New features of EULAG 2P/3P versions

A. Wyszogrodzki

Overview

Code reorganization

- ***Setup parameters***
- ***Reorganization: main driver, tinit, topolog split into horizontal/vertical/IMB***
- ***Common blocks***
- ***New variables***
- ***3D Jacobian of the coordinate transformation***

New I/O options

- ***Serial/Parallel netcdf output and restart***
- ***COSMO coupler (mesoscale applications)***
- ***WRF coupler (LES applications)***
- ***Runtime analysis (PBL/microphysics based on RICO/porous media)***

New parameterizations / technical options

- **Physics: surface layer, microphysics (ice AB, bin), radiation (Utah/CCM2/etc.)**
- **Numerics: steretching in SGS1, extended SGS2, POLES=1, polar absorbers, itras**
- **New model formulations: pseudo-incompress, incompress Euler, bussinesq, deep atmosphere; FUTURE: MHD, compressible**

Predefined TEST CASES

Code reorganization
Remove redundancy in parameters

```
#define MOISTMOD 0  /* 0=DRY, 1=BULK-WARM, 2=BULK-ICE A+B, 3=BIN-WARM      */
#define SGS 1       /* DIFFUSION: 0=ILES, 1=OLD DISSIP, 2=DISSIP AMR       */
#define CHEMIS 0    /* PASSIVE TRACER: 0=OFF/1=ON                          */
#define LAGDIS 1    /* PASSIVE TRACER: 0=OFF/1=ON                          */
#define IMRSB 0     /* IMMERSED BOUNDARY: 0=OFF/1=ON                       */
```

```
ccccccccccccccccccccccccccccccccccccccccccccccc
c Chemistry
c-----------------------------------------------
c ---    ichm - chemistry on (1) / off (0)
c ---    nspc - number of chemical species
c ---    ilgd - lagrangian dispalcements
ccccccccccccccccccccccccccccccccccccccccccccccc
      parameter (ichm=CHEMIS,nspc=3)
      parameter (ilgd=LAGDIS,nlgd=3)
      parameter (nthch=nspc*nthv)
```

```
ccccccccccccccccccccccccccccccccccccccc
control immersed boundary
ccccccccccccccccccccccccccccccccccccccc
      parameter(imrsb=IMRSB)
```

Code reorganization

Move ‘all’ options to parameter files/blockdata

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c GCRK solver and preconditioners
c-------------------------------------------------------------------------------
c --- lrd  - set the size of Krylov subspace lord=lrd
c --- itmn - minimum number of outer cycles of pressure iterations
c            e.g., calls to precon/lapc: see 200 loop below
c --- iprc - use preconditioneer (default on), igrid=1 (B grid) -> iprc=0
c --- ispcpr - use spectral preconditioneer (default off)
c-------------------------------------------------------------------------------
c --> see common/itero/   -> niter,nitsm,icount,miter,mitsm,jcount,eer,eem
c --> see common/itero/   -> itp0,epp0,itp1,epp1
c --> see common/precond/ -> beta,itr,line
c --> see common/indx/    -> e1,e2,e3
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      parameter(lrd=3)
      parameter(itmn=1)
      parameter(iprc=0)
      parameter(ispcpr=0)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c ---  MPDATA icontrol parameters
c-------------------------------------------------------------------------------
c--> iord0 - number of mpdata iterations; iord0=2 defult 2nd order mpdata
C--> isor  - isor=3 third order mpdata, isor=1 default for 2nd order mpdata
C--> nonos - nonos=1 nonoscillatory option (monotonicity ehnacement)
C--> idiv  - idiv=0 default for soundproof equations
c-------------------------------------------------------------------------------
c variables with "a"/"m"/"i" at the end for mpdata/mpdatm/inter respectively
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      parameter(iord0a=3,isora=3,nonosa=1,idiva=1) ! mpdata2/mpdata3
      parameter(         isorm=1,nonosm=1,idivm=1) ! mpdatm2/mpdatm3
      parameter(                 nonosi=1        ) ! inter2/inter3
      parameter(impli1d=1,iord1d=1,nonos1d=1     ) ! moist 1d advec
```

Code reorganization common blocks

```
#######
rm -f common.gwimpl
cat > common.gwimpl << '\eof'
      common/gwimpl/ dthe(1-ih:np+ih,1-ih:mp+ih,l,3)
'\eof'
#######
rm -f common.profm
cat > common.profm << '\eof'
      common/profm/tme(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
     .             qve(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
     .             qce(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
     .             qre(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms)
      common/prsss/qvs(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms)
'\eof'
#######
rm -f common.tinits
cat > common.tinits << '\eof'
      common/tinits/ initi,lipps
'\eof'
#######
rm -f common.ctherm
cat > common.ctherm << '\eof'
      common/ctherm/ rg,cp,cap,st,g,th00,tt00,pr00,rh00,u00,v00,u0z,v
      common/ctherd/ bv
'\eof'
```

```
      subroutine lstsq(th,z,nz,a,b)
#include "param.nml"
#include "msg.inc"
#include "incl/common.lsty"
      dimension th(nz),z(nz)

      sy=0
      sz=0
      syz=0
      szz=0
      do k=1,nz
        y(k)=alog(th(k))
      end do
      do k=1,nz
        sz  = sz  + z(k)
        szz = szz + z(k)*z(k)
        syz = syz + y(k)*z(k)
        sy  = sy  + y(k)
      end do
      a=(sy*szz-sz*syz)/(float(nz)*szz-sz*sz)
      b=(float(nz)*syz-sy*sz)/(float(nz)*szz-sz*sz)
      a=exp(a)

      return
      end
```

```
      subroutine rhngck(rho)
#include "param.nml"
#include "msg.inc"
#include "incl/common.sphere"
#include "incl/common.metric"
#include "incl/common.metrit"
#include "incl/common.adapt"

      dimension rho(1-ih:np+ih, 1-ih:mp+ih, l)

#if (TIMEPLT == 1)
      call ttbeg(22)
#endif
```

Code reorganization new variables

- New environmental variables: w_e, qc_e, qr_e

```
      common/profl/th0(1-ih:np+ih,1-ih:mp+ih,l),
     .             rho(1-ih:np+ih,1-ih:mp+ih,l),
     .             the(1-ih:np+ih,1-ih:mp+ih,l),
     .              ue(1-ih:np+ih,1-ih:mp+ih,l),
     .              ve(1-ih:np+ih,1-ih:mp+ih,l),
     .              we(1-ih:np+ih,1-ih:mp+ih,l),
     .             zcr(l)

     common/profm/tme(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
    .            qve(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
    .            qce(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms),
    .            qre(1-ih:nmsp+ih, 1-ih:mmsp+ih, lms)
```

- Full pressure

```
cccccccccccccccccccccccccc
c create actual full pressure variable
cccccccccccccccccccccccccc
#include "incl/common.pact"     /*pact,pbeg*/

#######
rm -f common.pact
cat > common.pact << '\eof'
      common/pactu/ pact(1-ih:np+ih,1-ih:mp+ih,l),
     .              pbeg(1-ih:np+ih,1-ih:mp+ih,l)
'\eof'
#######
```

Code reorganization
3D Jacobian

```
*/
#include "incl/common.jacobian" /* g11 .... g33, c11 ... c33 */


        g110=1./((1-icylind)*gmm(i,j,k)*cosa(i,j)+icylind*1.)
        g220=1./gmm(i,j,k)
      g11=strxx(i,j)*g110
      g12=stryx(i,j)*g110
      g13=(s13(i,j)*gmul(k)-h13(i,j))*gmus(k)*g110
      g21=strxy(i,j)*g220
      g22=stryy(i,j)*g220
      g23=(s23(i,j)*gmul(k)-h23(i,j))*gmus(k)*g220
      g33=gi(i,j)*gmus(k)
    ox(i,j,k,0)=g11*u(i,j,k,0)+g21*v(i,j,k,0)
    oy(i,j,k,0)=g12*u(i,j,k,0)+g22*v(i,j,k,0)
    oz(i,j,k,0)=g13*u(i,j,k,0)+g23*v(i,j,k,0)+g33*w(i,j,k,0)



    ox(i,j,k,0)=g11(i,j,k)*u(i,j,k,0)+g21(i,j,k)*v(i,j,k,0)
    oy(i,j,k,0)=g12(i,j,k)*u(i,j,k,0)+g22(i,j,k)*v(i,j,k,0)
    oz(i,j,k,0)=g13(i,j,k)*u(i,j,k,0)+g23(i,j,k)*v(i,j,k,0)
   .           +g33(i,j,k)*w(i,j,k,0)
```

g11…g33 – defined in subroutine metryc

Code reorganization
3D Jacobian

```
*/
#include "incl/common.jacobian" /* g11 .... g33, c11 ... c33 */
compute interior pressure forces
c--->      c = astri
c--->     fc = Gmod
c--->     fd = 1./etainv
      do 10 k=2-ibcz,l-1+ibcz
      do 10 j=1,mp
      do 10 i=1,np
            F2=.5*dt*fcr2(i,j)*c(i,j,k)*initprs
            F3=.5*dt*fcr3(i,j)*c(i,j,k)*initprs
            g110=1./((1-icylind)*gmm(i,j,k)*cosa(i,j)+icylind*1.)
            g220=1./gmm(i,j,k)
            g11=strxx(i,j)*g110
            g12=stryx(i,j)*g110
            g13=(s13(i,j)*gmul(k)-h13(i,j))*gmus(k)*g110
            g21=strxy(i,j)*g220
            g22=stryy(i,j)*g220
            g23=(s23(i,j)*gmul(k)-h23(i,j))*gmus(k)*g220
            g33=gi(i,j)*gmus(k)
            Rt=ft(i,j,k)
         a11= Rt*(g11+F3*g21)+fc(i,j,k)*(dthe(i,j,k,3)*g11
     +           +(F3*dthe(i,j,k,3)+F2*dthe(i,j,k,2))*g21)
         a12= Rt*(g12+F3*g22)+fc(i,j,k)*(dthe(i,j,k,3)*g12
     +           +(F3*dthe(i,j,k,3)+F2*dthe(i,j,k,2))*g22)
         a13= Rt*(g13+g23*F3-g33*F2)+fc(i,j,k)*(
     .       g23*F2*dthe(i,j,k,2)+(g13+g23*F3)*dthe(i,j,k,3) )
         a21= Rt*(g21*(1.+F2*F2)-g11*F3)+fc(i,j,k)*(-F2*g21*dthe(i,j,k,1)
     +      +(g21-F3*g11)*dthe(i,j,k,3))
         a22= Rt*(g22*(1.+F2*F2)-g12*F3)+fc(i,j,k)*(-F2*g22*dthe(i,j,k,1)
     +      +(g22-F3*g12)*dthe(i,j,k,3))
         a23= Rt*((1.+F2*F2)*g23-F3*g13+F2*F3*g33)
     .      +fc(i,j,k)*((g23-g13*F3)*dthe(i,j,k,3)-g23*F2*dthe(i,j,k,1))
         a31= Rt*(F2*g11+F2*F3*g21)-fc(i,j,k)*((g11+F3*g21)*dthe(i,j,k,1)
     +      -(F3*g11-g21)*dthe(i,j,k,2))
         a32= Rt*(F2*g12+F2*F3*g22)-fc(i,j,k)*((g12+F3*g22)*dthe(i,j,k,1)
     +      -(F3*g12-g22)*dthe(i,j,k,2))
         a33= Rt*(F2*g13+F2*F3*g23+(1+F3*F3)*g33)
     +      -fc(i,j,k)*((g13+F3*g23)*dthe(i,j,k,1)
     +      -(F3*g13-g23)*dthe(i,j,k,2))
            c11=g11*a11+g21*a21
            c12=g11*a12+g21*a22
            c13=g11*a13+g21*a23
            c21=g12*a11+g22*a21
            c22=g12*a12+g22*a22
            c23=g12*a13+g22*a23
            c31=g13*a11+g23*a21+g33*a31
            c32=g13*a12+g23*a22+g33*a32
            c33=g13*a13+g23*a23+g33*a33
           pfx(i,j,k)=u(i,j,k)-(c11*px(i,j,k)+c12*py(i,j,k)+c13*pz(i,j,k))
           pfy(i,j,k)=v(i,j,k)-(c21*px(i,j,k)+c22*py(i,j,k)+c23*pz(i,j,k))
        10 pfz(i,j,k)=w(i,j,k)-(c31*px(i,j,k)+c32*py(i,j,k)+c33*pz(i,j,k))
```

```
compute interior pressure forces
c--->      c = astri
c--->     fc = Gmod
c--->     fd = 1./etainv
      do 10 k=2-ibcz,l-1+ibcz
      do 10 j=1,mp
      do 10 i=1,np
       pfx(i,j,k)=u(i,j,k)-(c11(i,j,k)*px(i,j,k)
     +                     +c12(i,j,k)*py(i,j,k)
     +                     +c13(i,j,k)*pz(i,j,k))
       pfy(i,j,k)=v(i,j,k)-(c21(i,j,k)*px(i,j,k)
     +                     +c22(i,j,k)*py(i,j,k)
     +                     +c23(i,j,k)*pz(i,j,k))
    10 pfz(i,j,k)=w(i,j,k)-(c31(i,j,k)*px(i,j,k)
     +                     +c32(i,j,k)*py(i,j,k)
     +                     +c33(i,j,k)*pz(i,j,k))
```

c11…c33 – defined in subroutine coef0

Memory bound performance issues

Code reorganization
main driver reorganization

Work in progress:

- Splitting logically consistent parts into separate blocks/subroutines

- Goal – an ‘easy’ adaptation of new configurations (MHD/icompress/etc.)

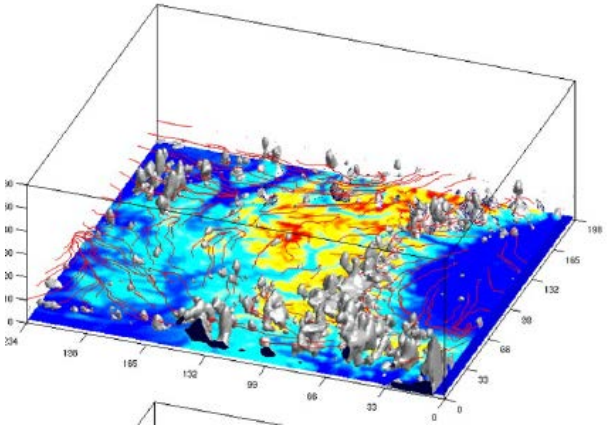
Model formulations

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c Basic physical/numerical parameters
c-------------------------------------------------------------------------------
c -->   ipsinc = 1 pseudoincompressible (Durran) equations.
c -->            0 anelastic/bussinesq in theta system (default)
c -->  incmpeul= 1 incompressible euler requires ipsinc = 0
c -->  ibousrho= 1 bussinesq in tho     requires ipsinc = 0
c -->   implgw = 1 advect theta perturbation (default)
c ---            0 advect full theta
c -->  isphere = 0 reference coordinates are Cartesian or Cylindrical
c ---             (need to specify dx,dy,dz in blanelas)
c ---            1 reference coordinates are spherical
c ---              (need to specify only dz in blanelas)
c -->  icylind = 0 reference coordinates are Cartesian or Spherical
c ---              reference coordinates are cylindrical
c -->   icorio = 0 no coriolis accelerations
c ---            1 incorporate coriolis accelerations
c ---              (need to set icorio even if isphere=1)
c -->   ideep  = 0 shallow atmosphere approximation: special
c ---              for small spheres, re Wedi & Smolar QJR, 2009
c ---            1 (default) deep atmosphere
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

I/O

COSMO coupler

- **Mesoscale applications**

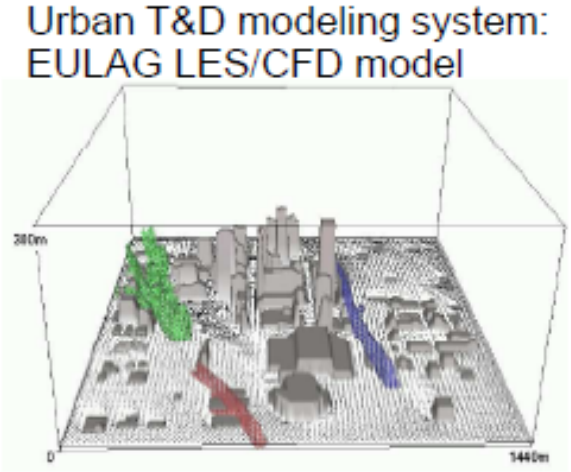


WRF coupler

- **LES apllications**
- **Small subset of WRF domian (5x5 grid points) drives LES BC**

Netcdf (serial/parallel) output and restart

- **tape.full**
- **tape.short**
- **tape.custom**



Predefined test cases

```
C=====================================================================
C NOW CHOOSE THE PREDEFINED TEST CASE (*) OR CREATE YOUR OWN EXPERIMENT (**)
C (*) UNCOMMENT THE PRESELECTED WORKING TEST CASE FROM THE FOLLOWING LIST
C  NOTE: ONLY THE LAST UNCOMMENTED LINE IS ACTIVE - SUBSTITUTE "C" with "#"
C (**) CREATE NEW ACTIVE TESTCASE NUMBER, REDEFINE ALL APPROPRIATE PARAMETERS
C=====================================================================
cdefine TESTCASE 1  /* SOLAR - Full MHD model in the future              */
cdefine TESTCASE 2  /* EARTH ... JW model imbalance                      */
cdefine TESTCASE 13 /* Held-Suarez test                                  */
cdefine TESTCASE 3  /* Exoplanets ... Zonal flow                         */
cdefine TESTCASE 4  /* Ocean model .....  not ready yet                  */
C---------------------------------------------------------------------*/
cdefine TESTCASE 5  /* PBL - CARTESIAN, CONVECTIVE                       */
cdefine TESTCASE 6  /* PBL - CARTESIAN, SHEAR DRIVEN/NEUTRAL             */
cdefine TESTCASE 7  /* PBL - CARTESIAN, STABLE                           */
C---------------------------------------------------------------------*/
cdefine TESTCASE 0  /* DEFAULT-CARTESIAN 2D OROGRAOPHICAL FLOW           */
cdefine TESTCASE 10 /* 3D real orographic flow: terrain-following/IMB    */
C---------------------------------------------------------------------*/
cdefine TESTCASE 20 /* NWP - COSMO COUPLING                              */
cdefine TESTCASE 21 /* LAM - WRF COUPLING                                */
cdefine TESTCASE 22 /* GCM - CAM COUPLING  .... not ready yet            */
C---------------------------------------------------------------------*/
cdefine TESTCASE 30 /* MICRO - rising buble in 2D ...                    */
cdefine TESTCASE 31 /* MICRO - DNS/chamber turbulence ... not ready yet  */
#define TESTCASE 32 /* MICRO - shallow covection (BOMEX/RICO/IMPACT)     */
#define TESTDATA  1 /* BOMEX  */
cdefine TESTDATA  2 /* RICO   */
cdefine TESTDATA  3 /* IMPACT */
cdefine TESTCASE 33 /* MICRO - deep convection ... not ready yet         */
C---------------------------------------------------------------------*/
cdefine TESTCASE 40 /* IMB - POROUS MEDIA .... not ready yet             */
cdefine TESTCASE 41 /* IMB - OBSTACLE/IDEL INPUT .... not ready yet      */
cdefine TESTCASE 42 /* IMB - URBAN FLOWS/POZNAN/IDEAL INPUT              */
cdefine TESTCASE 43 /* IMB - URBAN FLOWS/OKLAHOMA CITY/REAL SOUNDING     */
cdefine TESTCASE 44 /* IMB - URBAN FLOWS/OKLAHOMA CITY/WRF INPUT         */
cdefine TESTCASE 45 /* IMB - EULAG-D (DARCY) .... not ready yet          */
C---------------------------------------------------------------------*/
cdefine TESTCASE 11 /* TGV vortex 3D cyclic test                         */
cdefine TESTCASE 14 /* Convection over heated plane Piotrowskietal JCP`09  */
cdefine TESTCASE 15 /* Convection over heated sphere                     */
C---------------------------------------------------------------------*/
Cdefine TESTCASE 50 /* WAKE EDDIES                                       */
C#####################################################################
```

Predefined test cases

```
ccccccccccccccccccc
#if (TESTCASE == 15)
#define MOISTMOD 0
#define SGS 0
#define POLES 1
#define H_SUAREZ 0
#endif
ccccccccccccccccccc
#if (TESTCASE == 20)
#define SGS 1
#define POLES 0
#define PACTU 1
#define BCREF 1
#define VARENV 0
#define MOISTMOD 1
#endif
ccccccccccccccccccc
#if (TESTCASE == 30)
#define MOISTMOD 1
#define RADIAT 0
#define J3DIM 0
#define SGS 1
#endif
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     n,m,l  - model grid size
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
#if (TESTCASE == 0)   /* DEFAULT SETUP */
      parameter (n=256,m=1,l=125)
#endif
#if (TESTCASE == 1)
c      parameter (n=128,m=64,l=33) ! global code maximum l=40 because of rho
profile
      parameter (n=128,m=64,l=48) ! global code requires new rho profile
#endif
#if (TESTCASE == 2)
      parameter (n=128,m=64,l=47) ! global code
#endif
#if (TESTCASE == 3)
      parameter (n=128,m=64,l=181) ! global code
#endif
#if (TESTCASE == 5 || TESTCASE == 6)
      parameter (n=96,m=96,l=96) ! cartesian convective and shear driven PBL
#endif
#if (TESTCASE == 7)
      parameter (n=80,m=80,l=81) ! cartesian stable PBL
#endif
#if (TESTCASE == 11)
      parameter (n=64,m=64,l=64) ! Taylor-Green vortex
#endif
```

Predefined test cases

```
      do 4 k=1,l
      do 4 j=1,mp
      ja = (mpos-1)*mp + j
      do 4 i=1,np
      ia = (npos-1)*np + i
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
#if(TESTCASE == 0)
      u(i,j,k,0)=ue(i,j,k)
      v(i,j,k,0)=ve(i,j,k)
      w(i,j,k,0)=1.e-2*fz(i,j,k)
      th(i,j,k)=1.e-1*fz(i,j,k)
#endif
#if(TESTCASE == 1) /* convective layer only */
      u(i,j,k,0)=ue(i,j,k)
      v(i,j,k,0)=ve(i,j,k)
      w(i,j,k,0)=1.e2*fx(i,j,k)
      th(i,j,k) =1.e0*fx(i,j,k)    ! introduce theta perturbation
!     if (k.eq.1) th(i,j,k)=0.     !mihai
#endif
#if(TESTCASE == 2) /* IGEO=1 JW perturbation */
      u(i,j,k,0)=ue(i,j,k)
      coslmlc=cos(x(ia)-xlmbc)
      cosr=sinphic*sina(i,j)+cosphic*cosa(i,j)*coslmlc
      rpr=10.*acos(cosr)
      u(i,j,k,0)=u(i,j,k,0)+exp(-rpr**2)
      v(i,j,k,0)=ve(i,j,k)
      w(i,j,k,0)=1.e-2*fz(i,j,k)
       th(i,j,k)=1.e-1*fx(i,j,k)
#endif
```