Numerical Methods

We remind the user the equations are solved with a projection method. The numerical methods used in the code SUNFLUIDH are commented in the documents for the full version or documents for the educative version.

For solving the conservation equations for velocity and temperature, we choose a 2nd order scheme in space and time.

- The spatial discretization for the convective flux is a 2nd order centered scheme (conservative form)
- The spatial discretization for the viscous/conductive terms is also a 2nd order centered scheme (this way is imposed, no choice is possible)
- The time discretization is a 2nd order Backward Differentiation Formula (semi-implicit method build on the viscous/conductive terms)

The Poisson's equation is solved a Successive Over-Relaxation method coupled with a nV-cycle multigrid method for accelerating the convergence. As the flow is incompressible, a solver version with constant coefficient matrix can be selected.
The data setup is made with the following namelists :

```
  &Numerical_Methods  NS_NumericalMethod= "BDF2-SchemeO2"
,  !--- BDF2 + 2nd order centered scheme
                      MomentumConvection_Scheme="Centered-O2-Conservative"
,  !--- conservative form for solving the velocity (momentum) equation
                      TemperatureAdvection_Scheme="Centered-O2-Conservative"
,  !--- conservative form for solving the temperature (enthalpy) equation
                      Poisson_NumericalMethod="Home-SORMultigrid-
ConstantMatrixCoef"  / !--- SOR  + multigrid method (homemade release) for
solving the Poisson's equation with constant coefficient matrix
```

As the used solver is a “homemade” release (directly implemented in the code SUNFLUIDH without invoking any external scientific library), the parameters of the SOR + multigrid method are set in the namelist

```
   &HomeData_PoissonSolver  SolverName="SOR-Redblack",!--- Successive Over-
Relaxation (SOR) method based on the red-black algorithm
                            Relaxation_Coefficient= 1.7 , !--- Relaxation
coefficient of the SOR method ( 1 <= Relaxation_Coefficient < 2)
                            Number_max_Grid= 5,       !--- Number of grid
levels
                            Number_max_Cycle= 10,     !--- Number of
multigrid cycles
                            Number_Iteration= 0,      !--- Maximum number of
SOR iterations method applied for any grid level, if 0 (or removed) the 3
next data are considered
                            Number_Iteration_FineToCoarseGrid= 3,  !---
number of SOR iterations applied on any grid level during the restriction
step (before the coarsest grid computation)
                            Number_Iteration_CoarseToFineGrid= 15, !---
```

```
number of SOR iterations applied on any grid level during the prolongation
step (after the Coarsest grid computation)
                            Number_Iteration_CoarsestGrid= 15 ,    !---
number of SOR iterations applied on the coarsest grid
                            Convergence_Criterion= 1.D-08 /        !---
convergence tolerance on the residu of the Poisson's equation
```

The SOR-multigrid method is efficient but the tuning of parameters can be delicate. Some tips are presented in order to help the user :

- The maximum number of grid level is conditioned by the cell number of the finnest grid. The grid coarsening consists in dividing by two the cell number along each direction. The minimum number of cells on the coarsest grid is two per direction.
- The relaxation coefficient can improve the convergence efficiency but beware of the numerical instabilities. The choice of the relaxation coefficient depends on the problem. The maximum value are about 1.8 and the minimum value is 1.
- The red-black algorithm is able to keep the same solution accuracy in parallel computing regardless of the domain decomposition. It can be seen as the combination of Jacobi and Gauss-Seidel methods. As the consequence, the convergence is less efficient than a pure SOR method (or Over Relaxed Gauss-Seidel method) which doas not preserve the solution accuracy in parallel computing in regard to the domain decomposition.
- It could be useful to set the number of SOR iterations to different values for each part of a V-cycle of the multigrid method in order to reduce the total number of operations. Some numerical experiments are shown that only few SOR iterations could be sufficient during the restriction step. A fine setting of the iteration number can improve the performance of the code.
- Beware of the maximum number of multigrid cycles. more larger is this number more longer is the CPU time consuming if the convergence is too slow. Its maximum value is generally approximately 10. Beyond this value, it could be useful to try something else.
- Do not forget the purpose on solving the Poisson's equation : ensure a divergence-free velocity for incompressible flows or the mass preserving in low Mach number hypothesis. This goal is sometimes reached without solving the Poisson's equation with a too much small convergence tolerance…

Other numerical methods are available for solving the conservation equations as well as the Poisson's equation. For more details about them click here.