Software evolution around Arpège, Aladin, Arome

Claude Fischer

Content:

- Arpège/IFS collaboration
- Aladin collaboration
- Software interaction Arpège/Aladin
  - General overview and motivation
  - Rules
- Phasing
- Code management
- Implications on Arome
- Appendices:
  - Configuration overview; List of libraries; Some figures on code performance; Technical stuff

Arpège/IFS collaboration

- Technological collaboration: a new common cycle every 6 months
- Regular phone conferences and semestrial coordination meetings
- Many informal bi-lateral contacts
- Software agreement about code exchange, defining protected areas for each side

Arpège/IFS collaboration (2)

- Start of collaboration: 1987/1988
- Main dates for operations at EC:
  - IFS in March 1994 (T213/L31)
  - Daily EPS in May 1994
  - 3D-VAR in January 1996
  - MPI version in September 1996 (VPP700)
  - 4D-VAR in November 1997
  - VPP5000 in May 2000
  - SL TL/AD in inner loop in November 2000
  - IBM cluster in March 2003

Arpège/IFS collaboration (3)

- Main dates for operations at MF:
  - Arpège in September 1992 (T79/L15/c1)
  - Stretching in Arpège (c3.5) in October 1993
  - SL advection scheme in Arpège in October 1995
  - 3D-VAR in May 1997
  - MPI version in June 1998 (VPP700)
  - 4D-VAR in June 2000
  - Raw radiances in October 2002
  - Reduction of stretching (TL358/L41/c2.4) in June 2003

Aladin collaboration

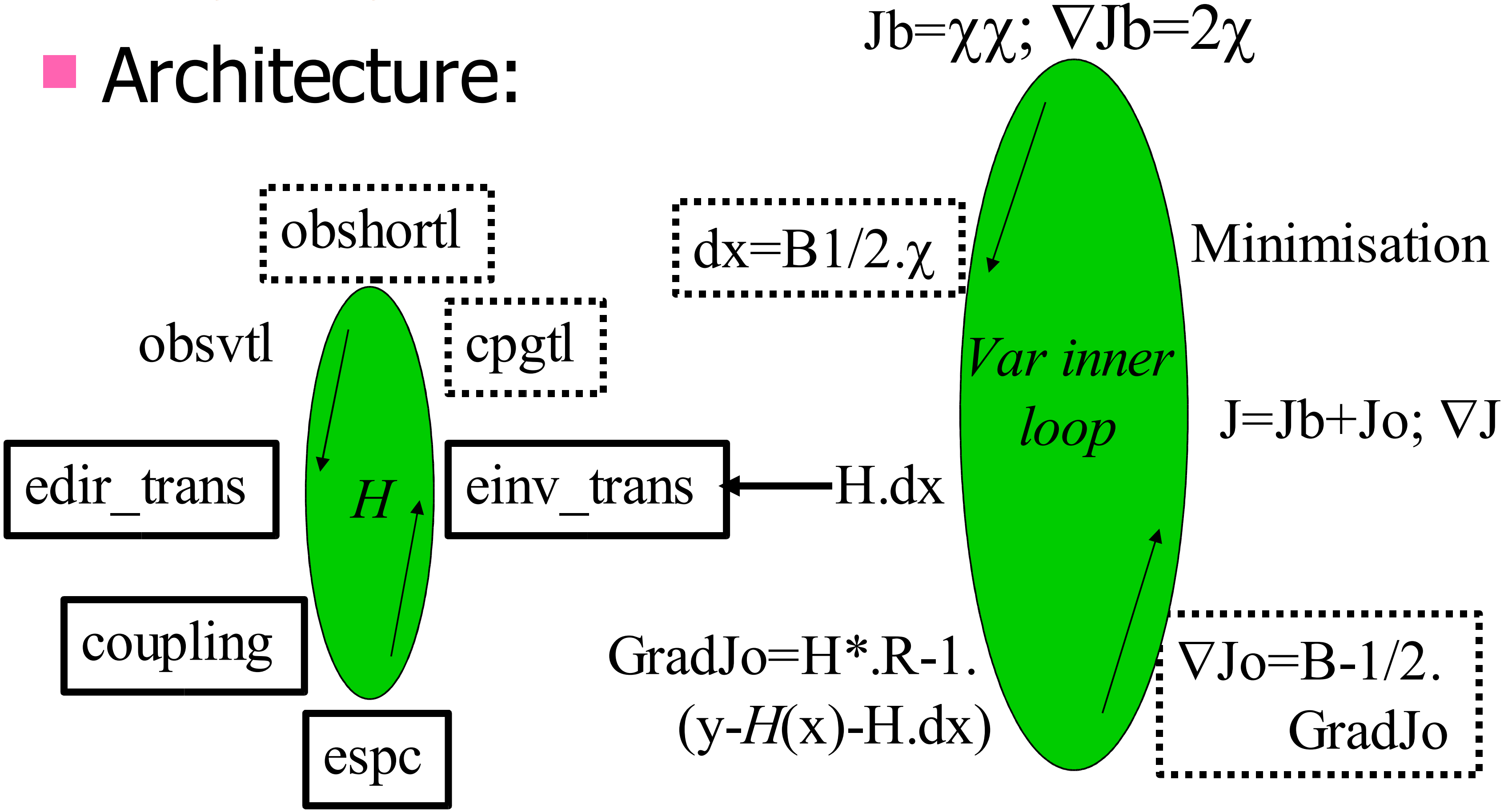
- Start: first visitors in May 1991
- Aladin/PECO in May 1994
- Aladin/France in March 1996
- Aladin/LACE in Toulouse in July 1996
- Aladin/France MPI version in June 1998
- LACE model moves to Prague computer in July 1998
- Aladin/France on VPP5000/60PE in October 2003

Software interaction Arpège/Aladin

- General view & motivation:
  - IFS evolution forces a continuous, sometimes drastic, code optimisation and cleaning
  - IFS provides powerful state-of-the-art software, both scientific and technological
  - In detail, for the LAM: observation operators are IFS, basic configurations, singular vectors etc…
  - Sometimes, first ideas are tested in the LAM: SL scheme, digital filters
- **This « marriage » creates rights and duties**

Software interaction Arpège/Aladin

- Architecture:



Jb=$\chi\chi$; ∇Jb=2χ

obshortl

dx=B1/2.χ

Minimisation

obsvtl

cpgtl

Var inner loop

J=Jb+Jo; ∇J

edir_trans

H

einv_trans

H.dx

coupling

GradJo=H*.R-1.(y-*H*(x)-H.dx)

∇Jo=B-1/2. GradJo

espc

Software interaction Arpège/Aladin

- Architecture:

obshortl	Slightly different code: LELAM key
coupling	Completely different code

Fully shared dataflow between IFS and Aladin (especially in gridpoint space), but quite separate dimensioning and addressing in spectral buffers (spherical versus bi-Fourier).
Coupling code is of course only LAM.

Software interaction Arpège/Aladin

- Rules or practice:
  - IF (LELAM) THEN; CALL ETOTO; ELSE; CALL TOTO; ENDIF
  - Aladin specific routines go into a separate fortran library, unseen from ECMWF
  - No LELAM key below the level of gridpoint scan (SCAN2MDM/TL/AD) => LRPLANE
  - Duplicated code must be avoided !!
  - The issue(s) for modularity: with respect to a functionality or a desired degree of freedom, not with respect to IFS v/s LAM

Software interaction Arpège/Aladin

- Calling tree (ex: 3D-VAR) is common at control level:

```
CNT0 -> SU0YOMA / SU0YOMB
        -> CVA1 -> SUOBS / CNT2 (trajectory)
                   -> SIM4D (simulator)
                      -> CHAVARIN
                      -> CNT3TL / CNT3AD
                      -> CHAVARINAD
                      -> M1QN3
```

Phasing (1)

- When and how long ?:
  - Twice per year (generally spring and fall)
  - « 6 weeks », but in practice 2-3 months
- How many ?:
  - In the old days: between 4 and 6/7 phasers invited, plus 2/4 Météo-France local stuff
  - Nowadays: 4/5 initial phasers, plus 2/3 additional « late » phasers
  - Phasing has become more time continuous, because of:
    - **Human turnover and need to train new phasers**
    - **More configurations with time**
    - **Increased complexity: observation database, NH**
- Phasing is NOT VERY POPULAR, and depends heavily on the people's willingness to leave home … for a sacrifice

Phasing (2)

- Principles:
  - At the very beginning: check and understand the modifications in a new IFS/Arpège cycle
  - Report (manually) Arpège changes in the Aladin counterparts when automatic
  - Perform code and scientific analysis, and then adapt to Aladin setup and/or LELAM keyed code when needed (= when not trivial)
  - Recode some Aladin at identical scientific content for code compliancy with Arpège
  - Report systematically changes in duplicated code !
  - Analyze and « uncode » code clashes

Phasing (3)

- Phasing examples:

```
SUBROUTINE TOTO
WEIGHT=1.+2**N <<inserted>>
Y=X/(2.*N+1.) <<changed into>>
Y=WEIGHT*X/(2.*N+1.)
```

```
SUBROUTINE ETOTO
WEIGHT=2**N * 2**M <<inserted>>
Y=X/(2.*FKSTAR(N,M)+1.) <<changed into>>
Y=WEIGHT*X/(2.*FKSTAR(N,M)+1.)
```

Phasing (4)

- Phasing examples:

```
SUBROUTINE TOTO(ZTAB,K)  <<changed into>>
SUBROUTINE TOTO(ZTAB,K1,K2)
REAL ZTAB(K) <<changed into>>
REAL ZTAB(K1,K2)
CALL TITI(ZTAB)  <<unchanged>>
          SUBROUTINE ETOTO(ZTAB,K)  <<changed into>>
          SUBROUTINE ETOTO(ZTAB,K1,K2)
          REAL ZTAB(K) <<changed into>>
          REAL ZTAB(K1,K2)
          CALL TITI(ZTAB)
```

Code management

- Source management under clearcase
- To Arpège CY28T1 corresponds the Aladin cycle AL28T1
- IFS has its own cycles in Reading (CY28R1, etc…)
- Support team for maintenance of cycles and libraries, and interfacing with operations: « GCO »
- More and more progress is made to build a user-friendly compilation environment on the high performance platforms (« gmkpack »)
- Export versions are defined and made available for the Aladin partners (generally based on Toulouse operational versions)

Implications on Arome

- Already a slightly renovated strategy of phasing for the present Aladin:
  - Go from a very concentrated period in time and one team of phasers towards more time-continuous phasing: 4-5 initial phasers, followed by 2-3 late phasers
  - Try to validate the basic configurations of Arpège and Aladin already in the common Arp/IFS cycle: phase Arpège and Aladin with the IFS at the same time

Implications on Arome (2)

- Separate **with observations/without observations** (assimilation configurations usually come after model ones)
- Separate **adiabatic/process-relying**:
  - adiabatic configurations could be adiabatic hydrostatic and NH models, a basic 3D-VAR, fullpos -> systematically phased and validated
  - Process-relying concerns Alaro or Arome physics, complicated Jb's or control variable, exotic TL/AD models -> not fully validated, not systematically phased, remote debugging at partner Centers
- Méso-NH physics routines are in a separate library, with specific interface code (especially to host Méso-NH modules and to catch up real/integer promotion conflicts

Final remarks

- … never forget the « big brother » (ECMWF)
- Documentation:
  - Aladin Tech'book about Aladin impacts in Arpège
  - GMAP and Aladin websites
  - Karim Yessad's extensive Arpège documentation
  - NH documentation by Pierre Bénard
  - 3D-VAR documentation by Claude Fischer
  - Several technical notes: internal technical notes, Alain Joly, ECMWF memorandums, Ryad's coding standards, etc…

Appendix A: overview of configurations

- 927: production of coupling fields
- 001: forecast (both hydrostatic and NH)
- 002: data screening and trajectory
- 701: OI data analysis (CANARI)
- 131: variational analysis (3D-VAR)
- 401: test of the adjoint model
- 501: test of the tangent linear model
- 601: singular vectors (Lanczos)
- 801: gradient computations w/r to I.C.

Appendix B: list of libraries

- arp: Arpège code
- ald: Aladin specific code
- xrd: auxiliary library
- tfl: IFS spectral transforms
- tal: bi-Fourier spectral transforms
- odb: Observation DataBase structure
- sat/coh/…: specific observation libraries for pre-treatment and observation operators

Appendix C: (personal) figures on performance (VPP5000)

- Forecast: 24 hours, hydrostatic, 9.5 km, 300*300*41: 750 sCPU, 2 Gbytes
- 3D-VAR minimisation: 4 PEs, 25 inner loops: 4*200 sCPU, 4*3 Gbytes

Appendix D: some known technical shortcomings

- Gridpoint computations through the extension zone
- Strong link of biperiodisation and Davies relaxation
- 1D/2D on-the-run model diagnostics bugged
- No shallow water model in Aladin … and never planned !
- 3D-FGAT to be validated and tested
- SL TL/AD for the hydrostatic model to be coded