Single precision in cycle 43

Ole Vignes, ALADIN/HIRLAM all staff meeting 2019

Overview

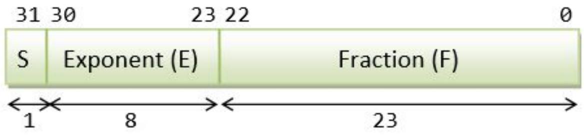
- What does single precision mean?
- Why is it attractive?
- How to activate it?
- Adaptations for ODB and data assimilation
- Results from validation against double precision experiments
- What remains before operationalization?
- Conclusions and outlook

What does single precision mean?

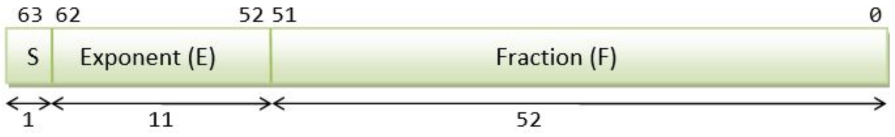
Single precision usually means (also in IA^{4}H) that floating point numbers are using 32 bits of storage instead of the traditional 64 bits.

With 32 bits a (IEEE 754) (non-zero) floating point number lies in the range $\mp\mathbf{1.17 \times 10^{-38}}$ to $\mp\mathbf{3.4 \times 10^{38}}$. Precision is around **7** digital digits.

With 64 bits those limits are $\mathbf{\sim 2.2 \times 10^{-308}}$ and $\mathbf{\sim 1.8 \times 10^{308}}$.
Precision is around **15** digital digits.



32-bit Single-Precision Floating-point Number



64-bit Double-Precision Floating-point Number

Why is single precision attractive?

The forecast model typically sees a run-time reduction of **35-40%**.

This is not because the computer necessarily computes faster in 32 bits, but mostly because the data volumes become smaller:

- The memory cache layers can hold larger portions of arrays (memory bandwidth becomes better)
- Amount of data involved in MPI message passing smaller (halved)

Note that file sizes (FA, grib) do not shrink the same way, since they are already packed to (much) less than 32 bit precision.

How to activate it?

Something like this (from Harmonie “makeup” config file for Intel):

```
FDEFS=... -DADDRESS64 -DPOINTER_64 -D_ABI64 …    (common flags)
ifeq ($(FP_PRECISION),single)
      FDEFS += -DPARKIND1_SINGLE -DB2O_HAVE_IFSAUX
      AUTODBL=
else
      FDEFS += -DREAL_8 -DREAL_BIGGER_THAN_INTEGER
      AUTODBL=-r8
endif
```

In arpifs/module/parkind1.F90:

```
#ifdef PARKIND1_SINGLE
INTEGER, PARAMETER :: JPRB = SELECTED_REAL_KIND(6,37)
#else
INTEGER, PARAMETER :: JPRB = SELECTED_REAL_KIND(13,300)
#endif
! Double real for C code and special places requiring higher precision.
INTEGER, PARAMETER :: JPRD = SELECTED_REAL_KIND(13,300)
```

Code adaptations

The forecast model was almost ready thanks to work by ECMWF (F. Vana *et. al.*) and Meteo-France (P. Marginaud). Some surfex adaptations since HIRLAM has introduced v. 8.1 in cy43h2.

Most of my adaptations were related to ODB/assimilation.

Around 390 files were modified, but many changes small and trivial, typically replace JPRB with JPRD in declarations. Some cases of constants out of bounds for 32 bit (outside 1E-38 to 1E+38).

ODB

In cycle 43 the C part of ODB is hardcoded to use *double* (64 bit) for data.
As a consequence ROBHDR/ROBODY in Fortran must be of kind JPRD.

It should have been possible to change the C code to allow float (via typedef), but another issue is the missing data indicator(s). They are typically set like this:

```
NMDI = 2147483647
RMDI = -2147483647_JPRB
```

or sometimes `RMDI = -NMDI.` Note that RMDI cannot be that precise with 32 bit reals. Since I was unsure about this truncation I decided to leave ODB in 64 bit. That lead to many changes under *obs_preproc, op_obs, bufr2odb, cma2odb.*

Inner products (important in assimilation)

It is a well-known fact from numerical linear algebra that the accumulation of inner products is susceptible to round-off error when vectors are long. Therefore, in e.g. `CONTROL_VECTORS_PARA_MOD` I use JPRD for the accumulation variable:

```
REAL(KIND=JPRB) FUNCTION DOT_PRODUCT_CV_CV(YDCV1,YDCV2)
TYPE (CONTROL_VECTOR), INTENT(IN) :: YDCV1,YDCV2
REAL(KIND=JPRD) :: ZZ
...
ZZ=0.0_JPRD
DO JJ=1,ILEN
  ZZ=ZZ+YDCV1%DATA(JJ)*YDCV2%DATA(JJ)
ENDDO
...
DOT_PRODUCT_CV_CV=REAL(ZZ,JPRB)
```

Results from validation against double precision

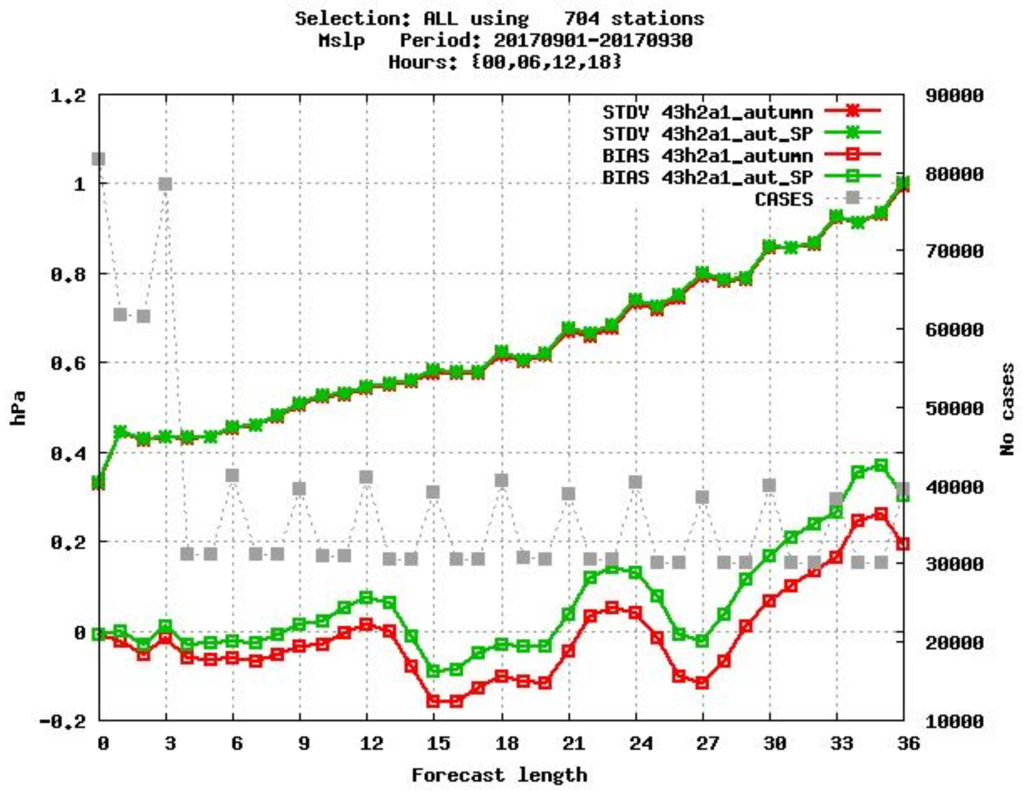
As part of the Harmonie validation of cycle 43, four periods (seasons) were run and compared against cycle 40. For the autumn period (september 2017) I reran the cy43 case with single precision reals.

Only conventional observations were assimilated.

In general, the 32 bit reals run reproduced the 64 bit run quite well. Most variables were almost indistinguishable in scores, except one notable difference:

- MSLP shows an additional positive bias in the single precision run.

Verification of MSLP

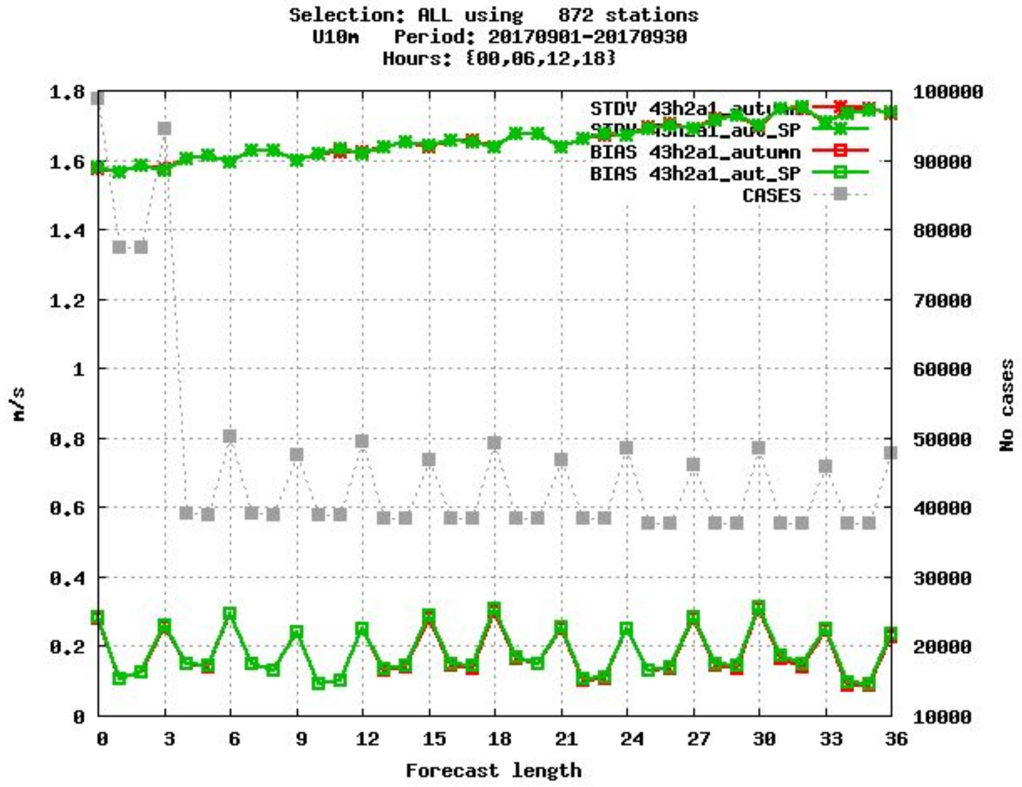


64 bit reals

32 bit reals

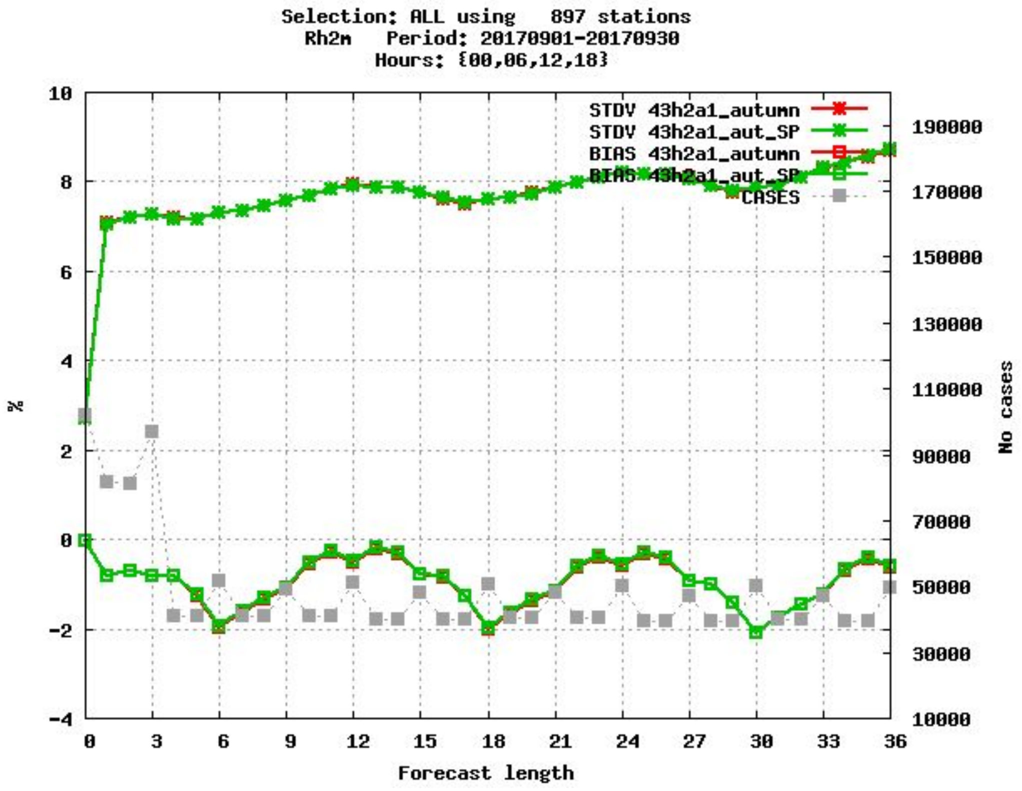
We see a clear additional positive bias increasing with forecast length

Wind at 10m



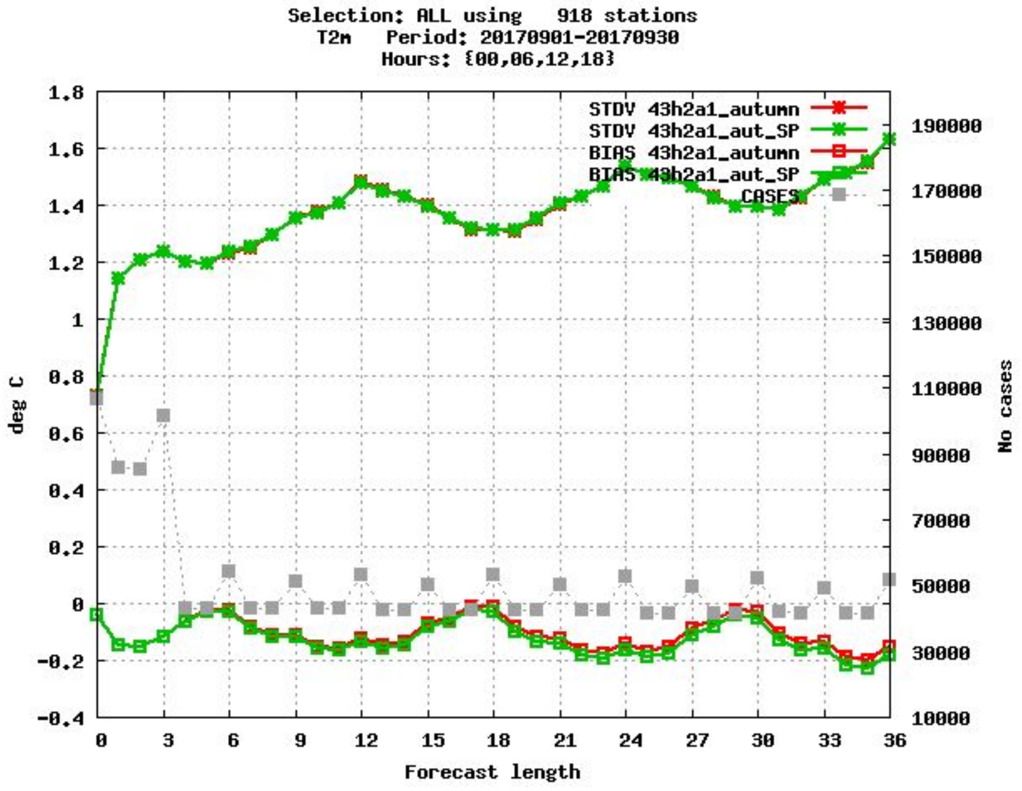
Looks completely fine

Relative humidity (at 2m)



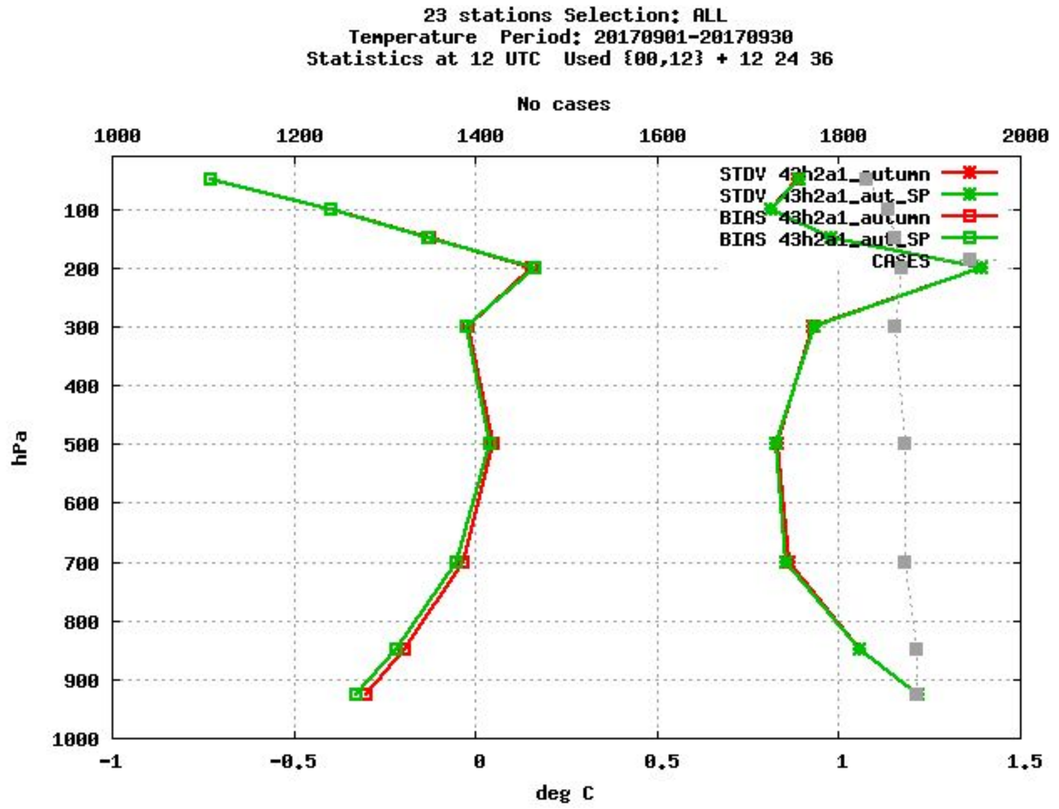
Looks fine too.
So do clouds
and precipitation
(not shown).

Temperature (at 2m)



A small signal (cooling) in temperature at 2m can be seen, also increasing with forecast length

Temperature profile



Can see a slight cooling throughout the troposphere, increasing with pressure.

Lead us to suspect some issue with radiation.

A first attempt at improving the MSLP bias

F. Vana *et. al.* (MWR, Feb. 2017) identified three areas in IFS where improvement was needed:

1. Radiation scheme
2. Legendre transformations (not used in LAM)
3. VFE setup

In Harmonie we have not (yet) used vertical finite elements, so the double precision run was without. It would be interesting to test VFE in the future.

Also, although AROME uses ECMWF radiation, it is an older version. I think radiation experts must be involved to look at those precision issues in AROME.

What remains before operationalization?

Not all types of observations are tested yet, or are known to work.

- AMSU-A, AMSU-B, MHS apparently works
- ASCAT apparently works
- IASI crashes (in radiation computations ...)
- Radar not tested
- GNSS not tested

Conclusions and outlook

- Harmonie cy43h2 (alpha) has been modified to enable runs with 32 bit reals
- Both forecast model and assimilation (Canari, Screening, 3D-VAR) run (mostly) in single precision
- Some work remains for remote sensing data (IASI, Radar, GNSS, …)
- Should understand and improve the temperature cooling and resulting positive MSLP bias

Next: would like to also test LAM 4D-VAR …

Also: unsure about phasing to later cycles since work has also been done upstream in cy45/46 in this area. Hope we can learn from each other.

Gracias por su atención!