

FINITE VOLUME METHOD

Enrico Nobile

Dipartimento di Ingegneria e Architettura
Università degli Studi Trieste, 34127 TRIESTE



3 marzo 2024



OUTLINE

Parte I

Generalities

- 1 INTRODUCTION
- 2 THE CONSERVATION EQUATION
- 3 THE BASIC IDEA
 - The computational grid
 - Cartesian grids for the finite volume method



Finite Volume Method for Incompressible fluids

- Cartesian grids for two-dimensional (2D) domains:
 - ▶ Generic conservation equation
 - ▶ Complete fluid flow-thermal problem
 - ▶ Modifications required for Cartesian grids in three-dimensional (3D) domains
 - ▶ One-dimensional (1D) case: elementary derivation left to the reader
- Unstructured grids:
 - ▶ Generic conservation equation
 - ▶ Geometric quantities and variables arrangement on the grid
 - ▶ Gradient reconstruction
 - ▶ Hybrid grids



Finite Volume Method (FVM)

- Algebraic equations obtained by enforcing conservation for each Control Volume (CV) or Cell;
- Popular method in Computational Fluid Dynamics and Numerical Heat Transfer (CFD);
- Used in several commercial and open source CFD packages;
- Simplicity and physical correspondance between FVM and conservation principles;
- Frequently adopted, also at undergraduate level, for the introduction of numerical techniques in Heat Transfer and fluid dynamics.



Comparison of FV and FD

Finite Difference method	Finite Volume method
Non inherently conservative	Inherently Conservative
Close to the equations	Physically meaningful
Simple geometries (in standard form)	Suitable for complex geometries
Derivation and interpolation	Integration and deconvolution
Higher order derivatives	Lower order derivatives
High order methods	High order difficult



All conservation (transport) equations, with the exception of the mass conservation equation, have a similar structure:

- Energy equation in conservative form:

$$\frac{\partial}{\partial \vartheta} (\rho c_p t) + \nabla \cdot (\rho \mathbf{w} c_p t) = \nabla \cdot (\lambda \nabla t) + \dot{q}$$

- Momentum equation in conservative form:

$$\frac{\partial}{\partial \vartheta} (\rho \mathbf{w}) + \nabla \cdot (\rho \mathbf{w} \mathbf{w}) = -\nabla p + \nabla \cdot (\mu \nabla \mathbf{w}) - \rho \beta (t - t_0) \mathbf{g}$$

General form of transport equation:

$$\frac{\partial}{\partial \vartheta} (\rho \phi) + \nabla \cdot (\rho \mathbf{w} \phi) = \nabla \cdot (\Gamma \nabla \phi) + s$$

ϕ is a generic scalar variable and Γ is the molecular transport property.



In other words:

$$(Accumulation) + (Convection) = (Diffusion) + (Source/Sink)$$

- Energy equation: for incompressible fluids with constant c , ϕ is the specific enthalpy (ct), Γ is λ/c , and the source term, s , includes the possible internal heat generation.
- Momentum equation: ϕ is the velocity component, Γ is the molecular dynamic viscosity μ and the source term includes, besides the component of the pressure gradient, also the buoyancy force, is present, and all other possible force fields.
- It will be sufficient to consider first the generic transport equation, and then the Navier-Stokes and continuity equations:
 - ▶ The velocity field \mathbf{w} , the thermophysical properties ρ and Γ , and the source term s will be assumed known.
 - ▶ The procedure described maintains its validity even when the velocity field is unknown, like in the case of the Navier-Stokes equations.



Use of the integral, or *finite*, formulation of the conservation equation, written for a generic *control volume* V :

$$\int_V \left[\frac{\partial}{\partial \vartheta} (\rho\phi) + \nabla \cdot (\rho\mathbf{w}\phi) - \nabla \cdot (\Gamma\nabla\phi) - s \right] dV = 0$$

Applying Gauss's theorem (also known as Ostrogradsky's theorem,.) with A *surface boundary* of the control volume V :

$$\int_V \frac{\partial}{\partial \vartheta} (\rho\phi) dV + \int_A \rho\phi\mathbf{w} \cdot \mathbf{n} dA = \int_A \Gamma\nabla\phi \cdot \mathbf{n} dA + \int_V s dV$$

ed in a more compact form:

$$\int_V \frac{\partial}{\partial \vartheta} (\rho\phi) dV + \int_A \mathbf{J}'' \cdot \mathbf{n} dA = \int_V s dV$$



With \mathbf{J}'' we indicate the *flux density* vector of the variable ϕ . It includes both the convective flux density, \mathbf{J}_c'' , and the diffusive flux density, \mathbf{J}_d'' . Therefore, $J'' = \mathbf{J}'' \cdot \mathbf{n}$ represents the flux density component normal to the surface:

$$\mathbf{J}'' = \mathbf{J}_c'' - \mathbf{J}_d'' = \rho\phi\mathbf{w} - \Gamma\nabla\phi$$

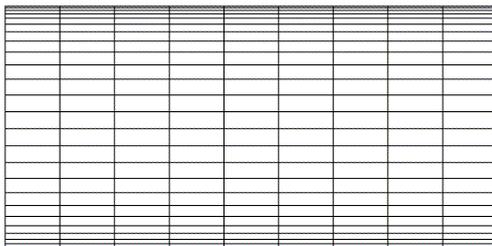
$$J'' = \mathbf{J}'' \cdot \mathbf{n} = \rho\phi\mathbf{w} \cdot \mathbf{n} - \Gamma\nabla\phi \cdot \mathbf{n}$$



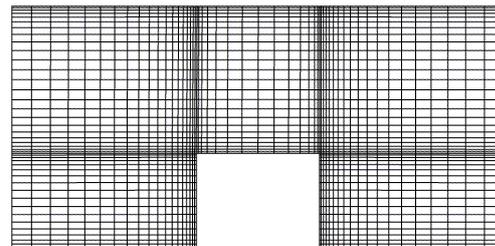
Structured Cartesian grids

- Constituted by families of, mutually orthogonal, parallel lines.
- The lines of each family, or the *cells* defined by these lines, are identified by a set of two indices (i, j) in 2D, or three indices (i, j, k) in 3D.
- Limited geometrical flexibility.
- Simplicity and efficiency of computing methods based on such grids.

Examples of structured Cartesian grids:



Single-block



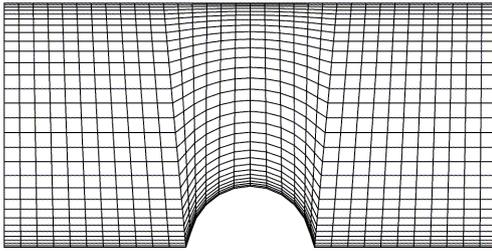
Multi-block



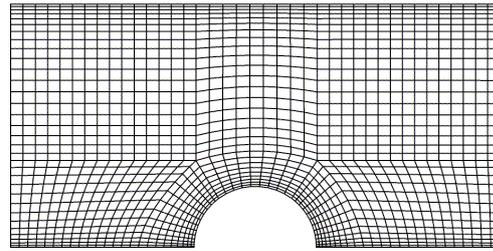
Structured curvilinear (boundary-fitted) grids

- Constituted by families of curvilinear lines, such that each line of a family will never intersect a line of the same family, and will cross only once the lines of the other families.
- They are identical to the Cartesian grids from the logical - e.g. indexing, data-structure - point of view.
- Greater geometrical flexibility compared to Cartesian grids.
- Special case: the families of the lines are mutually orthogonal.

Examples of structured curvilinear grids:



Single-block



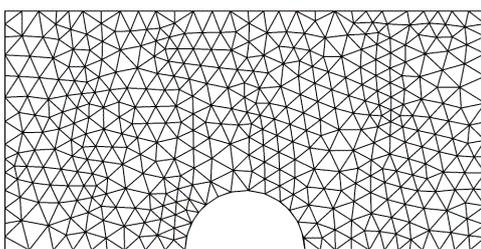
Multi-block



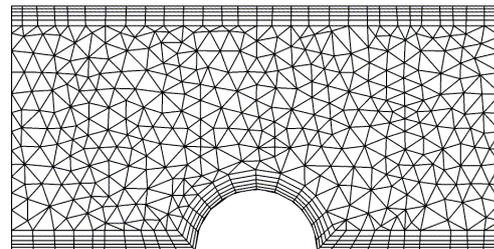
Unstructured grids

- Particularly suited to address complex geometries of industrial interest.
- The spatial domain is discretized into cells of arbitrary shape, e.g. *polyhedra*, or more typically triangles and quadrilaterals in 2D, and tetrahedra and hexaedra in 3D.
- Possibility to refine the grid, also automatically (*adaptive grids*), in specific zones of the domain.
- Greater complexity and computational costs, e.g. data-structures, memory requirements, limited choice of linear solvers.

Examples of unstructured grids:



Triangle-based grid

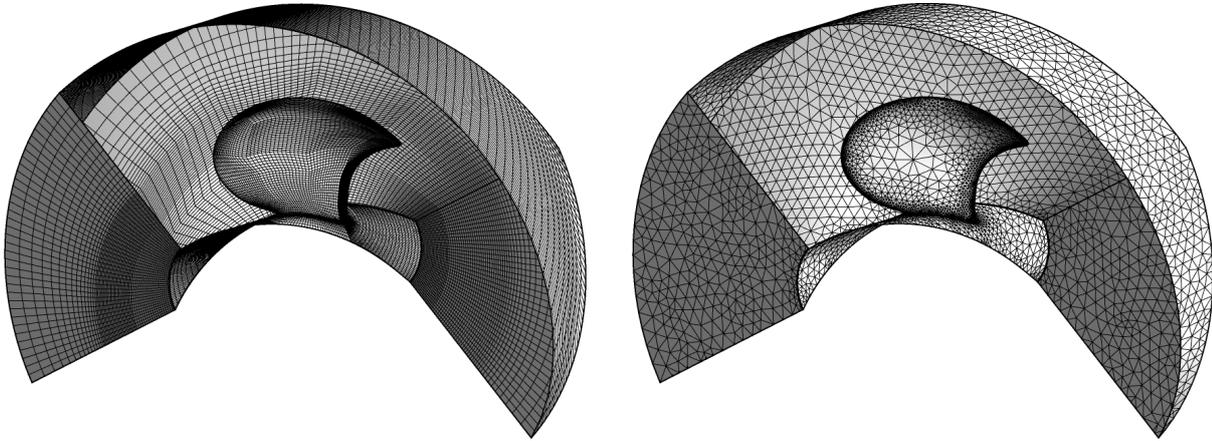


Hybrid grid



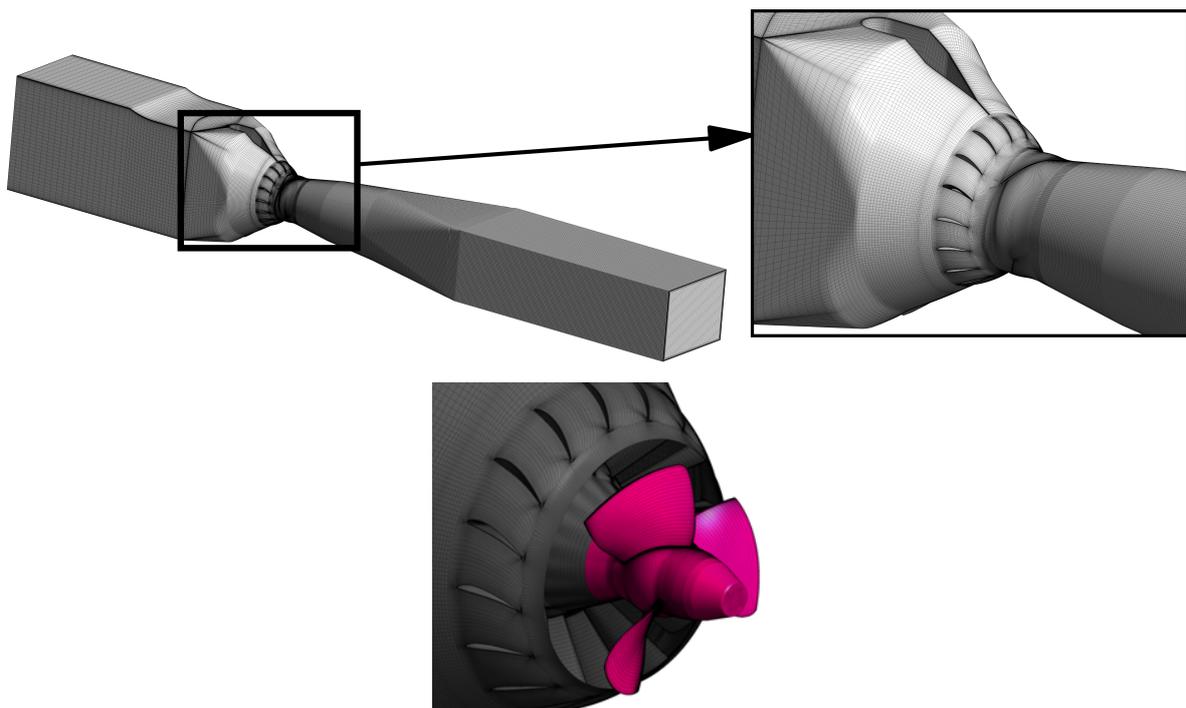
Some examples

Hexa and hybrid meshing for a naval propeller (M. Morgut, 2009).



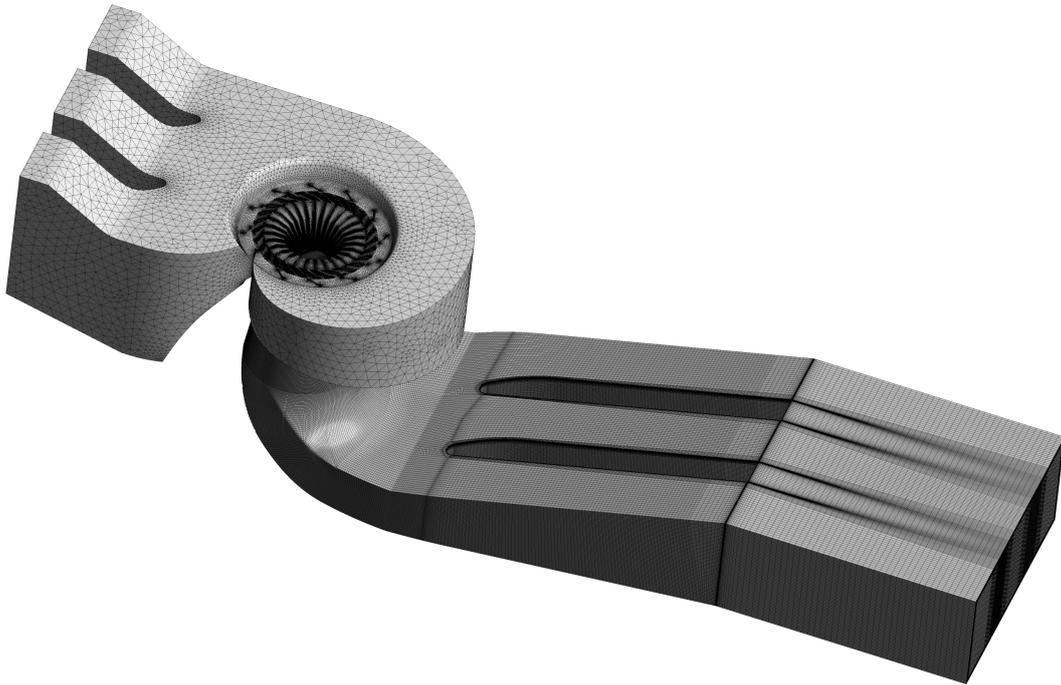
Some examples - cont.

Hexa meshing for a bulb turbine (ACCUSIM project, www.accusim.eu)



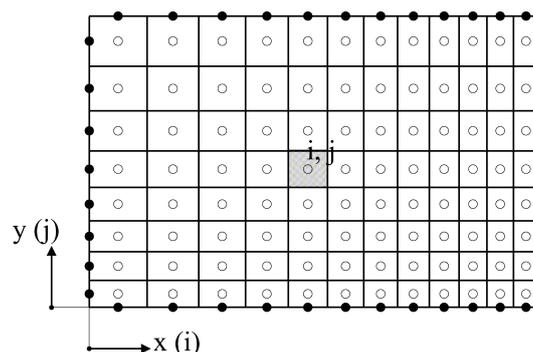
Some examples - cont.

Mixed – hybrid-hexa – meshing for a Kaplan turbine (ACCUSIM project, www.accusim.eu)



Cartesian grids for the finite volume method

- The physical domain is subdivided in a finite number of control volumes (CV), as illustrated in the figure.
- The grid, differently from the *Finite Difference* method, defines the *faces*, and not the *nodes*, of the grid, although the variables are still collocated in the nodes.
- The most common approach, which is also adopted for cell-centered FV unstructured grids, is to place the nodes in the centroids of the CVs.
- The nodes at the centre of the CVs are indicated with 'o', while the nodes at the boundary are represented by '•'.



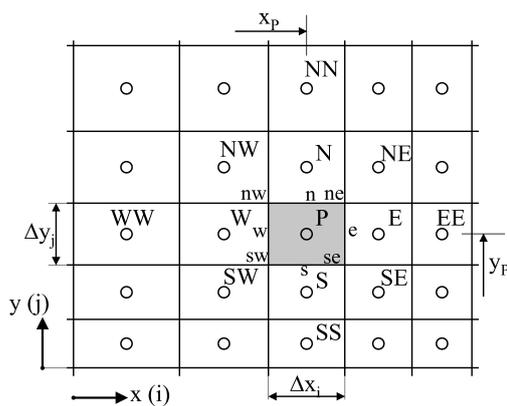
Fundamentals

- 4 SPATIAL DISCRETIZATION
 - Order of Accuracy
- 5 THE STEADY CASE
 - Surface integrals
 - Volume integrals
 - Source term
- 6 INTERPOLATION TECHNIQUES
 - Diffusive Flux
 - Convective Flux
 - Deferred Correction
- 7 FINAL ALGEBRAIC EQUATION
 - Examples of discretized equation
- 8 BOUNDARY CONDITIONS
- 9 UNDER-RELAXATION

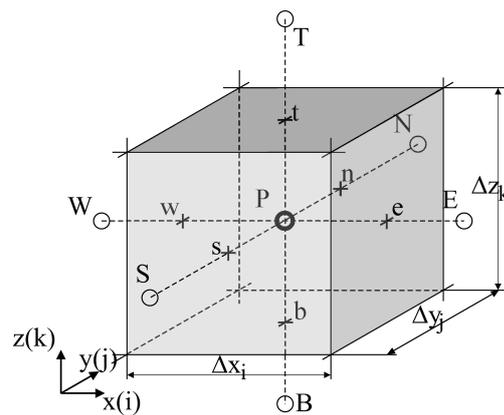


Notation

Typical CVs for a Cartesian grid, together with the adopted nomenclature:



Two-dimensional



Three-dimensional

- The 2D case can be seen as a special 3D case where all variables are independent of the z coordinate.
- Attention paid to the 2D case: simple extension, for Cartesian grids, to the 3D case.



Spatial variation approximation

- As will be seen later, fluxes at the faces and sources within the cell are evaluated using the *mean value approach*, i.e. using the value of ϕ at the centroid of the surface (midpoint rule) and cell, respectively.
- This approach, in addition to the *assumed* variation of ϕ in space around point P , determine the accuracy of the entire discretization procedure.
- In the *standard* Finite Volume method, ϕ is assumed to vary linearly in space, i.e.

$$\phi(\mathbf{x}) = \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla\phi)_P \quad \text{where } \phi_P = \phi(\mathbf{x}_P)$$

- ϕ_P represents the mean value of ϕ within the cell

$$\phi_P = \bar{\phi} = \frac{1}{V} \int_V \phi(\mathbf{x}) \, dV$$



Spatial variation approximation - *cont.*

- However, the spatial variation of ϕ within an element can be described via a Taylor series expansion around \mathbf{x}_P as

$$\begin{aligned} \phi(\mathbf{x}) = & \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla\phi)_P + \frac{1}{2} (\mathbf{x} - \mathbf{x}_P)^2 : (\nabla\nabla\phi)_P \\ & + \frac{1}{3!} (\mathbf{x} - \mathbf{x}_P)^3 :: (\nabla\nabla\nabla\phi)_P + \dots \\ & + \frac{1}{n!} (\mathbf{x} - \mathbf{x}_P)^n \underbrace{:: \dots ::}_{(n-1) \text{ times}} \left(\underbrace{\nabla\nabla \dots \nabla\phi}_{n \text{ times}} \right)_P + \dots \end{aligned}$$

- The term $(\mathbf{x} - \mathbf{x}_P)^n$ in the equation represents the n th tensorial product of the vector $(\mathbf{x} - \mathbf{x}_P)$ with itself, producing an n th tensor.
- The operator $(:)$ is the inner product of two 2nd rank tensors, $(::)$ is the inner product of two 3rd rank tensors, and $(::: \dots ::)$ is the inner product of two n th rank tensors, all yielding a *scalar*.
- The comparison between this Taylor series expansion, and the previous assumed variation of ϕ within the cell, indicates an error proportional to $|(\mathbf{x} - \mathbf{x}_P)^2|$, implying a *second order* accuracy.



The conservation equation, written in integral form, can be applied to any control volume, and in particular to every FV (cell) or to the entire domain:

- Summing up the equations obtained for all FVs, we are left with the global conservation equation, since fluxes over internal faces will cancel out: they will be evaluated with different sign for the two adjacent FVs.
- This property, however, holds only if the procedure for the evaluation of the face flux is unique, e.g. it is the same for both adjacent FVs.
- Although this property (*telescopic property*) seems obvious, in case of arbitrary unstructured *collocated* grids it is necessary to adopt some specific measures, in order to satisfy this rule, which constitutes the base, and the principal advantage, of the Finite Volume method.



The steady case

- Numerical approximation of surface and volume integrals that appear in the integral form of the conservation equation.
- Adoption of interpolation techniques, in order to evaluate some quantities in locations different from those where they are defined.

Conservation equation for the steady case

$$\int_A \rho \phi \mathbf{w} \cdot \mathbf{n} \, dA = \int_A \Gamma \nabla \phi \cdot \mathbf{n} \, dA + \int_V s \, dV$$



Net flux through the boundary of the CV

$$\begin{aligned}\int_A \mathbf{J}'' \cdot \mathbf{n} \, dA &= \sum_k \int_{A_k} \mathbf{J}'' \cdot \mathbf{n} \, dA \\ &= \sum_k \int_{A_k} \mathbf{J}_c'' \cdot \mathbf{n} \, dA - \sum_k \int_{A_k} \mathbf{J}_d'' \cdot \mathbf{n} \, dA\end{aligned}$$

with

$$\mathbf{J}'' = \mathbf{J}_c'' - \mathbf{J}_d'' = \rho\phi\mathbf{w} - \Gamma\nabla\phi$$

where \mathbf{J}'' is the overall specific flux, and \mathbf{J}_c'' and \mathbf{J}_d'' are the specific convective and diffusive fluxes, respectively.

- In order to guarantee *local* conservativity at the CV level, and *global* conservativity for the entire domain, CVs should not overlap:
 - ▶ Every face belongs to one CV, if it lies on the boundary, or belongs to two CVs if it is internal to the domain.
 - ▶ This property should be guaranteed also for more complex cases, like e.g. *sliding grids*, *overset (chimera) grids* et.
- In the following we will consider just one face, the one indicated with “e” in the figure; similar expressions can be derived for the other faces by proper index substitution.



Approximations

- To calculate the surface integral on face *e* exactly, it would be necessary to know the integrand ($\mathbf{J}'' \cdot \mathbf{n}$) everywhere on the face.
- This is not possible, since the variable ϕ , and therefore its associated fluxes, are known only on the nodes (centers) of the CV.
- Therefore, two distinct approximations are required:
 - ① The integral is approximated in terms of one, or more, variable value on the face;
 - ② The variable values on the cell face are approximated in terms of nodal (center) values.
- Referring later to the approximate derivation of face values in terms of node (center) variable values, we look first at the approximate evaluation of the integrals.



Approximations of surface integrals

Midpoint rule

$$F_e = \int_{A_e} \mathbf{J}'' \cdot \mathbf{n} \, dA = \overline{J''}_e A_e = \overline{J''}_e \Delta y_j \approx J''_e \Delta y_j$$

Trapezoidal rule

$$F_e = \int_{A_e} \mathbf{J}'' \cdot \mathbf{n} \, dA \approx \frac{\Delta y_j}{2} (J''_{ne} + J''_{se})$$

Higher order approximations: Simpson's rule

$$F_e = \int_{A_e} \mathbf{J}'' \cdot \mathbf{n} \, dA \approx \frac{\Delta y_j}{6} (J''_{ne} + 4J''_e + J''_{se})$$



Approximations of volume integrals

Midpoint rule (2^{nd} order)

$$S_P = \int_V s \, dV = \bar{s} \Delta V \approx s_P \Delta V$$

which in 2d can be expressed as:

$$S_P \approx s_P \Delta x_i \Delta y_j$$

Higher order approximation (4^{th} order)

$$S_P = \int_V s \, dV \approx \Delta x_i \Delta y_j \left[a_0 + \frac{a_3}{12} (\Delta x_i)^2 + \frac{a_4}{12} (\Delta y_j)^2 + \frac{a_8}{144} (\Delta x_i)^2 (\Delta y_j)^2 \right]$$

that, for uniform Cartesian grids, where $\Delta x_i \equiv \Delta x$ e $\Delta y_j \equiv \Delta y$, became:

$$S_P \approx \frac{\Delta x \Delta y}{36} \left[16 s_P + 4 s_s + 4 s_n + 4 s_w + 4 s_e + s_{ne} + s_{se} + s_{nw} + s_{sw} \right]$$



Source term integration

- Meaning of the source term s :
 - ▶ In the source term one should insert all other contributions that can not be part of transport (advection and diffusion) and unsteady terms.
- If s_P is known and it *not* depends on ϕ , there are no difficulties.
- If, as usual, it depends on the variable ϕ , it must be linearized:
 - ▶ It is appropriate - and convenient - to take into account such dependence in the formulation of the discretized equation;
 - ▶ Such dependence can be, at most, *linear*, since the final result will be incorporated in a linear system of equations.

Source term linearization

$$s_P = s_P^{rhs} + s_P^{lhs} \phi_P$$

with

$$s_P^{lhs} < 0 \text{ e } s_P^{rhs} > 0.$$



Source term linearization

- If s_P is a non-linear function of ϕ , it is necessary to linearize it, i.e. specify the values of s_P^{lhs} e s_P^{rhs} , which may themselves depend on ϕ :
 - ▶ At every iteration, or time-step, s_P^{lhs} e s_P^{rhs} would then be recomputed from the new values (*in store*) of ϕ_P ; in the following we indicate these, for brevity, as ϕ_P^* (ϕ_P^k or ϕ_P^n);
 - ▶ The chosen linearization for s_P should be a good representation of the $s_P = s_P(\phi_P)$ relationship.
- The basic rule $s_P^{lhs} < 0$ should be always be observed:
 - ▶ This rule would not be required if the final system of linear equations, at every iterative cycle or time-step, would be solved by a *direct* method.
 - ▶ Vice versa, this rule is of paramount importance if, as it is (almost) always the case in CFD, the solution of the linear system of equations is sought by an *iterative* method.

Let see some examples in the following.



Examples (1)

Example 1

Given:

$$s = 6 - 3\phi$$

let's see some possible linearizations:

- 1 $s_P^{rhs} = 6; s_P^{lhs} = -3$. Most obvious and recommended.
- 2 $s_P^{rhs} = 6 - 3\phi_P^*; s_P^{lhs} = 0$. Not efficient, but sometimes it represents the only choice for complex expressions of the source term (e.g. turbulence models).
- 3 $s_P^{rhs} = 6 + 5\phi_P^*; s_P^{lhs} = -8$. This proposed relationship is steeper than that given (remember that ϕ_P^* should be considered, for all purposes, a *constant*). It corresponds to an *under-relaxation*, and therefore it will lead to a slow down of the convergence of the (external) iterative cycle (see later). It is suitable if there are other, stiff non-linearities in the problem.



Examples (2)

Example 2

Given:

$$s = 4 + 11\phi$$

some possible linearizations are:

- 1 $s_P^{rhs} = 4; s_P^{lhs} = 11$. In general not acceptable, since it will make $s_P^{lhs} > 0$.
- 2 $s_P^{rhs} = 4 + 11\phi_P^*; s_P^{lhs} = 0$. This is the correct choice, since it is impossible to find a *natural* formulation which gives $s_P^{lhs} < 0$.
- 3 $s_P^{rhs} = 4 + 15\phi_P^*; s_P^{lhs} = -4$. This corresponds to an artificial creation of $s_P^{lhs} < 0$. It will slow down the convergence, but it can also be useful (robustness, reduced risks of divergence of the simulation).



Examples (3)

Example 3

Given:

$$s = 3 - 5\phi^3$$

- 1 $s_P^{rhs} = 3 - 5\phi_P^{*3}$; $s_P^{lhs} = 0$. Inefficient, does not take advantage of the dependence of S on ϕ .
- 2 $s_P^{rhs} = 3$; $s_P^{lhs} = -5\phi_P^{*2}$. This seems correct, but the given relationship $s_P = s_P(\phi_P)$ is steeper than this linearization.
- 3 Recommended method:

$$s = s^* + (ds/d\phi)_* (\phi_P - \phi_P^*) = 3 - 5\phi_P^{*3} - 15\phi_P^{*2} (\phi_P - \phi_P^*)$$

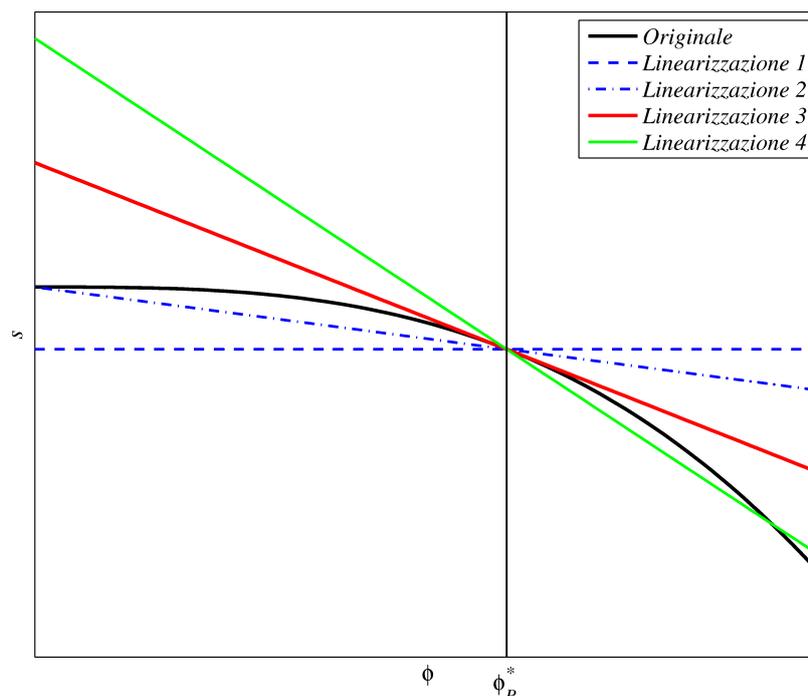
thus: $s_P^{rhs} = 3 + 10\phi_P^{*3}$; $s_P^{lhs} = -15\phi_P^{*2}$: tangent, in ϕ_P^* , to the curve $s_P = s_P(\phi_P)$.

- 4 $s_P^{rhs} = 3 + 20\phi_P^{*3}$; $s_P^{lhs} = -25\phi_P^{*2}$. This linearization is steeper than that given and it would slow down the convergence.

The four linearizations are illustrated in the next slide, together with the actual $s_P = s_P(\phi_P)$ curve.



Examples (4)



- The evaluation of the integrals requires the knowledge of the variables in locations different than those where they are defined, i.e. CV centers.
- The integrand is the product of several variables and/or their gradients: $J_c'' = \rho \phi \mathbf{w} \cdot \mathbf{n}$ for the convective - sometimes called *advective* - flux, and $J_d'' = \Gamma \nabla \phi \cdot \mathbf{n}$ for the diffusive flux.
- In order to compute these fluxes, it is first necessary to define the value of ϕ , and the face-normal component of its gradient, in one or more points of the face.
- Distinction between the face-normal component of the gradient - *diffusive flux* - and value of the variable - *convective flux*.



Schemes for the diffusive flux

CDS (Central Difference Scheme) - 2nd order

$$\phi_e = \phi_E \lambda_{e,PE} + \phi_P (1 - \lambda_{e,PE})$$

$$\lambda_{e,PE} = \frac{x_e - x_P}{x_E - x_P}$$

Face-normal component of the gradient:

$$\left(\frac{\partial \phi}{\partial x} \right)_e \approx \frac{\phi_E - \phi_P}{x_E - x_P}$$

CDS - 4th order

$$\phi(x) = a + b x + c x^2 + d x^3$$

For uniform grids:

$$\left(\frac{\partial \phi}{\partial x} \right)_e \approx \frac{27 \phi_E - 27 \phi_P + \phi_W - \phi_{EE}}{24 \Delta x}$$



- Use of 2^{nd} order schemes for the diffusive fluxes;
- Higher order schemes increase the computational cost: larger *computational molecule*;
- Higher order schemes are tendentially *less robust*;
- Use of the *deferred correction*;
- High order approximations do not necessarily guarantee higher accuracy:
 - ▶ The grid should be *sufficiently fine*;
 - ▶ Resort to *grid refinement*.



Diffusive flux

Face diffusivity

Value of face diffusivity Γ_e when the diffusivity exhibits spatial variations:

$$\Gamma_e = \left[\frac{\lambda_{e,PE}}{\Gamma_P} + \frac{(1 - \lambda_{e,PE})}{\Gamma_E} \right]^{-1}$$

For a uniform grid: *harmonic mean* of Γ_P e Γ_E :

$$\Gamma_e = \frac{2\Gamma_E\Gamma_P}{\Gamma_E + \Gamma_P}$$

Diffusive flux density

$$J''_{d,e} = (\Gamma \nabla \phi \cdot \mathbf{n})_e = \Gamma_e \left(\frac{\partial \phi}{\partial x} \right)_e$$



Derivation of the face diffusivity

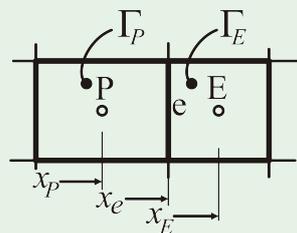
The previous expression for Γ_e can be easily found by extending the *electric analogy*, used in heat transfer, to a generic diffusion problem.

We can in fact express the specific *diffusive resistance* of a homogeneous layer of unitary area, having diffusivity Γ and thickness L , as

$$R''_{\phi} = \frac{L}{\Gamma} \quad \Rightarrow \quad q'' = \frac{\phi_P - \phi_E}{R''_{\phi}}$$

with q'' diffusive flux density.

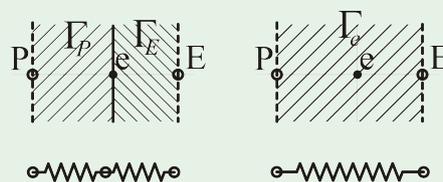
For materials with variable diffusivity, Γ_P and Γ_E represent the (average) diffusivity of cell P and cell E , respectively.



Equivalent diffusivity - cont.

Derivation of the face diffusivity - cont.

In our case, the diffusive resistance between P and E is given by the *series* of two diffusive resistances: therefore, the equivalent diffusivity must be such as to give rise to the same specific resistance and therefore of the flux density:



$$R''_{\phi} = \frac{x_e - x_P}{\Gamma_P} + \frac{x_E - x_e}{\Gamma_E} = \frac{x_E - x_P}{\Gamma_e}$$

and solving for Γ_e :

$$\Gamma_e = \frac{1}{\frac{\lambda_{e,PE}}{\Gamma_P} + \frac{(1 - \lambda_{e,PE})}{\Gamma_E}}$$



Basic schemes for the convective flux

Convective flux density:

$$J''_{c,e} = (\rho \phi \mathbf{w} \cdot \mathbf{n})_e = \rho \phi_e u_e$$

u_e is the velocity component normal to the face e , and it is assumed to be known.

CDS - Central Difference Scheme

$$\phi_e = \phi_E \lambda_{e,PE} + \phi_P (1 - \lambda_{e,PE})$$

UDS - Upwind Difference Scheme

$$\phi_e = \begin{cases} \phi_P & \text{se } (\mathbf{w} \cdot \mathbf{n})_e > 0 \\ \phi_E & \text{se } (\mathbf{w} \cdot \mathbf{n})_e < 0 \end{cases}$$



Numerical diffusion

- The UDS scheme guarantees absence of unphysical oscillations (dispersion), but it introduces a relevant *numerical diffusion*.
- Assuming $(\mathbf{w} \cdot \mathbf{n})_e > 0$:

$$\phi_e = \phi_P + (x_e - x_P) \left(\frac{\partial \phi}{\partial x} \right)_P + \frac{(x_e - x_P)^2}{2} \left(\frac{\partial^2 \phi}{\partial x^2} \right)_P + \dots$$

The UDS scheme contains only the first term on the right and therefore it is a first order scheme. The truncation error (second term on the right) reminds the expression for the diffusive flux.

- The UDS scheme, therefore, introduces a *false diffusion flux* given by:

$$J''_{Num} = \Gamma_{Num} \left(\frac{\partial \phi}{\partial x} \right)_P \quad \text{con} \quad \Gamma_{Num} = \rho u_e \frac{\Delta x_P}{2}$$

- Moreover, the artificial diffusion increases when the streamlines are not aligned with the grid lines.



Why the UDS scheme (or similar schemes like i.e. HY - Hybrid, which uses the CDS scheme in cases where the convective flow is modest, switching to UDS in other cases) is still available in most commercial and open source CFD products?

- Simplicity and absence of oscillations: more guarantees to obtain a preliminary (although very approximate, and frequently of no physical value) solution of the problem;
- Start of the simulation in difficult (stiff) cases, then continuing with more accurate schemes;
- Use of higher order schemes in *deferred correction* mode.



High order schemes for the convective flux

SOUDS - Second Order Upwind Difference Scheme

$$\phi_e = \begin{cases} \phi_P + [\lambda_{e,PW} (\phi_W - \phi_P)] & \text{if } (\mathbf{w} \cdot \mathbf{n})_e > 0 \\ \phi_E + [\lambda_{e,EE} (\phi_{EE} - \phi_E)] & \text{if } (\mathbf{w} \cdot \mathbf{n})_e < 0 \end{cases}$$

$$\lambda_{e,PW} = \frac{x_e - x_P}{x_W - x_P}; \quad \lambda_{e,EE} = \frac{x_e - x_E}{x_{EE} - x_E}$$

In case $(\mathbf{w} \cdot \mathbf{n})_e > 0$:

$$\phi_e = \phi_P + \frac{\phi_W - \phi_P}{x_W - x_P} (x_e - x_P)$$

QUICK - Quadratic Upstream Interpolation for Convective Kinematics

$$\phi_e = \begin{cases} \phi_P + [\gamma_{1e}\phi_W - (\gamma_{1e} + \gamma_{2e})\phi_P + \gamma_{2e}\phi_E] & \text{if } (\mathbf{w} \cdot \mathbf{n})_e > 0 \\ \phi_E + [\gamma_{3e}\phi_P - (\gamma_{3e} + \gamma_{4e})\phi_E + \gamma_{4e}\phi_{EE}] & \text{if } (\mathbf{w} \cdot \mathbf{n})_e < 0 \end{cases}$$

$$\gamma_{1e} = \lambda_{e,EW} \lambda_{e,PW} \quad \gamma_{2e} = \lambda_{e,PE} \lambda_{e,WE}$$

$$\gamma_{3e} = \lambda_{e,EP} \lambda_{e,EEP} \quad \gamma_{4e} = \lambda_{e,PEE} \lambda_{e,EEE}$$



Some considerations about high order schemes for the convective flux

- Although more *robust* than the CDS scheme, and in the mean time more *accurate* of the UDS scheme, high-order schemes may however give rise to non-physical oscillations of the solution (wiggles).
- To make high-order schemes more stable and oscillation-free, the most common approaches can be grouped into two classes, i.e. *flux blending methods* and *composite flux limiter methods*.
- *Flux blending methods*:
 - ▶ An *anti-diffusive* flux is added to a 1st order upwind scheme (eg. FCT - Flux Corrected Transport).
 - ▶ A certain amount of artificial diffusivity is added - in a selective way - to a high-order scheme to dampen oscillations. (eg. FRAM - Filtering Remedy and Methodology).

- *Composite flux limiter methods*.

The flux on the cell-face is modified via a criterion that imposes a limit value (boundedness):

- ▶ Total Variational Diminishing (TVD) flux limiters.
- ▶ Normalized Variable Formulation (NVF), or Normalized Variable and Space Formulation (NVSF).



Other high order schemes

- On this basis some other high-order schemes have been developed:
 - ▶ MSOU - Monotonic Second Order Upwind scheme;
 - ▶ HLP - Hybrid Linear/Parabolic Approximation;
 - ▶ SHARP - Simple High-Accuracy Resolution Program);
 - ▶ SMART - Sharp and Monotonic Algorithm for Realistic Transport;
 - ▶ NIRVANA - Nonoscillatory, Integrally Reconstructed Volume-Averaged Numerical Advection.
- Drawbacks of such schemes:
 - ▶ Large size of the calculation molecule, unless one uses the *deferred correction*;
 - ▶ Difficult extension to unstructured grids.



Example - description

Performance of some convective schemes for the test of the purely convective, two-dimensional transport for a step profile.

- Imposed and constant flow field, inclined with an angle $\alpha = \pi/4$ respect to the grid.
- Boundary conditions are indicated in the figure, and the transport equation considered is $\nabla \cdot (\rho \mathbf{w} \phi) = 0$ which becomes

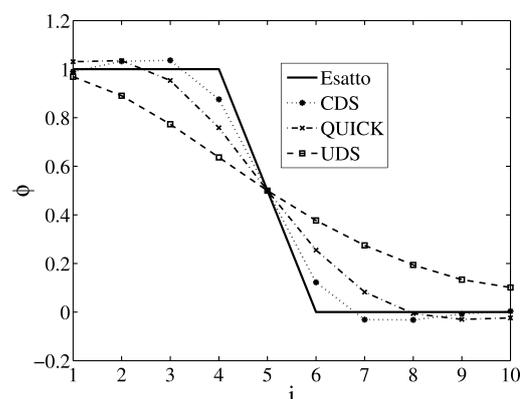
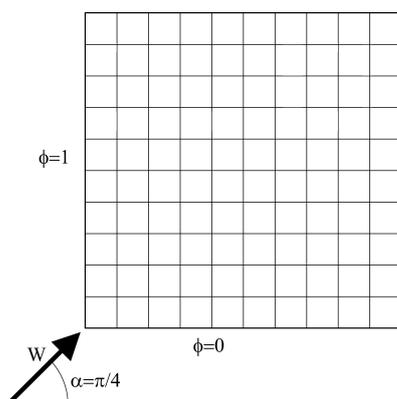
$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = 0$$

- Due to lack of diffusion, the *step* of the ϕ profile at the inlet, should be ideally transported, without diffusion or wiggles, indefinitely.
- The presence of artificial diffusion, or/and wiggles, may alter the result.



Example - results

I profili, ottenuti sulla sezione orizzontale posta a metà altezza, sono confrontati con il profilo *esatto*, cioè quello più accurato ottenibile con la griglia utilizzata.



- Lo schema UDS non dà luogo a oscillazioni, ma altera completamente il profilo, a seguito della diffusività numerica che lo caratterizza.
- Lo schema CDS preserva in modo accettabile il profilo (la griglia è particolarmente rada), ma introduce delle oscillazioni.
- Lo schema QUICK ha un comportamento intermedio fra UDS e CDS.



Deferred correction

- Difficulties in the use of high order schemes.
- Insufficient accuracy of simple (low order) schemes.

Deferred Correction

Indicating with k the k -th iteration we have

$$\phi_e^k = \underbrace{\phi_e^{ob, k}}_{lhs} + \underbrace{\left[\phi_e^{oe, k-1} - \phi_e^{ob, k-1} \right]}_{rhs}$$

ob indicates a *low order* scheme (e.g. UDS) and *oe* indicates a *high order* one (e.g. QUICK).

- Formulation of the expressions for SOUDS and QUICK.
- Low computational cost, better convergence, reduced possibility of non-physical oscillations of the solution.
- An iterative procedure already present for the solution of (non-linear) thermo-fluid problems, or, equivalently, time-steps for temporal integration.



Final algebraic equation

Two-dimensional problems

$$A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S = S_P$$

Three-dimensional problems

$$A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + A_T \phi_T + A_B \phi_B = S_P$$

Compact form

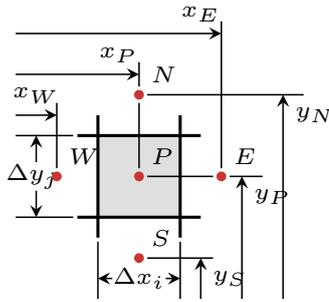
$$A_P \phi_P + \sum_{nb} A_{nb} \phi_{nb} = S_P$$

where *nb* (from *neighbor*) means that the summation includes all adjacent VCs.

The expressions to use for A_P , A_E , A_W etc. depend on the schemes adopted for the diffusive and convective terms, and on the relations used to evaluate the integrals.



Steady 2D diffusion



The conservation equation for 2D steady diffusion is

$$-\int_A \mathbf{J}_d'' \cdot \mathbf{n} \, dA = \int_V s \, dV$$

or

$$-\sum_k \int_{A_k} \mathbf{J}_d'' \cdot \mathbf{n} \, dA = \int_V s \, dV$$

and, since $\mathbf{J}_d'' = \Gamma \nabla \phi$, it can be written also as

$$\sum_k \int_{A_k} (-\Gamma \nabla \phi) \cdot \mathbf{n} \, dA = \int_V s \, dV$$

This can be approximated as

$$\sum_k (-\Gamma \nabla \phi)_k \cdot \mathbf{S}_k = (s_P^{rhs} + s_P^{lhs} \phi_P) \Delta x_i \Delta y_j$$

which can be expanded as

$$(-\Gamma \nabla \phi)_e \cdot \mathbf{S}_e + (-\Gamma \nabla \phi)_w \cdot \mathbf{S}_w + (-\Gamma \nabla \phi)_n \cdot \mathbf{S}_n + (-\Gamma \nabla \phi)_s \cdot \mathbf{S}_s = (s_P^{rhs} + s_P^{lhs} \phi_P) \Delta x_i \Delta y_j$$

And since

$$\mathbf{S}_e = \Delta y_j \mathbf{i}; \quad \mathbf{S}_w = -\Delta y_j \mathbf{i}; \quad \mathbf{S}_n = \Delta x_i \mathbf{j}; \quad \mathbf{S}_s = -\Delta x_i \mathbf{j}$$

we have, i.e., for the east face

$$\begin{aligned} (-\Gamma \nabla \phi)_e \cdot \mathbf{S}_e &= -\Gamma_e \Delta y_j \left(\frac{\partial \phi}{\partial x} \mathbf{i} + \frac{\partial \phi}{\partial y} \mathbf{j} \right)_e \mathbf{i} \\ &= -\Gamma_e \Delta y_j \left(\frac{\partial \phi}{\partial x} \right)_e \end{aligned}$$

Therefore, the previous balance equation can be written as

$$\begin{aligned} -\Gamma_e \Delta y_j \left(\frac{\partial \phi}{\partial x} \right)_e + \Gamma_w \Delta y_j \left(\frac{\partial \phi}{\partial x} \right)_w - \Gamma_n \Delta x_i \left(\frac{\partial \phi}{\partial y} \right)_n \\ + \Gamma_s \Delta x_i \left(\frac{\partial \phi}{\partial y} \right)_s = (s_P^{rhs} + s_P^{lhs} \phi_P) \Delta x_i \Delta y_j \end{aligned}$$

In particular, using the second order CDS

$$\begin{aligned} \left(\frac{\partial \phi}{\partial x} \right)_e &\approx \frac{\phi_E - \phi_P}{x_E - x_P}; & \left(\frac{\partial \phi}{\partial x} \right)_w &\approx \frac{\phi_P - \phi_W}{x_P - x_W} \\ \left(\frac{\partial \phi}{\partial y} \right)_n &\approx \frac{\phi_N - \phi_P}{y_N - y_P}; & \left(\frac{\partial \phi}{\partial y} \right)_s &\approx \frac{\phi_P - \phi_S}{y_P - y_S} \end{aligned}$$

Steady 2D diffusion - cont.

the balance equation can be approximated as

$$\begin{aligned} + \Gamma_e \frac{\Delta y_j}{x_E - x_P} (\phi_P - \phi_E) + \Gamma_w \frac{\Delta y_j}{x_P - x_W} (\phi_P - \phi_W) \\ + \Gamma_n \frac{\Delta x_i}{y_N - y_P} (\phi_P - \phi_N) + \Gamma_s \frac{\Delta x_i}{y_P - y_S} (\phi_P - \phi_S) = (s_P^{rhs} + s_P^{lhs} \phi_P) \Delta x_i \Delta y_j \end{aligned}$$

or

$$\begin{aligned} \left(\Gamma_e \frac{\Delta y_j}{x_E - x_P} + \Gamma_w \frac{\Delta y_j}{x_P - x_W} + \Gamma_n \frac{\Delta x_i}{y_N - y_P} + \Gamma_s \frac{\Delta x_i}{y_P - y_S} - s_P^{lhs} \Delta x_i \Delta y_j \right) \phi_P \\ - \frac{\Gamma_e \Delta y_j}{x_E - x_P} \phi_E - \frac{\Gamma_w \Delta y_j}{x_P - x_W} \phi_W - \frac{\Gamma_n \Delta x_i}{y_N - y_P} \phi_N - \frac{\Gamma_s \Delta x_i}{y_P - y_S} \phi_S = s_P^{rhs} \Delta x_i \Delta y_j \end{aligned}$$

In another form:

$$A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S = S_P$$

where

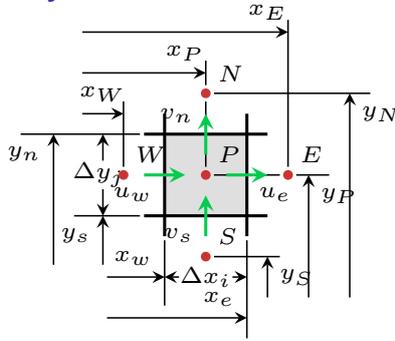
$$A_E = -\frac{\Gamma_e \Delta y_j}{x_E - x_P}; \quad A_W = -\frac{\Gamma_w \Delta y_j}{x_P - x_W}; \quad A_N = -\frac{\Gamma_n \Delta x_i}{y_N - y_P}; \quad A_S = -\frac{\Gamma_s \Delta x_i}{y_P - y_S}$$

$$A_P = -(A_E + A_W + A_N + A_S) - s_P^{lhs} \Delta x_i \Delta y_j$$

$$S_P = s_P^{rhs} \Delta x_i \Delta y_j$$



Steady 2D diffusion and convection



The conservation equation for 2D steady diffusion and convection is

$$\int_A \mathbf{J}_c'' \cdot \mathbf{n} dA - \int_A \mathbf{J}_d'' \cdot \mathbf{n} dA = \int_V s dV$$

or

$$\sum_k \int_{A_k} \mathbf{J}_c'' \cdot \mathbf{n} dA - \sum_k \int_{A_k} \mathbf{J}_d'' \cdot \mathbf{n} dA = \int_V s dV$$

and, since $\mathbf{J}_c'' = \rho \phi \mathbf{w}$ and $\mathbf{J}_d'' = \Gamma \nabla \phi$, it can be written also as

$$\sum_k \int_A \rho \phi \mathbf{w} \cdot \mathbf{n} dA - \sum_k \int_A \Gamma \nabla \phi \cdot \mathbf{n} dA = \int_V s dV$$

which can be approximated as

$$\sum_k (\rho \phi \mathbf{w}) \cdot \mathbf{S}_k + \sum_k (-\Gamma \nabla \phi) \cdot \mathbf{S}_k = \left(s_P^{rhs} + s_P^{lhs} \frac{\lambda_{n,PN}}{\phi_P} \right) \Delta x_i \Delta y_j$$

which can be expanded as

$$\begin{aligned} & (\rho \phi \mathbf{w})_e \cdot \mathbf{S}_e + (\rho \phi \mathbf{w})_w \cdot \mathbf{S}_w + (\rho \phi \mathbf{w})_n \cdot \mathbf{S}_n + (\rho \phi \mathbf{w})_s \cdot \mathbf{S}_s \\ & + (-\Gamma \nabla \phi)_e \cdot \mathbf{S}_e + (-\Gamma \nabla \phi)_w \cdot \mathbf{S}_w \\ & + (-\Gamma \nabla \phi)_n \cdot \mathbf{S}_n + (-\Gamma \nabla \phi)_s \cdot \mathbf{S}_s = \left(s_P^{rhs} + s_P^{lhs} \phi_P \right) \Delta x_i \Delta y_j \end{aligned}$$

Proceeding as for the 2D steady diffusion case for the diffusive fluxes, and using the second order CDS also for the convective fluxes, it results i.e. for the east face

$$\frac{\phi_E - \phi_P}{x_E - x_P} = \frac{\phi_e - \phi_P}{x_e - x_P}$$

It follows

$$\phi_e = \phi_P \left[1 - \frac{x_e - x_P}{x_E - x_P} \right] + \phi_E \left[\frac{x_e - x_P}{x_E - x_P} \right]$$

and similarly

$$\phi_w = \phi_W \lambda_{w,PW} + \phi_P (1 - \lambda_{w,PW})$$

$$\phi_n = \phi_N \lambda_{n,PN} + \phi_P (1 - \lambda_{n,PN})$$

$$\phi_s = \phi_S \lambda_{s,PS} + \phi_P (1 - \lambda_{s,PS})$$

$$\lambda_{e,PE} = \frac{x_e - x_P}{x_E - x_P}; \quad \lambda_{w,PW} = \frac{x_w - x_P}{x_W - x_P}$$

$$\lambda_{n,PN} = \frac{y_n - y_P}{y_N - y_P}; \quad \lambda_{s,PS} = \frac{y_s - y_P}{y_S - y_P}$$

Steady 2D diffusion and convection - cont.

And since

$$\mathbf{S}_e = \Delta y_j \mathbf{i}; \quad \mathbf{S}_w = -\Delta y_j \mathbf{i}; \quad \mathbf{S}_n = \Delta x_i \mathbf{j}; \quad \mathbf{S}_s = -\Delta x_i \mathbf{j}$$

we have, i.e., for the east face

$$\begin{aligned} (\rho \phi \mathbf{w})_e \cdot \mathbf{S}_e &= \rho \Delta y_j u_e \phi_e = \dot{m}_e \phi_e \\ &= \dot{m}_e \phi_E \lambda_{e,PE} + \dot{m}_e \phi_P (1 - \lambda_{e,PE}) \end{aligned}$$

and similarly for the other faces

$$(\rho \phi \mathbf{w})_w \cdot \mathbf{S}_w = \dot{m}_w \phi_W \lambda_{w,PW} + \dot{m}_w \phi_P (1 - \lambda_{w,PW})$$

$$(\rho \phi \mathbf{w})_n \cdot \mathbf{S}_n = \dot{m}_n \phi_N \lambda_{n,PN} + \dot{m}_n \phi_P (1 - \lambda_{n,PN})$$

$$(\rho \phi \mathbf{w})_s \cdot \mathbf{S}_s = \dot{m}_s \phi_S \lambda_{s,PS} + \dot{m}_s \phi_P (1 - \lambda_{s,PS})$$

with

$$\dot{m}_e = \rho u_e \Delta y_j; \quad \dot{m}_w = \rho u_w \Delta y_j$$

$$\dot{m}_n = \rho v_n \Delta x_i; \quad \dot{m}_s = \rho v_s \Delta x_i$$

Defining the coefficients for the convective fluxes

$$A_E^c = \dot{m}_e \lambda_{e,PE}; \quad A_W^c = -\dot{m}_w \lambda_{w,PW}$$

$$A_N^c = \dot{m}_n \lambda_{n,PN}; \quad A_S^c = -\dot{m}_s \lambda_{s,PS}$$

and, as already seen, the coefficients for the diffusive fluxes

$$A_E^d = -\frac{\Gamma_e \Delta y_j}{x_E - x_P}; \quad A_W^d = -\frac{\Gamma_w \Delta y_j}{x_P - x_W}$$

$$A_N^d = -\frac{\Gamma_n \Delta x_i}{y_N - y_P}; \quad A_S^d = -\frac{\Gamma_s \Delta x_i}{y_P - y_S}$$

we have

$$\begin{aligned} & [A_E^c + A_E^d] \phi_E + [A_W^c + A_W^d] \phi_W \\ & + [A_N^c + A_N^d] \phi_N + [A_S^c + A_S^d] \phi_S \\ & + \left\{ -[A_E^c + A_E^d] - [A_W^c + A_W^d] - [A_N^c + A_N^d] - [A_S^c + A_S^d] \right. \\ & \left. + \dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s \right\} \phi_P = \left(s_P^{rhs} + s_P^{lhs} \phi_P \right) \Delta x_i \Delta y_j \end{aligned}$$

Observing that

$$\sum_k \dot{m}_k = \dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s \equiv 0$$

and defining

$$A_E = A_E^c + A_E^d; \quad A_W = A_W^c + A_W^d$$

$$A_N = A_N^c + A_N^d; \quad A_S = A_S^c + A_S^d$$

$$A_P = -(A_E + A_W + A_N + A_S) - s_P^{lhs} \Delta x_i \Delta y_j$$

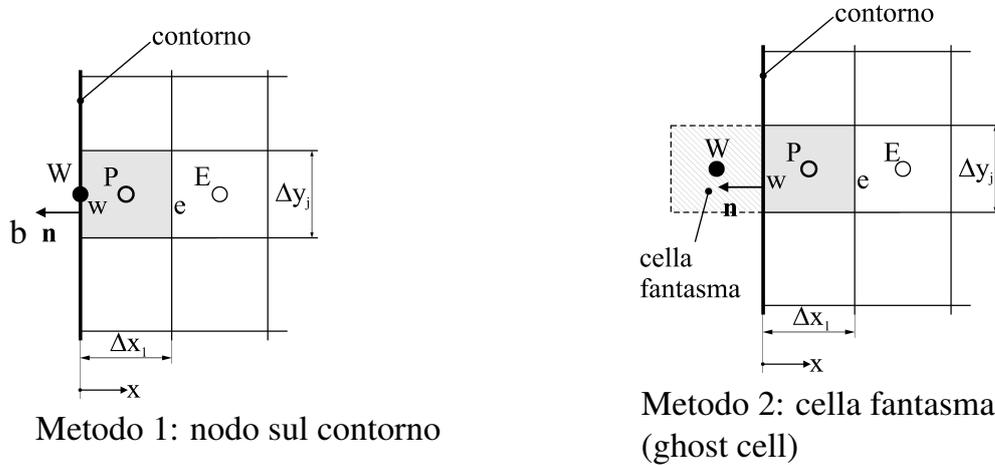
$$S_P = s_P^{rhs} \Delta x_i \Delta y_j$$

We are left with the final form of the equation

$$A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S = S_P$$

Condizioni al contorno

- Le condizioni al contorno vanno applicate prima di assemblare, e successivamente risolvere, il sistema di equazioni.
- Per le griglie Cartesiane, o più in generale *strutturate*, sono possibili due diversi approcci:



Condizione al contorno convettiva - metodo 1

$$\mathbf{n} \cdot \Gamma \nabla \phi + \alpha (\phi_{BC} - \phi_\infty) = 0$$

In altra forma:

$$-\Gamma \frac{\partial \phi}{\partial x} + \alpha \phi_{BC} = \alpha \phi_\infty$$

ed in termini discreti:

$$-\Gamma \frac{\phi_P - \phi_W}{\Delta x_1/2} + \alpha \phi_W = \alpha \phi_\infty$$

$$\phi_W = \phi_P \underbrace{\left[\frac{2\Gamma}{2\Gamma + \alpha \Delta x_1} \right]}_{C_{lhs}} + \underbrace{\left[\frac{\alpha \Delta x_1}{2\Gamma + \alpha \Delta x_1} \phi_\infty \right]}_{C_{rhs}}$$

Equazione algebrica per la cella P :

$$\underbrace{(A_P + A_W C_{lhs})}_{A_P} \phi_P + A_E \phi_E + A_N \phi_N + A_S \phi_S = \underbrace{(S_P - A_W C_{rhs})}_{S_P}$$



Condizione al contorno di Dirichlet - metodo 2

È la condizione al contorno più semplice, e corrisponde, nel caso dell'equazione dell'energia, al valore imposto della temperatura. Detto ϕ_{BC} il valore assegnato, si ha:

$$\phi_{BC} = \frac{\phi_P + \phi_W}{2}$$

da cui:

$$\phi_W = 2\phi_{BC} - \phi_P$$

Equazione algebrica per la cella P :

$$\underbrace{(A_P - A_W)}_{A_P} \phi_P + A_E \phi_E + A_N \phi_N + A_S \phi_S = \underbrace{(S_P - 2 A_W \phi_{BC})}_{S_P}$$



Under-relaxation of the iterative process

We have already seen that the general form of the discretization equation is

$$\begin{aligned} A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S &= S_P \\ A_P \phi_P + A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + A_T \phi_T + A_B \phi_B &= S_P \end{aligned}$$

for 2D and 3D problems, respectively. In a general compact form

$$A_P \phi_P + \sum_{nb} A_{nb} \phi_{nb} = S_P$$

For a general convection-diffusion problem, both Γ and s may be function of ϕ , which may lead to large variations in the coefficients and in the source term.

This, in turn, may cause divergence, or slow oscillatory convergence, of the iterative solution procedure.

To favor convergence, it may be useful to *slow down* the variations of ϕ between iterations by *under-relaxation*. This can be done in several ways, one of the most common is illustrated next.



Under-relaxation of the iterative process - cont.

The previous equation can be written as

$$\phi_P = \frac{-\sum_{nb} A_{nb}\phi_{nb} + S_P}{A_P}$$

If k is the iteration index, ϕ_P^{k-1} represents the variable at the previous iteration, which can be added and subtracted from the right hand side of the previous equation

$$\phi_P^k = \phi_P^{k-1} + \left(\frac{-\sum_{nb} A_{nb}\phi_{nb}^k + S_P}{A_P} - \phi_P^{k-1} \right)$$

where the term between parentheses is the change in ϕ_P obtained from the current iteration. This change can be modified by the introduction of a *relaxation factor* α_ϕ such that

$$\phi_P^k = \phi_P^{k-1} + \alpha_\phi \left(\frac{-\sum_{nb} A_{nb}\phi_{nb}^k + S_P}{A_P} - \phi_P^{k-1} \right)$$



Under-relaxation of the iterative process - cont.

The previous equation can be rewritten as

$$\boxed{\frac{A_P}{\alpha_\phi} \phi_P^k + \sum_{nb} A_{nb}\phi_{nb}^k = S_P + \frac{(1 - \alpha_\phi) A_P}{\alpha_\phi} \phi_P^{k-1}}$$

- At convergence $\phi_P^k \equiv \phi_P^{k-1} = \phi_P$ and its value is independent of α_ϕ and also satisfies the original equation.
- Depending on the value of α_ϕ , the equation can be
 - 1 Under-relaxed, i.e. $0 < \alpha_\phi < 1$
 - 2 Over-relaxed, i.e. $\alpha_\phi > 1$
- In CFD and CHT applications, under-relaxation is usually used.
- The *optimum* value of α_ϕ is problem dependent and there are not general rules. It depends, among other, by the type of problem, grid size and quality, iterative solver et.
- The value of α_ϕ can vary within the domain, and can change between iterations.



Parte III

Integrazione temporale

10 INTEGRAZIONE TEMPORALE

- Metodi base
- Metodi multilivello e metodi predictor-corrector
- Metodi particolari

11 APPLICATION TO THE GENERIC TRANSPORT EQUATION



Integrazione temporale

Equazione di trasporto nella formulazione generale tempo-variante:

$$\frac{\partial}{\partial \vartheta} (\rho \phi) + \nabla \cdot (\rho \mathbf{w} \phi) = \nabla \cdot (\Gamma \nabla \phi) + s$$

- Integrazione nel tempo alle *differenze finite*: l'intervallo temporale ϑ è suddiviso in un certo numero di intervalli temporali $\Delta \vartheta$.
- Integrazione temporale *marciando* nel tempo, nota la condizione *iniziale* al tempo $\vartheta = \vartheta_0$, e le condizioni al contorno in ciascun istante successivo $\vartheta = l \Delta \vartheta$, con $l = 1, \dots, n, n + 1, \dots$

L'equazione può essere scritta nella forma compatta:

$$\frac{\partial}{\partial \vartheta} (\rho \phi) = \mathcal{F} (\vartheta, \phi (\vartheta))$$

$$\mathcal{F} = -\nabla \cdot (\rho \mathbf{w} \phi) + \nabla \cdot (\Gamma \nabla \phi) + s$$

Questa ricorda la forma delle equazioni differenziali ordinarie ai valori iniziali.



ODE - Ordinary Differential Equations

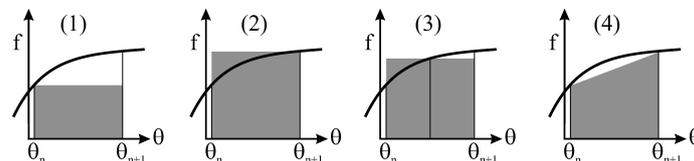
Equazione differenziale ordinaria (ODE) del primo ordine, assieme alla condizione iniziale al tempo ϑ_0 :

$$\frac{d\phi(\vartheta)}{d\vartheta} = f(\vartheta, \phi(\vartheta)) \quad \text{con} \quad \phi(\vartheta_0) = \phi^0$$

Si desidera determinare ϕ dopo un intervallo $\Delta\vartheta$ dall'istante iniziale, ottenendo così ϕ^1 all'istante $\vartheta_1 = \vartheta_0 + \Delta\vartheta$; successivamente si determinerà ϕ^2 all'istante $\vartheta_2 = \vartheta_1 + \Delta\vartheta$ e così via. I metodi più semplici sono ottenuti integrando la precedente fra ϑ_n e $\vartheta_{n+1} = \vartheta_n + \Delta\vartheta$:

$$\int_{\vartheta_n}^{\vartheta_{n+1}} \frac{d\phi(\vartheta)}{d\vartheta} d\vartheta = \phi^{n+1} - \phi^n = \int_{\vartheta_n}^{\vartheta_{n+1}} f(\vartheta, \phi(\vartheta)) d\vartheta$$

dove $\phi^{n+1} = \phi(\vartheta_{n+1})$. L'espressione scritta è esatta, tuttavia il termine a destra non può venire valutato senza conoscere la soluzione al nuovo istante, ed è quindi necessario approssimarla. Vediamo in breve i metodi più comuni.



(1) - Metodo di Eulero esplicito

$$\phi^{n+1} = \phi^n + f(\vartheta_n, \phi^n) \Delta\vartheta$$

(2) - Metodo di Eulero implicito

$$\phi^{n+1} = \phi^n + f(\vartheta_{n+1}, \phi^{n+1}) \Delta\vartheta$$

(3) - Metodo del punto medio

$$\phi^{n+1} = \phi^n + f(\vartheta_{n+1/2}, \phi^{n+1/2}) \Delta\vartheta$$

(4) - Metodo di Crank-Nicolson

$$\phi^{n+1} = \phi^n + \frac{1}{2} [f(\vartheta_n, \phi^n) + f(\vartheta_{n+1}, \phi^{n+1})] \Delta\vartheta$$



Stabilità ed accuratezza dei metodi base

- I metodi visti, a parte (3), sono detti *a due livelli*.
- A parte il metodo (1), tutti gli altri metodi richiedono la funzione incognita ad istanti diversi da ϑ_n , per il quale è nota, e quindi è necessario procedere ad approssimazioni e/o iterazioni.
- Il metodo (1) è detto *esplicito*, mentre gli altri sono metodi *impliciti*.
- Il metodo (1), come tutti i metodi espliciti, è più economico e semplice, ma non può essere utilizzato per valori di $\Delta\vartheta$ superiori al limite di stabilità:

$$\left| \Delta\vartheta \frac{\partial f(\vartheta, \phi)}{\partial \phi} \right| < 2 \quad (1)$$

- I metodi impliciti sono più costosi, ma sono *incondizionatamente stabili*:
 - ▶ metodi impliciti – scala temporale maggiore del limite di stabilità metodi espliciti;
 - ▶ metodi espliciti – scale temporali dello stesso ordine del limite di stabilità.
- I metodi (1) e (2) hanno un'accuratezza temporale del prim'ordine ($\Delta\vartheta$), mentre i metodi (3) e (4) hanno un'accuratezza del second'ordine, $((\Delta\vartheta)^2)$.
- Con metodi a due livelli, l'accuratezza può essere, al più, del second'ordine.



Metodi multilivello e metodi predictor-corrector

Metodi di ordine più elevato:

- Metodi *multilivello* – scelti fra gli istanti di tempo già considerati, e per i quali la soluzione è disponibile: metodi espliciti di *Adams-Bashfort*:

$$\phi^{n+1} = \phi^n + \frac{1}{2} [3 f(\vartheta_n, \phi^n) - f(\vartheta_{n-1}, \phi^{n-1})] \Delta\vartheta$$

$$\phi^{n+1} = \phi^n + \frac{1}{12} [23 f(\vartheta_n, \phi^n) - 16 f(\vartheta_{n-1}, \phi^{n-1}) + 5 f(\vartheta_{n-2}, \phi^{n-2})] \Delta\vartheta$$

- Metodi di *Runge-Kutta* – compresi fra ϑ_n e ϑ_{n+1} : metodo del quarto ordine:

$$\phi_{n+1/2}^* = \phi^n + \frac{\Delta\vartheta}{2} f(\vartheta_n, \phi^n)$$

$$\phi_{n+1/2}^{**} = \phi^n + \frac{\Delta\vartheta}{2} f(\vartheta_{n+1/2}, \phi_{n+1/2}^*)$$

$$\phi_{n+1}^* = \phi^n + \Delta\vartheta f(\vartheta_{n+1/2}, \phi_{n+1/2}^{**})$$

$$\phi^{n+1} = \phi^n + \frac{\Delta\vartheta}{6} [f(\vartheta_n, \phi^n) + 2 f(\vartheta_{n+1/2}, \phi_{n+1/2}^*) + 2 f(\vartheta_{n+1/2}, \phi_{n+1/2}^{**}) + f(\vartheta_{n+1}, \phi_{n+1}^*)]$$



Gli schemi visti sono fra quelli più utilizzati nei codici CFD general-purpose.

Schemi indirizzati ad applicazioni particolari o caratterizzati da particolari vantaggi:

- Per simulazioni non-stazionarie caratterizzate da elevata dinamicità, quali DES, LES e DNS, che necessitano di passi d'integrazione temporale ridotti, può risultare conveniente integrare il termine convettivo - non lineare - per via *esplicita*, trattando per via *implicita* il solo termine viscoso - lineare - eliminando quindi la necessità di sub-iterazioni ad ogni time-step;
- In LES e DNS, si utilizzano spesso schemi di discretizzazione temporale di ordine elevato, in particolare quando gli schemi di discretizzazione spaziale sono di ordine elevato: ciò è più comune per domini semplici, per i quali schemi spaziali di ordine elevato sono di più agevole applicazione.



Schema di Eulero implicito del secondo ordine - 1

Gli integrali che appaiono nella

$$\int_{\vartheta_n}^{\vartheta_{n+1}} \frac{d\phi(\vartheta)}{d\vartheta} d\vartheta = \phi^{n+1} - \phi^n = \int_{\vartheta_n}^{\vartheta_{n+1}} f(\vartheta, \phi(\vartheta)) d\vartheta$$

qua riportata nuovamente per comodità, possono venir valutati conoscendo i *valori medi* nell'intervallo di integrazione.

In particolare, lo schema di Eulero implicito del secondo ordine, o schema Gear, può essere costruito integrando su un intervallo di tempo $\Delta\vartheta$ centrato attorno a ϑ_{n+1} (cioè fra $\vartheta_{n+1} - \Delta\vartheta/2$ e $\vartheta_{n+1} + \Delta\vartheta/2$), ed applicando la regola del punto medio per ambedue i termini dell'equazione.

La derivata temporale a ϑ_{n+1} può essere approssimata differenziando una parabola passante per la soluzione nei tre istanti ϑ_{n-1} , ϑ_n e ϑ_{n+1} , ottenendo:

$$\left(\frac{d\phi}{d\vartheta}\right)^{n+1} \approx \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta\vartheta}$$



Schema di Eulero implicito del secondo ordine - 2

Lo schema risultante è il seguente:

$$\phi^{n+1} = \frac{4}{3}\phi^n - \frac{1}{3}\phi^{n-1} + \frac{2}{3}f(\vartheta_{n+1}, \phi^{n+1})\Delta\vartheta$$

Le sue caratteristiche principali sono:

- Implicito - f è valutato al *nuovo* istante;
- Accuratezza del secondo ordine;
- Più semplice ed economico da implementare rispetto allo schema di Crank-Nicolson, ma l'errore di troncamento è quattro volte maggiore di quest'ultimo;
- Il maggiore errore di troncamento è dovuto alla mancata considerazione della variazione temporale dei flussi, della quale viceversa tiene conto lo schema di Crank-Nicolson;
- Diffusione numerica indotta trascurabile per piccoli passi di tempo, come avviene, ad esempio, in LES.



Schema di Eulero implicito del secondo ordine - 3

Derivazione basata su sviluppo in serie di Taylor

Sviluppando ϕ^{n-1} e ϕ^n in serie di Taylor nell'intorno di ϕ^{n+1} si ha:

$$\phi^{n-1} = \phi^{n+1} - 2\left(\frac{d\phi}{d\vartheta}\right)^{n+1}\Delta\vartheta + 2\left(\frac{d^2\phi}{d\vartheta^2}\right)^{n+1}\Delta\vartheta^2 + O(\Delta\vartheta^3)$$

$$\phi^n = \phi^{n+1} - \left(\frac{d\phi}{d\vartheta}\right)^{n+1}\Delta\vartheta + \frac{1}{2}\left(\frac{d^2\phi}{d\vartheta^2}\right)^{n+1}\Delta\vartheta^2 + O(\Delta\vartheta^3)$$

Moltiplicando la seconda per 4 e sottraendola dalla prima:

$$\begin{aligned}\phi^{n-1} - 4\phi^n &= \phi^{n+1} - 4\phi^{n+1} - 2\left(\frac{d\phi}{d\vartheta}\right)^{n+1}\Delta\vartheta + 4\left(\frac{d\phi}{d\vartheta}\right)^{n+1}\Delta\vartheta \\ &\quad + 2\left(\frac{d^2\phi}{d\vartheta^2}\right)^{n+1}\Delta\vartheta^2 - 2\left(\frac{d^2\phi}{d\vartheta^2}\right)^{n+1}\Delta\vartheta^2 + O(\Delta\vartheta^3)\end{aligned}$$

da cui:

$$\left(\frac{d\phi}{d\vartheta}\right)^{n+1} \approx \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta\vartheta}$$



Metodi misti esplicito-implicito, rispettivamente per il flusso convettivo e per il flusso diffusivo, usati in simulazioni LES e DNS su geometrie semplici.

- Schema di Adams-Bashfort del second'ordine per il termine convettivo, e schema Gear per il contributo diffusivo:

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta\vartheta} = -\frac{1}{2} (3\nabla \cdot (\rho\mathbf{w}\phi^n) - \nabla \cdot (\rho\mathbf{w}\phi^{n-1})) + (\nabla \cdot (\Gamma\nabla\phi))^{n+1} + s^{n+1}$$

- Schema Runge-Kutta per i termini convettivi, e Crank-Nicolson o Gear per i contributi diffusivi.



Applicazione alla generica equazione di trasporto

Caso bidimensionale, metodo implicito di Eulero del prim'ordine e schema CDS :

$$A_P\phi_P^{n+1} + A_E\phi_E^{n+1} + A_W\phi_W^{n+1} + A_N\phi_N^{n+1} + A_S\phi_S^{n+1} = S_P$$

$$\dot{m}_e = \rho u_e \Delta y_j$$

$$A_E^c = \dot{m}_e \lambda_{e,PE}$$

$$A_E^d = -\frac{\Gamma_e \Delta y_j}{x_E - x_P}$$

$$A_E = A_E^c + A_E^d$$

$$A_P = \frac{\rho}{\Delta\vartheta} \Delta x_i \Delta y_j - (A_E + A_W + A_N + A_S) - s_P^{lhs} \Delta x_i \Delta y_j$$

$$S_P = \left(s_P^{rhs} + \frac{\rho \phi^n}{\Delta\vartheta} \right) \Delta x_i \Delta y_j$$

- Il termine non-stazionario aumenta il peso del coefficiente A_P .
- Utilizzo della forma non-stazionaria delle equazioni per problemi stazionari.
- Gli schemi espliciti non richiedono la soluzione di sistemi di equazioni, ma il passo di integrazione temporale è limitato.



Sistemi di equazioni

12 SISTEMI DI EQUAZIONI LINEARI

13 DIRECT METHODS

- Eliminazione di Gauss
- Decomposizione LU
- Metodi particolari

14 ITERATIVE METHODS

- Metodi di base
- Metodi basati sul gradiente
- Metodi multigriglia



Soluzione dei sistemi di equazioni - 1

Sistema di equazioni:

$$\mathbf{A}\Phi = \mathbf{S}$$

dove \mathbf{A} è la matrice dei coefficienti, Φ è il vettore delle incognite, e \mathbf{S} il vettore dei termini noti:

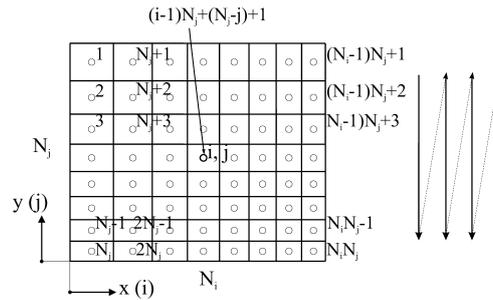
$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{bmatrix} \quad \Phi = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_N \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{pmatrix}$$

- La matrice \mathbf{A} è *sparsa*.
- La disposizione degli elementi non nulli, all'interno di \mathbf{A} , dipende dalla modalità di numerazione delle incognite.
- L'approccio più comune, per griglie strutturate, è quello di utilizzare la numerazione lessicografica:



Soluzione dei sistemi di equazioni - 2

Numerazione lessicografica:



Per un problema 2D con $N_j = 3$ ed $N_i = 4$:

$$\mathbf{A} = \left[\begin{array}{ccc|cc|}
 \begin{array}{c} A_{P1} \\ A_{N2} \end{array} & \begin{array}{c} A_{S1} \\ A_{P2} \\ A_{N3} \end{array} & & A_{E1} & & \\
 & & & & A_{E2} & \\
 & & & & & A_{E3} \\
 \hline
 A_{W4} & & & A_{P4} & A_{S4} & & A_{E4} & & \\
 & A_{W5} & & A_{N5} & A_{P5} & A_{S5} & & A_{E5} & \\
 & & A_{W6} & & A_{N6} & A_{P6} & & & A_{E6} \\
 \hline
 & & & A_{W7} & & & A_{P7} & A_{S7} & & A_{E7} \\
 & & & & A_{W8} & & A_{N8} & A_{P8} & A_{S8} & & A_{E8} \\
 & & & & & A_{W9} & & A_{N9} & A_{P9} & & A_{E9} \\
 \hline
 & & & & & & A_{W10} & & & A_{P10} & A_{S10} \\
 & & & & & & & A_{W11} & & A_{N11} & A_{P11} & A_{S11} \\
 & & & & & & & & A_{W12} & & A_{N12} & A_{P12}
 \end{array} \right]$$



Soluzione dei sistemi di equazioni - 3

La matrice \mathbf{A} possiede una struttura *polidiagonale*, detta anche *diagonale a blocchi* o *tridiagonale a blocchi*.

- Non è necessario memorizzare l'intera matrice.
- Per tali matrici esistono efficienti algoritmi di soluzione di tipo *iterativo* e, per casi particolari, anche *diretto*.
- La struttura della matrice non cambia – alcuni termini sono trattati in modalità *correzione differita* - lavorando con griglie curvilinee.
- La struttura della matrice si conserva, all'interno di ciascun blocco, anche per griglie strutturate multiblocco.
- La struttura della matrice, sempre sparsa, cambia utilizzando griglie non strutturate.
- Distinzione fra metodi *diretti* e metodi *iterativi*:
 - ▶ Nei primi la soluzione è ottenuta in un numero predeterminato di operazioni, funzione del numero di incognite.
 - ▶ Nei metodi iterativi la soluzione è ottenuta attraverso iterazioni successive partendo da un valore *di tentativo*, ed il numero di operazioni non è predeterminabile, ma dipende da diversi fattori.



Eliminazione di Gauss - 1

Fra i metodi diretti il più semplice è il metodo di Gauss. N è il numero totale delle incognite, pari a $N_i \times N_j$ per problemi 2D e $N_i \times N_j \times N_k$ in 3D.

- Eliminazione di A_{21} dalla matrice \mathbf{A} :
 - ▶ Si moltiplica la prima equazione (prima riga della matrice) per A_{21}/A_{11} e la si sottrae dalla seconda equazione.
 - ▶ Anche tutti gli altri elementi della seconda riga risulteranno modificati, così come il secondo elemento S_2 del termine noto.
- Poi si moltiplica la prima equazione per A_{31}/A_{11} e la si sottrae dalla terza.
- Si procede così per tutti gli altri elementi della prima colonna della matrice.

$$A'_{ij} = A_{ij} - \frac{A_{i1}}{A_{11}} A_{1j} \quad i = 2, \dots, N, j = 1, \dots, N$$
$$S'_i = S_i - \frac{A_{i1}}{A_{11}} S_1$$



Eliminazione di Gauss - 2

- Dopo questo primo passaggio la matrice risulta così modificata:

$$\mathbf{A}' = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ 0 & A'_{22} & \dots & A'_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & A'_{N2} & \dots & A'_{NN} \end{bmatrix}$$

- Nessuna delle equazioni $2, 3, \dots, N$ contiene più la la variabile ϕ_1 .
- Si passa quindi ad eliminare la variabile ϕ_2 da questo sistema ridotto, nello stesso modo, e si continua quindi per le colonne $3, 4, \dots, N - 1$.
- In termini generali:

$$A'_{ij} = A'_{ij} - \frac{A'_{ik}}{A'_{kk}} A'_{kj} \quad k = 1, \dots, N - 1, i = k + 1, \dots, N, j = k, \dots, N$$
$$S'_i = S'_i - \frac{A'_{ik}}{A'_{kk}} S'_k$$



Eliminazione di Gauss - 3

- I fattori A'_{ik}/A'_{kk} sono detti *moltiplicatori*, o coefficienti moltiplicativi, e la matrice risultante \mathbf{U} è in forma triangolare superiore:

$$\mathbf{U} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ 0 & A'_{22} & \dots & A'_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A'_{NN} \end{bmatrix} \quad \mathbf{S}' = \begin{Bmatrix} S_1 \\ S'_2 \\ \vdots \\ S'_N \end{Bmatrix}$$

- Il sistema così modificato è facilmente risolto. Osservando che:

$$\phi_N = \frac{S'_N}{A'_{NN}}$$

è sufficiente procedere all'indietro:

$$\phi_i = \frac{S'_i - \sum_{j=i+1}^N A'_{ij} \phi_j}{A'_{ii}} \quad i = N - 1, \dots, 1$$



Eliminazione di Gauss - 4

- La sequenza di operazioni che, partendo dalla matrice triangolare superiore, fornisce i valori delle incognite, è detta *sostituzione all'indietro* (backsubstitution).
- Per N sufficientemente grande il numero totale di operazioni richieste dal metodo di Gauss è proporzionale a $N^3/3$, sebbene la fase di sostituzione all'indietro richiede $N^2/2$ operazioni, ed è quindi molto meno costosa.
- Il costo elevato, in termini di numero di operazioni per N grande, dell'algoritmo di Gauss, giustifica la ricerca di metodi più economici per la soluzione dei sistemi di equazioni.
- Il metodo di Gauss trova scarso utilizzo, in questa forma, nei problemi di termofluidodinamica computazionale.



Decomposizione LU

L'idea è di *fattorizzare* la matrice \mathbf{A} :

$$\mathbf{A} = \mathbf{L} \mathbf{U}$$

Per rendere la fattorizzazione unica, si impone che gli elementi diagonali della matrice \mathbf{L} , L_{ii} , siano unitari (o, in modo equivalente, siano unitari gli elementi diagonali U_{ii} della matrice \mathbf{U}). La matrice \mathbf{U} è quella ottenuta nella fase di eliminazione in avanti del metodo di Gauss, mentre gli elementi della matrice \mathbf{L} sono i coefficienti moltiplicativi, A_{ji}/A_{ii} , utilizzati in tale fase.

$$\mathbf{U} \Phi = \mathbf{Y}$$

$$\mathbf{L} \mathbf{Y} = \mathbf{S}$$

La soluzione, in sequenza, di tali due sistemi consente di risolvere il problema. L'importanza del metodo di decomposizione LU deriva da:

- 1 la fattorizzazione non richiede la conoscenza del vettore \mathbf{S} dei termini noti;
- 2 la decomposizione LU costituisce la base di alcuni dei migliori metodi iterativi.



Metodi particolari

Casi speciali di sistemi lineari:

- limitati a geometrie semplici;
- efficienti per soluzioni ripetute del medesimo sistema con differenti termini noti.

Algoritmi di maggiore interesse nella termofluidodinamica computazionale:

- Algoritmi per sistemi tridiagonali e tridiagonali ciclici: Thomas, Temperton, *etc.*. Il numero di operazioni scala linearmente con N , anziché con N^3 , come nell'eliminazione di Gauss.
- Algoritmi per griglie Cartesiane (ed anche cilindriche e sferiche) 2D e 3D *uniformi*: basati su metodi FFT (Fast Fourier Transform) e di riduzione ciclica. Il numero di operazioni, escludendo la fase di fattorizzazione, è dell'ordine di $N \times (\ln_2 N)$.
- Algoritmi per griglie Cartesiane (cilindriche e sferiche) 2D e 3D *non uniformi*: decomposizione matriciale, per il quale il numero di operazioni, per la sola fase di soluzione, varia con $N^{4/3}$.



Metodi iterativi - 1

L'utilizzo di metodi iterativi è giustificato dalle seguenti osservazioni:

- Con il metodo LU, sebbene \mathbf{A} sia sparsa, le matrici \mathbf{L} ed \mathbf{U} sono *piene*.
- Gli errori di discretizzazione sono usualmente ben superiori agli errori di arrotondamento accumulati con i metodi diretti.
- Nei problemi applicativi è necessario risolvere sistemi di rilevante dimensione.

L'idea alla base dei metodi iterativi consiste nel partire da una soluzione di tentativo, e migliorarla sino al livello desiderato: dopo n iterazioni avremo una soluzione approssimata Φ^n , che non soddisfa l'equazione in modo esatto:

$$\mathbf{A}\Phi^n = \mathbf{S} - \mathbf{r}^n$$

Sottraendo questa dal sistema di equazioni di partenza si ottiene una relazione fra l'errore di convergenza ϵ^n , definito dalla:

$$\epsilon^n = \Phi - \Phi^n$$

ed il residuo \mathbf{r}^n :

$$\mathbf{A}\epsilon^n = \mathbf{r}^n$$

L'obiettivo di un metodo iterativo è quello di diminuire, sino a zero, il residuo, ed in tal caso anche l'errore di convergenza va a zero.



Metodi iterativi - 2

Schema iterativo:

$$\mathbf{M}\Phi^{n+1} = \mathbf{N}\Phi^n + \mathbf{B}$$

La soluzione, a convergenza avvenuta, deve soddisfare l'equazione di partenza. A convergenza, inoltre, $\Phi^{n+1} = \Phi$, da cui:

$$\mathbf{A} = \mathbf{M} - \mathbf{N} \quad \text{e} \quad \mathbf{B} = \mathbf{S}$$

o, più in generale:

$$\mathbf{P}\mathbf{A} = \mathbf{M} - \mathbf{N} \quad \text{e} \quad \mathbf{B} = \mathbf{P}\mathbf{S}$$

dove \mathbf{P} è una matrice (non singolare) di *precondizionamento*.

Versione alternativa dello schema iterativo:

$$\mathbf{M}(\Phi^{n+1} - \Phi^n) = \mathbf{B} - (\mathbf{M} - \mathbf{N})\Phi^n \quad \text{o} \quad \mathbf{M}\delta^n = \mathbf{r}^n$$

dove $\delta^n = \Phi^{n+1} - \Phi^n$ è la *correzione*, o aggiornamento, e rappresenta un'approssimazione dell'errore di convergenza.



Metodi iterativi - 3

- Un metodo iterativo è tanto più efficace quanto più economica è la soluzione del sistema di equazioni modificato, e quanto più rapida è la convergenza, cioè minore il numero di iterazioni necessarie.
- La velocità di convergenza è funzione del *raggio spettrale* λ_1 , definito come il modulo dell'autovalore massimo della matrice $\mathbf{M}^{-1}\mathbf{N}$.
- Un'espressione approssimata del numero di iterazioni necessario è dato dalla:

$$n \approx \frac{\ln \left(\frac{\delta}{a_1} \right)}{\ln \lambda_1}$$

dove a_1 è una costante e δ è la tolleranza richiesta per l'errore di convergenza. Da questa relazione approssimata per n si può notare che, per λ_1 prossimo all'unità, la convergenza può risultare molto lenta.



Metodi di base - 1

Metodo di Jacobi

\mathbf{M} è una matrice diagonale, i cui elementi sono quelli della diagonale principale di \mathbf{A} :

$$\phi_P^{n+1} = \frac{S_P - A_E \phi_E^n - A_W \phi_W^n - A_N \phi_N^n - A_S \phi_S^n}{A_P}$$

Il numero di iterazioni richiesto è proporzionale al quadrato del numero di VC in una direzione. È quindi un metodo piuttosto lento, e perciò di scarsa utilità.



Metodi di base - 2

Metodo di sovrarilassamento o SOR (Successive Over-Relaxation)

La matrice \mathbf{M} è costituita dalla porzione triangolare inferiore della matrice \mathbf{A} .

Seguendo l'ordine lessicografico:

$$\phi_P^{n+1} = (1 - \omega) \phi_P^n + \omega \frac{S_P - A_E \phi_E^n - A_W \phi_W^{n+1} - A_N \phi_N^{n+1} - A_S \phi_S^n}{A_P}$$

dove ω è il fattore di sovrarilassamento, ($1.5 \leq \omega \leq 1.85$), necessario per accelerare la convergenza.

Il metodo SOR è più efficiente del metodo di Jacobi poichè con il SOR si utilizzano i *valori nuovi* delle variabili non appena questi si rendono disponibili.

Metodo di Gauss-Seidel

Corrisponde allo schema SOR con $\omega = 1$:

$$\phi_P^{n+1} = \frac{S_P - A_E \phi_E^n - A_W \phi_W^{n+1} - A_N \phi_N^{n+1} - A_S \phi_S^n}{A_P}$$



Metodi di base - 3

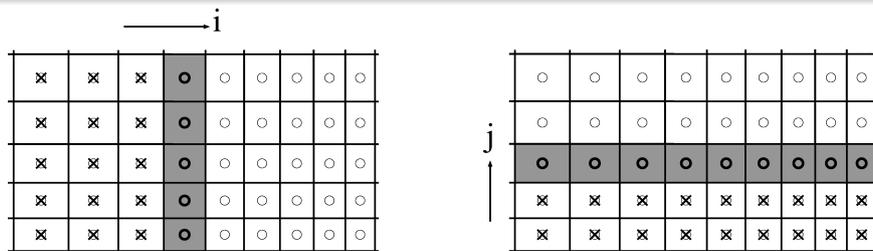
Metodo di sovrarilassamento per linee o SLOR (Successive Line Over-Relaxation)

Aggiornamento delle variabili di un'intera linea – riga o colonna – di VC.

\mathbf{M} è costituita da un sottoinsieme della \mathbf{A} , relativo ai soli coefficienti delle variabili sulla linea in esame.

Procedendo per colonne, e per valori di i crescenti:

$$A_P \phi_{i,j}^{n+1} + A_N \phi_{i,j+1}^{n+1} + A_S \phi_{i,j-1}^{n+1} = [S_P - A_E \phi_{i+1,j}^n - A_W \phi_{i-1,j}^n]$$



- Si ottiene un sistema tridiagonale.
- Possibile utilizzo di un coefficiente di sovrarilassamento ω .



Cenni sui metodi basati sul gradiente

- Trovano fondamento nel fatto che è possibile minimizzare una funzione, rispetto a più direzioni, lavorando su una direzione alla volta.
- Possono venire classificati in funzione del tipo di sistemi (matrici) lineari a cui si rivolgono:
 - ① *Sistemi simmetrici*. CG (Conjugate Gradient) e ICCG (Incomplete Cholesky Conjugate Gradient).
 - ② *Sistemi non simmetrici*. BCG (BiConjugate Gradient), CGS (Conjugate Gradient Squared), CGSTAB (CGS Stabilized) e GMRES (Generalized Minimal Residual).
- I metodi basati sul gradiente fanno uso di opportuni *precondizionatori* al fine migliorarne le prestazioni.
- Si ricorda infatti che la velocità di convergenza di tali metodi diminuisce all'aumentare dell'*indice di condizionamento* κ della matrice \mathbf{A} , definito dalla:

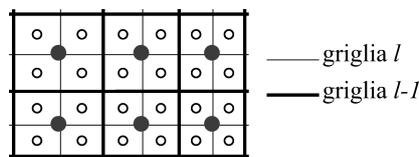
$$\kappa = \frac{\lambda_{max}}{\lambda_{min}}$$

- L'idea alla base delle tecniche di preconditionamento è quello di sostituire il problema in esame con un altro caratterizzato da un indice di condizionamento inferiore.



Cenni sui metodi multigriglia - 1

Basati sull'utilizzo di più griglie: la *griglia effettiva* di calcolo (livello l), ed una sequenza di griglie via via più rade ($l - 1, l - 2, \dots, 1$), dove ciascuna cella è ottenuta *sommando* (agglomerando) più celle della griglia immediatamente più fine.



L'idea alla base di tali metodi, limitando la descrizione al caso di due sole griglie, nasce dalle seguenti osservazioni:

- Tutti i metodi iterativi sono efficaci nella riduzione degli errori (residui) di *piccola lunghezza d'onda*, cioè degli errori di elevata frequenza. Si tratta di errori la cui lunghezza d'onda è comparabile con la dimensione della griglia.
- La riduzione degli errori di piccola lunghezza d'onda avviene nelle prime iterazioni.
- Viceversa, la riduzione degli errori di lunghezza d'onda via via più elevata (bassa frequenza) avviene più lentamente, ed in effetti, dopo le prime iterazioni, la velocità di convergenza si riduce.



Cenni sui metodi multigriglia - 2

- Riduzione degli errori di bassa frequenza, in modo molto più economico, su una griglia più rada (*interpolazione* o somma dei residui).
- Ottenuta la soluzione sulla griglia rada, questa viene *iniettata* o sommata sulla griglia fine.
- Le iterazioni sulla griglia più rada sono molto più economiche.
- Maggiore velocità di convergenza, poichè sulla griglia rada lo stesso errore ha frequenza più elevata.
- Il vantaggio aumenta utilizzando più griglie, o livelli.
- Idealmente, il numero di iterazioni *equivalenti*, e quindi il tempo di calcolo, varia linearmente con il numero N di variabili.

Metodi multigriglia di maggiore interesse in CFD:

- 1 Multigriglia di tipo *Additivo* - ACM (Additive Correction Multigrid).
- 2 Multigriglia di tipo *Algebrico* - AGM (Algebraic Multigrid).

Come risolutore iterativo nelle varie griglie, detto *smoother*, può essere utilizzato uno qualsiasi dei metodi iterativi visti.



OUTLINE

Parte V

Problemi termofluidodinamici - approccio FVM

- 15 **PROBLEMI TERMOFLUIDODINAMICI**
 - Disposizione delle variabili sulla griglia
 - Interpolazione di Rhie-Chow
- 16 **PROCEDURA AI VOLUMI FINITI**
 - Metodi segregati
 - Metodi accoppiati



PROBLEMI TERMOFLUIDODINAMICI - 1

- FVM: approccio comune per la risoluzione numerica di problemi termofluidodinamici.
- Ampia diffusione di codici commerciali basati su tale tecnica.
- Le equazioni algebriche ottenute dalla discretizzazione delle equazioni di Navier-Stokes, ed altre grandezze, potrebbero venire risolte *simultaneamente*, oppure in modo *sequenziale*: la soluzione procede, per ogni variabile, *congelando* le altre al passo di tempo o iterazione precedente.
- I primi codici FVM utilizzavano l'approccio sequenziale, o *segregato*, e sole griglie strutturate di tipo Cartesiano.
- Estensione alle griglie strutturate curvilinee, prima monoblocco, poi multi-blocco.
- FVM per griglie non strutturate:
 - ▶ generazione completamente *automatica* della griglia per geometrie complesse;
 - ▶ *importazione* di geometrie CAD;
 - ▶ *griglie adattive*;
 - ▶ metodologie di soluzione di tipo *segregato* ed *accoppiato*.



PROBLEMI TERMOFLUIDODINAMICI - 2

- Con il FVM, si utilizzano procedure di calcolo di tipo iterativo per problemi stazionari o, per problemi non-stazionari, si fa ricorso a tecniche di integrazione di tipo implicito.
- Metodo *SIMPLE* (Semi-Implicit Method for Pressure-Linked Equations), e derivati da questo, come SIMPLER, SIMPLEC, SIMPLEST *etc.*:
 - ▶ Non differiscono sostanzialmente dai procedimenti di calcolo di tipo *Projection*;
 - ▶ Per ragioni di omogeneità faremo riferimento ai metodi di proiezione.
- Metodi di Proiezione: determinazione di una equazione differenziale per la pressione a partire dal vincolo di conservazione della massa.
- Proiezione: il campo di velocità approssimato, trovato nella prima fase del calcolo, viene successivamente *proiettato* in un campo a divergenza nulla tenendo conto delle correzioni di pressione.
- Differenze fra le equazioni di Navier-Stokes e la generica equazione di trasporto: conseguenze delle possibili distribuzioni delle variabili velocità e pressione sulla griglia.



Disposizione delle variabili sulla griglia - 1

Le equazioni di Navier-Stokes e della continuità, in forma conservativa in coordinate Cartesiane bidimensionali:

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (\rho u u) + \frac{\partial}{\partial y} (\rho v u) = - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) - \rho_0 \beta (t - t_0) g_x$$

$$\frac{\partial}{\partial t} (\rho v) + \frac{\partial}{\partial x} (\rho u v) + \frac{\partial}{\partial y} (\rho v v) = - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) - \rho_0 \beta (t - t_0) g_y$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

- Termine sorgente - gradiente di pressione - richiede particolare attenzione.
- Il termine di galleggiamento non dà luogo a particolari problemi.



Disposizione delle variabili sulla griglia - 2

Consideriamo la prima delle due equazioni di Navier-Stokes.

- Con il FVM il termine relativo al gradiente di pressione viene usualmente espresso (approccio conservativo) come forza di superficie:

$$- \int_V \frac{\partial p}{\partial x} dV = - \left[\int_{A_e} p dA - \int_{A_w} p dA \right] \approx \Delta y_j (p_w - p_e)$$

- Usando l'interpolazione lineare (differenze centrali), avremo:

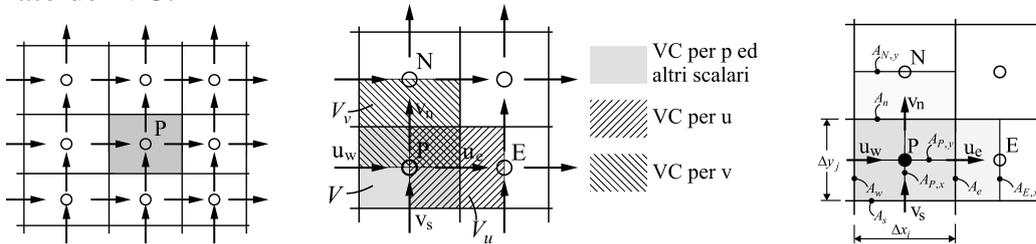
$$\Delta y_j (p_w - p_e) \approx \Delta y_j \left(\frac{p_W + p_P}{2} - \frac{p_P + p_E}{2} \right) = \Delta y_j \left(\frac{p_W - p_E}{2} \right)$$

- Risulta quindi che il termine relativo alla pressione viene valutato utilizzando una griglia due volte più grossolana.
- Debole accoppiamento fra il campo delle velocità e quello delle pressioni: campo di pressione a scacchiera (*checkerboard pressure field*).
- È quindi necessario eliminare questo inconveniente.



Disposizione delle variabili sulla griglia - 3

- Utilizzo di griglie diverse, per le componenti di velocità, rispetto a quella, principale.
- Griglie ottenute traslando (sfalsando) i VC della griglia principale, per ciascuna componente di velocità e nella direzione corrispondente, di una quantità pari a metà del lato del VC.



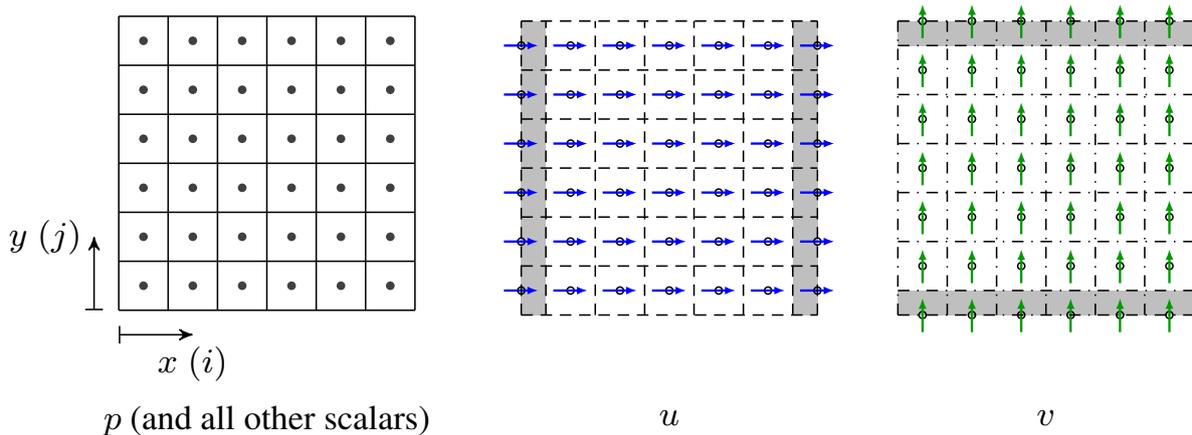
- Con questa disposizione delle variabili (*staggered grid*), l'integrazione del termine relativo al gradiente di pressione per u_e fornisce:

$$-\int_{V_e} \frac{\partial p}{\partial x} dV = -\left[\int_{A_E} p dA - \int_{A_P} p dA \right] \approx \Delta y_j (p_P - p_E)$$

- Assenza di campi di pressione *checkerboarded*: il gradiente viene valutato con riferimento alla pressione in due nodi adiacenti.



Staggered grids

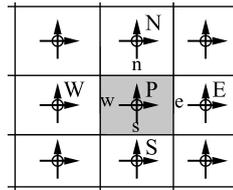


- Note the *half-cells* (shaded) for both u (x-velocity component) and v (y-velocity component).
- A similar cell pattern holds for w (z-velocity component).



Disposizione delle variabili sulla griglia - 4

- Le griglie sfalsate nel FVM corrispondono all'utilizzo di funzioni di forma di ordine inferiore per la pressione (*unequal order interpolation*) nel FEM.
- Ulteriori vantaggi offerti da tale tipo di griglia:
 - ▶ Alcuni termini delle equazioni, che richiederebbero altrimenti delle interpolazioni, possono venire valutati, con un'accuratezza del second'ordine, in modo semplice.
 - ▶ Viene garantita la conservazione dell'energia cinetica.
- Difficoltà di utilizzo delle griglie staggered di tipo curvilineo, ed ancor più non strutturate.
- Utilizzo della distribuzione *co-locata* (collocated) delle variabili:

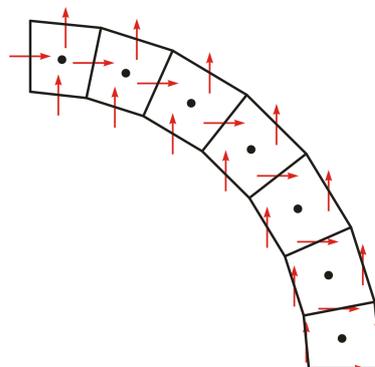


- La proprietà di conservazione dell'energia cinetica delle griglie sfalsate, le rende particolarmente indicate in LES e DNS.



Difficoltà nell'uso di griglie sfalsate - 1

- L'uso di griglie *staggered*, come visto, risolve il problema del *checkerboarding* per le griglie strutturate Cartesiane.
- Tuttavia essa non è facilmente applicabile al caso delle griglie curvilinee o non strutturate, come illustrato in figura:

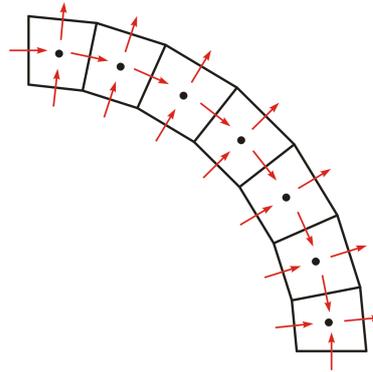


In tal caso, infatti, le componenti Cartesiane u e v della velocità, disposte sempre nei centroidi delle facce, non rappresentano più i flussi normali a queste.



Difficoltà nell'uso di griglie sfalsate -2

- Tale difficoltà potrebbe essere risolta utilizzando le componenti di velocità *controvarianti*, orientate localmente secondo la griglia:



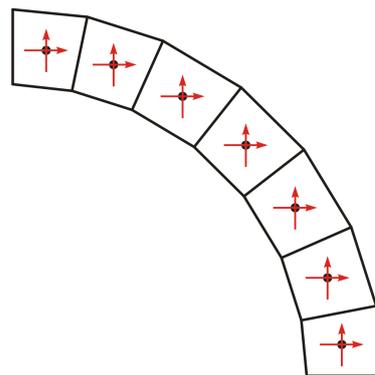
Sebbene tale approccio sia stato - ed in parte lo è ancora - utilizzato per griglie strutturate curvilinee, esso presenta almeno due inconvenienti:

- Le equazioni di conservazione, e la corrispondente derivazione delle equazioni discrete, diventa complessa, in particolare in 3D;
- È necessario memorizzare le informazioni geometriche (metriche) per 3 (in 2D) o 4 (in 3D) griglie;
- La sua estensione a griglie non-strutturate arbitrarie, se possibile, presenta non poche difficoltà.



Difficoltà nell'uso di griglie sfalsate -3

- La soluzione più semplice consiste nell'adozione della disposizione co-locata di *tutte* le variabili:



- Tuttavia sorge il problema del debole accoppiamento pressione-velocità, e del conseguente possibile campo di pressione *checker-boarded*).

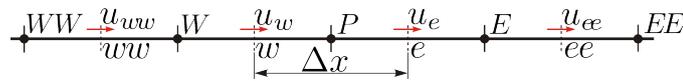
Che fare ?

- In quanto segue vedremo una possibile strategia per evitare questo inconveniente, limitando la discussione al solo caso Cartesiano, rimandando alla letteratura l'estensione al caso di griglie non strutturate arbitrarie.



Griglie co-locate - 1

- Prima di passare al modo di trattare l'accoppiamento velocità-pressione su griglie co-locate, ricordiamo brevemente il funzionamento della disposizione sfalsata delle variabili, che per semplicità illustriamo, con riferimento alla equazione di q.d.m. lungo x , su una griglia sfalsata 1D:



L'equazione discretizzata di q.d.m. lungo x per la velocità u_e può essere scritta, in forma compatta con notazione geografica:

$$A_{p_e}^u u_e + \sum_{nb} A_{nb}^u u_{nb} = \Delta y_j (p_P - p_E) + S_{p_e}^u$$

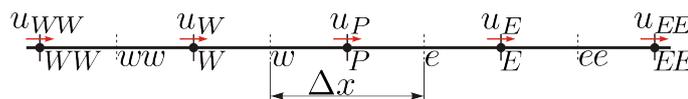
dove u_e rappresenta la componente di velocità lungo x disposta sulla faccia e .

- La differenza di pressione che contribuisce alla equazione di q.d.m. è $(p_P - p_E)$, che sta proprio a cavallo del nodo e :
 - la velocità u_e , lo ricordiamo, giace proprio sulla faccia e del volume di controllo principale;
 - l'equazione di continuità è applicata al volume di controllo della griglia principale per derivare l'equazione di correzione della pressione, nel metodo di proiezione o nei metodi basati sull'algoritmo SIMPLE.



Griglie co-locate - 2

- Consideriamo ora la disposizione co-locata delle variabili, illustrata anche qua, per semplicità, per una griglia 1D:



L'equazione discretizzata di q.d.m. lungo x per la velocità u_P risulta:

$$A_{P_P}^u u_P + \sum_{nb} A_{nb}^u u_{nb} = \Delta y_j (p_w - p_e) + S_P^u$$

- Avevamo già visto che, utilizzando un'interpolazione lineare, si ottiene:

$$\Delta y_j (p_w - p_e) \approx \left(\frac{p_W + p_P}{2} - \frac{p_P + p_E}{2} \right) = \Delta y_j \left(\frac{p_W - p_E}{2} \right)$$

che sostituita nella precedente fornisce:

$$A_{P_P}^u u_P + \sum_{nb} A_{nb}^u u_{nb} = \boxed{\frac{\Delta y_j}{2} (p_W - p_E)} + S_P^u$$



Griglie co-locate - 3

- Esplicitando u_P dalla precedente equazione si ha:

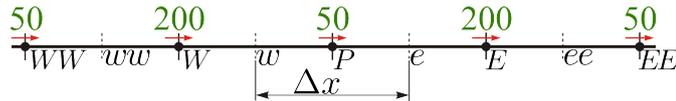
$$u_P = \frac{S_P^u - \sum_{nb} A_{nb}^u u_{nb}}{A_{PP}^u} + \frac{d_P^u}{2} (p_W - p_E)$$

con $d_P = \Delta y_j / A_{PP}^u$. Analogamente per la velocità u_E :

$$u_E = \frac{S_E^u - \sum_{nb} A_{nb}^u u_{nb}}{A_{PE}^u} + \frac{d_E^u}{2} (p_P - p_{EE})$$

con $d_E = \Delta y_j / A_{PE}^u$.

- Si consideri ora un campo di pressione *rapidamente variabile* come in figura:



- Ci si aspetterebbe che le differenze di pressione fra i nodi P ed E , e fra i nodi W e P , diano luogo a *flussi entranti* nel volume di controllo in P attraverso le facce e e w .
- Tuttavia, come visto, la differenza di pressione ($p_W - p_E$) è zero, e ad analoghe conclusioni si giunge considerando le equazioni per le celle W e E .
- Questo è il problema del *checkerboarding*, responsabile di soluzioni non fisiche come conseguenza dell'interpolazione lineare per le differenze di pressione nella disposizione co-locata delle variabili.



Griglie co-locate - 4

- Ricordiamo che le velocità sulle facce e e w sono usate nell'equazione di Poisson per il calcolo della correzione di pressione:
 - Com'è facilmente intuibile, se la differenza di pressione non è adeguatamente *rappresentata* nelle velocità sulle facce, è ragionevole attendersi che l'equazione di correzione della pressione *non* potrà fornire una soluzione corretta.
 - Ad esempio, la velocità u_e nella disposizione co-locata viene ottenuta utilizzando le velocità u_E e u_P fornite dalle relazioni appena viste.
 - Ne segue che l'espressione di u_e *non* conterrà il gradiente di pressione attraverso le facce del volume di controllo - ($p_W - p_P$) e ($p_P - p_E$) - ma piuttosto il gradiente valutato sulla griglia *rada* (o doppia) - ($p_W - p_E$) e ($p_P - p_{EE}$), con la conseguenza che la soluzione risultante possa avere carattere oscillatorio.
- Per ovviare a tale inconveniente diversi autori nel 1981 (Hsu; Prakash; Rhie) riportarono i primi successi nell'implementazione di schemi basati sull'equazione di correzione della pressione su griglie co-locate.
- La derivazione qua riportata segue il lavoro originale di Rhie e Chow (1983).



Schema di Rhie e Chow - 1

- Lo schema di Rhie e Chow consiste nell'includere un termine di ordine superiore per esprimere u_e nell'equazione di correzione della pressione:

Rhie & Chow interpolation

$$u_e = \left[\frac{u_P + u_E}{2} \right] + \frac{1}{2} (d_P + d_E) (p_P - p_E) - \frac{1}{4} d_P (p_W - p_E) - \frac{1}{4} d_E (p_P - p_{EE})$$

- La velocità u_e è ora legata alla differenza di pressione ($p_P - p_E$) attraverso la faccia.
 - Il primo termine a destra è il valore medio della velocità in e .
 - Dal confronto fra tale relazione e la somma delle espressioni viste in precedenza per u_E e u_P , è agevole verificare che gli ultimi due termini a destra rimuovono gli effetti indesiderati dovuti alle differenze di pressione fra celle non adiacenti: essi, infatti, sono rimpiazzati, nel secondo termine a destra, da un contributo che tiene conto della differenza di pressione ($p_P - p_E$) attraverso la faccia.
- Rimane da dimostrare che i tre termini aggiuntivi nella espressione di u_e ora vista, rappresentano un termine di ordine elevato, tale quindi da non pregiudicare l'accuratezza della soluzione.



Schema di Rhie e Chow - 2

- Assumendo per semplicità, ma senza nulla togliere alla generalità di quanto segue, che i termini " d " siano costanti, riscriviamo l'espressione di u_e come segue:

$$\begin{aligned} u_e &= \frac{u_P + u_E}{2} + d(p_P - p_E) - \frac{1}{4}d(p_W - p_E) - \frac{1}{4}d(p_P - p_{EE}) \\ &= \frac{u_P + u_E}{2} + \boxed{\frac{d}{4} [3p_P - 3p_E + p_{EE} - p_W]} \end{aligned}$$

- Per valutare il significato del termine di pressione aggiuntivo, consideriamo la derivata terza della pressione attraverso la faccia e :

$$\begin{aligned} \left. \frac{\partial^3 p}{\partial x^3} \right|_e &= \frac{\partial}{\partial x} \left(\frac{\partial^2 p}{\partial x^2} \right) \approx \frac{1}{\Delta x} \left[\left(\frac{\partial^2 p}{\partial x^2} \right)_E - \left(\frac{\partial^2 p}{\partial x^2} \right)_P \right] \\ &\approx \frac{1}{\Delta x} \left[\frac{p_{EE} - 2p_E + p_P}{\Delta x^2} - \frac{p_E - 2p_P + p_W}{\Delta x^2} \right] \\ &\approx \frac{1}{\Delta x^3} [p_{EE} - 3p_E + 3p_P - p_W] \end{aligned}$$



Schema di Rhie e Chow - 3

- Da quanto visto:

$$(p_{EE} - 3p_E + 3p_P - p_W) \approx \frac{\partial^3 p}{\partial x^3} \Delta x^3$$

- Confrontando quindi questa espressione con quella del termine di pressione aggiuntivo dell'interpolazione di Rhie e Chow, risulta che:

Ordine del termine di pressione aggiuntivo

$$u_e = \frac{u_P + u_E}{2} + \frac{d}{4} \frac{\partial^3 p}{\partial x^3} \Big|_e \Delta x^3$$

Ciò dimostra che l'interpolazione di Rhie e Chow consiste nell'aggiunta di un termine del terz'ordine del gradiente di pressione:

- Poiché il metodo è *complessivamente* al più accurato al secondo ordine, tale aggiunta non compromette l'accuratezza della soluzione.
- I suoi effetti benefici consistono nello smorzamento delle oscillazioni spurie causate dalla disposizione co-locata delle variabili, e per tale motivo esso è usualmente chiamato, in letteratura, *pressure smoothing term* o *added dissipation term* o ancora *pressure redistribution term*.
- Lo smorzamento è dovuto al ritrovato collegamento fra le differenze di pressione attraverso le facce della cella, e la velocità sulle facce che appare nell'equazione di correzione di pressione (continuità).



Schema di Rhie e Chow - 4

- La procedura di interpolazione di Rhie e Chow ora vista può essere estesa, senza particolari difficoltà, a griglie strutturate curvilinee ed a griglie non strutturate arbitrarie.
- Essa si è rivelata particolarmente efficace nelle griglie co-locate strutturate e non strutturate; per tale ragione essa è adottata, pur con numerose modifiche e varianti, in numerosi programmi commerciali ed Open Source ai volumi finiti.
- Ricordiamo nuovamente che l'interpolazione dev'essere usata **solo** nella valutazione delle componenti di velocità sulle facce per determinare l'errore di continuità nell'equazione di correzione della pressione, e **non** nel calcolo dei flussi nell'equazione di quantità di moto o nelle relazioni per il calcolo delle correzioni di velocità.



Metodi segregati - 1

- Deflussi bidimensionali di fluidi con proprietà costanti.
- Termine di galleggiamento: ipotesi di Oberbeck-Boussinesq.
- Calcolo delle componenti della velocità di tentativo (o di stima) u^* e v^* , assumendo la pressione di stima p^* pari a p^n :

$$\begin{aligned} \rho \frac{u^* - u^n}{\Delta\vartheta} + \nabla \cdot (\rho \mathbf{w}^n \gamma u^*) + \nabla \cdot (\rho \mathbf{w}^n (1 - \gamma) u^n) \\ = \nabla \cdot (\mu \nabla \gamma u^*) + \nabla \cdot (\mu \nabla (1 - \gamma) u^n) - \frac{\partial p^*}{\partial x} - \rho \beta (t^n - t_0) g_x \end{aligned}$$

$$\begin{aligned} \rho \frac{v^* - v^n}{\Delta\vartheta} + \nabla \cdot (\rho \mathbf{w}^n \gamma v^*) + \nabla \cdot (\rho \mathbf{w}^n (1 - \gamma) v^n) \\ = \nabla \cdot (\mu \nabla \gamma v^*) + \nabla \cdot (\mu \nabla (1 - \gamma) v^n) - \frac{\partial p^*}{\partial y} - \rho \beta (t^n - t_0) g_y \end{aligned}$$

dove i termini convettivi sono linearizzati secondo il metodo di *Picard*:

$$\rho \mathbf{w} \gamma \mathbf{w} \approx (\rho \mathbf{w}^n) \gamma \mathbf{w}$$



Metodi segregati - 2

- Portando a destra dell' uguale i termini che non contribuiscono alla matrice dei coefficienti, ed integrando sui rispettivi volumi di controllo si ottiene:

$$\begin{aligned} \frac{\rho}{\Delta\vartheta} \int_{V_u} u^* dV + \rho \gamma \int_{A_u} u^* \mathbf{w}^n \cdot \mathbf{n} dA - \mu \gamma \int_{A_u} \nabla u^* \cdot \mathbf{n} dA \\ = -\rho (1 - \gamma) \int_{A_u} u^n \mathbf{w}^n \cdot \mathbf{n} dA + \mu (1 - \gamma) \int_{A_u} \nabla u^n \cdot \mathbf{n} dA \\ - \left[\int_{A_{E,x}} p^* dA - \int_{A_{P,x}} p^* dA \right] - \rho \beta g_x \int_{V_u} (t^n - t_0) dV + \frac{\rho}{\Delta\vartheta} \int_{V_u} u^n dV \end{aligned}$$

$$\begin{aligned} \frac{\rho}{\Delta\vartheta} \int_{V_v} v^* dV + \rho \gamma \int_{A_v} v^* \mathbf{w}^n \cdot \mathbf{n} dA - \mu \gamma \int_{A_v} \nabla v^* \cdot \mathbf{n} dA \\ = -\rho (1 - \gamma) \int_{A_v} v^n \mathbf{w}^n \cdot \mathbf{n} dA + \mu (1 - \gamma) \int_{A_v} \nabla v^n \cdot \mathbf{n} dA \\ - \left[\int_{A_{N,y}} p^* dA - \int_{A_{P,y}} p^* dA \right] - \rho \beta g_y \int_{V_v} (t^n - t_0) dV + \frac{\rho}{\Delta\vartheta} \int_{V_v} v^n dV \end{aligned}$$



Metodi segregati - 3

- La correzione di pressione viene calcolata risolvendo l'equazione di Poisson corrispondente che, integrata sul volume di controllo della pressione fornisce:

$$\int_A \nabla p' \cdot \mathbf{n} dA = \frac{\rho}{\Delta\vartheta} \int_A \mathbf{w}^* \cdot \mathbf{n} dA$$

che in forma discreta, utilizzando la regola del punto medio, diventa:

$$\begin{aligned} \int_{A_e} \left. \frac{\partial p'}{\partial x} \right|_e dy - \int_{A_w} \left. \frac{\partial p'}{\partial x} \right|_w dy + \int_{A_n} \left. \frac{\partial p'}{\partial y} \right|_n dx - \int_{A_s} \left. \frac{\partial p'}{\partial y} \right|_s dx \\ = \left\{ [u_e^* - u_w^*] \Delta y_j + [v_n^* - v_s^*] \Delta x_i \right\} \frac{\rho}{\Delta\vartheta} \end{aligned}$$

- I gradienti delle pressioni di correzione sulle facce vengono valutati secondo uno schema alle differenze centrali, ottenendo:

$$\begin{aligned} \left[\frac{p'_E - p'_P}{x_E - x_P} - \frac{p'_P - p'_W}{x_P - x_W} \right] \Delta y_j + \left[\frac{p'_N - p'_P}{y_N - y_P} - \frac{p'_P - p'_S}{y_P - y_S} \right] \Delta x_i \\ = \left\{ [u_{i,j}^* - u_{i-1,j}^*] \Delta y_j + [v_{i,j}^* - v_{i,j-1}^*] \Delta x_i \right\} \frac{\rho}{\Delta\vartheta} \end{aligned}$$



Metodi segregati - 4

- Le componenti u' e v' delle correzioni di velocità vengono calcolate in funzione del gradiente della correzione di pressione:

$$\begin{aligned} \frac{\rho}{\Delta\vartheta} \int_{V_u} u' dV &= - \left[\int_{A_{E,x}} p' dA - \int_{A_{P,x}} p' dA \right] \\ \frac{\rho}{\Delta\vartheta} \int_{V_v} v' dV &= - \left[\int_{A_{N,y}} p' dA - \int_{A_{P,y}} p' dA \right] \end{aligned}$$

- In forma discreta, ed utilizzando le più semplici espressioni del second'ordine per l'integrazione e l'interpolazione, fornisce:

$$\begin{aligned} u'_e &= - \frac{\Delta\vartheta}{\rho} \frac{(p'_P - p'_E)}{(x_P - x_E)} \\ v'_n &= - \frac{\Delta\vartheta}{\rho} \frac{(p'_P - p'_N)}{(y_P - y_N)} \end{aligned}$$



Metodi segregati - 5

- I valori della pressione e delle componenti di velocità alla fine del passo temporale sono dati dalle:

$$\begin{aligned}p_P^{n+1} &= p_P^* + p'_P \\u_e^{n+1} &= u_e^* + u'_e \\v_n^{n+1} &= v_n^* + v'_n\end{aligned}$$

- Per ciò che riguarda la temperatura, questa viene ottenuta integrando sul VC l'equazione dell'energia, utilizzando lo stesso algoritmo di integrazione temporale utilizzato per le componenti di velocità:

$$\begin{aligned}\frac{\rho c_p}{\Delta\vartheta} \int_V t^{n+1} dV + \rho c_p \gamma \int_A t^{n+1} \mathbf{w}^n \cdot \mathbf{n} dA - \lambda \gamma \int_A \nabla t^{n+1} \cdot \mathbf{n} dA \\= -\rho c_p (1 - \gamma) \int_A t^n \mathbf{w}^n \cdot \mathbf{n} dA + \lambda (1 - \gamma) \int_A \nabla t^n \cdot \mathbf{n} dA \\+ \frac{\rho c_p}{\Delta\vartheta} \int_V t^n dV\end{aligned}$$



Metodi segregati - 6

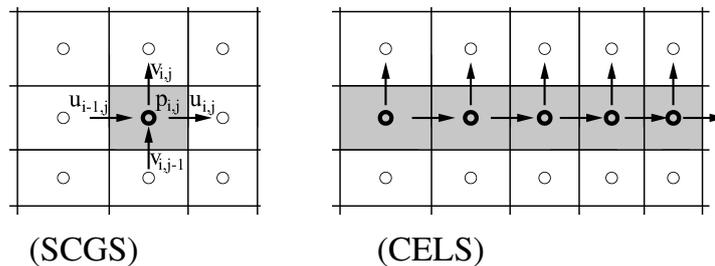
Osservazioni

- In analogia con il FEM, le proprietà termofisiche sono state ritenute costanti.
- L'algoritmo di integrazione temporale tipo *projection* ora visto presenta molte varianti possibili, quale ad esempio quella data dalla combinazione di uno schema Adams-Bashfort esplicito per il termine convettivo, ed uno schema Gear (Eulero second'ordine) implicito per il flusso diffusivo.
- La metodologia illustrata può servire, viste le notevoli somiglianze, come riferimento per altri algoritmi di soluzione, come il SIMPLE e derivati.



Metodi accoppiati - 1

- I metodi segregati possono dar luogo a problemi di convergenza quando l'accoppiamento fra le variabili è molto pronunciato.
- Necessità di dover adottare ridotti passi di integrazione temporale o valori piccoli dei coefficienti di sottorilassamento.
- Per queste situazioni può risultare conveniente utilizzare una strategia di soluzione che preservi, in modo *implicito*, l'accoppiamento fra le variabili.
- Le variabili risolte simultaneamente appartengono tipicamente ad opportuni sotto-domini, che vengono *visitati* in sequenza. I sotto-domini sono usualmente costituiti da:
 - 1 Un singolo VC. Metodo SCGS (Symmetrically Coupled Gauss-Seidel). Può essere utilizzato anche per griglie non strutturate.
 - 2 Una linea, riga o colonna, di VC. Ad esempio CELS (Coupled Equation Line Solver). Limitato a griglie strutturate.



Metodi accoppiati - 2

Metodo SCGS, sole variabili pressione e velocità nel caso 2D stazionario

$$\begin{aligned}
 & A_{P_{i,j}}^u u_{i,j} + A_{E_{i,j}}^u u_{i+1,j} + A_{W_{i,j}}^u u_{i-1,j} + A_{N_{i,j}}^u u_{i,j+1} \\
 & \quad + A_{S_{i,j}}^u u_{i,j-1} + (p_{i+1,j} - p_{i,j}) \Delta y_j = S_{i,j}^u \\
 & A_{P_{i-1,j}}^u u_{i-1,j} + A_{E_{i-1,j}}^u u_{i,j} + A_{W_{i-1,j}}^u u_{i-2,j} + A_{N_{i-1,j}}^u u_{i-1,j+1} \\
 & \quad + A_{S_{i-1,j}}^u u_{i-1,j-1} + (p_{i,j} - p_{i-1,j}) \Delta y_j = S_{i-1,j}^u \\
 & A_{P_{i,j}}^v v_{i,j} + A_{E_{i,j}}^v v_{i+1,j} + A_{W_{i,j}}^v v_{i-1,j} + A_{N_{i,j}}^v v_{i,j+1} \\
 & \quad + A_{S_{i,j}}^v v_{i,j-1} + (p_{i,j+1} - p_{i,j}) \Delta x_i = S_{i,j}^v \\
 & A_{P_{i,j-1}}^v v_{i,j-1} + A_{E_{i,j-1}}^v v_{i+1,j-1} + A_{W_{i,j-1}}^v v_{i-1,j-1} + A_{N_{i,j-1}}^v v_{i,j} \\
 & \quad + A_{S_{i,j-1}}^v v_{i,j-2} + (p_{i,j} - p_{i,j-1}) \Delta x_i = S_{i,j-1}^v \\
 & (u_{i,j} - u_{i-1,j}) \Delta y_j + (v_{i,j} - v_{i,j-1}) \Delta x_i = 0
 \end{aligned}$$

- Il termine relativo al gradiente di pressione è stato scritto in modo esplicito.
- Sistema lineare di cinque equazioni nelle incognite $u_{i-1,j}$, $u_{i,j}$, $v_{i,j-1}$, $v_{i,j}$ e $p_{i,j}$.



Metodi accoppiati - 3

Per poter semplificare il sistema, al fine di ottenere economicamente la soluzione per via analitica, si adotta la seguente strategia:

- ① Per le equazioni di quantità di moto, si portano nel termine noto tutti i contributi, con l'esclusione del termine diagonale A_P e del contributo della pressione $p_{i,j}$.
- ② L'equazione di continuità viene mantenuta in forma completa.

Così facendo si ottiene il seguente sistema:

$$\begin{bmatrix} A_{P_{i-1,j}}^u & 0 & 0 & 0 & \Delta y_j \\ 0 & A_{P_{i,j}}^u & 0 & 0 & -\Delta y_j \\ 0 & 0 & A_{P_{i,j-1}}^v & 0 & \Delta x_i \\ 0 & 0 & 0 & A_{P_{i,j}}^v & -\Delta x_i \\ -\Delta y_j & \Delta y_j & -\Delta x_i & \Delta x_i & 0 \end{bmatrix} \begin{pmatrix} u_{i-1,j} \\ u_{i,j} \\ v_{i,j-1} \\ v_{i,j} \\ p_{i,j} \end{pmatrix} = \begin{pmatrix} b_{i-1,j}^u \\ b_{i,j}^u \\ b_{i,j-1}^v \\ b_{i,j}^v \\ 0 \end{pmatrix}$$



Metodi accoppiati - 4

I termini noti contengono i contributi che non appaiono esplicitamente in forma matriciale, ad esempio:

$$b_{i,j}^u = -A_{E_{i,j}}^u u_{i+1,j}^* - A_{W_{i,j}}^u u_{i-1,j}^* - A_{N_{i,j}}^u u_{i,j+1}^* - A_{S_{i,j}}^u u_{i,j-1}^* - p_{i+1,j}^* \Delta y_j + S_{i,j}^u$$

Soluzione del sistema di equazioni:

$$r_1 = -\Delta y_j / A_{P_{i-1,j}}^u; \quad r_2 = \Delta y_j / A_{P_{i,j}}^u; \quad r_3 = -\Delta x_i / A_{P_{i,j-1}}^v; \quad r_4 = \Delta x_i / A_{P_{i,j}}^v$$

$$\text{DEN} = r_1 \Delta y_j - r_2 \Delta y_j + r_3 \Delta x_i - r_4 \Delta x_i$$

$$p_{i,j} = (r_1 b_{i-1,j}^u + r_2 b_{i,j}^u + r_3 b_{i,j-1}^v + r_4 b_{i,j}^v) / \text{DEN}$$

$$u_{i-1,j} = (b_{i-1,j}^u - \Delta y_j p_{i,j}) / A_{P_{i-1,j}}^u; \quad u_{i,j} = (b_{i,j}^u + \Delta y_j p_{i,j}) / A_{P_{i,j}}^u$$

$$v_{i,j-1} = (b_{i,j-1}^v - \Delta x_i p_{i,j}) / A_{P_{i,j-1}}^v; \quad v_{i,j} = (b_{i,j}^v + \Delta x_i p_{i,j}) / A_{P_{i,j}}^v$$

- SCGS applicabile anche alle griglie non strutturate, e altre variabili.
- La convenienza dei metodi accoppiati deriva, soprattutto, dal loro utilizzo contestuale ad un metodo *multigrid*.



Geometrie complesse: griglie non strutturate

17 GRIGLIE NON STRUTTURATE

- Considerazioni generali
- Mesh quality and metrics
- Scelta delle componenti di velocità
- Disposizione delle variabili sulla griglia
- Quantità geometriche
- Cell types
- Distribuzione spaziale delle variabili
- Integrazione
- Calcolo del gradiente
- Termine transitorio e termine sorgente
- Flusso convettivo
- Flusso diffusivo

18 INITIAL AND BOUNDARY CONDITIONS

- Condizioni iniziali
- Condizioni al contorno

19 FINAL ALGEBRAIC EQUATION

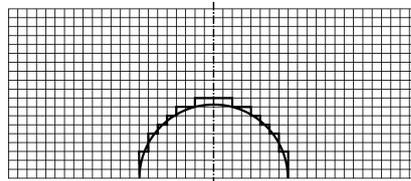
DATA STRUCTURE

- Data structure for cell-centered unstructured grids



Geometrie complesse

- La procedura illustrata per griglie strutturate Cartesiane è semplice ed accurata, e di agevole implementazione, ed infatti ha trovato applicazione in ricerche di base e nei primi codici commerciali.
- Difficoltà di utilizzo per geometrie complesse:

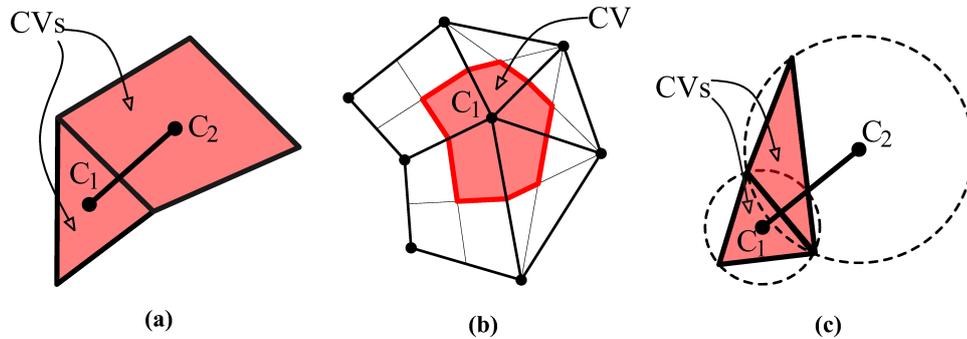


- Necessità di modificare il FVM in modo da affrontare anche tali problemi.
- Inizialmente ricorrendo a griglie strutturate curvilinee (approccio ancora utilizzato in qualche programma commerciale), passando poi a griglie non strutturate.
- Tendenza attuale è quella di utilizzare griglie non strutturate: l'uso di griglie strutturate, o Cartesiane, rimane garantito, sebbene con minore efficienza computazionale.
- Vista la maggiore generalità delle griglie non strutturate, in quanto segue considereremo solo queste.



Classificazione FV

- Svariate modalità di costruzione dei Volumi Finiti (celle) a partire da griglie non strutturate.
- Le più comuni sono:
 - (a) Cell-centered
 - (b) Vertex-centered (o node-centered)
 - (c) Circum-centered

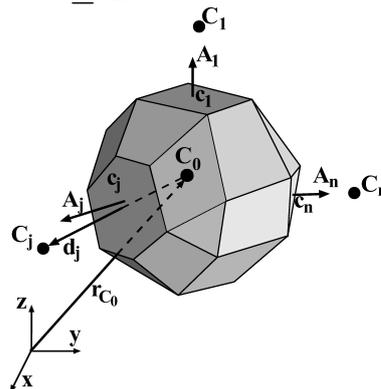


- Nel seguito, faremo riferimento al solo caso *cell-centered* (a).



Generalità e notazione

- Dominio discretizzato con una griglia non strutturata, costituita da VC di forma poliedrica arbitraria con un numero di facce $n \geq 4$:



- Unici requisiti sono che le celle devono essere *convesse* e le facce costituenti devono essere *piane*.
- Accuratezza maggiore con celle a forma di *esaedri* regolari.
- Generatori di griglia che producono mesh a poliedri:
 - ▶ L'uso di poliedri, in generale, può garantire una maggiore accuratezza, a causa del più elevato grado di ortogonalità della griglia (vedi in seguito).
 - ▶ Per griglie node-centered i VC sono già dei poliedri centrati sui nodi.
- Operando con FVM, non è necessario effettuare trasformazioni di coordinate.



Centroidi

Centroide C_0 (baricentro) del VC

$$\int_V (\mathbf{r} - \mathbf{r}_{C_0}) dV = 0$$

Centroide c_j della faccia A_j compresa fra C_0 e C_j

$$\int_{A_j} (\mathbf{r} - \mathbf{r}_j) dA = 0$$

\mathbf{A}_j è il *vettore area* della faccia:

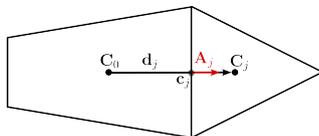
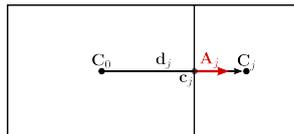
$$\mathbf{A}_j = A_j \mathbf{n}$$

- In generale $\mathbf{d}_j = \mathbf{r}_{C_j} - \mathbf{r}_{C_0}$ e \mathbf{A}_j non sono paralleli, e l'angolo da essi formato aumenta per griglie molto *distorte*, cioè *non ortogonali*.
- Il punto di applicazione di \mathbf{A}_j , cioè c_j , è in generale diverso dal punto ottenuto dall'intersezione di \mathbf{d}_j con la faccia stessa. Tale differenza costituisce la *skewness* (asimmetria, obliquità), e da la misura della deviazione della griglia attuale da una griglia ideale.

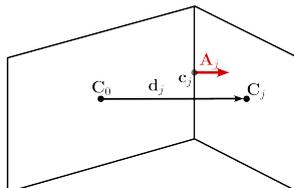


Mesh quality

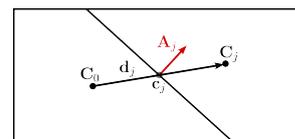
- Orthogonal and not skewed mesh



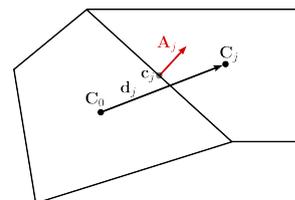
- Orthogonal and skewed mesh



- Non-orthogonal and not skewed mesh



- Non-orthogonal and skewed mesh

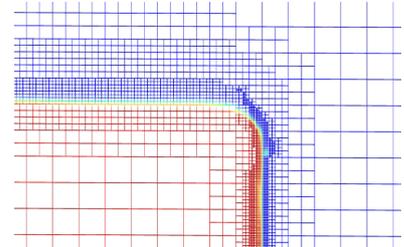
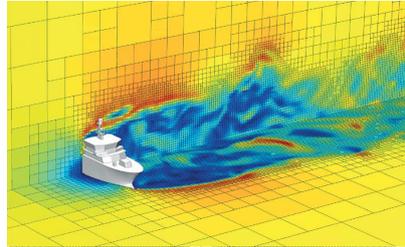
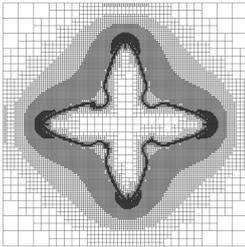


- The *non-orthogonality* add numerical diffusion to the solution, hence reducing the overall accuracy. It can also cause *wiggles*, which in turn may lead to nonphysical values and divergence.
- *Skewness* also adds numerical diffusion and reduces the accuracy. It may also leads to unboundedness, which in turn can lead to the divergence of the solution.

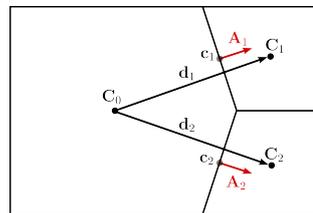
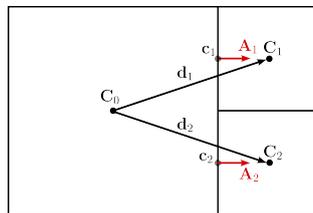


Improving the mesh

- There are several techniques available, in most mesh generators and/or CFD solvers, to improve the quality of the mesh, among these:
 - ▶ Smoothing nodes (e.g. Laplacian smoothing)
 - ▶ Swapping
 - ▶ Removing *slivers*, i.e. degenerate cells which are characterized by nodes that are nearly coplanar
- *Quadtree* and *octree* meshes:
 - ▶ They are frequently used for *adaptive meshing*



- ▶ It is possible to improve the orthogonality



Mesh metrics

- There are several ways - *metrics* - which can be used to assess the quality of the grid.
- These are usually available both in the mesher and in the solver.
 - ▶ Some of these are (from e.g. ANSYS Mesh): *Skewness*, *Orthogonal Quality*, *Element Quality*, *Aspect Ratio* for triangles or quadrilaterals, *Jacobian Ratio*, *Warping Factor*, *Parallel Deviation*, *Maximum Corner Angle*, *et.*
- For example, for *face orthogonality* a simple metric is

$$F_{ortho} = \frac{\mathbf{A}_j \cdot \mathbf{d}_j}{|\mathbf{A}_j| |\mathbf{d}_j|}$$

- The *skewness* could be based on the deviation from a normalized equilateral angle. This method can be applied to all cell and face shapes. It is defined as

$$F_{skew} = \max \left[\frac{\theta_{max} - \theta_e}{180 - \theta_e}, \frac{\theta_e - \theta_{min}}{\theta_e} \right]$$

where:

θ_{max} is the largest angle for the face or cell [deg]

θ_{min} is the smallest angle for the face or cell [deg]

θ_e is the angle for an equiangular face/cell (60° for a triangle, 90° for a square et.) [deg]



Equazione di trasporto

- È importante operare con griglie nelle quali i due vettori siano il più possibile paralleli e disposti lungo la stessa retta, e cioè con griglie il più possibile *ortogonali*.
- Nel FVM l'ortogonalità si misura fra congiungente i centroidi \mathbf{d}_j , e vettore area \mathbf{A}_j della faccia interposta.

Generica equazione di trasporto integrata sul VC:

$$\int_V \frac{\partial}{\partial t} (\rho\phi) dV + \int_A \rho\phi \mathbf{w} \cdot \mathbf{n} dA = \int_A \Gamma \nabla\phi \cdot \mathbf{n} dA + \int_V s dV$$

In analogia a quanto visto per le griglie Cartesiane, considereremo singolarmente i vari termini dell'equazione di trasporto, dopo aver prima definito alcuni aspetti della procedura di calcolo.



Scelta delle componenti di velocità

- Nel metodo FV, le equazioni della quantità di moto sono espresse in forma *conservativa*:
 - ▶ Tutti i termini delle equazioni possono venire rappresentati come divergenza di un vettore o di un tensore.
 - ▶ L'utilizzo della forma conservativa delle equazioni con il FVM, garantisce la conservazione globale della quantità di moto anche nel calcolo *discreto*.
 - ▶ Tuttavia, la forma conservativa delle equazioni richiede di esprimere le componenti della quantità di moto in un sistema di riferimento *fisso*, pena l'introduzione di campi di forza non-conservativi, secondo la definizione vista.
- Per tale ragione, è opportuno utilizzare componenti di tale tipo, ed in particolare Cartesiane, che sono le più semplici ed intuitive.
- Nel seguito, faremo riferimento a grandezze (vettori e tensori) espresse in termini di componenti Cartesiane.



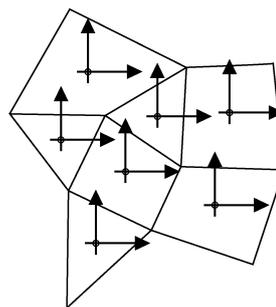
Disposizione delle variabili sulla griglia - 1

- La disposizione *sfalsata* (staggered) delle variabili sulla griglia garantisce un buon accoppiamento fra componenti di velocità e pressione.
- Ciò vale solo per griglie strutturate Cartesiane, per le quali le componenti della velocità risultano *normali* alle facce dei VC.
- Per griglie non strutturate, viceversa, sarebbe necessario memorizzare tutte le componenti della velocità su ciascuna faccia.
- L'utilizzo di componenti di velocità allineate localmente alla griglia (*componenti contravarianti*), risolve tale problema, ma non sarebbe garantita la conservatività globale della quantità di moto.



Disposizione delle variabili sulla griglia - 2

- Pratica diffusa co-locare tutte le variabili nel centroide dei VC:

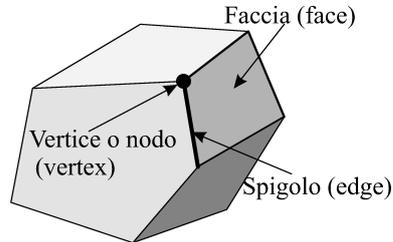


- Agevole gestione della struttura dati e programmazione (informazioni geometriche di un solo set di volumi di controllo).
- Utilizzo di opportune procedure di interpolazione per ovviare al debole accoppiamento fra il campo di velocità ed il campo di pressione.
- Nel FVM su griglie non strutturate, l'utilizzo di componenti Cartesiane della velocità, e la disposizione co-locata delle variabili, rappresenta un buon compromesso fra semplicità di implementazione, accuratezza e rispetto della conservatività (applicazione in numerosi codici, commerciali e *open source*).



Quantità geometriche

- Il dominio di calcolo viene suddiviso in un numero finito di VC contigui attraverso una griglia.
- La griglia, in analogia con il metodo degli Elementi Finiti, definisce i *vertici* (vertices) o *nodi*, che collegati due a due costituiscono a loro volta gli *spigoli* (edges) dei VC.



- Gli spigoli definiscono le *facce* (faces) dei VC, che non sono necessariamente piane; tuttavia le loro proiezioni sui piani Cartesiani risultano indipendenti dalla forma effettiva della faccia.
- Tali proiezioni sono le componenti Cartesiane del vettore area \mathbf{A} , univocamente definito sulla base delle coordinate dei vertici della faccia.

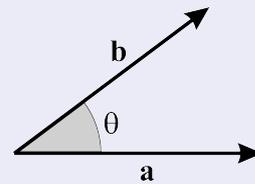


Richiami di calcolo vettoriale - 1

Prodotto scalare

Il prodotto scalare di due vettori \mathbf{a} e \mathbf{b} (*dot product*) è usualmente scritto $\mathbf{a} \cdot \mathbf{b}$, ed è definito dalla quantità scalare:

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$$



Alcune proprietà del prodotto scalare:

$$\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \cdot \mathbf{c}$$

$$\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2$$

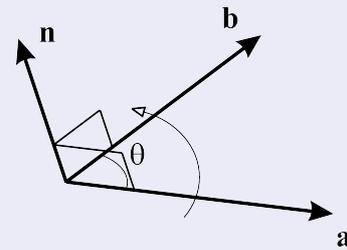
$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= (a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}) \cdot (b_x \mathbf{i} + b_y \mathbf{j} + b_z \mathbf{k}) \\ &= a_x b_x + a_y b_y + a_z b_z \end{aligned}$$



Prodotto vettoriale

Il prodotto vettoriale di due vettori \mathbf{a} e \mathbf{b} (*cross product*) è usualmente scritto $\mathbf{a} \times \mathbf{b}$ (o $\mathbf{a} \wedge \mathbf{b}$), ed è definito dalla quantità vettoriale:

$$\mathbf{a} \times \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \sin \theta \mathbf{n}$$



Alcune proprietà del prodotto vettoriale:

$$\mathbf{b} \times \mathbf{a} = -\mathbf{a} \times \mathbf{b}$$

$$\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$$

$$\mathbf{a} \times \mathbf{b} = (a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}) \times (b_x \mathbf{i} + b_y \mathbf{j} + b_z \mathbf{k})$$

$$= (a_y b_z - a_z b_y) \mathbf{i} + (a_z b_x - a_x b_z) \mathbf{j} + (a_x b_y - a_y b_x) \mathbf{k}$$

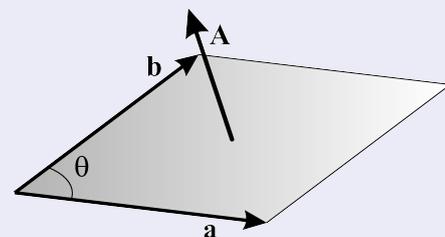
$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$



Richiami di calcolo vettoriale - 3

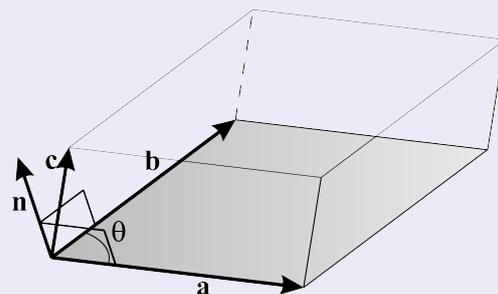
Vettore area \mathbf{A}

$$\mathbf{A} = \mathbf{a} \times \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \sin \theta \mathbf{n}$$



Volume di un parallelepipedo

$$V = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b}) = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$$



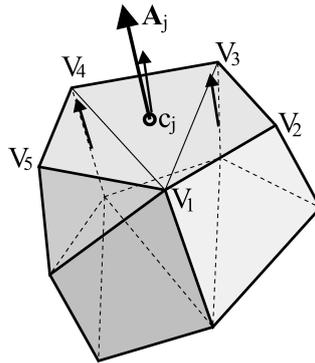
Area di una faccia

- Qualunque faccia congiungente N_V vertici V_l , con $l = 1 \dots N_V$ può essere decomposta in $N_V - 2$ triangoli con un vertice comune.

Vettore area \mathbf{A}_j della faccia

Ottenuto dalla somma dei vettori area dei triangoli che la compongono:

$$\mathbf{A}_j = \frac{1}{2} \sum_{l=3}^{N_V} [(\mathbf{r}_{V_{l-1}} - \mathbf{r}_{V_1}) \times (\mathbf{r}_{V_l} - \mathbf{r}_{V_1})]$$



Centroide di una faccia

- Per il triangolo definito dai vertici V_1 , V_2 e V_3 , la posizione del centroide è data dalla:

$$\mathbf{r}_{123} = \frac{1}{3} (\mathbf{r}_{V_1} + \mathbf{r}_{V_2} + \mathbf{r}_{V_3})$$

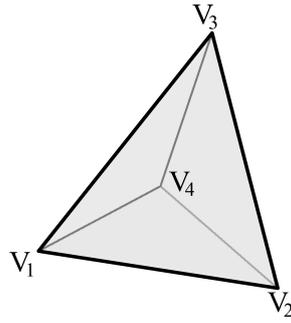
Centroide \mathbf{r}_j della faccia

Ottenuto dalla media, pesata per la rispettiva area, dei centroidi di tutti i triangoli utilizzati per la sua suddivisione:

$$\mathbf{r}_j = \frac{\frac{1}{3} \sum_{l=3}^{N_V} (\mathbf{r}_{V_1} + \mathbf{r}_{V_{l-1}} + \mathbf{r}_{V_l}) |(\mathbf{r}_{V_{l-1}} - \mathbf{r}_{V_1}) \times (\mathbf{r}_{V_l} - \mathbf{r}_{V_1})|}{\sum_{l=3}^{N_V} |(\mathbf{r}_{V_{l-1}} - \mathbf{r}_{V_1}) \times (\mathbf{r}_{V_l} - \mathbf{r}_{V_1})|}$$



Volume e centroide di un tetraedro



Volume di un tetraedro

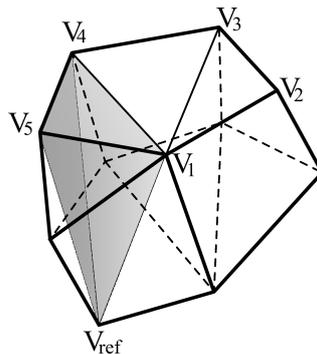
$$V_{1234} = \frac{1}{6} (\mathbf{r}_{V_3} - \mathbf{r}_{V_1}) \cdot [(\mathbf{r}_{V_2} - \mathbf{r}_{V_1}) \times (\mathbf{r}_{V_4} - \mathbf{r}_{V_1})]$$

Centroide di un tetraedro

$$\mathbf{r}_{1234} = \frac{1}{4} (\mathbf{r}_{V_1} + \mathbf{r}_{V_2} + \mathbf{r}_{V_3} + \mathbf{r}_{V_4})$$



Volume di un poliedro



Volume di un poliedro

Scomposizione in tetraedri e sommando il contributo di ciascuno di questi:

$$V = \frac{1}{6} \sum_{i=1}^{N_f} \sum_{l=3}^{N_{V,i}} (\mathbf{r}_{V_{i,1}} - \mathbf{r}_{V_{ref}}) \cdot [(\mathbf{r}_{V_{i,l-1}} - \mathbf{r}_{V_{ref}}) \times (\mathbf{r}_{V_{i,l}} - \mathbf{r}_{V_{ref}})]$$

con N_f numero facce del poliedro, $N_{V,i}$ numero vertici della faccia i -esima, e V_{ref} un vertice (o qualunque altro punto) di riferimento arbitrario.



Centroide di un poliedro

Dalla media pesata, con il rispettivo volume, dei vettori posizione dei centroidi dei tetraedri costituenti il poliedro:

$$\mathbf{r}_{C_0} = \frac{1}{24} \frac{1}{V} \sum_{i=1}^{N_f} \sum_{l=3}^{N_{V,i}} (\mathbf{r}_{V_{ref}} + \mathbf{r}_{V_{i,1}} + \mathbf{r}_{V_{i,l-1}} + \mathbf{r}_{V_{i,l}}) \{ (\mathbf{r}_{V_{i,1}} - \mathbf{r}_{V_{ref}}) \cdot [(\mathbf{r}_{V_{i,l-1}} - \mathbf{r}_{V_{ref}}) \times (\mathbf{r}_{V_{i,l}} - \mathbf{r}_{V_{ref}})] \}$$



Relazioni semplificate

- Sfruttando l'identità:

$$\nabla \mathbf{r} = \nabla (x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) \equiv \mathbf{3}$$

il volume della cella si può calcolare sfruttando il teorema di Gauss:

$$V = \frac{1}{3} \int_V \nabla \mathbf{r} \, dV = \frac{1}{3} \int_A \mathbf{r} \cdot \mathbf{n} \, dA = \frac{1}{3} \sum_{i=1}^{N_f} \mathbf{r}_i \cdot \mathbf{n}_i A_i$$

- Con considerazioni analoghe, attraverso una derivazione un pò più complessa, il centroide può valutarsi con la relazione:

$$\mathbf{r}_{C_0} = \frac{3 \sum_{i=1}^{N_f} (\mathbf{r}_i \cdot \mathbf{n}_i) \mathbf{r}_i A_i}{4 \sum_{i=1}^{N_f} \mathbf{r}_i \cdot \mathbf{n}_i A_i}$$

- Le espressioni viste sono in generale approssimate, ma risultano esatte per celle aventi facce triangolari o quadrilatere piane.



Sum of the vector areas of a triangle

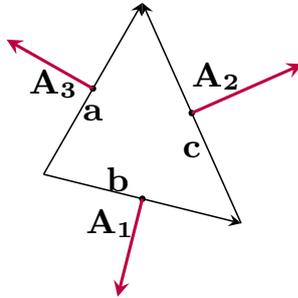
We can prove that the three vectors (with outward-pointing normals) of the faces of a triangle sum to zero.

Since

$$\mathbf{A}_1 = \mathbf{b} \times \mathbf{k}$$

$$\mathbf{A}_2 = \mathbf{c} \times \mathbf{k}$$

$$\mathbf{A}_3 = \mathbf{k} \times \mathbf{a}$$



It follows that

$$\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 = \mathbf{b} \times \mathbf{k} + \mathbf{c} \times \mathbf{k} + \mathbf{k} \times \mathbf{a}$$

but:

$$\mathbf{a} = \mathbf{b} + \mathbf{c}$$

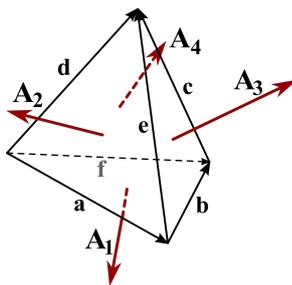
and therefore, remembering that $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$

$$\begin{aligned} \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 &= \mathbf{b} \times \mathbf{k} + \mathbf{c} \times \mathbf{k} + \mathbf{k} \times (\mathbf{b} + \mathbf{c}) \\ &= \mathbf{b} \times \mathbf{k} + \mathbf{c} \times \mathbf{k} + \mathbf{k} \times \mathbf{b} + \mathbf{k} \times \mathbf{c} \\ &= 0 \end{aligned}$$



Sum of the vector areas of a tetrahedron

We can prove that the four vector areas (with outward-pointing normals) of the faces of a tetrahedron sum to zero.



Therefore

$$\begin{aligned} \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4 &= \frac{1}{2} (\mathbf{b} \times \mathbf{a} + \mathbf{a} \times \mathbf{d} + \mathbf{b} \times \mathbf{e} + \mathbf{d} \times \mathbf{f}) \end{aligned}$$

but

$$\mathbf{d} = \mathbf{a} + \mathbf{e} \quad \mathbf{f} = \mathbf{a} + \mathbf{b}$$

it follows that

$$\mathbf{a} \times \mathbf{d} = \mathbf{a} \times (\mathbf{a} + \mathbf{e}) = \mathbf{a} \times \mathbf{a} + \mathbf{a} \times \mathbf{e}$$

$$\mathbf{d} \times \mathbf{f} = (\mathbf{a} + \mathbf{e}) \times (\mathbf{a} + \mathbf{b}) = \mathbf{a} \times \mathbf{a} + \mathbf{a} \times \mathbf{b} + \mathbf{e} \times \mathbf{a} + \mathbf{e} \times \mathbf{b}$$

and substituting

$$\begin{aligned} \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4 &= \frac{1}{2} (\mathbf{b} \times \mathbf{a} + \mathbf{a} \times \mathbf{e} + \mathbf{b} \times \mathbf{e} + \mathbf{a} \times \mathbf{b} + \mathbf{e} \times \mathbf{a} + \mathbf{e} \times \mathbf{b}) \equiv 0 \end{aligned}$$

$$\mathbf{A}_1 = \frac{1}{2} \mathbf{b} \times \mathbf{a}$$

$$\mathbf{A}_2 = \frac{1}{2} \mathbf{a} \times \mathbf{d}$$

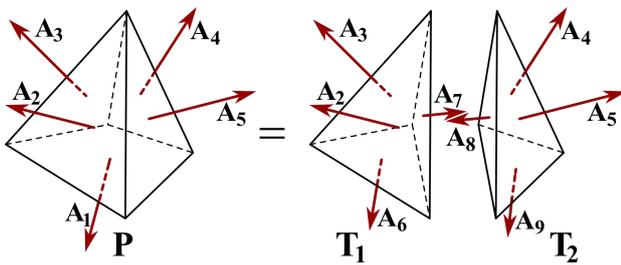
$$\mathbf{A}_3 = \frac{1}{2} \mathbf{b} \times \mathbf{e}$$

$$\mathbf{A}_4 = \frac{1}{2} \mathbf{d} \times \mathbf{f}$$



Sum of the vector areas of a pyramid

It is easy to see that the five vector areas (with outward-pointing normals) of the faces of a pyramid sum to zero by observing that it can be splitted in two tetrahedrons



\mathbf{P} = pyramid; \mathbf{T}_1 = tetrahedron; \mathbf{T}_2 = tetrahedron.

$$\text{For } \mathbf{T}_1 : \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_6 + \mathbf{A}_7 = 0$$

$$\text{For } \mathbf{T}_2 : \mathbf{A}_4 + \mathbf{A}_5 + \mathbf{A}_8 + \mathbf{A}_9 = 0$$

Summing up

$$\mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_6 + \mathbf{A}_7 + \mathbf{A}_4 + \mathbf{A}_5 + \mathbf{A}_8 + \mathbf{A}_9 = 0$$

and grouping

$$\mathbf{A}_2 + \mathbf{A}_3 + (\mathbf{A}_6 + \mathbf{A}_9) + (\mathbf{A}_7 + \mathbf{A}_8) + \mathbf{A}_4 + \mathbf{A}_5 = 0$$

Observing that

$$(\mathbf{A}_6 + \mathbf{A}_9) = \mathbf{A}_1$$

$$(\mathbf{A}_7 + \mathbf{A}_8) = 0$$

it results that

$$\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4 + \mathbf{A}_5 = 0$$

In other words, the rule

$$\sum_i \mathbf{A}_i = 0$$

holds also for a square pyramid.



Sum of the vector areas of a polyhedron

The results obtained for a *tetrahedron* and a *pyramid*, can be extended to a generic *polyhedron* by observing that it can also be decomposed in *tetrahedrons* and therefore:

- The surface area of a face of the polyhedron is equal to the sum of the triangular surface areas of the corresponding tetrahedrons.
- The contribution of surface areas of the tetrahedrons which are *inside* the polyhedron will cancel since, as shown for the pyramid, they are counted twice with different sign.

The same result can be obtained also by the *Gauss* theorem.

Let's consider a polyhedron V with a surface area A , characterized by $\mathbf{A}_i, i = 1, \dots, N$ surface area vectors ($\mathbf{A}_i = \mathbf{n}_i A_i$), and assume that \mathbf{u} is a constant vector. From Gauss's theorem:

$$\mathbf{u} \sum_i \mathbf{A}_i = \int_A \mathbf{u} \cdot \mathbf{n} dA = \int_V \nabla \cdot \mathbf{u} dV$$

But, since \mathbf{u} is constant, its divergence vanishes and consequently:

$$\mathbf{u} \sum_i \mathbf{A}_i = 0$$

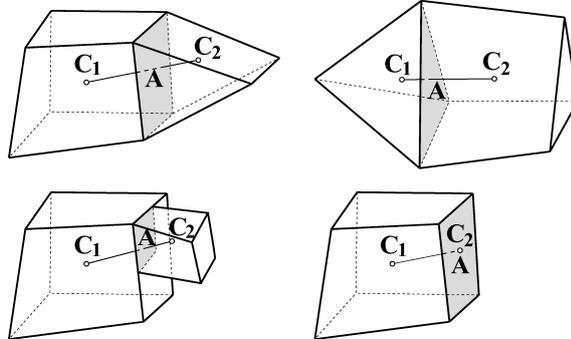
But \mathbf{u} is arbitrary, so this can happen only if:

$$\sum_i \mathbf{A}_i = 0$$



- Vista la possibilità di suddividere il dominio in VC caratterizzati da numero arbitrario di facce, è comune l'adozione, nell'implementazione, di una *struttura dati* basata sulle facce, anziché basata sui VC o sui nodi:

- ▶ Agevole considerare celle e combinazioni di celle quali quelle rappresentate in figura:

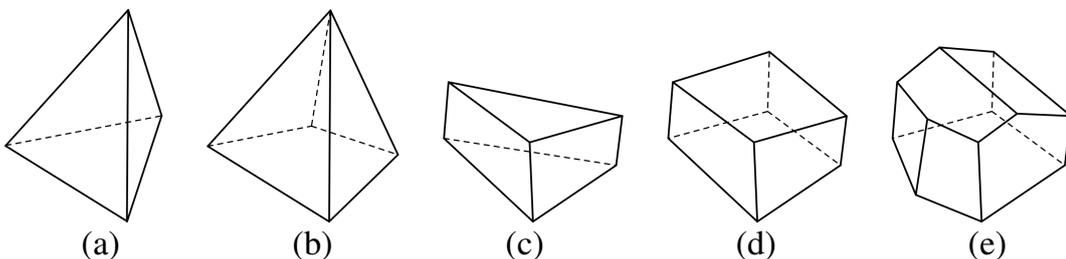


- ▶ Per ciascuna faccia è sufficiente conoscere le celle adiacenti C_1 e C_2 , ed il vettore d'area \mathbf{A} , per poter valutare i flussi convettivo e diffusivo.



Forme comuni delle celle

- Tipologie di celle più comuni - (a) tetraedro; (b) piramide; (c) prisma; (d) esaedro; (e) poliedro:



- La valutazione numerica degli integrali, di superficie e di volume, richiede la conoscenza delle coordinate dei centroidi dei VC, C_j , e delle facce, c_j .
- È inoltre necessario conoscere il vettore d'area di ogni faccia, \mathbf{A}_j , e il volume V del VC.
- Questi dati vengono in genere forniti dai generatori di griglia, altrimenti si può procedere alla loro valutazione, attraverso le relazioni viste, note le coordinate dei vertici V_j del VC.



Distribuzione spaziale delle variabili - 1

- Le variabili dipendenti, ed i valori delle proprietà termofisiche, sono definite nei centroidi dei VC.
- Tuttavia, per calcolare gli integrali di superficie, è necessario disporre del loro valore nei centroidi delle facce.
- Si suppone una variazione lineare della variabile ϕ all'interno della cella:

$$\phi = \phi_{C_0} + (\nabla\phi)_{C_0} \cdot (\mathbf{r} - \mathbf{r}_{C_0})$$

dove $(\nabla\phi)_{C_0}$ indica il gradiente valutato in C_0 , costante su tutto il volume di controllo.

- L'espressione vista fornisce un valore diverso di ϕ sulla faccia, a seconda di quale dei due VC questa si consideri appartenente.
- Formulazione più complessa che, in analogia al caso Cartesiano, garantisce perfetta simmetria.

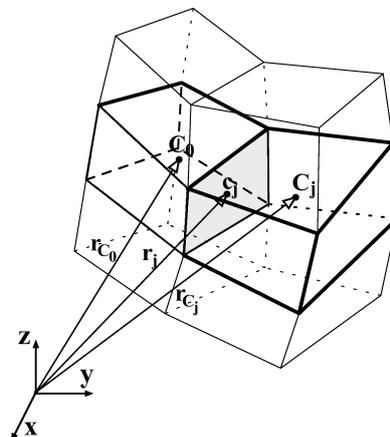


Distribuzione spaziale delle variabili - 2

Formulazione simmetrica - CDS

$$\phi_j = \frac{1}{2} (\phi_{C_0} + \phi_{C_j}) + \frac{1}{2} \left[(\nabla\phi)_{C_0} \cdot (\mathbf{r}_j - \mathbf{r}_{C_0}) + (\nabla\phi)_{C_j} \cdot (\mathbf{r}_j - \mathbf{r}_{C_j}) \right]$$

dove il pedice j indica la faccia compresa fra i VC C_0 e C_j , adiacenti a tale faccia.



Integrali di superficie e di volume

- L'equazione di trasporto dev'essere integrata su ogni VC.
- Adozione di schemi del second'ordine, adeguati per le applicazioni industriali della CFD.
- Utilizzo della formula del punto medio.

Integrazione

$$\int_V \phi \, dV \approx \phi_{C_0} \Delta V$$
$$\int_{A_j} \phi \, d\mathbf{A} \approx \phi_j \mathbf{A}_j$$
$$\int_{A_j} \nabla \phi \cdot d\mathbf{A} \approx (\nabla \phi)_j \cdot \mathbf{A}_j$$



Calcolo del gradiente

- Per poter calcolare i valori della variabile a centro-faccia, e per valutare termine diffusivo, è necessario conoscere il gradiente di ϕ nel centroide della faccia.
- Anche il calcolo del termine convettivo necessita della conoscenza del gradiente.
- Vi sono diverse strategie, adottate nei programmi commerciali, per ottenere (*ricostruire*) tale grandezza sulla base dei soli valori della variabile sui centroidi.
- Quelle indicati nel seguito rappresentano due delle alternative più diffuse.



Metodo di Gauss-Green

- ϕ_{x_0} , ϕ_{y_0} e ϕ_{z_0} sono le componenti del gradiente di ϕ in C_0 , e ϕ_{x_0} può essere considerato la divergenza del vettore $\phi \mathbf{i}$.
- Applicando il teorema di Gauss-Green:

$$\phi_{x_0} \Delta V \approx \int_V \phi_x dV = \int_V \nabla \cdot (\phi \mathbf{i}) dV = \int_A \phi \mathbf{i} \cdot d\mathbf{A} \approx \sum_j \phi_j A_{j,x}$$

- Procedendo in maniera analoga per le altre componenti del gradiente si ottiene:

$$\begin{aligned} \phi_{x_0} \mathbf{i} + \phi_{y_0} \mathbf{j} + \phi_{z_0} \mathbf{k} \\ \approx \frac{\sum_j \phi_j A_{j,x}}{\Delta V} \mathbf{i} + \frac{\sum_j \phi_j A_{j,y}}{\Delta V} \mathbf{j} + \frac{\sum_j \phi_j A_{j,z}}{\Delta V} \mathbf{k} \end{aligned}$$

Ricostruzione del gradiente con il metodo di Gauss-Green

$$(\nabla \phi)_{C_0} \approx \frac{\sum_j \phi_j \mathbf{A}_j}{\Delta V}$$

Rimane da stabilire come valutare il valore della variabile sulla faccia, ϕ_j .



Metodo di Gauss-Green *cell based*

Media aritmetica:

$$\phi_j = \frac{1}{2} (\phi_{C_0} + \phi_{C_j})$$

oppure media pesata (weighted interpolation):

$$\phi_j = \lambda \phi_{C_0} + (1 - \lambda) \phi_{C_j}$$

con

$$\lambda = \frac{|\mathbf{r}_{C_j} - \mathbf{r}_j|}{|\mathbf{r}_{C_j} - \mathbf{r}_{C_0}|}$$

Questo approccio è semplice ed economico da implementare, ma da luogo ad un errore del second'ordine solo se i vettori $(\mathbf{r}_{C_j} - \mathbf{r}_{C_0})$ e \mathbf{A}_j sono allineati.

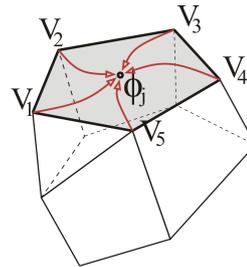
Altrimenti è possibile utilizzare una procedura di correzione di tipo iterativo.



Metodo di Gauss-Green *node based*

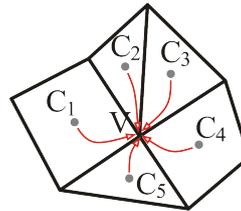
Media sui $N_{V,j}$ nodi (vertici) che definiscono la faccia j -esima:

$$\phi_j = \frac{1}{N_{V,j}} \sum_{l=1}^{N_{V,j}} \phi_l$$



Valori della variabile sui nodi (vertici) tramite *media pesata* del valore di ϕ nei centroidi delle celle che definiscono il nodo. Nella formulazione più semplice il peso è assunto pari all'inverso della distanza fra il nodo (vertice) ed il centroide della cella:

$$\phi_V = \frac{\sum_{j=1}^{N_C} \frac{\phi_{C_j}}{|\mathbf{r}_V - \mathbf{r}_{C_j}|}}{\sum_{j=1}^{N_C} \frac{1}{|\mathbf{r}_V - \mathbf{r}_{C_j}|}}$$

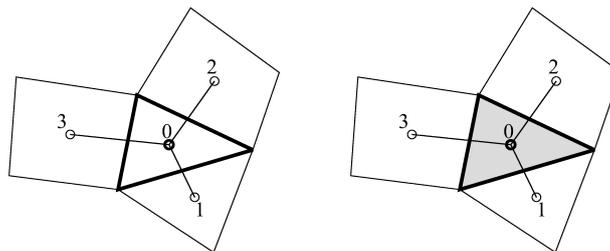


dove N_C è il numero delle celle che definiscono il nodo.



Metodo di Gauss-Green *cell based* - esempio - 1

La relazione trovata fornisce un valore del gradiente, in C_0 , che *non dipende* da ϕ_{C_0} .



Esempio 2D

Si ottiene

$$\phi_{x_0} \approx \frac{1}{\Delta V} \left[\frac{(\phi_{C_0} + \phi_{C_1})}{2} A_{1,x} + \frac{(\phi_{C_0} + \phi_{C_2})}{2} A_{2,x} + \frac{(\phi_{C_0} + \phi_{C_3})}{2} A_{3,x} \right]$$

$$\phi_{y_0} \approx \frac{1}{\Delta V} \left[\frac{(\phi_{C_0} + \phi_{C_1})}{2} A_{1,y} + \frac{(\phi_{C_0} + \phi_{C_2})}{2} A_{2,y} + \frac{(\phi_{C_0} + \phi_{C_3})}{2} A_{3,y} \right]$$



Esempio 2D (continua)

Semplificando le espressioni trovate:

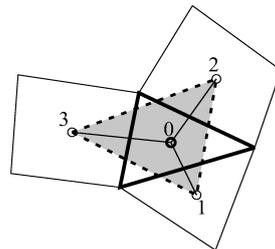
$$\phi_{x_0} \approx \frac{1}{2\Delta V} [\phi_{C_1} A_{1,x} + \phi_{C_2} A_{2,x} + \phi_{C_3} A_{3,x}]$$

$$\phi_{y_0} \approx \frac{1}{2\Delta V} [\phi_{C_1} A_{1,y} + \phi_{C_2} A_{2,y} + \phi_{C_3} A_{3,y}]$$



Metodo di Gauss-Green *cell based* - esempio - 3

Adozione di un dominio d'integrazione *esteso*: riduce gli errori associati alla valutazione dei valori della variabile sulle facce tramite interpolazione.



Esempio 2D - Dominio d'integrazione esteso

$$\phi_{x_0} \approx \frac{1}{2\Delta V} [\phi_{C_1} (y_{C_2} - y_{C_3}) + \phi_{C_2} (y_{C_3} - y_{C_1}) + \phi_{C_3} (y_{C_1} - y_{C_2})]$$

$$\phi_{y_0} \approx \frac{1}{2\Delta V} [\phi_{C_1} (x_{C_3} - x_{C_2}) + \phi_{C_2} (x_{C_1} - x_{C_3}) + \phi_{C_3} (x_{C_2} - x_{C_1})]$$

Anche in questo caso, però, le componenti del gradiente in C_0 non dipendono da ϕ_{C_0} .



- Il metodo di Gauss-Green *cell based*, per la valutazione dei gradienti, è accurato solo per griglie Cartesiane, o comunque strutturate *ortogonali*. Esso è piuttosto impreciso per griglie non-uniformi e griglie non strutturate costituite da celle di tipo diverso, quale è il caso delle *griglie ibride*.
- Il metodo di Gauss-Green *node based* - in particolare adottando la media pesata proposta da D. G. Holmes and S. D. Connell, 1989 - è più accurato del metodo *cell based* su griglie irregolari, tuttavia richiede un maggiore onere computazionale.
- Anche il metodo *ai minimi quadrati*, descritto nel seguito, è in grado di garantire un'accuratezza migliore di quella del metodo di Gauss-Green *cell based*, in particolare per griglie distorte ed irregolari, ma anch'esso richiede un onere computazionale leggermente maggiore, comparabile o inferiore a quello del metodo *node based*.



Metodo ai minimi quadrati - 1

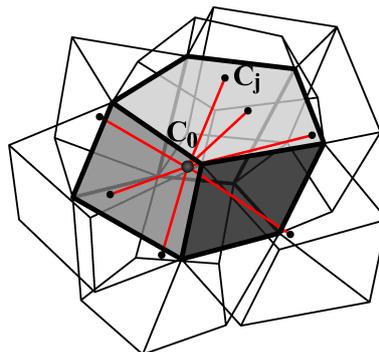
La relazione vista (non simmetrica) che fornisce la legge di variazione lineare della generica variabile ϕ all'interno della cella, può essere scritta in forma esplicita:

$$\phi = \phi_{C_0} + \phi_{x_0} (x - x_{C_0}) + \phi_{y_0} (y - y_{C_0}) + \phi_{z_0} (z - z_{C_0})$$

Imponendo che questa relazione sia valida non solo sulla cella in esame, ma descriva l'andamento spaziale della variabile sino ai centroidi delle celle adiacenti, si ha:

$$\phi_{C_j} = \phi_{C_0} + \phi_{x_0} (x_{C_j} - x_{C_0}) + \phi_{y_0} (y_{C_j} - y_{C_0}) + \phi_{z_0} (z_{C_j} - z_{C_0}) - r_j$$

dove r_j è il residuo: sistema *sovradeterminato*.



Metodo ai minimi quadrati - 2

In forma compatta:

$$r_j = \phi_{x_0} d_{xj} + \phi_{y_0} d_{yj} + \phi_{z_0} d_{zj} - \Delta_j$$

dove:

$$\begin{aligned}\Delta_j &= \phi_{C_j} - \phi_{C_0} \\ d_{xj} &= x_{C_j} - x_{C_0} \\ d_{yj} &= y_{C_j} - y_{C_0} \\ d_{zj} &= z_{C_j} - z_{C_0}\end{aligned}$$

Il quadrato del residuo si traduce quindi nella:

$$\begin{aligned}r_j^2 &= \Delta_j^2 + (\phi_{x_0} d_{xj})^2 + (\phi_{y_0} d_{yj})^2 + (\phi_{z_0} d_{zj})^2 \\ &\quad - 2(\Delta_j \phi_{x_0} d_{xj} + \Delta_j \phi_{y_0} d_{yj} + \Delta_j \phi_{z_0} d_{zj}) \\ &\quad + 2(\phi_{x_0} \phi_{y_0} d_{xj} d_{yj} + \phi_{x_0} \phi_{z_0} d_{xj} d_{zj} + \phi_{y_0} \phi_{z_0} d_{yj} d_{zj})\end{aligned}$$

che sommato per tutte le nb (*neighbours*) celle adiacenti fornisce:

$$R = \sum_{j=1}^{nb} r_j^2$$



Metodo ai minimi quadrati - 3

Minimizzando ai minimi quadrati:

$$\begin{aligned}\frac{\partial R}{\partial \phi_{x_0}} &= 0 \\ \frac{\partial R}{\partial \phi_{y_0}} &= 0 \\ \frac{\partial R}{\partial \phi_{z_0}} &= 0\end{aligned}$$

Sviluppando, raccogliendo ed esprimendo in forma matriciale:

$$\begin{bmatrix} \sum_{j=1}^{nb} (d_{xj})^2 & \sum_{j=1}^{nb} d_{xj} d_{yj} & \sum_{j=1}^{nb} d_{xj} d_{zj} \\ \sum_{j=1}^{nb} d_{xj} d_{yj} & \sum_{j=1}^{nb} (d_{yj})^2 & \sum_{j=1}^{nb} d_{yj} d_{zj} \\ \sum_{j=1}^{nb} d_{xj} d_{zj} & \sum_{j=1}^{nb} d_{yj} d_{zj} & \sum_{j=1}^{nb} (d_{zj})^2 \end{bmatrix} \begin{Bmatrix} \phi_{x_0} \\ \phi_{y_0} \\ \phi_{z_0} \end{Bmatrix} = \begin{Bmatrix} \sum_{j=1}^{nb} \Delta_j d_{xj} \\ \sum_{j=1}^{nb} \Delta_j d_{yj} \\ \sum_{j=1}^{nb} \Delta_j d_{zj} \end{Bmatrix}$$



Metodo ai minimi quadrati - 4

Con simbologia compatta

$$\mathbf{D}_0 \nabla \phi_{C_0} = \mathbf{f}_0$$

Ricostruzione del gradiente con il metodo ai minimi quadrati

$$\nabla \phi_{C_0} = \mathbf{D}_0^{-1} \mathbf{f}_0$$

- \mathbf{D}_0 è una matrice simmetrica 3×3 , i cui elementi dipendono solo dalla griglia. Pertanto è necessario calcolare la matrice inversa \mathbf{D}_0^{-1} , con griglia indeformabile, solo all'inizio del calcolo.
- La ricostruzione del gradiente col metodo dei minimi quadrati richiede quindi di memorizzare, ed invertire, la matrice \mathbf{D}_0 per ogni cella, per un totale di $6 \times N$ numeri reali, con N numero totale di celle.



Metodo ai minimi quadrati - commenti

- Per griglie molto distorte la matrice \mathbf{D}_0 può risultare mal condizionata:
 - ▶ Adozione di una metodologia di soluzione basata sulla fattorizzazione QR (fattorizzazione della matrice di partenza nel prodotto di una matrice ortogonale Q e una matrice triangolare superiore R tramite il processo di Gram-Schmidt).
 - ▶ Introduzione di *pesi* inversamente proporzionali alla distanza delle celle vicine.
- Il metodo è caratterizzato da un'accuratezza del primo ordine per griglie costituite da celle di forma arbitraria, ed è inoltre *consistente*, cioè il gradiente di una funzione lineare è calcolato *esattamente* per qualunque tipo di cella. Esso è perciò particolarmente adatto a trattare *griglie ibride*.



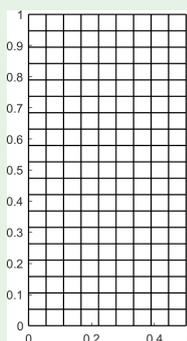
In the following, we will see how well the previous schemes can compute (*reconstruct*) the gradient on several types of grids. In particular:

- The domain selected is a single 2D rectangle, in order to provide the possibility to use also Cartesian and regular grids.
- The grids are characterized by the same number of cells, e.g. 200. This low number has been selected since it can more easily favor a graphical comparison between results.
- The comparison is performed in case of linear, quadratic, cubic and sinusoidal (trigonometric) functions.
- Since the value of the scalar variable ϕ is obtained by means of an analytical function, the latter is used to set its value at the *cell centroids*, at the *face centroids* and (when required) at the *nodes* (vertices).

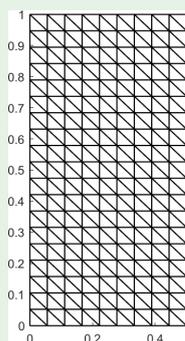


Gradient reconstruction - numerical tests - cont. 1

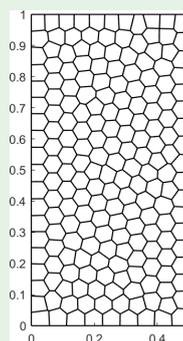
Grids and functions considered



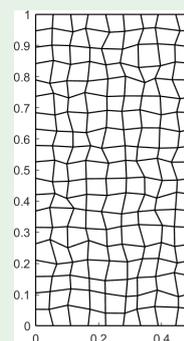
Cartesian



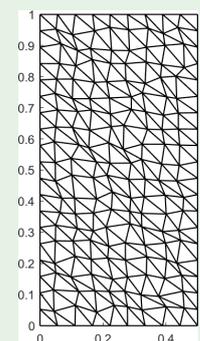
Triangular



Polygonal



Distorted Cartesian



Distorted
Triangular

- 1 Linear: $\phi = x + y$
- 2 Quadratic: $\phi = x^2 + y^2 + 2x + y + 1$
- 3 Cubic: $\phi = x^3 + 2y^3 + xy + 2$
- 4 Composite trigonometric function: $\phi = \sin(9x) + \cos(15y) + 5xy$



Max error of gradient reconstruction schemes for the various grids

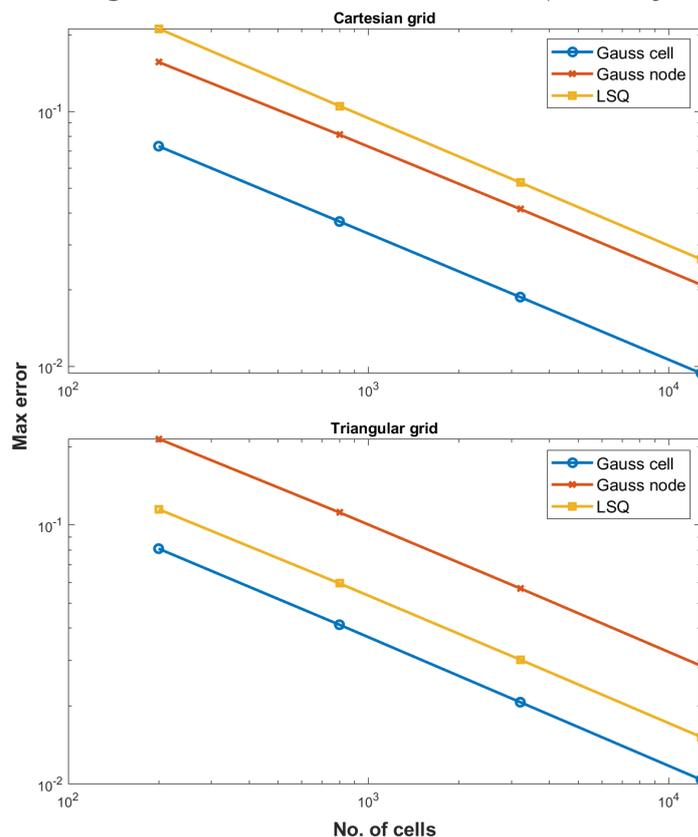
		Cartesian	Triangular	Polygonal	Distorted Cartesian	Distorted Triangular
Linear	<i>GCell</i>	5.44×10^{-15}	3.99×10^{-6}	0.093	0.270	0.375
	<i>GNode</i>	5.01×10^{-6}	4.44×10^{-6}	0.198	0.194	0.213
	<i>LSq</i>	2.44×10^{-15}	5.00×10^{-15}	1.67×10^{-15}	2.22×10^{-15}	4.22×10^{-15}
Quadratic	<i>GCell</i>	0.014	0.019	0.284	0.766	0.846
	<i>GNode</i>	0.043	0.070	0.443	0.423	0.525
	<i>LSq</i>	0.039	0.027	0.057	0.061	0.060
Cubic	<i>GCell</i>	0.073	0.081	0.532	0.755	0.962
	<i>GNode</i>	0.156	0.215	0.882	0.450	0.794
	<i>LSq</i>	0.211	0.115	0.245	0.249	0.165
Trigonometric	<i>GCell</i>	1.425	2.076	1.735	2.759	7.082
	<i>GNode</i>	2.886	4.617	3.208	3.533	5.450
	<i>LSq</i>	3.116	2.222	4.171	3.982	3.460

GCell = Gauss-Green Cell-based; *GNode* = Gauss-Green Node-based; *LSq* = Least Square



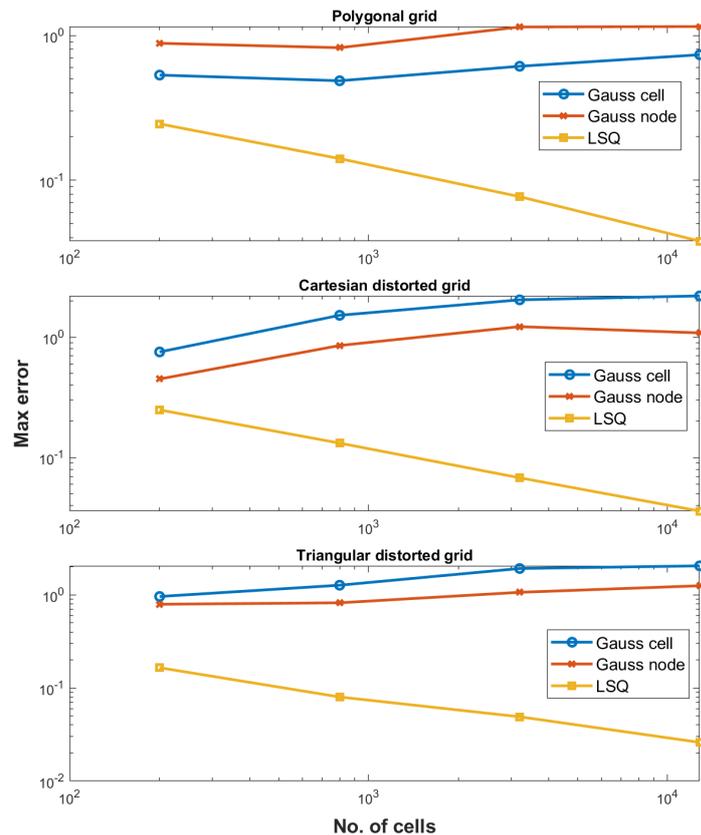
Gradient reconstruction - numerical tests - cont. 3

Convergence test for the cubic function $\phi = x^3 + 2y^3 + xy + 2$



Gradient reconstruction - numerical tests - cont. 4

Convergence test for the cubic function $\phi = x^3 + 2y^3 + xy + 2$



Termine transitorio

- La procedura, ed il risultato, dipende dallo schema di integrazione temporale adottato.
- Per problemi stazionari o con lente variazioni temporali, è sufficiente l'adozione dello schema di Eulero implicito del primo ordine.
- Per flussi con rapide variazioni temporali, risulta conveniente lo schema di Crank-Nicolson del second'ordine.

Integrazione del termine transitorio

$$\int_V \frac{\partial}{\partial \vartheta} (\rho \phi) dV \approx \frac{\partial}{\partial \vartheta} (\rho_{C_0} \phi_{C_0} \Delta V) \approx \rho_{C_0} \frac{\Delta V}{\Delta \vartheta} (\phi_{C_0}^{n+1} - \phi_{C_0}^n)$$



Integrazione del termine sorgente

$$\int_V s dV \approx s_{C_0} \Delta V$$

Qualora il termine sorgente dipenda dalla variabile stessa, si procede alla sua linearizzazione secondo quanto visto nel caso delle griglie Cartesiane.



Flusso convettivo - 1

Il termine convettivo viene calcolato considerando il contributo di tutte le facce costituenti il VC:

$$\int_A \rho \phi \mathbf{w} \cdot d\mathbf{A} = \sum_j \int_{A_j} \rho \phi \mathbf{w} \cdot d\mathbf{A}$$

e, sulla base della formula del punto medio:

Flusso convettivo

$$\sum_j \int_{A_j} \rho \phi \mathbf{w} \cdot d\mathbf{A} \approx \sum_j \dot{m}_j \phi_j$$

dove \dot{m}_j rappresenta la portata di massa attraverso la faccia A_j :

$$\dot{m}_j = \int_{A_j} \rho \mathbf{w} \cdot d\mathbf{A} \approx \rho_j (\mathbf{w}_j \cdot \mathbf{A}_j)$$

Linearizzazione del termine convettivo attraverso il metodo di Picard.



Flusso convettivo - 2

Valutazione di ϕ_j :

Schema CDS

$$\phi_j = \frac{1}{2} (\phi_{C_0} + \phi_{C_j}) + \frac{1}{2} \left[(\nabla\phi)_{C_0} \cdot (\mathbf{r}_j - \mathbf{r}_{C_0}) + (\nabla\phi)_{C_j} \cdot (\mathbf{r}_j - \mathbf{r}_{C_j}) \right]$$

- Instabilità ed oscillazioni (wiggles), e difficoltà di convergenza.
- Cura: raffinamento selettivo della griglia.

Schema UDS

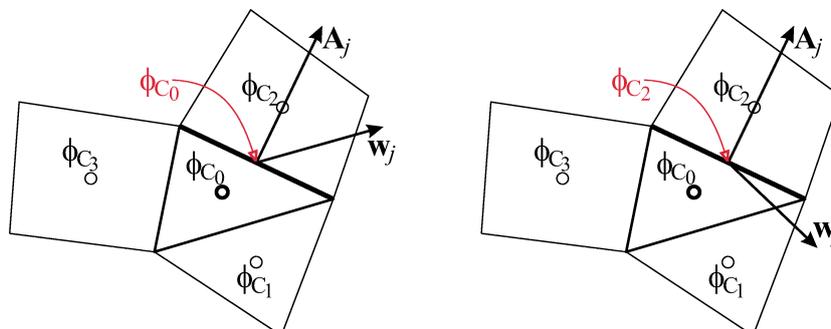
$$\phi_j = \begin{cases} \phi_{C_0} & \text{se } (\mathbf{w}_j \cdot \mathbf{A}_j) > 0 \\ \phi_{C_j} & \text{se } (\mathbf{w}_j \cdot \mathbf{A}_j) < 0 \end{cases}$$

- Introduzione di *diffusività artificiale*: usare, se proprio necessario, con cautela.
- Accettabile per equazioni di trasporto *source-dominated*, es. modelli di turbolenza.



Flusso convettivo - 3

Rappresentazione grafica dello schema UDS:



Schema *upwind* di ordine più elevato

$$\phi_j = \begin{cases} \phi_{C_0} + (\nabla\phi)_{C_0} \cdot (\mathbf{r}_j - \mathbf{r}_{C_0}) & \text{se } (\mathbf{w}_j \cdot \mathbf{A}_j) > 0 \\ \phi_{C_j} + (\nabla\phi)_{C_j} \cdot (\mathbf{r}_j - \mathbf{r}_{C_j}) & \text{se } (\mathbf{w}_j \cdot \mathbf{A}_j) < 0 \end{cases}$$

Ricorso a schemi misti, attraverso un fattore di *blending* γ compreso fra 0 e 1:

Schema misto

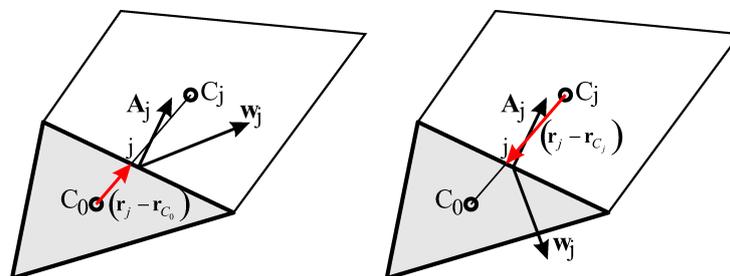
$$\phi_j = \gamma\phi_j^{oe} + (1 - \gamma)\phi_j^{ob} = \phi_j^{ob} + \left[\gamma \left(\phi_j^{oe} - \phi_j^{ob} \right) \right]$$

- $\gamma = 1$ corrisponde allo schema CDS, e $\gamma = 0$ corrisponde allo schema UDS.
- *ob* indica uno schema di ordine basso (UDS), e *oe* indica uno schema di ordine elevato (CDS).
- Utilizzo in modalità *correzione differita*.



Flusso convettivo - 5

Rappresentazione grafica dello schema *upwind* di ordine elevato:



Flusso diffusivo - 1

Il termine diffusivo viene calcolato considerando il contributo di tutte le facce costituenti il VC:

$$\int_A \Gamma \nabla \phi \cdot d\mathbf{A} = \sum_j \int_{A_j} \Gamma \nabla \phi \cdot d\mathbf{A}$$

e, sulla base della formula del punto medio::

$$\sum_j \int_{A_j} \Gamma \nabla \phi \cdot d\mathbf{A} \approx \sum_j \Gamma_j (\nabla \phi)_j \cdot \mathbf{A}_j$$

Il valore di $(\nabla \phi)_j$ viene approssimato tramite la:

$$(\nabla \phi)_j \approx \frac{1}{2} [(\nabla \phi)_{C_0} + (\nabla \phi)_{C_j}]$$

che sostituito nella precedente risulta:

$$\sum_j \int_{A_j} \Gamma \nabla \phi \cdot d\mathbf{A} \approx \frac{1}{2} \sum_j \Gamma_j [(\nabla \phi)_{C_0} + (\nabla \phi)_{C_j}] \cdot \mathbf{A}_j$$



Flusso diffusivo - 2

Raccogliendo, ed osservando che $\sum_j \Gamma_j \mathbf{A}_j \approx 0$, ed in particolare $\Gamma \sum_j \mathbf{A}_j = 0$ nel caso di proprietà termofisiche costanti, otteniamo:

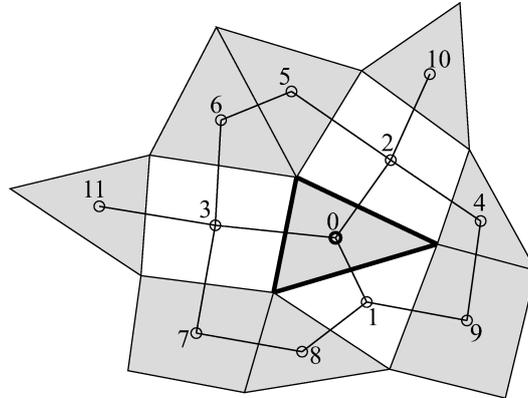
$$\int_A \Gamma \nabla \phi \cdot d\mathbf{A} \approx \frac{1}{2} \sum_j \Gamma_j (\nabla \phi)_{C_j} \cdot \mathbf{A}_j$$

L'espressione ora ricavata presenta due inconvenienti:

- 1 Abbiamo incluso VC *non adiacenti* (second neighbors) nella discretizzazione.
- 2 Sebbene i contributi dei VC *adiacenti* (first neighbors) siano formalmente presenti, essi si cancellano a vicenda: soluzioni con oscillazioni non fisiche (*wiggles* o *checkerboarding*).



Discretizzazione del termine diffusivo per la cella C_0 : solo i VC ombreggiati contribuiscono con coefficienti non nulli.



Flusso diffusivo - 4

Utilizzo di un'espressione *modificata* per il gradiente, che include l'effetto delle celle adiacenti nella discretizzazione, ad esempio:

$$\widetilde{(\nabla\phi)}_j = (\nabla\phi)_j + \left(\frac{\phi_{C_j} - \phi_{C_0}}{|\mathbf{d}_j|} \frac{\mathbf{A}_j}{|\mathbf{A}_j|} - \frac{(\nabla\phi)_j \cdot \mathbf{d}_j}{|\mathbf{d}_j|} \frac{\mathbf{A}_j}{|\mathbf{A}_j|} \right)$$

Con tali modifiche, l'espressione discreta del flusso diffusivo, scritta utilizzando la *correzione differita*, diventa:

Flusso diffusivo

$$\int_A \Gamma \nabla\phi \cdot d\mathbf{A} \approx \sum_j \Gamma_j \left\{ \frac{\phi_{C_j} - \phi_{C_0}}{|\mathbf{d}_j|} |\mathbf{A}_j| + \left[(\nabla\phi)_j - \frac{(\nabla\phi)_j \cdot \mathbf{d}_j}{|\mathbf{d}_j|} \frac{\mathbf{A}_j}{|\mathbf{A}_j|} \right] \cdot \mathbf{A}_j \right\}$$

Il primo termine a destra viene trattato in modo implicito, mentre l'espressione in parentesi quadra, nota con il nome di *termine cross-diffusivo*, è trattata in modo esplicito.



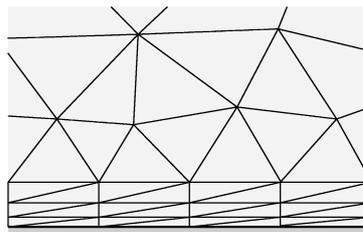
La procedura vista per la discretizzazione del termine diffusivo costituisce la base di alcuni codici commerciali, ma presenta alcuni difetti:

- L'accuratezza del metodo, e le proprietà di convergenza, si riducono per griglie lontane dall'ortogonalità: aumenta il contributo del termine *cross-diffusivo*.
- Per griglie molto distorte il termine cross-diffusivo può addirittura diventare negativo, con conseguenze *catastrofiche* per variabili definite positive, ad esempio l'energia cinetica turbolenta.
- Le griglie a *tetraedri* sono meno accurate, con la discretizzazione vista, rispetto a griglie costituite da soli *esaedri*. Le prime, tuttavia, sono spesso preferite per ragioni legate alla facilità di generazione automatica della griglia, ed eventuale utilizzo di griglie *adattive*.



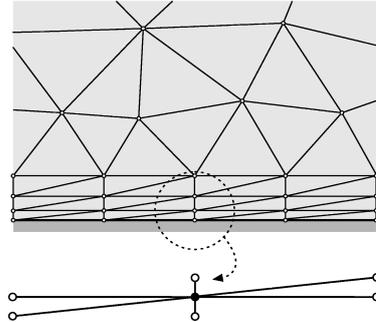
Griglie ibride - 1

- L'approccio FVM per griglie non strutturate è molto diffuso ed è adottato, pur con qualche variante, in numerosi pacchetti CFD.
- Esso presenta un inconveniente qualora venga utilizzato con griglie a tetraedri, in 3D, o a triangoli in 2D.
- Nel calcolo di flussi turbolenti con modelli RANS o LES, il centroide delle celle di calcolo disposte sulle pareti solide deve trovarsi ad una distanza y^+ , da queste ultime, compresa in un range opportuno.
- Ciò si traduce nella necessità di *schacciare* le prime celle in prossimità delle pareti:



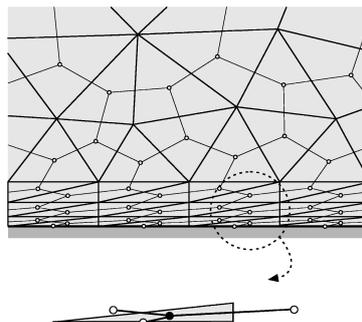
Griglie ibride - 2

- Con il metodo FEM non vi sarebbero problemi di stabilità poiché tutti i nodi, indicati nel particolare, darebbero un contributo alla riga della matrice corrispondente al nodo in esame, indicato con simbolo pieno.
- Vi sarebbero, eventualmente, problemi legati alla ridotta accuratezza di elementi molto *allungati*).



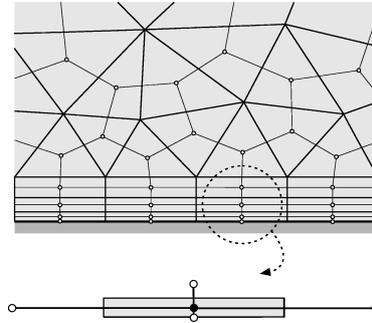
Griglie ibride - 3

- Con il metodo FVM descritto la molecola di calcolo risulta molto deformata: alcune rette congiungenti i centri delle celle formano un angolo rilevante con le normali alle facce e, nel caso limite di celle molto allungate, possono risultare quasi parallele alle facce stesse:
 - ▶ Aumenta il contributo del termine diffusivo esplicito, e si riduce il termine implicito: difficoltà di convergenza.
 - ▶ Aumenta il contributo dei termini cross-diffusivi che possono talvolta diventare negativi: ridotta accuratezza.
 - ▶ Per le equazioni di quantità di moto, diminuisce l'accuratezza nel calcolo del gradiente di pressione in prossimità delle pareti solide.



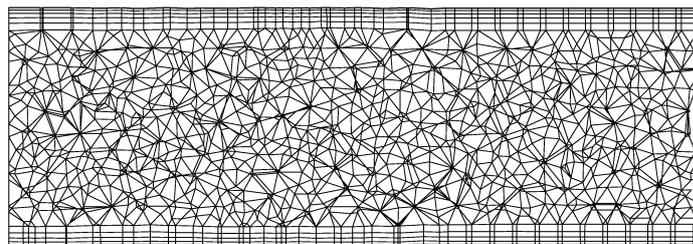
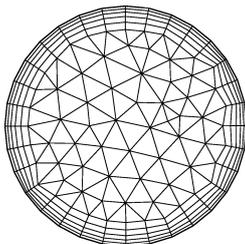
Griglie ibride - 4

- Soluzione: utilizzo di *griglie ibride*, cioè esaedri e/o prismi (rettangoli in 2D) per i primi strati di celle, e tetraedri (triangoli in 2D) al centro del dominio.
- La molecola di calcolo è molto più regolare, a causa dell'ortogonalità fra le normali alle facce e le congiungenti i centroidi.



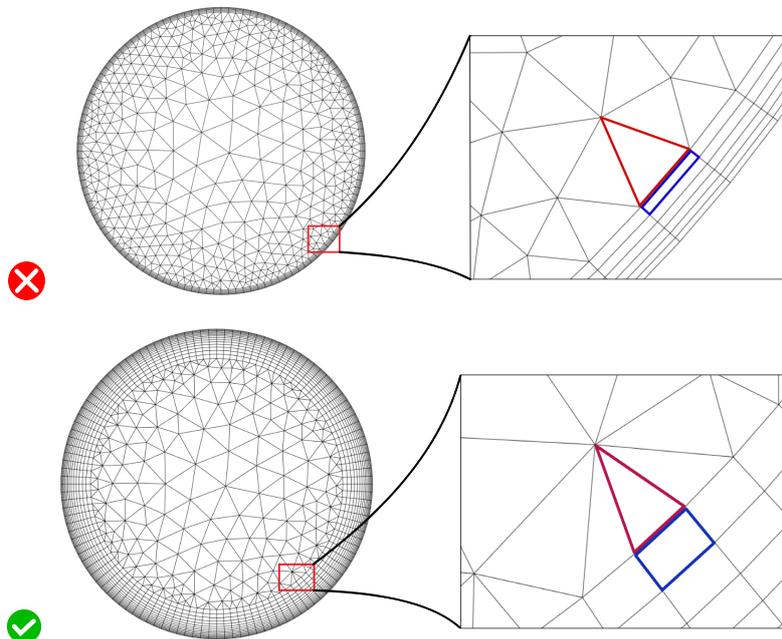
Griglie ibride - 5

- Opportunità (necessità) di utilizzare griglie ibride, anziché griglie a soli tetraedri, come confermato dalla gran parte dei prodotti commerciali, che ne prevedono la generazione automatica quale opzione.
- Adozione, qualora possibile, di griglie non strutturate o strutturate a soli esaedri: possibilità offerta da numerosi pacchetti CFD, e prodotti specifici per la generazione di griglie.



Griglie ibride - 6

- The volume of the cells of the final extruded layer should be similar to that of the first cells attached to them.
- If this does not happen, avoid to increase the growth-ratio above 1.2-1.3, but rather increase the number of layers.



Condizioni iniziali

- Per problemi non stazionari, è necessario fissare i valori delle variabili all'istante iniziale.
- Se interessa solo il *comportamento periodico* a regime (es. motori a combustione interna, organi rotanti di macchine ecc.), tali valori possono essere arbitrari, ma è necessario integrare per un tempo adeguato, in modo da eliminare gli effetti dovuti alle condizioni iniziali.
- Per innescare alcuni fenomeni, quali ad esempio separazioni non stazionarie degli strati limite, è conveniente perturbare la soluzione.
- Per problemi stazionari, l'utilizzo di condizioni iniziali il più possibile prossime alla soluzione cercata rende più economico il calcolo, e riduce il rischio di fallimento (divergenza).
- Tali condizioni iniziali possono ottenersi risolvendo, ad esempio:
 - ▶ Problemi semplificati (es. flussi e scambi termici puramente diffusivi)
 - ▶ Interpolazione/estrapolazione della soluzione di problemi più economici (es. griglia rada, problema bidimensionale).

- Le componenti Cartesiane della velocità non sono generalmente allineate con le facce delle celle sul contorno.
- Non si fa uso di *celle fantasma*, ed in generale gli integrali sulle facce dei VC sul contorno vanno espressi, come nel caso Cartesiano, in funzione delle condizioni al contorno e dei valori delle variabili nei centri dei medesimi VC:
 - ▶ Con condizioni al contorno di *Dirichlet*, il flusso convettivo è immediatamente disponibile, mentre il flusso diffusivo richiede di approssimare il gradiente sul contorno.
 - ▶ Con condizioni al contorno di *Neumann*, è immediatamente disponibile il valore del flusso diffusivo, mentre il valore della variabile, necessario nel caso di condizione al contorno *convettiva*, viene ricavato in base all'approssimazione (discreta) del gradiente.
- Le condizioni al contorno sulle pareti solide, nei problemi di flusso turbolento, dipendono dal modello di turbolenza adottato, e dalle modalità di trattamento della zona di parete.



Final discretization equation

- After assembling, for every CV, all terms of the transport equation integrated over the CV itself (unsteady term(s), diffusive and convective fluxes, source terms, boundary conditions et.), the final equation is obtained, here written in compact form

$$A_0 \phi_0 + \sum_{nb} A_{nb} \phi_{nb} = S_0$$

where, again, with nb we intend that the summation is extended to all neighboring CVs, whose number, differently from the case of structured grids, varies from one CV to the other.

- The coefficients in the equation depends on the approximations chosen for the various terms, e.g. convective and diffusive fluxes, unsteady term and source term.
- The resulting global coefficient matrix is *sparse*, and is usually solved by one of the available iterative methods.
- Introducing an under-relaxation coefficient α_ϕ within the iterative procedure, as already seen for Cartesian grids, the final discretization equation becomes

$$\underbrace{\frac{A_0}{\alpha_\phi}}_{A_0} \phi_0^k + \sum_{nb} A_{nb} \phi_{nb}^k = S_0 + \underbrace{\frac{(1 - \alpha_\phi) A_0}{\alpha_\phi}}_{S_0} \phi_0^{k-1}$$

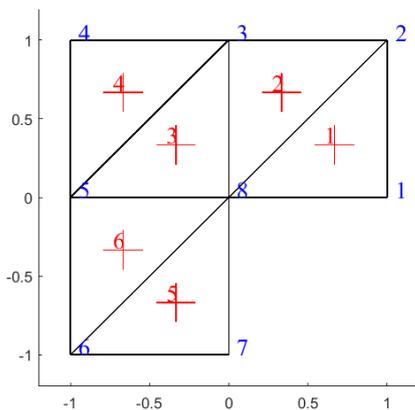


Mesh description

- The mesh is composed of FVs - cells - defined by a set of vertices and bounded by faces;
- Information about mesh topology:
 - ▶ List of vertices;
 - ▶ List of faces, defined in term of vertices;
 - ▶ List of cells, defined in term of faces (3D) or vertices (2D);
 - ▶ Element, face and vertex connectivity.
- The mesh boundary is divided into numbered patches of boundary faces: these patches are used to define the physical boundary conditions for the problem at hand.



Mesh: 2D example



List of nodes (vertices):

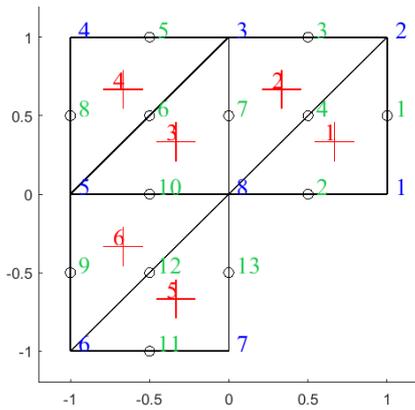
8		
1	1.0	0.0
2	1.0	1.0
3	0.0	1.0
4	-1.0	1.0
5	-1.0	0.0
6	-1.0	-1.0
7	0.0	-1.0
8	0.0	0.0

Element connectivity (list of elements):

6			
1	1	2	8
2	3	8	2
3	8	3	5
4	4	5	3
5	7	8	6
6	5	6	8



Mesh: 2D example - cont.



Vertex to element connectivity:

8					
1	0	0	0	0	0
1	2	0	0	0	0
2	3	4	0	0	0
4	0	0	0	0	0
3	4	6	0	0	0
5	6	0	0	0	0
5	0	0	0	0	0
1	2	3	5	6	6

Edge connectivity:

13				
1	1	2	1	-1
2	1	8	1	-1
3	2	3	2	-1
4	2	8	1	2
5	3	4	4	-1
6	3	5	3	4
7	3	8	2	3
8	4	5	4	-1
9	5	6	6	-1
10	5	8	3	6
11	6	7	5	-1
12	6	8	5	6
13	7	8	5	-1

Neighbor connectivity:

6						
1	-1	2	-1	1	4	2
2	3	1	-1	7	4	3
3	2	4	6	7	6	10
4	-1	3	-1	8	6	5
5	-1	6	-1	13	12	11
6	-1	5	3	9	12	10

