freeCappuccino input file specification guide

January 14th 2023

Description of input parameters and their default values, possible values, description, discussion on status of the code related to it, etc.

The description below refers to fortran namelist file e.g. 'input.nml' that is required to configure the simulation. Not all the parameters have to be present, nor do they have to be in specific order.

Parameter key	Values	Description
title	String e.g. 'Laminar flow in cavity'	Descriptive name of the case which will be written in monitor file.
mesh_format	'nativeMesh' 'foamMesh'	Mesh format 'nativeMesh' for native polyMesh format or 'foamMesh' for OpenFOAM polyMesh.
lread	T/F	Read restart file - continue simulation from saved state?
ltest	T/F	Verbosity for linear solver convergence (for troubleshooting).
calcU	T/F	Activate Velocity field calculation?
urfU	Real array, size(3), e.g. 0.7, 0.7, 0.7	Under-relaxation factor for momentum equations.
gdsU	Real, e.g. 1.0	gamma-deferred correction parameter for velocity.
cSchemeU	String e.g. 'linearUpwind'	Convection scheme for momentum equation.
dSchemeU	String, e.g. 'skewness'	Diffusion scheme for momentum equation

(continued next page)

(from previous page)

Parameter key	Values	Description
nrelaxU	Int, e.g. 1	Integer parameter for diffusion scheme - advanced
lSolverU	Sting, e.g. 'bicgstab'	Linear algebraic solver for momentum equation
maxiterU	Int, e.g. 10	Max no of iterations for linear U-V-W equations
tolAbsU	Real, e.g. 1e-13	Absolute tolerance level for residual for linear U-V-W equations
tolRelU	Real, e.g. 0.025	Relative tolerance level for residual for linear U-V-W equations
pscheme	Sting, e.g. 'linear'	Interpolation of pressure to faces. Possible choices are: 'linear'(default), 'weighted', 'central'.
calcP	T/F	Activate Pressure field calculation? True/False.
urfP	Real, eg. 0.3	Under-relaxation factor for pressure
lSolverP	Sting, e.g. 'iccg'	Linear algebraic solver for pressure/pressure correction
maxiterP	Int, e.g. 50	Max no of iterations for pressure/pressure correction
tolAbsP	Real, e.g. 1e-13	Absolute tolerance level for residual for pressure/pressure correction

(continued next page)

(from previous page)

Parameter key	Values	Description
tolRelP	Real, e.g. 0.025	Relative tolerance level for residual for pressure/pressure correction
TurbModel	String, e.g. 'none', 'k_epsilon_std'	Turbulence model - see discussion at the end of the document.
TurbModel%Scalar(n)%urf	Real, e.g. 0.7	Under-relaxation factor, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%gds	e.g. 1.0	Deferred correction factor, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%cScheme	String, e.g. 'linearUpwind'	Convection scheme - default is second order upwind, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%dScheme	String, e.g. 'skewness'	Diffusion scheme, i.e. the method for normal gradient at face uncorrected/skewness/offset, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%nrelax	Int, e.g. 0	Relaxation parameter non-orthogonal correction for face gradient, {-1/0/1}, , $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%lSolver	Sting, e.g. 'bicgstab'	Linear algebraic solver, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%maxiter	Int, e.g. 20	Max number of iterations in linear solver, $n \in \{1, ..., 8\}$..
TurbModel%Scalar(n)%tolAbs	Real, e.g. 1e-13	Absolute residual level, $n \in \{1, ..., 8\}$.
TurbModel%Scalar(n)%tolRel	Real, e.g. 0.01	Relative drop in residual to exit linear solver, $n \in \{1, ..., 8\}$.
TurbModel%urfVis	Real, e.g. 1.0	Under-relaxation for eddy-viscosity.

(continued next page)

(from previous page)

Parameter key	Values	Description
calcVis	T/F	Activate dynamic viscosity recalculation (Non-Newtonian flows)
non_newtonian_model	String, e.g. 'PowerLaw'	There are many available models, see discussion at the end of the document.
npow	Real, e.g. 0.4	Exponent for power-law fluids
Consst	Real, e.g. 10.0	Consistency index for power-law fluids
shearmin	Real, e.g. 1e-3	Lower limit for the shear rate magnitude for power-law fluids
Tau_0	Real	Yield stress
megp	Real	Exponential growth parameter
muplastic	Real	Plastic viscosity
muzero	Real	Zero shear rate viscosity
muinfty	Real	Infinity shear rate viscosity
lamtime	Real	Natural time - time parameter
urfVis	Real, e.g. 0.8	Under-relaxation parameter
calcT	T/F	Activate Temperature equation calculation? True/False.
urfT	Real, e.g. 0.7	Under-relaxation factors.
gdsT	Real, e.g. 1.0	Deferred correction factor.

(continued next page)

(from previous page)

Parameter key	Values	Description
cSchemeT	String, e.g. 'linearUpwind'	Convection scheme - default is second order upwind.
dSchemeT	String, e.g. 'skewness'	Diffusion scheme, i.e. the method for normal gradient at face skewness/offset.
nrelaxT	Int, e.g. 0	Relaxation parameter non-orthogonal correction for face gradient. Advanced.
lSolverT	String, e.g. 'bicgstab'	Linear algebraic solver.
maxiterT	Int, e.g. 20	Max number of iterations in linear solver.
tolAbsT	Real, e.g. 1e-15	Absolute residual level.
tolRelT	Real, e.g. 0.01	Relative drop in residual to exit linear solver.
sigt	Real, e.g. 0.9	sigma_t - Prandtl-Schmid number for temperature.
calcEn	T/F	Activate Energy field calculation? True/False.
urfEn	Real, e.g. 0.9	Under-relaxation factors.
gdsEn	Real, e.g. 1.0	Deferred correction factor.
cSchemeEn	String, e.g. 'linearUpwind'	Convection scheme - default is second order upwind.
dSchemeEn	String, e.g. 'skewness'	Diffusion scheme, i.e. the method for normal gradient at face skewness/offset.

(continued next page)

(from previous page)

Parameter key	Values	Description
nrelaxEn	Int, e.g. 0	Relaxation parameter non-orthogonal correction for face gradient.
lSolverEn	String, e.g. 'bicgstab'	Linear algebraic solver type (see list below).
maxiterEn	Int, e.g. 20	Max number of iterations in linear solver.
tolAbsEn	Real, e.g. 1e-15	Absolute residual level.
tolRelEn	Real, e.g. 0.01	Relative drop in residual norm to exit linear solver.
sigtEn	Real, e.g. 0.9	Prandtl-Schmid number for energy equation.
solveTotalEnergy	T/F	What we solve here - default is total energy, but other options are...
solveInternalEnergy	T/F	internal energy, or...
solveEnthalpy	T/F	enthaply.
addViscDiss	T/F	Add viscous dissipation term? True/False.
lbuoy	T/F	Buoyancy activated? True/False.
boussinesq	T/F	Bousinesq approximation for buoyancy.
tref	Real, e.g. 300.0	Reference temperature for buoyant flows.
gravx	Real, e.g. 0.0	Three components of gravity vector.
gravy	Real, e.g. -9.81	-
gravz	Real, e.g. 0.0	-

(continued next page)

(from previous page)

Parameter key	Values	Description
calcCon	T/F	Activate passive scalar concentration field calculation? True/False.
urfCon	Real, e.g. 0.7	Under-relaxation factors.
gdsCon	Real, e.g. 1.0	Deferred correction factor.
cSchemeCon	String, e.g. 'linearUpwind'	Convection scheme - default is second order upwind.
dSchemeCon	String, e.g. 'skewness'	Diffusion scheme, i.e. the method for normal gradient at face skewness/offset.
nrelaxCon	Int, e.g. 0	Type of non-orthogonal correction for face gradient minimal/orthogonal/over-relaxed. 1/0/-1.
lSolverCon	String, e.g. 'bicgstab'	Linear algebraic solver.
maxiterCon	Int, e.g. 20	Max number of iterations in linear solver.
tolAbsCon	Real, e.g. 1e-15	Absolute residual level.
tolRelCon	Real, e.g. 0.01	Relative drop in residual to exit linear solver.
sigCon	Real, e.g. 0.9	Prandtl-Schmidt number for passive scalar concentration
calcEpot	T/F	Activate Electric potential field calculation? True/False.
urfEpot	Real, e.g. 0.8	Under-relaxation factor.

(continued next page)

(from previous page)

Parameter key	Values	Description
gdsEpot	Real, e.g. 1.0	Deferred correction factor.
lSolverEpot	Sting, e.g. 'iccg' ,	Linear algebraic solver.
maxiterEpot	Int, e.g. 20	Max number of iterations in linear solver.
tolAbsEpot	Real, e.g. 1e-15	Absolute residual level.
tolRelEpot	Real, e.g. 0.01	Relative drop in residual to exit linear solver.
densit	Real, e.g. 1.0	Fluid density.
viscos	Real, e.g. 1e-2	Molecular dynamic viscosity.
pranl	Real, e.g. 0.71	Prandtl coefficient for specific fluid.
beta	Real, e.g. 1e-3	Thermal expansion coefficient.
lsgdh	T/F	Simple gradient hypothesis for heat-fluxes, or...
lggdh	T/F	Generalized gradient hypothesis for heat fluxes, or...
lafm	T/F	Algebraic flux modelling for heat fluxes.
facnap	Real, e.g. 1.0	Under-relaxation factor for Reynolds stresses.
facflx	Real, e.g. 1.0	Under-relaxation factor for heat fluxes.
ltransient	T/F	Unsteady simulation True/False and chose ONE algorithm below

(continued next page)

(from previous page)

Parameter key	Values	Description
bdf	T/F	Backward-Euler; First-Order Implicit, or...
bdf2	T/F	Second-Order Backward Euler; Second-Order Implicit, or...
bdf3	T/F	Third-order backard, or...
CN	T/F	Crank-Nicolson.
autotime	T/F	Automatic timestep for pseudotransient runs.
lenscale_option	String	Length scale option for pseudotransient runs: 'conservative' or 'aggressive'.
lstsq	T/F	Gradient approximation, chose ONE: Unweighted Least-Square gradient approximation, or...
lstsq_qr	T/F	Least square gradients based on thin QR decomposition, or...
lstsq_dm	T/F	Distance-square weighted version of Least-Square Gradient, or...
node_gauss	T/F	Node based Gauss gradient with pseudolaplacian interpolation.
gauss	T/F	Cell based Gauss gradient with simple linear interpolation.
nigrad	Int, e.g. 1	Number of iterations for Gauss gradient approximation.

(continued next page)

(from previous page)

Parameter key	Values	Description
limiter	String, e.g. 'Venkatakrishnan'	Gradient limiter - global for all gradients (none, Barth-Jespersen, Venkatakishnan, multidimensional, R3)
SIMPLE	T/F	Pressure-velocity coupling method - SIMPLE, or...
PISO	T/F	Pressure-velocity coupling method - PISO.
ncorr	Int, e.g. 1	Number of PISO corrections - only relevant for PISO.
npcor	Int, e.g. 1	Number of iterations for pressure/pressure correction equation - Number of Nonorthogonal corrections.
pRefCell	Int, e.g. 1	Reference cell for setting pressure level (since we have pure Neumann problem)
tolerance	Real, e.g. 1e-5	Desired level of convergence for SIMPLE iterations.
numstep	Int, e.g. 1000	Total number of timesteps.
timestep	Real, e.g. 1e-3	Timestep size - also known as dt.
nzapis	Int, e.g. 100	Program writes output files every NZAPIS timesteps.
maxit	Int, e.g. 1	Number of iterations in SIMPLE or PISO sequential equation solution loop.

(continued next page)

(from previous page)

Parameter key	Values	Description
CoNumFix	T/F	Adaptive timestep size based on target Courant number? True/False.
CoNumFixValue	Real, e.g. 1.0	If CoNumFix=True then set target maximum Courant number here.
const_mflux	T/F	Do we have constant mass flow in the domain? True/False.
magUbar	Real, e.g. 1.0	If const_mflux=True then set target bulk velocity for constant mass flow situation.

Possible **linear algebraic solvers** are: 'gauss_seidel', 'dpcg', 'iccg', 'bicgstab'. Most computational effort goes into solution of pressure Poisson equation, which is symmetric, positive definite. Use Incomplete Cholesky Conjugate Gradient (ICCG) for that one. For other transport equations with non-symmetric matrices Bi-CGStab is good option. Using LIS library enables greater choice for precodnitioner/solver combinations, when necessary (eg. '-i cg -p saamg -tol 0.01 -maxiter 10' for CG with Smoothed Agglomeration Algebraic Multigird preconditioner with relative residual norm reduction target of 0.01 and maximum of 10 iterations). For LIS solvers all solver parameters, as seen in this example, are given within single text string.

Possible **convection schemes** are: 'cds', 'central', 'linearUpwind', kappa, muscl, umist, smart, boundedLinearUpwind, boundedCentral, etc. The list is long and you may consult the source code, just look at the 'interpolation.f90' module. If you use 'linearUpwind' it is good to have gradient (slope) limiter activated, such as 'Barth-Jespersen', 'Venkatakrishnan', 'multidimensional' or 'R3'.

The **rheology** may be prescribed using the various non-Newtonian models. If we are interested in materials exhibiting such a behaviour we may set 'calcVis' to true, to recalculate the effective viscosity at each iteration, and choose an appropriate Non-Newtonian model. The options are: 'PowerLaw' (Power law constitutive Model), 'HerschelBulkley' (Herschel-Bulkley constitutive regularized (Papanastasiou) Model), 'Bingham' (Bingham model), 'BinghamPapanastasiou' (Bingham-Papanastasiou model), 'Carreau' (Carreau constitutive model), 'CarreauYasuda' (Carreau-Yasuda constitutive model), 'Casson' (Casson constitutive model), 'CassonPapanastasiou' (Casson constitutive regularized (Papanastasiou) Model with consistency temperature dependance), 'CrossModel' (Cross constitutive model).

Choosing **turbulence model** is of course a big topic. Here is the short list of possibilities. Eddy viscosity RANS models are: 'k_epsilon_std', 'k_omega_sst', 'Spalart_Allmaras', 'k_epsilon_rng',

'k_epsilon_rlzb' (Realizable $k-\epsilon$ model), 'k_epsilon_rlzb_2lewt' (Realizable $k-\epsilon$ model with two-layer approach for wall cells and with enhanced wall functions using Reichardt blending function), 'k_epsilon_std_2lewt' (Standard $k-\epsilon$ with two layer approach and enhanced wall functions). The LES models are 'WALE', and one equation 'k_eqn_eddy'. The hybrid models are: 'IDDES_k_omega_sst', 'DDES_k_omega_sst'. Many more to come soon!
The **TurbModel** is a variable of derived data type. In the namelist files it is initialized by a list of data, delimited using comma, representing values of the data members in sequential order as they are listed in definition of the derived data type.
The 'TurbModel%name' is first in the sequence so if we put 'TurbModel = k_omega_sst', it will only initialize the first data member of the derived type, which is the model name. This is fortunate circumstance and leads to compact initialization style. Turbulence model may have several (we often deal with two) equations which define the model, each for a separate scalar field. Example is $k-\epsilon$ model, where we solve separate equations for turbulence kinetic energy and turbulence kinetic energy dissipation rate. For every field we have in the model, we may set a value for any available parameter as seen in the table above. This can be done in any order, it is irrelevant. For example writing 'TurbModel%Scalar(1)%urf = 0.5' will set the under-relaxation factor for the first scalar field in the turbulence model to 0.5. 'TurbModel%Scalar(1)%cScheme = 'boundedLinearUpwind" will set the interpolation scheme for convection different from the default one ('linearUpwind') for the first scalar field in the model.