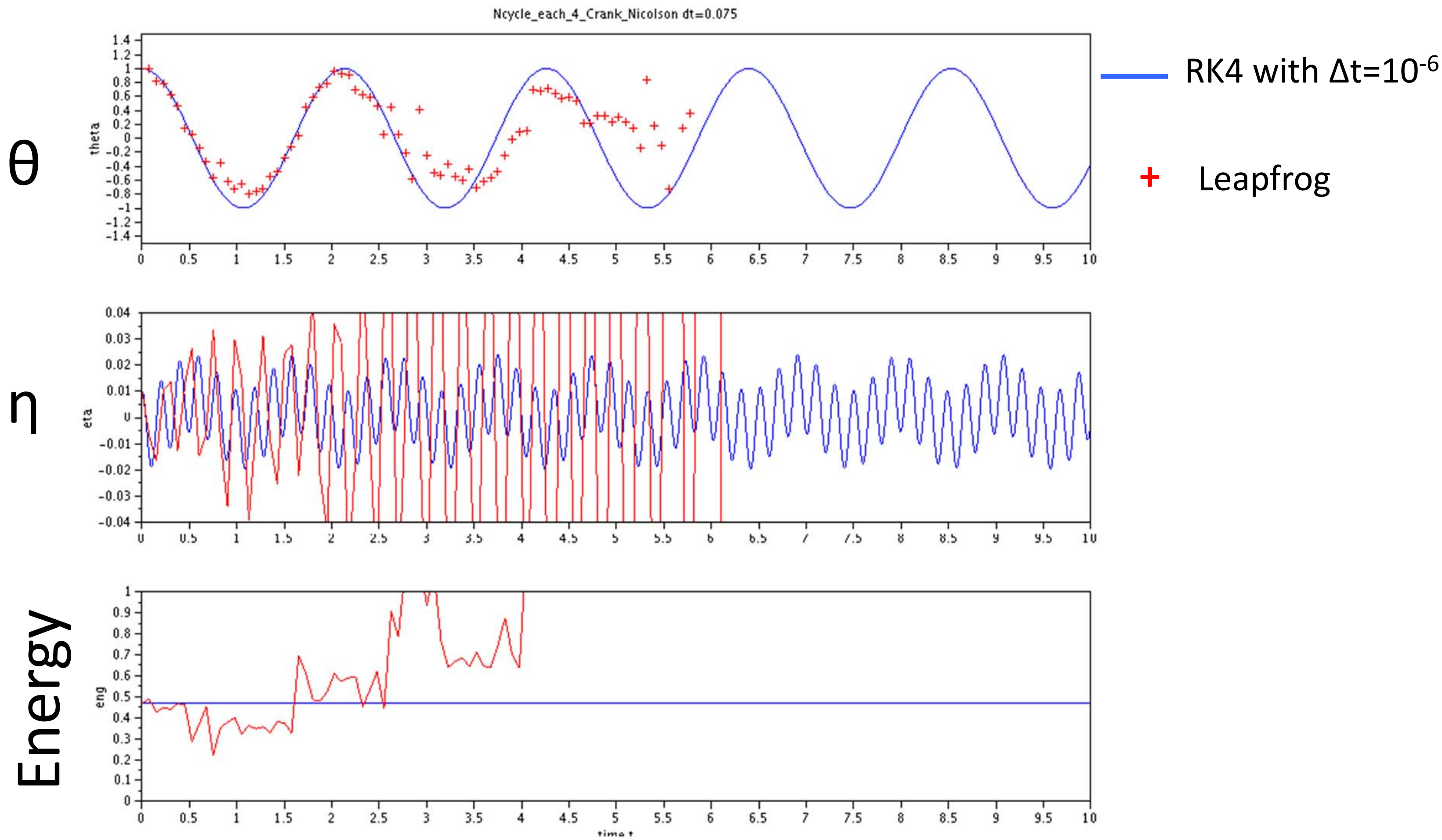


4-cycle with semi-implicit correction on every 4 steps (Backward)

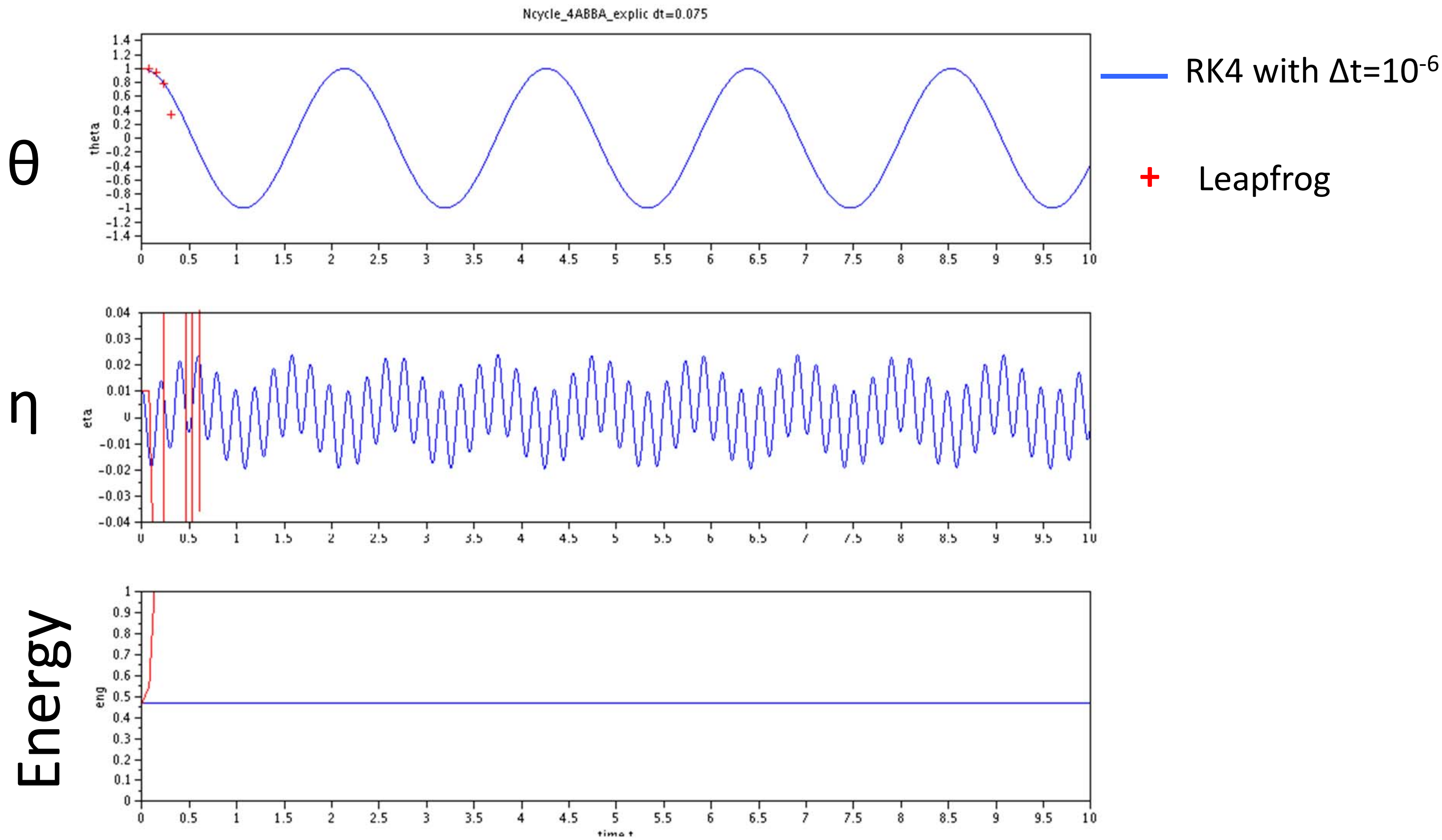
➔ stable but very dissipative



4-cycle with semi-implicit correction on every time step → unstable



Explicit 4-cycle
➔ unstable



Summary

Consistent with the linear analysis,

- N-cycle with semi-implicit on each time step
  - ➔ unstable
- N-cycle with semi-implicit on every N steps
  - ➔ Crank-Nicolson: stable, but comparable accuracy with Leapfrog
  - ➔ Backward Euler: stable, very dissipative

Reference:

- Williams, P. D. 2011: The RAW Filter: An Improvement to the Robert–Asselin Filter in Semi-Implicit Integrations, *Mon. Weath. Rev.*, **139**, 1996--2007

Runge-Kutta 4th-order scheme with semi-implicit correction

Introduction

- I implemented semi-implicit Runge-Kutta 4th-order scheme to SPEEDY model and found that
  - for Crank-Nicolson, the model blows up, even for very small dt (1min).
  - for Backward Euler, the model is too diffusive that the baroclinic-wave dynamical core test fails to produce the baroclinic wave.
- Explicit Runge-Kutta 4th-order scheme with small dt (5min) works fine for the dynamical core test.
- → examine stability using the toy-model

Method

- Following Durran (1991, 1999; MWR) and Williamson (2011; MWR), apply semi-implicit modification to the second term of the equation:

$$\frac{du}{dt} = i\omega_L u + i\omega_H u$$

- Examine the modulus of Amplification factor *A*.
- If |*A*| < 1, the scheme is stable.
- Range of interest:

$$\omega_L \Delta t < 0.5, \quad \omega_H \Delta t > 2$$

i.e., CFL condition is met for low-frequency part but is violated for high-frequency part.

Algorithm:
RK4 for ω_L , Crank-Nicolson for ω_H

$$
\begin{aligned}
h^1 &= i\omega_L u^n \\
u_2^* &= u^n + \tfrac{\Delta t}{2} h^1 \\
h^2 &= i\omega_L u_2^* \\
u_3^* &= u^n + \tfrac{\Delta t}{2} h^2 \\
h^3 &= i\omega_L u_3^* \\
u_4^* &= u^n + \Delta t h^3 \\
h^4 &= i\omega_L u_4^* \\
G &= \tfrac{1}{6}(h^1 + 2h^2 + 2h^3 + h^4) \\
\frac{u^{n+1} - u^n}{\Delta t} &= G + \left\{\beta i\omega_H u^{n+1}(1-\beta) i\omega_H u^n\right\}
\end{aligned}
$$

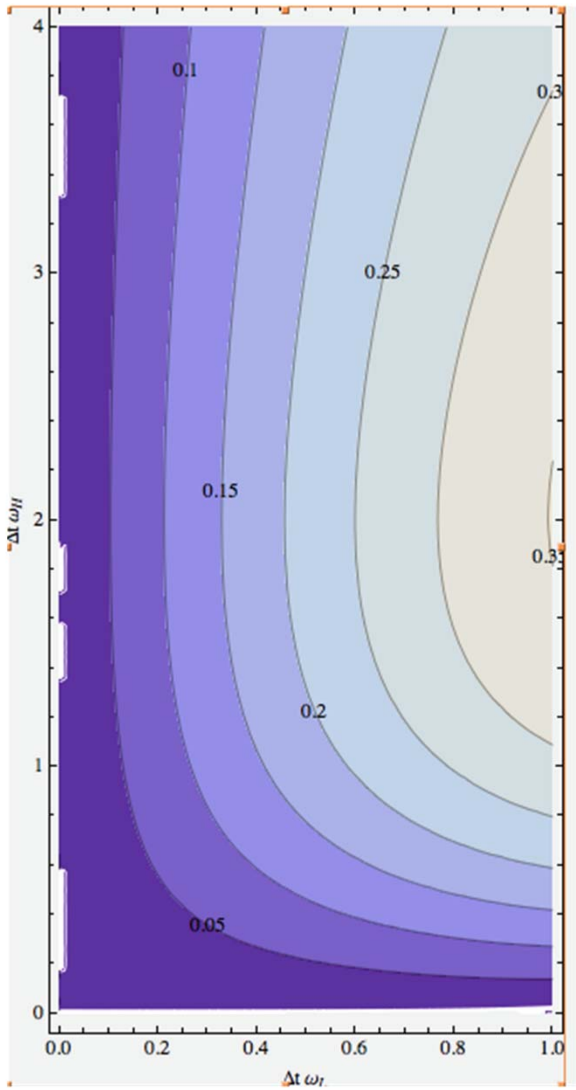
Truncation Error:

$$
\frac{u^{n+1} - u^{Exact}}{u^n} = \frac{1}{2}\omega_H \left((1-2\beta)\omega_H - 2(\beta-1)\omega_L\right)\Delta t^2 + O\left(\Delta t^3\right)
$$

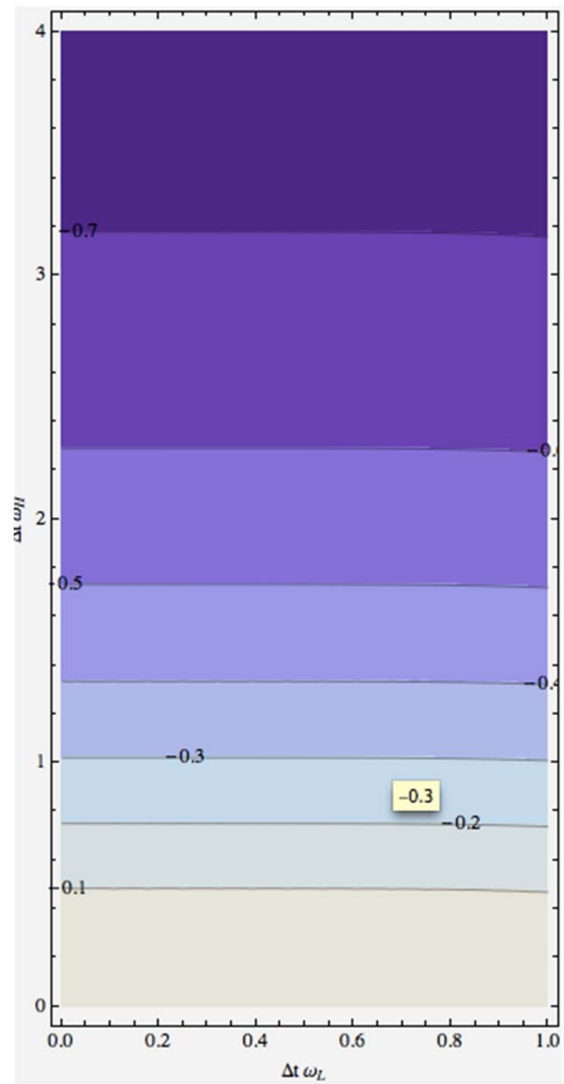
the accuracy is only 1st order

Plot of |A|-1

β=1/2: Crank-Nicolson
-> Absolutely unstable



β=1: backward Euler
-> Absolutely stable,but extremely dissipative



Summary

- Runge-Kutta 4^{th}-order scheme with semi-implicit time is
  - only of first order (with respect to fast modes)
  - absolutely unstable if Crank-Nicolson is used
  - absolutely stable if Backward Euler is used, but the numerical damping is too strong (more than halving on every time step)
- all of the above are consistent with what I found for the SPEEDY model.

Conclusion

- Runge-Kutta 4th-order scheme with semi-implicit time-stepping for gravity waves is impossible (either unstable or too dissipative)
- The accuracy becomes only of 1st order.

- Since the motivation for implementing Runge-Kutta 4th-order scheme is to produce a reference solution, I will not try to resolve this issue, and use the explicit scheme with small dt.

Stability analysis of Forward Euler "split physics" for Leapfrog, Lorenz 4-cycle and RK4

Toy model: linear advection-diffusion equation

- Undiscretized equation:

$$\frac{du}{dt} = i\omega u - \beta u$$

- The first term of the RHS simulates the dynamics of AGCM, and the second the physics.

Discretization: Leapfrog(dyn) + Euler(phys)

- Discretized equation:

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = i\omega u^n - \beta u^{n-1}$$

- The amplification factor *A* satisfies the following equation:

$$\frac{A^2 - 1}{2\Delta t} = i\omega A - \beta$$

$$\Rightarrow A = i\omega\Delta t \pm \sqrt{1 - (\omega\Delta t)^2 - 2\Delta t\beta}$$

- (+) and (-) correspond, respectively, to physical and computational modes.
- The scheme is stable if the maximum of the moduli of them is less than 1.

Discretization: Leapfrog for both dyn & phys

- Discretized equation:

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = i(\omega + i\beta)u^n$$

- The amplification factor *A* satisfies the following equation:

$$\frac{A^2 - 1}{2\Delta t} = i(\omega + i\beta)A$$

$$\Rightarrow A = i(\omega + i\beta)\Delta t \pm \sqrt{1 - (\omega + i\beta)^2 \Delta t^2}$$

- (+) and (-) correspond, respectively, to physical and computational modes.
- The scheme is stable if the maximum of the moduli of them is less than 1.

Discretization: Lorenz 4-cycle (dyn) + Euler(phys)

- Discretized equation:

$$u^{n+N} = u^* - N\beta\Delta t u^n,$$
$$\text{where} \quad u^* = u^n \left(\sum_{k=0}^{N} \frac{1}{k!} (iN\omega\Delta t)^k \right)$$

- Amplification factor (per time step):

$$A = \left\{ \left(\sum_{k=0}^{N} \frac{1}{k!} (iN\omega\Delta t)^k \right) - \beta N\Delta t \right\}^{1/N}$$

Discretization: Lorenz 4-cycle for both dyn & phys

- Discretized equation:

$$u^{n+N} = u^n \left(\sum_{k=0}^{N} \frac{1}{k!} (iN(\omega + i\beta)\Delta t)^k \right)$$

- Amplification factor (per time step):

$$A = \left(\sum_{k=0}^{N} \frac{1}{k!} (iN\omega\Delta t)^k \right)^{1/N}$$

Discretization: RK4(dyn) + Euler(phys)

- Discretized equation:

$$u^{n+1} = u^* - \beta \Delta t u^n,$$

$$\text{where} \quad u^* = u^n \left(\sum_{k=0}^{N} \frac{1}{k!} (i\omega \Delta t)^k \right)$$

- Amplification factor :

$$A = \left(\sum_{k=0}^{N} \frac{1}{k!} (i\omega \Delta t)^k \right) - \beta \Delta t$$

Discretization: RK4 for both dyn&phys

- Discretized equation:

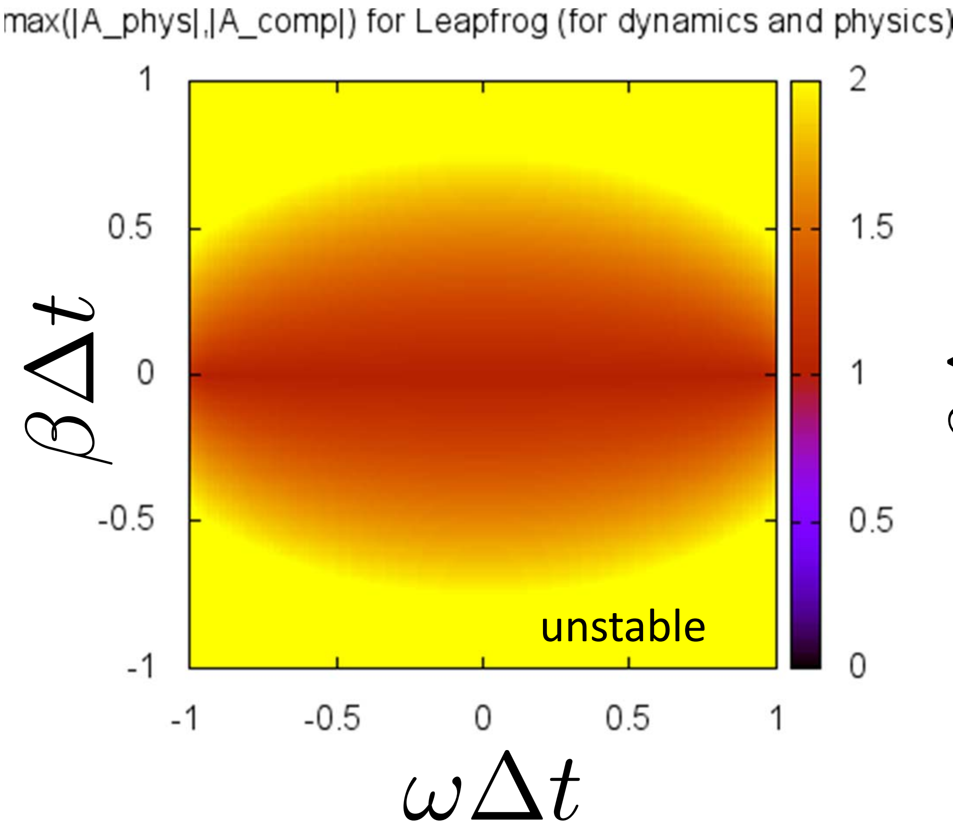
$$u^{n+1} = u^n \left(\sum_{k=0}^{N} \frac{1}{k!} (i(\omega + i\beta)\Delta t)^k \right)$$

- Amplification factor (per time step):

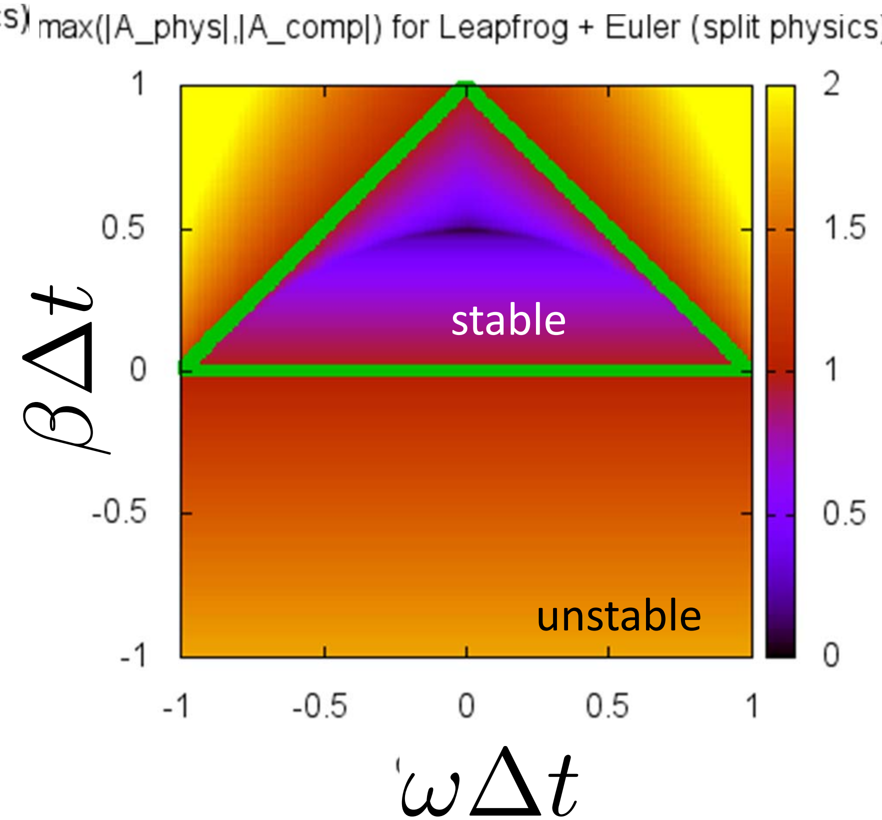
$$A = \sum_{k=0}^{N} \frac{1}{k!} (i\omega \Delta t)^k$$

Leapfrog: "split physics" stabilizes the otherwise absolutely unstable scheme

Leapfrog for both dyn & phys: absolutely unstable

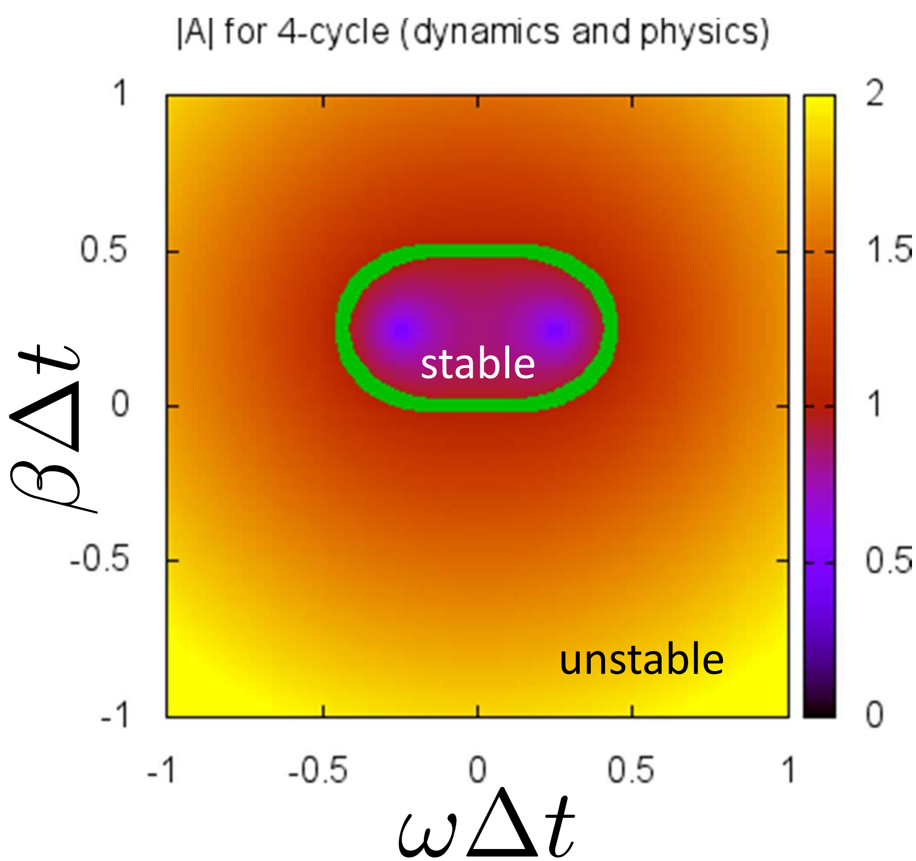


Leapfrog (dyn) + Euler (phys): Stable within the triangle

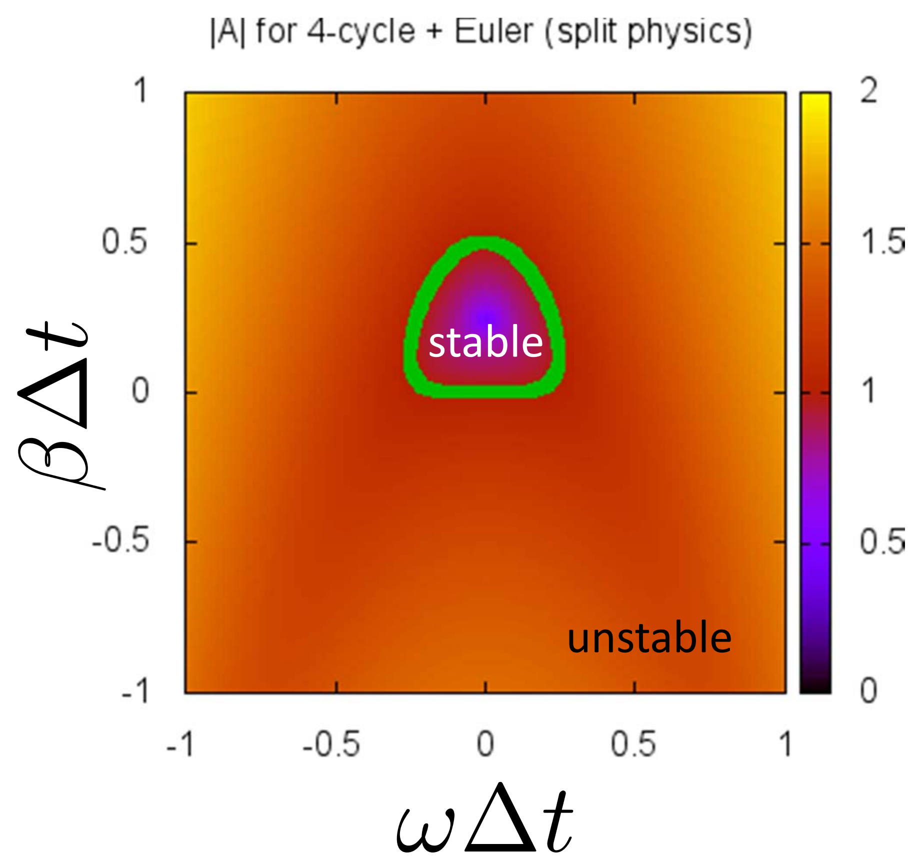


Lorenz 4-cyle:
"split physics" destabilizes the scheme

Lorenz 4-cycle for dyn & phys: absolutely unstable

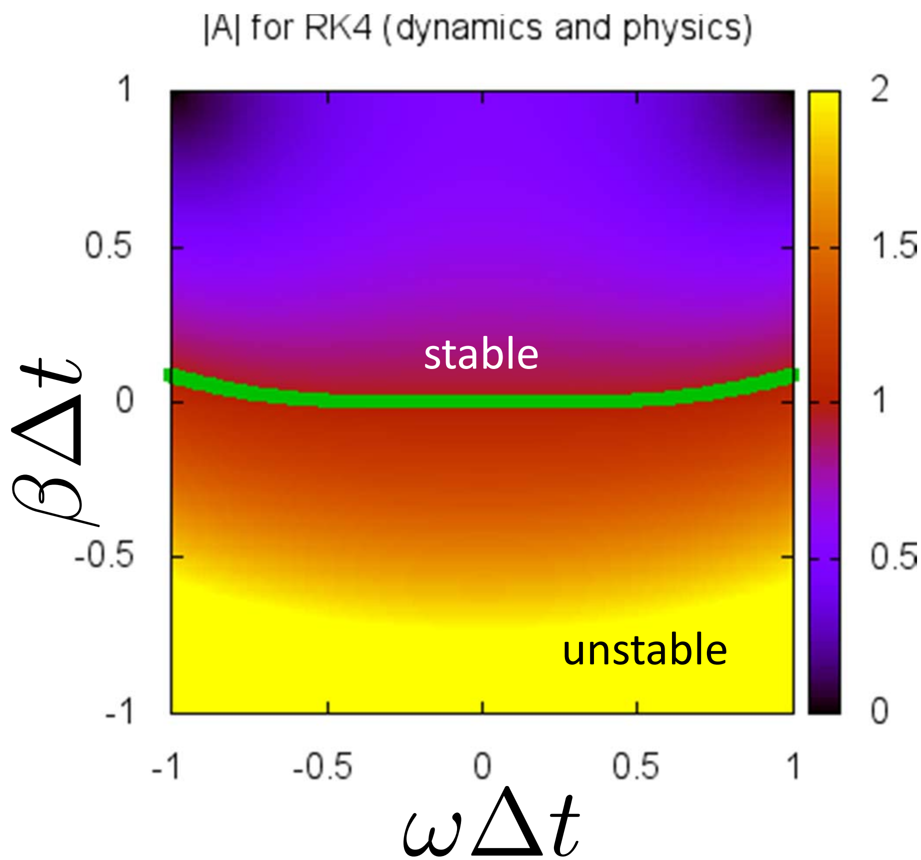


L4-cycle (dyn) + Euler (phys): Stable within the triangle

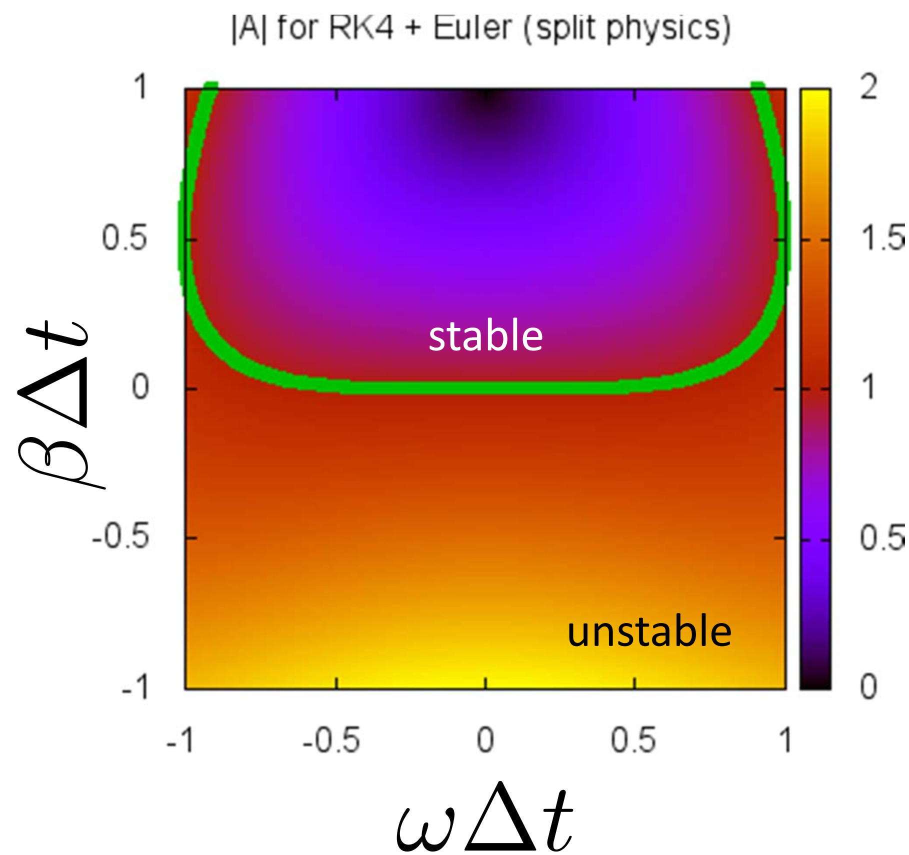


RK4:
again, "split physics" destabilizes the scheme

RK4 for dyn & phys:
absolutely unstable



RK4 (dyn) + Euler (phys):
Stable within the triangle



Summary

- For leapfrog, doing “split physics” works very well (stabilizes the scheme)
- However, for Lorenz 4-cycle (and equivalently, for RK4), “split physics” acts to destabilized the scheme.
- The latter is consistent with what I found by doing “split physics” with SPEEDY model.

The Bug (1)

Leapfrog (default)

```
PROGRAM agcm
CALL iniall ()
IDAY=0; CALL FORDATE()
 CALL STEPONE() ! 1st step by Euler Forward
 DO  ! loop over a month
   DO ! loop over a day
     CALL FORDATE()
     CALL STLOOP() ! integrate for a day
   END DO
END DO
```

N-cycle (with bug)

```
PROGRAM agcm
CALL iniall ()



DO  ! loop over a month
   DO ! loop over a day
     CALL FORDATE()
     CALL STLOOP() ! integrate for a day
   END DO
END DO
```

The Bug (2)

N-cycle (fixed)

```
PROGRAM agcm
CALL iniall ()
IDAY=0; CALL FORDATE()

DO  ! loop over a month
  DO ! loop over a day
    CALL FORDATE()
    CALL STLOOP() ! integrate for a day
  END DO
END DO
```

N-cycle (with bug)

```
PROGRAM agcm
CALL iniall ()


DO  ! loop over a month
  DO ! loop over a day
    CALL FORDATE()
    CALL STLOOP() ! integrate for a day
  END DO
END DO
```