MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Cellular Neural Networks for Data Assimilation Asymptotic behavior

Haroldo F. de Campos Velho[1], César M. O. Oliveira Jr.[2], Antonio M. Saraiva[2], Alexandre C. B. Delbem[2], Gerônimo G. Z. Lemos[3]

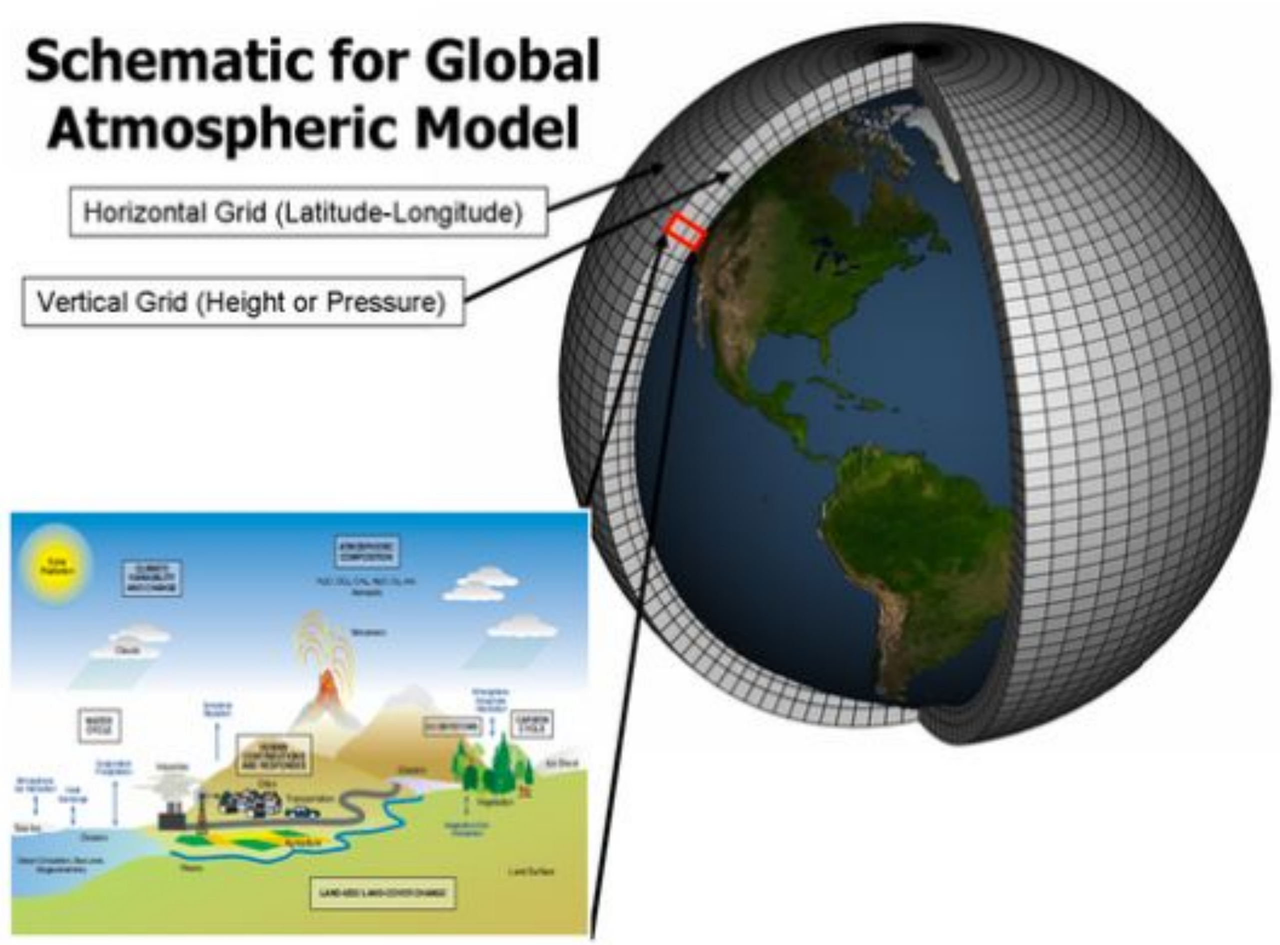
[1] INPE: National Institute for Space Research, Brazil

[2] USP: University of Sao Paulo - Brazil

Thanks for 21th EnKF workshop organizers

Weather and climate prediction

2. Dynamical core (Navie-Stokes equation "solver")



Weather and climate prediction

- Dynamical core (Navie-Stokes equation "solver")

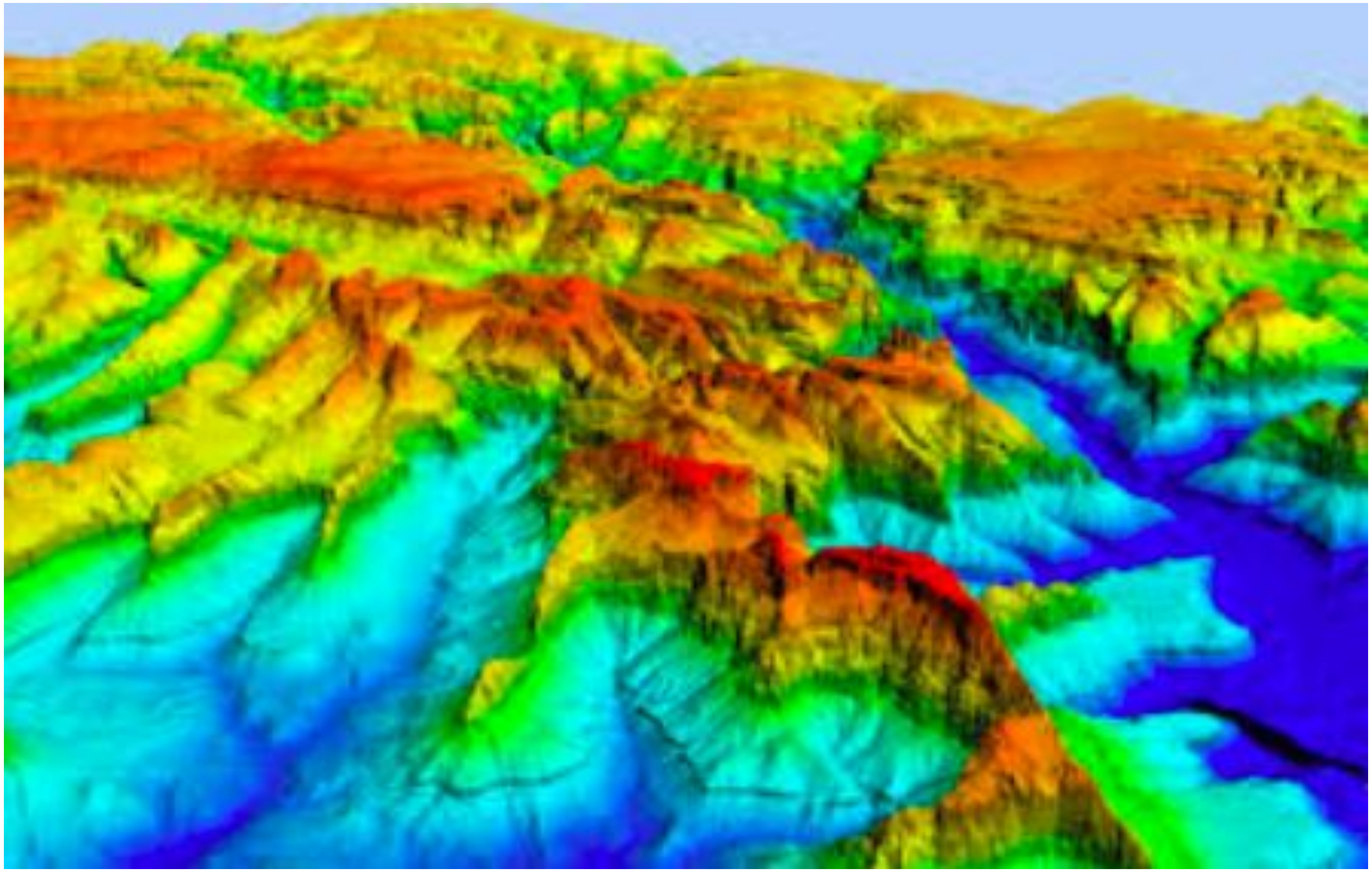
$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= -\nabla \cdot (\zeta + f)\vec{v}_H - \vec{k} \cdot \nabla \times \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) \\
\frac{\partial D}{\partial t} &= \vec{k} \cdot \nabla \times (\zeta + f)\vec{v}_H - \nabla \cdot \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) - \nabla^2 \left(\phi + \frac{\vec{v}_H \cdot \vec{v}_h}{2} \right) \\
\frac{\partial T}{\partial t} &= -\nabla \cdot (T\vec{v}_H) + TD + \dot{\sigma}\gamma - \frac{RT}{c_p}\left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} + H_T \right) \\
\frac{\partial q}{\partial t} &= -\vec{v}_H \cdot \nabla q - D - \frac{\partial \dot{\sigma}}{\partial \sigma} \\
\sigma \frac{\partial \phi}{\partial \sigma} &= -RT \\
\frac{\partial r}{\partial t} &= -\nabla \cdot (r\vec{v}_H) + rD - \dot{\sigma}\frac{\partial r}{\partial \sigma} + M
\end{aligned}
$$

{with: $q = \log(p_0)$}

{with: $\phi = gh$; and: $\sigma = p/p_0$}

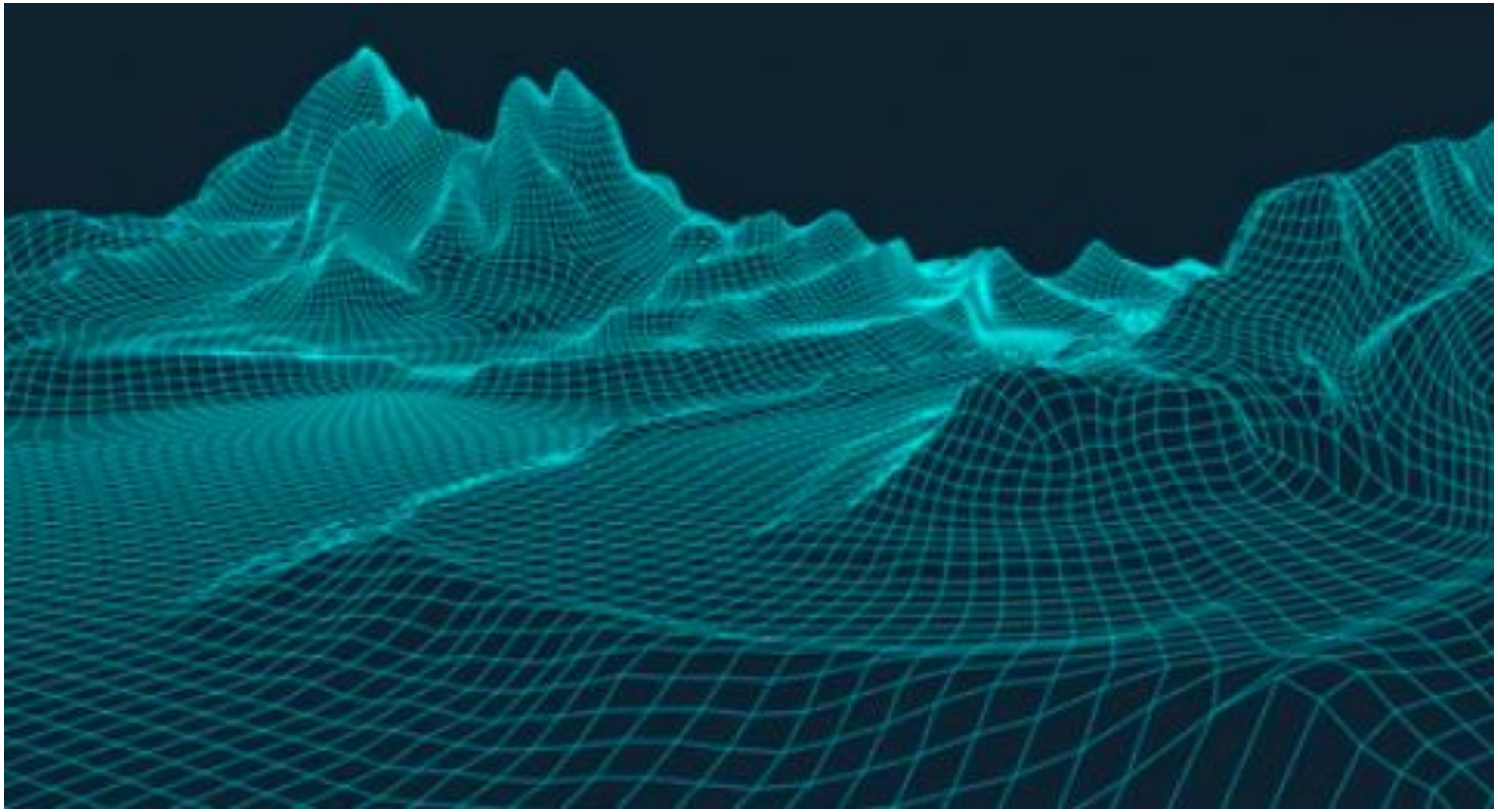
Weather and climate prediction

1.1 Topography data / surface covering map (NDVI)



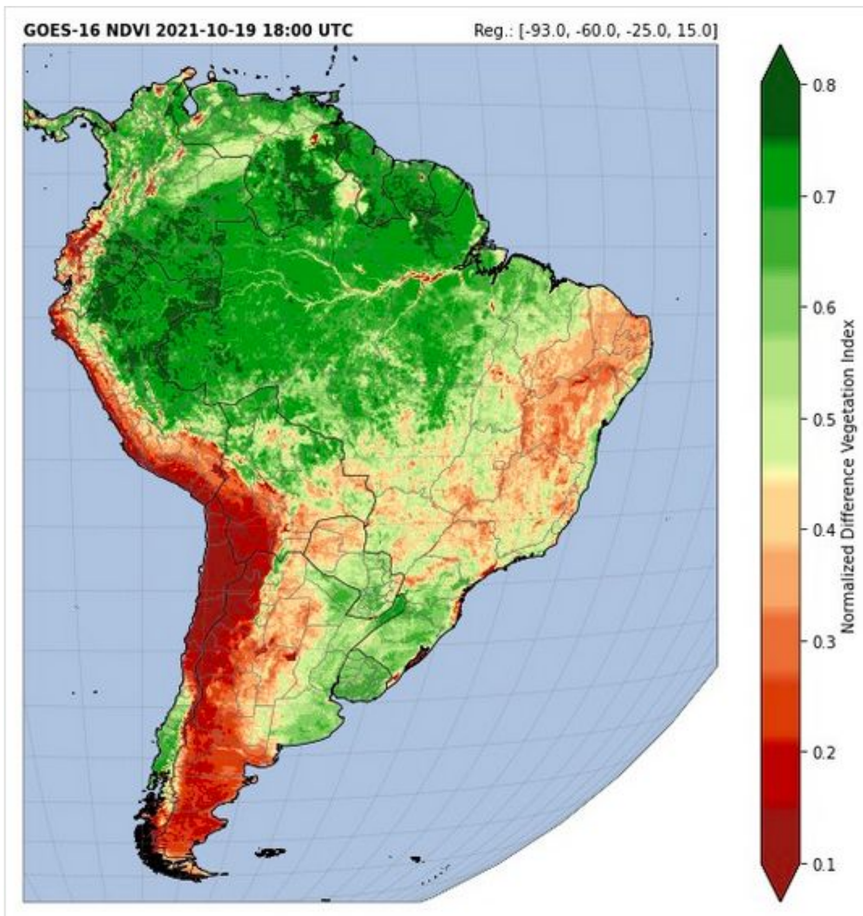
Weather and climate prediction

1.1/1.2 Topography data



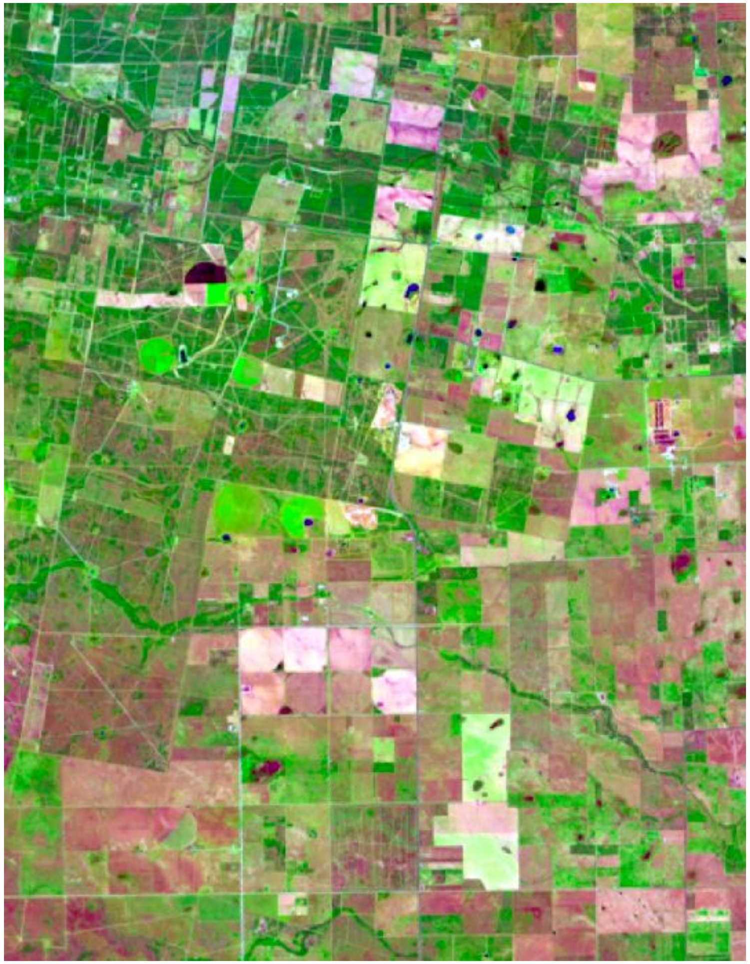
Weather and climate prediction

1.2 NDVI mapping



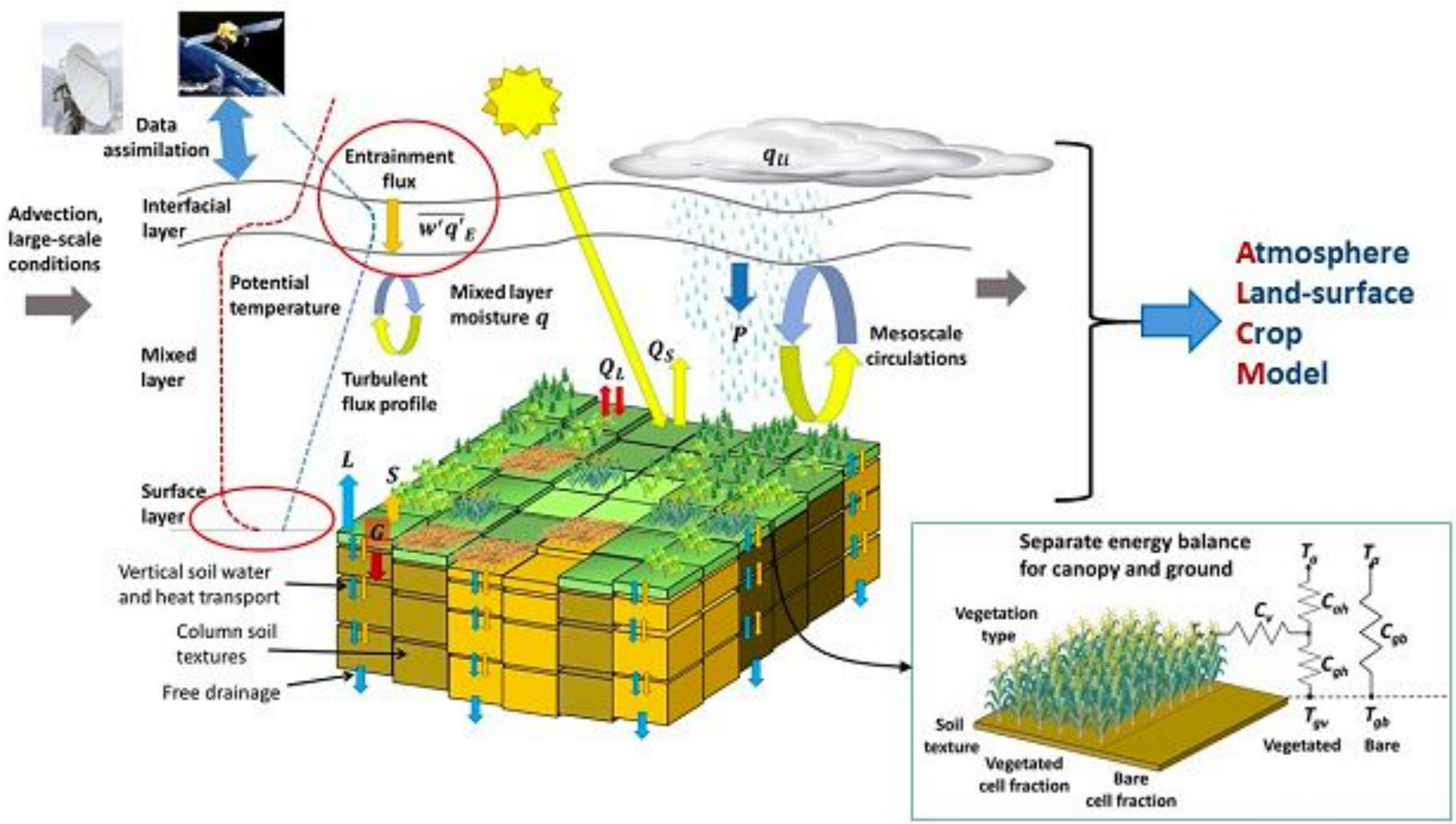
Weather and climate prediction

1.2 NDVI mapping



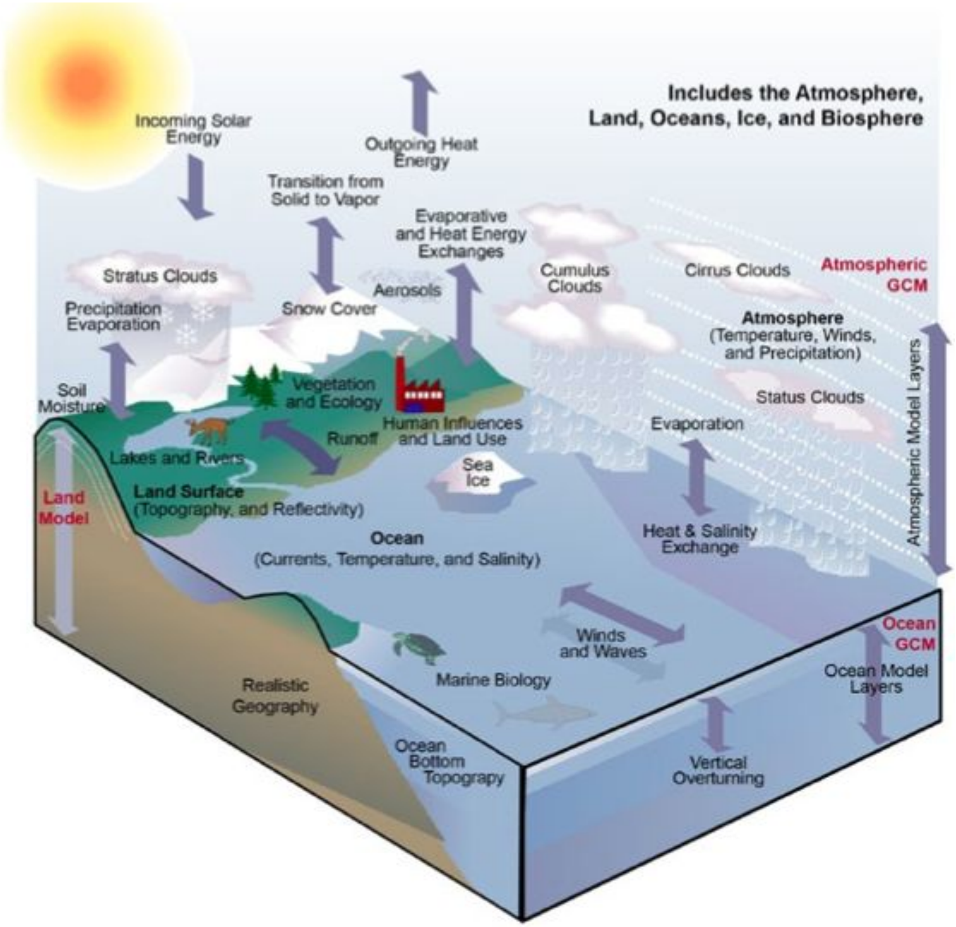
Weather and climate prediction

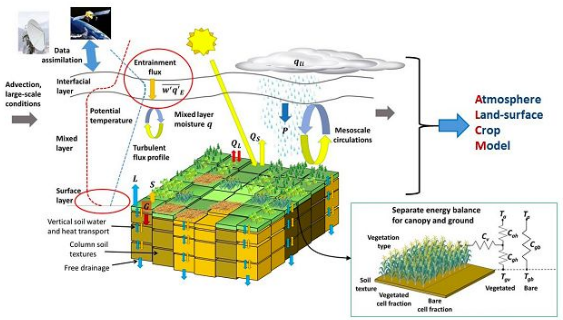
1.3 Soil type and soil moisture:



Weather and climate prediction

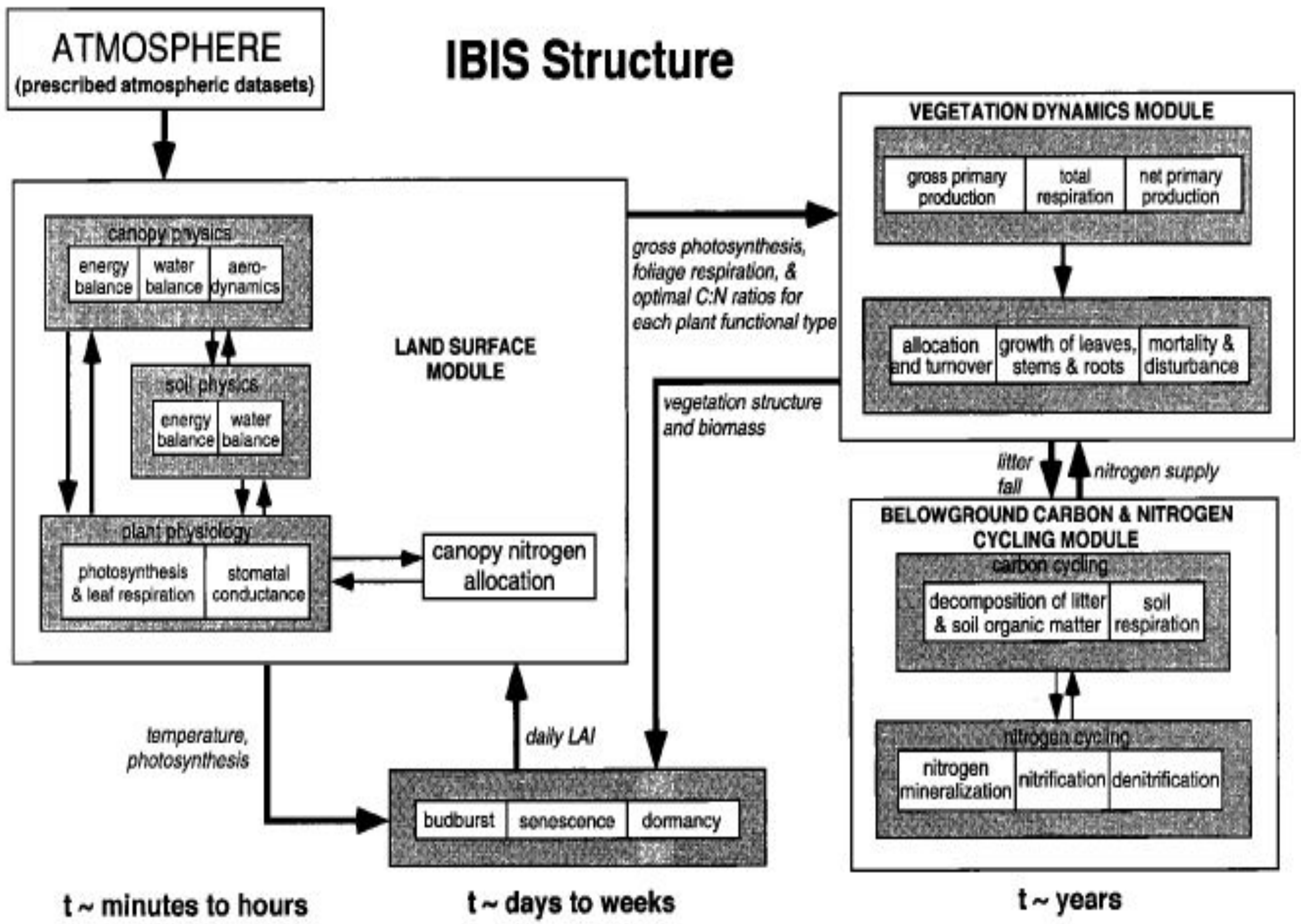
3.1 Model physics: Surface model





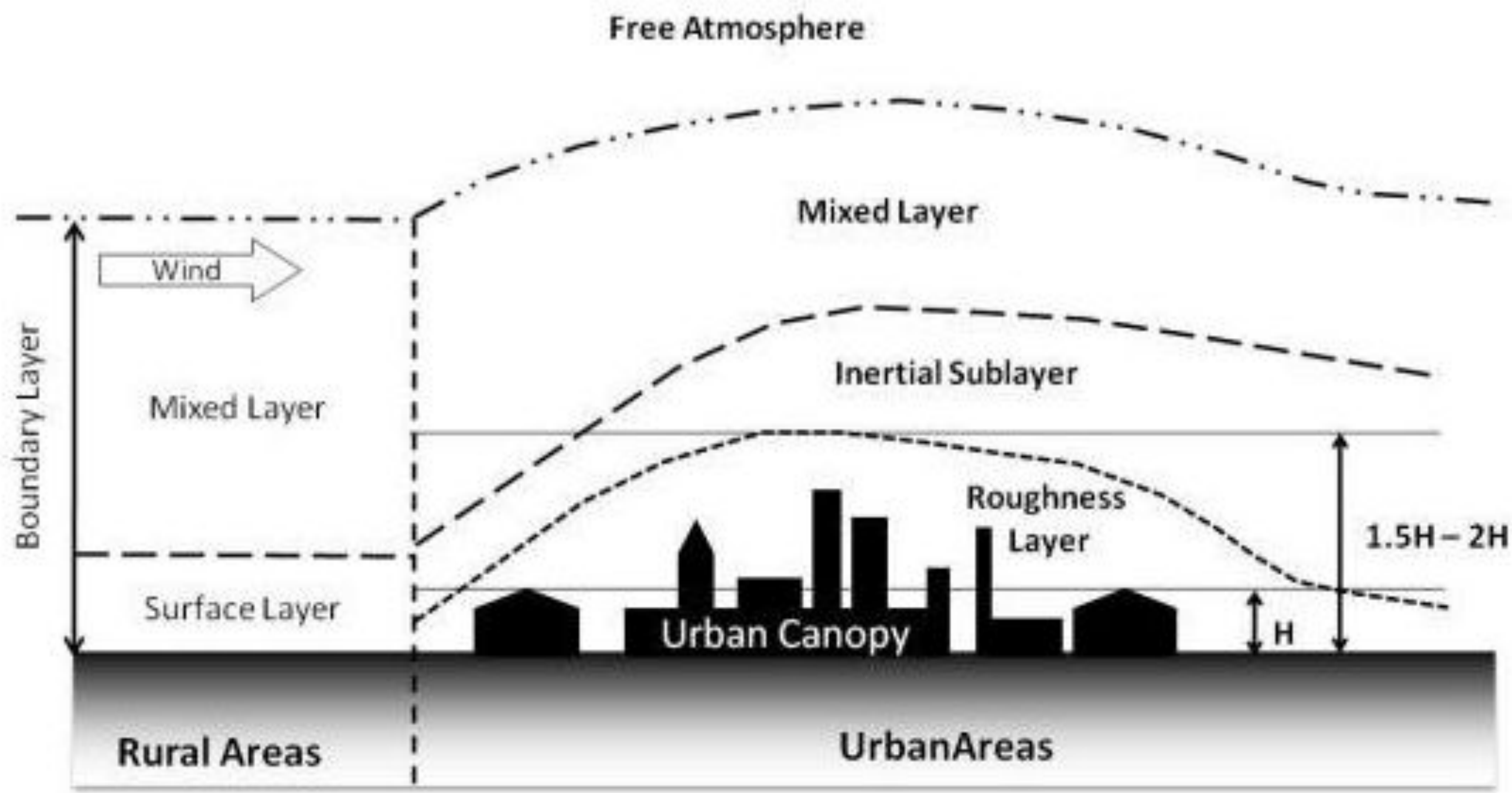
Weather and climate prediction

3.1 Model physics: Surface model



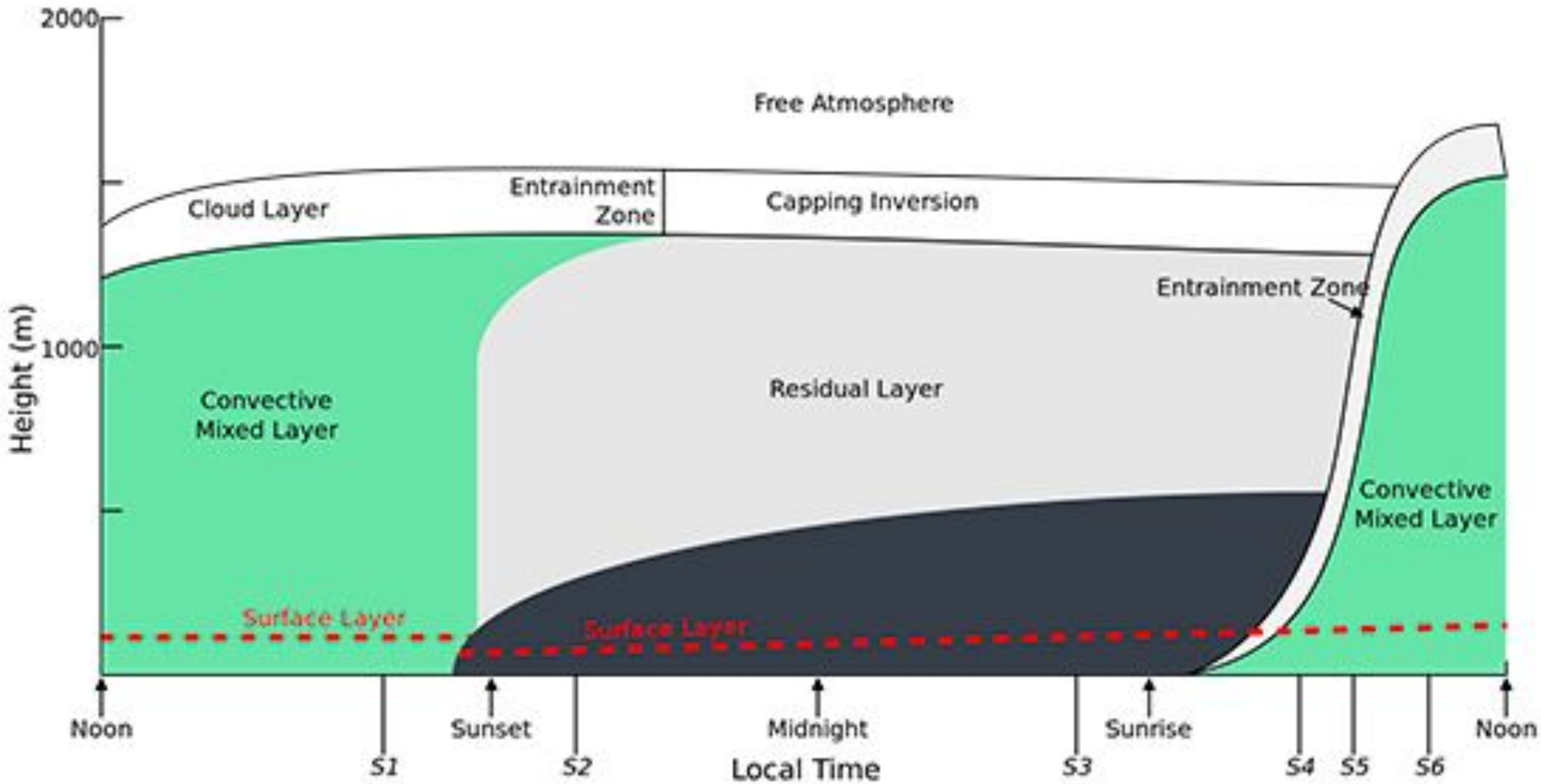
Weather and climate prediction

3.2 Model physics: Turbulence model



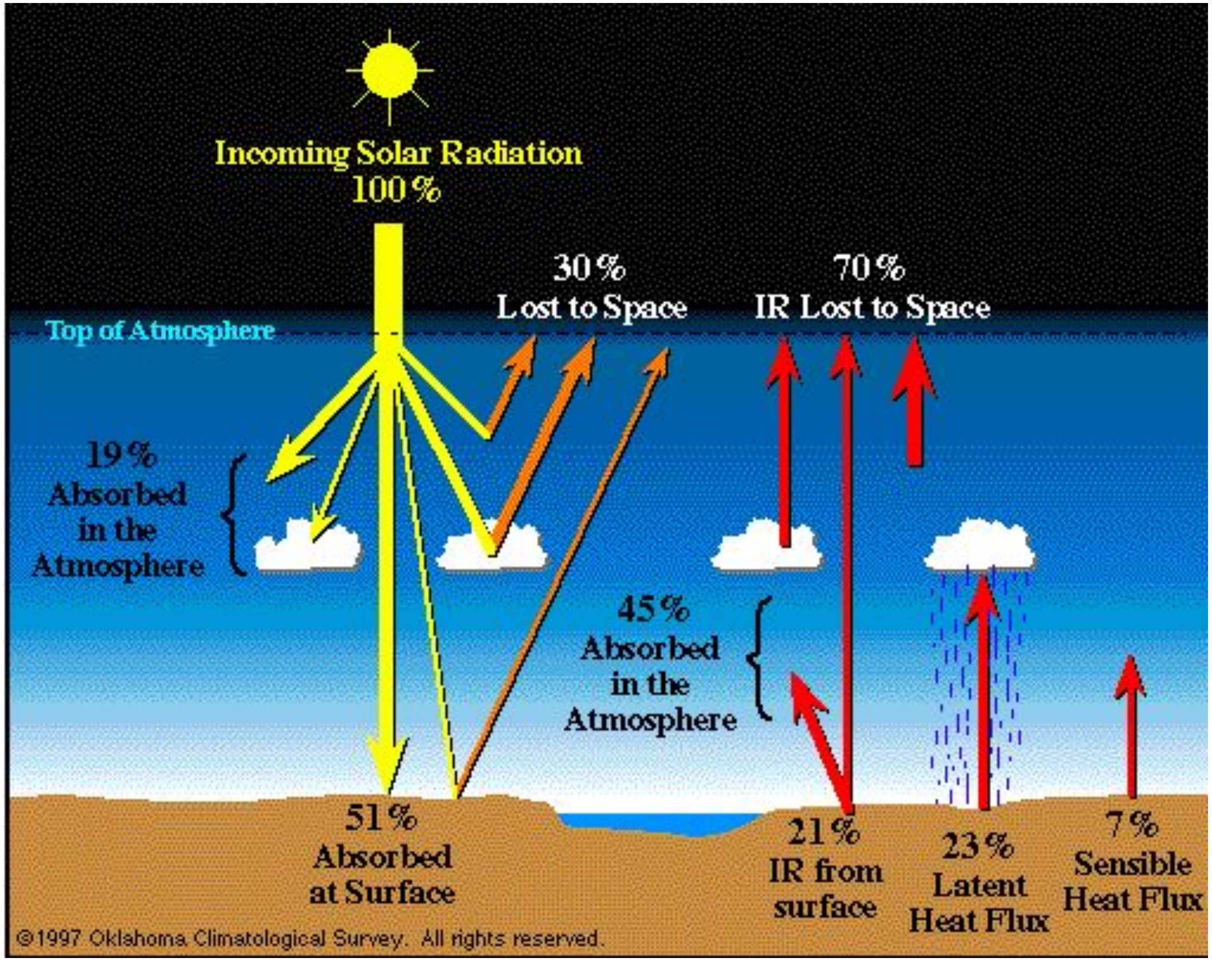
Weather and climate prediction

3.2 Model physics: Turbulence model



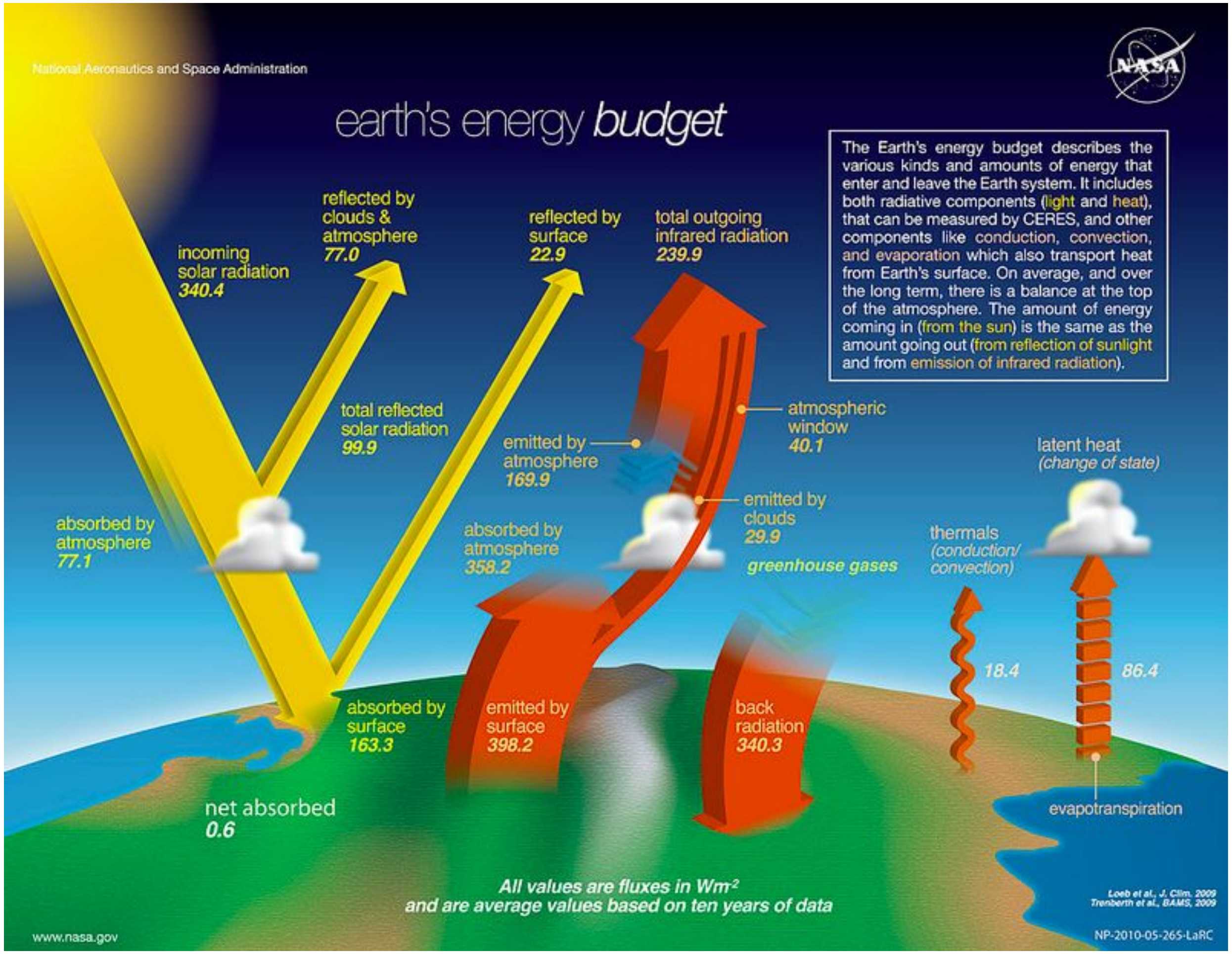
Weather and climate prediction

3.3 Model physics: Radiative transfer



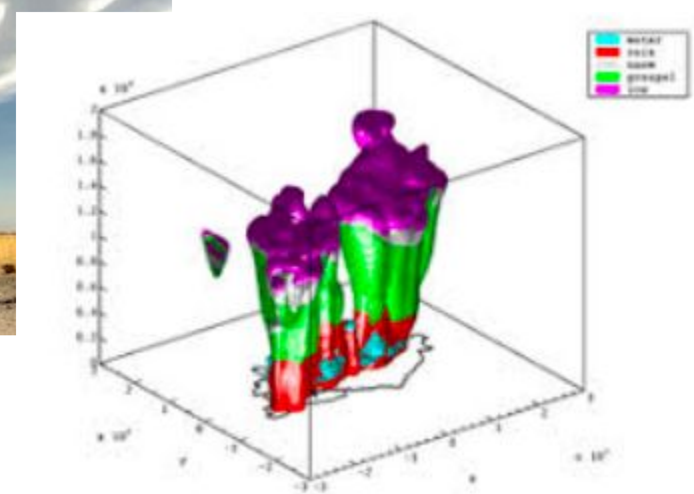
Weather and climate prediction

3.3 Model physics: Radiative transfer



Weather and climate prediction

3.3 Model physics: Cloud and precipitation



Weather and climate prediction

1. We have our mathematical model.
2. We have our representation for physical processes.
3. We have the representation of surface covering.
4. We have our numerical method.

Weather and climate prediction

1. We have our mathematical model.
2. We have our representation for physical processes.
3. We have the representation of surface covering.
4. We have our numerical method.

Are we prepare to do our numerical weather prediction?

Weather and climate prediction

1. We have our mathematical model.
2. We have our representation for physical processes.
3. We have the representation of surface covering.
4. We have our numerical method.

Are we prepare to do our numerical weather prediction?

No! Not at all. We do not have the initial condition!

Data assimilation: what's that?

Data assimilation cycle

- Machine learning (ML) algorithms for data assimilation.

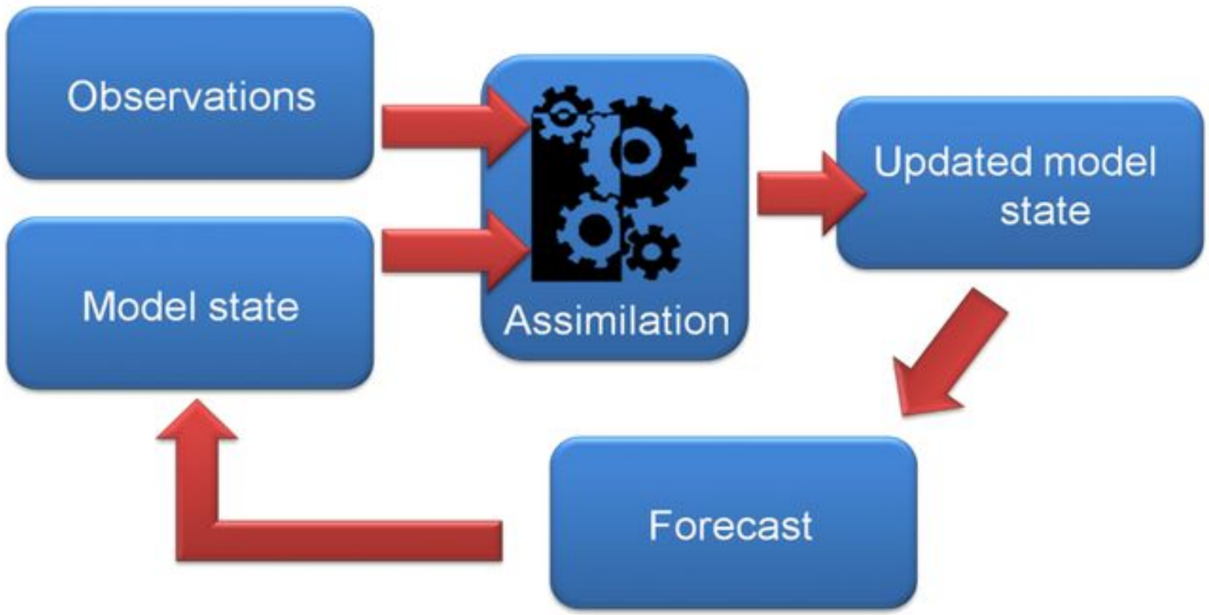
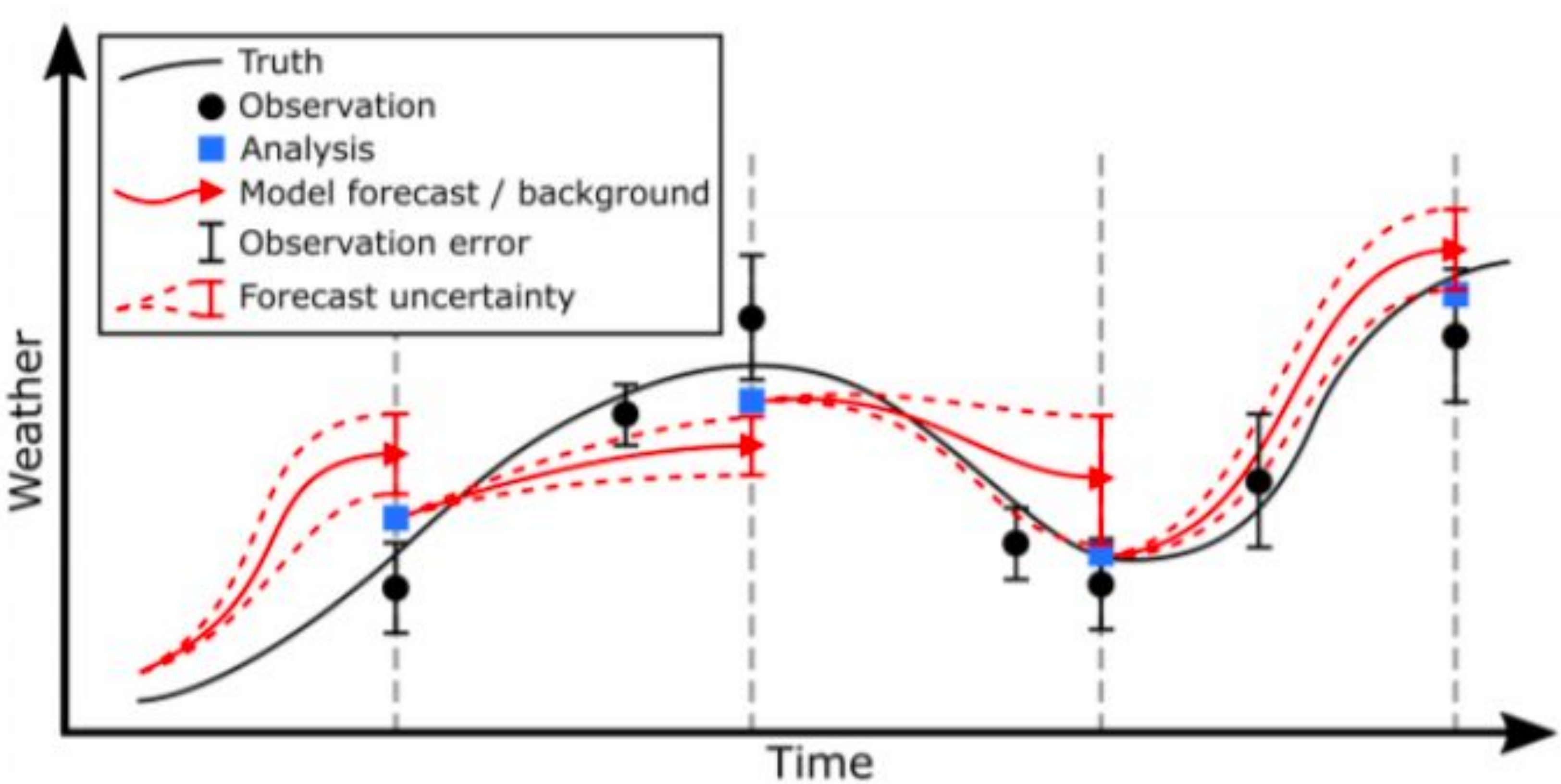


Figure Copyright S. L. Dance - University of Reading 2022
Used with permission

Data assimilation cycle



Weather and climate prediction

- Numerical weather prediction needs a supercomputer!

$$\frac{\partial \zeta}{\partial t} = -\nabla \cdot (\zeta + f)\vec{v}_H - \vec{k} \cdot \nabla \times \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right)$$

$$\frac{\partial D}{\partial t} = \vec{k} \cdot \nabla \times (\zeta + f)\vec{v}_H - \nabla \cdot \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) - \nabla^2 \left(\phi + \frac{\vec{v}_H \cdot \vec{v}_h}{2} \right)$$

$$\frac{\partial T}{\partial t} = -\nabla \cdot (T\vec{v}_H) + TD + \dot{\sigma}\gamma - \frac{RT}{c_p}\left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} + H_T \right)$$

$$\frac{\partial q}{\partial t} = -\vec{v}_H \cdot \nabla q - D - \frac{\partial \dot{\sigma}}{\partial \sigma}$$

{with: $q = \log(p_0)$}

$$\sigma\frac{\partial \phi}{\partial \sigma} = -RT$$

{with: $\phi = gh$; and: $\sigma = p/p_0$}

$$\frac{\partial r}{\partial t} = -\nabla \cdot (r\vec{v}_H) + rD - \dot{\sigma}\frac{\partial r}{\partial \sigma} + M$$

Weather and climate prediction

- Initial condition: **more than 50% of CPU-time** is taken!

$$\frac{\partial \zeta}{\partial t} = -\nabla \cdot (\zeta + f)\vec{v}_H - \vec{k} \cdot \nabla \times \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right)$$

$$\frac{\partial D}{\partial t} = \vec{k} \cdot \nabla \times (\zeta + f)\vec{v}_H - \nabla \cdot \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) - \nabla^2 \left(\phi + \frac{\vec{v}_H \cdot \vec{v}_h}{2} \right)$$

$$\frac{\partial T}{\partial t} = -\nabla \cdot (T\vec{v}_H) + TD + \dot{\sigma}\gamma - \frac{RT}{c_p}\left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} + H_T \right)$$

$$\frac{\partial q}{\partial t} = -\vec{v}_H \cdot \nabla q - D - \frac{\partial \dot{\sigma}}{\partial \sigma}$$

{with: $q = \log(p_0)$}

$$\sigma\frac{\partial \phi}{\partial \sigma} = -RT$$

{with: $\phi = gh$; and: $\sigma = p/p_0$}

$$\frac{\partial r}{\partial t} = -\nabla \cdot (r\vec{v}_H) + rD - \dot{\sigma}\frac{\partial r}{\partial \sigma} + M$$

Data assimilation (DA) – methods

- **DA is a type of inverse problem (IP)**
- **Inverse problem: methods**
  - Regularized solution
  - Variational approaches
  - Bayesian approach: Kalman filter
  - Bayesian approach: Particle filter

Data assimilation (DA) – methods

- DA is a type of inverse problem (IP)
- Inverse problem: methods
  - Regularized solution
  - Variational approaches
  - Bayesian approach: Kalman filter
  - Bayesian approach: Particle filter
  - Machine learning approach

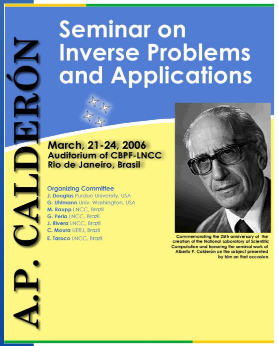
Inverse problem: Regularization method

COMPUTATIONAL & APPLIED MATHEMATICS
Volume 25, N. 2-3, pp. 1–24, 2006

ISSN 0101-8205
www.scielo.br/cam

A unified regularization theory: the maximum non-extensive entropy principle

HAROLDO F. DE CAMPOS VELHO[1], ELCIO H. SHIGUEMORI[2]
FERNANDO M. RAMOS[1] and JOÃO C. CARVALHO[3]



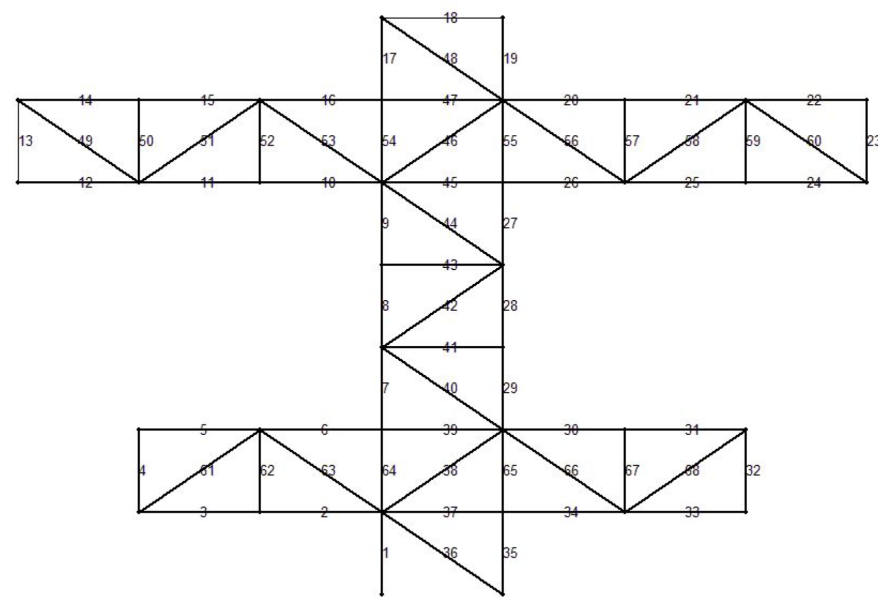
A.P. Calderon - Seminar on Inverse Problems and Applications – LNCC, Rio de Janeiro (RJ), March-2006

Inverse problem: variational approach

Damage detection: International Space Station

New methodology:

Hybrid method: Epidemic Genetic Algprithm + Variational



Available online at www.sciencedirect.com

ScienceDirect

PERGAMON

Acta Astronautica 62 (2008) 592–604

ACTA ASTRONAUTICA

www.elsevier.com/locate/actaastro

Damage assessment of large space structures through the variational approach

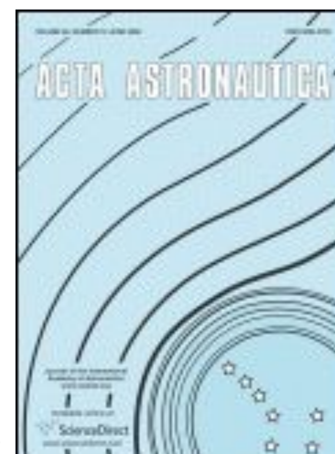
Leonardo D. Chiwiacowsky[a], Paolo Gasbarri[b], Haroldo F. de Campos Velho[c,*]

[a]*Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA), Universidade do Vale do Rio dos Sinos (UNISINOS), Av. Unisinos, 950, 93022-000 São Leopoldo, RS, Brazil*

[b]*Dipartimento di Ingegneria Aerospaziale e Astronautica, Università Degli Studi di Roma - "La Sapienza", Via Eudossiana, 18,00184 Rome, Italy*

[c]*Laboratório Associado de Computação e Matemática Aplicada (LAC), Instituto Nacional de Pesquisas Espaciais (INPE), Av. dos Astronautas, 1758, P.O. Box 515, 12245-970, São José dos Campos, SP, Brazil*

Received 2 February 2006; received in revised form 8 January 2007; accepted 10 January 2008
Available online 11 April 2008



Inverse problem: particle filter

Journal of Computational Biology > Vol. 22, No. 7 > Research Articles

Estimation of Tumor Size Evolution Using Particle Filters

Jose M.J. Costa, Helcio R.B. Orlande, Haroldo F. Campos Velho, Suani T.R. de Pinho, George S. Dulikravich, Renato M. Cotta, and Silvio H. da Cunha Neto

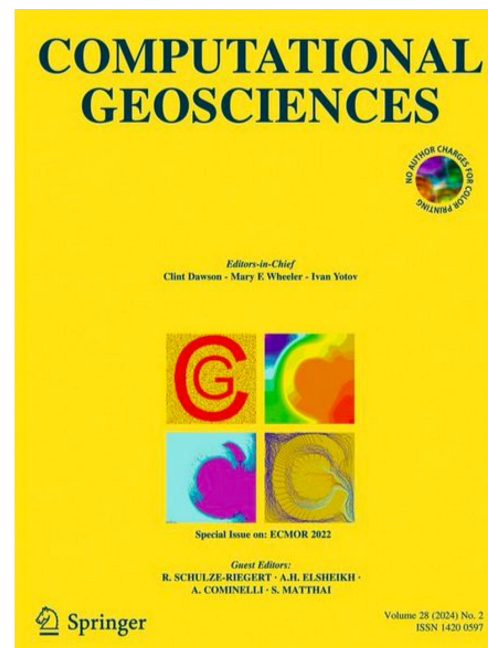
Published Online: 10 Jul 2015 | https://doi.org/10.1089/cmb.2014.0003

Computational Geosciences (2021) 25:931–944
https://doi.org/10.1007/s10596-020-10031-0

ORIGINAL PAPER

Using a machine learning proxy for localization in ensemble data assimilation

Johann M. Lacerda[1] · Alexandre A. Emerick[2] · Adolfo P. Pires[3]



Inverse problem: neural network

Int. J. Information and Communication Technology, Vol. x, No. x, xxxx 1

Atmospheric temperature retrieval using a Radial Basis Function Neural Network

E.H. Shiguemori

Laboratório Associado de Computação e Matemática Aplicada – LAC,
Instituto Nacional de Pesquisas Espaciais – INPE,
São José dos Campos, SP, Brazil

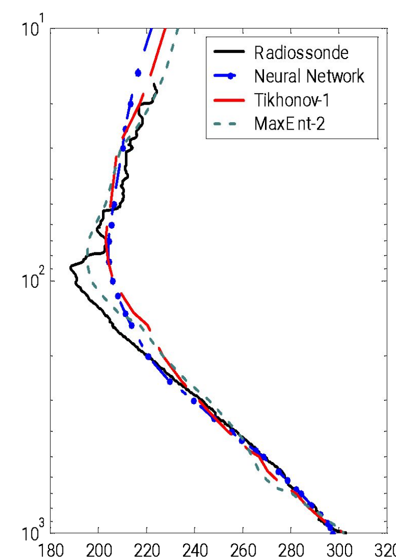
Instituto de Estudos Avançados – IEAv,
Comando-Geral de Tecnologia Aeroespacial – CTA,
São José dos Campos, SP, Brazil
E-mail: elcio@lac.inpe.br

J.D.S. da Silva and H.F. de Campos Velho

Laboratório Associado de Computação e Matemática Aplicada – LAC,
Instituto Nacional de Pesquisas Espaciais – INPE,
São José dos Campos, SP, Brazil
E-mail: demisio@lac.inpe.br E-mail: haroldo@lac.inpe.br

J.C. Carvalho

Superintendência de Administração
da Rede Hidrometeorológica – SAR,
Agência Nacional de Águas – ANA, Brasília, DF, Brazil
E-mail: joao.carvalho@ana.gov.br



Forecasts Scores

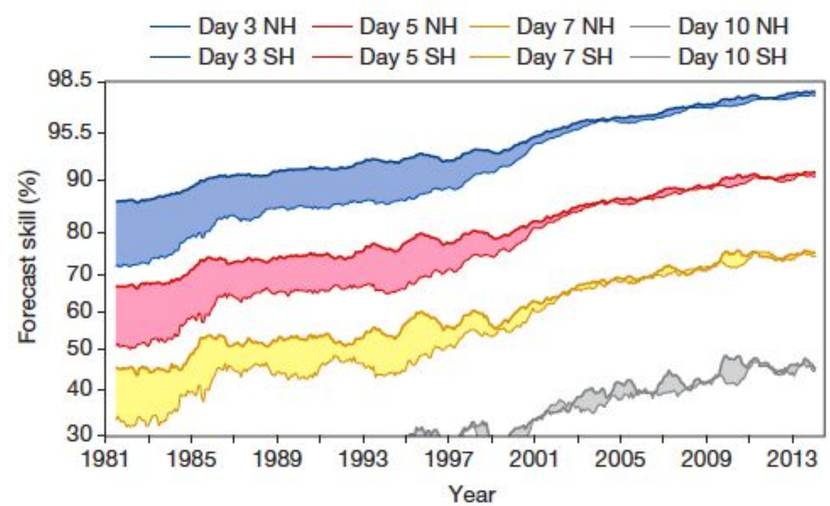
ECMWF

REVIEW

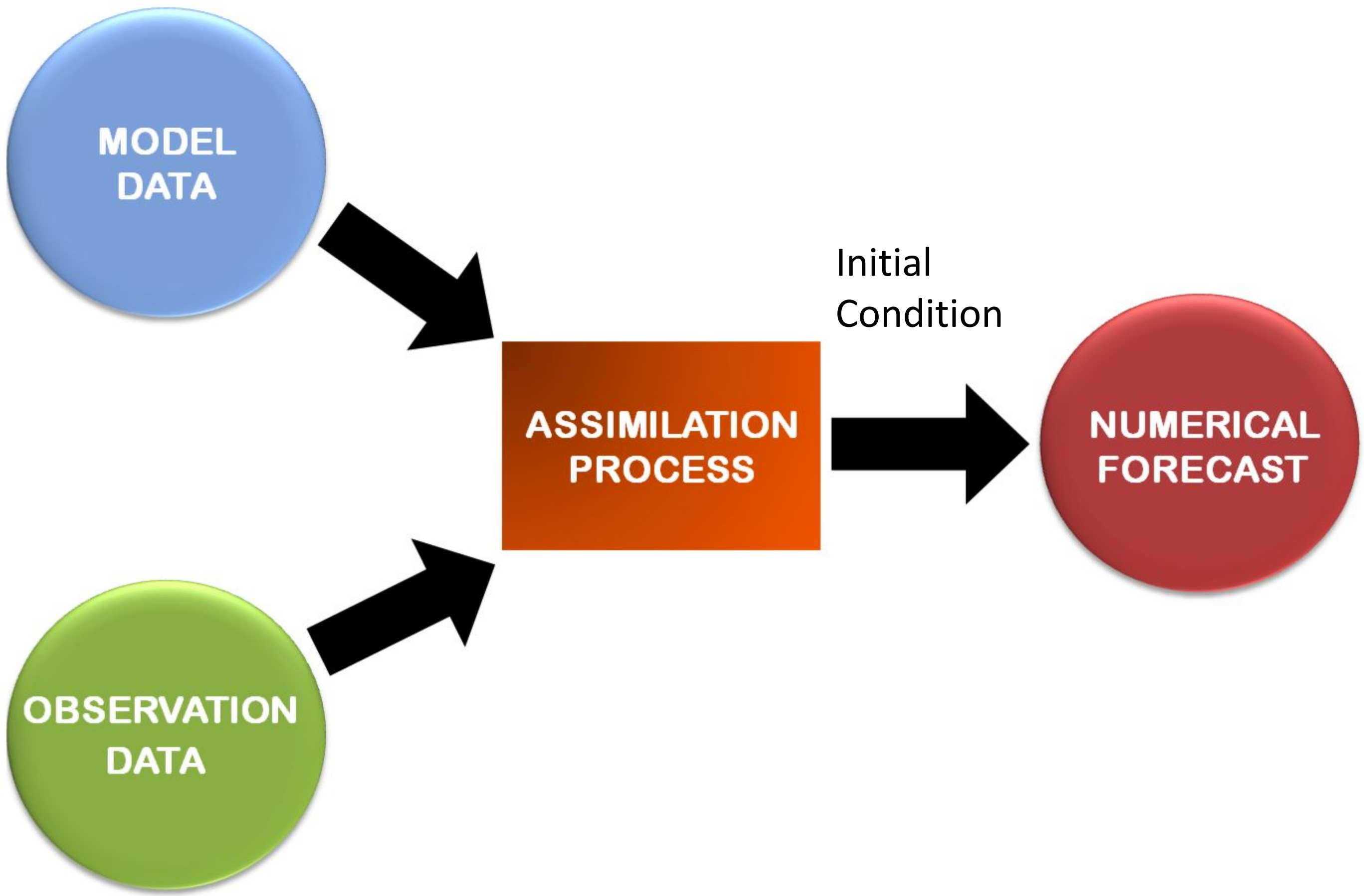
doi:10.1038/nature14956

The quiet revolution of numerical weather prediction

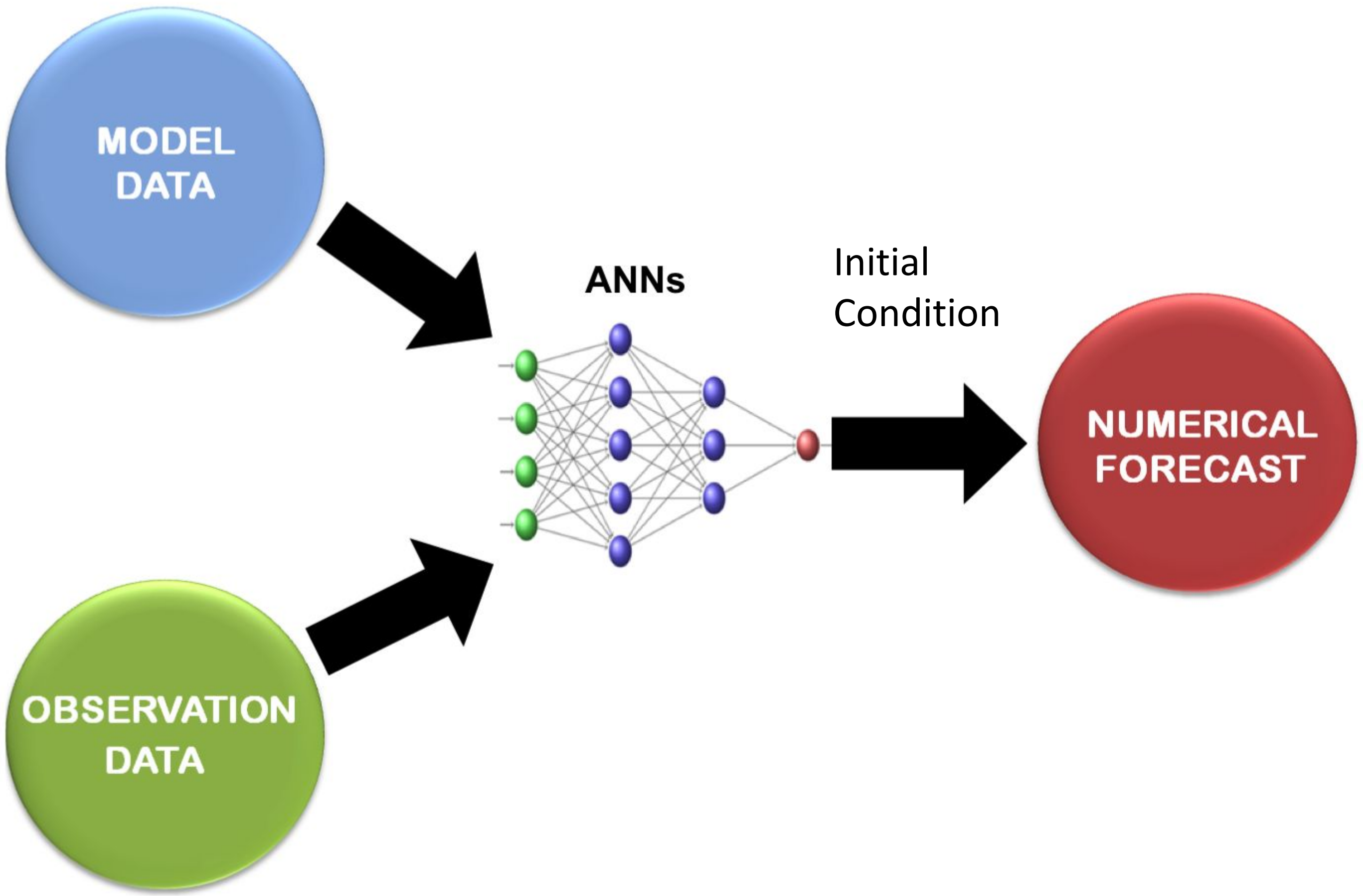
Peter Bauer[1], Alan Thorpe[1] & Gilbert Brunet[2]



Data assimilation – concept



Data assimilation – concept



Finding an OPTIMAL neural network

- Supervised neural network: Multi-Layer Perceptron (MLP)

MPCA solution

# hidden layers	# neurons layer-1	# neurons layer-2	# neurons layer-3	Activation function	Momentum ratio	Learning ratio

Parameters	Vallue
Number of hidden layers	|1| |2| |3|
Number of neurons for each layer	|1| ... |32|
Learning ratio	|0| ... |1|
Momentum	|0| ... |0.9|
Activation function	|Tanh| |Log| |Gauss|

Finding an OPTIMAL neural network

- Design of supervised neural network:

  Optimization problem – cost function:

$$E_{trein} = \frac{1}{N}\sum_{k=1}^{N}(d_k - s_k)^2$$

$$E_{gen} = \frac{1}{(M-N+1)}\sum_{k=N+1}^{M}(d_k - s_k)^2$$

$$F_{obj} = penalty * \frac{\left[\rho_1 * E_{trein} + \rho_2 * E_{gen}\right]}{\rho_1 + \rho_2}$$

$$penalty = \underbrace{\left(c_1 * \left(e^{\#neuron}\right)^2\right)}_{\text{complexity factor-1}} \times \underbrace{\left(c_2 * (\#epoch)\right)}_{\text{complexity factor-2}} + 1$$

MPCA: Multi-Particle Collision Algorithm

Available for download:
www.epacis.net/jcis/PDF_JCIS/JCIS11-art.01.pdf

PAN-AMERICAN ASSOCIATION OF COMPUTATIONAL INTERDISCIPLINARY SCIENCES

Journal of Computational Interdisciplinary Sciences (2008) 1(1): 3-10

ISSN 1983-8409
http://epacis.org

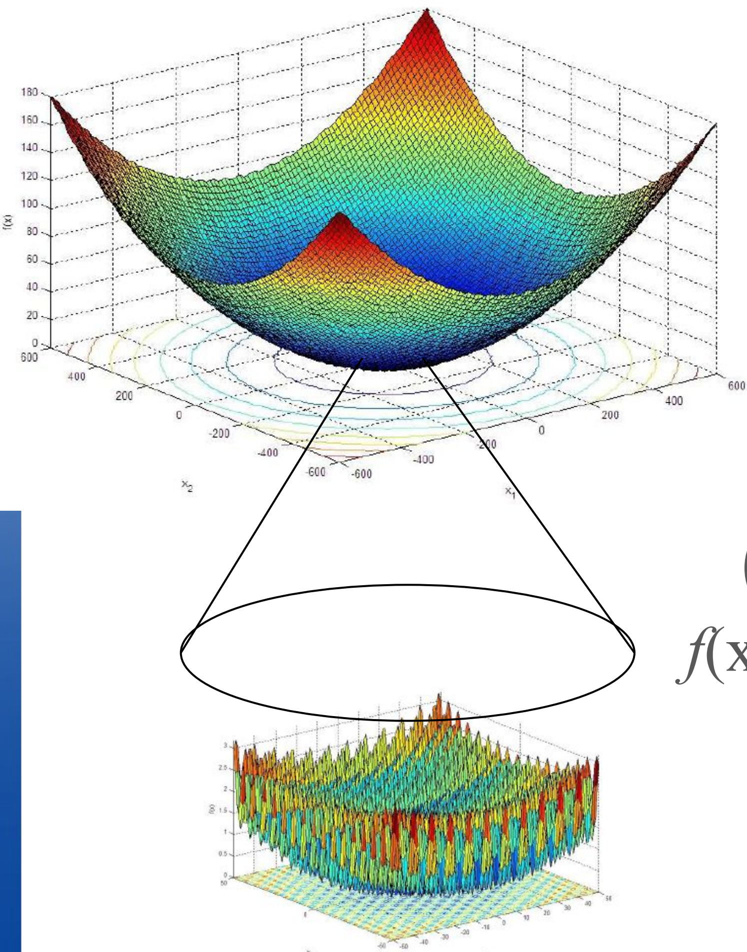
A new multi-particle collision algorithm for optimization in a high performance environment

Eduardo Fávero Pacheco da Luz, José Carlos Becceneri and Haroldo Fraga de Campos Velho

Manuscript received on July 31, 2008 / accepted on October 5, 2008

PCA vs MPCA (2)

Griewank function



$$f(x_1,\ldots,x_n) = 1 + \sum_{j=1}^{n} \frac{x_1^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$$\left\|(x_1,\ldots,x_2)\right\|_2^2 \leq 600$$

$$\min : (0,\cdots,0), \quad f(0,) = 0$$

PCA	MPCA
$(-3.14, 4.43)$	$(-1.8\text{x}10^{-8}, -3.3\text{x}10^{-8})$
$f(x_1,x_2) = 7.4 \times 10^{-3}$	$f(x_1,x_2) = 3.3 \times 10^{-16}$

DA: machine learning approaches

- Machine learning (ML) algorithms for data assimilation.
- ML algorithms:

a) Supervised (self-configuring) NN: MLP by MPCA (black box (??))

b) Supervised (Deep Learning) NN: TensorFlow (black box (??)

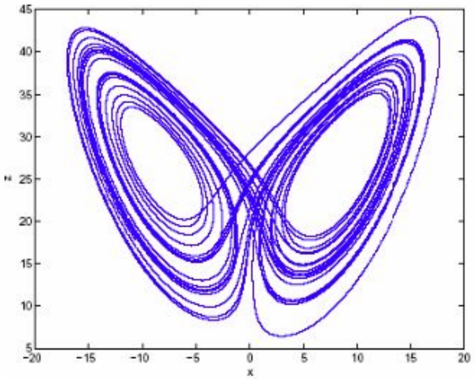
b) Supervised Decision Tree (DT): XGBoost algorithm (white box (??))

c) Unsupervised NN: celular-NN.

DA: cellular neural network (unsupervised NN)

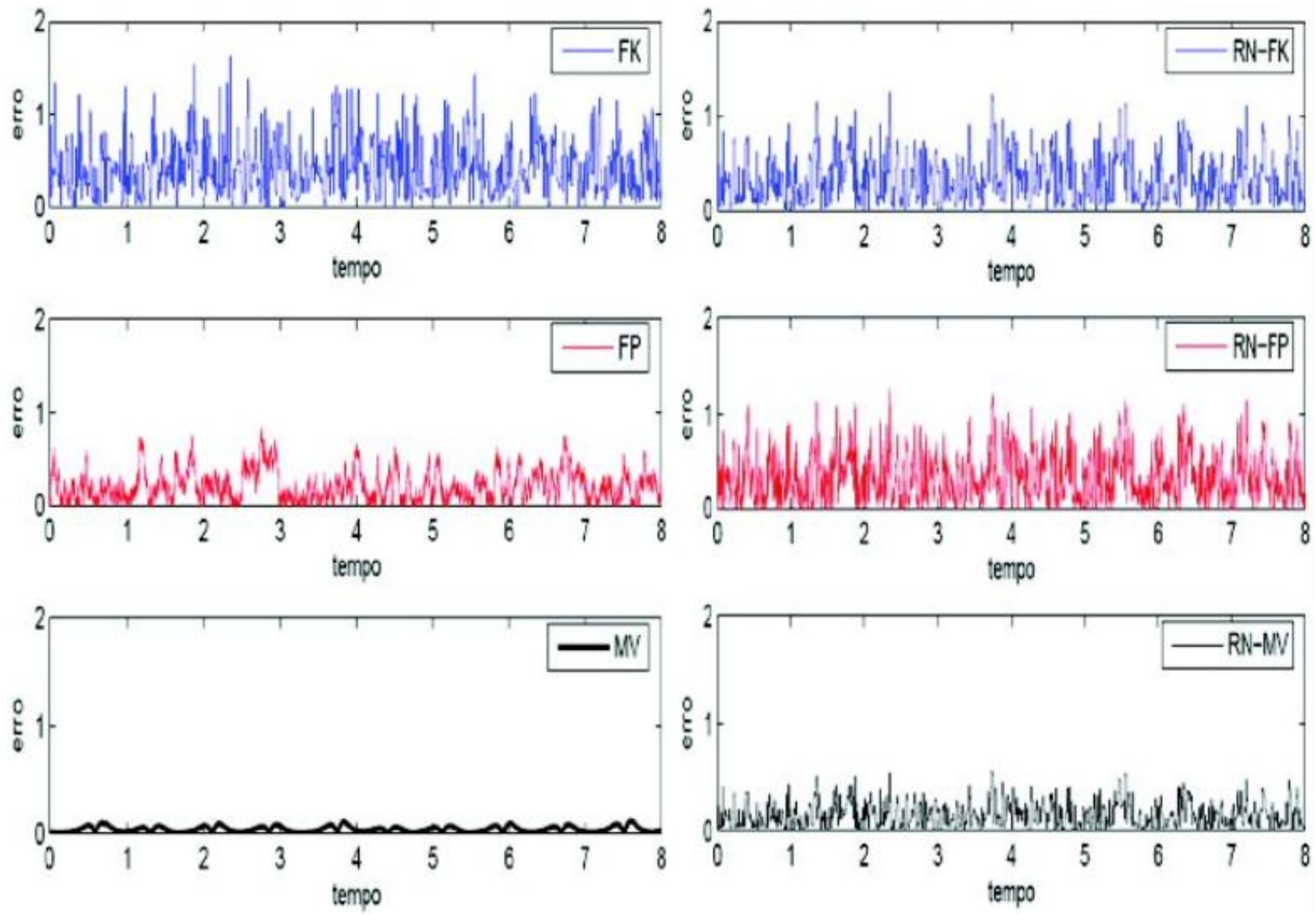
- **Dynamical system: Lorenz chaotic system**

$$
\begin{aligned}
\frac{dX}{dt} &= \sigma(Y - X) \\
\frac{dY}{dt} &= X(\rho - Z) - Y \\
\frac{dZ}{dt} &= XY - \beta Z,
\end{aligned}
$$



Data assimilation – comparison

- NN emulating: Kalman filter, particle filter, variational (error)



Data assimilation – NN emulating variational

- NN emulating variational method: **Lorenz's system**

IOPscience iopscience.iop.org

Dynamic Days South America 2010 IOP Publishing
Journal of Physics: Conference Series **285** (2011) 012036 doi:10.1088/1742-6596/285/1/012036

Neural networks for emulation variational method for data assimilation in nonlinear dynamics

Helaine C. Morais Furtado
Haroldo F. de Campos Velho
Elbert E. Macau

Data assimilation – NN emulating PF

- NN emulating Particle filter: **Lorenz's system**

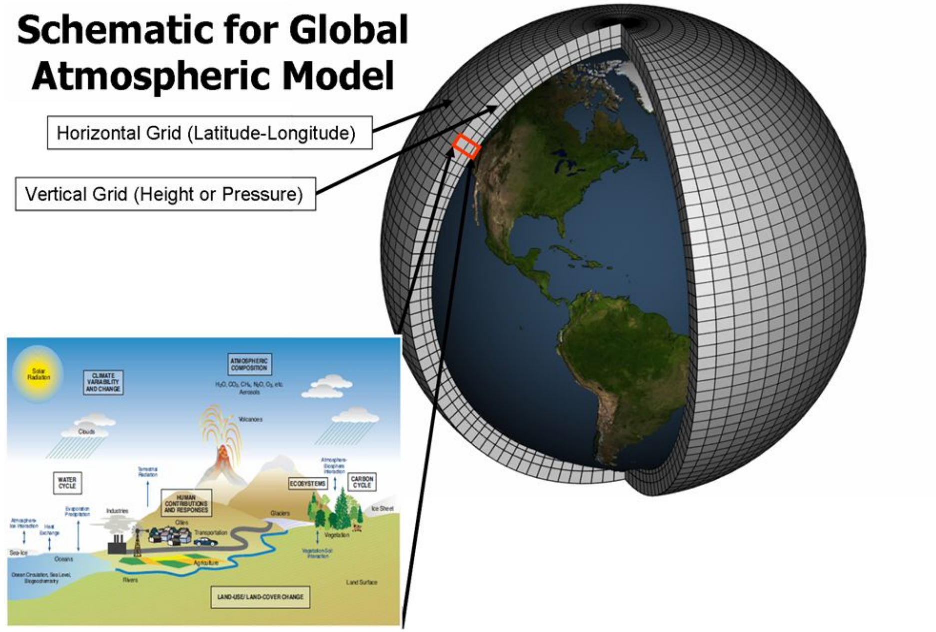
IOPscience iopscience.iop.org

6th International Conference on Inverse Problems in Engineering: Theory and Practice IOP Publishing
Journal of Physics: Conference Series **135** (2008) 012073 doi:10.1088/1742-6596/135/1/012073

Data assimilation: particle filter and artificial neural networks

Helaine Cristina Morais Furtado,
Haroldo Fraga de Campos Velho,
Elbert Einstein Nehrer Macau

SPEED model



Forward model (x^f):

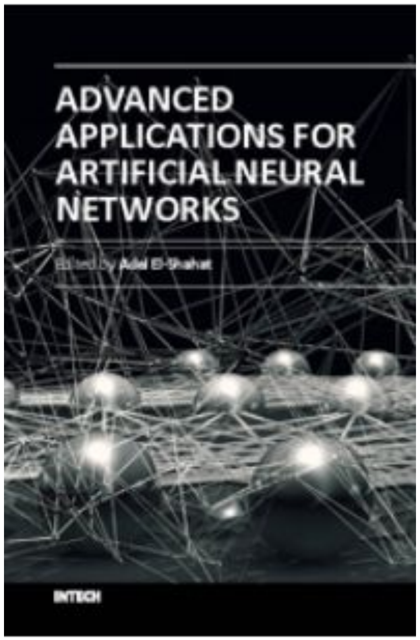
SPEED model

- Atmospheric general circulation model
- 3D spectral model
- simplified parameterization

Vertical coordinates: $\sigma = p_s/p$.
Horizontal coordinates: (lat , long) on a Gaussian grid

The spectral model: T30 horizontal resolution and 7 vertical levels
Observations: 12035 (00 and 12 UTC) = 415 x 4 x 7 + 415
Observations: 2075 (00 and 12 UTC) = 415 x 5 (only surface)

SPEED model



Chapter 14

Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model

Rosangela Saher Cintra and
Haroldo F. de Campos Velho

Additional information is available at the end of the chapter

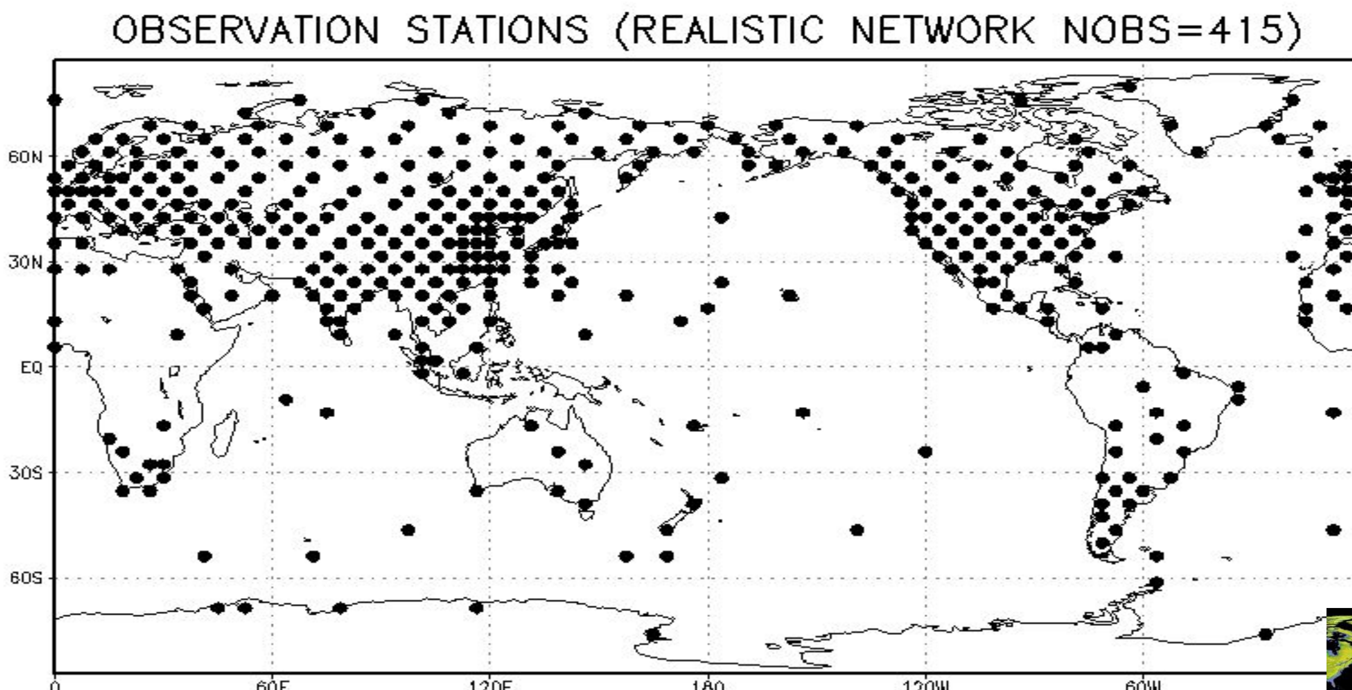
http://dx.doi.org/10.5772/intechopen.70791

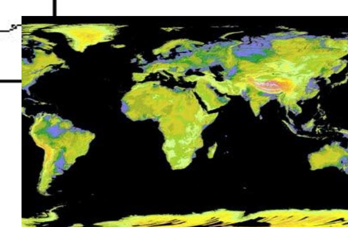
2018

SPEED: atmospheric general circulation model

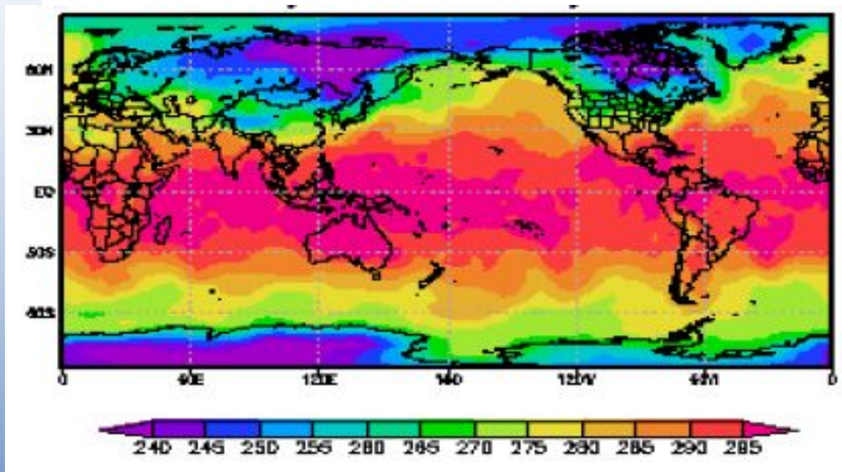
Spectral 3D model, with simplified parameterization

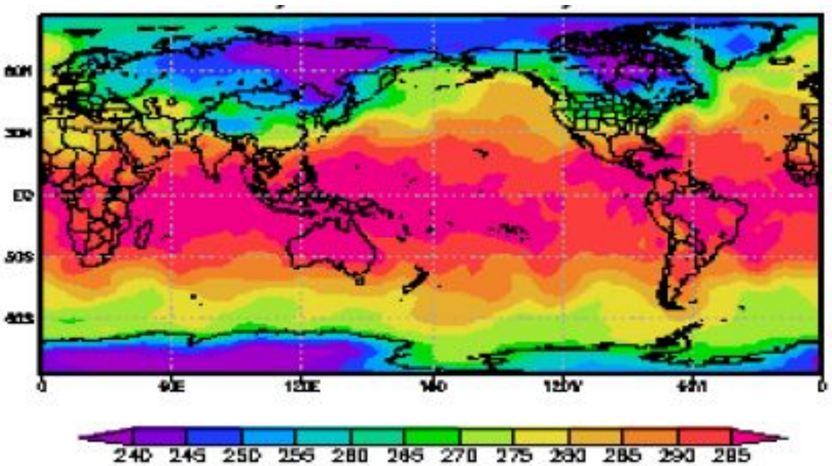
Observation grid: NN emulating LEnTKF

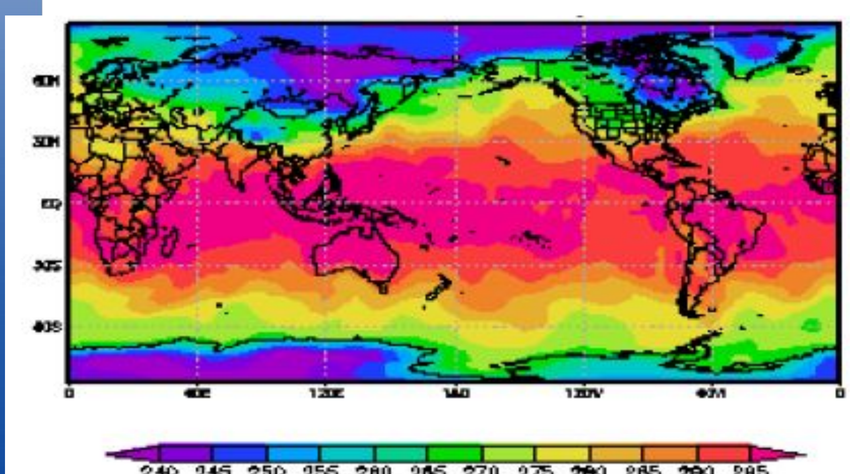




Temperature: assimilation experiment







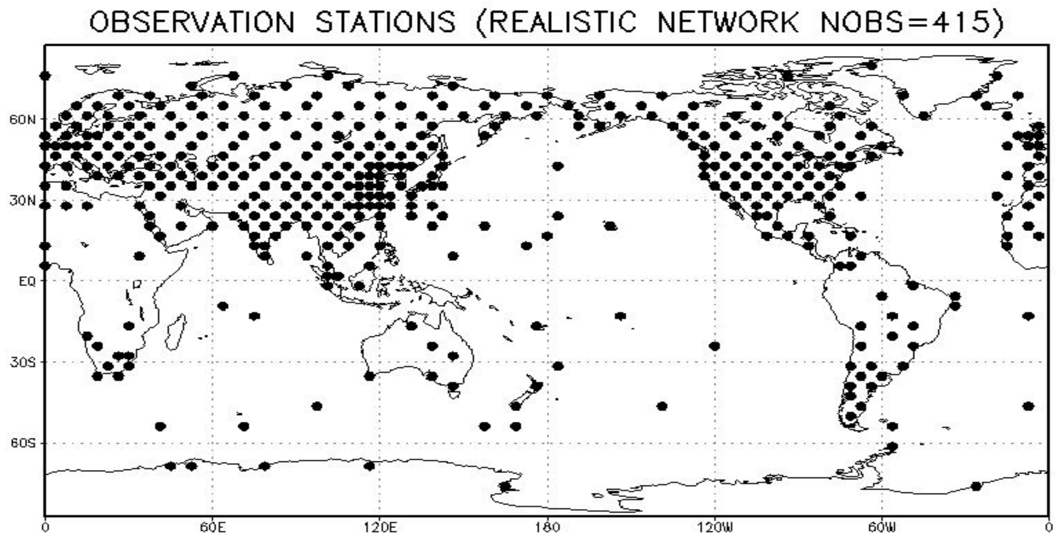
LETKF

neural network

True

Results from Rosangela Cintra PhD thesis (2011)

Experiment: LETKF and neural network



Execution time	
LETKF method	ANN method
04:20:39	00:02:53
hours : minutes : seconds	

General atmospheric Circulation Modelo 3D (spectral model):

SPEEDY (Simplified Parameterizations primitivE Equation DYnamics)

Gaussian grid: 96 x 48 (horizontal) x 7 lvels (vertical) = T30L7

Total grid points: 32.256 Total de variáveis: 133.632

Observations: (00, 06, 12, 18 UTC) – radiosonders "OMM stations"

Observations: 12035 (00 e 12 UTC) = 415 x 4 x 7 + 415

Observations: 2075 (00 e 12 UTC) = 415 x 5 (only surface)

Results from the Rosangela Cintra's PhD thesis (2011)

Global model for NWP

- **FSU-COAPS global model:** equations

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= -\nabla\cdot(\zeta+f)\vec{v}_H - \vec{k}\cdot\nabla\times\left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f}\right)\\
\frac{\partial D}{\partial t} &= \vec{k}\cdot\nabla\times(\zeta+f)\vec{v}_H - \nabla\cdot\left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f}\right) - \nabla^2\left(\phi + \frac{\vec{v}_H\cdot\vec{v}_h}{2}\right)\\
\frac{\partial T}{\partial t} &= -\nabla\cdot(T\vec{v}_H) + TD + \dot{\sigma}\gamma - \frac{RT}{c_p}\left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} + H_T\right)\\
\frac{\partial q}{\partial t} &= -\vec{v}_H\cdot\nabla q - D - \frac{\partial \dot{\sigma}}{\partial \sigma}\\
\sigma\frac{\partial \phi}{\partial \sigma} &= -RT\\
\frac{\partial r}{\partial t} &= -\nabla\cdot(r\vec{v}_H) + rD - \dot{\sigma}\frac{\partial r}{\partial \sigma} + M
\end{aligned}
$$

{with: $q = \log(p_0)$}

{with: $\phi = gh$; and: $\sigma = p/p_0$}

{with: r moisture, and: M source/sink}

Data assimilation: LETKF x ANN (FSU model)

Execution of 124 cycles	MLP-DA (hour:min:sec)	LETKF (hour:min:sec)
Analysis time	00:02:29	11:01:20
Ensemble time	00:00:00	15:50:40
Parallel model time	00:27:20	00:00:00
Total Time	00:29:49	26:52:00

266 times faster

55 times faster

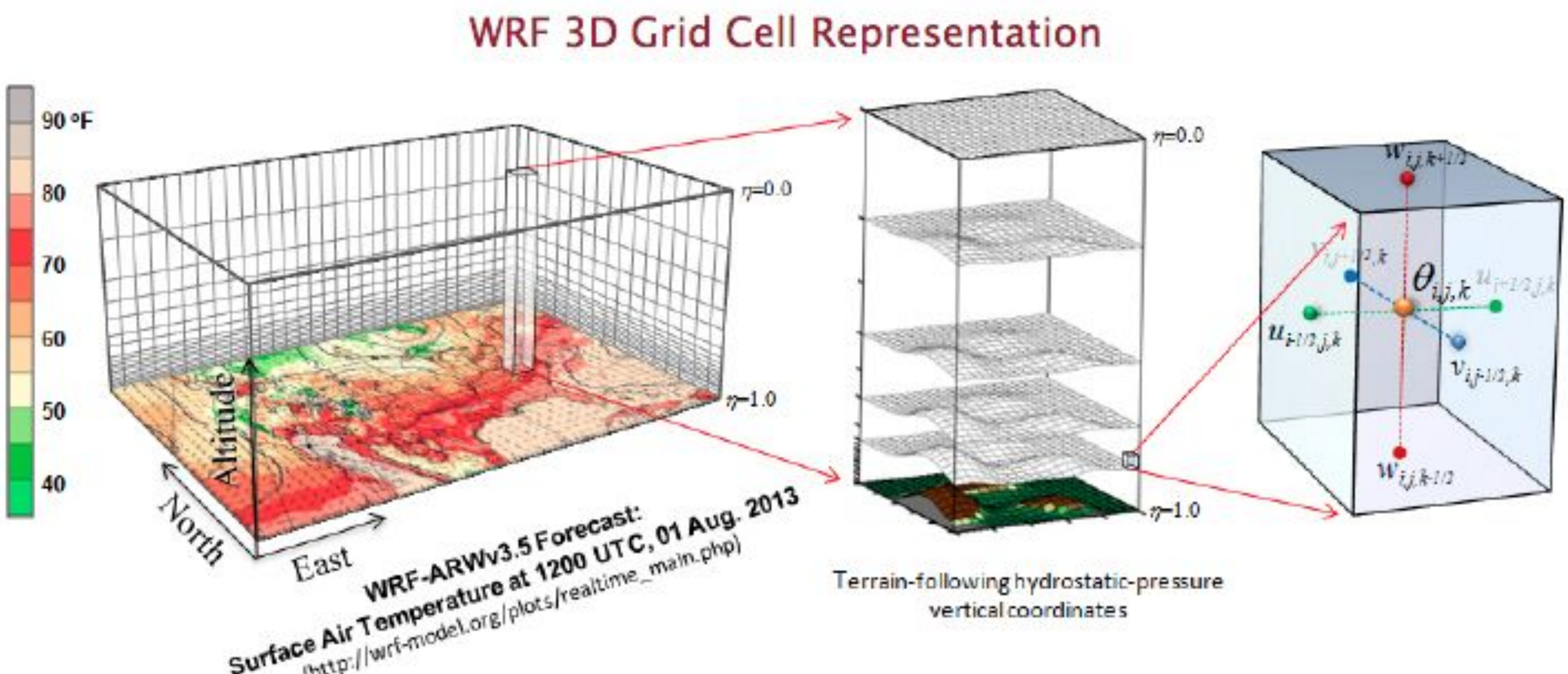
The LETKF analysis runs on 40 nodes at Cray XT/16
(1280 nodes, each node with 2 Opteron 12 cores, total of 30720 cores)
(http://www.cptec.inpe.br/supercomputador)).

MLP-DA computed analyses for the FSUGSM model:

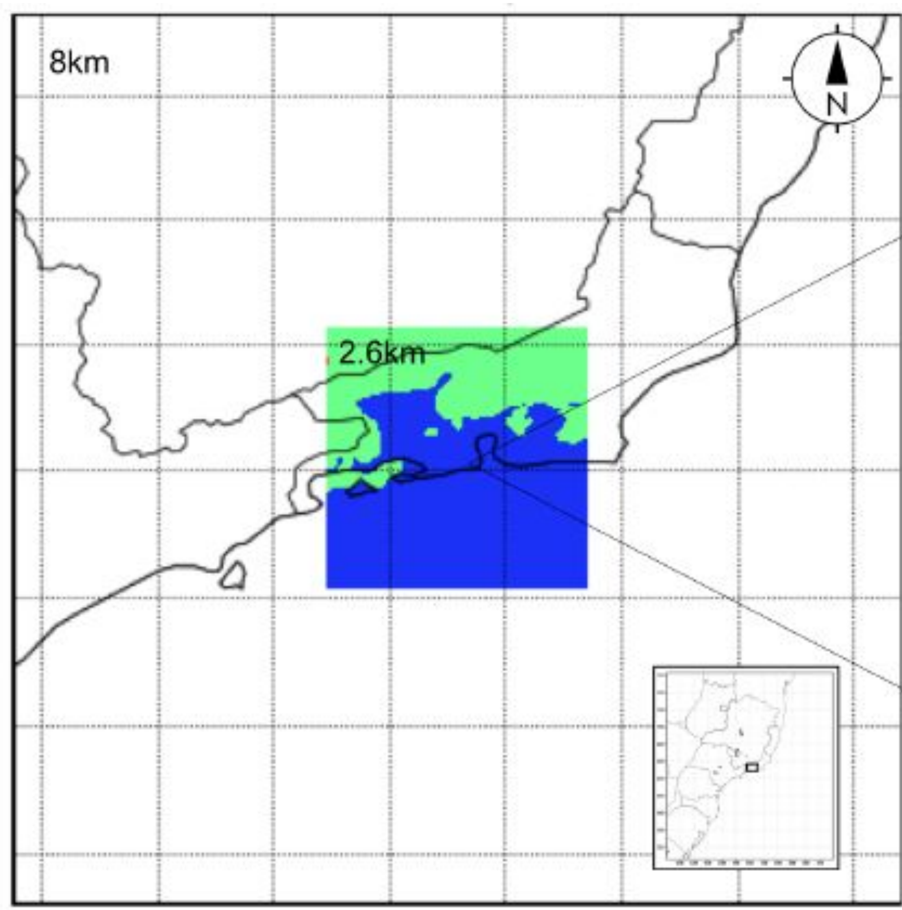
- **Analyses with similar LETKF quality**
- **Analysis with better computer performance.**

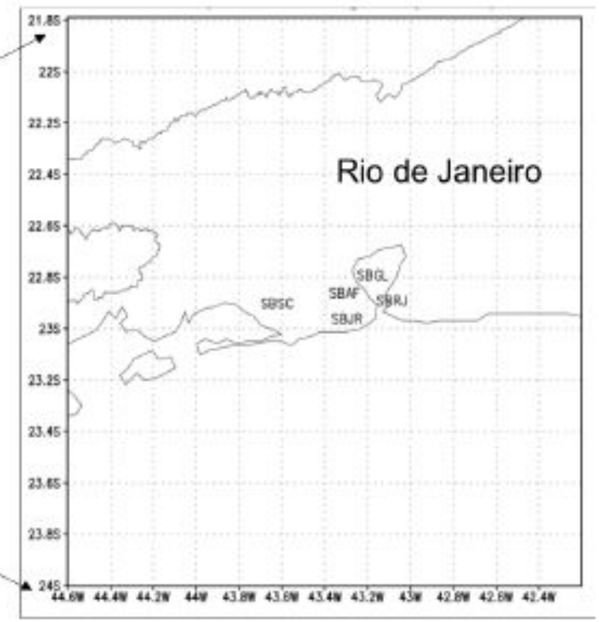
WRF: data assimilation by NN

- Cooperation:
  - CODPT-INPE (BR)
  - Universities (BR): UFPel + IFI-Bagé + UFOPA + UFRJ
  - LNCC (BR)



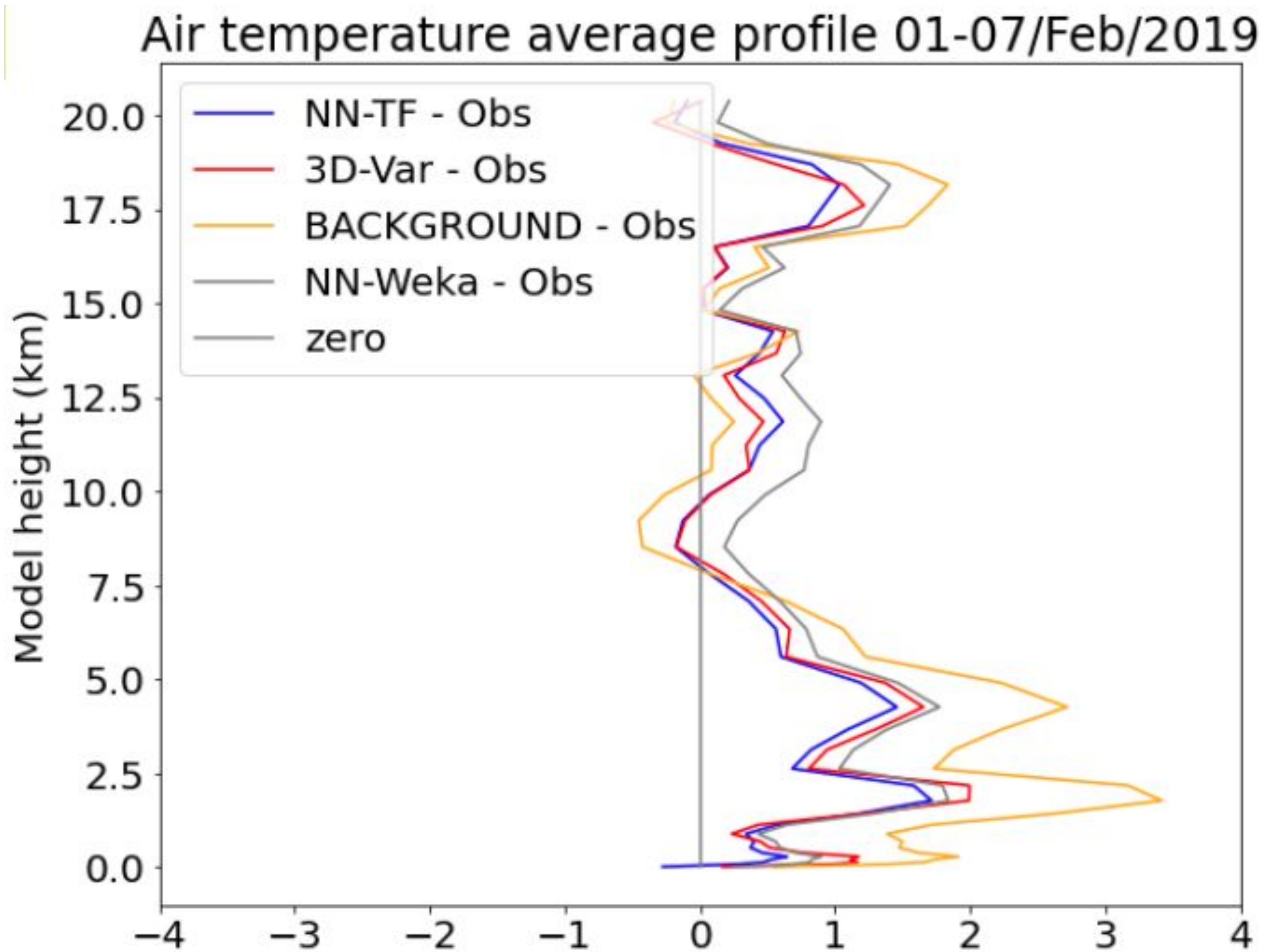
WRF-NCAR model





SBSC, SBAF, SBGL, SBRJ, SBJR are airports within the study area

WRF-NCAR model



WRF-NCAR model

- **CPU-time**

	Time/cycle	Total
3D-Var	00:01:11	00:33:08
NN-MPCA	00:00:01	00:00:28

71 times faster

Data assimilation: NN vs. “standard” methods

- **CPU-time**

MODEL	SPEED (G)	COAPS-FSU (G)	WRF (R)
Method	LEnTKF | NN	LEnTKF | NN	3D-Var | NN
CPU-time	04:20:39 | 00:02:53	26:52:00 | 00:29:49	00:33:08 | 00:00:28

95 times faster

55 times faster

71 times faster

Cooperation: INPE - Imperial College London (UK)

Data Assimilation by Optimized Neural Network for the Fluidity Code

Marcelo Paiva Ramos,[1,2,a)] Luiz Alberto Vieira Dias,[1,b)] Linfeng Li,[3,c)] Fangxin Fang,[3,d)] and Haroldo Fraga de Campos Velho[2,e)]

[1)] *Technological Institute of Aeronautics (ITA)*

[2)] *National Institute for Space Research (INPE)*

[3)] *Imperial College London (ICL)*

(Dated: 17 March 2025)

Fluidity code

- Developed by the Applied Computing and Modelling Group at Imperial College London (UK)
  - Multiscale and Multiphysics Problems
  - Discretization: Finite Elements (FE) / Control Volume (CV)
  - Adaptive Mesh: Gradient-Based Method
  - Reduced order modeling
- Applications:
  - Geophysics: ocean and atmospheric dynamics
  - Environment: urban air pollution / chemistry transport
  - Enigineering: fluid-structure coupling

DA: Optimal supervised neural network by MPCA

Cooperation: INPE–ITA & ICL

INPE
National Institute for Space Resarch - Brazil

ITA
Technological Institute of Aeronautics - Brazil

ICL:
Imperial College London - UK

Data assimilation by NN for Fluidity

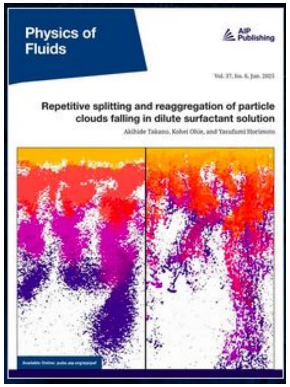
Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Data assimilation by optimized neural network for the fluidity code

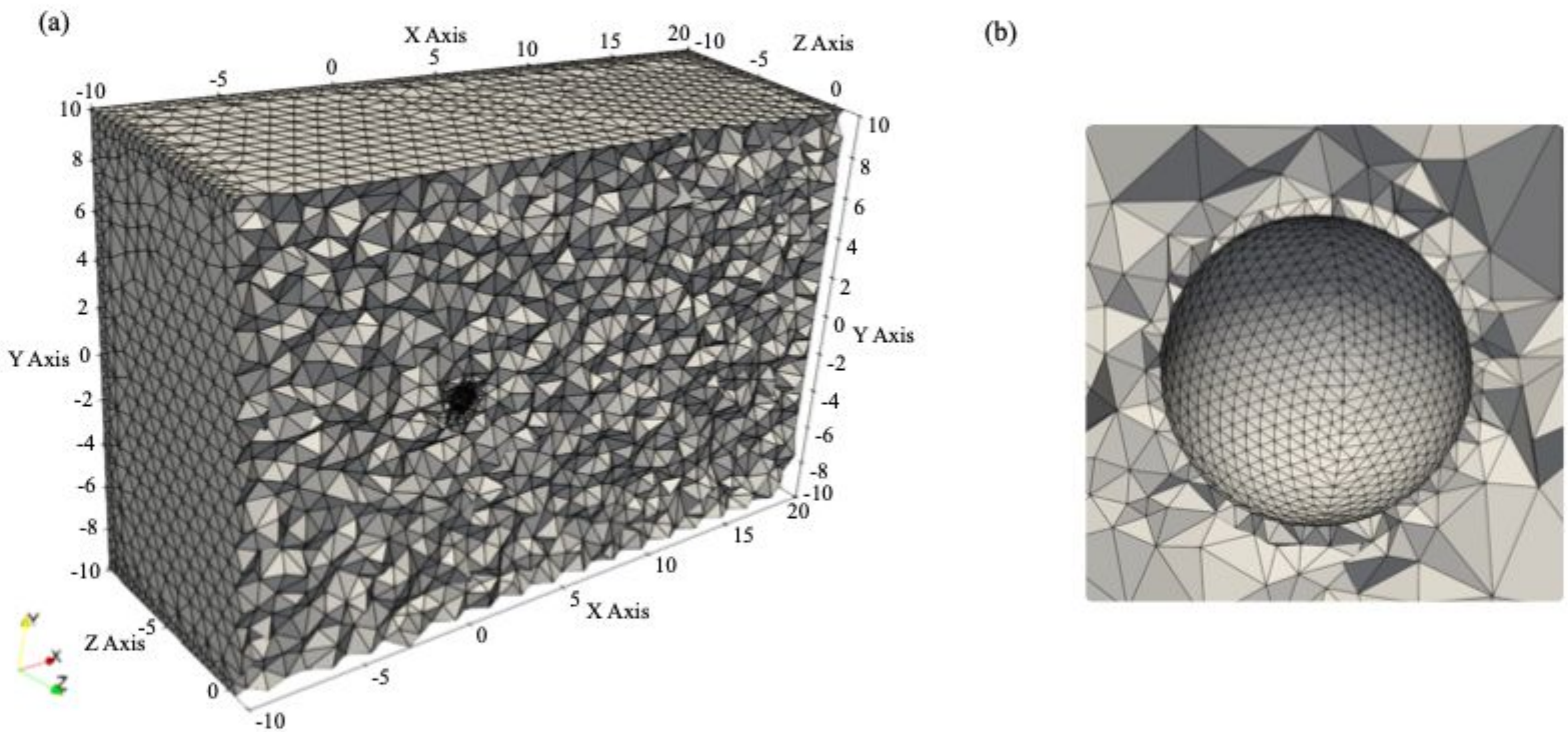
Cite as: Phys. Fluids **37**, 077186 (2025); doi: 10.1063/5.0273157
Submitted: 28 March 2025 · Accepted: 15 June 2025 ·
Published Online: 29 July 2025

View Online Export Citation CrossMark

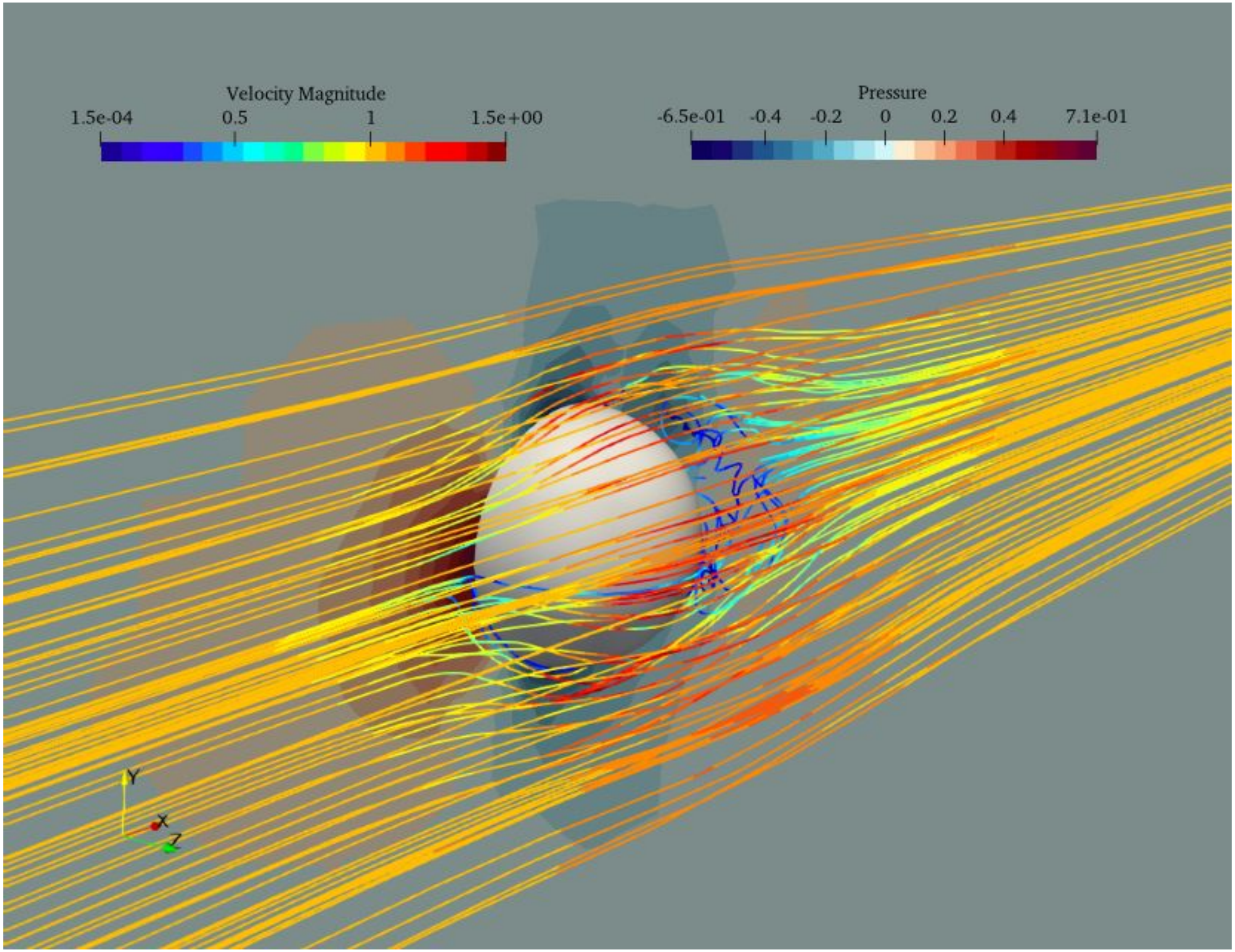
Marcelo Paiva Ramos,[1,2,a)] Luiz Alberto Vieira Dias,[1,b)] Linfeng Li,[3,c)] Fangxin Fang,[3,d)]
and Haroldo Fraga de Campos Velho[4,e)]



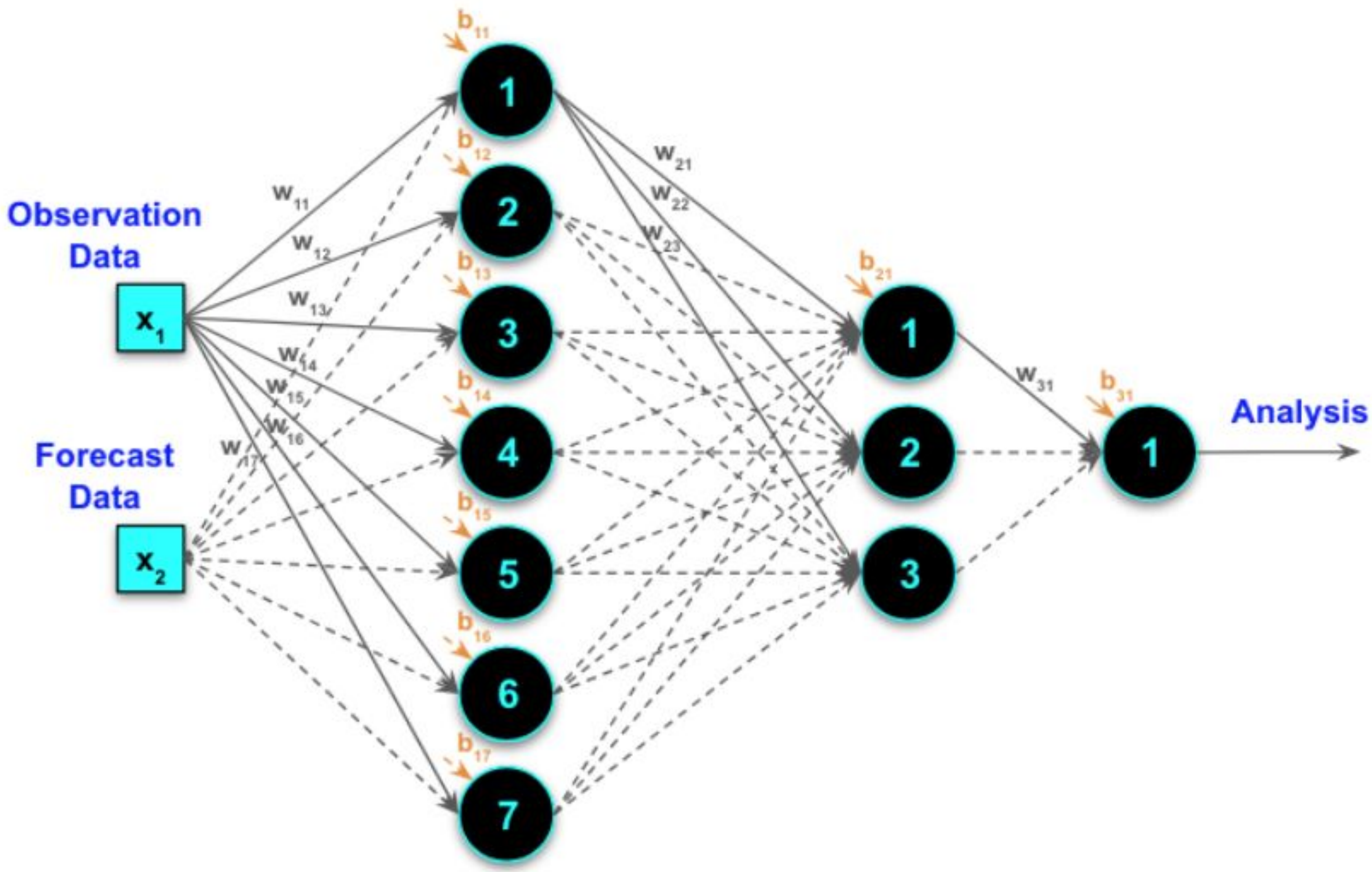
Cooperation: INPE - Imperial College London (UK)



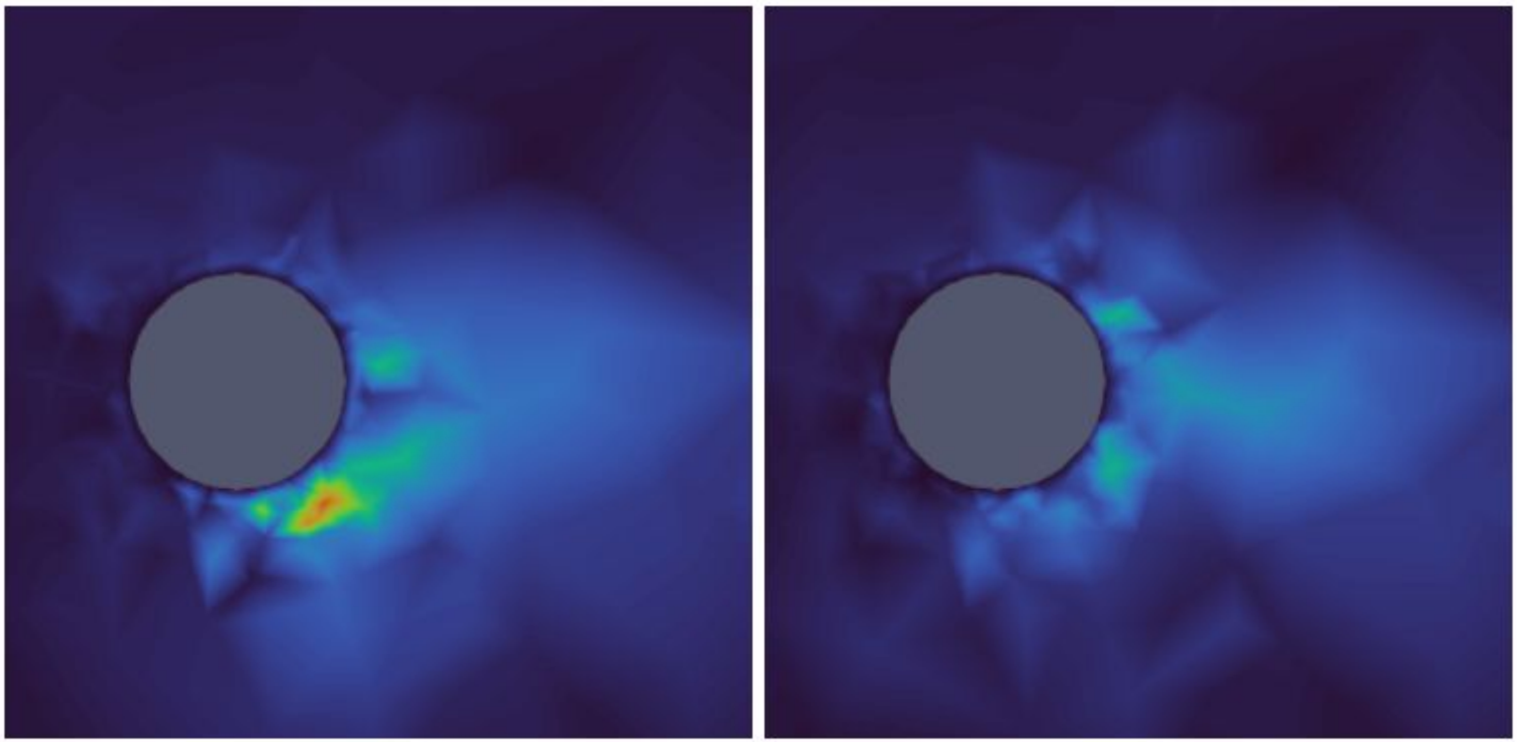
Cooperation: INPE - Imperial College London (UK)



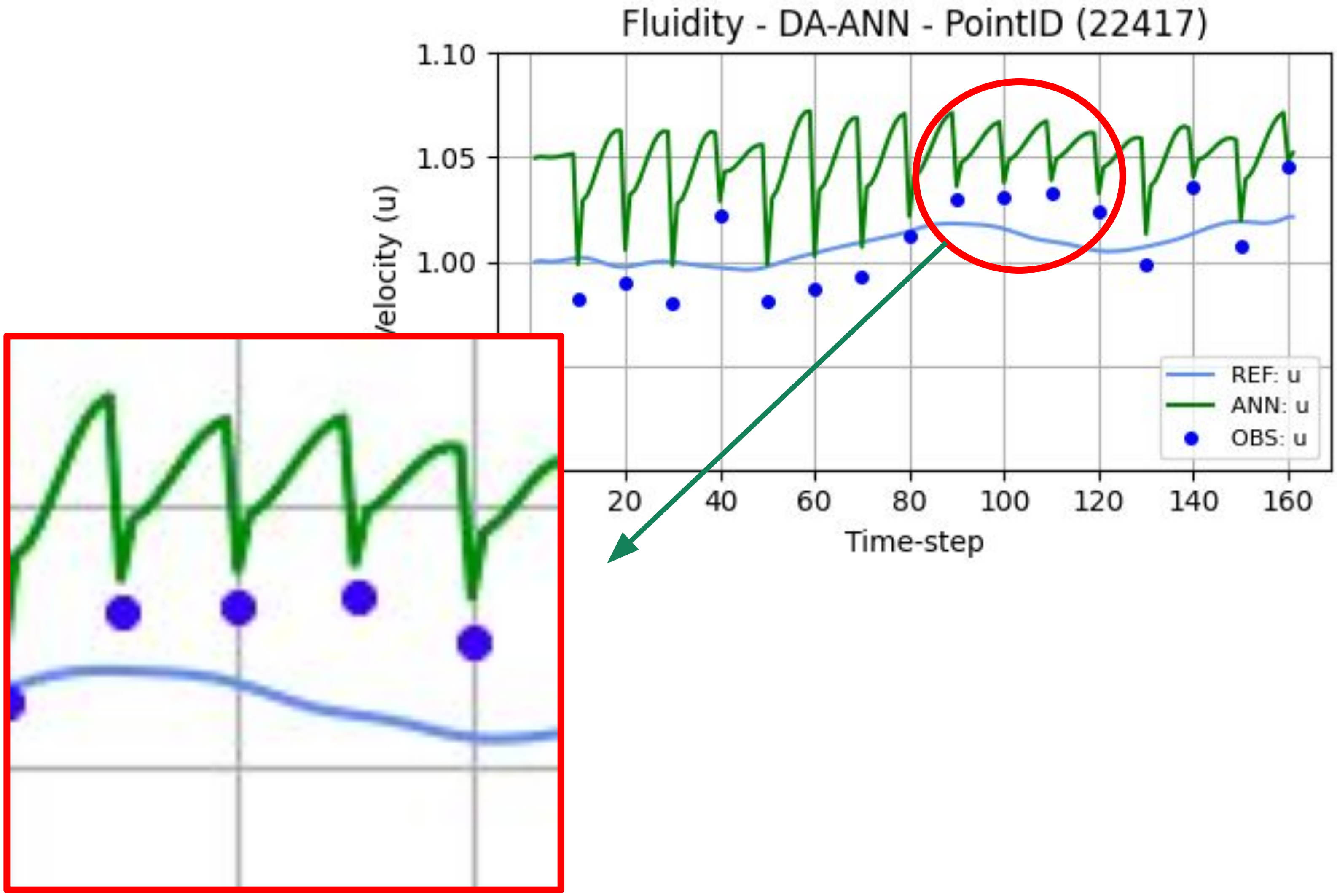
Cooperation: INPE - Imperial College London (UK)



Cooperation: INPE - Imperial College London (UK)



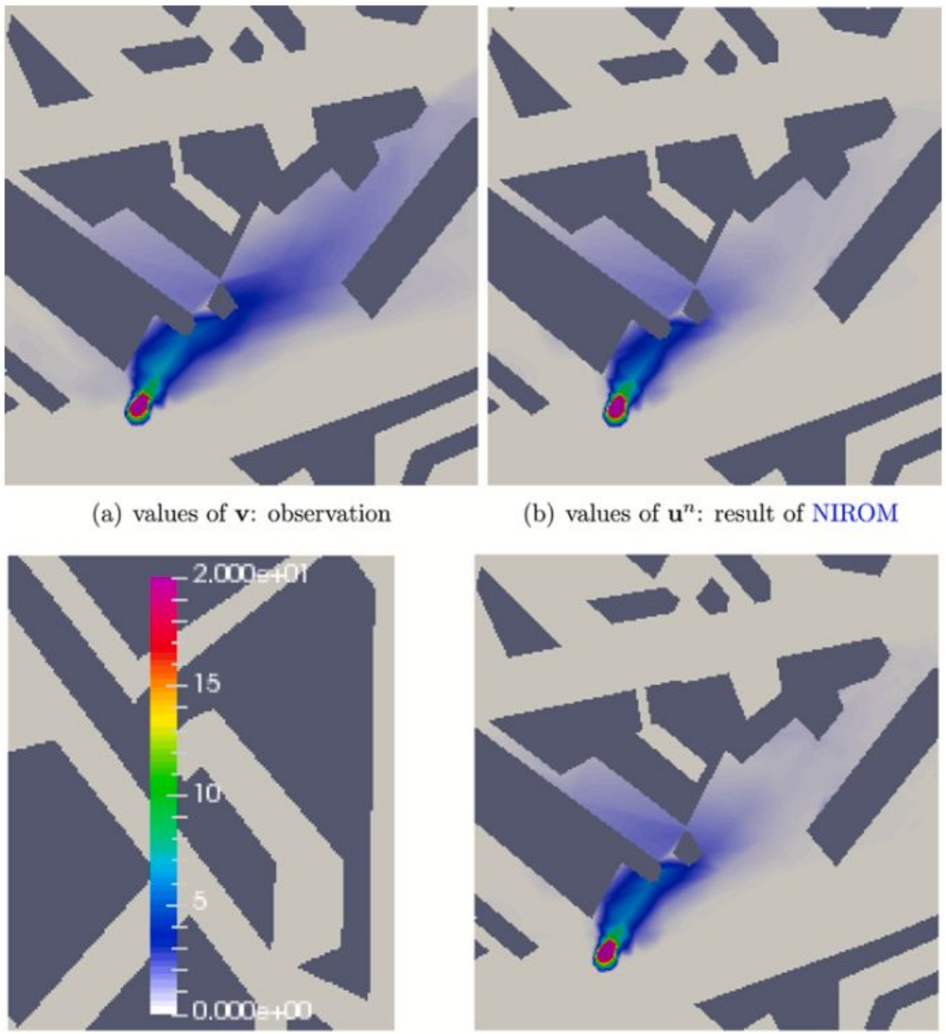
Fluidity - Assimilação de Dados



Fluidity: Data Assimilation

Important application:
Air pollutant on urban scenario

Fluidity - Air pollution on urban scenario



(a) values of $\mathbf{v}$: observation

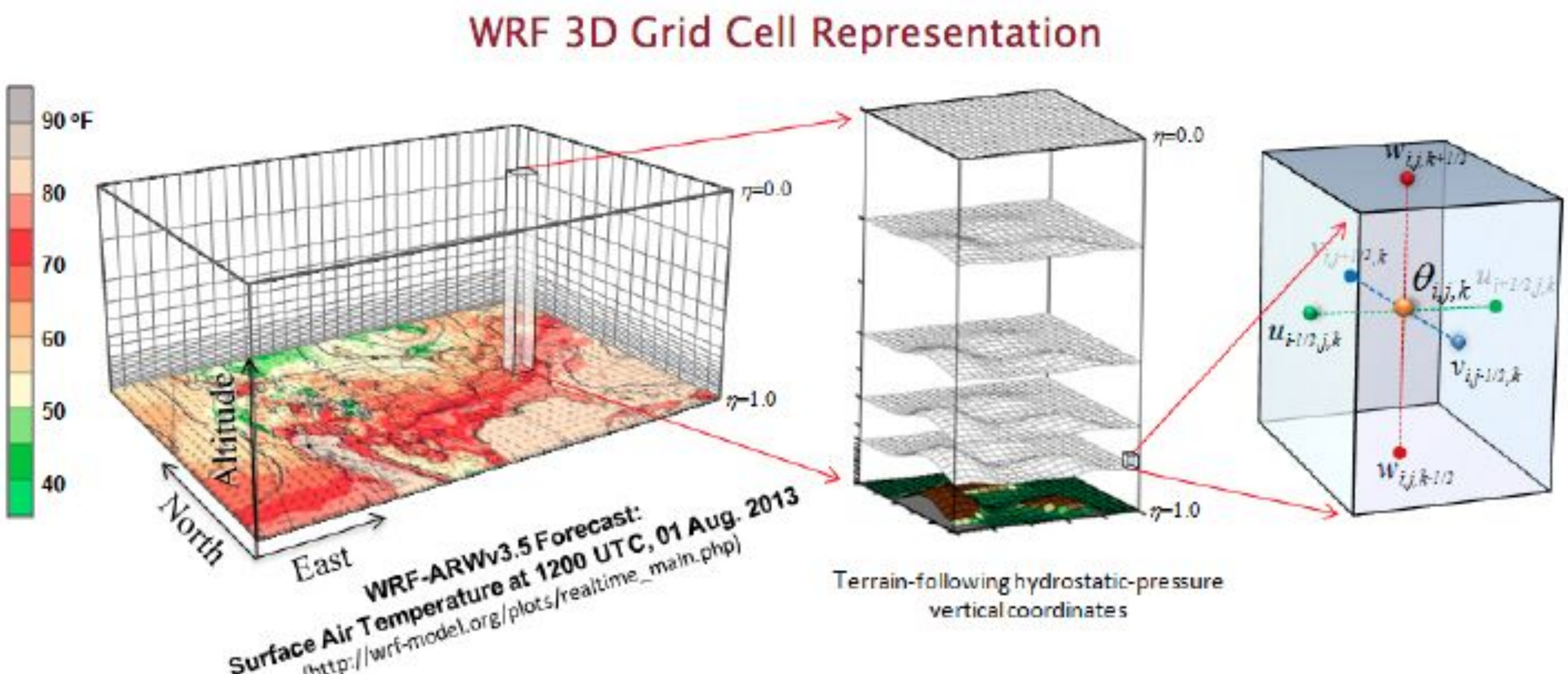
(b) values of $\mathbf{u}^n$: result of NIROM

(c) legend

(d) values of $\mathbf{u}^{RODA}$: result of RODA

WRF: data assimilation by DT

- Cooperation:
  - CODPT-INPE (BR)
  - Universities (BR): UFPel + IFI-Bagé + UFOPA + UFRJ
  - LNCC (BR)



If you do not like NN?

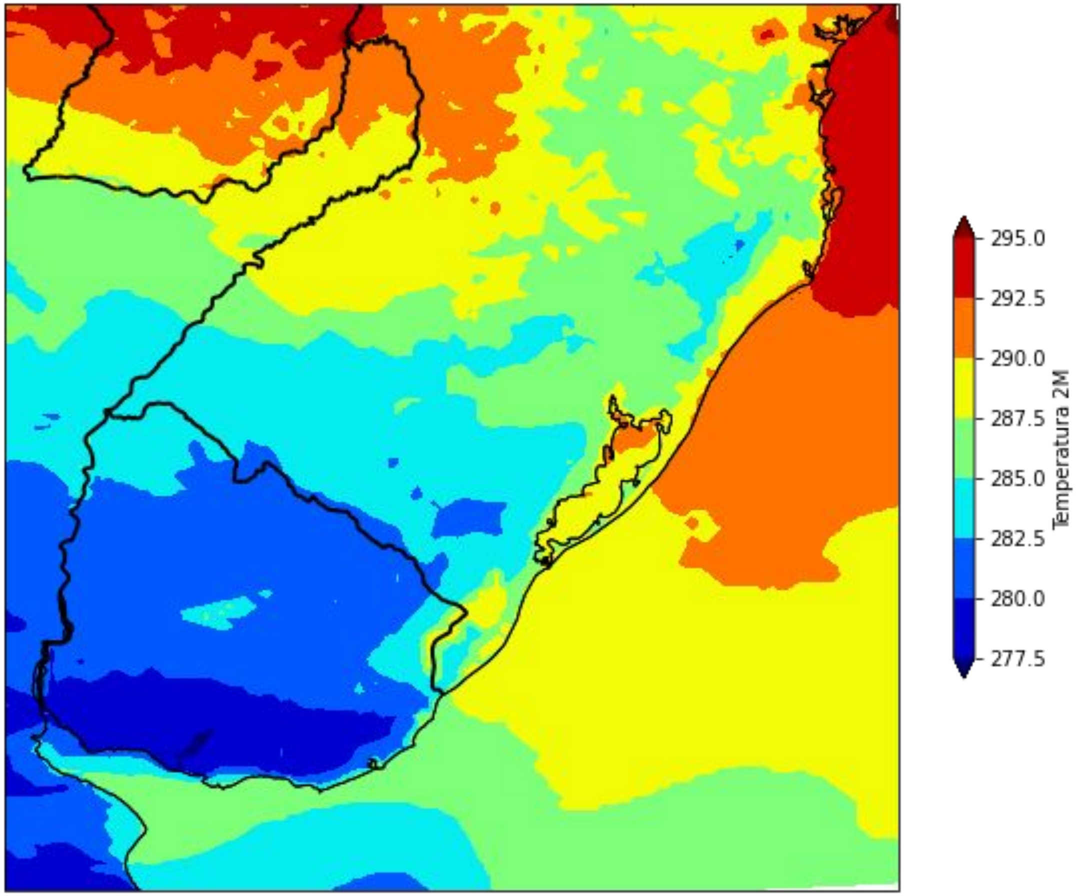
You can use decision tree - for example

2023

If you do not like NN?

You can use decision tree (DT) - for example

Surface Temperature

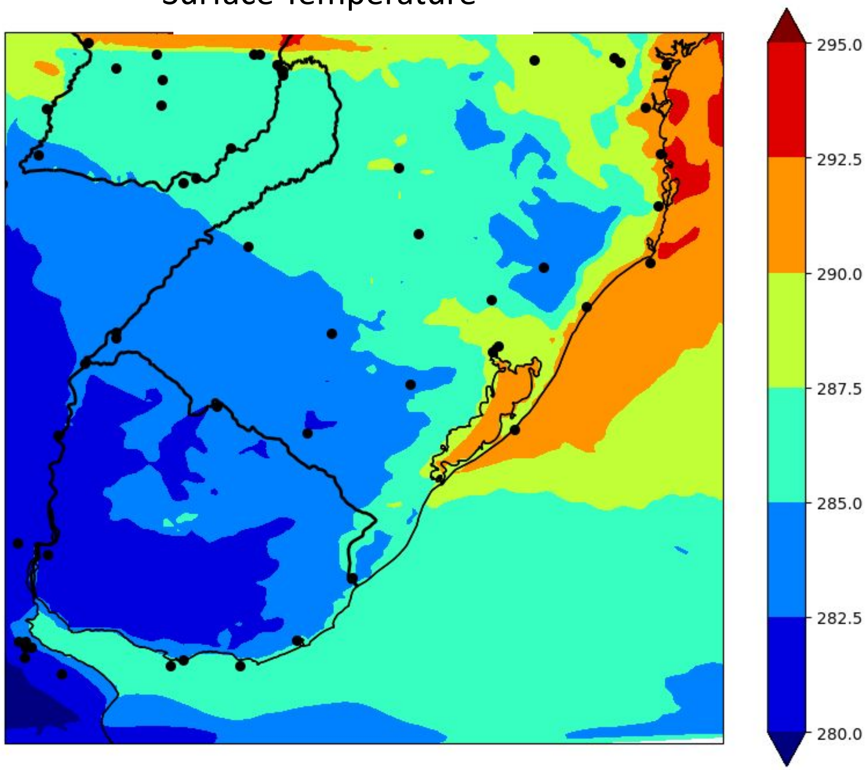


2023

If you do not like NN?

You can use decision tree (DT) - for example

Surface Temperature



Analysis using WRF:

a) Two Methods:

- **3D-Var**
- **DT: XGBoost**

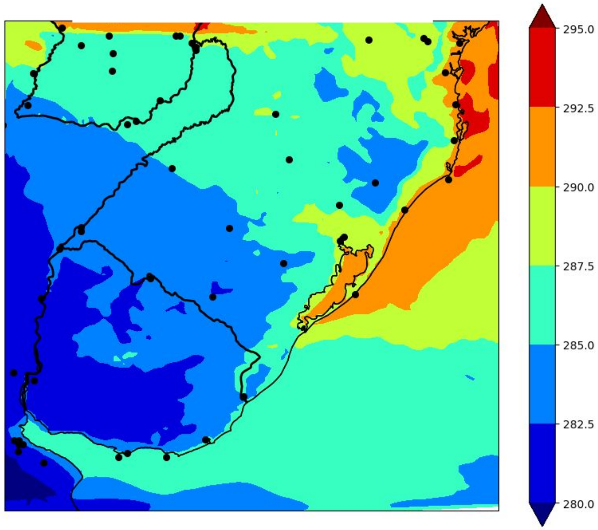
b) **Dot points:** observations

2023

If you do not like NN?

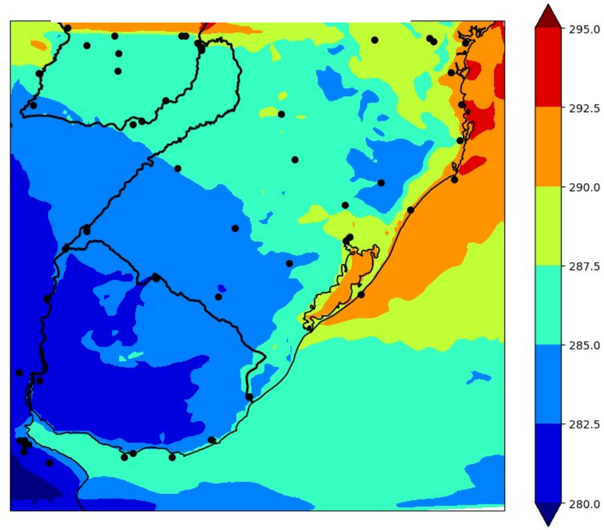
You can use decision tree - for example

Surface Temperature



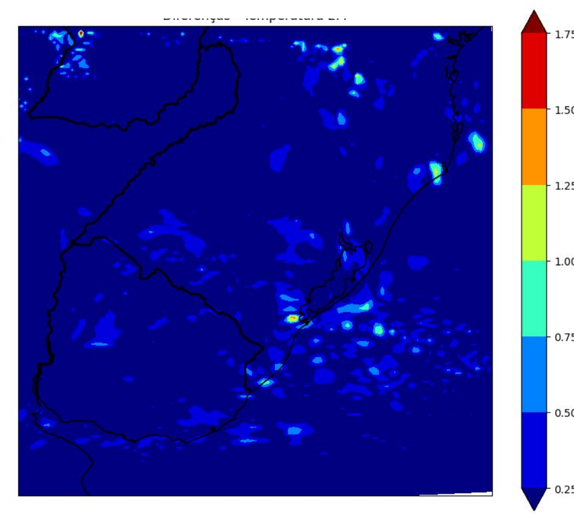
3D-Var

Surface Temperature



XGBoost

Surface Temperature



Diff. = |3D-Var - XGBoost|

2023

DA: cellular neural network (unsupervised NN)

Learning the Daniel Everett's lesson ...

Everett tryied to make contact with Pirahã (Pirahan) people in the Amazonian region.

DA: cellular neural network (unsupervised NN)

Learning the Daniel Everett's lesson ...

The language from the Pirahã people doesn't make sense to him.

Therefore, tried to learn the language like a child. After that, the language started to make sense.

DA: cellular neural network (unsupervised NN)

Learning the Daniel Everett's lesson ...

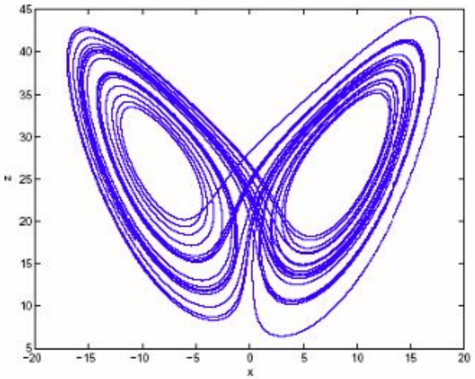
The lesson is:

Instead of trying to match the data with the theory, the theory should be adapted to match the data.

DA: cellular neural network (unsupervised NN)

- **Dynamical system: Lorenz chaotic system**

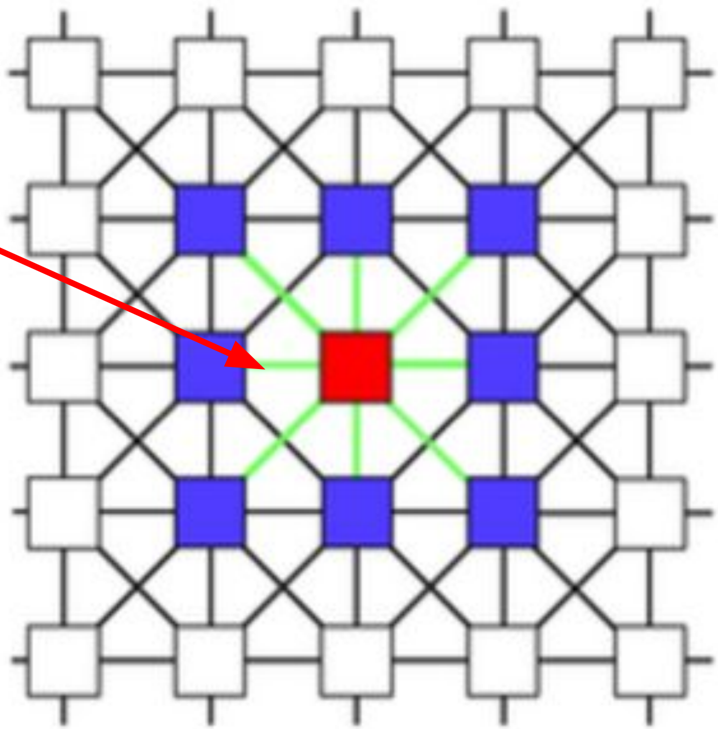
$$\begin{aligned}\frac{dX}{dt} &= \sigma(Y - X)\\ \frac{dY}{dt} &= X(\rho - Z) - Y\\ \frac{dZ}{dt} &= XY - \beta Z,\end{aligned}$$



DA: cellular neural network (unsupervised NN)

- **The Future: unsupervised – Ex.: Cellular network**

$\boldsymbol{\mu}_{ij}$



$$[\,\boldsymbol{\mu}_{ij}\,] = \begin{array}{|c|c|c|} \hline X & X & X \\ \hline Y & Y & Y \\ \hline Z & Z & Z \\ \hline \end{array}$$

DA: cellular neural network (unsupervised NN)

- **DA method: Cellular neural network**

$$\frac{d\mu^{r}_{ij}(\tau)}{d\tau} = -\mu^{r}_{ij}(\tau) + \sum_{(k,l)\in N_r(i,j)} A^{\mathrm{DA}}_{ij,kl}\, v_{kl}(\tau) + \sum_{(k,l)\in N_r(i,j)} B^{\mathrm{DA}}(\tau)_{ij,kl}\, u_{kl}$$

$$A^{\mathrm{DA}}_{ij,kl} = \begin{bmatrix} \alpha_A & 0 & 0 \\ 0 & \alpha_A & 0 \\ 0 & 0 & \alpha_A \end{bmatrix}$$

$$B^{\mathrm{DA}}_{ij,kl}(\tau) = \begin{bmatrix} B_{11} & B_{12} & B_{31} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}, \text{ with: } \{B_{mn}\}^{3}_{m,n=1} = \left(\frac{\mu^{o}_{ii}}{\mu^{r}_{ij} w_b}\right)$$

DA: cellular neural network (unsupervised NN)

- **DA method: Cellular neural network**

$$\frac{d\mu^r_{ij}(\tau)}{d\tau} = -\mu^r_{ij}(\tau) + \sum_{(k,l)\in N_r(i,j)} A^{\mathrm{DA}}_{ij,kl}\, v_{kl}(\tau) + \sum_{(k,l)\in N_r(i,j)} B^{\mathrm{DA}}(\tau)_{ij,kl}\, u_{kl}$$

$$v_{kl}(\tau) = \begin{cases} -1\ , & \text{if:} \quad (\mu^o_{ij} - \mathcal{H}\{\mu^r_{ij}\}) < -1\ ; \\ (\mu^o_{ij} - \mathcal{H}\{\mu^r_{ij}\})\ , & \text{if:} \quad -1 \le (\mu^o_{ij} - \mathcal{H}\{\mu^r_{ij}\}) \le 1\ ; \\ 1\ , & \text{if:} \quad (\mu^o_{ij} - \mathcal{H}\{\mu^r_{ij}\}) > 1\ ; \end{cases}$$

$$u_{kl}(\tau) \equiv \mu^o_{ij}/\mu^r_{ij}(\tau) - \text{with: } \mu^r_{ij}(0) = \mu^{\mathrm{b}}_{ij}(t)\ .$$

DA: cellular neural network (unsupervised NN)

- **DA method: Cellular neural network**

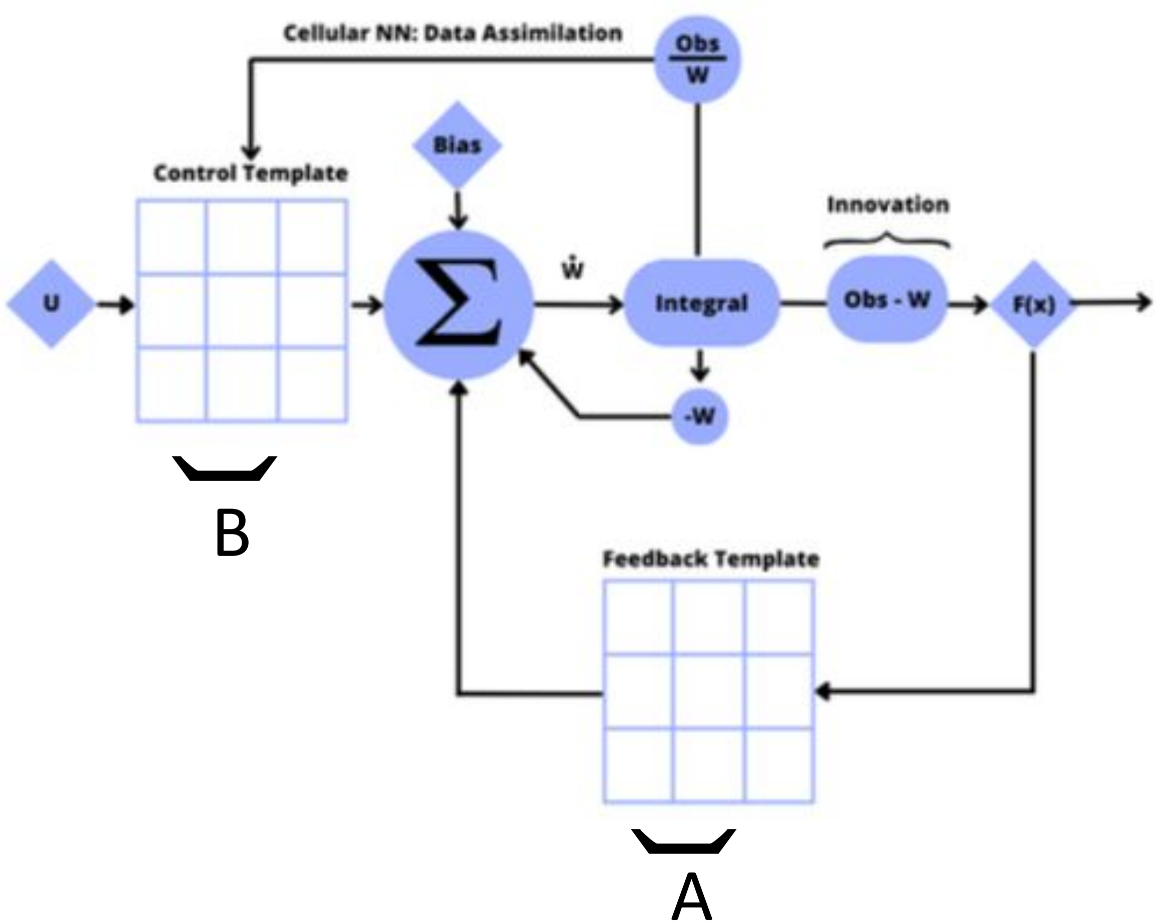
$$\frac{d\mu_{ij}^{r}(\tau)}{d\tau} = -\mu_{ij}^{r}(\tau) + \sum_{(k,l)\in N_r(i,j)} A_{ij,kl}^{\mathrm{DA}}\, v_{kl}(\tau) + \sum_{(k,l)\in N_r(i,j)} B^{\mathrm{DA}}(\tau)_{ij,kl}\, u_{kl}$$

$$\text{For: } |\mu_{ij}^{r}| \;\; \leq 10^{-6} \;\; \begin{cases} \mu_{ij}^{r} > 0 \implies \mu_{ij}^{r} = \mu_{ij}^{r} + \epsilon_r \\ \mu_{ij}^{r} < 0 \implies \mu_{ij}^{r} = \mu_{ij}^{r} - \epsilon_r \end{cases}$$

a) $r = r_{\max}$.
b) $\|\mathbf{w}^{r+1} - \mathbf{w}^{r}\|_2 / \|\mathbf{w}^{r+1}\|_2 \leq \epsilon$.

DA: cellular neural network (unsupervised NN)

- **DA method: Cellular neural network**



DA: cellular neural network (unsupervised NN)

- **DA method: 3D-Var**

$$J(\mathbf{w}) = \frac{1}{2}\left[\mathbf{w}-\mathbf{w}^b\right]^T (\mathbf{P}^b)^{-1}\left[\mathbf{w}-\mathbf{w}^b\right]$$

$$+\frac{1}{2}\left[\mathbf{w}^o-\mathcal{H}\{\mathbf{w}\}\right]^T (\mathbf{R})^{-1}\left[\mathbf{w}^o-\mathcal{H}\{\mathbf{w}\}\right]$$

$$\begin{array}{rclcrcl} \mathbf{P}^b & = & E\{\zeta_n\,\zeta_n^T\} & \quad & \mathbf{w}^b(t_n) & \equiv & \mathbf{w}(t_n)+\zeta_n \\ \mathbf{R} & = & E\{\epsilon_n\,\epsilon_n^T\} & \quad & \mathbf{w}^o(t_n) & \equiv & \mathcal{H}\{\mathbf{w}(t_n)\}+\epsilon_n \end{array}$$

DA: cellular neural network (unsupervised NN)

- **DA method: 3D-Var**

$$J(\mathbf{w}) = \frac{1}{2}\left[\mathbf{w} - \mathbf{w}^b\right]^T (\mathbf{P}^b)^{-1}\left[\mathbf{w} - \mathbf{w}^b\right]$$

$$+\frac{1}{2}\left[\mathbf{w}^{\mathrm{o}} - \mathcal{H}\{\mathbf{w}\}\right]^T (\mathbf{R})^{-1}\left[\mathbf{w}^{\mathrm{o}} - \mathcal{H}\{\mathbf{w}\}\right]$$

$$\boxed{\mathbf{w}^a = \mathbf{w}^b + [(\mathbf{P}^{\mathrm{b}})^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}]^{-1}[\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}]\{\mathbf{w}^o - \mathbf{H}\,\mathbf{w}^b\}}$$

DA: cellular neural network (unsupervised NN)

- **Numerical results:** synthetic observations

$$\mathbf{w}^{o}(t) \equiv \mathbf{w}(t)\left[1 + \sigma_n^2 \times \nu(t)\right]$$

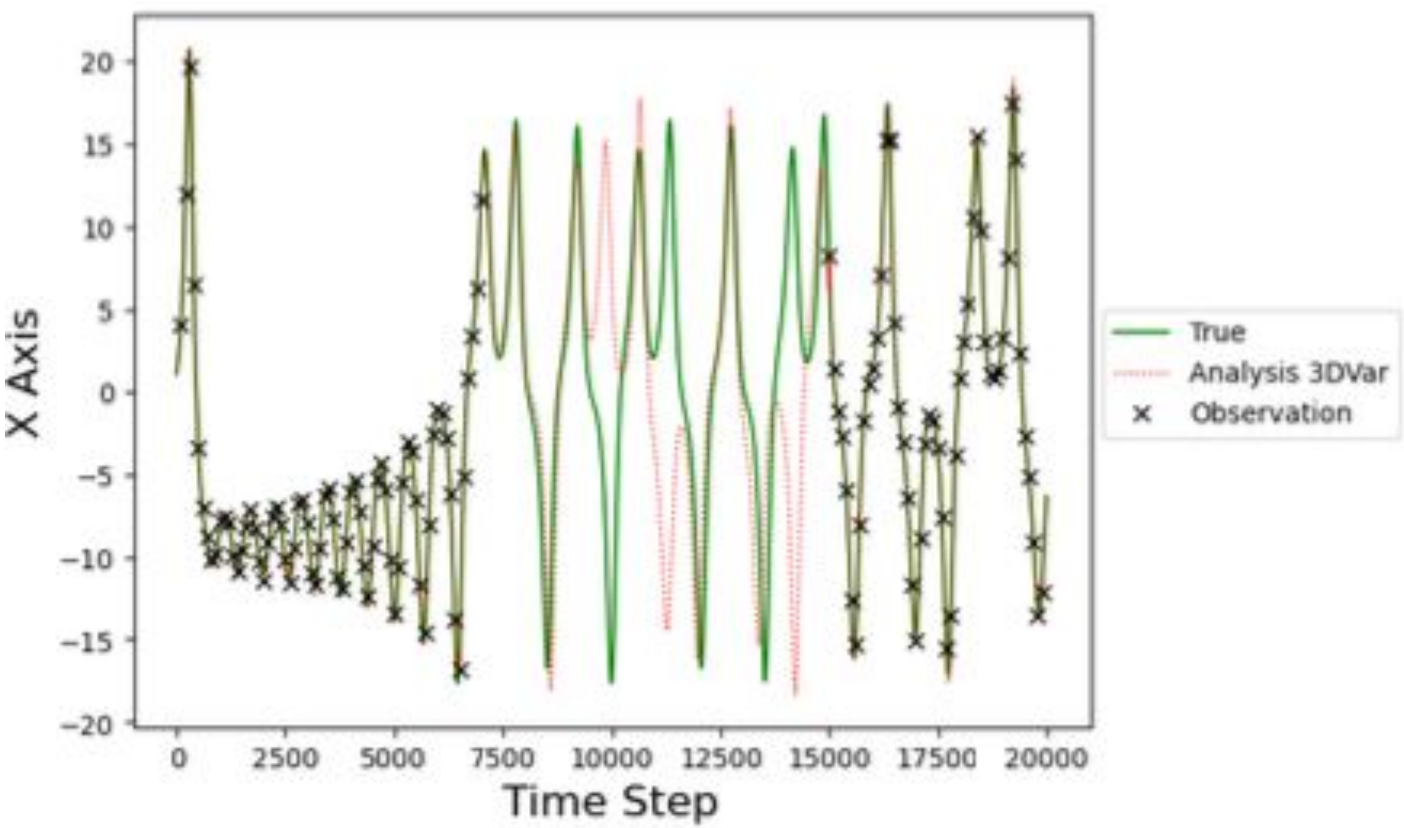
(level of noise) (random value)

$$\mathbf{P}^b = \begin{bmatrix} (0.5)^2 & 0 & 0 \\ 0 & (0.5)^2 & 0 \\ 0 & 0 & (0.5)^2 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} (0.1)^2 & 0 & 0 \\ 0 & (0.1)^2 & 0 \\ 0 & 0 & (0.1)^2 \end{bmatrix}$$

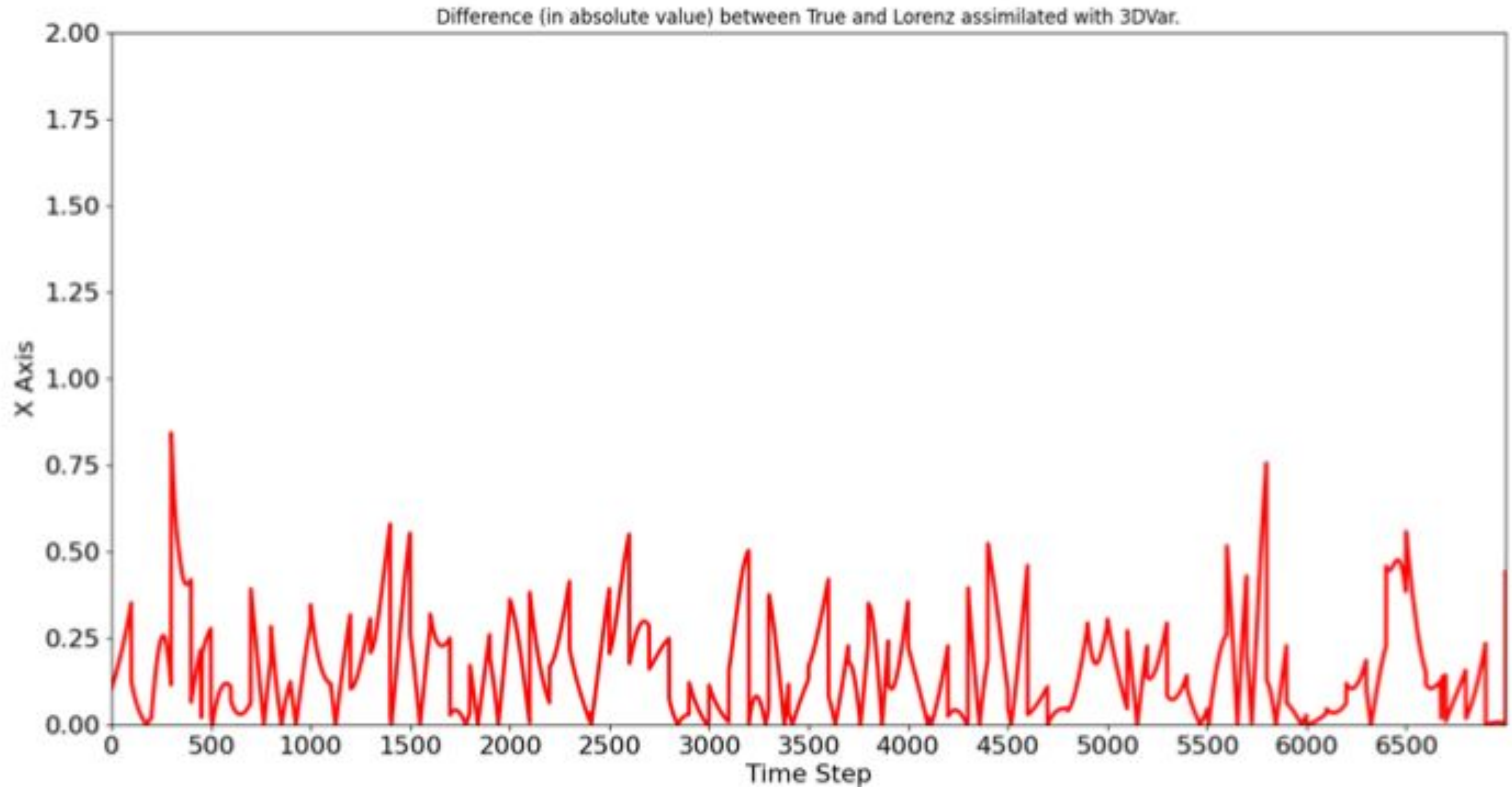
DA: cellular neural network (unsupervised NN)

- **Numerical results:** 3D-Var



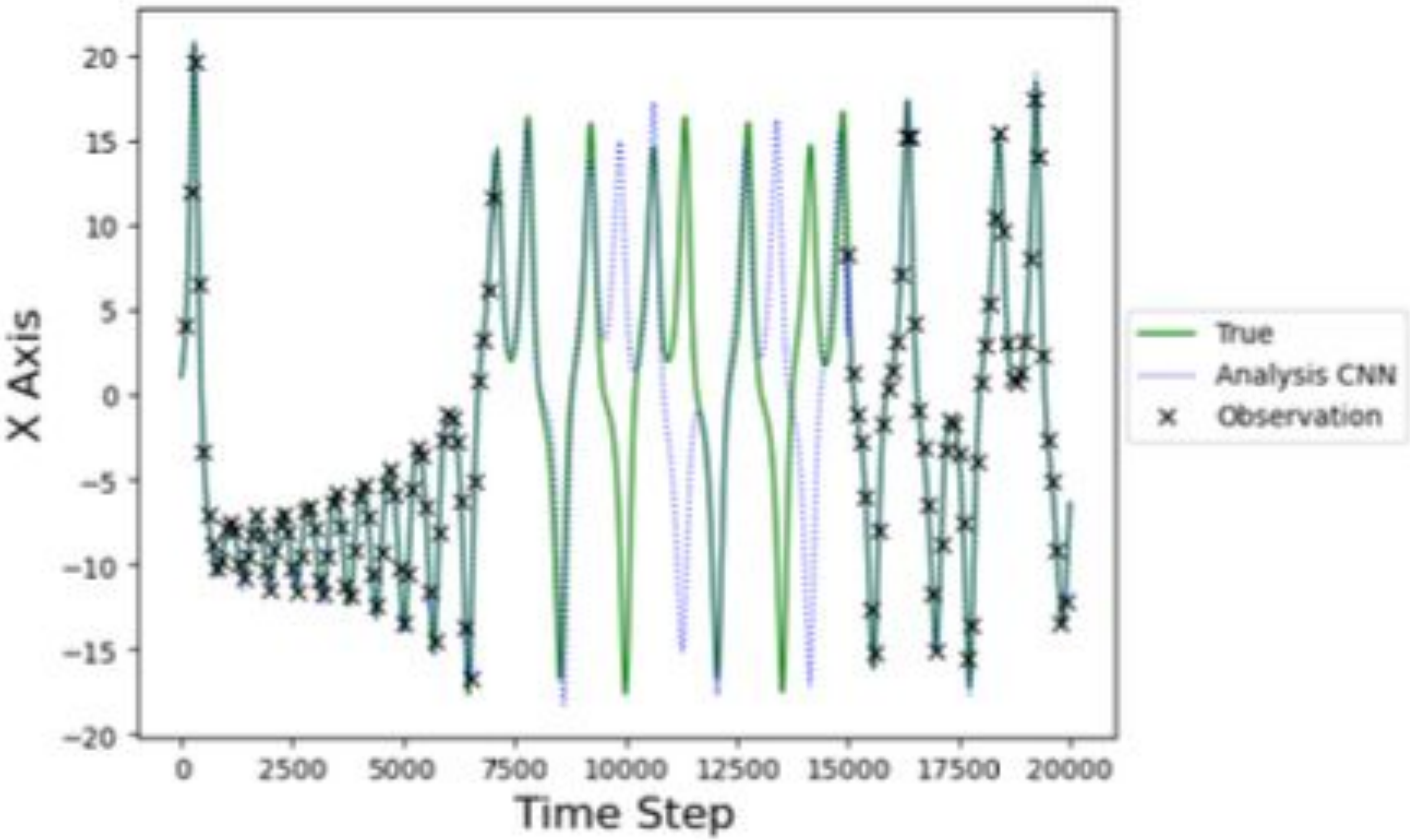
DA: cellular neural network (unsupervised NN)

- **Numerical results:** 3D-Var



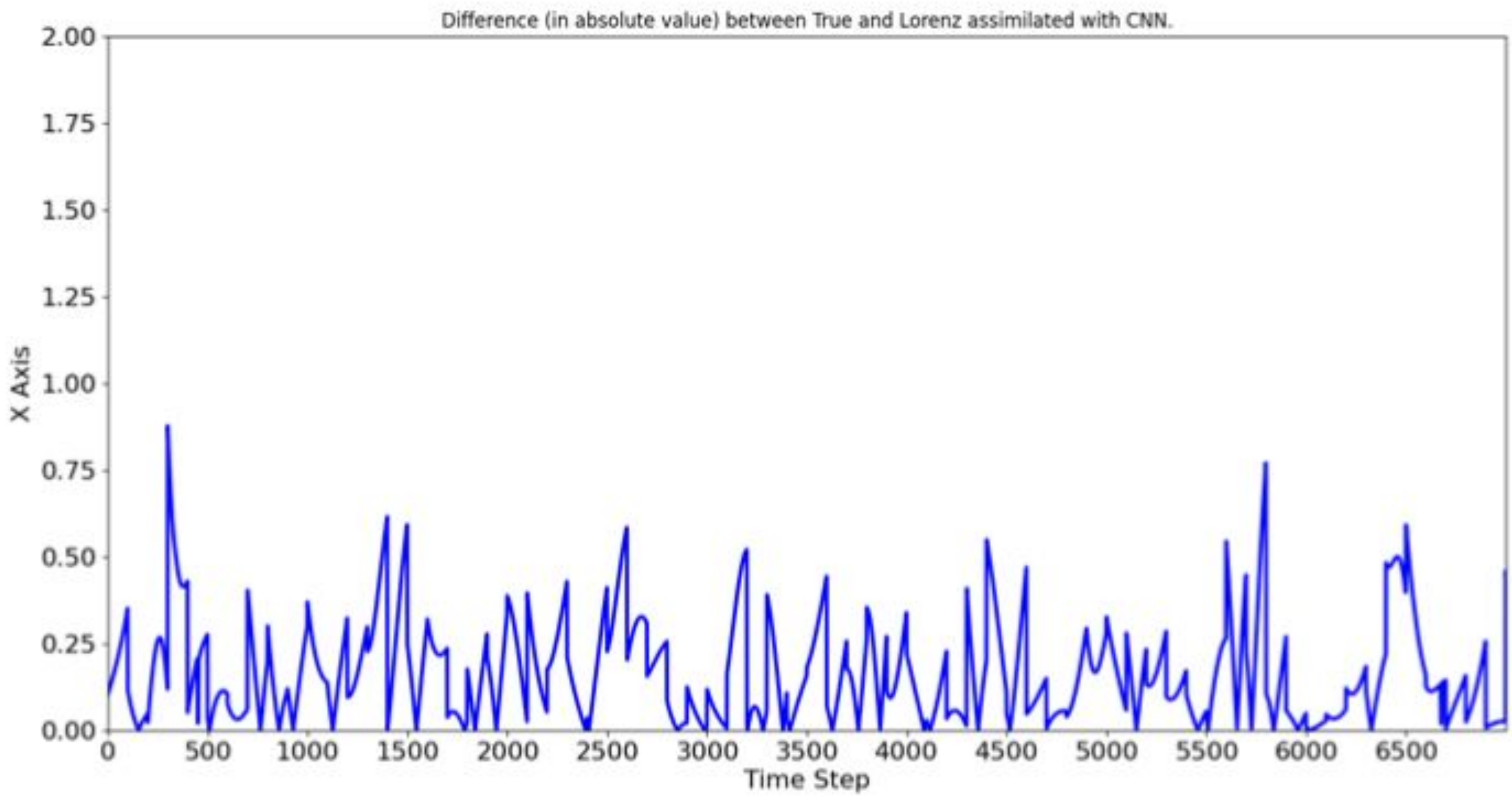
DA: cellular neural network (unsupervised NN)

- **Numerical results:** Cellular NN



DA: cellular neural network (unsupervised NN)

- **Numerical results:** Cellular NN



DA: cellular neural network (unsupervised NN)

Journal of Computational Science 90 (2025) 102587

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Data assimilation by cellular neural network applied to Lorenz-63 system

César Magno Leite de Oliveira Júnior [a],*, Antonio Mauro Saraiva [a], Alexandre Cláudio Botazzo Delbem [a], Haroldo Fraga de Campos Velho [b], Gerônimo Gallarreta Zubiaurre Lemos [b], Fabrício Pereira Härter [c]

DA: cellular neural network (unsupervised NN)

Conclusions:

1. Unsupervised neural network as a data assimilation method.

1. Complexity 3D-Var ~ $O(N^2)$
   Complexity Cell-NN ~ $O(N)$

1. 3D-Var errors ~ Cell-NN errors

CeNN-DA: Asymptotic behavior

$$\frac{d\mu_{ij}^{r}(\tau)}{d\tau} = -\mu_{ij}^{r}(\tau) + \sum_{(k,l)\in N_r(i,j)} A_{ij,kl}^{\mathrm{DA}}\, v_{kl}(\tau) + \sum_{(k,l)\in N_r(i,j)} B^{\mathrm{DA}}(\tau)_{ij,kl}\, u_{kl}$$

Assuming DA is working (v_{kl} ~ 0):

$$\lim_{\tau\to\infty}\left\{\frac{d\mu_{ij}(\tau)}{d\tau}\right\} \to \text{zero}$$

$$\sum_{(k,l)\in N_r(i,j)} A_{ij,kl}^{\mathrm{DA}}\, v_{kl}(\tau) + \sum_{(k,l)\in N_r(i,j)} B^{\mathrm{DA}}(\tau)_{ij,kl}\, u_{kl}(\tau) = \mu_{ij}(\tau)$$

CeNN–DA: Asymptotic behavior

$$\frac{1}{w_b}\sum \mathbf{B}_{ij}\,\boldsymbol{u} = \frac{1}{w_b}\sum \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{ij} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix} = \begin{bmatrix} u^s_{11} & u^s_{12} & u^s_{13} \\ u^s_{21} & u^s_{22} & u^s_{23} \\ u^s_{31} & u^s_{32} & u^s_{33} \end{bmatrix} \equiv \boldsymbol{u}^s$$

$\boldsymbol{u}^s$ is the term $\sum \mathbf{B}_{ij}\,\boldsymbol{u}$ … and the CeNN equation:

$$\frac{d\boldsymbol{\mu}(\tau)}{d\tau} = -\boldsymbol{\mu} + \boldsymbol{u}^s$$

with the solution:

$$\boldsymbol{\mu}(\tau) = e^{-\boldsymbol{\mu}\,\tau}\,\boldsymbol{\mu}(0) + \int_0^{\tau} e^{-\boldsymbol{\mu}\,(\tau - s)}\boldsymbol{u}^s\,ds$$

CeNN-DA: Asymptotic behavior

Changing the variable: $\tau - s = \lambda \implies ds = -d\lambda$

$$\begin{aligned}\boldsymbol{\mu}(\tau) &= e^{-\boldsymbol{\mu}\tau}\boldsymbol{\mu}(0) - \left[\int_0^\tau e^{-\boldsymbol{\mu}\lambda} d\lambda\right] \boldsymbol{u}^s \\ &= e^{-\boldsymbol{\mu}\tau}\boldsymbol{\mu}(0) - \left[e^{-\boldsymbol{\mu}\tau} - \boldsymbol{I}\right] \left(\boldsymbol{\mu}^{-1}\boldsymbol{u}^s\right) \\ &= e^{-\boldsymbol{\mu}\tau}\left[\boldsymbol{\mu}(0) - \boldsymbol{\mu}^{-1}\boldsymbol{u}^s\right] + \boldsymbol{\mu}^{-1}\boldsymbol{u}^s\end{aligned}$$

CeNN-DA: Asymptotic behavior

For: $\tau \longrightarrow \infty$

$$\boxed{\boldsymbol{\mu}(\infty) \sim \boldsymbol{\mu}^{-1}(\infty)\, \boldsymbol{u}^s \implies \boldsymbol{\mu}(\infty) \sim (\boldsymbol{u}^s)^{1/2}}$$

Going back to the CeNN equation .....

$$\frac{d\boldsymbol{\mu}(\tau)}{d\tau} = -\boldsymbol{\mu} + \boldsymbol{A}\,\boldsymbol{\mu} + \boldsymbol{B}\,\boldsymbol{u}$$

CeNN-DA: Asymptotic behavior

Another path for asymptotic behavior: explicit Euler time integration:

$$\boldsymbol{\mu}^{n+1} = (1-\Delta\tau)\boldsymbol{\mu}^{n} + \Delta\tau\,[\boldsymbol{A}\,\boldsymbol{v}^{n} + \boldsymbol{B}^{n}\,\boldsymbol{u}]$$

Moving: $\boldsymbol{\mu} \rightarrow \boldsymbol{X}$

$$\boldsymbol{X} = c_1\,\boldsymbol{X} + c_2\,[\boldsymbol{A}\,\boldsymbol{g}(\boldsymbol{X}) + \boldsymbol{B}(\boldsymbol{X})\,\boldsymbol{u}] \equiv h(\boldsymbol{X})$$

Now, it'd be necessary to proof that function $h(\boldsymbol{X})$ is a contraction: $h(\boldsymbol{X})$ follows the Lipschitz condition.

CeNN-DA: Asymptotic behavior

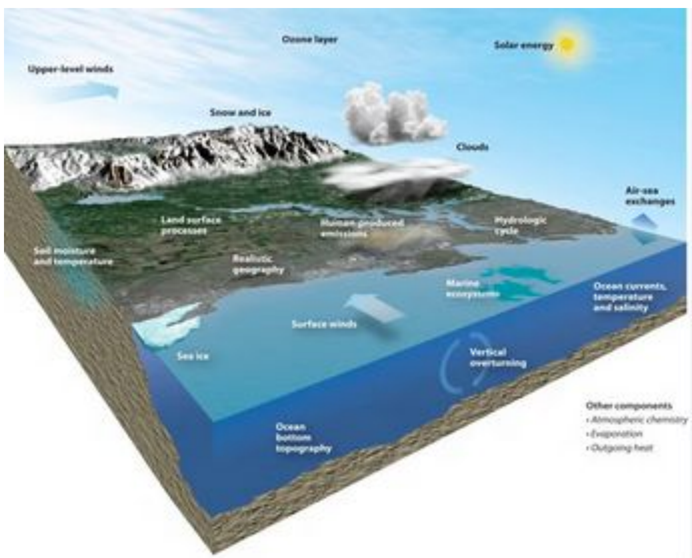
Lipschitz condition:

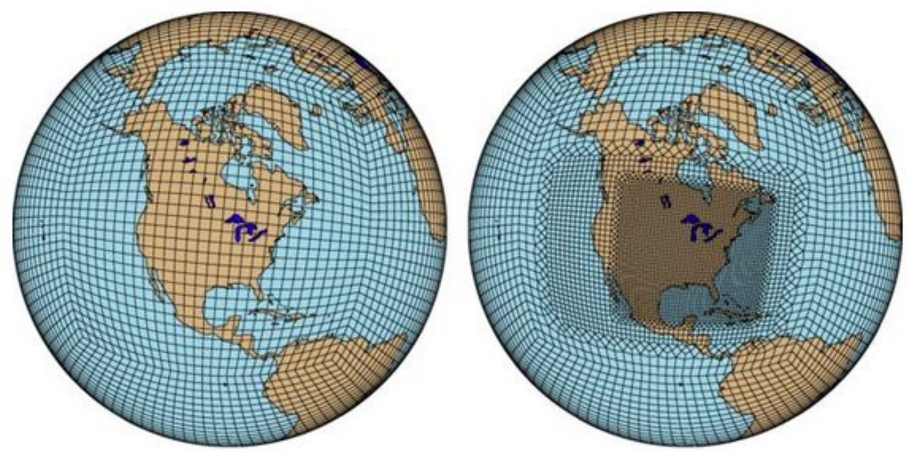
$$\|h(\boldsymbol{X}) - h(\boldsymbol{Y})\| < L \, \|\boldsymbol{X} - \boldsymbol{Y}\|$$

DA: cellular neural network (unsupervised NN)

INPE–USP & CMCC: CESM2 Global model

CESM2: Community Earth System Model - version 2





DA: cellular neural network (unsupervised NN)

INPE–USP & CMCC: CESM2 Global model

CMCC operational data assimilation scheme:

Ensemble Adjustment Kalman Filter (EAKF)

INPE–USP & CMCC: coupled strategy

EAKF: background + observations-1 ⇒ pre-analysis

Cell-NN: pre-analysis + observations-2 ⇒ analysis

DA: cellular neural network (unsupervised NN)

INPE-USP & CMCC: CESM2 Global model

CMCC operational data assimilation scheme:

Ensemble Adjustment Kalman Filter (EAKF)

INPE-USP & CMCC: coupled strategy

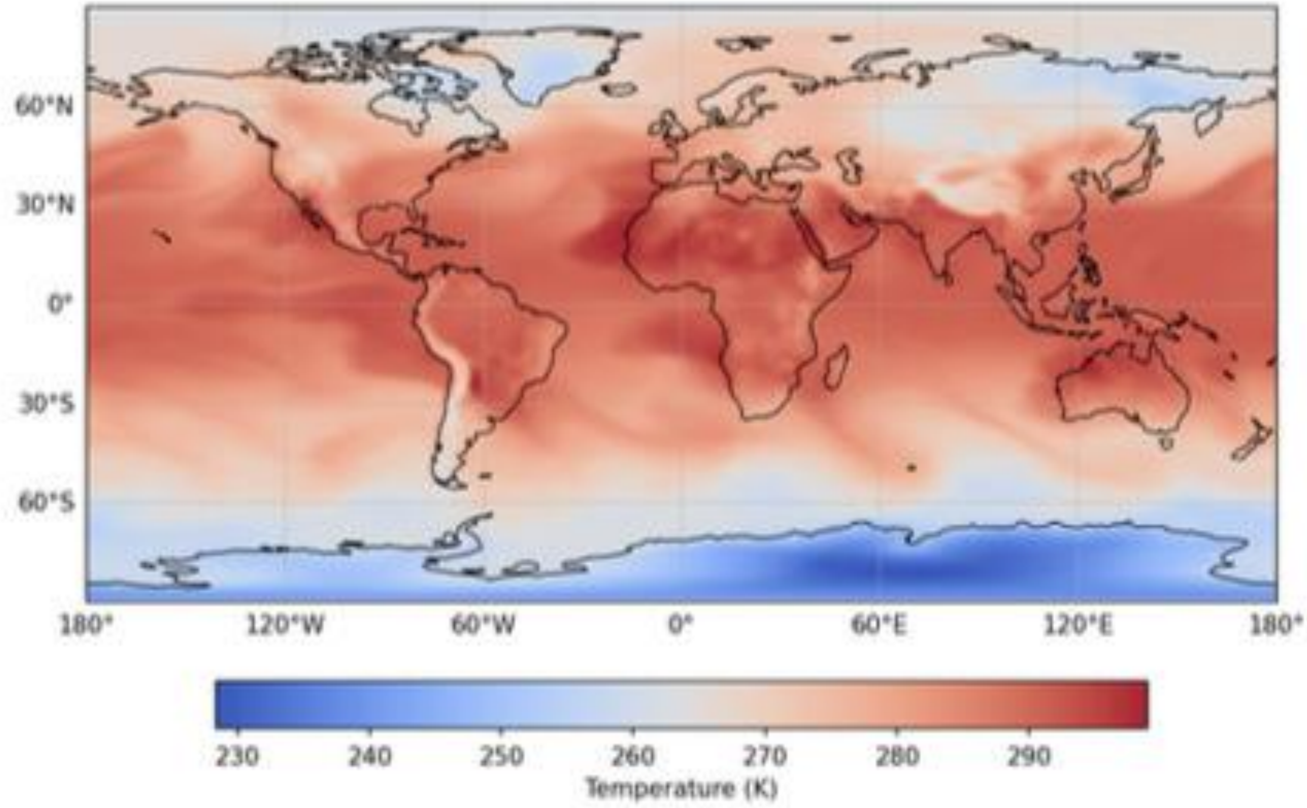
EAKF: background + observations-1 ⇒ pre-analysis

Cell-NN: pre-analysis + observations-2 ⇒ analysis

(**Spoiler:** CeNN is 13 faster than EAKF)

DA: cellular neural network (unsupervised NN)

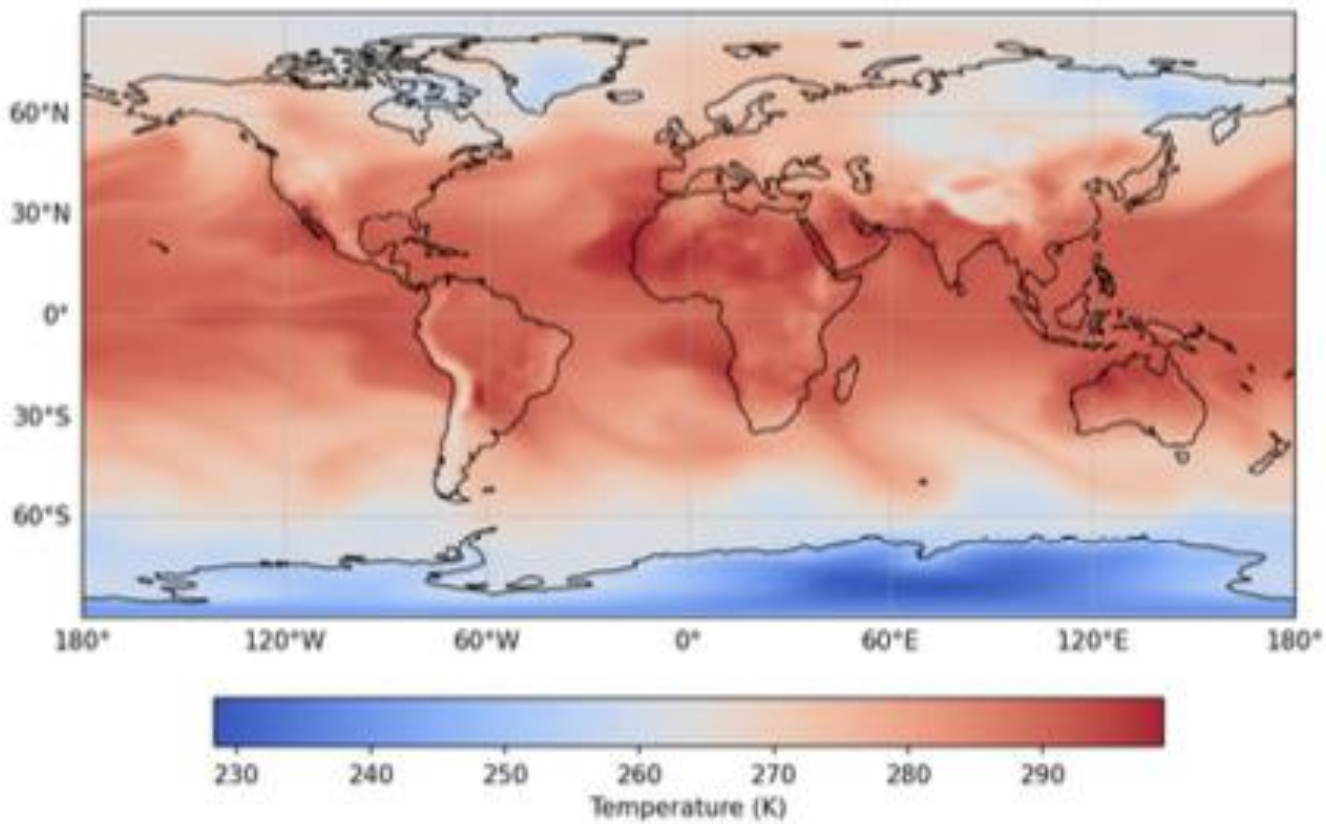
CESM2: assimilating AMSU-A by **EAKF**



(b) SPREADS: Temperature (T)

DA: cellular neural network (unsupervised NN)

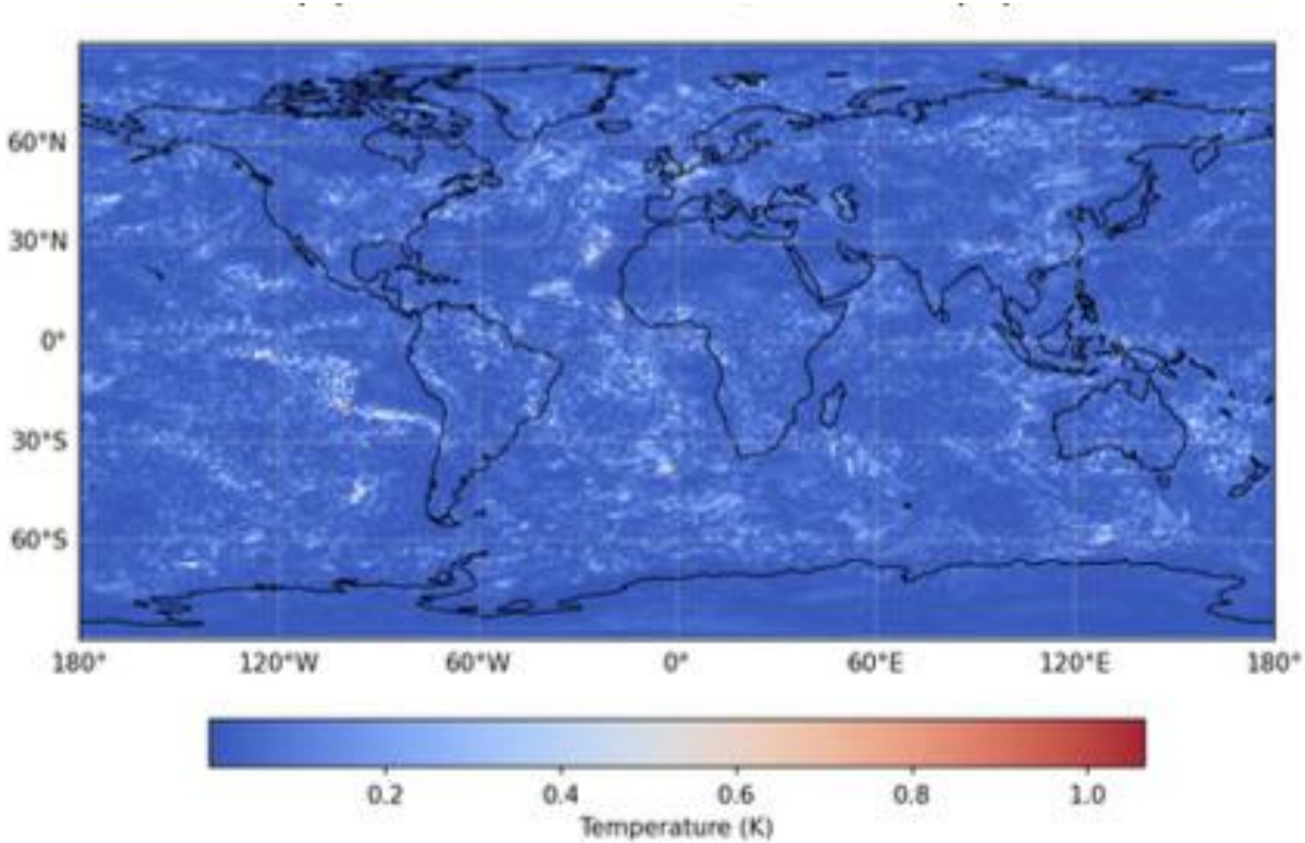
CESM2: assimilating AMSU-A by **CeNN**



(a) CeNN: Temperature (T)

DA: cellular neural network (unsupervised NN)

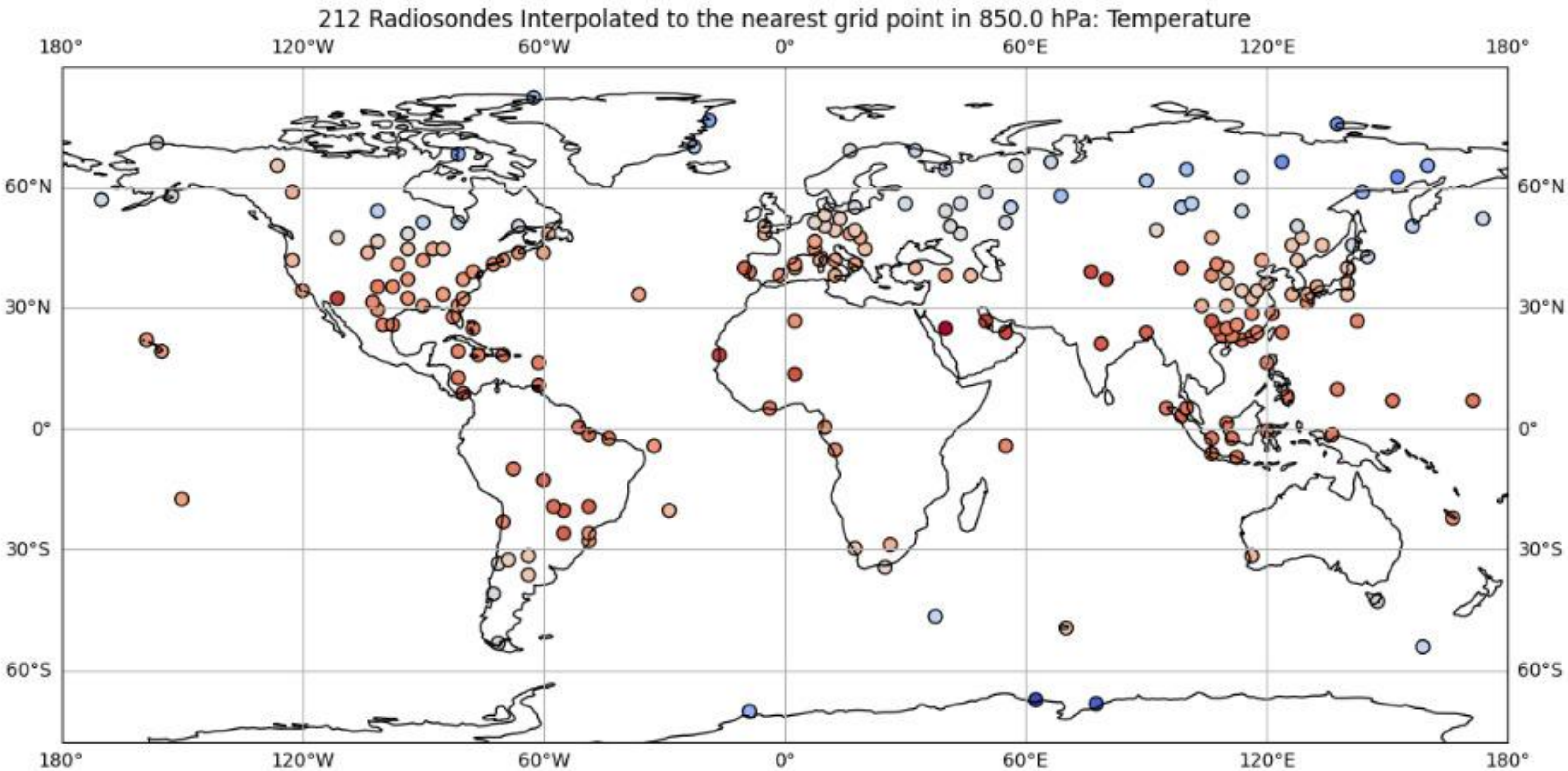
CESM2 (prediction 72hs ahead): **EAKF - CeNN**



(c) Absolute Difference $|CeNN - SPREADS|$

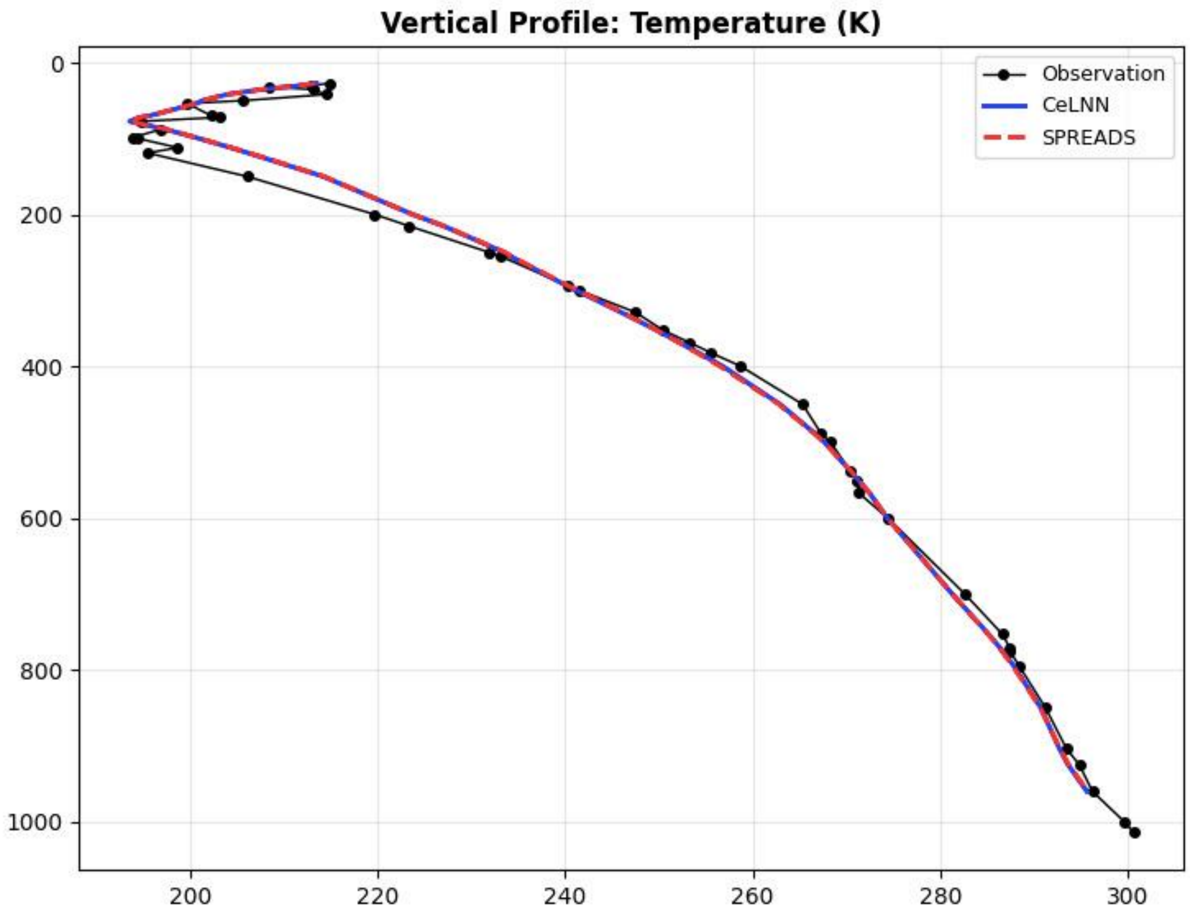
DA: cellular neural network (unsupervised NN)

CESM2 (prediction 72hs ahead): radiosonde

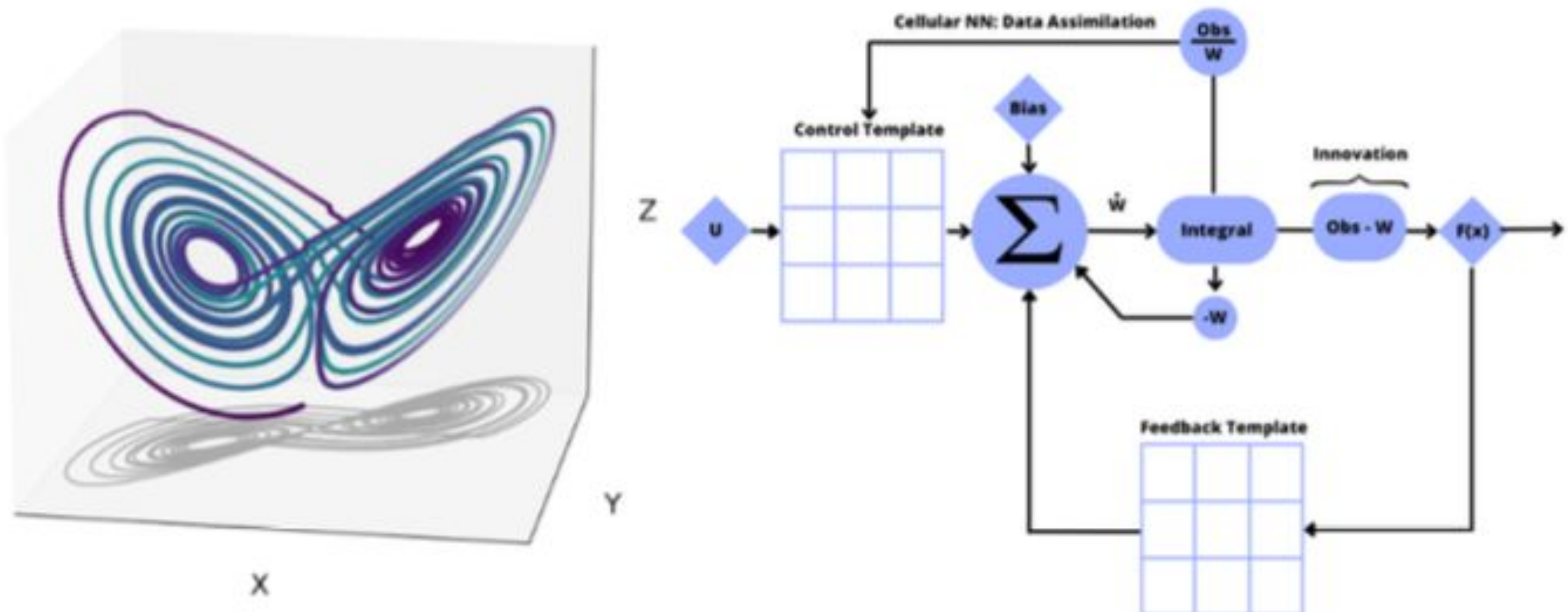


DA: cellular neural network (unsupervised NN)

CESM2 (prediction 72hs ahead): radiosonde



Data Assimilation by Cellular Neural Network



Thank you

For

Your Attention!