Report on the porting of ALADIN CY29T2 export to NEC SX-6

Martin Janoušek, CHMI

original 3 October 2005, modified on 22 December 2005

The export package of CY29T2 was ported for NEC SX-6 using the cross-compiler environment Rev 305 on Linux machine. The locally developed source code analyzer and dependency tool (developed by Tomáš Kalibera) was used.
The aim of this limited porting action was to install and compile the code on NEC in order to run configuration 001 (including fullpos) and ee927. Next step will be to install ODB part and validate data assimilation configurations.

Preparation of source code

CY29T2 contains new projects mpa and mse containing MésoNH physics subroutines.

mnh ➔ F90

The first step was to change suffices of source code files from *.mnh* to *.F90*.

Limited subset of mpa/mse

Then only necessary modules referenced by ALADIN/AROME interface routines to MesoNH physics were left in the project directories mpa/ and mse/ because this limited port of CY29T2 did not suppose to really call MesoNH physics. The following modules are needed in order to compile apr and ald:

mpa/turb/module/modi_shumanaro.F90
mpa/turb/module/modi_turb.F90
mpa/turb/module/modi_ini_cturb.F90
mpa/micro/module/modd_cst.F90
mpa/micro/module/modd_param_ice.F90
mpa/micro/module/modi_rain_ice.F90
mpa/micro/module/modi_ini_cst.F90
mpa/micro/module/modd_les.F90
mpa/micro/module/modi_ini_rain_ice.F90
mpa/micro/module/modd_rain_ice_descr.F90
mpa/micro/module/modi_ice_adjust.F90
mpa/micro/module/modd_conf.F90
mpa/micro/module/modd_parameters.F90
mpa/micro/module/modd_rain_ice_param.F90
mse/module/modd_type_date_surf.F90
mse/module/modi_fmread.F90
mse/module/modi_fmwrit.F90
mse/module/modi_ini_sw_setup.F90
mse/module/modi_ini_sun.F90

Further modifications of source code

Subroutines with too long lines

The following subroutines contained lines over 132 characters long and it was necessary to split them:

- arp/phys_dmn/accvimp.F90
- ald/module/eggpack.F90
- ald/module/eggangles.F90
- tal/programs/test_adjoint.F90
- tal/programs/aatestprog.F90
- xrd/ddh/ddhr.F90

Kind of literal constants in USNUNP

In arp/c9xx/usnunp.F90 some literal constants cannot be evaluated in the default kind INTEGER (4 bytes):

I39=2**39		I39=2_JPIB**39_JPIB
I48=2**48	➔	I48=2_JPIB**48_JPIB
I53=2**53		I53=2_JPIB**53_JPIB

Misplaced IMPLICIT NONE

In arp/pp_obs/reflsim.F90 the IMPLICIT NONE statement was placed in wrong place.

Loop variable

In arp/var/sualcos.F90 a loop-control variable JOTP is passed as an actual argument to the subroutine CMOCTMAP where it has INOUT status. This may cause a change of a loop-control variable which is prohibited, at least by NEC compiler. The same for the variable JCTP. Therefore loop local variables were introduced as a work-around.

SX-6 compiler options

SX-6 compiler options were added to ald/adiab/elascaw.F90 and arp/setup/suhdf.F90. These options must be placed on the lines before SUBROUTINE statement (no empty lines allowed) so it cannot be implemented via #ifdef SX4 CPP macro.

Local modifications of ALADIN physics

ACDIFUS

Modified calculation of $g\,z_H$ term

```
…
ZIUSTAR0=10.
…
!     VI - CALCULS POUR LA QUANTITE DE MOUVEMENT : CONDITION DE SURFACE,
!     CALCUL DE "CHARNOCK" PUIS SUBSTITUTION AVEC CALCUL DES FLUX A TOUS
!     LES NIVEAUX.
…
DO JLON=KIDIA,KFDIA
  ZELIM=ZXURO(JLON,KLEV-1)*ZIPOI(JLON,KLEV)
  ZMUL=1.0_JPRB/(1.0_JPRB+ZELIM*(1.0_JPRB-ZSUB1(JLON,KLEV-1))+ZXDRO(JLON)&
   & *ZIPOI(JLON,KLEV))
  ZN1(JLON,KLEV)=ZMUL*(PU(JLON,KLEV)+ZELIM*ZN1(JLON,KLEV-1))
  ZN2(JLON,KLEV)=ZMUL*(PV(JLON,KLEV)+ZELIM*ZN2(JLON,KLEV-1))
  ZMERL=(1.0_JPRB-PITM(JLON))*(1.0_JPRB &
   & -MAX(0.0_JPRB,SIGN(1.0_JPRB,TMERGL-PTS(JLON))))
  PGZ0F(JLON)=PGZ0F(JLON)+ZMERL &
   & *(RG*VZ0CM*PCD(JLON)/PCDN(JLON)+VCHRNK*PCD(JLON)&
   & *SQRT((ZN1(JLON,KLEV)**2+ZN2(JLON,KLEV)**2)&
   & *(PU(JLON,KLEV)**2+PV(JLON,KLEV)**2))-PGZ0F(JLON))
!!PGZ0HF(JLON)=(1.0_JPRB-ZMERL)*PGZ0HF(JLON)+ZMERL*PGZ0F(JLON)*RZHZ0M
  PGZ0HF(JLON)=PGZ0HF(JLON)+ZMERL&
   &*(PGZ0F(JLON)*EXP(-SQRT(PCD(JLON)&
   &*SQRT((ZN1(JLON,KLEV)**2+ZN2(JLON,KLEV)**2)&
   &*(PU(JLON,KLEV)**2+PV(JLON,KLEV)**2)))*ZIUSTAR0)-PGZ0HF(JLON))
  PSTRTU(JLON,KLEV)=PCDROV(JLON)*ZN1(JLON,KLEV)
  PSTRTV(JLON,KLEV)=PCDROV(JLON)*ZN2(JLON,KLEV)
ENDDO
…
```

ACHMT

Passing the modified calculation of $g\,z_H$ term

```
…
!      CAS DE DEUX LONGUEURS DE RUGOSITE (SUR CONTINENT NON TOTALEMENT
!      COUVERT DE NEIGE).
!      CASE OF TWO ROUGHNESS LENGTHS (ON PARTIALLY SNOW-FREE LAND ONLY).

  IF (LLZ0H) THEN

    IF (LNEIGE) THEN
      IF (LSNV) THEN
        PGZ0H(JLON)=SQRT((1.0_JPRB-PNEIJV(JLON))*PGZ0HF(JLON)**2+&
         & PNEIJV(JLON)    *&
         & (Z0CR**2+(PGZ0RLF(JLON)*PGZ0HF(JLON)/PGZ0F(JLON))**2))
      ELSE
        PGZ0H(JLON)=PGZ0HF(JLON)+(Z0CR-PGZ0HF(JLON))*PITM(JLON)&
         & *PSNS(JLON)/(PSNS(JLON)+WCRIN*(1+ZUZ0CN*PGZ0HF(JLON)))
      ENDIF
    ELSE
      PGZ0H(JLON)=PGZ0HF(JLON)
    ENDIF

! PGZ0H = ZGZ0 OVER SEA UNLESS REQUIRED IN NAMPHY1 (CLIMATE RUNS)

!!! PGZ0H(JLON)=MIN(PGZ0(JLON),PITM(JLON)*PGZ0H(JLON)&
!!!  & +(1.0_JPRB-PITM(JLON))*PGZ0(JLON)*RZHZ0M)
    PGZ0H(JLON)=PITM(JLON)*PGZ0H(JLON)&
     & +(1.0_JPRB-PITM(JLON))*PGZ0HF(JLON)

  ELSE

    PGZ0H(JLON)=PGZ0(JLON)

  ENDIF
…
```

ACCLPH

Martina Tudor's modification of the Ayotte-Geleyn-Piriou scheme with further amendment of Jean-François Geleyn (declaration of new local arrays is not quoted here):

```
…
DO JLON=KIDIA,KFDIA
  PCLPH(JLON)=0.0_JPRB
  ZCLPU(JLON)=0.0_JPRB
  ZCLPV(JLON)=0.0_JPRB
  KCLPH(JLON)=KLEV
  ZINT(JLON)=0.0_JPRB
  ZX_PREV(JLON)=0.0_JPRB
  ZLCH_PREV(JLON)=ZVAL_LCH
  ZTHETAV(JLON,KLEV+1)=MAX(PTHETAVS(JLON),PTHETAV(JLON, KLEV))
  ZTHETAVS(JLON,KLEV+1)=ZTHETAV(JLON,KLEV+1)
  ZPHI(JLON,KLEV+1)=PAPHI(JLON,KLEV)
  ZU(JLON,KLEV+1)=0.0_JPRB
  ZV(JLON,KLEV+1)=0.0_JPRB
ENDDO
DO JLEV=KTDIA,KLEV
  DO JLON=KIDIA,KFDIA
    ZTHETAV(JLON,JLEV)=PTHETAV(JLON,JLEV)
    ZPHI(JLON,JLEV)=PAPHIF(JLON,JLEV)
    ZU(JLON,JLEV)=PU(JLON,JLEV)
    ZV(JLON,JLEV)=PV(JLON,JLEV)
  ENDDO
ENDDO
!MT shear linked convection modification
DO JLEV=KLEV,KTDIA,-1
  DO JLON=KIDIA,KFDIA
    ZTHETAVS(JLON,JLEV)=ZTHETAVS(JLON,JLEV+1)&
     &+ZTHETAV(JLON,JLEV)-ZTHETAV(JLON,JLEV+1)&
     &-0.5_JPRB*(ZTHETAV(JLON,JLEV)+ZTHETAV(JLON,JLEV+1))&
     &*((ZU(JLON,JLEV)-ZU(JLON,JLEV+1))**2+&
     &  (ZV(JLON,JLEV)-ZV(JLON,JLEV+1))**2)&
     &/(ZPHI(JLON,JLEV)-ZPHI(JLON,JLEV+1))
  ENDDO
ENDDO
```

```
…
! UPWARD INTEGRATIONS.
!-------------------------------------------------

DO JLEV=KLEV,KTDIA,-1
  DO JLON=KIDIA,KFDIA
    !
    !-------------------------------------------------
    ! INTEGRALE DE THETAV.
    ! THETAV INTEGRAL.
    !-------------------------------------------------
    !
    ZINT(JLON)=ZINT(JLON)+(ZPHI(JLON,JLEV)-ZPHI(JLON,JLEV+1)) &
     & *0.5_JPRB*(ZTHETAVS(JLON,JLEV)+ZTHETAVS(JLON,JLEV+1))
    !
    !-------------------------------------------------
    ! ECART THETAV' ENTRE THETAV DU NIVEAU COURANT
    ! ET LA MOYENNE DE THETAV ENTRE LA SURFACE ET LE NIVEAU COURANT.
    ! COMPUTE THETAV', DIFFERENCE BETWEEN CURRENT THETAV VALUE
    ! AND ITS INTEGRAL BETWEEN SURFACE AND CURRENT LEVEL.
    !-------------------------------------------------
    !
    ZX=ZTHETAVS(JLON,JLEV)-ZINT(JLON)/(ZPHI(JLON,JLEV)-ZPHI(JLON,KLEV+1))
…
```

ACDRAG

Better vertical distribution of the orographic drag force along the slopes:

```
…
!     -----------------------------------------------------------------
!     V - BOUCLE DESCENDANTE POUR LA COMBINAISON DES EFFETS ET LES
!     SECURITES NUMERIQUES CONDUISANT AU CALCUL DEFINITIF DES FLUX.

!         DESCENDING LOOP FOR THE COMBINATION OF EFFECTS AND THE
!     NUMERICAL SECURITIES LEADING TO THE FINAL FLUX CALCULATION.

!     FLUX PROVISOIRES PRENANT EN COMPTE LES TROIS EFFETS.
!     PROVISIONAL FLUXES TAKING INTO ACOUNT ALL THREE EFFECTS.

DO JLEV=KTDIA,KLEV
  DO JLON=KIDIA,KFDIA
    ZRAPP(JLON,JLEV)=ZDS2(JLON)*MAX(0.0_JPRB,PAPRS(JLON,JLEV)&
     & -ZPR2(JLON))&
     & +MAX(0.0_JPRB,MIN(ZSTM(JLON),ZRAPP(JLON,JLEV)&
     & +ZDS1(JLON)*MAX(0.0_JPRB,PAPRS(JLON,JLEV)&
     & -ZPR1(JLON)))-ZSTS(JLON))
!!! ZRZ=MIN(1.0_JPRB,(PAPRS(JLON,KLEV)-PAPRS(JLON,JLEV))&
!!!  & /MAX(ZEPS4,ZZBB(JLON)*HOBST*PGETRL(JLON)&
!!!  & *PGWDCS(JLON)))
    ZRZ=(PAPRS(JLON,KLEV)-PAPRS(JLON,JLEV))&
     & /MAX(ZEPS4,ZZBB(JLON)*HOBST*PGETRL(JLON)&
     & *PGWDCS(JLON))

    IF (.NOT.LNEWD) THEN
      ZZCR(JLON)=ZRAPP(JLON,JLEV)
    ENDIF

    ZRZR=SQRT(ZZBB(JLON)/(1.0_JPRB+ZRZ**2*ZZBB(JLON)*(1.0_JPRB-ZZBB(JLON))))
    ZRAPP(JLON,JLEV)=ZRAPP(JLON,JLEV)+ZZCR(JLON)*(ZZAA(JLON)&
     & *SQRT(MAX(0.0_JPRB,(1.0_JPRB-ZRZ*ZRZR)**3/(1.0_JPRB+ZRZ*GWDPROF&
     & *ZZBB(JLON)))))
  ENDDO
ENDDO
…
```

ACPLUIE

Correction of a formula bug:

```
…
!     CALCULS COMPLETS D'EVAPORATION ET FONTE/(GEL) SI ZSIGMA >0,
!     RECALCUL DE ZSIGMA ET INITIALISATION EVENTUELLE DES PROPORTIONS
!     NEIGEUSES DU FLUX DANS LE CAS CONTRAIRE.
!     COMPLETE COMPUTATIONS OF EVAPORATION AND MELTING/(FREEZING) IF
!     ZSIGMA >0, RECOMPUTATION OF ZSIGMA AND POTENTIAL INITIALIZATION
!     OF THE SNOW PROPORTIONS OF THE FLUX IN THE OPPOSITE CASE.
```

```
  IF (ZSIGMA > 0.0_JPRB) THEN
    DO JLON=KIDIA,KFDIA
      ZEVA=EVAP*SQRT(1.0_JPRB-ZRME(JLON)*(1.0_JPRB-REVGSL))
      ZFON=FONT*SQRT(1.0_JPRB-ZRME(JLON)*(1.0_JPRB-REVGSL))
      ZRPTH=SQRT(MAX(ZFPTH(JLON),ZFPTB(JLON)))
      ZRPTB=MAX(0.0_JPRB,ZRPTH-ZEVA*MAX(0.0_JPRB,ZDQ(JLON))*(ZIPH(JLON)&
       & -ZIPB(JLON)))
      ZFPTB(JLON)=MAX(ZFPTB(JLON),ZRPTB**2)
…
```

Validation tests

The ported code was compiled and a set of validation tests against CY28T3stratus executable which is currently operational at CHMI was performed.

Adiabatic run

In order to receive the same spectral norms in both cycles it was necessary

- to modify setting LLNEWGRP to .TRUE. in CPG_GP of CY28T3stratus in order to activate new calculation of pressure gradient (which is the default in CY29T2)
- to have the initial and coupling files with new format of cadre (made by ee927 with NCADFORM=1) in order to avoid call of EGGX which gives slightly different results in both cycles (probably NEC SX-specific feature).

Diabatic run

The norms were equal up to the end of the first coupling interval, then they started to diverge. The reason was found in the missing GFL(…,YGFLC,…) storage before and restoring after the call of ELSIRF and ELSWA3 in ERLBC. This local storage/restoring were added into ERLBC in CY29T2 but it is obviously in place also in CY28. After this fix of CY28T3 both cycles give identical results.

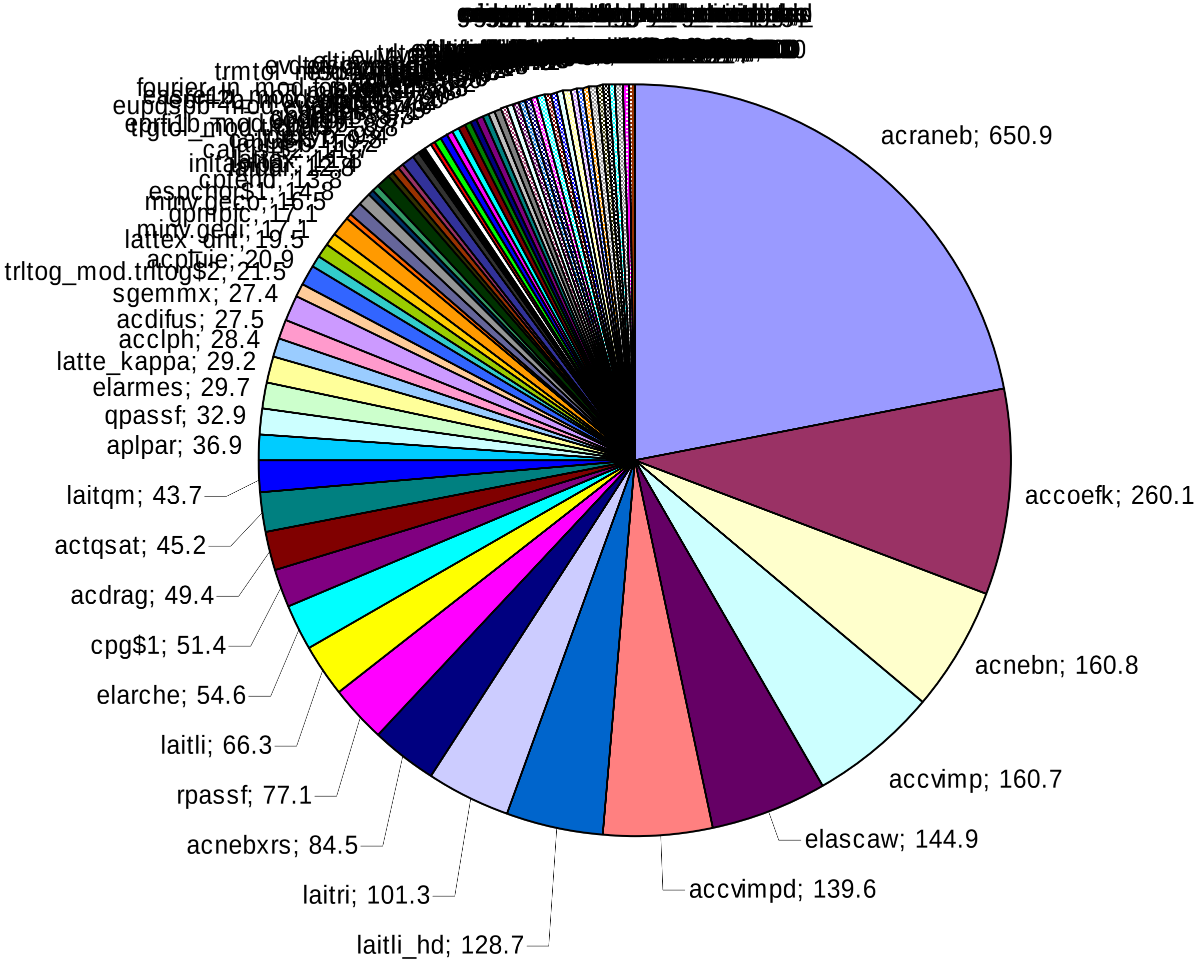
Benchmarks

OpenMP

CY29T2 provides the same results if run under OpenMP parallelization like under MPI. This is a change from CY28T3 where the results differed. Benchmark tests on 4-processor NEC SX-6B shows an acceptable speed-up on 4 OpenMP threads (ratio user/real time was 3.61) compared to 4-processes MPI run (ratio user/real time 3.94) with dramatic savings in memory (4224 MB OpenMP vs. 8402 MB MPI).
When switching from 1 to 4 threads the total user (=CPU) time increases by about 10% (similar to MPI).
ALADIN/CE 24 hour forecast with 4 threads was run in 870 s on a nearly-empty SX-6.

Profile

A breakdown of the subroutines costs was obtained by the reference 4-thread 24 hour forecast run, see the next page.
Some routines are suspiciously expensive (ACCOEFK, ACNEBXRS) which calls for additional vectorization check. The profile published here is therefore not definitive.



acraneb; 650.9
accoefk; 260.1
acnebn; 160.8
accvimp; 160.7
elascaw; 144.9
accvimpd; 139.6
laitli_hd; 128.7
laitri; 101.3
acnebxrs; 84.5
rpassf; 77.1
laitli; 66.3
elarche; 54.6
cpg$1; 51.4
acdrag; 49.4
actqsat; 45.2
laitqm; 43.7
aplpar; 36.9
qpassf; 32.9
elarmes; 29.7
latte_kappa; 29.2
acclph; 28.4
acdifus; 27.5
sgemmx; 27.4
trltog_mod.trltog$2; 21.5
acpluie; 20.9
lattex_dnt; 19.5