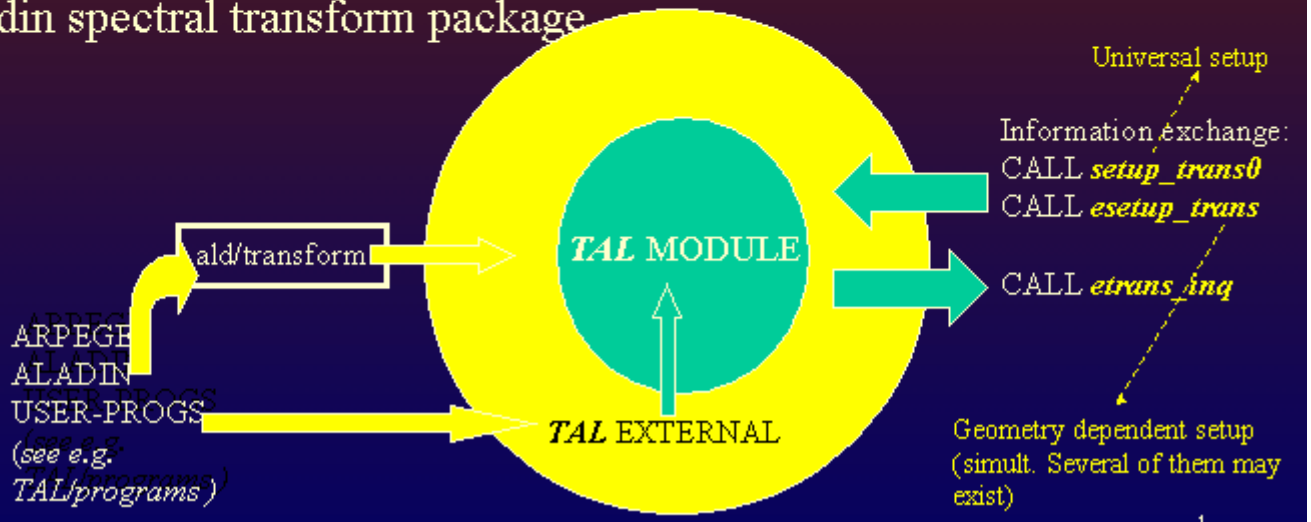
Transformation Package *TAL*

ALADIN homepage:www.cnrm.meteo.fr/aladin/
→Publications→Doc.-s on Arp. Ald.→More doc→technical support
→Aladin spectral transform package



Transformation Package *TAL*

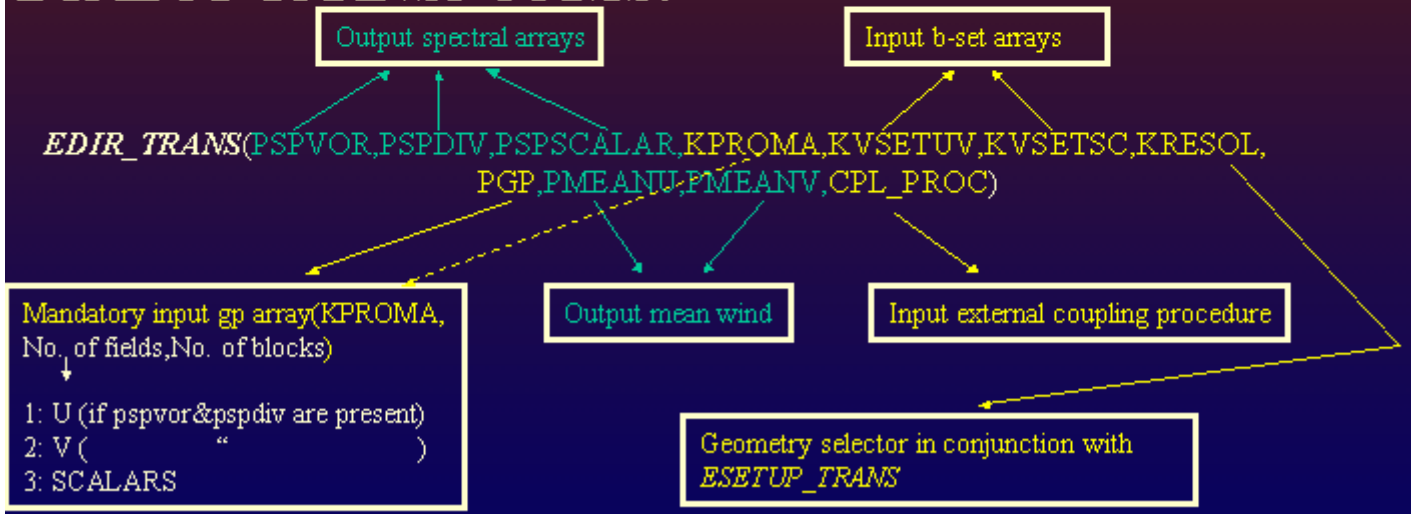
Some advantages:

- Separation (model part is not spoiled by internal business of transf.)
- Transf. Can be activated outside the model
- Modularity
- New coding features: (intensive use of optional arguments, structures...)

Transformation Package *TAL*

Most important for a user: *external*

DIRECT TRANSFORMS:



Transformation Package *TAL*

INVERSE TRANSFORMS:

EINV_TRANS(PSPVOR,PSPDIV,PSPSCALAR,FSPGL_PROC,LDSCDERS,LDVORGP,
LDDIVGP,LDUVDER,KPROMA,KVSETUV,KVSETSC,KRESOL,PGP,PMEANU,PMEANV)

Only ARPEGE: ext. proc. in FS

Logical switches indicating request for derivatives and vor/div in GP space

Mandatory GP array, field order:
- vorticity : IF_UV_G fields (if psvor present and LDVORGP)
- divergence : IF_UV_G fields (if psvor present and LDDIVGP)
- u : IF_UV_G fields (if psvor present)
- v : IF_UV_G fields (if psvor present)
- scalar fields : IF_SCALARS_G fields (if pspscalar present)
- N-S derivative of scalar fields : IF_SCALARS_G fields (if pspscalar present and LDSCDERS)
- E-W derivative of u : IF_UV_G fields (if psvor present and and LDUVDER)
- E-W derivative of v : IF_UV_G fields (if psvor present and and LDUVDER)
- E-W derivative of scalar fields : IF_SCALARS_G fields (if pspscalar present and LDSCDERS)

Here IF_UV_G is the GLOBAL number of u/v fields as given by the length of KVSETUV (or by PSPVOR if no split in spectral 'b-set' direction
IF_SCALARS_G is the GLOBAL number of scalar fields as giben by the length of KVESETSC (or by number of fields in PSPSCALAR if no spectral 'b-set' split

Transformation Package *TAL*

Other routines in *external:*

(not used by ALADIN, but useful for 'handmade' programs)

Diagnostics: *especnorm*

I/O distribution/gathering: *edist_spec,egath_spec*
edist_grid,egath_grid

Transformation Package *TAL*

```
if(myproc == 1) then
      open(unit=95,form='unformatted')
      do j=1,nflevg
            read(95) zvorg(j,1:nspec2g)
            read(95) zdivg(j,1:nspec2g)
            read(95) zspecg(j,1:nspec2g)
            enddo
endif
ifrom(1:nflevg)=1
call edist_spec(pspecg=zspecg,kfdistg=nflevg,kfrom=ifrom,pspec=zspec,kvset=ivset)
call edist_spec(pspecg=zvorg,kfdistg=nflevg,kfrom=ifrom,pspec=zvor,kvset=ivset)
call edist_spec(pspecg=zdivg,kfdistg=nflevg,kfrom=ifrom,pspec=zdiv,kvset=ivset)
```

Transformation Package *TAL*

```
SUBROUTINE EDIST_SPEC(PSPECG,KFDISTG,KFROM,KVSET,KRESOL,PSPEC)
!**** *EDIST_SPEC* - Distribute global spectral array among processors
!    Purpose.
!    -----
!      Interface routine for distributing spectral array
!**   Interface.
!    ------
!    CALL EDIST_SPEC(...)
!    Explicit arguments :
!    --------------------
!    PSPECG(:,:) - Global spectral array
!    KFDISTG     - Global number of fields to be distributed
!    KFROM(:)    - Processor resposible for distributing each field
!    KVSET(:)    - "B-Set" for each field
!    KRESOL      - resolution tag which is required ,default is the
!                  first defined resulution (input)
!    PSPEC(:,:)  - Local spectral array
```

Some remarks on local installation problems

- Start either from export version or Ccase→extract
- Use as general as possible Makefiles
- Start with xrd/module
- Start compilation of each project with modules
- Be aware of dependencies inside module directory
- Decide your default integer byte length (see *xrd/module/parkind*, xrd/include/precision.h*)
- Pay special attention on some directories of XRD (*support,grib_mf,...)*
- Problems may arise from intrinsic functions, mpi defaults…