

DRAVVYA



SIMULATIONS and ENGINEERING

Pvt. Ltd.

25th July 2018

Web www.dravvya.co.in
Email dravvya@dravvya.co.in
Phone [+81 \(82\) 831-1190](tel:+81(82)831-1190)

WHY Wildkatze Solver? Numerics

Following document explains the general numerical methods and procedure available to physics models.

Linear Solvers

Wildkatze solver provides user with various options for solving linear systems. Some of these options are not available in other commercial solvers at the time of writing this document.

Linear solvers play cardinal role in stability of solution. Wildkatze provides very robust linear solvers due to this reason.

Algebraic Multigrid Solvers

Additive-corrective multigrid method is provided. In this method fine level equations are grouped into coarse level equations. The prolongation of correction is equal from coarse equation to fine level equations. Similarly the restriction is constructed by summing the residuals of fine level equations.

Smoothers

Standard Gauss Seidel, Diagonal ILU smoothers are providing along with special options of Chebyshev Polynomial and Auto-Smoother as shown here as an

example from simtree file:

```
continuity-amg-smoother    gauss-seidel    0    4    gauss-
seidel    cheby-poly    d-ilu    auto-smoother User
```

Chebyshev Polynomial

It is well known that when the number of processors increase the performance of smoother as Gauss-Seidel drops to the point that simulation can even diverge. To address this problem Polynomial smoother is provided. Polynomial smoothers depend on matrix vector product and thus, their performance does not drop when number of processors is increased.

The implemented Chebyshev Polynomial smoother is described in paper “**Parallel multigrid smoothing: polynomial versus Gauss-Seidel**”, from authors Mark Adams, Marian Brezina, Jonathan Hu and Ray Tuminaro (Journal of Computational Physics, Volume 188, Issue 2, 1 July 2003).

Auto-Smoother

In industrial simulations it is often not possible to generate ideal meshes. Mesh quality decides stability and accuracy of simulation. For the purpose of accuracy one shall generate the mesh with highest quality possible. However since good quality mesh isn't always possible to generate, convergence and accuracy issues cannot be avoided when solving CFD problems.

Many times, problems in mesh quality cause linear solver failure, due to appearance of **zero on matrix diagonal** or failure to satisfy Scarborough criteria (diagonal dominance). For such situations Auto-Smoother is suggested.

As name suggests Auto-Smoother is automatic in the sense that it checks at each coarse level for convergence. **If divergence is detected on any coarse level then it may drop the smoother completely and will try to solve again with some other robust method.**

Due to this reason Auto-Smoother is expensive but it can work with severely degenerate linear systems. User shall use it with care.

Note: The diagonal at any coarse level will never become 0 in Wildkatze solver's linear system. This is taken care of by 0 being replaced by suitable non-zero value without affecting the accuracy of solution. Diagonal 0 problems are reported in other commercial solvers and for this reason this solution is provided as preemptive measure to avoid any such possibilities.

Krylov Solvers

Conjugate Gradient and Bi-Conjugate Gradient solvers are also available. They use chosen smoother as preconditioner.

Gradient Computation

Gradient computation plays a very important role in accuracy and stability of the solution. Following methods are available for gradients calculation:

- **Green Gauss**

Standard and default option that is useful for most of the situations. This option's accuracy is affected by mesh quality.

- **Weighted Least Square**

Gradient computation is done based on solving least square problem for gradients for a cell center using this cell's immediate neighbors. The weight is based on inverse distance weighting.

- **Least Square - Green Gauss**

Weighted Least Square method is much more accurate compared to Green Gauss method and shows much more improved performance when mesh quality degrades. However the standard method still suffers from mesh quality problems. To address this and to improve the accuracy further a hybrid method is developed in Wildkatze solver.

In this method, the closest point to face center on cell to cell center line is located. The value of variable is interpolated just as Green Gauss method. As it is known that due to skew, this value is not accurate. Therefore using the neighbors of cells on both side of face (left and right cell), a weighted least square problem (Least Square method as mentioned above) is solved. Now having obtained gradient at this point, the value at face center is interpolated.

This method shows much improved accuracy compared to both above methods, but at the expense of solving least square problem at every face point. Thus this method is quite expensive.

Note: Unlike many other CFD softwares, Wildkatze lets user to choose different Gradient computation method for each physics model. User can freely chose Green Gauss method for one set of physics model, while using Weighted Least Square method for another set of physics model.

Time Stepping Methods

Solver can be set to transient simulation by changing in Wildkatze **simulation-type** option from Steady to Transient. Two modes of transient simulation are provided:

- **Transient**

Implicit transient method is provided with two options: **Implicit Euler** and **Three Level Second Order** method. This method runs fixed number of inner iterations per time-step.

- **Transient-Fractional**

Fractional solver is provided for shortening the simulation time. This option normally has shown to drop calculation time by two to three times over Transient option.

Transient-Fractional solver is a combination of Fractional and Transient iterative methods. In this method solver will be run with 'Transient' option at certain frequency specified by the user. Typically this frequency is 5 to 10.

The idea behind hybrid solver comes from the fact that though fractional solvers simulations are fast, but they tend to show instability for complicated and difficult problems, while iterative solvers are stable. By solving implicit step after certain frequency the solver shows much greater stability and thus making it usable for difficult simulations too.

- **Variable Time Stepping**

The simulations with multiphase models use explicit methods forcing the time stepping to be variable and based on Courant number. This could be activated by setting option '**variable-time-step**' to 1. In this situation the minimum and maximum time-step is controlled by setting options 'minimum-time-step' and 'maximum-time-step'.

Transient and Transient-Fractional options are available for variable time stepping too.

Motion

Wildkatze provides motion in three forms:

- **Moving Reference Frame**

The effect of rotational motion in steady state mode can be realized by use of Moving Reference Frame (MRF) model.

- **Rigid Body Motion**

Sliding mesh problems could be done using Rigid Body Motion. One shall note that Wildkatze supports Grid-Grid Interface (GGI) for these types of motions as well as for steady state simulations.

- **Immersed Boundary Method**

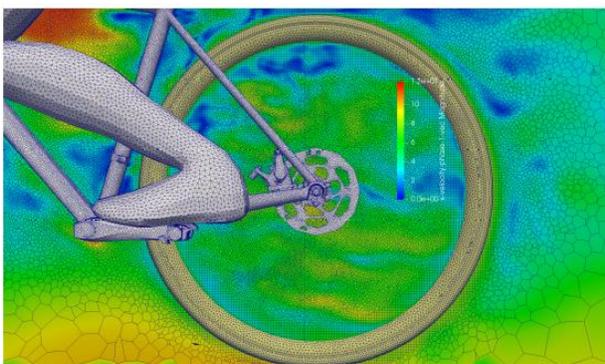
Immersed Boundary Method is available as a framework to the main solver for motion (transient) as well as steady simulation.

The Immersed Region does not need to be moving, it could be stationary and marked once. Hence, it is very easy to add or remove objects from the simulation using this method.

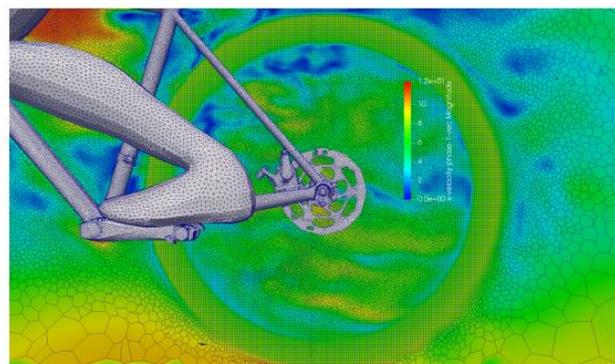
Immersed Boundary Methods could support various types of motions which are difficult to simulate with other mesh motion technics, like morphing or chimera grid methods.

Wildkatze's **fast and robust solid marking algorithm** allows one to use this method without much penalty on simulation time. It is a **parallelized feature which works robust on real cases** as seen in image below.

As Immersed Boundary Method example, consider the following motion of bike tire on road. This type of motion is very easy to simulate with Immersed Boundary Method of Wildkatze.



Tire motion with Immersed Boundary



Immersed Region with interpolated velocity field
