Challenges for using PREP and PGD operationally

Tayfun Dalkılıç, Daan Degrauwe, Piet Termonia

Contents

- PREP
  - memory consumption
  - cpu time

- PGD

Technical information

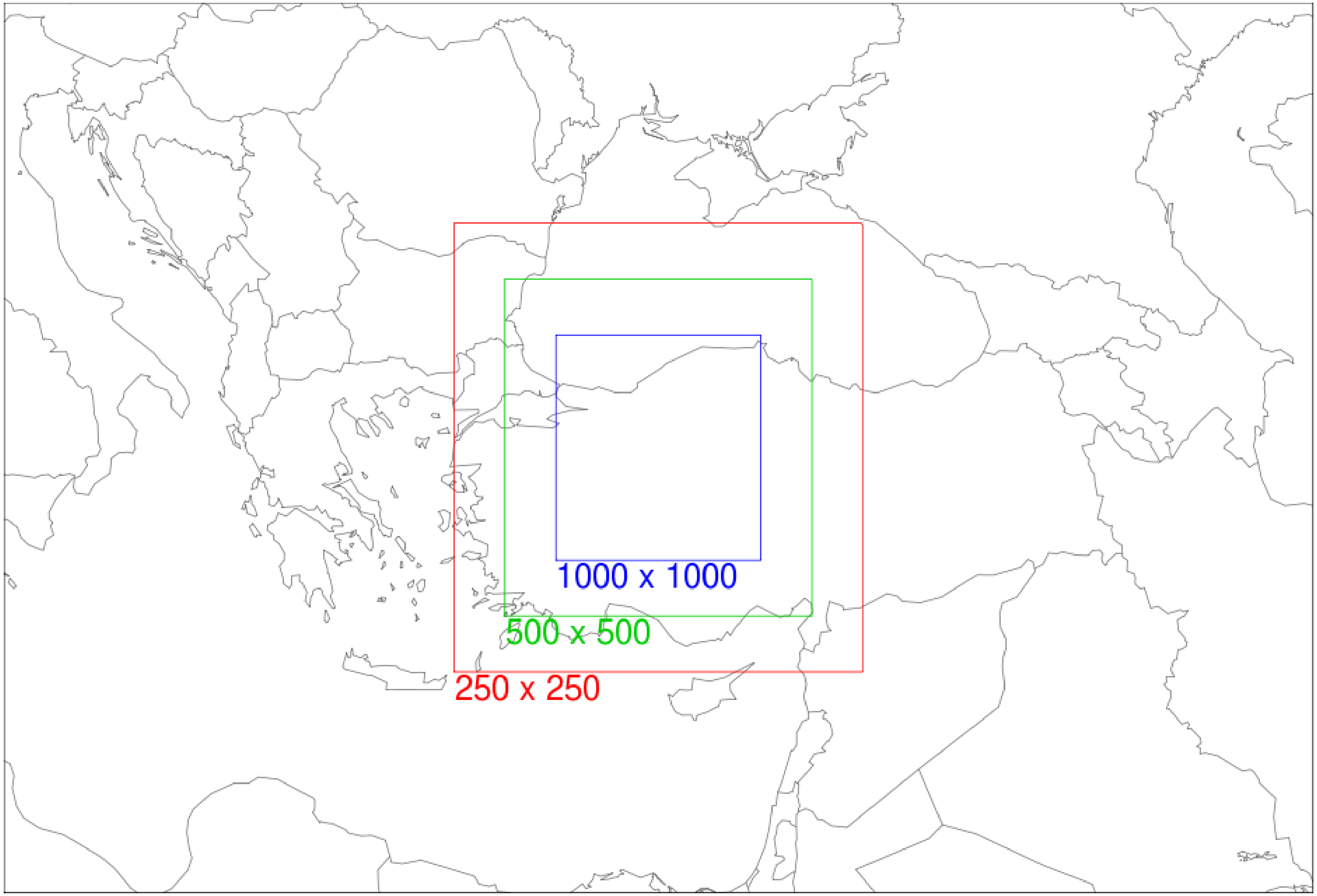
- cy39_t1.03, including Surfex 7.2

- SGI-UV2000: 256-core 2.4 Ghz **1TB shared memory** machine

PREP: CPU time

- Using PREP to interpolate from operational Turkish domain (709 x 439) to domains of different dimensions

  - Oper@4.5km
  - 250@4km
  - 500@1.5km
  - 1000@0.5km
  - 1500@0.5km



*** 1500 domain coincides with 500 domain

PREP: CPU time

- Profiling with Dr Hook for 1500 x 1500 domain:

```
Profiling information for program='./PREP', proc#1:
   Memory usage : 27314 MBytes (heap), 27311 MBytes (rss)
   Wall-time is 1790.10 sec on proc#1 (1 procs, 1 threads)

  #    %         Time (s)   # of calls     Routine@<thread-id>
  1    52.91    947.087             19     MODI_AV_PGD:AV_PATCH_PGD_1D:PART2@1
  2    38.59    690.852            924     BILIN@1
  3     3.41     61.109             63     MODI_AV_PGD:AV_PGD_1D@1
 ...
```

- So 3 routines take about 95% of CPU time!

PREP: CPU time
source code optimizations (1)

- `AV_PGD_1D` and `AV_PATCH_PGD_1D`: loop over gridpoints inside loop over all covers. The number of covers is 573, i.e. all covers that exist globally. But inside a LAM domain, only a limited number of covers will actually be present (e.g. 30 over Belgium, 100 over Turkey)

→ **Solution**: we added an extra argument to these subroutines to indicate which covers are present, and only perform the loop over these covers.

This also reduces memory consumption.

PREP: CPU time source code optimizations (2)

- Number of calls to `BILIN` seems quite large (924)
- A majority of these calls is related to different vegetation types (12).
- However, in our setup, we only have 1 patch type, meaning that the fields being interpolated are the same for all 12 vegetation types. [Concretely, in the routine `PREP_HOR_ISBA_FIELD`, the array `ZFIELDIN` is dimensioned as (NGP x 20 x 12), but it contains 12 copies of the same NGP x 20 array.]

→ **Solution**: We replace the loop over the vegetation types (12) by a loop over the patch types (1) in `PREP_HOR_ISBA_FIELD`. This reduces the number of calls to `BILIN` from 924 to 242.

*** The results remain exactly the same after these source code optimizations.

PREP: CPU time
Use of OpenMP

- The 3 most expensive routines contain easy-to-parallelize loops over the gridpoints.

- Introducing OpenMP directives allows to run PREP on multiprocessor (shared-memory) machines.

PREP: CPU time

- Results: CPU time (s)

Target Geometry	ORIG	OPT	OPT + OpenMP
250 x 250	98	47	25
500 x 500	219	64	32
1000 x 1000	909	148	56
1500 x 1500	1790	323	106

ORIG : Original PREP
OPT : Source code optimizations
OpenMP : optimal number of threads between 1 and 32

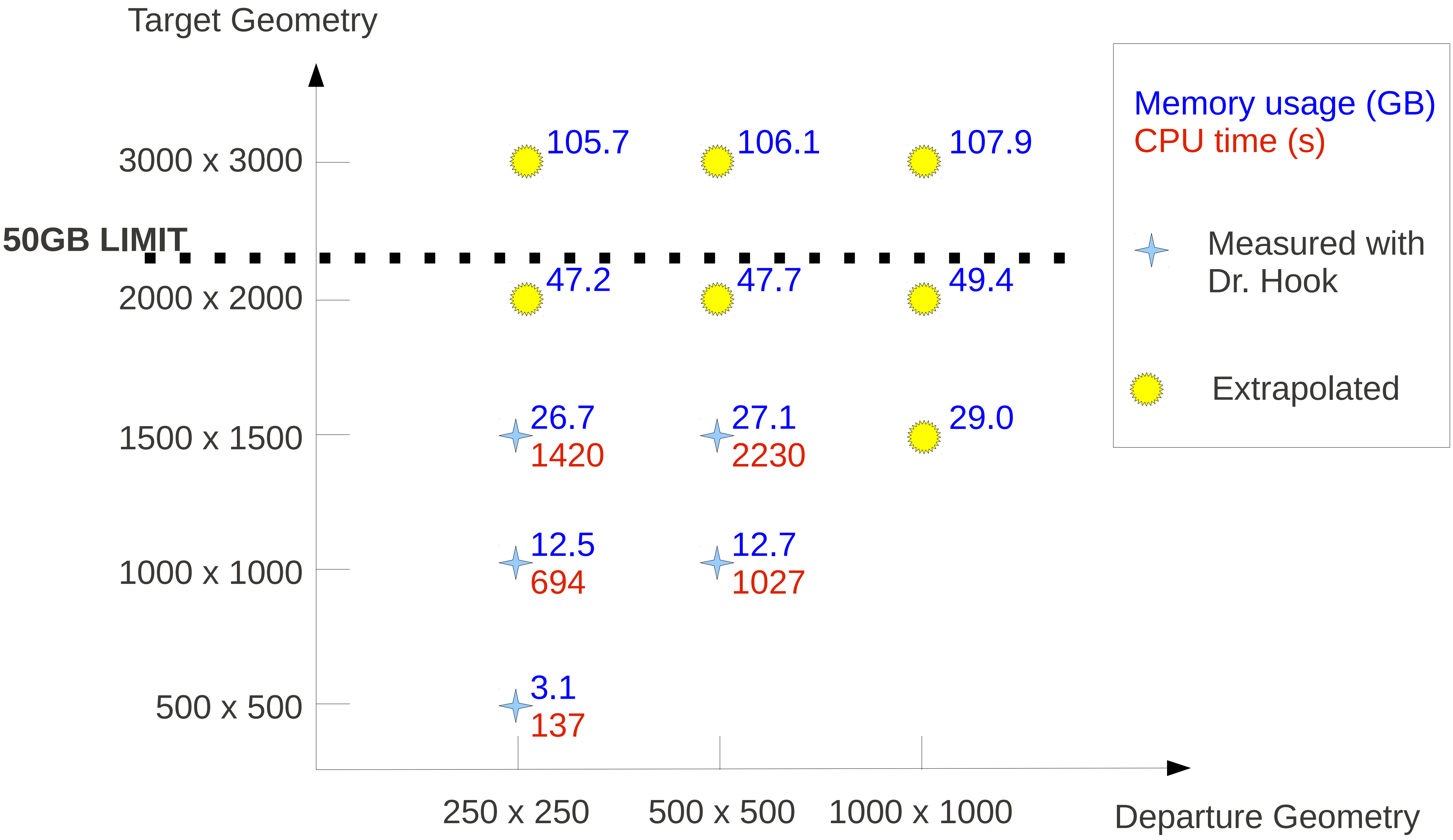
PREP: memory consumption

- We consider the "*Scientific Upper Bound*", i.e. the most demanding scenario from the scientific point of view: all 573 covers present; 12 patches.

- We determine the relationship between number of gridpoints (target and departure) and memory usage.

- We consider 50GB shared memory as the reference (ECMWF machine)

PREP: memory consumption



PREP: memory consumption

- Remark: we discovered large memory leaks in PREP due to the use of pointers, which can be reallocated without deallocating.

Remarks on PGD

We experienced problems with creating PGD files for large domains (e.g. 2000 x 2000)

- *CPU time*: routine `INTERPOL_NPTS` seems to contain a double loop over the gridpoints!

- *Memory consumption*: ~3 times as much as for PREP, so this may become problematic in the near future!

*** But we didn't have a look at possible optimizations yet ...