University of Leicester

Analytical determination of the carbon cycle equilibrium in coupled climate and carbon cycle models

Jörg Kaduk, Andrea Hurtado De Mendoza Rosales, Hao Wang

Department of Geography, University of Leicester
Centre for **Landscape** and **Climate Research**

Outline

- Motivation
- Some maths background
- Application
- Results
- Summary
- Outlook

University of Leicester

Motivation

- Carbon cycle models: predict future carbon (C) fluxes
- Simulation started with some initial state of the C pools (C in live biomass, litter, soil)
- Simulation results should reflect the response of the C system to the environmental variability during the simulation period
- Selection of initial state of C pools is crucial to avoid spurious C fluxes that result from adjustment of the system to “inappropriate” initial state
- Initial state often chosen as steady state of system (equilibrium)

University of Leicester

Motivation

- **Equilibrium**
- **Difficult** to determine for a complex system
- Run model with the same boundary conditions and forcing for a long time and wait for convergence to steady state (100s to 1000s of years): **Expensive**
- Not a good initial value for simulations from now into the future, rather for, say 19th century: simulation into the future needs to capture current disequilibrium of the C system from the past 150-200 years
- Need another 150-200 years simulation years - expensive

University of Leicester

Motivation

- **Equilibrium**
- Expensive
- Limits exploration of carbon cycle system
- Data assimilation (data assimilation of net C flux needs well established C state of the system)

- Direct determination from system dynamics?
- Needs to be computationally efficient

University of Leicester

Maths background

- Carbon cycle systems are being represented by

$$\frac{dx}{dt}(t) = A(t,x) + g(t), x(t_0) = x_0, t \geq t_0$$

- Frequently linear:

$$\frac{dx}{dt}(t) = A(t)x(t) + g(t), x(t_0) = x_0, t \geq t_0$$

- x are the carbon pools, A environmental effects, g input, e.g. photosynthesis
- Soil respiration in JULES

$$d(\mathrm{DPM})/dt = \alpha_{dr}\Lambda_c - R_{\mathrm{DPM}}$$
$$d(\mathrm{RPM})/dt = (1-\alpha_{dr})\Lambda_c - R_{\mathrm{RPM}}$$
$$d(\mathrm{BIO})/dt = 0.46\cdot\beta R_s - R_{\mathrm{BIO}}$$
$$d(\mathrm{HUM})/dt = 0.54\cdot\beta R_s - R_{\mathrm{HUM}}$$

University of Leicester

Maths background

$$\frac{dx}{dt}(t) = A(t)x(t) + g(t), x(t_0) = x_0, t \geq t_0$$

- Solution:

$$x(t) = \Psi(t, t_0)x_0 + \int_{t_0}^{t} \Psi(t, s)\, g(s)ds$$

- $\Psi(t, t_0)$ is the state transition matrix

- For constant A:

$$\Psi(t, t_0) = exp(A(t - t_0)) = e^{A(t-t_0)}$$

- State transfer between time points

$$\Psi(t_2, t_0) = \Psi(t_2, t_1)\Psi(t_1, t_0)$$

University of Leicester

Maths background

- Equilibrium:
- Steady state such that after one year the C pools are the same as at the beginning of the year:

$$x_0 = x(t_0 + \omega) = \Psi(t_0 + \omega, t_0)x_0 + \int_{t_0}^{t_0+\omega} \Psi(t, s)\, g(s) ds$$

ω period of one year

- So, for a steady state:

$$x_0 = (Id - \Psi(t_0 + \omega, t_0))^{-1} \int_{t_0}^{t_0+\omega} \Psi(t_0 + \omega, s)\, g(s) ds$$

University of Leicester

Application

- Carbon cycle model is run time stepped with time step Δt
- A and g constant during time step: A_i and g_i

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-s)}\, g(s)ds$$

- From one time step to the next:

$$x_{i+1} = e^{A_i \Delta t}x_i + g_i \int_{t_i}^{t_{i+1}} e^{A_i(t-s)}\, ds$$

- **Recursion**

University of Leicester

Application

- Resolve recursion:

$$\Psi(t_0 + \omega, t_0) = \prod e^{A_i \Delta t}$$

$$\int_{t_0}^{t_0+\omega} \Psi(t_0 + \omega, s)\, g(s) ds = \sum g_i \int_{t_i}^{t_{i+1}} e^{A_i (t-s)}\, ds$$

- Computationally very efficient:
  - For n C pools add only one *n x n* matrix and one *n* vector to model code for each grid cell and PFT.
  - No need to output additional data or save any additional data in file for later calculations.

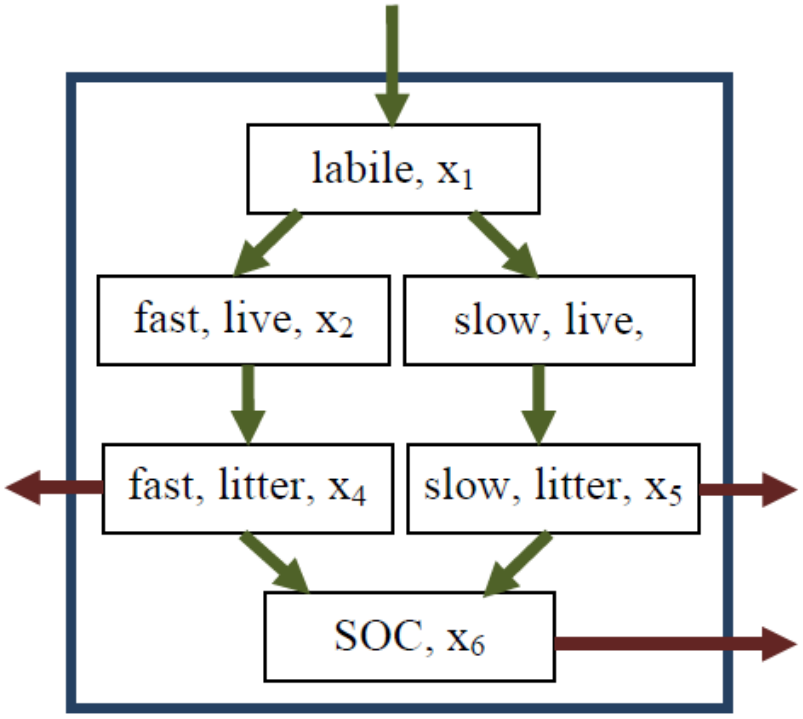
University of Leicester

Application

Example

- Synthetic system

$$A = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 & 0 \\ 0 & a_{42} & 0 & a_{44} & 0 & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} & 0 \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} \end{pmatrix}$$



a_{11}=-abs(max(10^{-5},sin(t*pi/nt)/nt))
a_{22}=0.25*a_{11}; a_{21}=-0.5*a_{11}; a_{33}=1/16*a_{11}; a_{31}=-0.5*a_{11};

g(t)=5000*(max(0,sin(t*pi/nt)*sin(t*2*pi/(24*60)-pi/2))/nt 0 0 0 0 0)^T

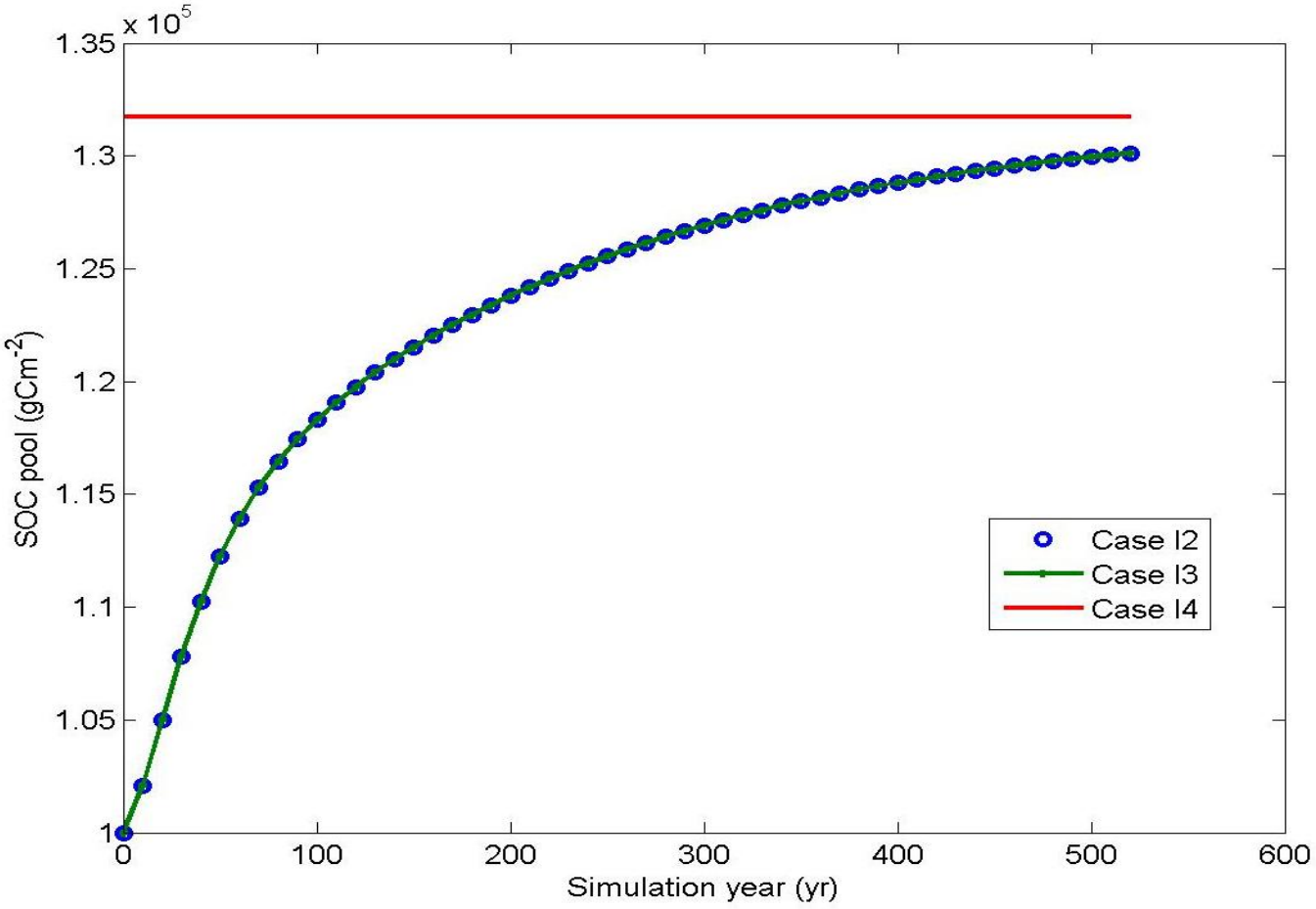
University of Leicester

Results

Case	Short description	Simulation years $\Delta x_T<0.01$	cputime in matlab (s)	C_{Tot} von qss	C_{Tot} when $\Delta x_T<0.01$
I1	Numerical higher order solver, ode45	528	809	152883.70	151333.09
I2	Explicit forward iteration with A and g	528	336	152814.02	151269.73
I3	$\Psi(T,0)$ and $\tilde{x}$ determined using similarity transform, eig(T) and inv(T), iterated for all simulation years	528	2.4	152814.02	151269.72
I4	$\Psi(T,0)$ and $\tilde{x}$ determined using similarity transform, eig(T) and inv(T), solved for qss	1	2.4	152814.02	

Results

- "SOC" pool x_6



University of Leicester

Results

- Different methods deliver the same steady state
- New approach needs less than 1% of the time of other methods for steady state
- With a more computationally complex A and g (e.g. JULES) absolute time savings would be very significant

University of Leicester

Results

Non linear example

- Use linearization: Taylor-expansion

- Simple canopy integration:

$$p_0*(1-e^{-kLAI})/k$$

- p_0 top leaf photosynthesis
- Specific leaf area links to fast turnover pool:

$$f(x_2(t)) = \frac{1}{k}(1 - e^{-k\gamma x_2(t)})$$

- Linearize f

$$f(x_2(t)) = f(x_2^0) + f'(x_2^0)(x_2(t) - x_2^0)$$
$$= \frac{1}{k}(1 + exp(-k\gamma x_2^0)) + \gamma exp(-k\gamma x_2^0)(x_2(t) - x_2^0)$$

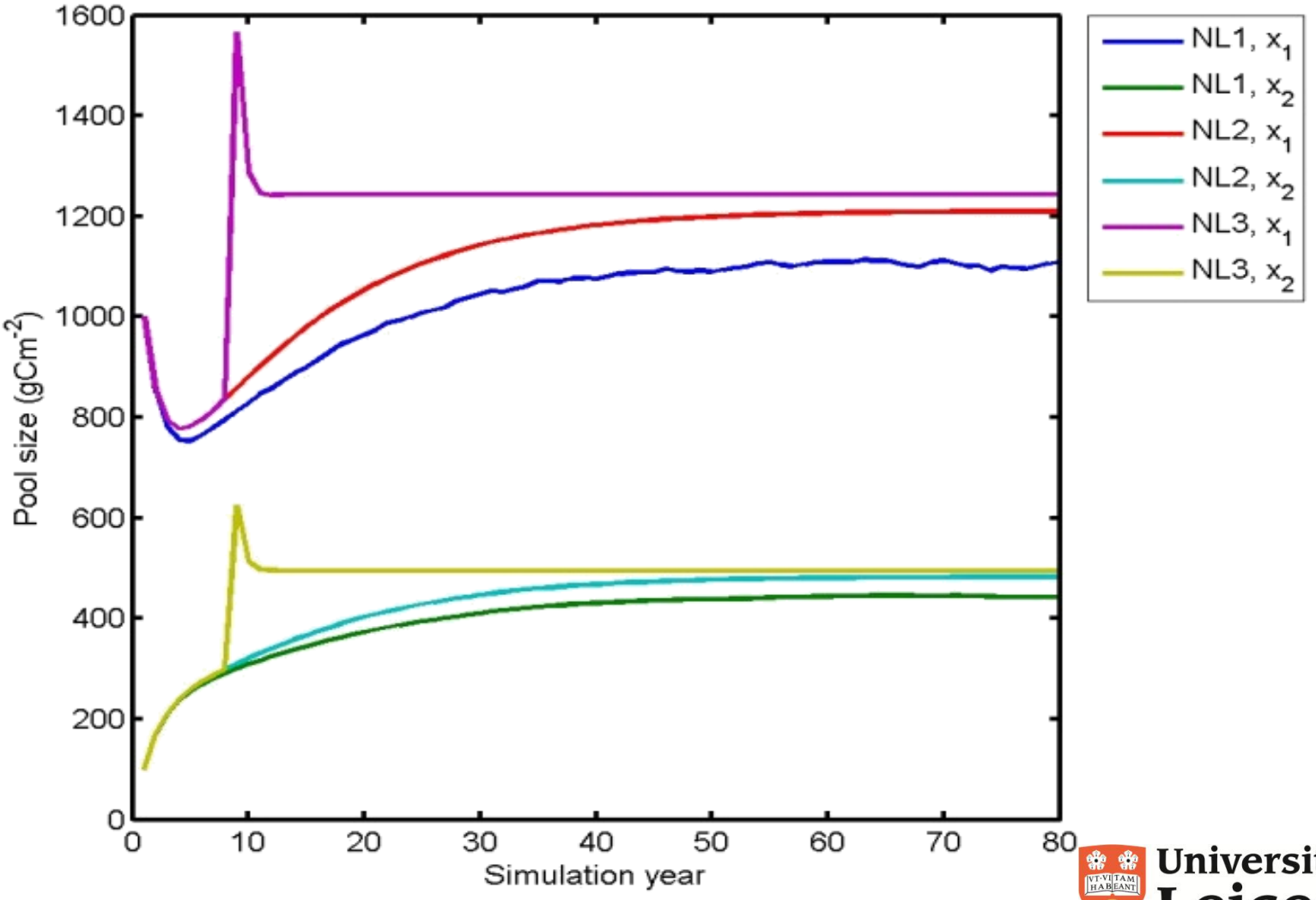
University of Leicester

Results

	NL1	NL2	NL3
Short description	Numerical method, ode 45	Explicit forward iteration with A and g	8 years NL2, then 73 years of using $\Psi(T,0)$ and $\tilde{x}$ using eig(T) and inv(T), solving for qss
No years for qss estimate, yr_{qss}	5000	5000	80
C_{Tot} of qss (gCm^{-2})	48993.1	53149.7	54479.0
No of years until $\Delta x_T < 0.01$, $yr_{0.01}$	1020	999	11
C_{Tot} ($yr_{0.01}$) (gCm^{-2})	49489.6	53628.0	54608.1
dCTot/dt ($gCm^{-2}yr^{-1}$) in year $yr_{0.01}$	9.0	2.7	123.2
time for $yr_{0.01}$ yrs (s)	267.7	378.8	15.6

Results

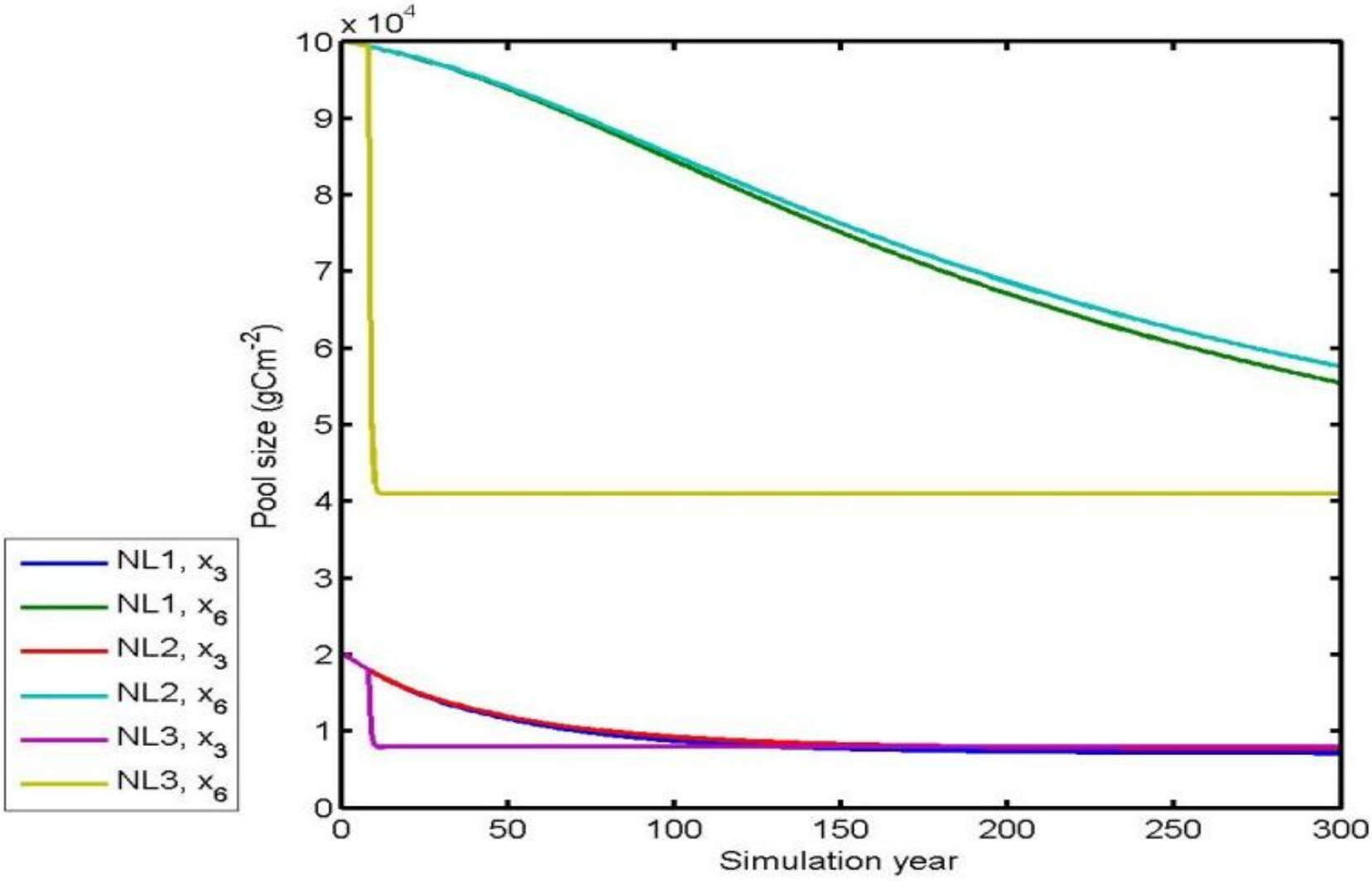
- Fast pools x_1 & x_2



University of Leicester

Results

- Slow pools x_3 & x_3



Results

- Different methods deliver different steady states
- New method needs for steady state of linearized system less than 5% of the time required by 1st order forward method
- Steady state difference between 1st order forward and new method is due to linearization
- Running 210 years 1st order forward method after steady state estimation from new method for steady state of 1st order forward method still needs less than 25% of the time required for 1st order forward method

University of Leicester

Summary

- Steady state of time varying linear system can be calculated directly from data of one simulation year
- Time savings are 99 – 99.99% for linear systems
- Virtually no additional resources (computational or manpower) required
- Can be applied to non-linear systems work through linearizations of the carbon system
- Difference between linearized and original system can be reduced by additional simulations
- Time savings of more than 75% possible
- Real time savings very large as A and g expensive

University of Leicester

Outlook

- For new method model carbon cycle system could be split into linear and nonlinear part
- New method facilitates exploration of climate carbon cycle feedbacks
- Allows improved data assimilation
- Currently being implemented in JULES
- Summited to GMDD very soon

Contact: jk61@le.ac.uk

University of Leicester