

SWITCHING TOWARDS TFL VERSION OF SUGAW EVERYWHERE.

YESSAD Karim.

May 13, 2009

Version 1.

1 Introduction and purpose.

There are two versions of **SUGAW** (calculation of Gaussian weights and latitudes) in CY36: a version in project ARP/IFS and a version in TFL. Our purpose is to use only the TFL version.

2 Current status (in CY36).

There are currently two versions of routine **SUGAW**:

- The ARP/IFS one: arp/setup/sugawa.F90 (formerly named arp/setup/sugaw.F90). This routine is called by arp/setup/sugem1a.F90, arp/utility/grid_from_grib.F90, arp/fullpos/sufpg1.F90.
- The TFL one: tfl/module/sugaw_mod.F90. It is called under tfl/external/setup_trans.F90.

Additionally, we can notice that:

- We cannot call directly tfl/module/sugaw_mod.F90 in **SUGEM1A** because required TFL set-up has not been called at this level of the code.
- **SU0YOMB** currently computes twice Gaussian weights: the first one is done under **SUGEM1A** (call to arp/setup/sugawa.F90); the second one is done under **SUTRANS** (TFL set-up, call to tfl/module/sugaw_mod.F90).
- In CY36 the content of arp/setup/sugawa.F90 matches with the content of tfl/module/sugaw_mod.F90 (new way of computing Gaussian weights). But we have kept the old way of computing Gaussian weights in arp/setup/sugawa35.F90.

3 Action to switch towards a call to tfl/module/sugaw_mod.F90 everywhere.

This is possible to call tfl/module/sugaw_mod.F90 once the TFL set-up is done at least for the following variables:

- **RA** in the TFL module **TPM_CONSTANTS**.
- **NOUT** in the TFL module **TPM_GEN**.

Both quantities do not depend on the horizontal resolution: that means that, once these quantities are computed, we can use the TFL version of **SUGAW** for the current horizontal resolution (cf. **SUGEM1A**) but also for other horizontal resolutions (cf. **GRID_FROM_GRIB** and **SUFPG1**).

That leads to split the TFL set-up in two parts: a first part for the set-up necessary to compute the Gaussian weights and latitudes, and a second part for the set-up necessary to compute the Legendre polynomials and the other quantities.

The current organigramme of arp/setup/sutrans.F90 is:

```
SUTRANS ->
* SETUP_TRANSO
* SETUP_TRANS ->
  - SET_RESOL
  - SET_DIMS
```

```

- SUMP_TRANS_PRELEG
- SULEG ->
  * Calculation of ZFNLAT (input to SUGAW) and ZFN (input to SUPOL)
  * SUGAW
  * SUPOL
  * SUTRLE
- SETUP_GEOM
- SUMP_TRANS
- SUFFT

```

The proposal is to rewrite it as follows:

```

SUTRANS ->
* SETUP_TRANS0 (called if KIND=0 or 1)
* SETUP_TRANS1 (called if KIND=0 or 1) ->
  - SET_RESOL
  - SET_DIMS
  - SUMP_TRANS_PRELEG
  - Calculation of ZFNLAT (input to SUGAW)
  - SUGAW
* SETUP_TRANS2 (called if KIND=0 or 2) ->
  - SULEG ->
    * Calculation of ZFN (input to SUPOL)
    * SUPOL
    * SUTRLE
  - SETUP_GEOM
  - SUMP_TRANS
  - SUFFT

```

Call to **SUGAW** goes out of **SULEG**. **SUTRANS** will be given a dummy argument **KIND**.

- **KIND=0**: call to **SETUP_TRANS0**, **SETUP_TRANS1** and **SETUP_TRANS2**.
- **KIND=1**: call to **SETUP_TRANS0** and **SETUP_TRANS1** only.
- **KIND=2**: call to **SETUP_TRANS2** only.

A specific caller to **SUGAW** will be coded: `tfl/external/sugaw_caller.F90`. This routine will have dummy arguments **KN**, **PL**, **PDL**, **PW**. It computes **ZFNLAT** (**PFN** input argument of **SUGAW**), then calls **SUGAW**.

ARPEGE/IFS routines calling **SUGAWA** will be modified as follows:

- **SUGEM1A**: the pre-calculation of **ZFNLAT** and the call to **SUGAWA** will be replaced by a call to **SUTRANS** with **KIND=1**. A call to **TRANS_INQ**(**PMU=RMU(1);PGW=RW**) will then be done (currently done in **SUGEM**).
- **SUFPG1**: the pre-calculation of **ZFNLAT** and the call to **SUGAWA** will be replaced by a call to **SUGAW_CALLER**.
- **GRID_FROM_GRIB**: the pre-calculation of **ZFNLAT** and the call to **SUGAWA** will be replaced by a call to **SUGAW_CALLER**.

Additionally to that, the call to **SUTRANS** done in **SU0YOMB** will be replaced by a call to **SUTRANS** with **KIND=2**; the call to **TRANS_INQ** will be removed in **SUGEM**.

Remark: after the above modification:

- all the first part of **SUGEM** can be moved into **SUGEM1A**.
- In **SU0YOMB**, the call to **SUGEM** can be replaced by a call to **SUGEM1B**.

4 Conclusion.

The above proposal will allow to remove the ARPEGE/IFS version of **SUGAW**, and to call only once **SUGAW** under **SU0YOMB**. Coding can be done for **CY37**. Additionally, there is nothing to do for **ALADIN** (no Gaussian weight nor Legendre polynomial; routines **SUETRANS** and **ESETUP_TRANS** remain unchanged).

Appendix A: new routine tfl/external/sugaw_caller.F90.

```
SUBROUTINE SUGAW_CALLER(KN,PL,PDL,PW)

!**** *SUGAW_CALLER* - Caller routine to initialize the Gaussian
!                          abccissa and the associated weights

!   Purpose.
!   -----
!       Initialize arrays PL,PDL and PW (quadrature abscissas and weights)

!**  Interface.
!   -----
!   CALL SUGAW_CALLER(...)

!   Explicit arguments :
!   -----
!       INPUT:
!           KN      : Number of Gauss abscissas
!       OUTPUT:
!           PL (KN) : abscissas of Gauss
!           PDL(KN) : idem in double precision
!           PW (KN) : Weights of the Gaussian integration

!   PL (i) is the abscissa i starting from the northern pole, it is the
!   cosine of the colatitude of the corresponding row of the collocation grid.

!   Method.
!   -----
!       See documentation

!   Externals.
!   -----

!   Reference.
!   -----

!   S.L. Belousov, Tables of normalized associated Legendre Polynomials,
!   Pergamon Press (1962)
!   P.N. Swarztrauber, On computing the points and weights for Gauss-Legendre
!   quadrature, SIAM J. Sci. Comput. Vol. 24 (3) pp. 945-954 (2002)

!   Author.
!   -----
!       Mats Hamrud *ECMWF* and K. Yessad

!   Modifications.
!   -----
!       Original : May 2008
!   -----

USE PARKIND1 ,ONLY : JPIM      ,JPRB

#ifdef INTERFACE
USE SUGAW_MOD
USE YOMHOOK ,ONLY : LHOOK,   DR_HOOK
#endif INTERFACE

!   -----

IMPLICIT NONE

INTEGER(KIND=JPIM),INTENT(IN) :: KN
REAL(KIND=JPRB),INTENT(OUT) :: PL(:)
```

```

REAL(KIND=JPRB),INTENT(OUT) :: PDL(:)
REAL(KIND=JPRB),INTENT(OUT) :: PW(:)

! -----

!ifndef INTERFACE

INTEGER(KIND=JPIM) :: IDGLG,ISTART,IK,IODD,JN
REAL(KIND=JPRB),ALLOCATABLE :: ZFN(:,,:), ZFNLAT(:)
REAL(KIND=JPRB) :: ZFNN

! -----
IF (LHOOK) CALL DR_HOOK('SUGAW_CALLER',0,ZHOOK_HANDLE)
! -----

! * preliminary calculations to compute input quantity ZFNLAT
! (k.y.: coded after what I found in tfl/module/suleg_mod.F90).
IDGLG=KN
ISTART=1
ALLOCATE(ZFN(0:IDGLG,0:IDGLG))
ALLOCATE(ZFNLAT(0:IDGLG/2))
! Belousov, Swarztrauber use ZFN(0,0)=SQRT(2._JPRB)
! IFS normalisation chosen to be 0.5*Integral(Pnm**2) = 1
ZFN(0,0)=2._JPRB
DO JN=ISTART,IDGLG
  ZFNN=ZFN(0,0)
  DO JGL=1,JN
    ZFNN=ZFNN*SQRT(1._JPRB-0.25_JPRB/REAL(JGL**2,JPRB))
  ENDDO
  IODD=MOD(JN,2)
  ZFN(JN,JN)=ZFNN
  DO JGL=2,JN-IODD,2
    ZFN(JN,JN-JGL)=ZFN(JN,JN-JGL+2) &
      & *REAL((JGL-1)*(2*JN-JGL+2),JPRB)/REAL(JGL*(2*JN-JGL+1),JPRB)
  ENDDO
ENDDO

IODD=MOD(IDGLG,2)
IK=IODD
DO JGL=IODD,IDGLG,2
  ZFNLAT(IK)=ZFN(IDGLG,JGL)
  IK=IK+1
ENDDO

! * call to SUGAW:
CALL SUGAW(KN,ZFNLAT,PL,PDL,PW)

! * deallocations:
DEALLOCATE(ZFN)
DEALLOCATE(ZFNLAT)

! -----
IF (LHOOK) CALL DR_HOOK('SUGAW_CALLER',1,ZHOOK_HANDLE)
!endif INTERFACE
END SUBROUTINE SUGAW_CALLER

```