COSTA in a nutshell

Nils van Velzen

IMPOSE-Chimere meeting
Februari 2009

TU Delft
VORTECH BV
COMPUTING AND SIMULATION SOFTWARE TECHNOLOGY

Outline

- Background
- COSTA
  - Components
  - Dynamic models in COSTA
- Using COSTA
  - Automatic parallelization
  - Using parallel models
- Applications of COSTA
  - WAQUA/TRIWAQ
  - Lotos-Euros

Background

- Usage of data assimilation in various areas:
  - Meteorology
  - Oceanography
  - Chemistry
  - etc
- Custom implementations for data assimilation and calibration

Background

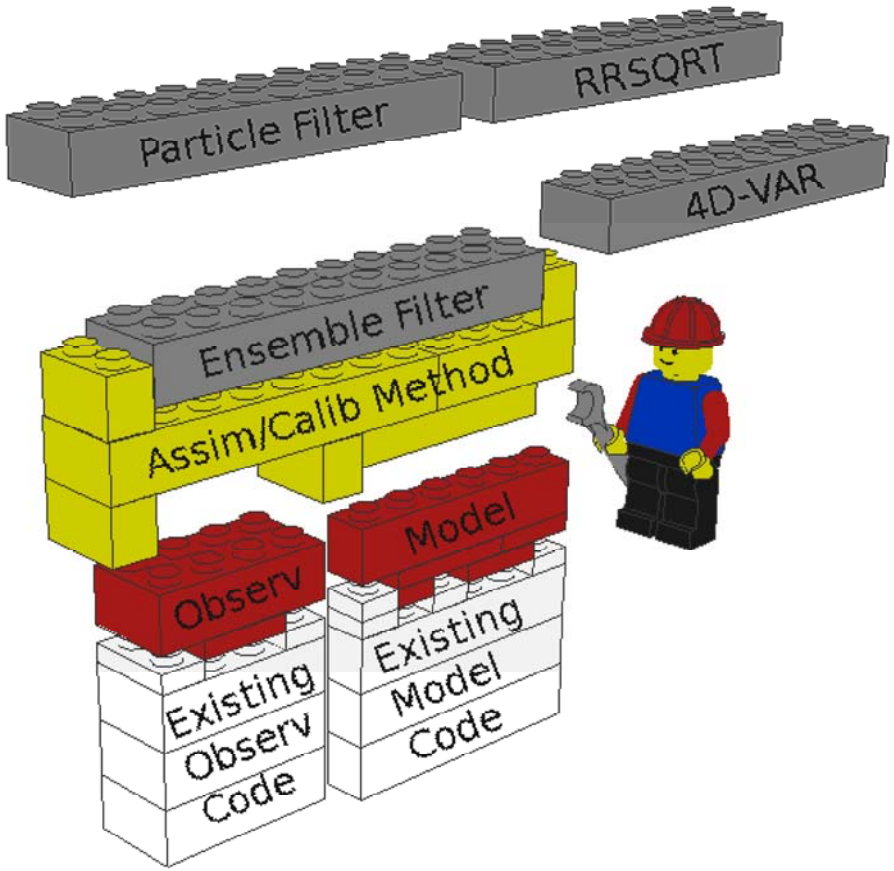
- Why people think they have to develop a custom implementation of a DA method:
  - Computational efficient
  - Need to handle model specific issues
  - Independent
- Are these assumptions correct? Issues are:
  - Which method to implement?
  - Expensive
  - Incompatible
  - Potentially full of bugs

COSTA

- A problem solving environment for data assimilation and calibration
  - Components and their interface
  - Data assimilation methods
  - (Software) development philosophy
  - Platform for exchanging models and methods
- Free software (LGPL)

COSTA components

- Model, observations and da-method



Model Component

- Model in COSTA (formal)

$$\frac{d\mathbf{x}(t)}{dt} = M\left(\mathbf{x}(t), \mathbf{p}, \mathbf{u}(t), \mathbf{w}(t)\right)$$

- State of a model instance $\mathbf{x}, \mathbf{u}, \mathbf{p}, \mathbf{w}, t$
- Methods to get or change the model-state

Model Component

- Propagate the model state-vector

$$\mathbf{x}(t) = \int_{t}^{t+\Delta t} M\left(\mathbf{x}(t), \mathbf{p}, \mathbf{u}(t), \mathbf{w}(t)\right) dt$$

- Get, set, axpy for $\mathbf{x}(t), \mathbf{p}, \mathbf{u}(t), \mathbf{w}(t)$

- GetObsValues: $y(t) = H\left(\mathbf{x}(t)\right)$

Steps to create a COSTA model

- Identify the state-vector of your model and those parameters and forcings (you think) you want to use
- Isolate the model timestep
- This is 90% of the work and has to be done for a custom data assimilation implementation as well
- COSTA provides tools to simplify this work

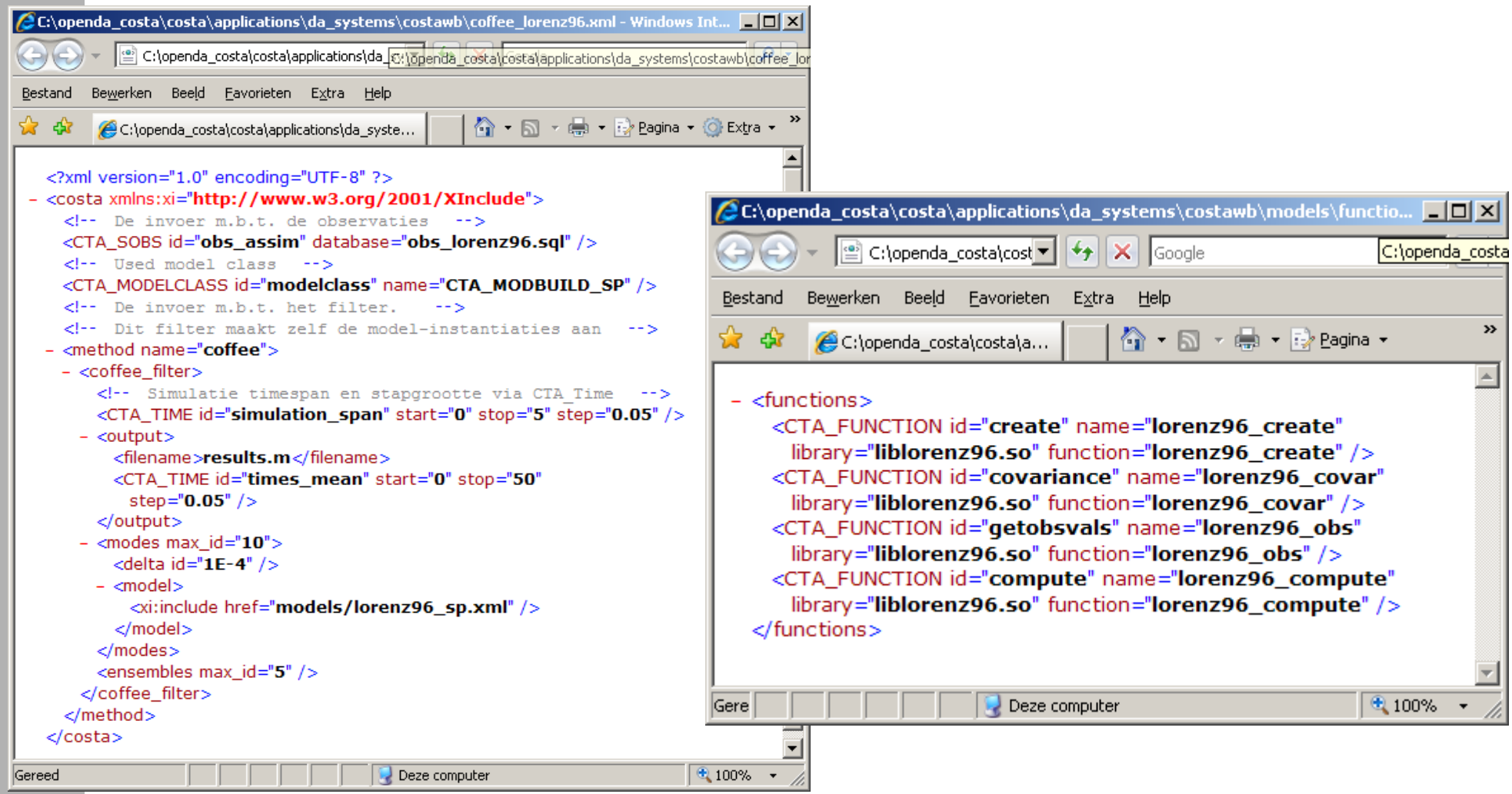
Tools for creating COSTA models

- Simple model builder
  - Handles the “object oriented” aspects of the model
  - You only need to implement 4 functions
- Model combiner
  - Combine multiple COSTA models in one composite model
  - Create a stochastic model from a deterministic model
- Black Box model builder
  - Create a COSTA component for a model without changing the model code
- Support for parallel models

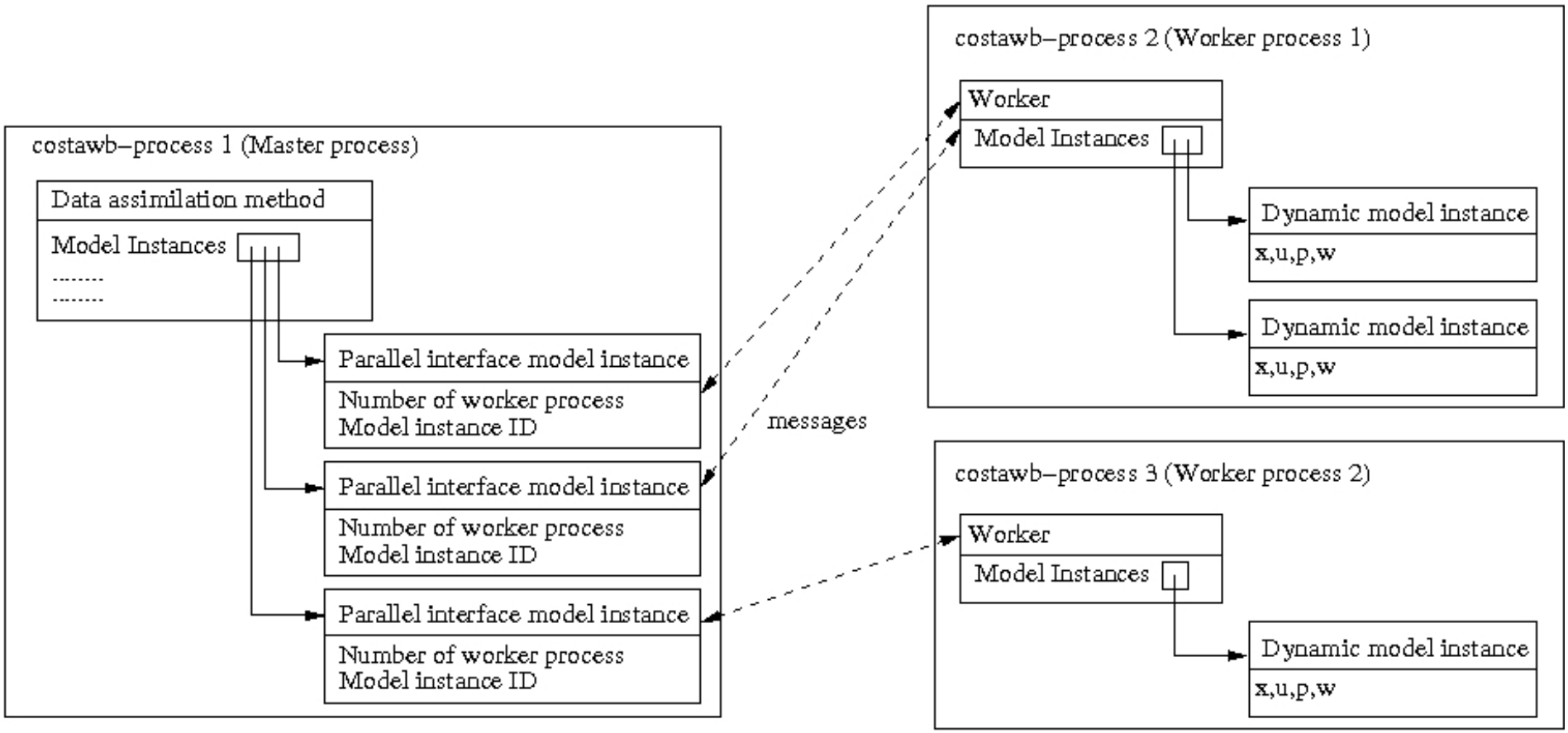
Using COSTA

- Use the “costawb” program to combine a model with a da-method without any programming
- Model components are dynamic libraries and are linked to the “costawb” program

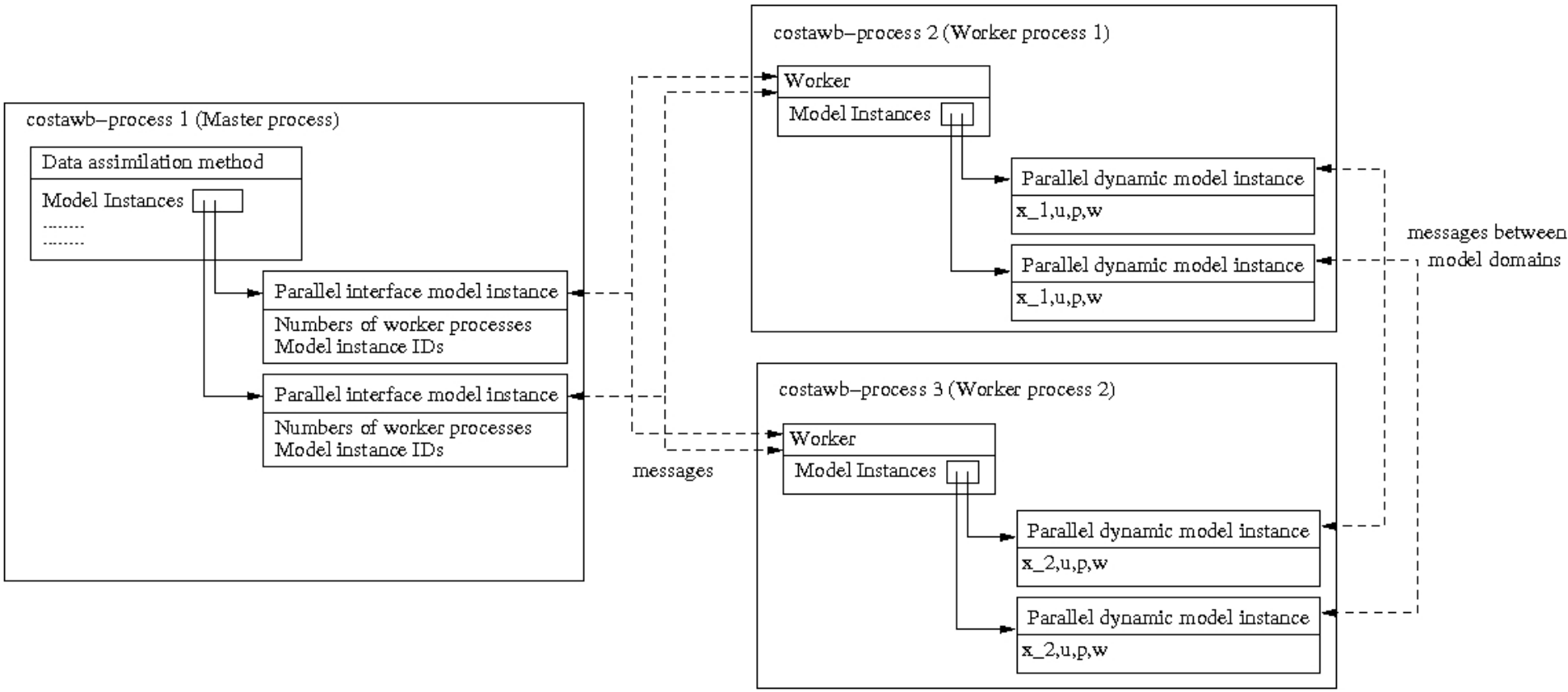
Using COSTA


C:\openda_costa\costa\applications\da_systems\costawb\coffee_lorenz96.xml - Windows Int...
Bestand Bewerken Beeld Favorieten Extra Help
<?xml version="1.0" encoding="UTF-8" ?>
<costa xmlns:xi="http://www.w3.org/2001/XInclude">
<!-- De invoer m.b.t. de observaties -->
<CTA_SOBS id="obs_assim" database="obs_lorenz96.sql" />
<!-- Used model class -->
<CTA_MODELCLASS id="modelclass" name="CTA_MODBUILD_SP" />
<!-- De invoer m.b.t. het filter. -->
<!-- Dit filter maakt zelf de model-instantiaties aan -->
<method name="coffee">
<coffee_filter>
<!-- Simulatie timespan en stapgrootte via CTA_Time -->
<CTA_TIME id="simulation_span" start="0" stop="5" step="0.05" />
<output>
<filename>results.m</filename>
<CTA_TIME id="times_mean" start="0" stop="50"
step="0.05" />
</output>
<modes max_id="10">
<delta id="1E-4" />
<model>
<xi:include href="models/lorenz96_sp.xml" />
</model>
</modes>
<ensembles max_id="5" />
</coffee_filter>
</method>
</costa>
Gereed
Deze computer
100%
C:\openda_costa\costa\applications\da_systems\costawb\models\functio...
Bestand Bewerken Beeld Favorieten Extra Help
<functions>
<CTA_FUNCTION id="create" name="lorenz96_create"
library="liblorenz96.so" function="lorenz96_create" />
<CTA_FUNCTION id="covariance" name="lorenz96_covar"
library="liblorenz96.so" function="lorenz96_covar" />
<CTA_FUNCTION id="getobsvals" name="lorenz96_obs"
library="liblorenz96.so" function="lorenz96_obs" />
<CTA_FUNCTION id="compute" name="lorenz96_compute"
library="liblorenz96.so" function="lorenz96_compute" />
</functions>
Deze computer
100%

Automatic parallelization



Using parallel models

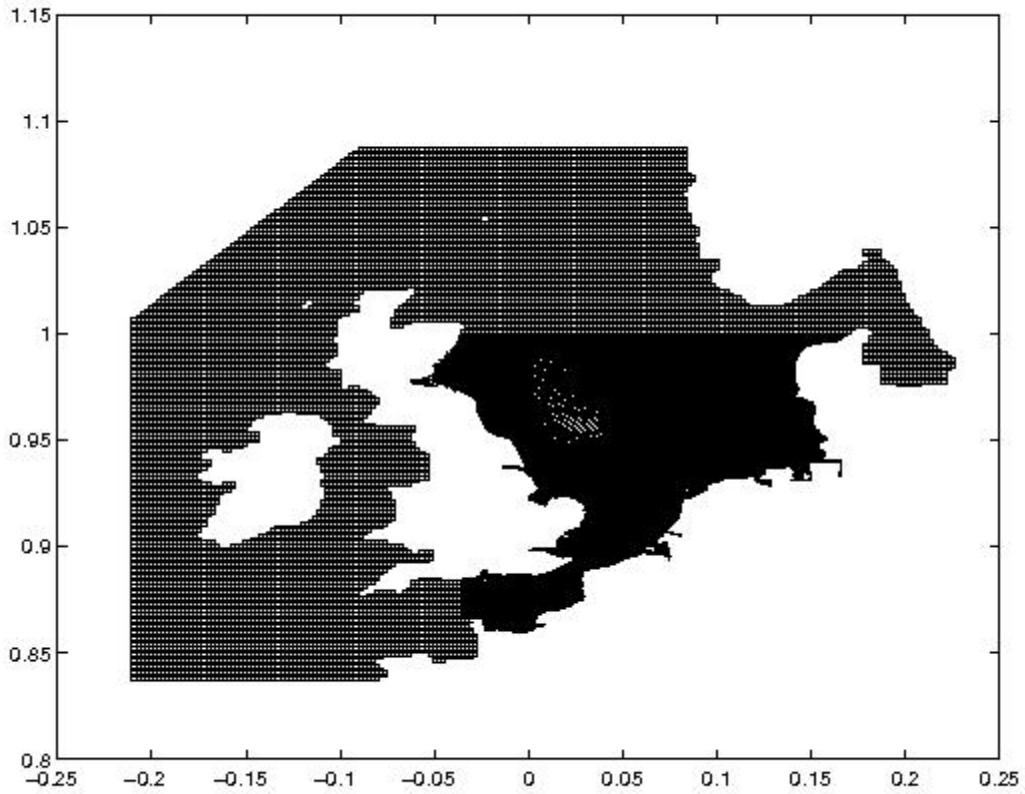


WAQUA/TRIWAQ

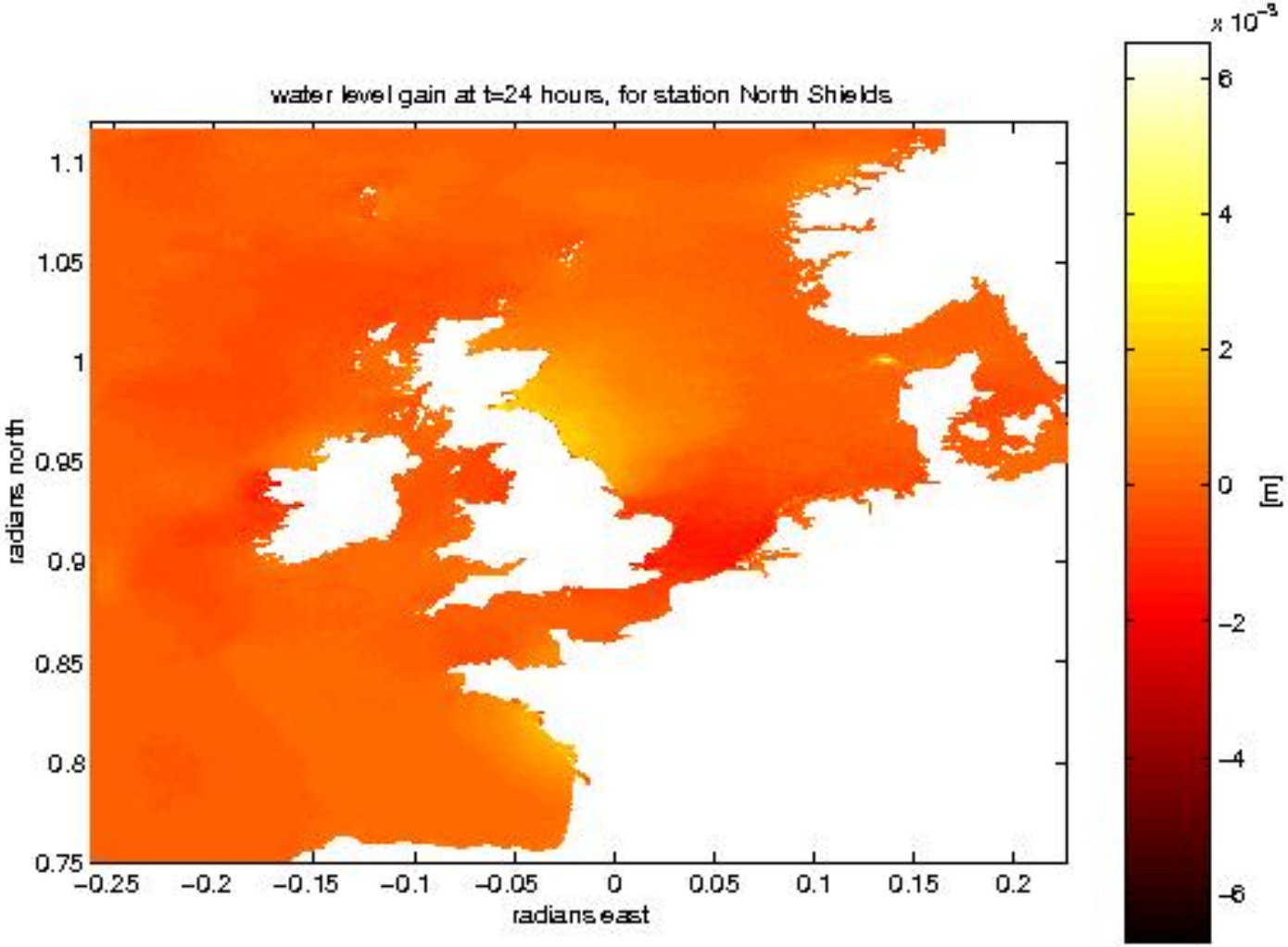
- Simulation model for shallow water
- Used operational by the Dutch Rijkswaterstaat
- Long history of data assimilation
  - Too expensive
  - Infexible

WAQUA/TRIWAQ

- Domain decomposition and parallel computing
- COSTA model component
- RRSQRT Filter



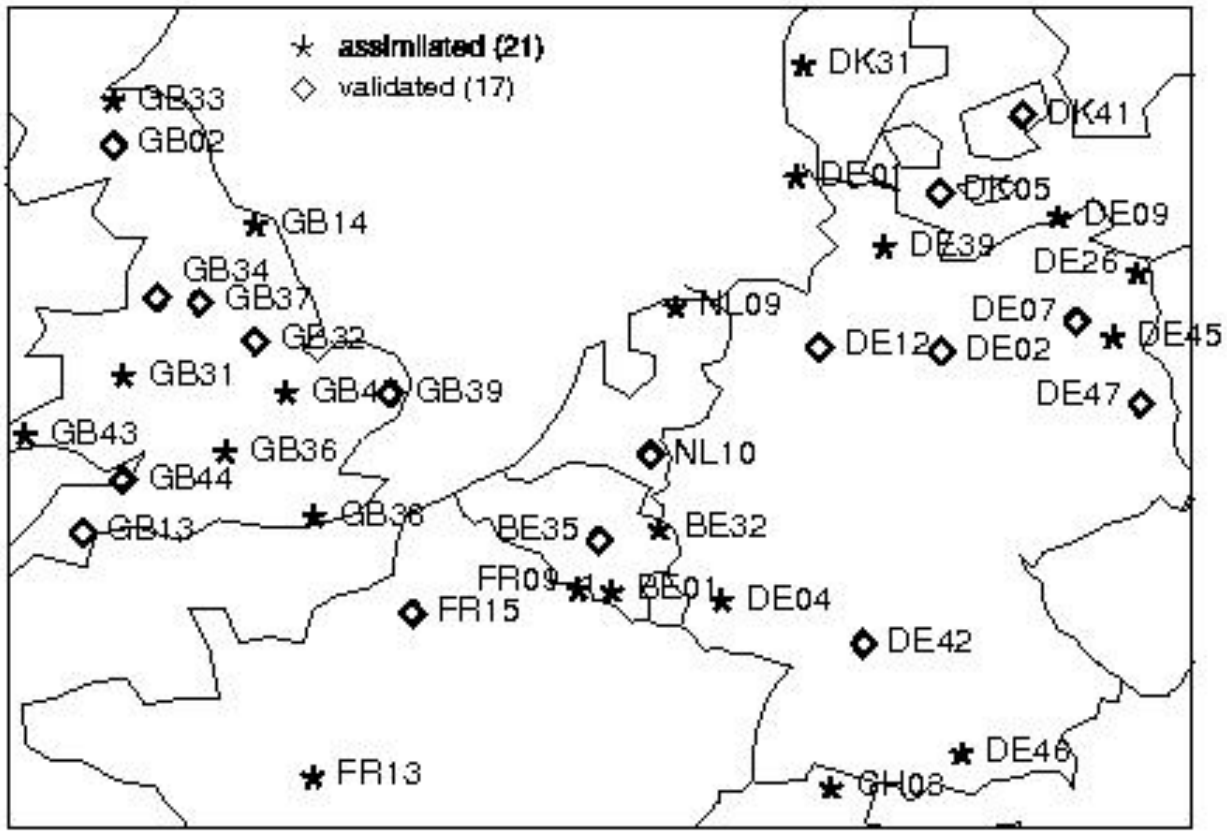
WAQUA/TRIWAQ



Lotos-Euros

- The LOTOS-EUROS model is an operational air quality focused on modeling the lower part of the troposphere
- Which da-method performs best?
- What noise model to be used
- Set up an ozone test-case

Lotos-Euros


assimilated (21)
validated (17)
GB33
GB02
GB14
GB34
GB37
GB32
GB31
GB40
GB39
GB43
GB36
GB44
GB13
GB38
FR13
FR15
FR09
BE01
BE35
BE32
NL09
NL10
DK31
DK41
DE01
DK05
DE09
DE39
DE26
DE07
DE45
DE12
DE02
DE47
DE04
DE42
DE46
GH08

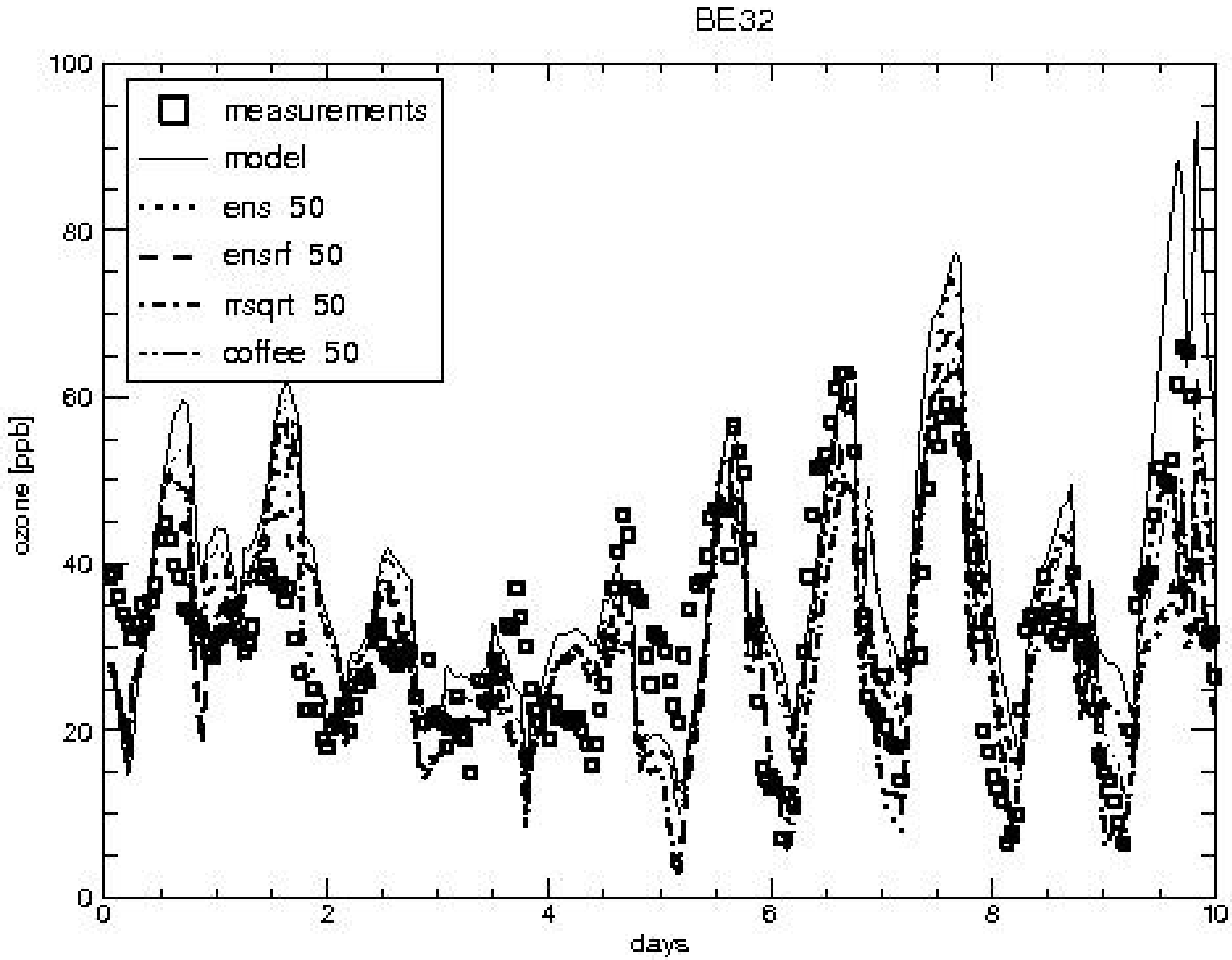
Lotos-Euros

- Nw Europe 40x40x4 grid cells
- Simulation period of one month
- Noise on the emissions

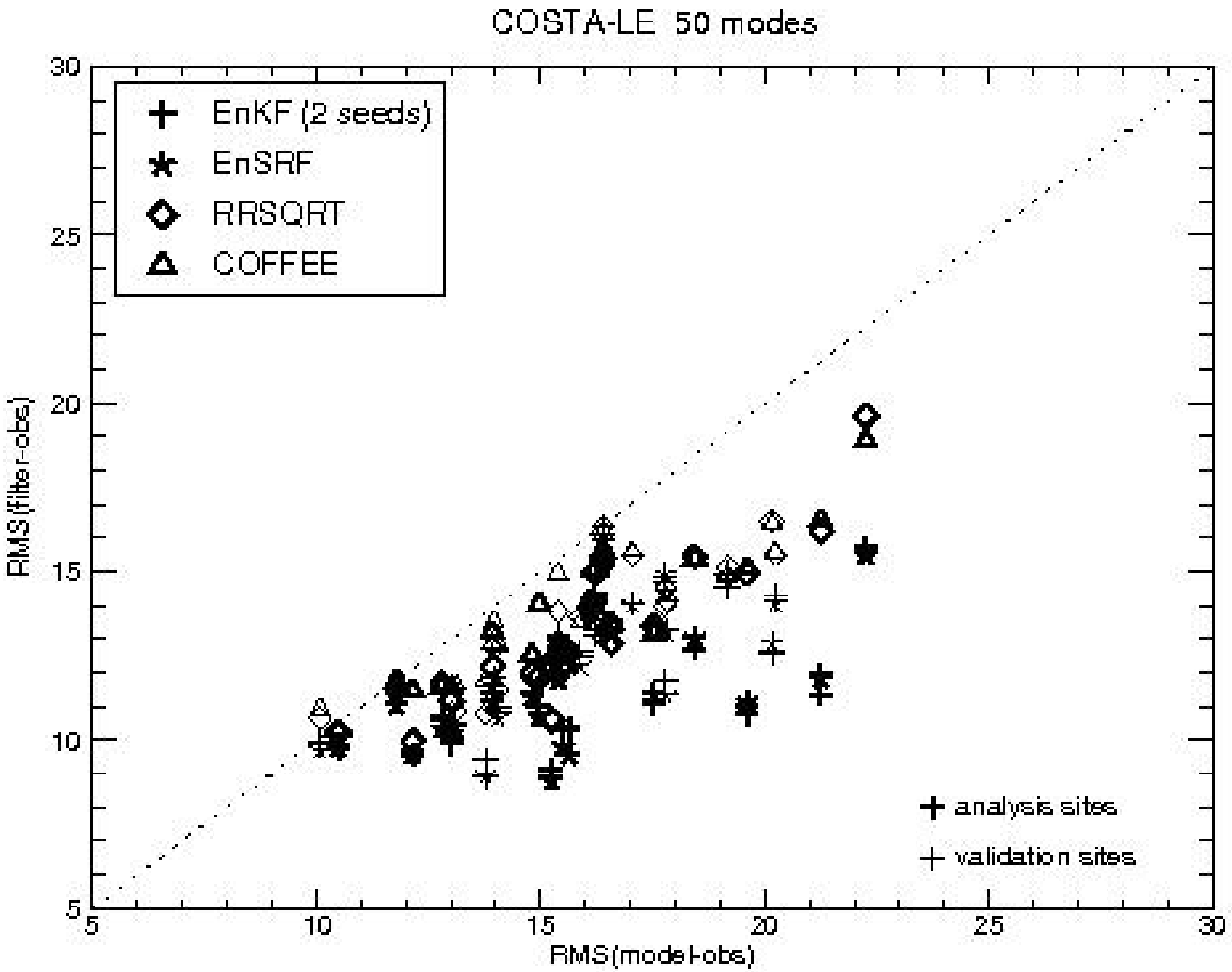
$$e_s^j\left(t_k\right)=\overline{e}_s^j\left(t_k\right)\left[1+\eta_s^j\left(t_k\right)\right],\ s=\{NO_x,VOC\}$$

- Compare Ensemble Kalman, Ensemble Kalman Square Root, RRSQRT and COFFEE

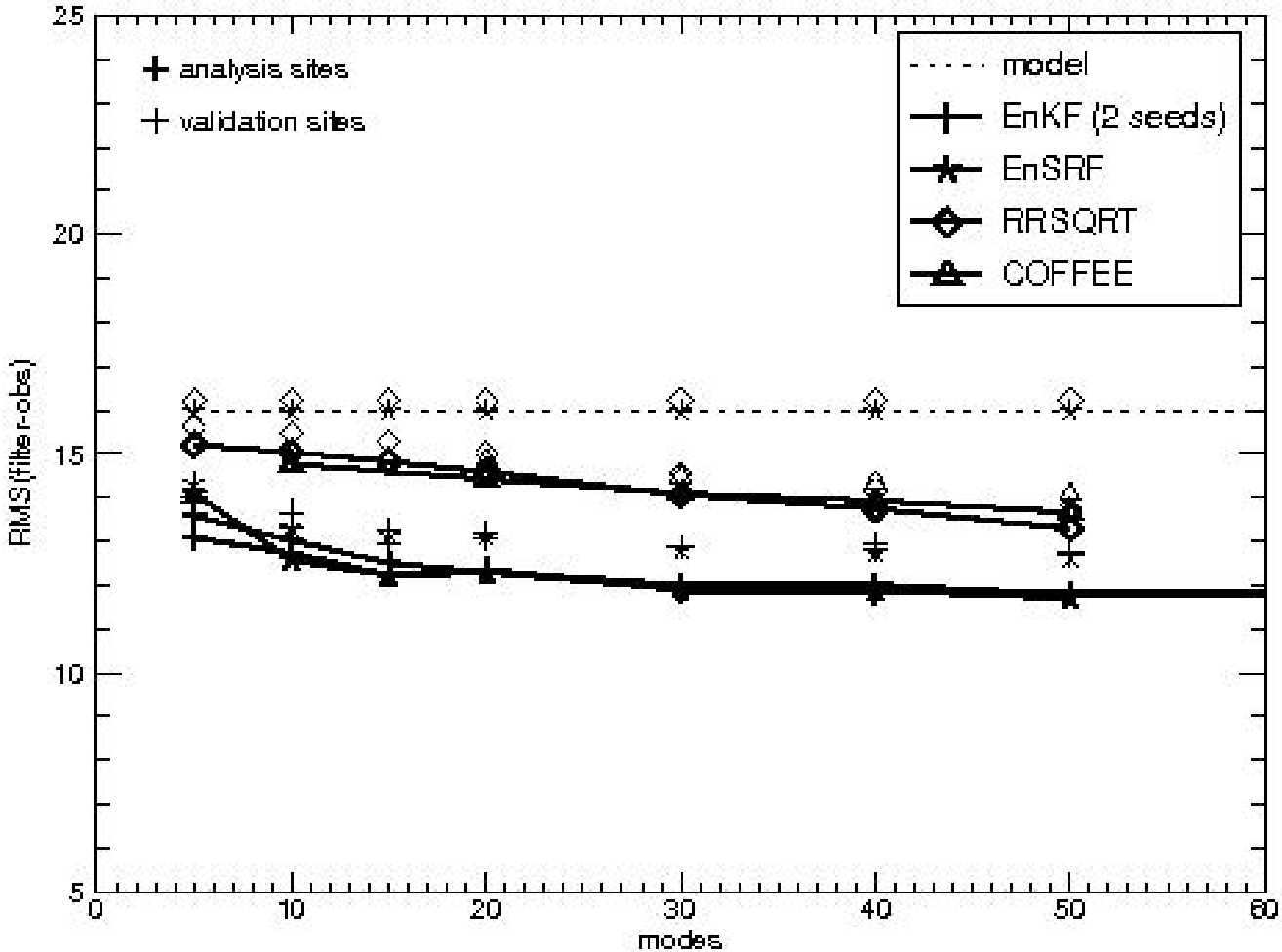
Lotos-Euros


BE32
measurements
model
ens 50
ensrf 50
rmsqrt 50
coffee 50
ozone [ppb]
days
0
20
40
60
80
100
0
2
4
6
8
10

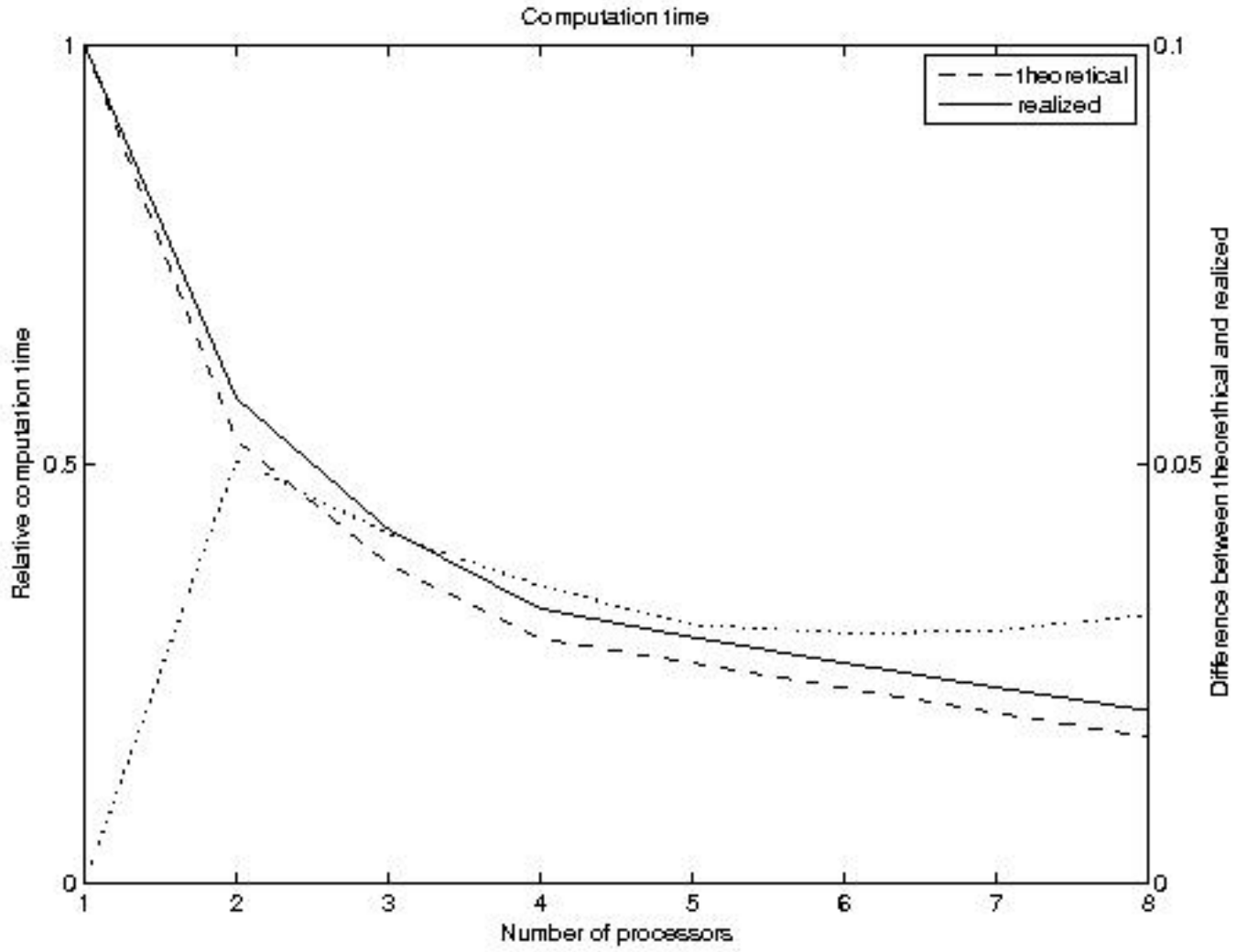
Lotos-Euros


COSTA-LE 50 modes
EnKF (2 seeds)
EnSRF
RRSQRT
COFFEE
analysis sites
validation sites
RMS(filter-obs)
RMS(model-obs)
5
10
15
20
25
30

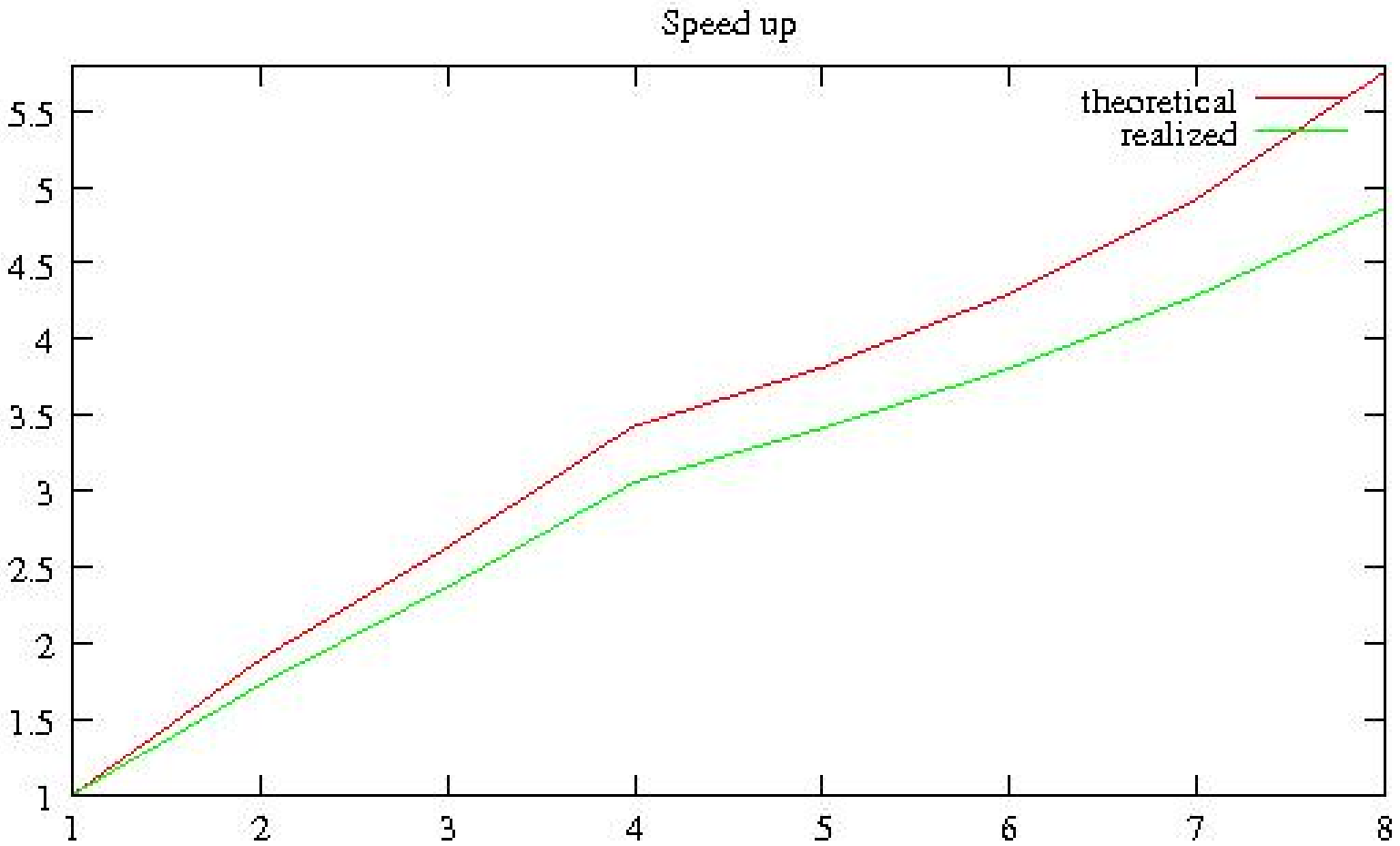
Lotos-Euros


analysis sites
validation sites
model
EnKF (2 seeds)
EnSRF
RRSQRT
COFFEE
RMS(filter-obs)
modes
25
20
15
10
5
0
10
20
30
40
50
60

Lotos-Euros


Computation time
theoretical
realized
Relative computation time
Difference between theorethical and realized
Number of processors
1
0.5
0
0.1
0.05
0
1
2
3
4
5
6
7
8

Lotos-Euros



Conclusions

- COSTA offers a flexible framework for using and developing data assimilation methods.
- Successful coupling with “real” models:
  - WAQUA/TRIWAQ
  - Lotos-Euros
  - Chimere

Conclusions

- Easy to use
- Complete basic functionality available:
  - Implementation available for all basic building blocks
  - Growing number of available methods
- Support for parallel computing