CRS4

Centre for Advanced Studies,

Research and Development in Sardinia

Uta - (CA)

The CFD Code Karalis

Date: August 31^{th} 2000

by:

M. Mulas, S. Chibbaro, G. Delussu, I. Di Piazza and M. Talice

Fluid Dynamics and Combustion Area

abstract

Karalis is a parallel MPI, Finite-Volume, multiblock CFD code which solves the fully compressible Euler and Navier-Stokes equations where all couplings between dynamics and thermodynamics are allowed. This is the most general mathematical model for all fluid flows.

The code solves the coupled system of continuity, momentum and full energy equation for the velocity components, pressure and temperature. Once u, v, w, p and T are updated, arbitrary thermodynamics is supplied. The second order Roe's upwind TVD scheme is used to compute convective fluxes through the Finite-Volume cell interfaces. A V-cycle Coarse Grid Correction Multi-Grid algorithm is used, together with a 5-stage Runge-Kutta explicit time-marching method, to accelerate convergence to a steady state. This formulation, typical of aerodynamic flows, shows an eccellent efficiency even for incompressible flows as well as for flows of incompressible fluids (typically buoyancy flows), once equipped with a preconditioner. Merkle's preconditioner has been chosen because it can be easily formulated for arbitrary equations of state given as a functional relation of two independent thermodynamic variables (typically the pressure p and the temperature T), or even in tabular form, read in as an input file and used with bilinear interpolations.

Karalis implements two among the most popular turbulence models, namely the one-equation model by Spalart and Allmaras and the two-equation model by Wilcox, the $\kappa - \omega$ model, which allow a good compromise between accuracy, robustness and stability of turbulent calculations.

Code validation is presented for some typical benchmark test cases of incompressible fluid dynamics. Comparison with solutions obtained with a few popular commercial CFD codes is also presented.

Contents

1 Mathematical model **2**

1.1 The Navier-Stokes equations 2
1.2 Fluid and flow compressibility 3
1.3 Low speed flows and preconditioning 4
1.4 Arbitrary equation of state 7

2 Numerical model **10**

2.1 Algorithms' structure 10
2.2 Numerical methods 11
2.3 Turbulence models 11

3 Code structure **14**

3.1 Data structure 14
3.2 Code structure 15
3.3 Code parallelization 16
3.4 Programming philosophy 17

4 Code validation **18**

4.1 The lid driven cavity 18
4.2 The backward facing step 21
4.3 The anular loop 26
4.4 The buoyancy driven cavity 33

5 References **38**

A Preconditioning matrix **39**

B Eigenvectors and eigenvalues **40**

1 Mathematical model

1.1 The Navier-Stokes equations

The code **Karalis** solves the so-called fully compressible Navier-Stokes system of equations.

$$\boxed{\begin{aligned}
&\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \\
&\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j + \delta_{ij} p - \tau_{ij}) = \rho g_i \\
&\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j}(\rho H u_j - u_i \tau_{ij} + q_j) = \rho \dot{Q}
\end{aligned}}$$

The three equations of continuity, momentum and energy represent the basic principles of mechanics and thermodynamics, namely the conservation of mass, Newton's second law of dynamics and the conservation of the total energy. The equations are written in terms of the so-called conservative variables (density ρ, total energy per unit volume ρE, and the three components of momentum ρu_i) as the dependent variables.[1] Total enthalpy per unit volume is represented by $\rho H \equiv \rho E + p$. Constitutive relations are given by Newton's and Fourier's laws of viscosity and heat conductivity, which represent a general model for the majority of fluids of engineering interest:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k}$$

$$q_j = -\kappa \frac{\partial T}{\partial x_j}$$

The above equations relate the viscous stress tensor τ_{ij} and the conductive heat flux q_j to the velocity gradient tensor and to the temperature gradient respectively. They represent the two physical diffusion terms, namely the molecular transport of momentum and energy, and allow a quantitative correct evaluation of the thermodynamic irreversible processes that occur in real flows, satisfying the second principle of thermodynamics.

[1] The Einstein tensor notation is used whenever possible, with subscipts i, j and k which can assume the values 1, 2 and 3. Whenever a term contains twice the same index, this implies summation over that index. Alternatively, the cartesian notation will be used ($u_1 \equiv u$, $u_2 \equiv v$ and $u_3 \equiv w$)

The total energy per unit mass E includes the internal and the kinetic energy as well as the gravitational potential $\Phi = gz$. The system is fully coupled by a general equation of state of the form

$$a = a(p, T)$$

where a represents any thermodynamic variable, such as density, enthalpy or speed of sound for instance, expressed in terms of pressure p and temperature T.

1.2 Fluid and flow compressibility

The above system of equations is the most general for it allows the reversible exchange between kinetic energy and internal energy, which can occur at the expense of density variations, thus including the fluid compressibility. This mechanism, hidden in the formulation of the equations, can however be clearly identified by generating a transport equation for the kinetic energy $V^2/2$ (making the dot product of the velocity vector and the momentum equation) and subtracting it from the total energy equation to get an equation for the internal energy e. The exchange term is responsible for the coupling between momentum and total energy equation.

Another way to identify the actual flow-fluid compressibility is by differentiating the equation of state:

$$\frac{D\rho}{Dt} = \left(\frac{\partial \rho}{\partial p}\right)_T \frac{Dp}{Dt} + \left(\frac{\partial \rho}{\partial T}\right)_p \frac{DT}{Dt}$$

which allows to relate the density changes associated to a fluid particle moving with the flow, to pressure and temperature changes, rather than to a particular term in the partial differential equations. The two partial derivatives of the density represent the fluid compressibility coefficients at constant temperature and at constant pressure respectively (often represented by $\rho\chi$ and $-\rho\beta$).

Incompressible fluids have $\beta = \chi = 0$, which implies that the density is constant throughout. This in turn inhibits the possibility of exchange between kinetic and internal energy, and the energy equation can be dropped out of the system. On the other hand, in typical compressible flows of gas dynamics, density changes occur in association with pressure changes, through the compressibility coefficient at constant temperature:

$$\left(\frac{\partial \rho}{\partial p}\right)_T = \rho\,\chi \approx \frac{1}{c^2} \qquad \Longrightarrow \qquad \frac{D\rho}{\rho} \approx M^2$$

where c is the speed of sound and M the Mach number. Therefore, density changes in flows of compressible fluids generally occur if the fluid speed is high enough compared to the local speed of sound.

A third class of fluid flows is represented by buoyancy flows of incompressible fluids. Typical buoyancy flows occurs at very low speeds, compared to the local speed of sound. Due to a volumetric heat addition density changes are entirely due to the temperature changes via the compressibility coefficient β, and not to the high fluid speed. It is for this reason that buoyancy flows are mistakenly called "incompressible", with the true meaning of very low Mach number flows. As a result however, the exchange mechanism between internal and kinetic energy, switched on by the gravitational source term of the momentum equation, is the only responsible of fluid motion and the whole system is back fully coupled. A typical equation of state for buoyancy flows can be approximated as $\rho = \rho(T)$, neglecting the very small effect of pressure changes. The most general model for buoyant flows is to add the body gravitational force as a source term in the momentum equation, as ρg_i, where g_i is the gravity vector component in the i- direction. The hydrostatic pressure gradient can be separated from the the total pressure gradient and expressed as $-\rho_0 g_i$, where ρ_0 is evaluated at a reference state; thus the source term simply becomes $(\rho - \rho_0)g_i$. Keeping the compressibility coefficient β constant in a narrow range, a general equation of state can be derived in the form $\rho = \rho(T) = \rho_0 \ exp[-\beta \ (T - T_0)]$. Expanding in Taylor series for small values of the argument $\beta \ (T - T_0)$, the so-called Boussinesq approximation can be derived, the source term being $(\rho - \rho_0)g_i \approx -\rho_0 g_i \beta (T - T_0)$. With this modified form of the source term the density can be kept constant throughout the domain. These alternative formulations are all implemented in **Karalis**.

1.3 Low speed flows and preconditioning

Though the system of the Navier-Stokes equations is parabolic from a mathematical point of view, at high Reynolds number regimes the viscous diffusion terms, associated with the irreversible transport phenomena due to viscosity and heat conductivity, play the minor role of re-distributing energy and momentum among the streamlines. These phenomena are essentially confined within the boundary layers, while the core flow retain the hyperbolic character of the advection dominated flows typical, as said, of high Reynolds number regimes. The Euler system of equation, the ideal and reversible portion of the whole system of the Navier-Stokes equations, represents the pure wave propagation nature of fluid flows. The eigenvalues of the Euler matrix of the equations identify the different wave speeds which govern conservation of mass, momentum and energy, which are represented by the particle speed (fluid velocity) and by the two acoustic speeds.

In supersonic flows all the eigenvalues of the Euler matrix are of the order of the local velocity, and the system is well-conditioned. In transonic flows one of the acoustic speed approaches zero, while the other eigenvalues are all of the order of the local speed of sound; the matrix of the system becomes ill-conditioned. As the Mach number approaches zero the acoustic perturbation speed is much bigger than the propagation velocity of the perturbations moving with fluid particles. In mathematical words, as the fluid speed becomes lower and lower the eigenvalues of the system of equations become very much spread giving rise to an ill conditioned Euler matrix at very low speeds: the ratio of acoustic speed to particle speed grows unbounded. In order to apply the numerical fully compressible formulation also to incompressible flows (or to flows of incompressible fluids) and to transonic flows, a preconditioning techniques must then be used. Multiplication of the matrix of the system of the Navier-Stokes equations by a preconditioning matrix artificially changes the characteristic speeds (matrix eigenvalues) at which signals propagate in fluids: the use of the preconditioning technique alters the acoustic perturbation speed, making it of the same order of magnitude of the fluid velocity. In other words, the real incompressible world is transferred into a highly compressible one, in which the compressible formulation of the numerical algorithm does recover its original efficiency.

The preconditioned system of the Navier-Stokes equations, in compact vector form, is given by:

$$\frac{\partial Q}{\partial t} + \mathbf{P} \left(\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \right) + \mathbf{P}\ (viscous\ \ fluxes) = 0$$

where F_x, F_y and F_z represent the three components of the inviscid flux vector $\vec{F}$:

$$F_x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho u H \end{pmatrix} \qquad F_y = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ \rho v H \end{pmatrix} \qquad F_z = \begin{pmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ \rho w H \end{pmatrix}$$

Merkle's preconditioning technique [1] was chosen because it can easily formulated for abitrary equations of state. Merkle's preconditioning matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \mathbf{M}\mathbf{M}_m^{-1}$$

where $\mathbf{M}$ represents the Jacobian matrix of the vector of conservative variables Q with respect to the vector of the so-called viscous-primitive variables Q_v:

$$\mathbf{M} = \left(\frac{\partial Q}{\partial Q_v} \right) \qquad Q = \begin{pmatrix} \rho \\ \rho\vec{V} \\ \rho E \end{pmatrix} \qquad Q_v = \begin{pmatrix} p \\ \vec{V} \\ T \end{pmatrix}$$

and where $\mathbf{M}_m$ represents a modified version of $\mathbf{M}$. Note that no modification (i.e. $\mathbf{M} = \mathbf{M}_m$ and $\mathbf{P} = \mathbf{I}$) brings back to the original not preconditioned system.

$$\mathbf{M} = \begin{pmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ u\rho_p & \rho & 0 & 0 & u\rho_T \\ v\rho_p & 0 & \rho & 0 & v\rho_T \\ w\rho_p & 0 & 0 & \rho & w\rho_T \\ H\rho_p - (1 - \rho h_p) & \rho u & \rho v & \rho w & H\rho_T + \rho h_T \end{pmatrix}$$

The matrix $\mathbf{M}$ contains arbitrary thermodynamics in terms of derivatives of density and enthalpy with respect to pressure and temperature (ρ_p ρ_T h_p h_T), while the matrix $\mathbf{M}_m$ contains "modified" thermodynamics in terms of ρ_p^m and ρ_T^m. Rescaling of the characteristic speeds is obtained with proper choice for "modified" values of the fluid compressibility coefficients. To keep the condition number of $O(1)$, it can be shown [2] that a proper choice of the modified compressibility coefficients is: $\rho_T^m = \rho_T$; $\rho_p^m = 1/V_r^2$, where V_r is an appropriate reference velocity. If V_r is made varying through the domain, the preconditiong matrix $\mathbf{M}_m$ changes point by point, and a *local* preconditioning technique is applied.

A good choice of V_r is crucial for convergence. V_r should be as low as possible, but not smaller than any local transport velocity for stability considerations. Therefore V_r is chosen as the maximum between the following velocities:

[1] the local convective velocity v

[2] the local momentum diffusion velocity ν/Δ_x

[3] the local heat diffusion velocity $(\nu/\Delta_x) \cdot (1/Pr)$

The first criterium is actually dominant in turbulent flows, at high Reynolds numbers. Nevertheless, in the boundary layers or in laminar low Reynolds number flows, the diffusion criteria may play a role. For liquid metals, the Prandtl number ($Pr = \nu/\alpha$, ratio of kinematic viscosity to thermal diffusivity) is much lower than one (e.g. for liquid sodium $Pr \approx 10^{-3}$), and the criterium based on the heat diffusion velocity may become important.

Moreover, V_r should be not smaller than other characteristic speeds such as:

[4] the local $\sqrt{\Delta p/\rho}$

[5] the global so-called Brunt-Väisälä velocity $(\nu/D)\sqrt{Gr}$

where the first one represents the characteristic speed of propagation of pressure changes, and the second one, typical of buoyant flows, represents the maximum velocity of gravity waves in a turbulent flow. Criterium [4] was introduced in [12] and it is found in [13] as well. Effectiveness of the fifth criterium is currently under investigation.

If the resulting value of V_r (from the above criteria [1] to [5]) is higher than the speed of sound, then c is chosen as reference velocity and the modified constant temperature compressibility coefficient is redefined as:

$$\rho_p^m = \frac{1}{V_r} - \frac{\rho_T}{\rho h_T}$$

This is the case of supersonic flows, where no preconditioning is needed and the modified matrix $\mathbf{M}_m$ recovers its physical meaning ($\rho_p^m = \gamma/c^2 = \rho_p \Longrightarrow \mathbf{M}_m = \mathbf{M}$), so that the preconditioning is locally and automatically switched off.

The advantadges of local preconditioning have been evidenced by many authors; for example Lee [3] in his Ph-D thesis gives a wide historical excursus of the research in this field, and stresses that the matrix of the Merkle's family are developed by the analysis and optimization of the eigenvalues of the system. More efficient preconditiong matrix can be obtained by focussing the attention on the orthogonality of the eigenvectors; this may be important expecially in the stagnation points of the flow, and alternative preconditioning techniques will be probably implemented in Karalis in the future.

At steady-state (i.e. $\partial Q/\partial t = 0$) the preconditioned system shares the same solution of the original non preconditioned system.

Updating is done in terms of the viscous primitive variables Q_v, namely pressure, temperature and the velocity components. This is done multiplying the preconditioned system by $\mathbf{M}^{-1}$ to the left:

$$\frac{\partial Q_v}{\partial t} + \mathbf{M}_m^{-1} \left(\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \right) + \mathbf{M}_m^{-1} \; (viscous \quad fluxes) = 0$$

1.4 Arbitrary equation of state

In the above described mathematical framework, the choice of the working fluid is totally arbitrary. Any thermodynamics can be supplied through the matrix $\mathbf{M}$. More precisely, the derivatives of density ρ and enthalpy h, with respect to pressure p and temperature T, have to be provided. After defining:

- the compressibility coefficient at constant temperature χ:

$$\chi = \frac{\rho_p}{\rho} \equiv \frac{1}{\rho} \left(\frac{\partial \rho}{\partial p} \right)_T$$

- the compressibility coefficient at constant pressure β:

$$\beta = -\ \frac{\rho_T}{\rho} \equiv -\ \frac{1}{\rho}\ \left(\frac{\partial \rho}{\partial T}\right)_p$$

- and the specific heat at constant pressure c_p:

$$c_p = h_T \equiv \left(\frac{\partial h}{\partial T}\right)_p$$

then, for any pure substance, the required derivatives are given by:

- $\rho_p = \rho\ \chi$
- $\rho_T = -\ \rho\ \beta$
- $h_T = c_p$
- $h_p = \dfrac{1 - \beta\ T}{\rho}$

Karalis implements three options:

[1] **ideal gas**

defined in terms of the gas constant R and the specific heat at constant pressure,

$$\rho = \rho(p, T) = \frac{p}{R\ T}$$

$$\beta = \frac{1}{T}$$

$$\chi = \frac{1}{p} \equiv \frac{1}{\rho\ R\ T}$$

$$h_p \equiv 0 \qquad \Longrightarrow \qquad h = h(T) = c_p(T)\ T$$

[2] **liquid (or incompressible fluid)**

defined in terms of a reference state (h_0, T_0, p_0) and constant fluid properties:

$$d\rho = \rho_p\ dp + \rho_T\ dT$$

$$\Longrightarrow \qquad \rho = \rho(p, T) = \rho_0\ exp[-\beta\ (T - T_0) + \chi\ (p - p_0)]$$

$$dh = h_p\ dp + h_T\ dT$$

$$\Longrightarrow \qquad h = h(p, T) = h_0 + c_p\ (T - T_0) + \frac{1 - \beta\ T}{\rho}\ (p - p_0)$$

where β and χ can be assigned any value, including zero. If $\beta = \chi = 0$ then the density is strictly constant and $c_p = c_v \equiv c$, where $c_v = c_p - R$ represents the specific heat at constant volume.

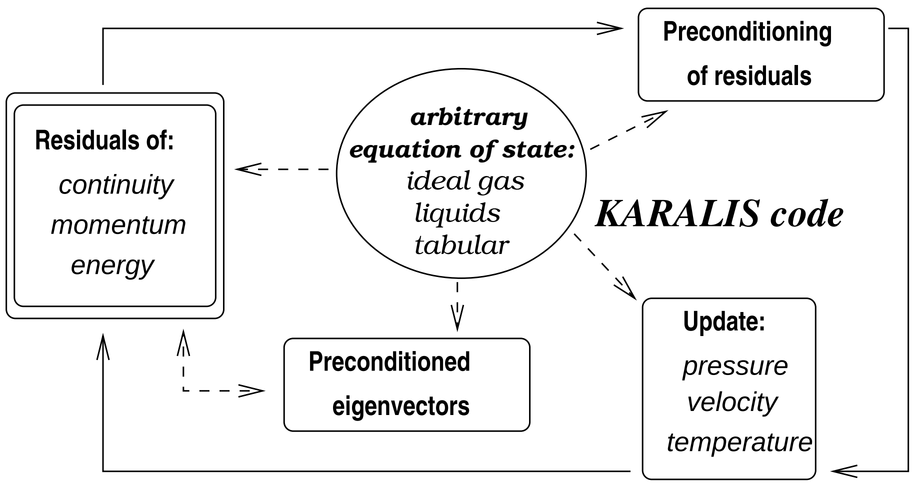
[3] **tabular equation of state**

an input file is supplied with density, enthalpy and their derivatives with respect to p and T tabulated for an arbitrary chosen set of values of pressure and temperature. Bilinear interpolation are carried out to extract the desired thermodynamic variables for the actual p and T values. This option is useful for treating, for instance, reactive fluids in thermochemical equilibrium where the tabular equation of state can be obtained from typical chemical equilibrium codes.

2 Numerical model

2.1 Algorithms' structure

The following picture represents skematically **Karalis** structure.



In the fully explicit Finite-Volume framework at each cycle (iteration) the solution algorithm is given by:

$$\Delta Q_v = \frac{\Delta t}{\Omega} \, \mathbf{M}_m^{-1} \, (RES_{inv}^m + RES_{vis})$$

and it is made of the following 4 steps (Ω represents the cell volume):

- calculation of the conservative residuals of continuity, momentum and energy;

  viscous residuals are unchanged;

  inviscid residuals are modified if TVD schemes are used [2];

- preconditioning of all residuals

- updating in terms of p, $\vec{V}$, and T

- arbitrary thermodynamics is supplied at each step

[2]see next paragraph

2.2 Numerical methods

A V-cycle Coarse Grid Correction Multi-Grid algorithm is used, together with a 5-stage Runge-Kutta explicit time-marching method, to accelerate convergence to a steady state. This formulation is typical of aerodynamic flows, and shows an excellent efficiency even for incompressible flows, once equipped with Merkle's preconditioner. Convective fluxes are computed either by TVD schemes (also called matrix dissipation schemes), or by typical scalar dissipation schemes, such as *quick* scheme for instance ([4], [5], [6]). The former are based on the decomposition of the Euler equations into waves (decomposition into characteristic variables) so that proper upwinding can be applied to each wave depending on the sign of the corresponding wave speed. This implies an eigenvector decomposition of the (now preconditioned) Euler matrix, done at a cell interface identified by its cosines n_x, n_y and n_z:

$$\mathbf{D}_p = \mathbf{PD} \equiv \mathbf{P}\ (\mathbf{A}n_x + \mathbf{B}\ n_y + \mathbf{C}n_z)$$

where:

$$\mathbf{A} = \left(\frac{\partial F_x}{\partial Q}\right) \quad \mathbf{B} = \left(\frac{\partial F_y}{\partial Q}\right) \quad \mathbf{C} = \left(\frac{\partial F_z}{\partial Q}\right)$$

so that the numerical evaluation of the flux vector $\vec{F}$:

$$F \equiv \vec{F} \cdot \vec{n} = F_x\ n_x + F_y\ n_y + F_z\ n_z$$

is given by:

$$F^*_{i+1/2} = \frac{F_i + F_{i+1}}{2} - \frac{1}{2}\ \mathbf{P}^{-1}\ \mathbf{R}_p\ |\Lambda_p|\ \mathbf{L}_p\ (Q_{i+1} - Q_i)$$

where $\mathbf{R}_p$ and $\mathbf{L}_p$ represent the matrices of right and left eigenvectors of $\mathbf{D}_p$, and Λ_p represents the diagonal matrix whose elements are $\mathbf{D}_p$ eigenvalues. All matrices contain arbitrary thermodynamics through the preconditioning matrix $\mathbf{P}$. Again, if $\mathbf{M}_m = \mathbf{M}$, then $\mathbf{P} = \mathbf{I}$ and the non-preconditioned system is recovered.

2.3 Turbulence models

Karalis implements two turbulence models: the one-equation model by Spalart & Allmaras [7], and the two-equation $\kappa - \omega$ model by Wilcox [8]. The first one is very much robust and easy to implement. It represents a simpler alternative, and more accurate also, to the widely used $\kappa - \epsilon$ model with wall functions. The second one is a low-Reynolds model, which integrates both turbulent quantities down to the solid boundary, and shows good advantages with respect to typical low-Reynolds models: essentially robustness and ease of implementation. Both models are very much popular and well known, and are here summarized:

$\kappa - \omega$ model:

$$\mu_t = C_\mu f_\mu \rho \frac{\kappa}{\omega}$$

$$\rho \frac{D\kappa}{Dt} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\kappa}\right)\frac{\partial \kappa}{\partial x_i}\right] + P_\kappa - D_\kappa$$

$$\rho \frac{D\omega}{Dt} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_i}\right] + P_\omega - D_\omega$$

P_κ	D_κ	P_ω	D_ω
$C_{1\kappa}\mu_t\Omega^2$	$C_{2\kappa}\rho\omega\kappa$	$C_{1\omega}\frac{\omega}{\kappa}P_\kappa$	$C_{2\omega}\rho\omega^2$

C_μ	f_μ	$C_{1\kappa}$	$C_{2\kappa}$	$C_{1\omega}$	$C_{2\omega}$	σ_κ	σ_ω
1.0	1.0	1.0	$\frac{9}{100}$	$\frac{5}{9}$	$\frac{3}{40}$	0.5	0.5

where Ω represents the absolute value of vorticity.

Spalart & Allmaras model

$$\mu_t = \rho\ \tilde{\nu}\ f_{v1}$$

$$\frac{D\tilde{\nu}}{Dt} = C_{b1}\ \tilde{\Omega}\ \tilde{\nu} + \frac{1}{\sigma}\ \frac{\partial}{\partial x_i}\ \left[(\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_i}\right] + \frac{C_{b2}}{\sigma}\ \frac{\partial \tilde{\nu}}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k} - C_{w1}\ f_w \left[\frac{\tilde{\nu}}{d}\right]^2$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$

$$\chi = \frac{\tilde{\nu}}{\nu}$$

$$\tilde{\Omega} = \Omega + \frac{\tilde{\nu}}{\kappa^2\ d^2}\ f_{v2}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi\ f_{v1}}$$

$$f_w = g\ \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}$$

$$g = r + c_{w2}(r^6 - r)$$

$$r = \frac{\tilde{\nu}}{\tilde{\Omega}\ \kappa^2\ d^2}$$

$$c_{w1} = \frac{C_{b1}}{\kappa} + \frac{1 + C_{b2}}{\sigma}$$

C_{b1}	C_{b2}	σ	κ	C_{w3}	C_{w2}	C_{v1}
0.135	0.622	2/3	0.41	2	0.3	7.1

where Ω represents again the absolute value of vorticity, and d the distance from the closest solid boundary.

3 Code structure

3.1 Data structure

A one-dimensional `work` array is used to store dynamically all of the global `real` data: the first part is reserved for the data which need to be stored permanently; the space left at the bottom of the `work` array is instead used dynamically to stored temporary data.

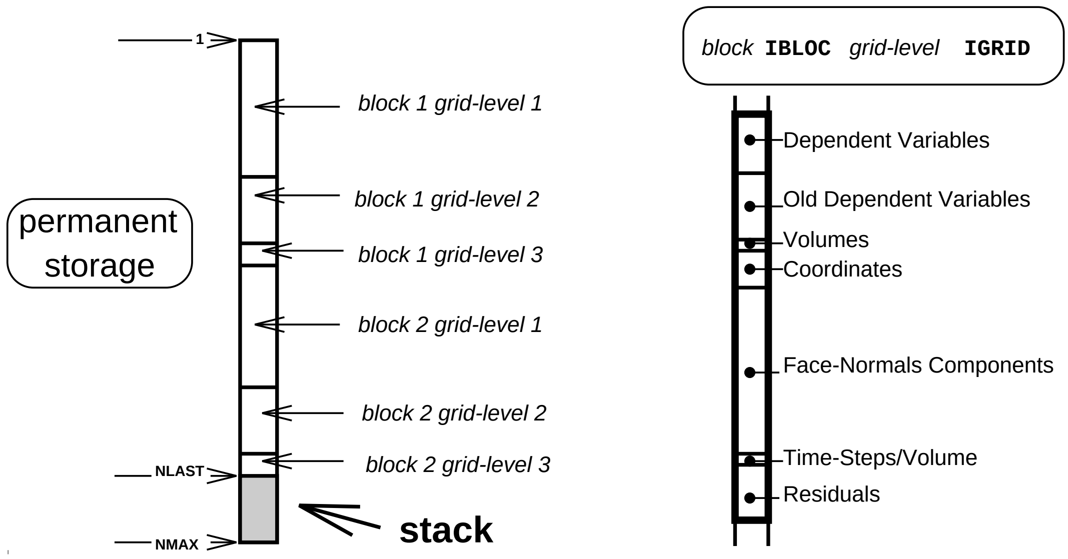


Figure 1: The `work` array structure

As skematically shown in fig. 1, each block/grid-level is assigned a portion of the storage area within the `work` array, which is organized to only store the data which do need permanent storage (shown in the left part of the picture). All data can be accessed by means of arrays of pointers, stored in a `COMMON` block.

Two layers of ghost cells are used to store the field information from adjacent blocks. The block/grid-level dimensions ($NI \times NJ \times NK$) are also stored in the `COMMON` block as block/grid-level arrays.

The free part of the `work` is used dynamically for temporary fields arrays. Temporary access to this part of the memory is allowed by means of temporary pointers, also stored in the `COMMON` block. This area is called the `stack` because memory locations can be *pushed* and *poped*, allowing allocation and de-allocation of memory space. Such a dynamic management of the stack is always used for a single block/grid-level at a time. Coupled with the multi-block environment, this approach allows to minimize the total memory requirement for a given numerical simulation: further sub-division of the computational domain in a higher number of blocks, results in fact in a smaller requirement of temporary storage.

3.2 Code structure

The tree-like code structure of **Karalis** is skematically shown in fig. 2. The upper part of the tree controls the three main sections of *initialization*, *core solver* and *output*. The three sections are made of so-called *high-level* routines. The bottom part of the structure consists instead of a collection of modules called *low-level* routines.

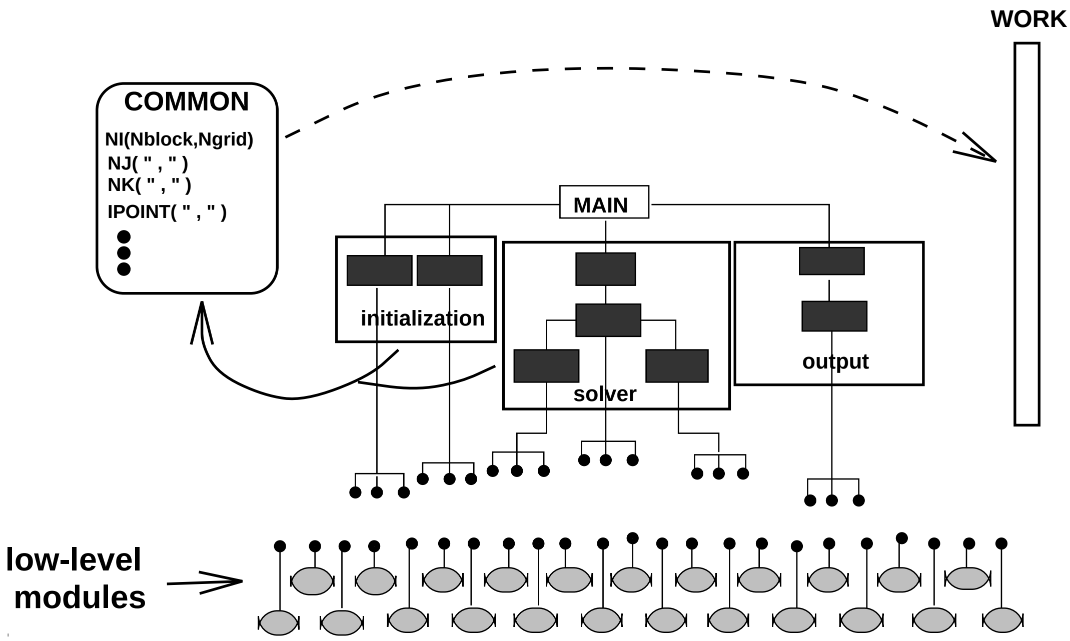


Figure 2: Tree-like structure of **Karalis**

The *high-level* routines are characterized by the fact that they have direct access to all data contained in the `work` array by means of the arrays of pointers stored in the `COMMON` block.

Inside a *high-level* routine total freedom is left to the user for looping through the grid levels and/or the blocks. The idea behind this structure is to construct the code with maximum degree of modularity and flexibility. In case, in fact, that new modules containing new functionalities, are needed, they can be easily introduced in the upper part of the structure without modifying other existing modules.

Low-level routines represent the code elementary building blocks: they perform very specific jobs on a given block/grid-level in the three-dimensional "world" (i.e. all arrays are seen as 3D arrays with the usual nested loops over the 3 indeces i, j

and k), in contrast to the one-dimensional "world" of *high-level* routines where only the whole 1D `work` array is visible.

3.3 Code parallelization

Karalis code allows the most general multi-block layout, each block having an independent (i, j, k) orientation with block faces which can be subdivided into segments. Block connectivities are allowed between any couple of face segments, regardless of their orientations. In the design of a multi-block code for use on distributed memory machines, care must be taken concerning how and where inter-block communications have to be implemented. The parallelization issues are in fact concerned with the type of domain decomposition that has to be used, and with the data dependencies among different sub-domain.

The first issue is handled assigning each block (or group of blocks) to a different processor. There are three types of data dependencies: **explicit** (information from block B at the previous time step are needed by block A and viceversa, at block boundaries, see fig. 3). These represent by far the majority of data dependencies in an explicit code, and they are handled using the two layers of ghost cells. **Implicit** dependencies occur for instance within the implicit smoothing algorithm used in the prolongation side of the multi-grid. Finally, **global** dependencies are required to determine the convergence of the solution, and in case of time accurate calculations, the time step should be uniform throughout the sub-domains.

The exchange of explicit dependencies among blocks is very localized and done for all blocks at the beginning of each Runge-Kutta stage. The idea is to allow all blocks to have all information needed before starting the numerical flux evaluation.

The data exchange is done in two steps which are skematically shown in fig. 3 for a 2-block configuration which involves only one connectivity between a full face of block **1** and a face segment of block **2**. During the first step the memory locations corresponding to two layers of ghost cells are filled with values of the corresponding inner field dependent variables. These represent all information needed to compute the inviscid fluxes through connected boundaries.

The second step, only needed for viscous calculations, is concerned with the computation of gradients of the dependent variables, evaluated by means of Gauss integration over a control volume centered at the center of the cell interface. At block boundaries however, only half control volume is available, the missing half being in fact part the adjacent block. To this purpose, 6 permanent boundary arrays (PBA_s) are provided (one per block face) within the `work` storage; the half contribution to the boundary gradients is calculated and stored in 6 temporary boundary arrays (TBA_s). All faces of all blocks are done. The above information are then transferred from TBA_s to the corresponding connected face segment of the PBA_s, and the 6

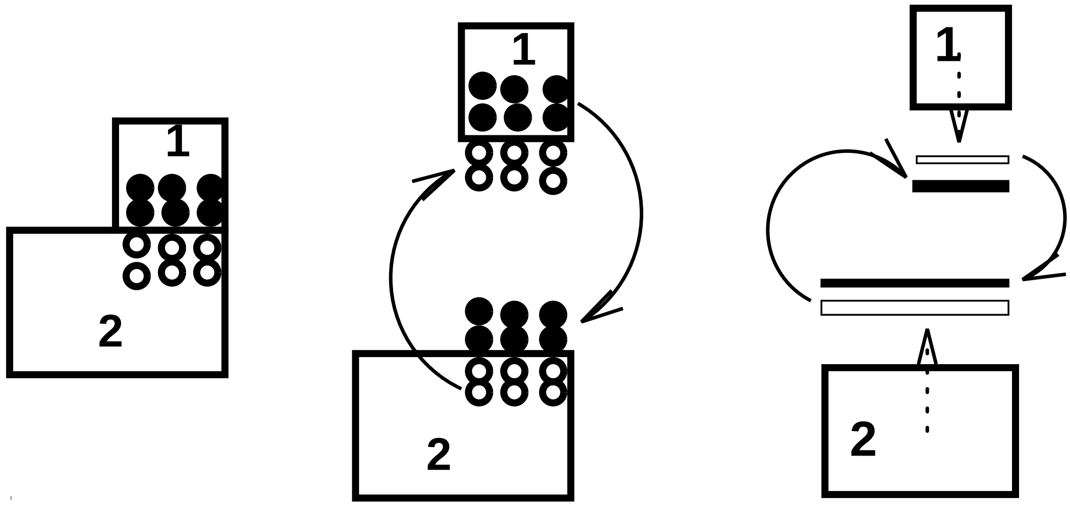


Figure 3: Data exchange: dependent variables and boundary gradients

TBA_s can be de-allocated. Finally, during viscous flux evaluation, the missing half contribution of the boundary gradients is found in the PBA_s and added. The solid arrows shown in fig. 3 represent the communication process. It has to be noted that, by using the described approach no geometrical information concerning ghost cells is needed. This allows to avoid all problems related to the construction of volume and surface normals for the ghost cells, especially when face segmentation is used.

3.4 Programming philosophy

Though the code is written in FORTRAN-77, its programming philosophy is fully Object-Oriented with heavy use of dynamic allocation and pointers.

The main object is represented by the *Domain*, or simply the Finite-Volume block, no matter to which grid-level it belongs. All grid-levels used in the multi-grid structure share in fact the same hierarchical rank. Each *Domain* owns its own data (through pointers) and functions (represented by the *low-level* routines previously described). Proper data are collected with the use of pointers stored in the `COMMON` block.

The *high-level* portion of the code can be seen as a collection of derived classes which handle the chosen algorithms, such as for instance the Runge-Kutta time advancing method or the association of extra equations for turbulence modelling. All pointers, and consequently, all data are always available at this level.

Easy implementation of new features, new functionalities as well as new transport equations has demonstrated, over the past years, the code maintenability and understandability, which represent the key features of Object-Oriented-Programming.

4 Code validation

4.1 The lid driven cavity

With reference to fig. 4, the two-dimensional motion in a squared cavity is driven by the top wall moving at constant speed along the x-axis. This is a very popular test case for incompressible fluid dynamics and **Karalis** results are compared with the results obtained by Ghia [7] using a Finite-Element incompressible Navier-Stokes code.

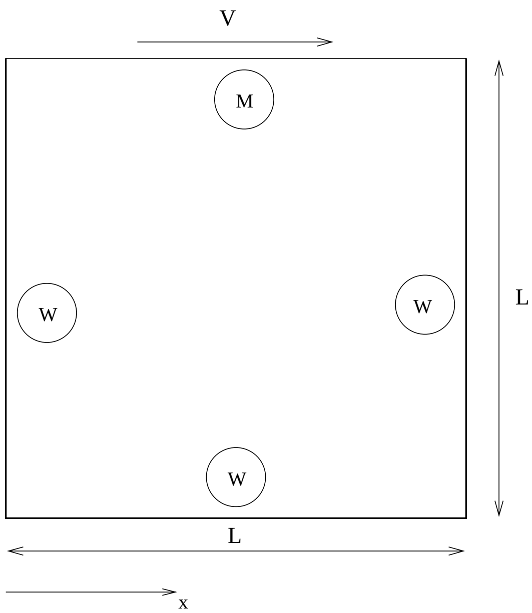


Figure 4: Geometrical definition of test case

Calculations have been performed for two values of the Reynolds number based on the top wall speed, $Re = 100$ and $Re = 1000$. The meshes are cartesian and uniform, made of 64x64 and 128x128 cells respectively for the two Reynolds number investigated. The table below shows the assigned values of density, dynamic viscosity and wall speed.

fluid	Reynolds n.	density	μ	wall speed
water	100	1000.0	10^{-3}	0.0001
	1000	1000.0	10^{-3}	0.001

Comparison to Ghia's data is done plotting the profiles of the two components of the field velocity, u e v, along the y and x mid sections of the cavity: $u(y)$ e $v(x)$ respectively.

Figures 5 and 6 show the normalized velocity profiles for $Re = 100$ and $Re = 1000$ respectively.

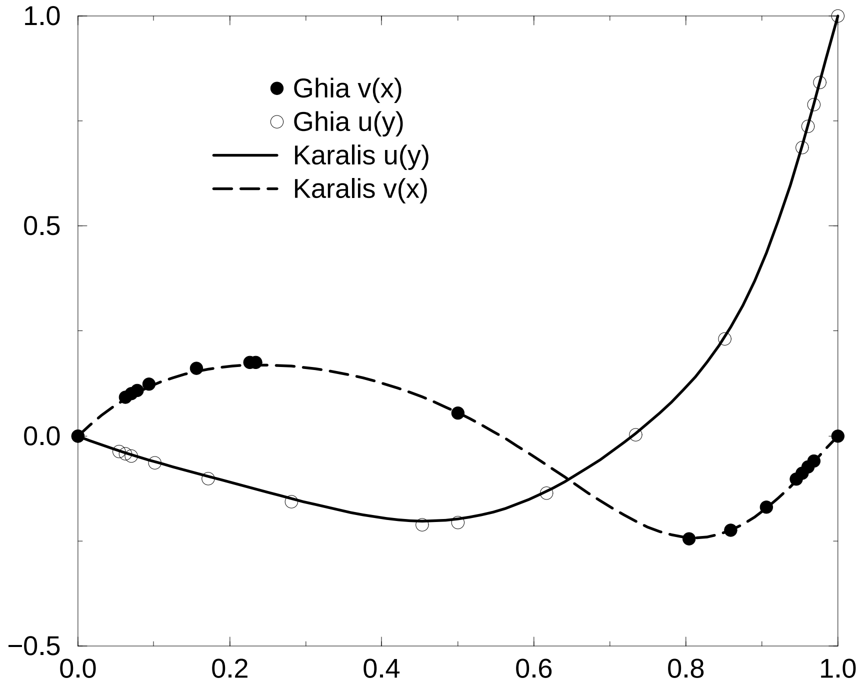


Figure 5: velocity profiles for $Re = 100$

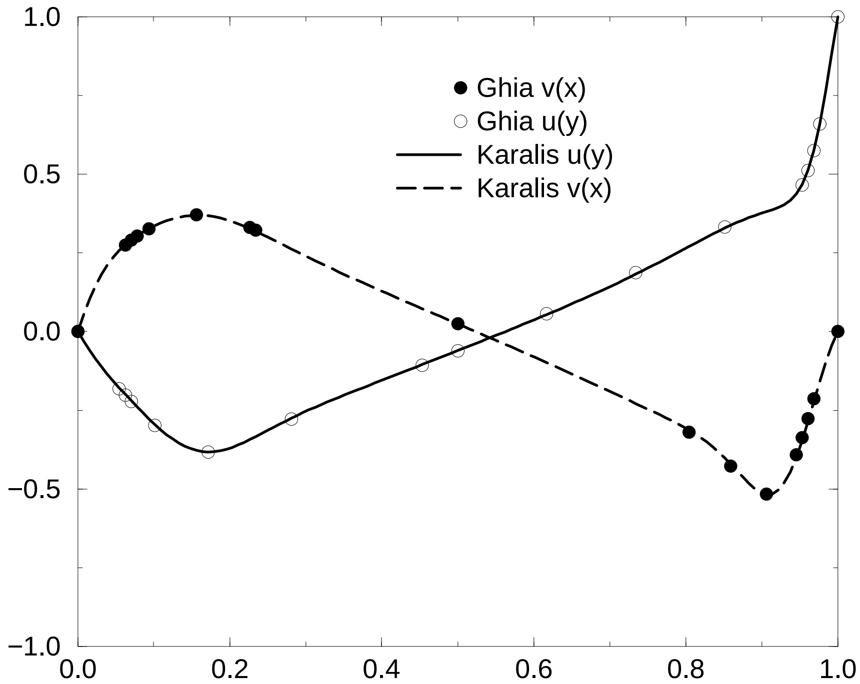


Figure 6: velocity profiles for $Re = 1000$

4.2 The backward facing step

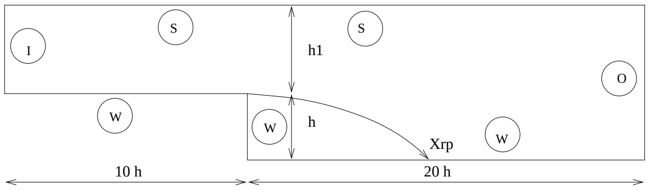


Figure 7: Problem definition

The backward-facing-step test case is one of the most widely used for the validation of CFD codes in general and for the assessment of turbulence models in particular. The reference solution chosen is represented by the DNS solution of Moin [8] for a Reynolds number $Re = 5,100$ based on the centerline velocity and on the step height [9].

The test case is described by figure 7: the step determines the flow separation and the ricirculating zone, before the boundary layer re-attachment. The DNS data found the re-attachment point at a distance of $x/h = 6.28$ from the step.

The computational domain, shown in the same fig. 7, starts at $x/h = 10$ upstream of the step location, and ends at $x/h = 20$ downstream. A two-block mesh has been used with 48×40 and 64×80 cells in the two blocks (the first one upstream, and the second one downstream of the step). A severe mesh stretching is provided close to the solid boundaries with a value of the non-dimensional grid spacing y^+ of the order unity.

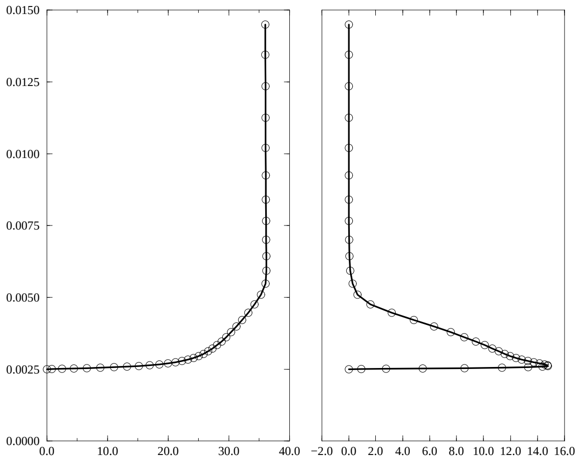


Figure 8: Inlet assigned profiles: $U(y)$ left, $\kappa(y)$ right

Table 1 shows the boundary conditions of the test case.

I	Inlet
O	Pressure Outlet
W	Adiabatic Solid Wall
S	Symmetry Plane

Table 1: Boundary conditions

At the inlet boundary the velocity and turbulent kinetic energy profiles coming from the DNS simulation have been assigned (fig. 8).

Numerical simulations have been carried out with four codes: the commercial codes **CFX** and **StarCD**, and the CRS4 in-house developed codes **Karalis** and **Tanit**. All codes but **Karalis** employ typical incompressible fractional-step numerical algorithms.

Table 2 shows the calculated re-attachment points:

Code	model	re-attachment point [x/h]	error %
Moin	DNS	6.28	-
CFX	$\kappa - \epsilon$	5.52	12.10
StarCD	RNG $\kappa - \epsilon$	5.97	4.93
Karalis	$\kappa - \omega$	6.19	1.43
Tanit	$\kappa - \epsilon$	5.14	18.15

Table 2: Calculated re-attachment points

Solutions are presented in terms of velocity and Reynolds Stress profiles at four different locations downstream of the step (fig. 9): the large differences are mostly due to error in the re-attachment evaluations. Fig. 10 shows the same profiles with the x-coordinates shifted in order to match the re-attachment points.

It has to be noted that all codes but **Karalis** make use of wall functions at solid boundaries: their comparison against the $\kappa - \omega$ model might then seem unfair. However, as already mentioned, the standard $\kappa - \omega$ model [3] can be compared to the $\kappa - \epsilon$ with wall functions as far as ease of implementation and robustness are concerned. **Karalis** and **Star-CD** give quite satisfactory solutions in the framework of RANS modelling (as opposed to the DNS). They use superior turbulence modelling compared to the standard $\kappa - \epsilon$ with wall functions udes by **CFX** and **Tanit**.

[3] the standard $\kappa - \omega$ model does not implement any of the typical low-Reynolds modifications of two-equation models.

However, **Tanit** flow field and Reynolds Stresses appear satisfactory when plotted relatively to the calculated re-attachment point. **CFX** Reynolds Stress profiles are both quantitavely and qualitatively wrong, particularly close to the solid boundary. This behaviour is due to the inability to reach convergence for this particular test case.

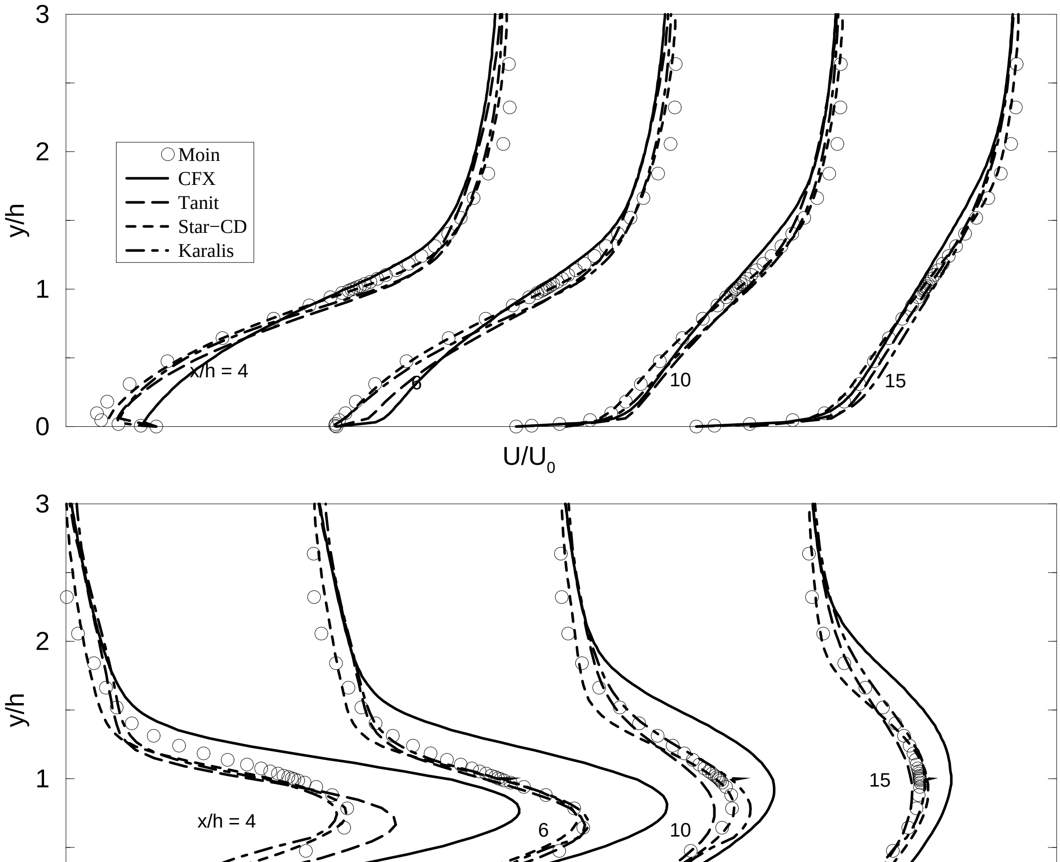


Figure 9: Velocity and Reynolds Stresse profiles at $x/h = 4, 6, 10$ and 15

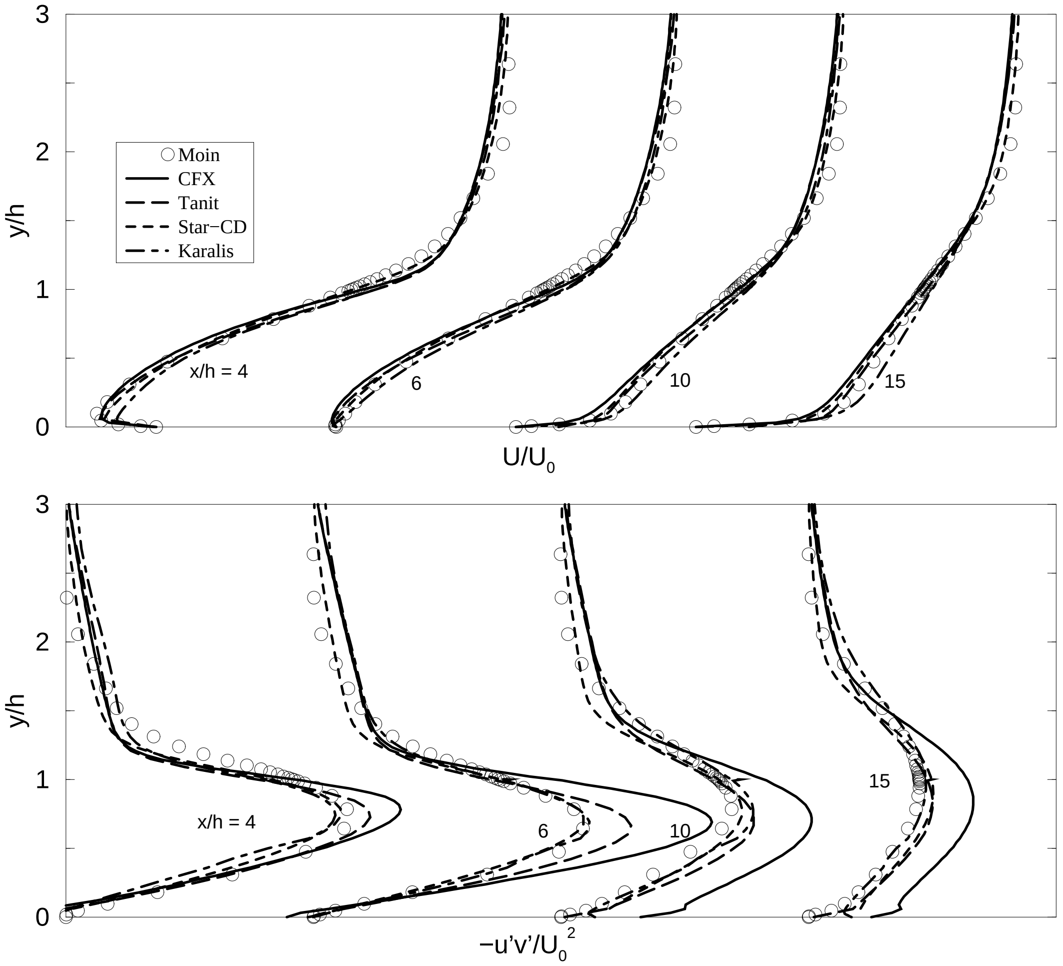


Figure 10: Velocity and Reynolds Stresse profiles at same distance from re-attachment point

4.3 The anular loop

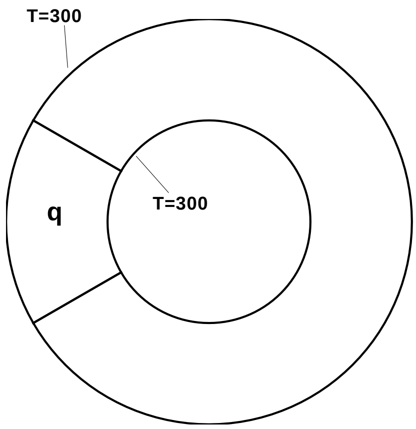


Figure 11: Reference geometry

With reference to fig. 11, the two-dimensional motion in the anular loop is buoyancy-driven, with the walls kept at a constant temperature, and a volumetric heat generation supplied in a sector of the domain at a rate $\rho\dot{Q}$. The enclosure is filled with a liquid metal (the eutettic Li17-Pb83) whose Prandtl number is $Pr = 0.0321$. The physical properties of the fluid are summarized in the table below.

Property	symbol	value(SI units)
molecular viscosity	μ	$2.2 \cdot 10^{-3}$
density	ρ	9000
specific heat	c_p	190
thermal conductivity	λ	13
thermal expansion coefficient	β	$1.68 \cdot 10^{-4}$
Prandtl number	Pr	$3.21 \cdot 10^{-2}$

Table 3: Physical properties of the Li17-Pb83 fluid at a reference temperature $T = 573K$

With this test case, the capability of **Karalis** to treat laminar or turbulent buoyancy-driven flows of low-Prandtl number fluids with internal heat generation is shown. The fluid is practically incompressible, but the flow is solved by the general compressible algorithm, letting the density vary as a function of temperature and pressure by the general equation of state $\rho = \rho(p,T) = \rho_0 \, exp[-\beta\,(T-T_0)+\chi\,(p-p_0)]$, though the pressure dependence is certainly negligible. The source term in the momentum equation is expressed as $(\rho - \rho_0)g$ (y direction), with the reference hydrostatic pressure gradient separated from the total pressure gradient and included in the source term (see section 1.2). A comparison is shown with results from two popular commercial codes: **CFX** and **Fluent**, the first one having the classical incompressible formulation with a SIMPLE family algorithm for the pressure-velocity coupling, while the latter one having a compressible formulation very similar to that of **Karalis** [13].

The internal radius is 10^{-2}, while the external radius $2 \cdot 10^{-2}$. The grid used for the calculations is shown in fig. 12, with 120 points in the poloidal direction and 40 points in the radial direction.

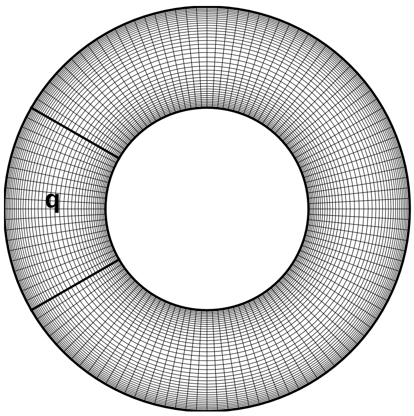


Figure 12: Grid used for the calculations

The test cases simulated are summarized in table 4. The Grashof number Gr is based on the maximum purely conductive temperature drop $\Delta T_c = \rho q D^2/8\lambda$, and is defined as : $Gr = g\beta\Delta T_c D^3/\nu^2$, where D is the distance beteween active walls (in the present case $D = 10^{-2}$) . The square root of the Grashof number represents the ratio of buoyant to viscous forces. The Brunt-Väisälä velocity $v_{ref} = \nu/D \cdot \sqrt{Gr}$ is a typical reference scale for buoyant flows.

In figs. 13 and 14 the temperature distribution and the velocity vector plot are shown for the case A.

In the temperature plot (fig. 13), the temperature drop between consecutive levels is 1/30 of the conductive temperature drop. Actually, temperature behaves as

case	$\dot{Q}$[W/kg]	ΔT_c	Gr	code	regime	model	$v_{ref} = \nu/D \cdot \sqrt{Gr}$
A	83.5	0.722	$2 \cdot 10^4$	**Karalis**	laminar	-	$3.45 \cdot 10^{-3}$
B	8350	72.2	$2 \cdot 10^6$	**Karalis**	turbulent	Spalart-Almaras	$3.45 \cdot 10^{-2}$
C	8350	72.2	$2 \cdot 10^6$	**Fluent**	turbulent	Spalart-Almaras	$3.45 \cdot 10^{-2}$
D	8350	72.2	$2 \cdot 10^6$	**CFX**	turbulent	rng $k - \epsilon$	$3.45 \cdot 10^{-2}$

Table 4: Test cases

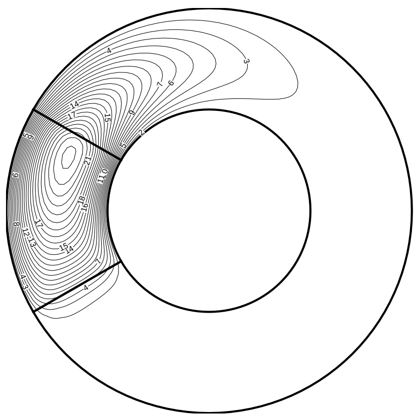


Figure 13: Temperature distribution for the case A

a passive scalar (with internal energy per unit mass generated at a rate $\dot{Q}$ in the sector on the left) and is convected around in the domain. The maximum temperature is represented by the isoline 24, with a resulting relative drop of $\Delta T_c/\Delta T_{max} \approx 1.36$; this ratio can be considered also as the overall Nusselt number, and represents a measure of the reduction of the maximum temperature caused by the convection.

While the temperature field develops, the corresponding density variations will generate buoyancy forces, and the fluid will move clockwise. In the vector velocity field shown in fig. 14, a reference vector $= 5v_{ref}$ is drawn to fix a scale.

The temperature distributions in the cases B (**karalis**), C (**Fluent**) and D (**CFX**), are shown in figs. 15, 16 and 17 respectively.

The fields relative to cases B (**Karalis**) and C (**Fluent**) are almost coincident. In fact they are obtained with the same turbulence model and a similar compressible formulation. The isoline of maximum temperature is the number 9, leading to an overall Nusselt number $\Delta T_c/\Delta T_{max} \approx 3.3$. This great enhancement of the heat

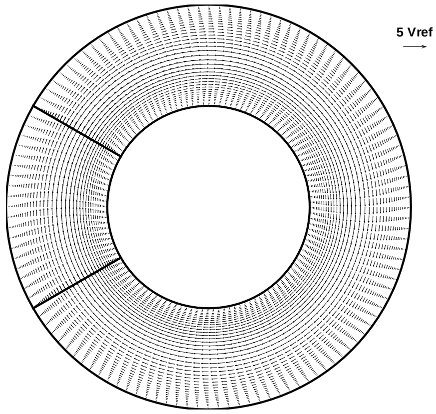


Figure 14: Velocity vectors for the case A

transfer is obiouvlsly due to the turbulent convective transport phenomena. The result of **CFX** (fig. 17) is obtained by a different turbulence model, and thus the isotherms differ slightly.

Finally, the vector plot and the turbulent viscosity distribution obtained by **Karalis** are shown in figs. 18 and 19 respectively.

It should be noticed that the Brunt-V*äisälä* velocity introduced is a good scale for the motion at any Grashof number. It is however, for this test case, of the same order of the heat conduction velocity scale and it is not yet clear whether it can be important in different test cases. The turbulence viscosity field has a maximum value of 0.025 (isoline 13) in the centre; this value is almost 10 times the molecular value, as it is reasonable at this higher Grashof number.

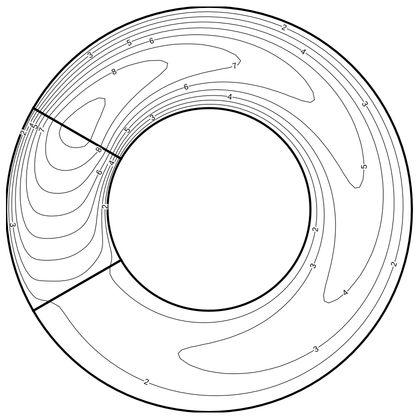


Figure 15: Temperature distribution for the case B (**Karalis**)

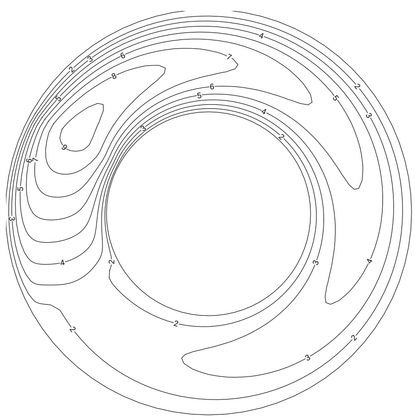


Figure 16: Temperature distribution for the case C (**Fluent**)

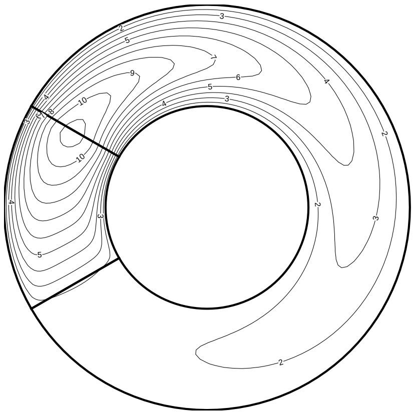


Figure 17: Temperature distribution for the case D (**CFX**)

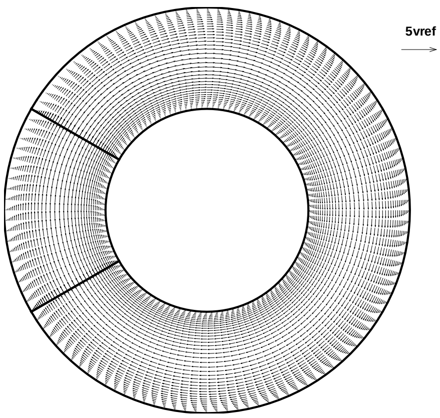


Figure 18: Vector plot distribution for the case B

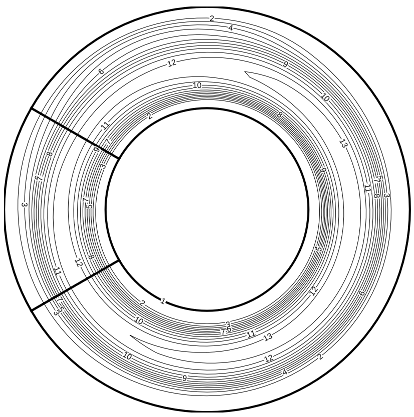


Figure 19: Turbulent viscosity distribution for the case B

4.4 The buoyancy driven cavity

The aim of this test case is to show the capability of **Karalis** to treat laminar buoyant flows with three different approaches: fully compressible, hydrostatic pressure gradient separated, and the so-called Boussinesq approximation. The first formulation is the most general: the gravity source term $\rho\vec{g}$ is added to the momentum equation; the second formulation is obtained by separating a reference "hydrostatic" pressure p_0 from the pressure gradient term:

$$-\nabla p + \rho\vec{g} \equiv -\nabla(p - p_0) + (\rho - \rho_0)\vec{g} \equiv -\nabla p' + (\rho - \rho_0)\vec{g}$$

This formulation is equivalent to the general one, but it allows to avoid the complication of the pressure boundary condition when treating external flows (the hydrostatic pressure gradient need not be taken into account as far as the boundary condition is concerned). The Boussinesq formulation takes into account the density variations which originate the buoyant flow only through the approximated momentum source term given by:

$$-\nabla p + \rho\vec{g} \equiv -\nabla(p - p_0) + (\rho - \rho_0)\vec{g} \approx -\nabla p' + \rho_0\beta(T - T_0)\vec{g}$$

leaving the $\rho = const = \rho_0$ approximation throughout the whole governing system of equations.

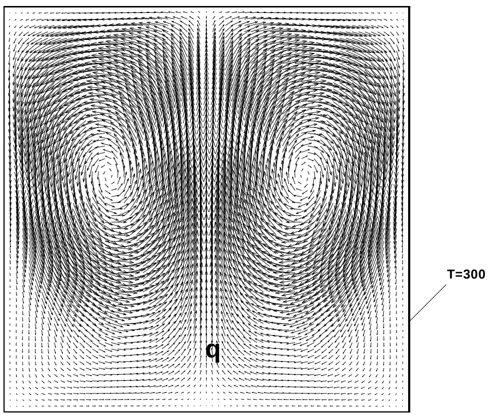


Figure 20: Velocity field

With reference to fig. 20, the two-dimensional motion in the square cavity (side $D = 2 \cdot 10^{-3}$) is buoyancy-driven, with the left and right walls kept at a constant temperature and the top and bottom walls adiabatic. A volumetric heat generation is supplied in the domain at a rate $\rho\dot{Q}$. The cavity is filled with a liquid metal (the eutettic Li17-Pb83) whose Prandtl number is $Pr = 0.0321$. The physical properties of the fluid are summarized in the table 3 in section 4.3 . The Grashof number is fixed to $2 \cdot 10^5$, well inside the laminar stationary range. An upward rising flow is established in the cavity center and goes down along the walls where an outgoing heat flux occurs. The exiting heat flux must be in equilibrium with the volumetric heat addtion in order to reach the sought steady state situation.

The test cases simulated are summarized in table 5.

case	$\dot{Q}$[W/kg]	ΔT_c	Gr	S_m (momentum source)	$v_{ref} = \nu/D \cdot \sqrt{Gr}$
A	$2.6 \cdot 10^6$	902.5	$2 \cdot 10^5$	ρg	10^{-2}
B	$2.6 \cdot 10^6$	902.5	$2 \cdot 10^5$	$(\rho - \rho_0)g$	10^{-2}
C	$2.6 \cdot 10^6$	902.5	$2 \cdot 10^5$	$-\rho_0 g\beta(T - T_0)$	10^{-2}

Table 5: Test cases

The three cases show a similar convergence history, and a comparison of the temperature fields is shown in figs. 21, 22 and 23.

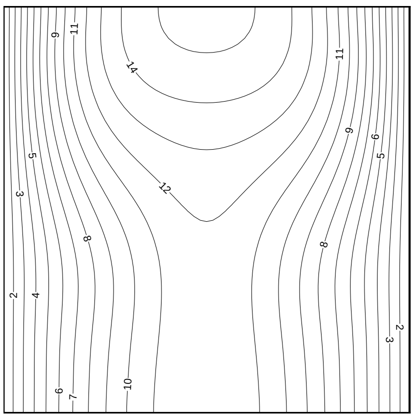


Figure 21: Temperature distribution for the case A ($S_m = \rho g$)

The temperature drop between consecutive levels is 1/15 of the conductive temperature drop. The convective transport modify the horizontal stratification induced

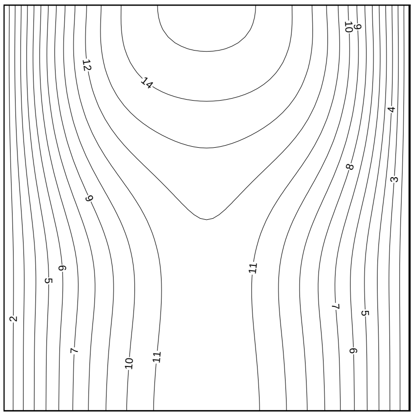


Figure 22: Temperature distribution for the case B ($S_m = (\rho - \rho_0)g$)

by diffusion. At this low Grashof number the convection is not sufficient to decrease the maximum conductive drop; on the contrary, hot fluid is accumulated in the central-top, leading to an overall Nusselt number lower than 1 (see also Di Piazza [11]). The temperature fields in the cases A($S_m = \rho g$) and B($S_m = (\rho - \rho_0)g$) correctly show identical results. The temperature field in the case C(Boussinesq) differs a little from the cases A and B, because the Boussinsq approximation is valid for $\beta(T - T_0) << 1$; in this case $\beta(T - T_0) \approx 0.15$, and the condition is not fully satisfied. Moreover the coupling between density and velocity (continuity equation) is missing ($\rho = const = \rho_0$ and $\nabla \cdot \vec{V} = 0$).

A comparison of the pressure fields in the case A(fig. 24) and B(fig. 25), shows that the hydrostatic pressure gradient naturally appears in the formulation A as a result of the calculation.

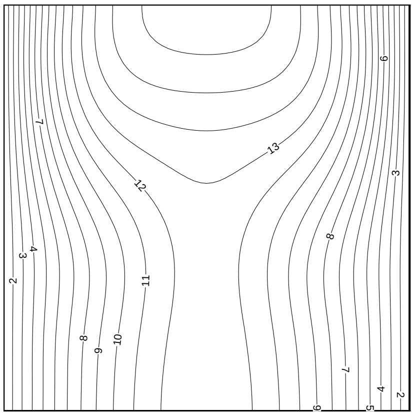


Figure 23: Temperature distribution for the case C ($S_m = -\rho_0 g \beta (T - T_0)$, Boussinesq)

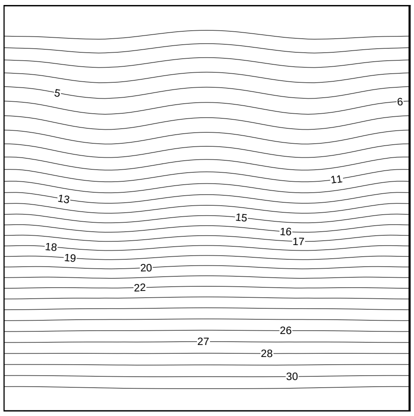


Figure 24: Pressure distribution for the case A ($S_m = \rho g$)

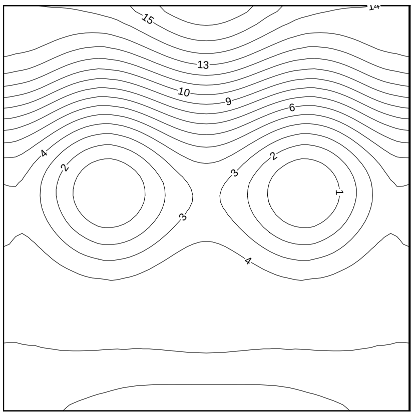


Figure 25: Pressure distribution for the case B ($S_m = (\rho - \rho_0)g$)

5 References

[1] Merkle C.L. *et al*: *Computation of flows with arbitrary equation of state*,AIAA J., vol.36, no.4, 1998.

[2] Merkle C. : *Preconditioning methods for viscous flow calculations*, ??, vol.??, no.?, 19??.

[3] Lee D. : *Local preconditioning of the Euler and Navier-Stokes equations*, Ph-D thesis, University of Michigan 1996.

[4] Hirsch Ch. : *Numerical computation of internal and external flows*, Volume 2, Wiley 1990.

[5] Roe P.L. : *Approximate Riemann Solvers, Parameters Vectors and Difference Schemes*, J. of Computational Physics, vol.43, 1981.

[6] Yee H.C. : *Construction of Explicit and Implicit Symmetric TVD Schemes and Their Apllications*, J. of Computational Physics, Vol.68, 1987.

[7] Spalart P.R. and Allmaras S.R. : *A one-equation turbulence model for aero-dynamics flows*, La Recherche Aerospatiale, no.1, 1994.

[8] Wilcox D.C. : *Reassessment of the scale-determining equation for advanced turbulence models*, AIAA J., vol.26, no.11, 1988.

[9] Ghia, U., Ghia, K.N., and Shin, C.T. : *High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method*, J. of Computational Physics, vol.48, 1982.

[10] Le H., Moin P. and Kim J.: *Direct Numerical Simulation of Turbulent Flow over a Backward Facing Step*, J. of Fluid Mechanics, vol.330, 1997.

[11] Di Piazza I. : *Prediction of free convection in liquid metals with internal heat generation and/or magnetohydrodynamic interactions*, Ph-D thesis, University of Palermo, Italy 2000.

[12] Venkateswaran S., Merkle L.: *Analysis of preconditioning methods for the Euler and the Navier-Stokes equations*, 30^{th} Computational Fluid Dynamics, VKI LS 1999-03.

[13] Weiss J.M., Maruszewski J.P., Smith W.A.: *Implicit solution of preconditioned Navier-Stokes equations using algebraic multi-grid*, AIAA J., vol.37, no.1, 1999.

A Preconditioning matrix

In section 1.3 Merkle's preconditioning matrix $\mathbf{P}$ was introduced:

$$\mathbf{P} = \mathbf{M}\mathbf{M}_m^{-1}$$

$$\mathbf{M} = \begin{pmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ u\rho_p & \rho & 0 & 0 & u\rho_T \\ v\rho_p & 0 & \rho & 0 & v\rho_T \\ w\rho_p & 0 & 0 & \rho & w\rho_T \\ H\rho_p - (1-\rho h_p) & \rho u & \rho v & \rho w & H\rho_T + \rho h_T \end{pmatrix}$$

$$\mathbf{M}^{-1} = \begin{pmatrix} \dfrac{\rho h_T + \rho_T(H - V^2)}{d} & \dfrac{\rho_T}{d}u & \dfrac{\rho_T}{d}v & \dfrac{\rho_T}{d}w & -\dfrac{\rho_T}{d} \\ -\dfrac{u}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0 \\ -\dfrac{v}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{w}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ \dfrac{-\rho_p(H - V^2) + 1 - \rho h_p}{d} & -\dfrac{\rho_p}{d}u & -\dfrac{\rho_p}{d}v & -\dfrac{\rho_p}{d}w & \dfrac{\rho_p}{d} \end{pmatrix}$$

where d is given by:

$$d = \rho\rho_p h_T + \rho_T(1 - \rho h_p)$$

The modified versions $\mathbf{M}_m$ and $\mathbf{M}_m^{-1}$ are obtained from the previous ones substituting ρ_T and ρ_p with ρ_T^m and ρ_p^m, so that d becomes d^m accordingly.

B Eigenvectors and eigenvalues

In section 2.2 the numerical inviscid flux evaluation scheme was introduced:

$$F^*_{i+1/2} = \frac{F_i + F_{i+1}}{2} - \frac{1}{2}\,\mathbf{P}^{-1}\,\mathbf{R}_p\,|\Lambda_p|\,\mathbf{L}_p\,(Q_{i+1} - Q_i) \tag{1}$$

where $\mathbf{R}_p$ and $\mathbf{L}_p$ represent the right and left eigenvector matrices in conservative variables of the preconditioned Euler matrix $\mathbf{D}_p$:

$$\mathbf{D}_p = \mathbf{PD} \equiv \mathbf{P}\,(\mathbf{A}n_x + \mathbf{B}\,n_y + \mathbf{C}n_z)$$

and Λ_p is a diagonal matrix whose elements are the eigenvalues of $\mathbf{D}_p$:

$$\lambda_1 = \lambda_2 = \lambda_3 = V_n \equiv un_x + vn_y + wn_z$$

$$\lambda_{4,5} = V_n\left(\frac{d + d^m}{2\,d^m}\right) \pm \sqrt{V_n^2\left(\frac{d - d^m}{2\,d^m}\right)^2 + \frac{\rho h_T}{d^m}}$$

Non-preconditioned eigenvalues are recovered when $d = d^m$ noticing that the speed of sound is given by $c = \sqrt{\rho h_T/d}$. Eigenvectors in the chosen primitive variables can be much simpler than the corresponding eigenvectors in conservative variables. Considering that:

$$\mathbf{L}_p\mathbf{D}_p\mathbf{R}_p \equiv \mathbf{L}_p\,(\mathbf{M}\mathbf{D}^v_p\mathbf{M}^{-1})\,\mathbf{R}_p \equiv (\mathbf{L}_p\mathbf{M})\,\mathbf{D}^v_p\,(\mathbf{M}^{-1}\mathbf{R}_p) \equiv \mathbf{L}^v_p\mathbf{D}^v_p\mathbf{R}^v_p = \Lambda_p$$

than:

$$\mathbf{P}^{-1}\,\mathbf{R}_p \equiv \mathbf{M}_m\mathbf{M}^{-1}\mathbf{R}_p \equiv \mathbf{M}_m\mathbf{R}^v_p$$

$$\mathbf{L}_p\,(Q_{i+1} - Q_i) \equiv \mathbf{L}^v_p\mathbf{M}^{-1}\,(Q_{i+1} - Q_i) \equiv \mathbf{L}^v_p\,(Q^v_{i+1} - Q^v_i) = W_{i+1} - W_i$$

where Q^v and W represent the primitive variables already introduced in section 1.3 and the characteristic variables respectively, and the superscript v refers to the eigenvector matrices in primitive variables. The numerical flux evaluation scheme becomes:

$$F^*_{i+1/2} = \frac{F_i + F_{i+1}}{2} - \frac{1}{2}\,\mathbf{M}_m\,\mathbf{R}^v_p\,|\Lambda_p|\,\mathbf{L}^v_p\,(Q^v_{i+1} - Q^v_i) \tag{2}$$

Equations 1 and 2 are equivalent: they both provide the fluxes (and so the residuals) for the conserved quantities (mass, momentum and total energy per unit volume). The former is expressed using the eigenvectors in conservative variables, the latter makes use of eigenvectors in primitive variables, which allows easier algebra and less intensive computational load.

The matrix of left eigenvectors in primitive variables is given by:

$$\mathbf{L}_p^v = \begin{pmatrix} -\frac{\rho_T}{d^m}\ n_x & 0 & \frac{\mathbf{A}}{\Delta^m}\ n_z & -\frac{\mathbf{A}}{\Delta^m}\ n_y & -\frac{\mathbf{A}}{T}\ n_x \\ -\frac{\rho_T}{d^m}\ n_y & -\frac{\mathbf{A}}{\Delta^m}\ n_z & 0 & \frac{\mathbf{A}}{\Delta^m}\ n_x & -\frac{\mathbf{A}}{T}\ n_y \\ -\frac{\rho_T}{d^m}\ n_z & \frac{\mathbf{A}}{\Delta^m}\ n_y & -\frac{\mathbf{A}}{\Delta^m}\ n_x & 0 & -\frac{\mathbf{A}}{T}\ n_z \\ \frac{1}{2} + \mathbf{B} - \mathbf{C}\ hp & \frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_x & \frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_y & \frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_z & \frac{1}{2}\mathbf{C}\ \rho h_T \\ \frac{1}{2} - \mathbf{B} + \mathbf{C}\ hp & -\frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_x & -\frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_y & -\frac{1}{2}\frac{\mathbf{A}}{\Delta^m}\ n_z & -\frac{1}{2}\mathbf{C}\ \rho h_T \end{pmatrix}$$

where:

$$hp = (1 - \rho\ h_p)$$

$$\Delta^m = \sqrt{V_n^2 \left(\frac{d - d^m}{2\ d^m}\right)^2 + \frac{\rho h_T}{d^m}}$$

$$\mathbf{A} = \frac{\rho^2\ h_T}{d^m}$$

$$\mathbf{B} = \frac{1}{4}\ \frac{V_n}{\Delta^m}\ \frac{d - d^m}{d^m}$$

$$\mathbf{C} = \frac{V_n}{\Delta^m}\ \frac{\rho_T - \rho_T^m}{d^m}$$

In case of no preconditioning:

$$d^m = d \quad \Longrightarrow \quad \Delta^m = \Delta = c \qquad \mathbf{A} = \rho\ c^2 \quad \text{and} \quad \mathbf{B} = \mathbf{C} = 0$$

It has to be noticed that, when deriving the eigenvector matrices, the elements (1,5), (2,5) and (3,5) of $\mathbf{L}_p^v$ take the following form:

$$\frac{\rho\ \rho_T\ h_T}{d^m\ (1 - \rho\ h_p)}$$

which would generate a 0/0 term when using incompressible fluids with $\beta = \chi = 0$. However, for any pure substance, the above term is equivalent to (see section 1.4):

$$\frac{\rho^2\ h_T}{d^m\ T}$$

In the matrix $\mathbf{L}_p^v$, the first three rows represent linear combinations of the entropy and the two shear waves, all of them propagating with characteristic speed given by $\lambda_{1,2,3} = V_n$. The last two rows represent the two acoustic waves with characteristic speeds given by λ_4 and λ_5. When the ideal gas law is considered (with no preconditioning), the well known left eigenvector matrix $\mathbf{L}^v$ in primitive variables is recovered:

$$\mathbf{L}^v = \begin{pmatrix} (\gamma-1)\ n_x & 0 & \rho\ c\ n_z & -\rho\ c\ n_y & -\dfrac{\rho\ c^2}{T}\ n_x \\ (\gamma-1)\ n_y & -\rho\ c\ n_z & 0 & \rho\ c\ n_x & -\dfrac{\rho\ c^2}{T}\ n_y \\ (\gamma-1)\ n_z & \rho\ c\ n_y & -\rho\ c\ n_x & 0 & -\dfrac{\rho\ c^2}{T}\ n_z \\ \dfrac{1}{2} & \dfrac{1}{2}\rho\ c\ n_x & \dfrac{1}{2}\rho\ c\ n_y & \dfrac{1}{2}\rho\ c\ n_z & 0 \\ \dfrac{1}{2} & -\dfrac{1}{2}\rho\ c\ n_x & -\dfrac{1}{2}\rho\ c\ n_y & -\dfrac{1}{2}\rho\ c\ n_z & 0 \end{pmatrix}$$

The matrix of $\mathbf{M}_m\ \mathbf{R}_p^v$ is given in the following page split into two pieces: the first piece reduces to $\mathbf{R}$, the Euler right eigenvector matrix in conservative variables, when the ideal gas law is applied and without preconditioning:

$$\mathbf{R} = \frac{1}{c^2}\begin{pmatrix} n_x & n_y & n_z & 1 & 1 \\ un_x & un_y - cn_z & un_z + cn_y & u + cn_x & u - cn_x \\ vn_x + cn_z & vn_y & vn_z - cn_x & v + cn_y & v - cn_y \\ wn_x - cn_y & wn_y + cn_x & wn_z & w + cn_z & w - cn_z \\ \begin{array}{c}\dfrac{V^2}{2}n_x + \\ c(vn_z - wn_y)\end{array} & \begin{array}{c}\dfrac{V^2}{2}n_y + \\ c(wn_x - un_z)\end{array} & \begin{array}{c}\dfrac{V^2}{2}n_z + \\ c(un_y - vn_x)\end{array} & H + cV_n & H - cV_n \end{pmatrix}$$

The second part of the matrix contains the extra terms due to the preconditioning, which all vanish as soon as the preconditioning is switched off, and where:[4]

$$\mathbf{D} = \frac{T}{\rho}V_n(\rho_T - \rho_T^m) \qquad \mathbf{E} = V_n(1 - \rho h_p)\frac{\rho_T - \rho_T^m}{d^m} \qquad \mathbf{F} = \frac{1}{2}V_n\frac{d - d^m}{d^m}$$

[4]All preconditioned matrices were derived with two modified parameters ρ_p^m and ρ_T^m. All calculations presented are however obtained with one modification only (with $\rho_T^m = \rho_T$ and $\mathbf{C} = \mathbf{D} = \mathbf{E} = 0$), for no effect of ρ_T^m has been observed.

$$
\mathbf{M}_m \mathbf{R}_p^v = \frac{d^m}{\rho h_T}
\left(
\begin{array}{ccccc}
-\dfrac{T\rho_T^m}{\rho} n_x & -\dfrac{T\rho_T^m}{\rho} n_y & -\dfrac{T\rho_T^m}{\rho} n_z & 1 & 1 \\
\hline
-\dfrac{T\rho_T^m}{\rho} u n_x & -\dfrac{T\rho_T^m}{\rho} u n_y - \Delta^m s_z & -\dfrac{T\rho_T^m}{\rho} u n_z + \Delta^m s_y & u + \Delta^m n_x & u - \Delta^m n_x \\
\hline
-\dfrac{T\rho_T^m}{\rho} v n_x + \Delta^m s_z & -\dfrac{T\rho_T^m}{\rho} v n_y & -\dfrac{T\rho_T^m}{\rho} v n_z - \Delta^m s_x & v + \Delta^m n_y & v - \Delta^m n_y \\
\hline
-\dfrac{T\rho_T^m}{\rho} w n_x - \Delta^m s_y & -\dfrac{T\rho_T^m}{\rho} w n_y + \Delta^m s_x & -\dfrac{T\rho_T^m}{\rho} w n_z & w + \Delta^m n_z & w - \Delta^m n_z \\
\hline
\begin{array}{c} -T\left(\dfrac{H\rho_T^m}{\rho} + h_T\right) n_x \\ +\Delta^m (v n_z - w n_y) \end{array} &
\begin{array}{c} -T\left(\dfrac{H\rho_T^m}{\rho} + h_T\right) n_y \\ +\Delta^m (w n_x - u n_z) \end{array} &
\begin{array}{c} -T\left(\dfrac{H\rho_T^m}{\rho} + h_T\right) n_z \\ +\Delta^m (u n_y - v n_x) \end{array} &
H + \Delta^m V_n & H - \Delta^m V_n
\end{array}
\right) +
$$

$$
+ \quad \frac{d^m}{\rho h_T}
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
\mathbf{D} n_x^2 & \mathbf{D} n_x n_y & \mathbf{D} n_x n_z & (\mathbf{E} - \mathbf{F}) n_x & (\mathbf{E} - \mathbf{F}) n_x \\
\mathbf{D} n_x n_y & \mathbf{D} n_y^2 & \mathbf{D} n_y n_z & (\mathbf{E} - \mathbf{F}) n_y & (\mathbf{E} - \mathbf{F}) n_y \\
\mathbf{D} n_x n_z & \mathbf{D} n_y n_z & \mathbf{D} n_z^2 & (\mathbf{E} - \mathbf{F}) n_z & (\mathbf{E} - \mathbf{F}) n_z \\
\mathbf{D} V_n n_x & \mathbf{D} V_n n_y & \mathbf{D} V_n n_z & (\mathbf{E} - \mathbf{F}) V_n & (\mathbf{E} - \mathbf{F}) V_n
\end{pmatrix}
$$