Namelist "Numerical_Methods" (new version)

Not for the release SUNFLUIDH_EDU

This new data setup is also devoted to the selection of the numerical methods and schemes used in order to solve the conservation equations for velocity components, temperature, species mass fractions and density (in particuliar cases) and Poisson equation for the pressure. Some parameter setting could be also considered in respect with the numerical method selected. The data are divided in three groups in order to define:

- the numerical method applied for solving the conservation equations (for velocity, temperature, species, ...)
- the choice of advective or convective flux discretization (for 2nd order schemes only). The viscous, conductive or diffusive fluxes are always discretized with a centered 2nd order or 4th order scheme according to the previous choice.
- the numerical methods (and associated parameters) for solving the Poisson's equation according to the type of problem considered. Several methods are presented based on direct or iterative approaches.

- This version allows the user to select the numerical methods by means of characters strings instead of option numbers as in the old version Numerical_Methods (old version) .
- For the sake of clarity, the iterative method parameters used for solving the Poisson's equation are set in individual namelists :
    - HomeData_PoissonSolver for methods directly implemented in the code ("homemade" development)
    - HypreData_PoissonSolver for methods provided by the HYPRE library
- Click here to get some examples illustrating how to use the namelists "Numerical_Methods", "HomeData_PoissonSolver" or "HypreData_PoissonSolver"

Full data set of the namelist

```
 &Numerical_Methods  NS_NumericalMethod= "BDF2-Scheme02",
                     MomentumConvection_Scheme="Centered-02-Conservative" ,
                     TemperatureAdvection_Scheme="Centered-02-Conservative"
,
                     SpeciesAdvection_Scheme="Centered-02-Convective" ,
                     Poisson_NumericalMethod="Hypre-VariableMatrixCoef"  /
```

- Depending on the problem, some numerical methods are better adapted than others, this point is precised for each available option.
- The partial diagonalization method (Poisson_NumericalMethod = “Home-PartialDiagonalization”) used for solving the Poisson equation is only permitted for separable problems.
- The HYPRE library solvers for solving the Poisson's equation are only available if the HYPRE library has been installed.

Definition of the data set

Solving the conservation equations

NS_NumericalMethod

(equivalent to “Numerical_Scheme” in the previous release, see Numerical_Methods (old version))

- Type: character string
- Selection of the numerical scheme for solving the conservation equations :
  - “BDF2-SchemeO2” : 2nd order Backward Differentiation Formula for the time discretization. Semi-implicit scheme on the viscous or diffusion terms. 2nd order spatial discretization (centered for the viscous/diffusion terms, different schemes are available for the convection/advection terms - see further).
    Preferentially used for incompressible or low Mach-number flows without multi-species components (except two phase incompressible flows).
  - “CN-SchemeO2” : 2nd order Crank-Nicolson scheme (semi-implicit scheme on the viscous or diffusion terms. 2nd order spatial discretization (centered for the viscous/diffusion terms, different schemes are available for the convection/advection terms - see further).
    Preferentially used for incompressible or low Mach-number flows without multi-species components (except two phase incompressible flows). **Not for the release SUNFLUIDH_EDU** .
  - “BDF1-SchemeO2” : 1st order Backward Differentiation Formula. Semi-implicit scheme on the viscous or diffusion terms. 2nd order spatial discretization (centered for the viscous/diffusion terms, different schemes are available for the convection/advection terms - see further). **Not for the release SUNFLUIDH_EDU** .
  - “ExplicitPredCorO2-SchemeO2-Knio” : Explicit predictor-corrector scheme : 2nd order Adams Bashforth - Runge-Kutta scheme (Knio et al., JCP 1998).
    This scheme is mainly used for low Mach Number flows with multi-species components.
    Not for the release SUNFLUIDH_EDU .
  - “ExplicitPredCorO2-SchemeO2-Njam” : 2nd order Explicit scheme based on Strang's operator-splitting with a multi-timestep procedure on the viscous/diffusion terms (2nd order Adams Bashforth - Runge-Kutta schemes) The density is explicitly solved from the differential formulation of the equation of state and the Temperature is deduced from the

  equation of state. The projection method is based on the Chorin's formulation by considering Poisson's Operator with constant coefficients (Njam et al., JCP 1999).
  This scheme is mainly used for reactive flows under low Mach number hypothesis. **Not for the release SUNFLUIDH_EDU** .
  - “ExplicitPredCorO2-SchemeO2-Rixen” : 2nd order Explicit scheme based on Strang's operator-splitting with multistep a multi-timestep procedure on the viscous/diffusion terms (2nd order Adams Bashforth - Runge-Kutta schemes). The temperature is explicitly solved from the enthalpy equation and the density is deduced from the equation of state. The projection method is based on the Goda's formulation by considering Poisson's Operator with variable coefficients (Rixin Yu et al., JCP 2012). **Not for the release SUNFLUIDH_EDU** .
  - “ExplRKO3-CompactSchemeO4” : 4th order hermittian scheme in space + 3rd order explicit Runge-Kutta scheme in time. The projection method is based on the Goda's formulation by considering Poisson's Operator with constant coefficients (Knikker, ijnmf 2008, 2009).
    This numerical scheme is used for incompressible or low Mach number flows with or without heat transfer. **Not for the release SUNFLUIDH_EDU** .
  - “CN-CompactSchemeO4” : 4th order hermittian scheme in space + semi-implicit Crank-Nicolson scheme in time. The projection method is based on the Goda's formulation by considering Poisson's Operator with constant coefficients (Knikker, ijnmf 2008, 2009).
    This numerical scheme is used for incompressible or low Mach number flows with or without heat transfer. **Not for the release SUNFLUIDH_EDU** .
  - “RK3CN-CompactSchemeO4” : 4th order hermittian scheme in space + hybrid RK3/Crank-Nicolson scheme in time. The projection method is based on the Goda's formulation by considering Poisson's Operator with constant coefficients (Knikker, ijnmf 2008, 2009).
    This numerical scheme is used for incompressible or low Mach number flows with or without heat transfer. **Not for the release SUNFLUIDH_EDU** .
  - “CN-SchemeO2-SpecialLowMachFlow”: Crank-Nicolson semi-implicit scheme (as the option 2) with a predictor-corrector procedure for solving species mass fractions and temperature. This numerical scheme is used for low Mach number flows. **Not for the release SUNFLUIDH_EDU** .

MomentumConvection_Scheme

(equivalent to “Convective_Flux_Discretization_Type” in the previous release, see Numerical_Methods (old version))

- Type : character string
- Selection of the 2nd order spatial discretization for the convection flux in the momentum equations. The options are :
  - “Centered-O2-Conservative” : 2nd order centered scheme in the conservative formulation.
  - “Centered-O2-Convective” : 2nd order centered scheme in the advective formulation.
  - “Centered-O2-Skewsymmetric”: 2nd order centered scheme in the skew-symemetric formulation (developer's test).
  - “Upwind-O1-Conservative” : not commented (developer's test)
  - “Quick-O2-Conservative” : QUICK scheme in the conservative formulation. **Not for the release SUNFLUIDH_EDU** .
  - “Upwind-O2-Conservative” : 2nd order Upwind scheme in the conservative formulation. **Not for the release SUNFLUIDH_EDU** .

- "Eno-O2-Conservative" : 2nd order ENO scheme in the conservative formulation (in progress). **Not for the release SUNFLUIDH_EDU** .

TemperatureAdvection_Scheme

(equivalent to "Temperature_Advective_Flux_Discretization_Type" in the previous release)

- Type : character string
- Selection of the 2nd order spatial discretization for the advection flux in the temperature/enthalpy equation. The options are the same as previously.

SpeciesAdvection_Scheme

(equivalent to "Species_Advective_Flux_Discretization_Type" in the previous release)

- Type : character string. **Not for the release SUNFLUIDH_EDU** .
- Selection of the 2nd order spatial discretization for the advection flux in the species equations. The options are the same as previously.

Explicit_Solving_of_Density

- Type : integer value. **Not for the release SUNFLUIDH_EDU** .
- Selection of numerical schemes based on flux limiters with TVD properties for solving the density. Some of proposed methods are "experimental" and they must be used with caution. The options are :
  - 0 : The mass equation is not solved. For low Mach number flows, the density is deduced from the equation of state. This supposes that species mass fractions or/and temperature are calculated from their conservation equations if it is required.
    - 1, 2, 4, 5 : Obsolete methods
    - 3 : Solving the density from the differential equation of state. This option is automatically selected when the numerical scheme of Njam et al. is used (Numerical_Scheme=5)
    - 6 : The mass equation is solved with the Lax-Wendroff + Superbee TVD scheme (test).
    - 7 : The mass equation is solved with the Lax-Wendroff + SuperC TVD scheme (test).
    - 8 : The mass equation is solved with the Lax-Wendroff + HyperC TVD scheme (test).
    - 9 : The mass equation is solved with the Lax-Wendroff + Van Leer TVD scheme (test).
    - 10 : The mass equation is solved with the Lax-Wendroff + a TVD scheme defined by the user (test).
    - 11 : The mass equation is solved with a WENO5 scheme (test).

Solving the Poisson's equation

Poisson_NumericalMethod

(equivalent to “Numerical_Method_Poisson_Equation ” in the previous release, see Numerical_Methods (old version))

- Type : Character string
- Selection of the numerical method for solving the Poisson equation in accordance to the projection method. The solution is the pressure time increment ($\Phi= P^{n+1}-P^{n}$, Goda's method) used to update the velocity field according to the principle of the projection method (When the numerical method of Njam et al. is used to solve the Navier-Stokes equations, the pressure is solved in place of its time increment (Chorin's method). The options are :
  - “Home-SORMultigrid-ConstantMatrixCoef” : Successive Over-Relaxed method (SOR) coupled with a nV-cycle multigrid method in order to accelerate the convergence. The matrix elements depend on the cell size only (constant elements). This method is directly implemented in the code (“homemade” development). No external library is required to use it. The associated parameters are set in the namelist HomeData_PoissonSolver .
  - “Home-HelmholtzApproximation” : Poisson's operator (constant matrix elements) is approximated by a Helmholtz's operator (experimental method proposed by J.L. Guermond) - For incompressible flow only. This method is directly implemented in the code (“homemade” development). No external library is required to use it. No parameter setting is needed.
  - “Home-PartialDiagonalization” : Partial diagonalisation of the Laplacian operator (constant elements). BEWARE : the problem must be separable. This method is directly implemented in the code (“homemade” development). No external library is required to use it. No parameter setting is needed.
  - “Home-SORMultigrid-VariableMatrixCoef” : SOR iterative method coupled with a multigrid method in order to accelerate the convergence. The matrix elements depend on the cell size and the density, the source term is defined from the divergence of the velocity. This method is directly implemented in the code (“homemade” development). No external library is required to use it. The associated parameters are set in the namelist HomeData_PoissonSolver .
  - “Hypre-ConstantMatrixCoef” : Poisson's equation is solved by the HYPRE library tools. The matrix elements are constants. The parameters of the HYPRE library are set in the namelist HypreData_PoissonSolver .
  - “Hypre-VariableMatrixCoef”: Poisson's equation is solved by the HYPRE library tools. The matrix elements are non-constants. The parameters of the HYPRE library are set in the namelist HypreData_PoissonSolver .

Simulation control

We here resort to a specific namelist named “Simulation_Management. It is also used in the next section “Data acquisition”. We specify here some parameters in order to define the numerical time step as well as stop criteria and recording rates related to backup and check files. Two examples are given. The first one corresponds to a simulation starting at t= 0 with a variable time step.

```
 &Simulation_Management Restart_Parameter= 0                            ,!--- Option value for starting the simulation from t=0.
       Steady_Flow_Stopping_Criterion_Enabled = .false. ,!--- Stop criterion for steady flows. When it is enabled, residues between two successive flow fields are computed
       Steady_Flow_Stopping_Criterion = 1.D-20          ,!--- convergence tolerance threshold for a steady flow solution (it works only when the previous parameter is enabled)
       Temporal_Iterations_Number = 10                  ,!--- maximum value of time iterations before stopping the computation
       Final_Time = 3.D+01                              ,!--- Maximum value of time before stopping the computation
       TimeStep_Type = 1 ,                              ,!--- Option value for specifying a variable time-step computed from a CFL criterion
       CFL_Min = 0.05                                   ,!--- Minimum value of the CFL criterion imposed by the user when the simulation starts
       CFL_Max = 0.4                                    ,!--- Maximum value of the CFL criterion imposed by the user after n time iterations (here n=100, see the next parameter)
       Iterations_For_Timestep_Linear_Progress= 100     ,!--- Number of time iterations over which the CFL criterion increase linearly between CFL_Min and CFL_Max
       Simulation_Backup_Rate                 = 1000    ,!--- Recording rate (in time-iteration units) for generating backup files (save_fld_xxxxx_y.d , save_var_xxxxx_y.d)
       Simulation_Checking_Rate               = 200     /!--- Recording rate (in time-iteration units) for writing some relevant check data in a file checkcalc_xxxxx.d
```

The second example corresponds to a restart of the previous simulation with a uniform time step.

```
&Simulation_Management Restart_Parameter= 3             ,!--- Option value for resuming the simulation from the end of a previous computation.
       Steady_Flow_Stopping_Criterion_Enabled = .false. ,!--- Stop criterion for steady flows. When it is enabled, residues between two successive flow fields are computed
       Steady_Flow_Stopping_Criterion = 1.D-20          ,!--- convergence tolerance threshold for a steady flow solution (it works only when the previous parameter is enabled)
       Temporal_Iterations_Number = 1000                ,!--- maximum value of time iterations before stopping the computation
       Final_Time = 6.D+01                              ,!--- Maximum value of time before stopping the computation
       TimeStep_Type = 0 ,                              ,!--- Option value for specifying a constant time-step
       Timestep_Max = 1.e-3,                            ,!--- Value of the time step
       Iterations_For_Timestep_Linear_Progress= 100     ,!--- Number of time iterations over which the CFL criterion increase linearly between CFL_Min
```

```
and CFL_Max
       Simulation_Backup_Rate                  = 1000    ,!--- Recording rate
(in time-iteration units) for generating backup files (save_fld_xxxxx_y.d ,
save_var_xxxxx_y.d)
       Simulation_Checking_Rate                = 200     /!--- Recording rate
(in time-iteration units) for writing some relevant check data in a file
checkcalc_xxxxx.d
                         Fields_Recording_Rate = 1.D+00 ,
                         Probe_Recording_Rate                  = 10     ,
                         Start_Time_For_Statistics= 1.D+2              ,
                         Time_Range_Statistic_Calculation = 1.D+00  /
```

For more information on this data set, click here.

Keep in mind the time step must be chosen with caution because it can generate numerical instabilities when it is too much large. The numerical stability depends on the property of the numerical methods used for solving the conservation equations It often relies on the CFL criterion which have not to exceeded a reference value. This value depends on the numerical scheme properties as well as the computational problem.

- For semi-implicit schemes proposed here, a maximum CFL-value about 0.5 is generally prescribed for usual computations, but it could be smaller for problems with strong gradients.
- For explicit schemes, the CFL criterion also depends on the viscous/diffusive time scales as well as the space dimension of the problem. As a consequence, the CFL value prescribed is generally between 0.5^{n-1} and 0.5^n, where n is the dimension of the problem.

When the time-step value is constant, the user can verify if the CFL criterion is respected by checking regularly the file checkcalc_xxxxx.d