# A Local Grid Refinement Algorithm on Modern High-Performance Computers

Martin Maihöfer*, Albert Ruprecht

Institute of Fluid Mechanics and Hydraulic Machinery
University of Stuttgart
10, Pfaffenwaldring st., Stuttgart, 70550, Germany

## Introduction

The numerical prediction of turbulent flow in industrial applications is essential today. In technical applications complex flow problems have to be solved. This requires great resources in computer power. To obtain a reasonable response time these simulations have to be performed in parallel on modern high-performance computers. Because of the increasing complexibility of modern parallel computers, CFD-Codes have to be adapted in order to get the full performance on all types of architectures. Therefore aspects of vectorization, shared- and distributed-memory parallelization have to be taken into account in order to guarantee a certain flexibility on modern computers.

But from an engineering point of view the use of more and more computer power is often not enough to find a solution in a reasonable time. Many times only special effects or regions of the considered flow domain are of interest so that the computational grid has to be fine enough in those regions. In such cases, like flows with thin vortex structures or boundary layer flows, the use of local grid refinements is a very efficient tool to reduce the overall computational time. The local grid refinement algorithm presented in this paper is realized with non-matching grids. Under the aspect of grid quality and automatization of the method with hexahedral elements, this is the most promising way to do. But the use of non-matching grids normally leads to significant changes in the underlying algorithm which are often not very suitable for parallelization or vectorization. Therefore the main focus of this paper is the description of the presented nonmatching grid algorithm on modern high performance computers. At first we describe the numerical scheme of the finite element flow simulation program FENFLOSS [1-2], developed at the Institute of Fluid Mechanics and Hydraulic Machinery of the University of Stuttgart, and consider aspects of parallelization and vectorization. As application the tip vortex flow of a rotating ship propeller is studied.

## Grid refinement method

Local grid refinements are useful to augment the resolution of the computational grid in certain regions of the flow domain. The elements to refine are normally identified by a user specified criterion that is applied to a previously calculated solution. Instead of refining all element, this approach is much more efficient. Local refinements can either be done with matching or non-matching elements, whereas the matching approach normally leads to a loss of grid quality. Especially in complex geometries, where some elements of the original grid are already deformed. Maintaining the quality of the original mesh can only be achieved by using embedded non-matching grids.

The following local grid refinement algorithm is based on a domain decomposition method with non-overlapping elements. In this special case the method is referred as a Schur-complement method. An overview of several domain decomposition methods is given in [3].

Those methods are normally used for solving linear systems of equations in parallel. Thereby it is usually assumed, that all subdomains consist of approximately the same number of

*E-mails: mai@ihs.uni-stuttgart.de, ruprecht@ihs.uni-stuttgart.de

unknowns. In opposite to locally embedded grids, this requirement is fulfilled when automated grid partitioning tools like *Metis* [4] are applied. Before explaining the procedure to overcome this problem, the underlying Schur-complement method is presented. Therefore two non-matching subdomains, shown in figure 1 are considered.

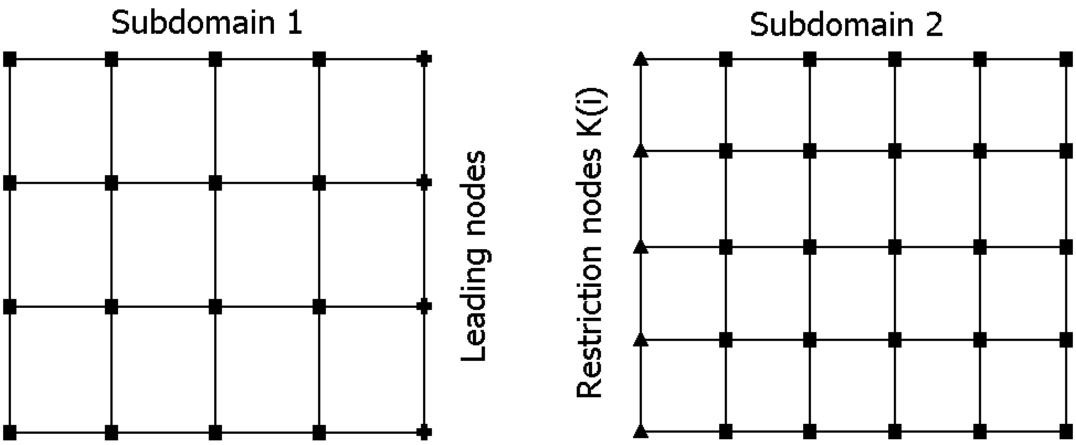

**Fig. 1.** Domain decomposition with two non-matching grids.

As a result of the non-overlapping approach, there are to many degrees of freedom along the nonmatching interface. Therefore the interface is divided into so called leading and restriction nodes. In opposite to the restriction nodes, the leading nodes are treated as degrees of freedom. Coupling of the two subdomains is achieved by introducing a Lagrange multiplier λ for each restriction node $K(i)$. The procedure is similar to optimization problems with constraints. The resulting linear equation system can be written in the form

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} & \mathbf{B}_1^T \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{B}_2^T \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{0} \end{pmatrix}, \tag{1}$$

with the matrices $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ of the two subdomains, the unknowns $\mathbf{x}_1$ and $\mathbf{x}_2$ and the vectors of the right hand side $\mathbf{b}_1$ and $\mathbf{b}_2$. The restriction operators $\mathbf{B}_1$ and $\mathbf{B}_2$ are obtained by expressing the restriction nodes in terms of the leading nodes. Thus, system (1) is completely defined and has to be solved. Applying the Schur-complement method, system (1) is transformed into

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} & \mathbf{B}_1^T \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{B}_2^T \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_{\lambda} \end{pmatrix}, \tag{2}$$

with the Schur-complement matrix **S**. After solving the Schur-complement system

$$\mathbf{S}\boldsymbol{\lambda} = \mathbf{b}_{\lambda} \tag{3}$$

for the Lagrange multipliers $\boldsymbol{\lambda}$, the unknowns $\mathbf{x}_\mathrm{i}$ can be calculated in parallel without further communication. As previously mentioned in this section, this solution procedure requires that all subdomains consist of approximately the same number of unknowns. Otherwise severe load balancing problems will occur. While using locally embedded grids this requirement is not necessarily fulfilled. Furthermore the local grid refinement should also be applicable within a serial computation with only one processor. Under those circumstances the Schur-complement method necessitates to administrate the matrices of several subdomains within the serial code. But normally CFD-Codes are designed to administrate the systems of only one computational domain. Hence this approach would require severe changes in the already vectorized and parallelized code. Therefore the presented solution procedure of the original Schur-complement method is not suitable for local refinements with non-matching grids. Concerning the code structure, it would be straightforward to store system (1) in only one single matrix. But solving this system with Krylov subspace methods is impossible because of the zero entries on the diagonal. Krylov subspace methods require that all diagonals are present with non-zero values. In order to avoid this problem, the entries of the matrices $\mathbf{B}_1^T$ and $\mathbf{B}_2^T$ in system (1) are eliminated by simple operations with the corresponding rows. The rows and columns of the restriction nodes are treated similar. The described procedure leads to the following system

$$\begin{pmatrix} \mathbf{A}_{11}^* & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22}^* \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1^* \\ \mathbf{b}_2^* \end{pmatrix}, \tag{4}$$

As a result of the elimination process, the matrices $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ and the right hand vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ of the original system are modified and therefore denoted $\mathbf{A}_{11}^*$, $\mathbf{A}_{22}^*$, $\mathbf{b}_1^*$ and $\mathbf{b}_2^*$. The matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ are representing the coupling information of the two subdomains. In opposite to system (1), system (4) is suitable for the Krylov subspace methods. The advantage of the presented method is, that system (4) can formally be treated like one single matrix resulting from a completely connected mesh. Beside the elimination process no other changes are necessary in the code. Therefore the code arrangements concerning vectorization, shared-memory and distributed-memory parallelization can be maintained without further restrictions.

In order to show the numerical behavior, the flow over a backward facing step with two local refinements in the recirculation zone is considered. The geometry and the and the embedded grids are shown in figure 2.

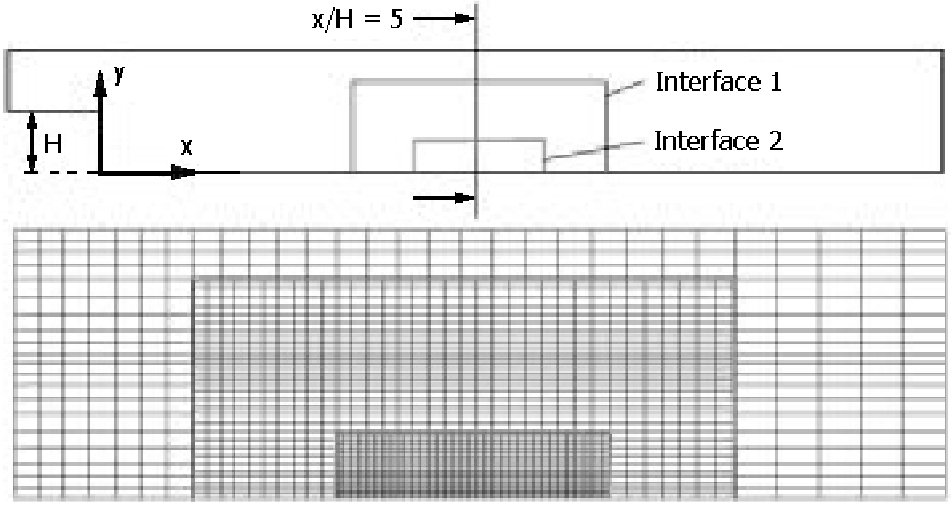

**Fig. 2.** Backward facing step with local refinements.

The calculations are carried out on a coarse grid a fine grid and the grid shown in figure 2. The coarse and the fine grid are completely connected whereas the density of the fine grid corresponds to mesh surrounded by interface 2. The density of the coarse grid is conform to the outer mesh shown in figure 2.

The calculations are carried out at a Reynolds number of *Re*=11000 with a *k-ε* model. The obtained velocity profiles at $x/H = 6$ are shown in figure 3. In can be seen that the profiles of the fine and locally refined grid are almost identical. Especially in the region of the local refinements are no differences between the two grids. Furthermore can be seen that the profile obtained with the locally refined grid is completely smooth. The non-matching interfaces at $y/H = 0.5$ and $y/H = 1.5$ are not detectable.

The convergence behavior is investigated in figure 4 where the residuals of the continuity equation are presented. It can be seen that the local refinement does not have any influence on the convergence behavior. The characteristics of all grids are similar and the solution is reached within almost the same number of nonlinear iterations.

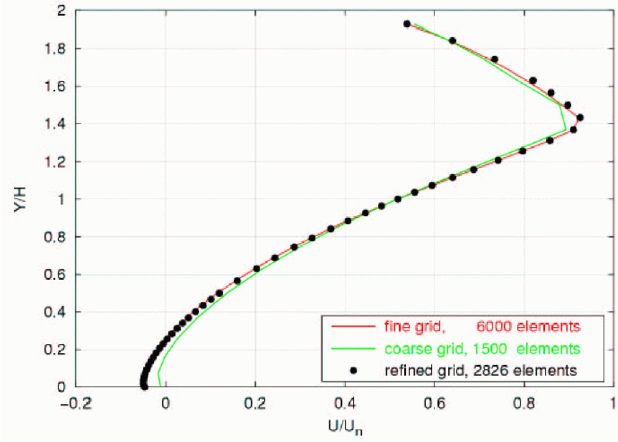

**Fig. 3.** Velocity profiles at 6 / = H x

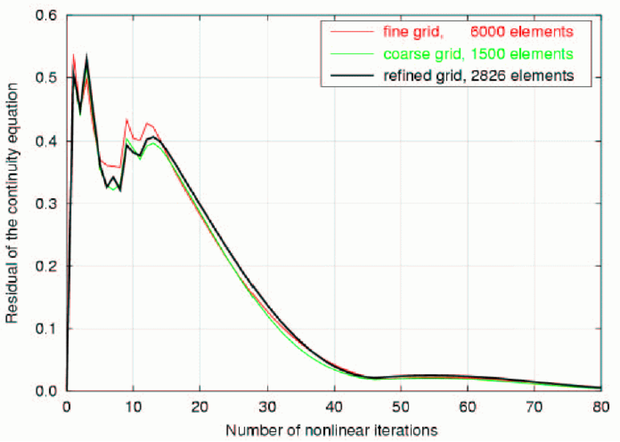

**Fig. 4.** Residual of the continuity equation

## Application

In this section the tip vortex flow field of a marine propeller is investigated. Tip vortex flows are complicated three-dimensional viscous flow phenomena and still a challenging problem in the field of computational fluid dynamic [5]. The tip vortex cavitation is the major source of noise and therefore of high interest. In order to avoid or reduce tip vortex cavitation the underlying flow field has to be studied in advance.

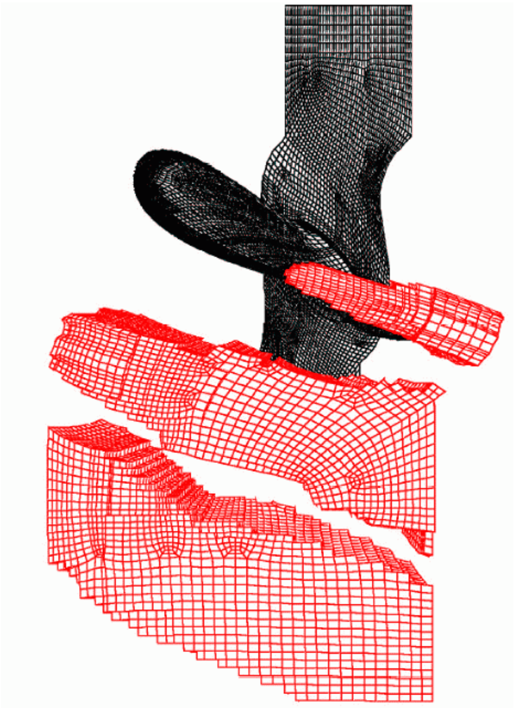

**Fig. 5.** Local embedded grid in the tip vortex region.

In order to investigate the influence of the grid resolution in the tip vortex region a local refinement is used. Based on the coarse mesh simulation the tip vortex is detected and the grid is refined along the tip vortex line. The resulting mesh respectively the locally embedded grid is presented in figure 5. The results will be discussed is the final paper.

## References

1. Ruprecht, A., *Numerische Strömungssimulation am Beispiel hydraulischer Strömungsmaschinen*, Habilitationsschrift, Universität Stuttgart, 2003.
2. Maihöfer, M. *Effiziente Verfahren zur Berechnung dreidimensionaler Strömungen mit nicht-passenden Gittern,* Dissertation, Universität Stuttgart, 2002.
3. Chan, T.F., Mathew, T., *Domain decomposition algorithms,* Acta Numerica, 61:61-144, 1994.
4. Karypis, G., Kumar, V., *A software package for partitioning unstructured graphs, partitioning meshes and computing fill-reduced orderings*, User's Manual Version 4.0.
5. Hsiao, C-T., Pauley, L.L., *Numerical computation of Tip vortex flow generated by a marine propeller*, J. Fluid Eng., 121:638-645, 1999.