University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture
Department of Energy, Power and Environmental Engineering

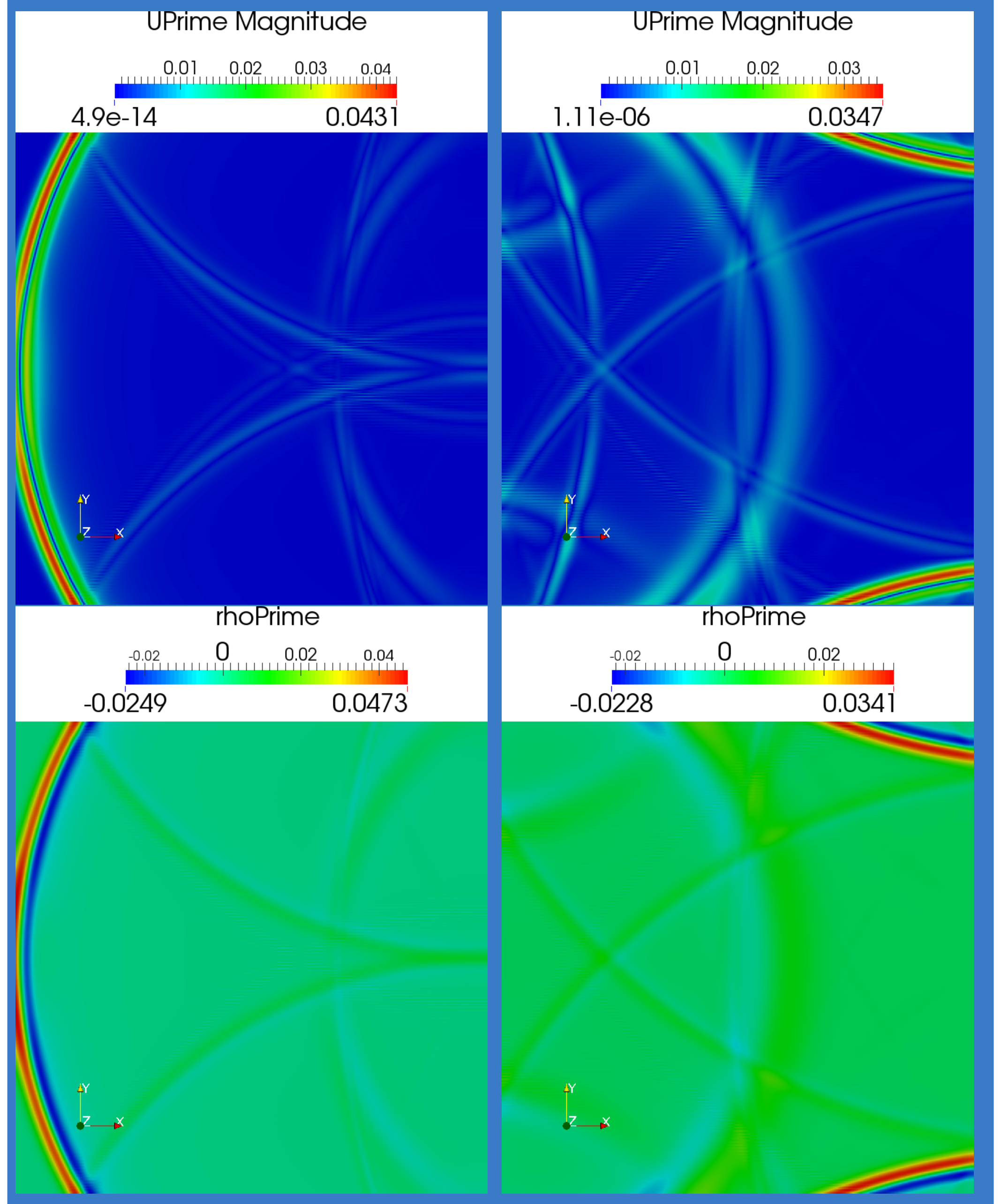
Implementation of Discontinuous Galerkin in OpenFOAM

Hrvoje Jasak
Gregor Cvijetić
Tessa Uroić

Introduction

OpenFOAM is an open source Computational Fluid Dynamics (CFD) toolbox based on the Finite Volume Method (FVM). The advantage of FVM is a straightforward and relatively easy formulation of equations and operators for computational meshes consisting of arbitrary polyhedral cells. The method is conservative and 2nd order accurate in space, which is adequate for most engineering applications.

There are some examples when FVM discretisation is insufficiently accurate such as acoustic wave propagation, shown in the figure below, and highly accurate simulations of fluid flow, e.g. using the Large Eddy Simulation (LES) approach for turbulence modelling. For such cases, Discontinuous Galerkin (DG) methods are becoming a popular choice. DG represents a combination of FVM and the Finite Element Method (FEM). The unknowns are represented by polynomials which are locally defined for each cell, which is consistent with FEM. The discontinuous values on a face shared by two cells are treated by using a numerical flux, which is taken from FVM. The order of polynomials is arbitrary and can provide highly accurate solutions. The formulation presented here was derived by prof. M. Oberlack et al. and it satisfies the polyhedral cell support requirement, provided in OpenFOAM.



FVM simulations of flow induced noise, conducted by a Master student at FSB, Zagreb: velocity and density fluctuations in time.

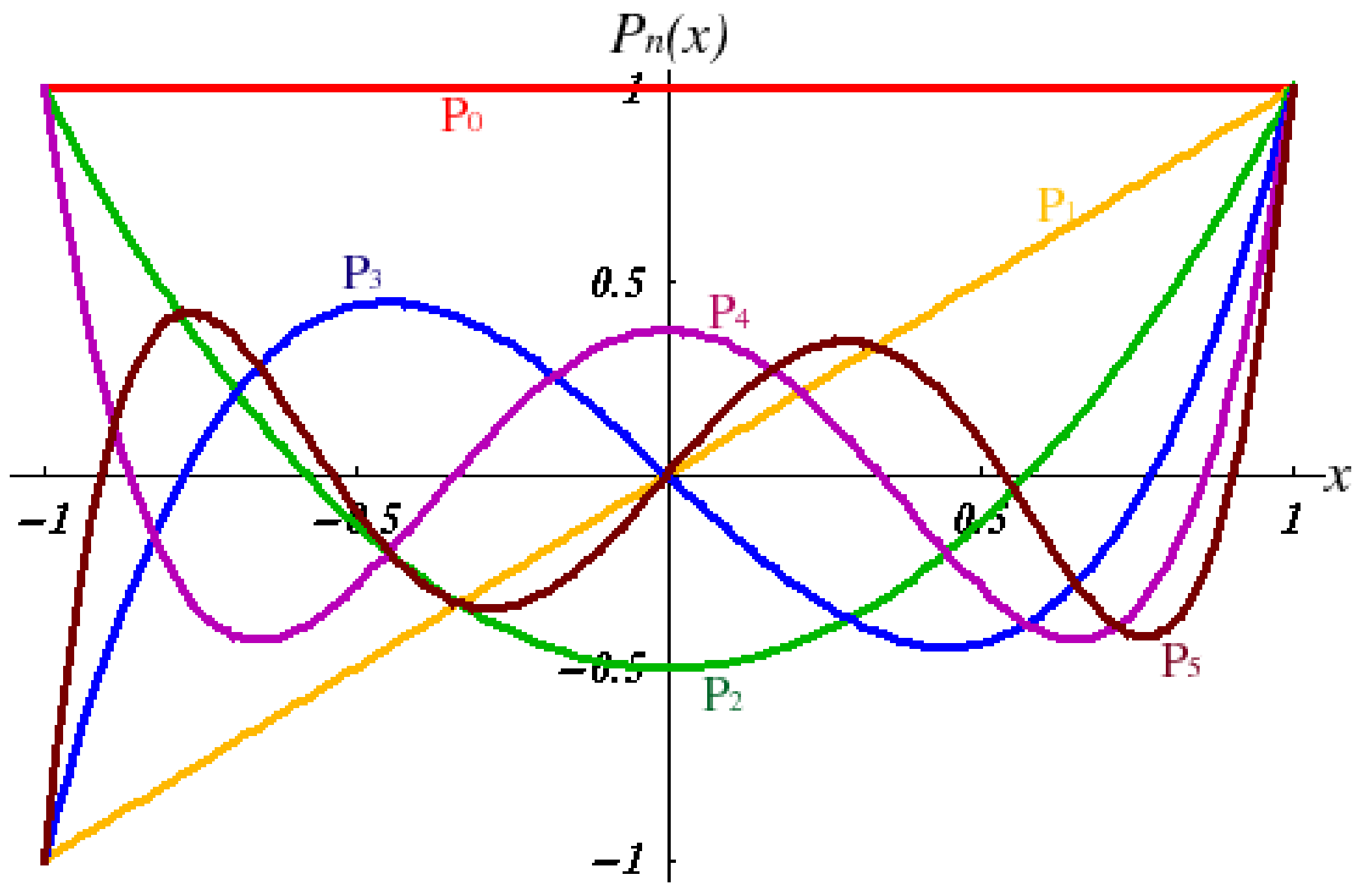
Methodology

The solution is represented by a function using a Legendre modal *polynomial basis* $\Phi = (\Phi_0, ..., \Phi_{N_p-1})$. The solution can then be expressed as a linear combination:

$$u(x) = \sum_{n=0}^{N_p-1} \Phi_n(x)\tilde{u}_n,$$

where $\tilde{u}_n$ are solution variable *weights on modal values*. The solution is a combination of polynomials of different degree: constant, linear, parabolic..., which are calculated using the hierarchical formula:

$$P_{n+1}(x) = \frac{1}{n+1}((2n+1)\cdot xP_n(x) - n\cdot P_{n-1}(x)).$$



Legendre polynomials up to 5th degree.

The equations are written in a *weak formulation*, i.e. finding the solution of the system is equivalent to minimising the overall error (residual). The error is minimised on the approximation (test) function space. We use the *Bubnov–Galerkin* approach, where the solution basis and test functions are the same. The integration of surface and volume integrals over the cell and the face is done with *Gaussian quadrature rule*.

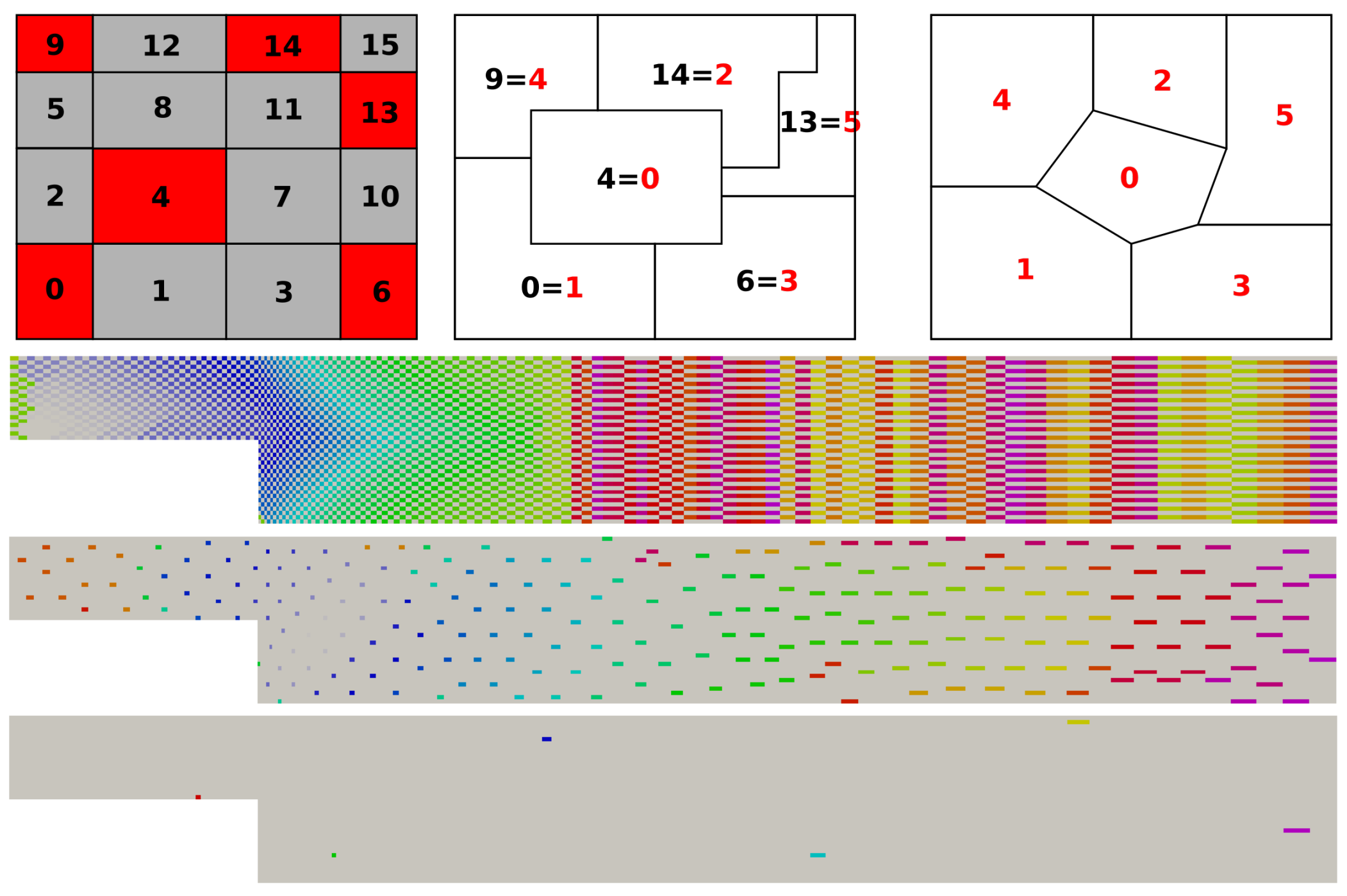
Matrix support

The system of equations to be solved can be written as:

$$\mathbf{M} \cdot \tilde{u} = \tilde{f}$$

where $\mathbf{M}$ is a block–matrix: since the unknowns are represented using polynomials of arbitrary degree, the dimensions of the matrix are $K \cdot N_p$, where K is the number of cells and N_p is the order of the polynomial. The support for block–systems in OpenFOAM is advanced and flexible, enabling easy construction of block–matrices based on the face addressing from the computational mesh. Also, an extensive library of block–linear solvers is available in OpenFOAM, with the state–of–the-art multigrid solvers, namely the selective algebraic multigrid, implemented by Jasak and Uroić (Computers&Fluids, 2018).

Selective multigrid was tested with the implicitly–coupled incompressible turbulent flow solver and theoretical convergence of one order of magnitude per iteration was achieved for complex cases with highly unstructured meshes.



Example of creating coarse matrix levels in selective AMG.

Example - DG discretisation of a Laplacian

We are assembling a linear system which is a discrete representation of the following problem:

$$\begin{cases} \Delta u = \nabla\!\bullet\!(\nabla u) = g_\Omega \text{ in } \Omega \\ u = g_D \text{ on } \Gamma_D\text{, Dirichlet B.C.} \\ \nabla u \cdot n_{\partial\Omega} = g_N \text{ on } \Gamma_N\text{, Neumann B.C.} \end{cases}$$

It was shown that the system is *consistent* and *stable* if it satisfies the coercitivity condition. Nitsche's method (1971), called Symmetric Interior Penalty (SIP) is one possibility which satisfies this condition:

$$\begin{aligned} a_{SIP}(u,v) = &\int_\Omega \underbrace{\nabla u \cdot \nabla v}_{\text{volume term - 1}} \quad dV \\ &- \oint_{\Gamma\setminus\Gamma_N} \underbrace{\{\nabla u\}\cdot n_\Gamma[|v|]}_{\text{consistency term - 2}} + \underbrace{\{\nabla v\}\cdot n_\Gamma[|u|]}_{\text{symmetry term - 3}} \; dA \\ &+ \oint_{\Gamma\setminus\Gamma_N} \underbrace{\eta[|u|][|v|]}_{\text{penalty term - 4}} \quad dA \end{aligned}$$

Here, term 2 ensures consistency. Term 3 is added for symmetry (and the method is truly variational → the discrete solution minimizes $\frac{a(u,u)}{2} - \int g_\Gamma v \; dV$). Thus, $a_{SIP}(-,-)$ is symmetric. The penalty term 4 guarantees stability (Nitsche proved that if η is taken as C/h, where h is the element size and C a sufficiently large constant, discrete solution converges to the exact solution with optimal order). It prevents spurious solutions by controlling the gradients. Considering that $a_{SIP}(-,-)$ is linear:

$$\underbrace{\begin{bmatrix} a_{SIP}(\Phi_0^0,\Phi_0^0) & \dots & a_{SIP}(\Phi_{N_p-1}^{K-1},\Phi_0^0) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ a_{SIP}(\Phi_0^0,\Phi_{N_p-1}^{K-1}) & \dots & a_{SIP}(\Phi_{N_p-1}^{K-1},\Phi_{N_p-1}^{K-1}) \end{bmatrix}}_{=:M_{SIP}} \cdot \underbrace{\begin{bmatrix} \tilde{u}_0^0 \\ \vdots \\ \vdots \\ \tilde{u}_{N_p-1}^{K-1} \end{bmatrix}}_{=:\tilde{u}} = \underbrace{\begin{bmatrix} \int g_\Omega \Phi_0^0 \, dV \\ \vdots \\ \vdots \\ \int g_\Omega \Phi_{N_p-1}^{K-1} \, dV \end{bmatrix}}_{=:\tilde{g}}$$

The unknowns are sorted in a cell–by–cell manner, to satisfy a block–structure of $\mathbf{M}_{SIP}$.

8th Floor CFD@FSB

Zagreb, 2018

www.fsb.hr/cfd

Feel free to contact us at cfd@fsb.hr or take a look at our YouTube channel: 8th Floor CFD@FSB.