A block coupled solver development for hydraulic machinery applications

L Mangani, M Buchmayr and M Darwish
Hochschule Luzern, Technik und Architektur, CH-6048 Horw, Switzerland,
Andritz AG, Andritzer Reichstrasse 68B, A-8045 Graz, Austria;
Department of Thermal Turbomachinery - TU Graz, Infeldgasse 25a, A-8010 Graz, Austria
Department of Mechanical Engineering, American University of Beirut, P.O. Box 11-0236,
Riad El Solh, Beirut 1107 2020, Lebanon
E-mail: luca.mangani@hslu.ch

Abstract. In this paper, a fully coupled block algorithm for the solution of three-dimensional incompressible turbulent flows for use in hydraulic machinery is presented. Due to the simultaneous solution of momentum and continuity equations, implicit block coupling of pressure and velocity variables leads to faster convergence in respect to classical, loosely coupled, segregated algorithms of the SIMPLE family of algorithms. Special linear solver developments were also performed in order to extend the new fully coupled solver to high efficiency HPC simulations. These modifications allow the coupled solver to retain its improved performance and robustness in addition to mesh size scalability while solving hydraulic machinery type applications.

1. Introduction
OpenFOAM® is a finite volume based open source CFD toolbox that can be used for the resolution of a wide range of flow problems. Its basic solvers that are used for solving the Navier-Stokes equations use a segregated approach. Over the past few years the popularity of OpenFOAM® has increased substantially not only in the academic community but also in a number of industries. The main drivers behind this increased adoption rate has been the accessibility to the full source code that allows complete customization of solvers, in addition to the low cost of running the code, especially when running large number of seats in parallel without the need to the usual licensing costs of commercial CFD software. The main pressure-velocity coupling technology, which the OpenFOAM® CFD library is based on, is the SIMPLE family of algorithms and related segregated procedures. Although segregated solvers have been used successfully in the past years and decades, several issues impose limits to the adoption of SIMPLE based solvers for HPC applications.

The detrimental effect of increasing problem size on the performance of the current generation of segregated pressure-algorithms, which continue to suffer from a breakdown in convergence rate when applied to the solution of large scale problems remains an issue that needs to be revisited. This is mainly due to the weak resolution of the Navier-Stokes equations’ velocity-pressure coupling using SIMPLE family-like algorithms. This issue has been addressed in detail by Darwish et al. [4] In their paper the authors tried to resolve the still existing dichotomy
between segregated and coupled approaches, clarifying the numerical differences and the issues that come along with each of them.

Addressing the weaknesses inherent to segregated solvers can be achieved by means of numerics that reflect the strong variable coupling present in the NS equations, a goal that can be accomplished by solving the system of discretized equations simultaneously, thus ensuring that the coupling of the equations is preserved during each solution step. This stands in contrast to the segregated approach used in the SIMPLE algorithms, where the equations are solved sequentially.

In a related paper [12], the authors demonstrated the advantages of using a fully implicit coupled approach in resolving the pressure-momentum coupling that arises in the Navier-Stokes equations, namely better performance, improved robustness and near linear mesh size scalability. This was demonstrated by solving four test problems on increasingly refined meshes. Mesh size scalability was defined as the characteristic of an algorithm to have its computational cost increase linearly with mesh size. In this paper the fully implicit coupled algorithm is extended to allow for the solution of turbulent problems with multi-block meshes in multiple reference frames (MFR), a feature that is critical for the simulation of turbo-machinery applications. To preserve the convergence behaviour and scalability it is essential to use a fully implicit treatment for the multi block interfaces with both conforming and non-conforming mesh topologies. To this end the implicit multi-block procedure of Darwish et al. [3] has been adopted. Since turbo-machinery flows are generally turbulent, the \( k-\omega \) SST turbulence model [14] is used with the coupled solver.

In the remainder of the paper the governing equations in a rotating reference frame are first presented, then the key features of the MFR coupled solver are pointed out, namely the concept of block coupled discretization, as well as boundary conditions and mesh interfaces for fully implicit algorithms. Finally the performance and mesh scalability of the algorithm is evaluated for a variety of industrial turbo-machinery applications.

2. The governing equations

In a rotating reference frame with constant angular velocity \( \Omega = 0 \) the Navier-Stokes equations for steady flows can be reformulated as [6]

\[
\nabla \cdot \mathbf{u}_r = 0
\]

\[
\nabla \cdot (\mathbf{u}_r \mathbf{u}_r) + 2 \Omega \times \mathbf{u}_r + \Omega \times \Omega \times \mathbf{r} = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu_{\text{eff}} (\nabla \mathbf{u}))
\]

where \( \Omega \) is the angular velocity of the rotating frame, \( \mathbf{u}_r \) the relative velocity in the rotational system. The absolute or stationary frame velocity, \( \mathbf{u} \), is written

\[
\mathbf{u} = \mathbf{u}_r + \Omega \times \mathbf{r}
\]

In equation (2) the second and third terms are the Coriolis and centripetal forces respectively. To facilitate the numerical discretisation and resolution of these equations in multiple rotating frames, the equations are re-cast in terms of the stationary, or absolute, velocity[8] yielding

\[
\nabla \cdot \mathbf{u} = 0
\]

\[
\nabla \cdot (\mathbf{u} \mathbf{u}) + \Omega \times \mathbf{u} = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu_{\text{eff}} (\nabla \mathbf{u}))
\]

In equation (5) the effective kinematic viscosity is the sum of the laminar and turbulent kinematic viscosities \( (\nu_{\text{eff}} = \nu + \nu_t) \). The convecting flux is written in terms of the relative velocity while the convected velocities are expressed in the stationary frame. As in equation (3) the relation between the relative volume flux \( \dot{V}_r \) and absolute volume flux \( \dot{V} \) can be written as

\[
\dot{V} = \dot{V}_r + (\Omega \times \mathbf{r}) \cdot \mathbf{S}_f
\]
3. Turbulence Model
Together with the coupled pressure-velocity coupling, in order to solve turbulent flows, a special turbulence model was developed based on the SST $k - \omega$ model. In a rotational reference frame the $k - \omega$ SST turbulence model becomes

$$\nabla \cdot (u_r k) - \nabla \cdot [(\nu + \nu_t \alpha_k) \nabla k] = \frac{1}{\rho} P_k - \beta^* \omega k$$

(7)

$$\nabla \cdot (u_r \omega) - \nabla \cdot [(\nu + \nu_t \alpha_\omega) \nabla \omega] = \frac{C_1 P_k}{\mu_t} - C_2 \omega^2 + \frac{2\alpha_e (1 - F_1)}{\omega} \nabla k \cdot \nabla \omega$$

(8)

where in equations (7,8) the effects of curvature or rotation [7],[15] are not accounted for.

The solution sequence of the $k - \omega$ SST turbulence model follows the segregated solution of the $k - \omega$ variables but with a special treatment of the wall function in order to improve the stability for the higher time step allowed from the coupled pressure-velocity coupling. Additionally a blended wall function has been implemented following the idea of Menter [14], using Kader’s universal law [1]. Its implementation is presented in more detail in [12].

Validation of the boundary layer reconstruction for a zero pressure gradient flow over a flat plate has been performed for different $y^+$ and compared against the analytical solution. Linear, log and wake law were used for comparison against the numerical results:

![Dimensionless boundary layer velocity profile](image)

(a) Dimensionless boundary layer velocity profile $y^+ \sim 1$.

![Dimensionless boundary layer velocity profile](image)

(b) Dimensionless boundary layer velocity profile $y^+ \sim 30$.

![Dimensionless boundary layer velocity profile](image)

(c) Dimensionless boundary layer velocity profile $y^+ \sim 100$.

**Figure 1.** Mesh independence validation: dimensionless boundary layer velocity profiles.

For all the $y^+$ the agreement with the analytical solution is very good and fits perfectly the slope of the log law. Thus the mesh independence is translated into the ability of the novel code to predict wall shear forces correctly also for multi-component simulations in which the control of the mesh at the wall is far to be controlled by the user in terms of mesh resolution.

4. Numerical discretisation
The concept of discretisation in a block coupled framework is to cast a whole set of governing equations into a single linear matrix-vector syem, as opposed to segregated algorithms where each governing equation is discretized and cast into a seperate matrix vector system. In this particular case the set of equations under consideration are composed by the momentum equations and continuity equation. Due to implicit discretization of the governing terms many additional extra-diagonal matrix coefficients arise. A draft of such a system of equations is given by Equation 9. The resulting matrix-vector system is not necessarily diagonally dominant, and therefore it solution needs special attention. The linear system solver that has been developed for solving
this system of equations is presented in section 5.

\[
\begin{bmatrix}
    a_{uu}^C & a_{uv}^C & a_{uw}^C & a_{up}^C \\
    a_{vu}^C & a_{vv}^C & a_{vw}^C & a_{vp}^C \\
    a_{wu}^C & a_{wv}^C & a_{ww}^C & a_{wp}^C \\
    a_{pu}^C & a_{pv}^C & a_{pw}^C & a_{pp}^C \\
\end{bmatrix}
\cdot
\begin{bmatrix}
    u_C \\
    v_C \\
    w_C \\
    p_C \\
\end{bmatrix}
+ \sum_{NB}
\begin{bmatrix}
    a_{uu}^{NB} & a_{uv}^{NB} & a_{uw}^{NB} & a_{up}^{NB} \\
    a_{vu}^{NB} & a_{vv}^{NB} & a_{vw}^{NB} & a_{vp}^{NB} \\
    a_{wu}^{NB} & a_{wv}^{NB} & a_{ww}^{NB} & a_{wp}^{NB} \\
    a_{pu}^{NB} & a_{pv}^{NB} & a_{pw}^{NB} & a_{pp}^{NB} \\
\end{bmatrix}
\cdot
\begin{bmatrix}
    u_{NB} \\
    v_{NB} \\
    w_{NB} \\
    p_{NB} \\
\end{bmatrix}
= \begin{bmatrix}
    b_u \\
    b_v \\
    b_w \\
    b_p \\
\end{bmatrix}
\tag{9}
\]

A detailed description of the block coupled discretization of all the governing equations' terms is given in papers by the authors [12][13].

Boundary conditions also add on to these discretized terms. Most boundary conditions yield the same coefficient regardless whether coupled or segregated algorithms are considered. This is the case for both Dirichlet and von Neumann type boundary conditions. However, it is critical to ensure that the coefficients are linearised whenever possible. Boundary conditions that act on various primitive variables at a time, such as the total pressure boundary condition or wall boundary condition or Generic Grid Interface (AMI), have to be treated implicitly in order to preserve the benefit of block coupling. Derivations of such boundary conditions can be found in [12][13].

The discretization of the turbulence equations, which are solved for in a segregated fashion after the Navier-Stokes equations, is not presented in this paper for the sake of brevity.

5. Implementation of the Block Linear Solver

The discretized block coupled set of equations now ready to be solved, requires special attention. The reason for this is simply that the sparse matrix system to simultaneously solve the momentum and continuity equations contains now 16 times more entries than the corresponding segregated matrices for a single variable would have. This means that for the block coupled approach each matrix vector multiplication is 16 times more time consuming than a vector matrix multiplication for segregated matrices with the same number of cells. It is clear that a linear solver that doesn’t scale nearly linearly with the number of cells would deprecate the overall convergence, and the gain that we wish to obtain from block coupling would vanish quickly.

For the given case a block algebraic multi-grid solver has been developed. For unstructured grids algebraic multi-grid methods are very well suited because by definition no specified mesh structure is needed for the restriction. The linear solver is based on the additive correction approach for block coupled systems of equations (BlockAMG) following Hutchinson [9] or Keller [10]. Thereby a special block-ILU smoother is used to calculate the coarse grid corrections. An exact LU decomposition is used to solve for the correction on the coarsest grid. The concept of

---

**Figure 2.** Additive correction block-AMG cycle with restriction, prolongation and pre/post smoothing.
this approach can be seen in figure (2). A more detailed description of the implemented block-
AMG solver has been outlined in [12]. Moreover the multi-grid solver has to be constructed in
a way as to fit into a well scaling parallelization framework. Its parallel performance will be
demonstrated in section 6.

6. Parallel efficiency computing
Despite of the advantage of the coupled solutions compared to SIMPLE algorithms, a key role
for a successful usage of the coupled framework is the ability to have high parallel efficiency in
distributed computations. Algebraic multigrid solvers are rather problematic in terms of parallel
scalability with a high number of processors, mainly due to the coarsest level smoothing in which
the cpu time requires for the matrix-vector multiplication is shorter than the MPI messaging
for data exchange. Special algorithms were developed for the coarser levels in order to overcome
the common parallel efficiency deficiency of the algebraic parallel multigrid solver. Several tests

![Figure 3](image)

(a) Parallel performances in terms of number of (b) Parallel performances in terms of number of
cores.

were performed comparing the standard OpenFOAM® multigrid solver named GAMG and the
newly developed block coupled version BlockAMG. As a reference together with the theoretical
standard linear law, the smooth solver performance is taken. The smooth solver is the most
meaningful solver to compare implicit solvers to, since it represents the minimum operations to
be performed for solving the implicit linear system. Figure 3(a) shows the parallel performance
using a mesh of 90 million elements and using up to 512 cores. The scaling of the new BlockAMG
linear solver is really close to the parallel scaling of the smooth solver with an clear improvement
in respect to the standard GAMG. Figure 3(b) illustrates the performance considering the
number of elements per core. Two meshes were chosen (15 and 90 million elements) in order to
take into account the number of parallel communications due to the partitioning of the mesh.
As expected there is a slight degradation of the performance as soon as the number of cells per
core decreases but an overall good performance and a net improvement compared to the original
GAMG remains.
7. Applications
7.1. Steady state simulations: Mesh size scalability

The Backward facing step test case has been chosen to demonstrate the good scalability of the outlined solver in respect to the number of grid cells. The test case has been carried out with 3 different grid sizes. For all grid sizes the same flow field has been obtained for both the block coupled and the segregated algorithm. The difference in performance and scalability is outlined in Table 1. The runs have been stopped at a specified RMS convergence threshold of $10^{-5}$ [12].

| # CV | (C)time [s] | (C)time/CV [s] | (S)time [s] | (S)time/CV [s] | S/C |
|------|-------------|----------------|-------------|----------------|-----|
| 12k  | 4.3         | 0.000351       | 32.5        | 0.002653       | 7.6 |
| 48k  | 24.1        | 0.000493       | 393.7       | 0.008051       | 16.3|
| 195k | 139.9       | 0.000715       | 5888.0      | 0.030102       | 42.1|

Table 1. Backward facing step - Performance comparison of the coupled (C) and segregated (S) algorithm

\[ \text{Scale Factor} = \frac{(\text{time/c.v.})_{n\text{Cells}}}{(\text{time/c.v.})_{\text{ref}}} \] (10)

![Scaling](Scaling.png)

Figure 4. Backward facing step - mesh size scaling

From Table 1 it can clearly be seen that the block coupled algorithm outperforms the segregated algorithm in terms of convergence speed. More importantly the Backward facing step test case proves the superiority of the coupled solver over the segregated solver in respect to scalability with increasing grid size. In Figure 4 a scaling factor is plotted as a function of the grid size in order to show the superiority of the coupled solver over the segregated one. It can be seen that the coupled solver scales almost linearly, whereas the segregated solver’s convergence behaviour deteriorates a lot with increasing mesh sizes [12].
7.2. Steady state simulations: Turbo-machinery applications

The coupled solvers’ performance in terms of mesh size scalability and convergence rate compared to segregated solvers has also been demonstrated in a wide range of turbo-machinery related applications. Among others its performance regarding stationary components such as draft tubes and Pelton distributors [12], spiral casings with distributors [5], as well as regarding rotating parts (using the rotating reference frame formulation) such as pump impellers, Kaplan or Francis runners [13] has been investigated. Due to its strong inter-equation coupling the coupled solver always outperformed the state-of-the-art segregated reference solvers. The benefit of block coupling in respect to segregated approaches can be seen in Table 2 for some rotating components.

| test case | grid incr. factor | (C)time/CV[s] | (S)time/CV[s] | S/C |
|-----------|-------------------|---------------|---------------|-----|
| Pump      | 1                 | 0.002111      | x             | x   |
|           | 2.60              | 0.002991      | 0.025250      | 8.48|
|           | 4.08              | 0.002495      | 0.028858      | 11.56|
| Francis   | 1                 | 0.000665      | 0.012543      | 18.86|
|           | 2.76              | 0.000726      | 0.025499      | 35.11|
|           | 5.00              | 0.001082      | 0.034051      | 31.48|
| Kaplan    | 1                 | 0.003489      | 0.021215      | 6.08 |

Table 2. Test case summary of mesh size scaling for coupled (C) and segregated (S) algorithm [13]

It is imperative that obtained results are physically correct. Obviously the best way to validate a code is to compare results with measurements. However, in the case of turbo-machinery applications it tends to be very costly to generate a qualitative flow analysis by measurements. Therefore in the scope of this work a comparative flow analysis between the presented coupled solvers, a widely used commercial coupled solver and a thoroughly evaluated segregated solver has been carried out to validate the physical correctness of the coupled solvers applied to turbomachinery related applications.

Figure 5 shows e.g. the qualitative flow field at a cut through a spiral casing (Devals et al. [5]).

![Figure 5. Comparison of the viscosity ratios at a cut through a spiral casing [5]](image)

Quantitative results were also found to be close to measurements as well as to the reference solvers (commercial coupled, state-of-the-art segregated) [5].
7.3. Transient simulations: deforming mesh analysis

The coupled pressure velocity coupling was also modified to perform fluid-structure interaction (FSI) simulations for hydraulic machines with deforming meshes, especially for transient stay-vanes-rotor interaction analysis. The new coupled solver has been successfully applied to a transient simulation during the start-up procedure of a modern reversible pump-turbine. Starting from speed-no load conditions and gradually increasing the guide-vanes opening angle the full transient process was simulated. The simulation domain includes a spiral casing, stay vanes, guide vanes, a turbine runner and a draft tube discretized with an overall 6 millions mesh. The simulation used 64 cores (4 nodes with 16 Intel Xeon E5-2660 (2.2 GHz) processors each equipped with 32 GB DDR3-1600 MHz RAM): with 5 deg angular resolution it took 23 hours to complete 25 revolutions. Particular attention was given to the vortices formed in the vaneless space between runner and guide vanes, additional details can be found in [2] where comparisons to measurements including global data like frequencies of the rotating stall cells and local flow patterns were presented against numerical results.

![Computational mesh used for the transient time-accurate simulations with spiral, guide-vane, rotor, diffuser zones highlighted (left) and zoom in the guidevane/rotor interface zone (right).]

![Comparison of normalized Jacobian quality index before (top) and after (bottom) mesh deformation for $\delta = 10$ deg guide-vane rotation [2].]
7.4. Navier Stokes Engine for Optimization
Since the presented block coupled solvers are fast, accurate and robust at a time, they are eligible for usage inside optimization procedures. Mentioned properties are very important when dealing with optimization loops where thousands of individual calculations are usually carried out to obtain an optimal solution for a given problem. When hydraulic components are optimized, very robust and fast CFD solvers are necessary to guarantee a fast and stable overall performance of an optimization loop. Kyriacou et al. [11] therefore have used outlined solvers as Navier-Stokes engines for optimizations of pump impeller and turbine runner hydraulics, employing the evolutionary optimizer EASY. An example of a Pareto front and an optimized pump impeller arising from this work is shown in Figure 8.

Figure 8. Pareto front of 2-objective optimization loop (left). Pump impeller shape; original geometry (shaded) vs. optimized geometry (dotted) [11]

8. Conclusion
A pressure based fully implicit coupled solver was developed and implemented within the OpenFOAM® framework for hydraulic machines applications. Extensions for steady, MRF, transient and FSI simulations were developed together with the development of a dedicated linear solver with HPC capabilities. The coupled solver demonstrated substantially improved performance in terms of cpu time and number of iterations until convergence with regard to segregated algorithms for a wide range of applications. For turbo-machinery design purposes almost all necessary features are present and implemented in a block coupled fashion in this work. However, one important boundary condition, namely the so-called mixing plane interface, has not yet been implemented in a block coupled manner. The authors intend to start working on its realization in the near future.

References
[1] Kader B 1982 Temperature and concentration profiles in fully turbulent boundary layers International Journal of Heat and Mass Transfer 24(2): 1541–1544
[2] Casartelli E, Mangani L, Romanelli G and Staubli T 2014 Transient simulation of speed-no load conditions with an open-source based c++ code IAHR 27th Symposium: Hydraulic Machinery and Systems (Montreal, Canada)
[3] Darwish M, Geahchan W and Moukalled F 2010 Fully implicit coupling for non-matching grids AIP ICNAAM 2010: International Conference of Numerical Analysis and Applied Mathematics 2010 1281 47–50
[4] Darwish M, Sraj I and Moukalled F 2009 A coupled finite volume solver for the solution of incompressible flows on unstructured grids Journal of Computational Physics 228(1):180–201
[5] Devals C, Zhang Y, Dompierre J, Vu T C, Mangani L and Guibault F 2014 3d casing-distributor analysis with a novel block coupled openfoam solver for hydraulic design application IAHR 27th Symposium: Hydraulic Machinery and Systems
[6] Hauger W, Schnell W and Gross D 2002 Technische Mechanik 3. (Springer, Berlin)
[7] Hellsten A 1998 Some improvements in menter’s k-omega sst turbulence model AIAA 98-2554
[8] Hirsch C 1991 Numerical Computation of Internal and External Flows (Wiley, NY)
[9] Hutchinson B R and Raithby G D 1986 A multigrid method based on the additive correction strategy Numerical Heat Transfer
[10] Keller S 2004 Additive correction multigrid method applied to diffusion problems with unstructured grids Braz. Soc. of Mechanical Sciences and Engineering, Proceedings of the 10a Brazilian Congress of Thermal Sciences and Engineering
[11] Kyriacou S, Kontoleontos E, Weissenberger S, Mangani L, Casartelli E, Skouteropoulou I, Gattringer M, Gehrer A and Buchmayr M 2014 Evolutionary algorithm based optimization of hydraulic machines utilizing a state-of-the-art block coupled cfd solver and parametric geometry and mesh generation tools IAHR 27th Symposium: Hydraulic Machinery and Systems (Montreal, Canada)
[12] Mangani L, Buchmayr M and Darwish M 2014 Development of a novel fully coupled solver in openfoam: Steady-state incompressible turbulent flows Numerical Heat Transfer, Part B: Fundamentals 66(1):1–20
[13] Mangani L, Buchmayr M and Darwish M 2014 Development of a novel fully coupled solver in openfoam: Steady state incompressible turbulent flows in rotational reference frames Numerical Heat Transfer, Part B: Fundamentals
[14] Menter F R 1993 Zonal two equation k – ω turbulence models for aerodynamic flows AIAA Paper 93-2906
[15] Smirnov P E and Menter F R 2009 Sensitization of the sst turbulence model to rotation and curvature by applying the spalart-shur correction term ASME Journal of Turbomachinery