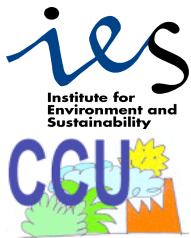


4D-Var Speedup

Arjo Segers
Peter Bergamaschi
Matteo Corazza

European Commission Joint Research Centre, Institute for Environment and Sustainability, Climate Change Unit, I-21027 Ispra (VA), Italy



Why necessary ?

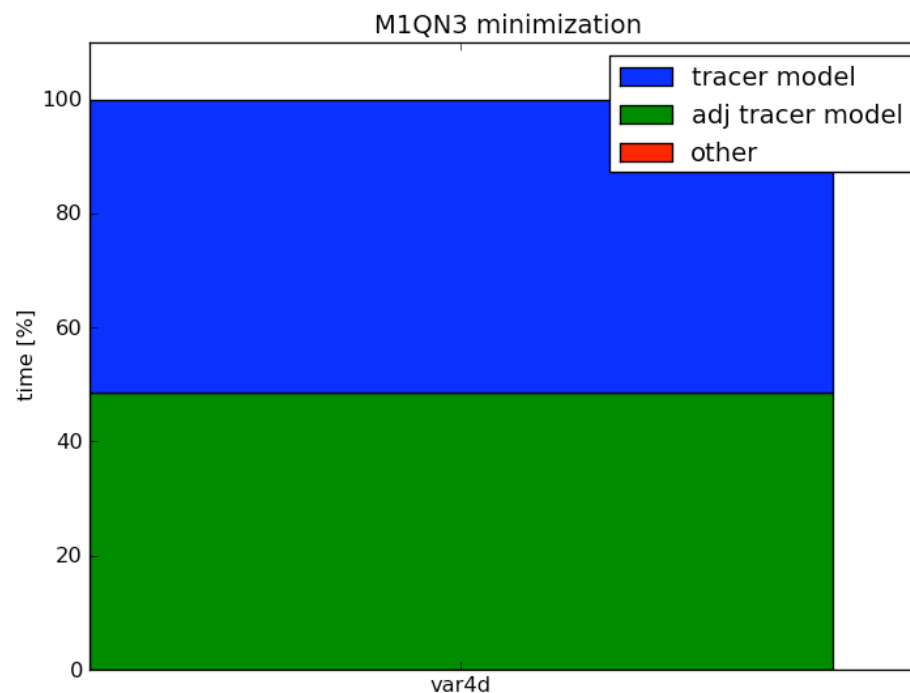
- Ambition level increasing:
 - higher resolutions
 - more iterations (temporal resolution increases!)
- Users are impatient:
 - runs could take 1-2 weeks
 - too long for semi-operational inversions (MACC)





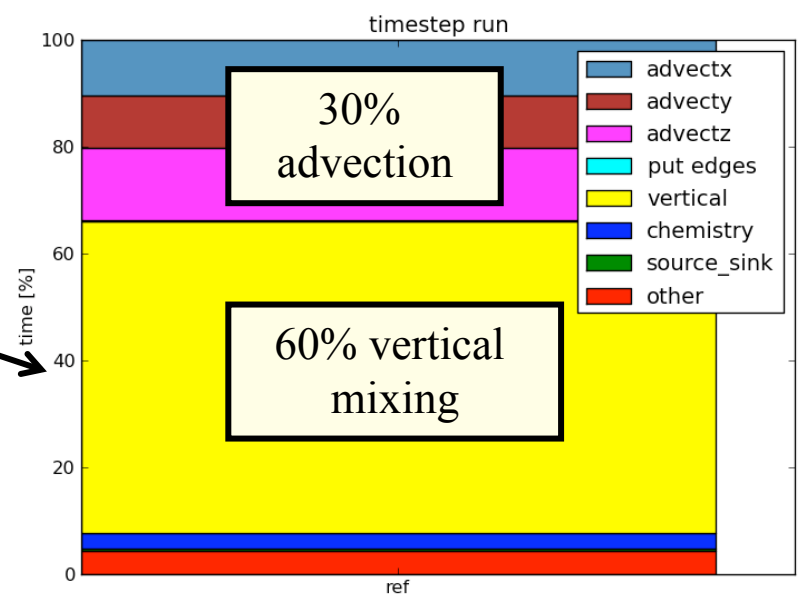
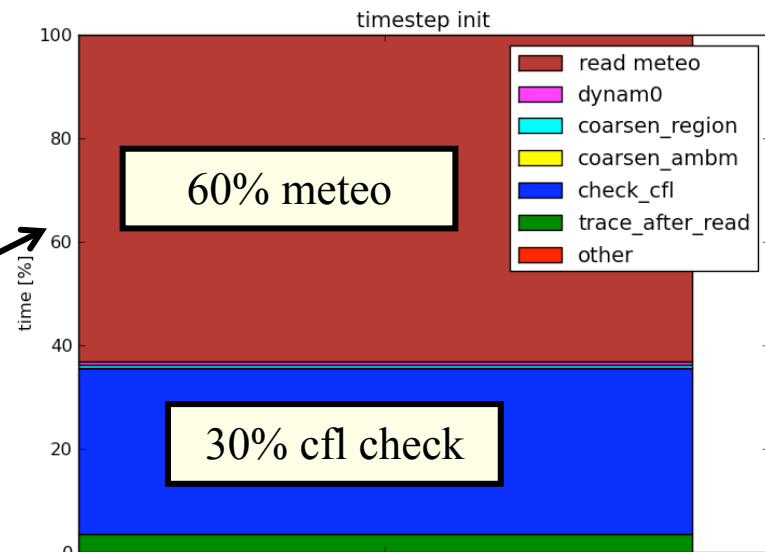
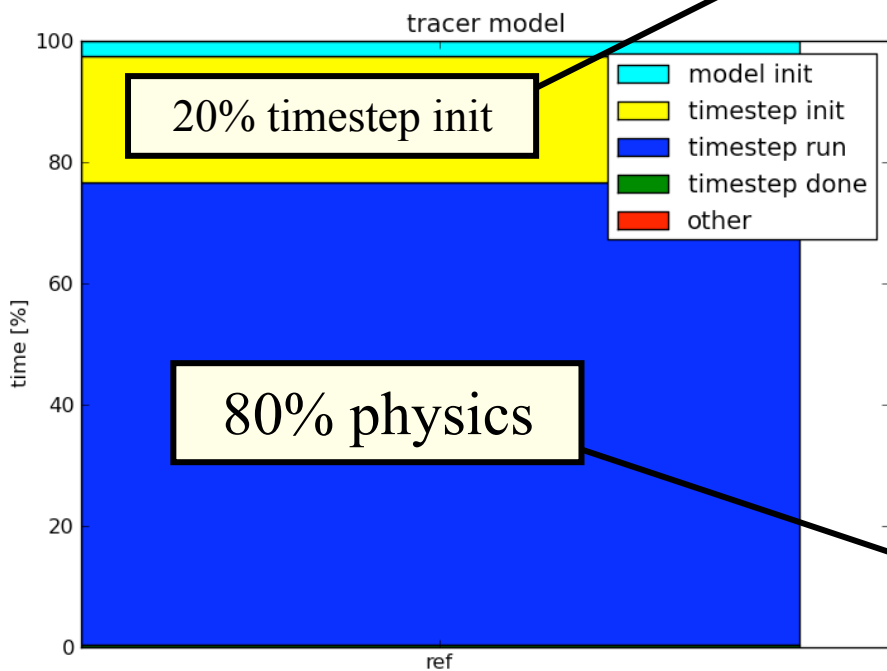
1. Check runtimes

- macc-mode 4D-var run:
 - global 6x4, zoom europe
 - 6+2 months
- ~100% of run time spent on **forward** and **adjoint** model runs





Spee-D-Var

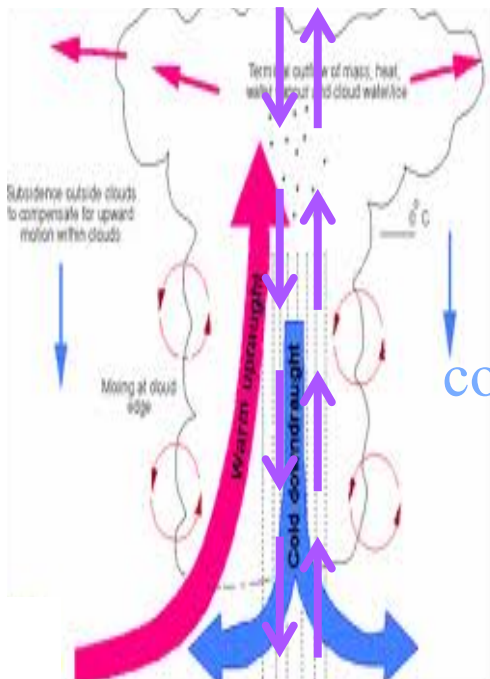




2. Serial optim: vertical mixing

Joint Research Centre

vertical diffusion



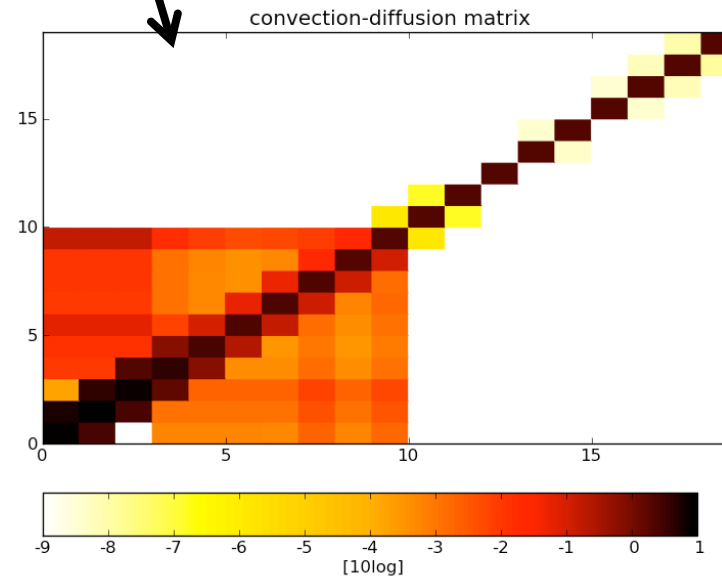
convective mixing



2. Serial optim: vertical mixing

- convection/diffusion solved with linear system:

$$C \underline{x}[t+1] = \underline{x}[t]$$

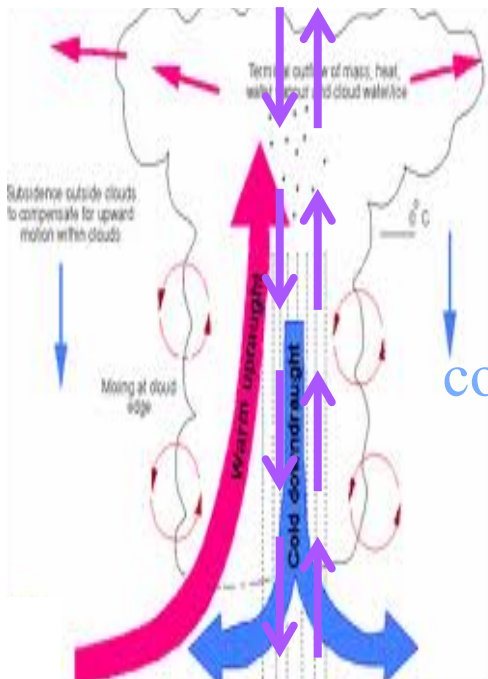


“lmax_conv”
layers
(19 of 25)

Joint Research Centre

vertical
diffusion

convective
mixing





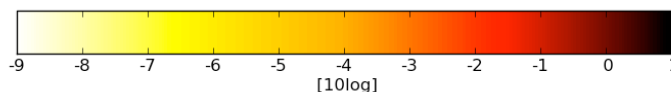
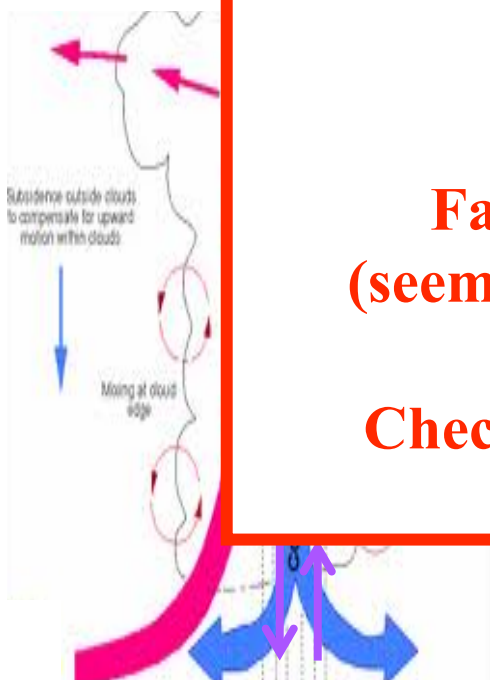
2. Serial optim: vertical mixing

WARNING
Some 4D-var codes around that
use calls to single precision lapack routines:

call sgetrf(...)

Faster, but recipe for segmentation faults!
(seems no problem on 64-bit machines however)

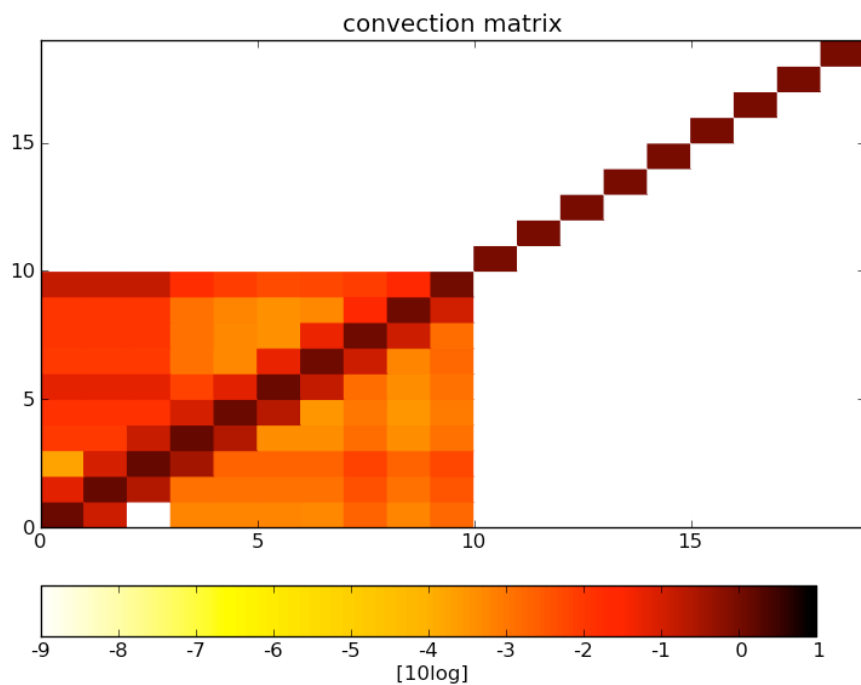
Check if you use the double precision versions!



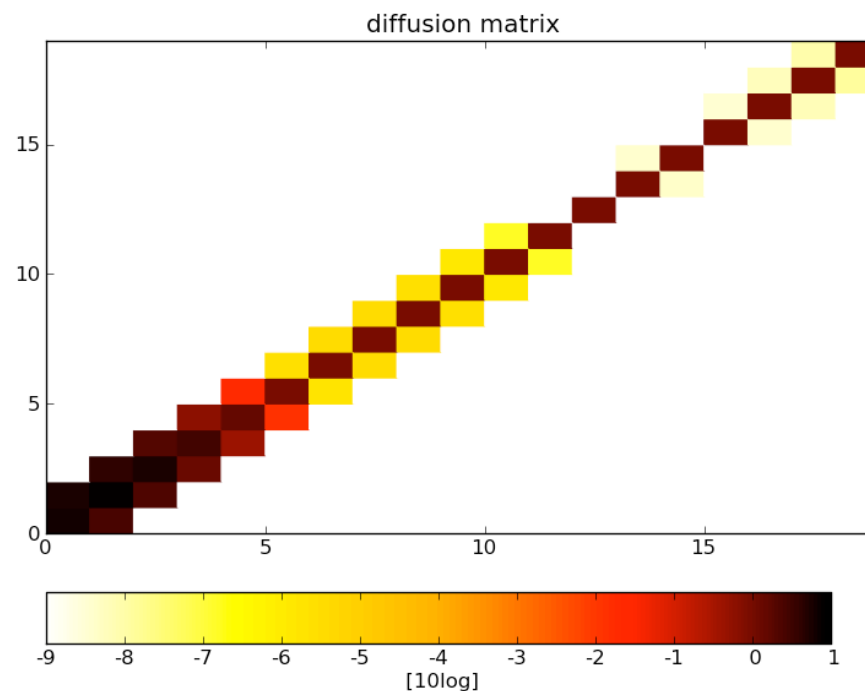
..._conv”
layers
(19 of 25)



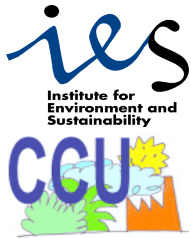
- Split matrix in original parts for convection and diffusion:



full matrix + identity

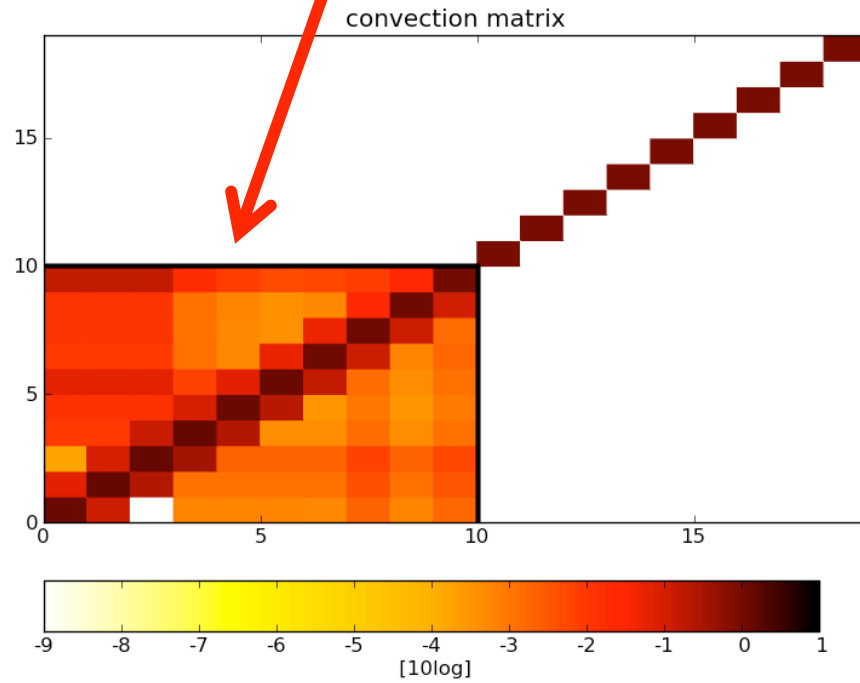


tri-diagonal matrix

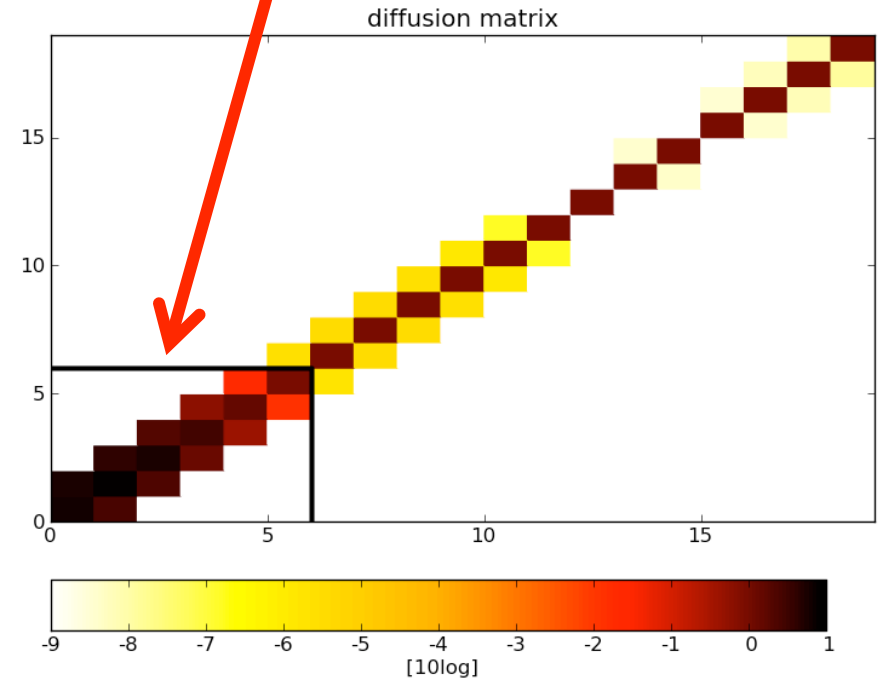


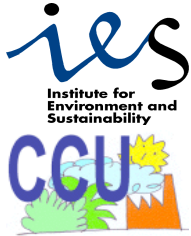
- Solve only up to layer where:

convection is present

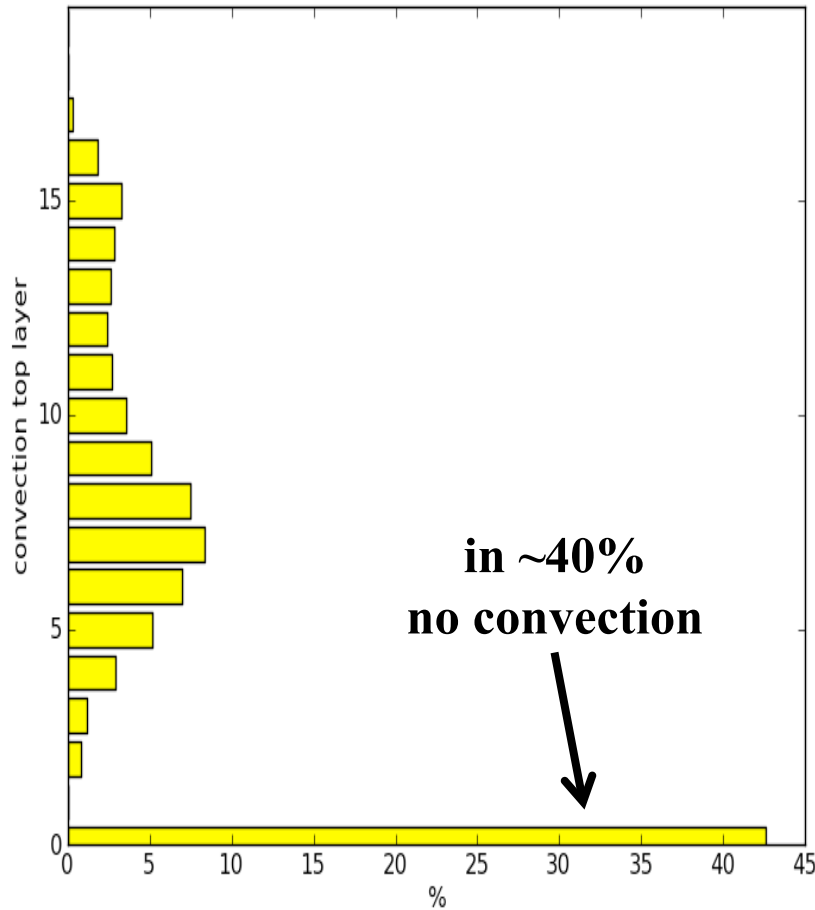


diagonal elements $> \epsilon_{ps}$

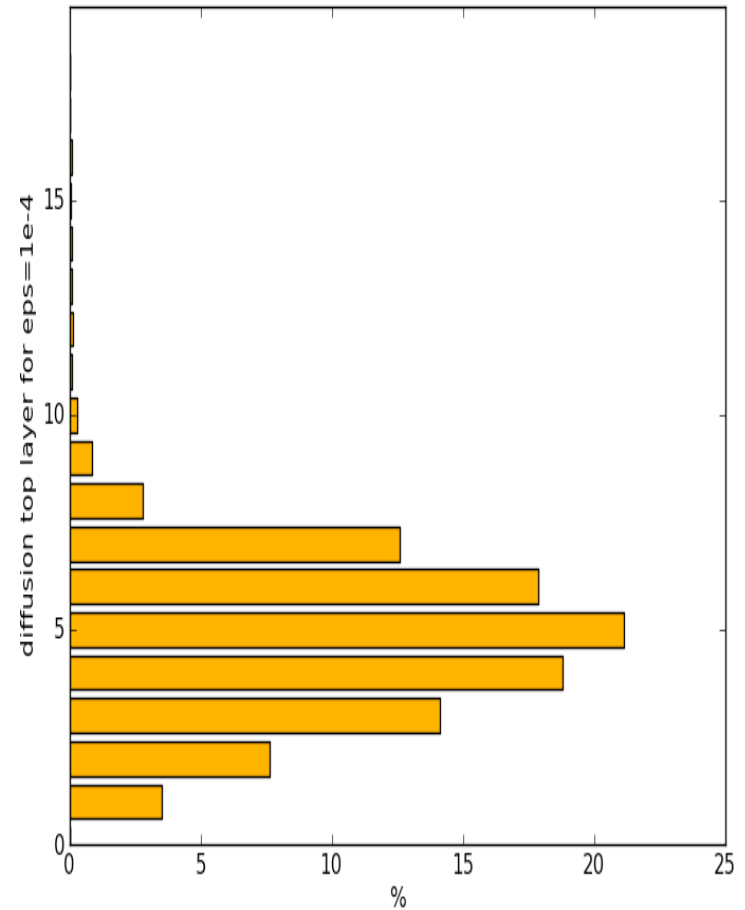




convection top layer

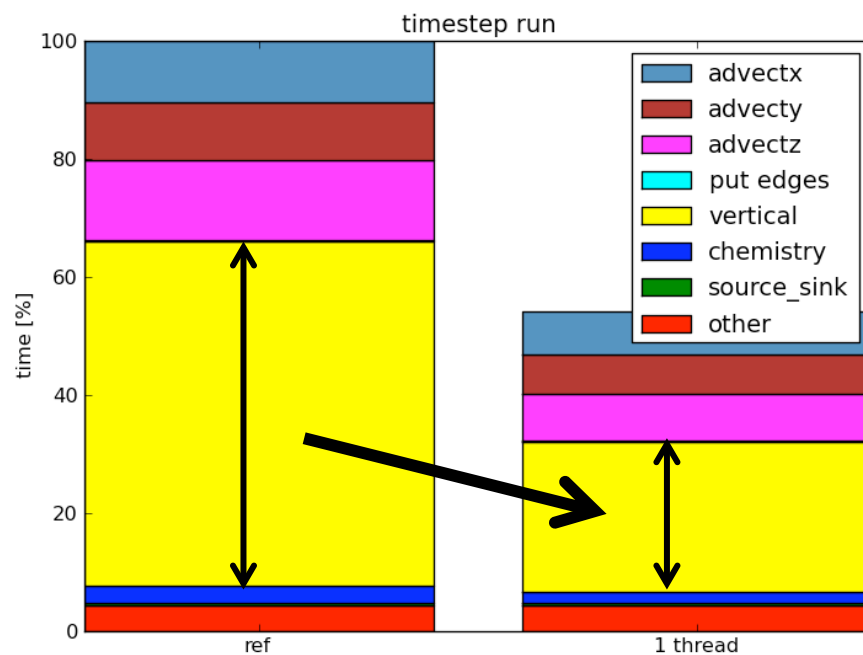


diffusion top layer for $\epsilon_{ps} < 1e-4$



2. Serial optim: convection+diffusion

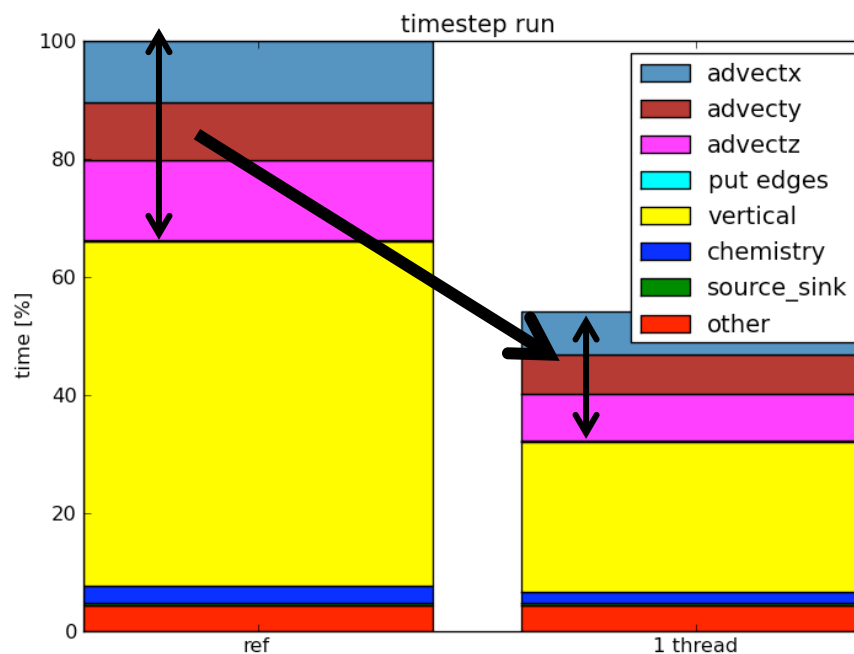
- 50% less time spent on “vertical”





2. Serial optim: advection

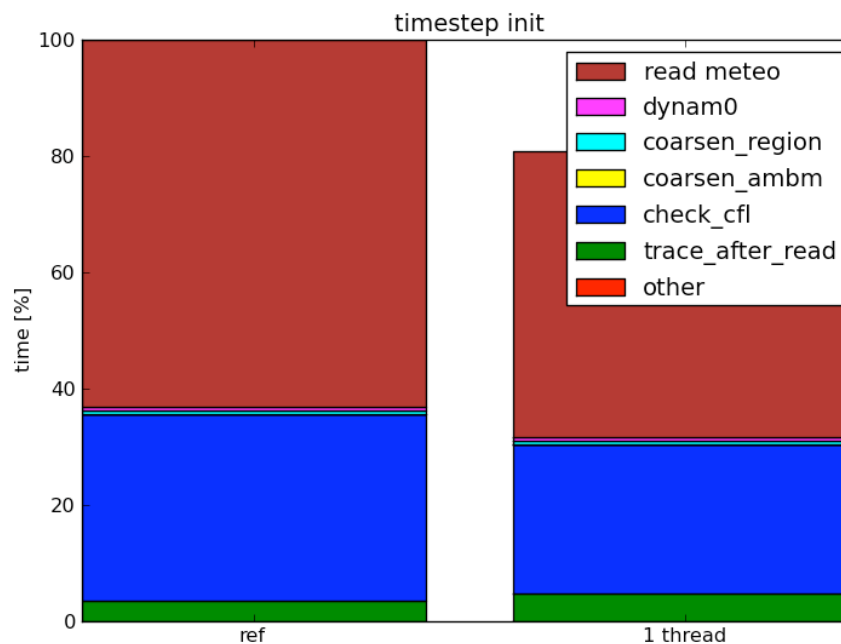
- skip second cfl check
- changed some loop orders etc
- 40% less time spent on advection





2. Serial optim: timestep initialization

- cleaned-up cfl check (-20% runtime)
- new meteo format (-20%)



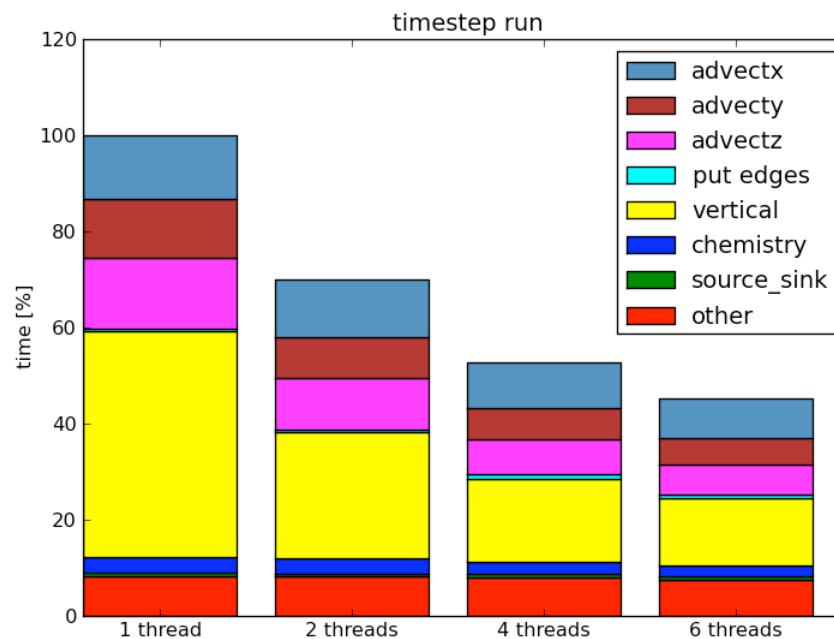
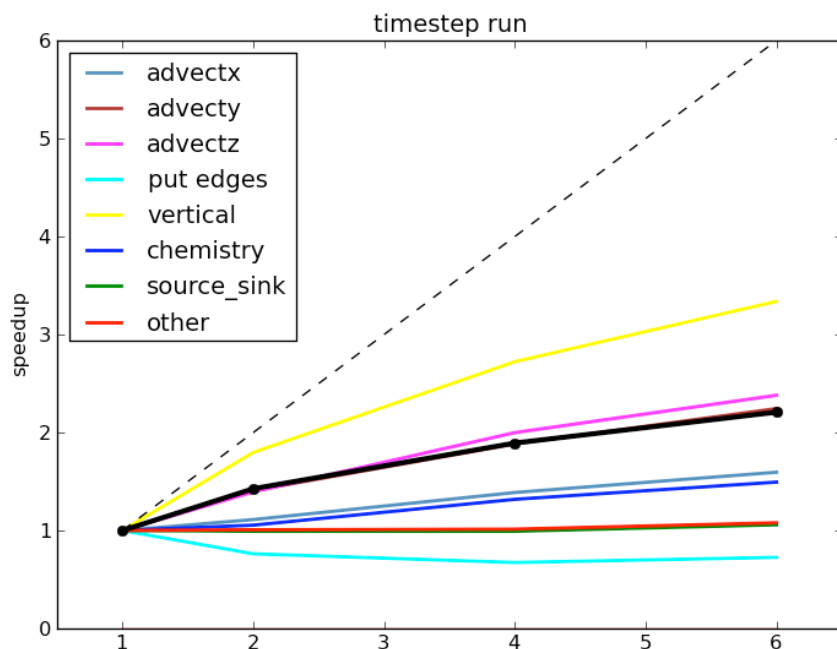


3. OpenMP

- TM5 MPI parallelization (tracers/levels) not useful here:
 - only one tracer: CH₄, N₂O
 - hardly chemistry (only OH-sink)
- Use OpenMP directives:
 - cleaner code
 - TM5 release3 directives available for inspiration



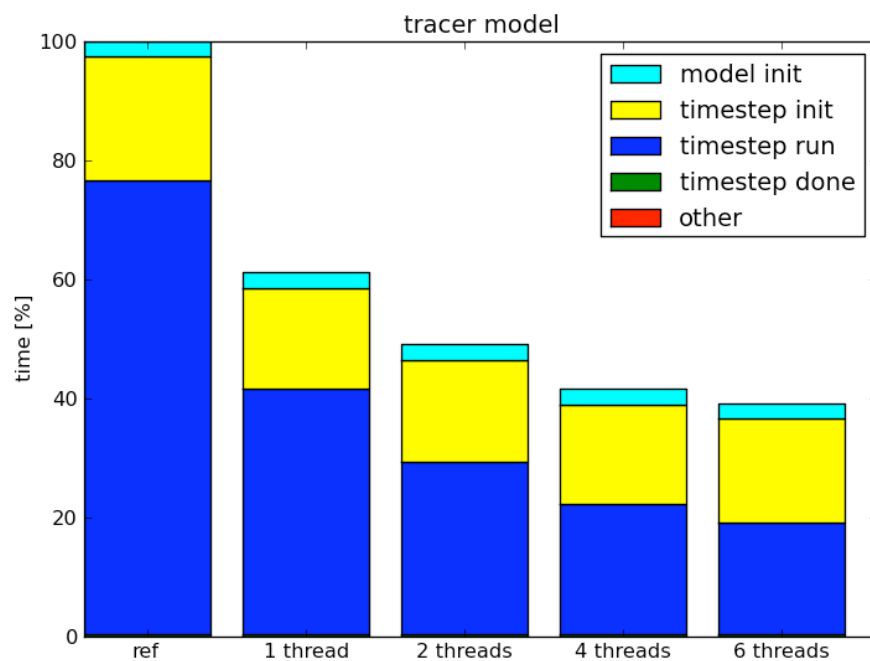
- added to: *advection, convection, diffusion, 'chemistry' (OH-sink)*
- also: *cfl check, mix edges*





Total

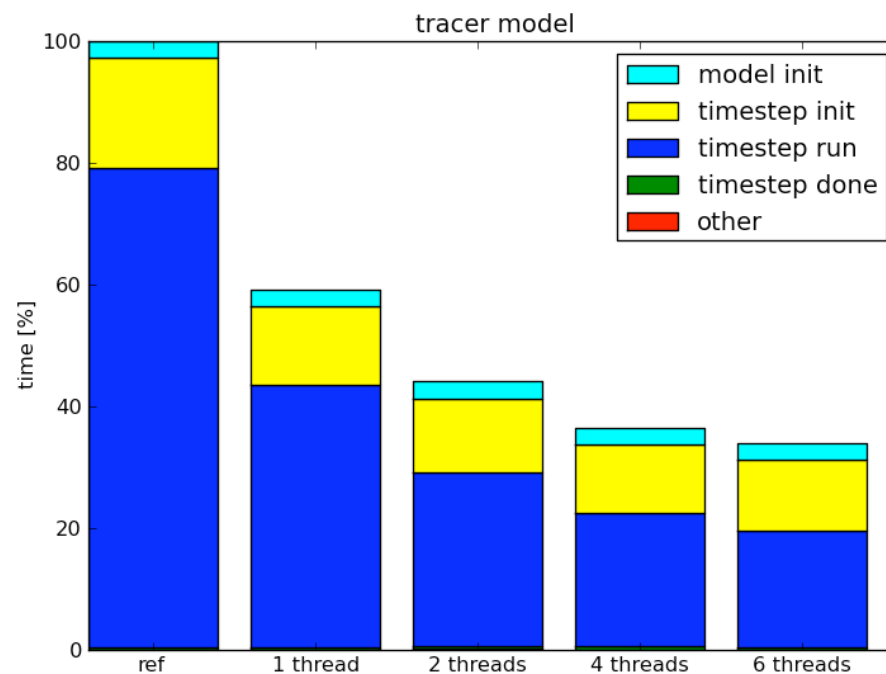
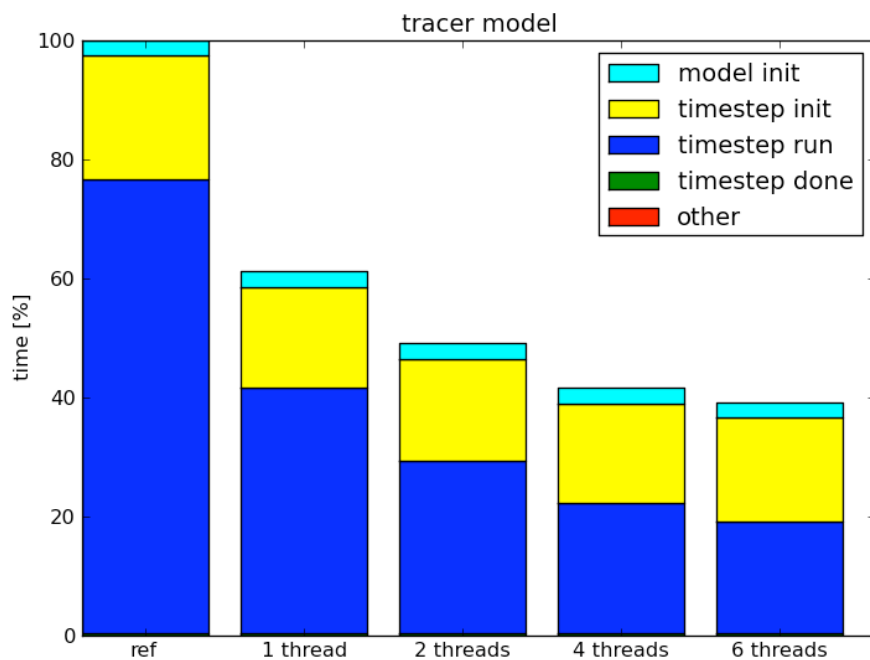
- serial -40%, openmp -20%



Total

- serial -40%, openmp -20%

idem for run
on global 3x2





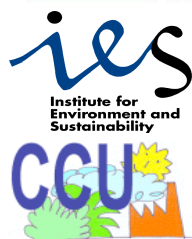
Summary

- Serial optim -40%, openmp -20%
- Input remains problem
- Future:
 - use mpi with model node(s) and io node ?
 - domain decomposition ?



Update on meteo format

From HDF to NetCDF ?



TM5 meteo files

from previous meeting

- **current format:**
 - hdf
 - internal compression
 - saves 25% of disk space
 - useful when we only used workstations with small disks
 - multiple 3D records in a file
 - 8-10 years old ?
- **testing a new format:**
 - NetCDF4
 - = based on HDF5
 - ! bug in HDF5 for IBM AIX machines (ecmwf ...);
therefore tests with 'classic' NetCDF
 - 4D records
 - uncompressed
 - allows parallel i/o



TM5 meteo files

from previous meeting

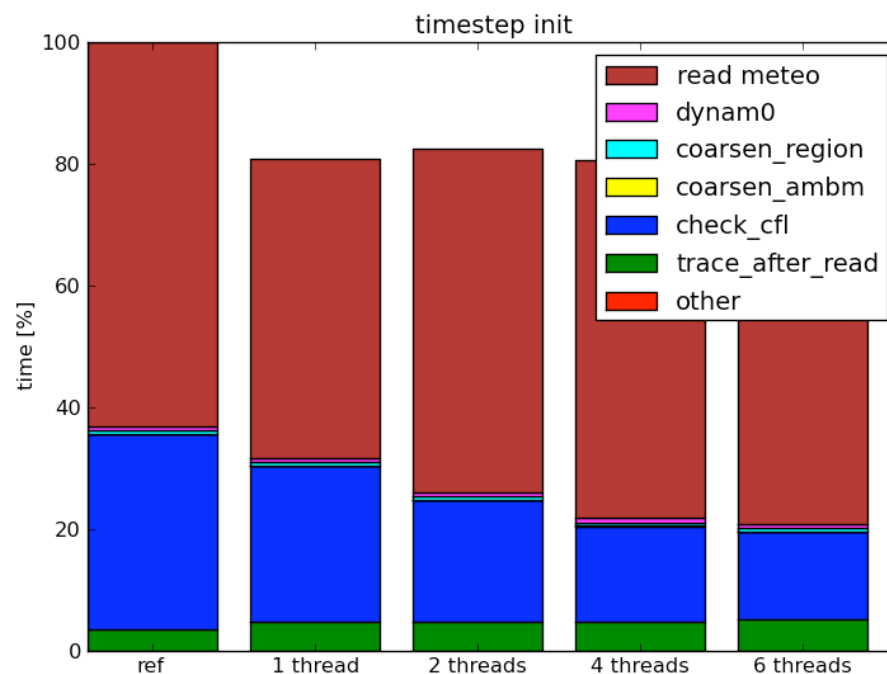
- **current format:**
 - hdf
 - internal compression
 - saves 25% of disk space
 - useful when we only used workstations with small disks
 - multiple 3D records in a file
 - 8-10 years old ?

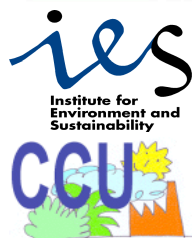
- **testing a new format:**
 - NetCDF4
 - = based on HDF5
 - ! bug in HDF5 for IBM AIX machines (not hdf ...); therefore tests with 'old' NetCDF
 - 4D records
 - uncompressed
 - allows parallel i/o

solved!

- Compared runs with old and new meteo:
 - tiny difference in land/sea mask (% is not a CF unit); round to integer values 0-100 ...
 - run output exactly the same

- run time for reading meteo: -20%
 - ... but slightly increasing when using more threads (memory?)
 - similar for input from glb3x2 and glb1x1
- ! overall time strongly depended on file system !**





- **Proposal: commit branch into trunk:**
 - backwards compatible
 - cleaner
 - ready for use in future (far, far away ...)
- **Proposal: start converting archives:**
 - era-interim
- ... at the same time : clean up ALL archives !