Computers & Fluids xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes

R. Abgrall [a,*], G. Baurin [b], P. Jacq [a], M. Ricchiuto [a]

[a] INRIA Bordeaux Sud Ouest and University of Bordeaux, Team Bacchus, 351 cours de la Libération, 33 405 Talence cedex, France
[b] SNECMA Villaroche, Rond Point René Ravaud, Réau, 77550 Moissy-Cramayel, France

ARTICLE INFO

Article history:
Received 2 February 2011
Received in revised form 14 April 2011
Accepted 15 May 2011
Available online xxxx

Keywords:
Residual distribution schemes
High order
Compressible fluid mechanics
Unstructured and hybrid meshes

ABSTRACT

Our aim is to report some recent advances in the development of residual distribution (RD) schemes using unstructured meshes: we present here some 3D results using pure tet meshes with a third order accurate scheme and 3D results using meshes with hex only. These latter meshes originate from ONERA where they have been used for Euler simulations with the Elsa code. Elsa only uses block structured meshes so that we have transformed the "ijk" format of the mesh into a nonstructured one without modifying the location of vertices and the connectivity of the mesh, so that it is exactly the same mesh.

1. Introduction

The RD scheme has been originally motivated by Roe in his fluctuation and signal framework. A one dimensional example of this framework is provided by his 1981 JCP paper [15] and a multidimensional extension is given in [8,16]. The idea is to design schemes where upwinding is a design principle. This is relatively easy in 1D, much less trivial in multiD since the signals are, for most waves, genuinely multidimensional, except for the steady supersonic case. This exact path has been followed for a some years by Roe and co-workers, in particular the group at VKI, and has led to a some very interesting non-oscillatory second order schemes for steady problems, and also a couple of questions. It has quite early been recognized that the RD schemes have many connections with the stabilized finite element methods see [10,9], and some formalizations of these methods have been done in [1,2]. It has also been possible to increase the accuracy from second to a priory any order, first on triangular meshes [6,5] and then more systematically on hybrid meshes in two dimensions [3]. Extensions to unsteady problems have been proposed in [4,14] and a very efficient one is proposed in [13]. The purpose of this paper is to show that these techniques extend without any problem to 3D, and that the schemes are easily parallelisable.

In a first part, we recall our construction of the high order RD schemes. We consider the equations of compressible fluid mechanics (perfect gas), on a domain $\Omega \subset \mathbb{R}^3$,

$$\begin{aligned} &\operatorname{div} \mathbf{f}(\mathbf{u}) = 0 \quad \text{if } x \in \Omega, \\ &\vec{u}\cdot\vec{n} = 0 \quad \text{on the solid walls} \\ &\mathbf{u} = \mathbf{u}_\infty \quad \text{on the inflow/outflow boundaries of } \partial\Omega \end{aligned} \tag{1a}$$

with

$$\mathbf{u} = (\rho, \rho\vec{u}, E)^T \text{ and } \mathbf{f} = (\rho\vec{u}, \rho\vec{u}\otimes\vec{u} + p\mathrm{Id}, (E+p)\vec{u})^T \tag{1b}$$

with $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho\vec{u}^2$. The parallel implementation of the scheme is also described. The second part is devoted to 3D numerical illustrations.

2. Schemes and parallelization issues

2.1. Description of the methods

We consider a conformal mesh $\mathcal{T}_h$ of Ω. The elements are denoted by K and the vertices of $\mathcal{T}_h$ by $\{M_i\}_{i=1,n_e}$. We assume here that $\Omega = \cup_{K\in\mathcal{T}_h} K$. The aim is to approximate the solution $\mathbf{u}$ of (1) by $\mathbf{u}^h \in V^h$ where, denoting by $\mathbb{P}^r(K)$ the set of polynomial of degree r defined on K,

$$V^h = \{V \text{ continuous function over } \Omega, \text{ such that for any } K \in \mathcal{T}_h, V_{|K} \in \mathbb{P}^r(K)^5\}.$$

* Corresponding author.
E-mail addresses: remi.abgrall@math.u-bordeaux1.fr, abgrall@math.u-bordeaux.fr (R. Abgrall).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.05.014

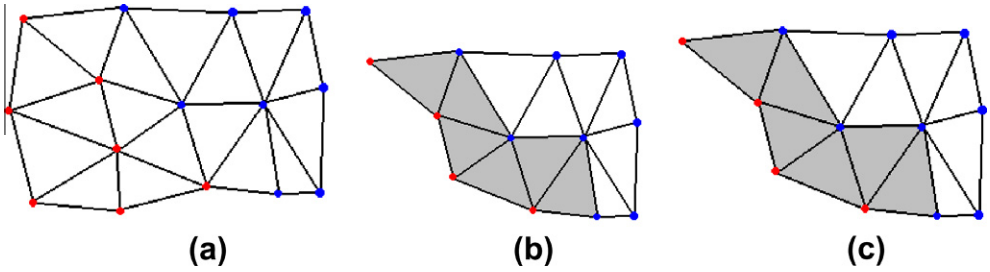

Fig. 1. 2 domains partitioning (P1 scheme): (a) Processor's 0 unknowns in red, Processor's 1 unknowns in blue, (b): Overlaps creation (P1 scheme): Processor's 0 calculated unknowns in red, unknowns received from Processor 1 in blue, Overlap's cells in gray, (c): Overlaps creation (P1 scheme): Processor's 1 calculated unknowns in blue, unknowns received from Processor 0 in red, Overlap's cells in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\mathbb{P}^r(K)$ is of dimension N_r and we introduce N_r Lagrange points σ on K. For the two cases studied in this paper, we have

- for tetrahedrons, the Lagrange points have the barycentric coordinates $\left(\frac{i}{r},\frac{j}{r},\frac{k}{r},\frac{l}{r}\right)$ with i,j,k integers such that $i+j+k+l=r$. The dimension of $\mathbb{P}^r(K)$ is $N_r=\frac{(r+1)(r+2)(r+3)}{3}$.
- for hexahedrons, we proceed as follows. The hex K is mapped onto $[0,1]^3$ (this can be done under some geometrical conditions, such as the convexity of K). The image of the σs by this mapping have the coordinates $\left(\frac{i}{r},\frac{j}{r},\frac{k}{r}\right)$ with i,j,k integers smaller than r. The dimension of $\mathbb{P}^r(K)$ is r^3.

Similar approximation spaces can be used for prisms. For other elements, such as pyramids, not considered here, the approximation space is different, see for example [7], but the same method can be adapted a priory. The solution of (1) can be equivalently represented by the approximations of $\mathbf{u}$ at the degrees of freedom, generically σ. We set $\mathbf{u}^h_\sigma := \mathbf{u}(\sigma)$. The purpose of the RD scheme is to evaluate $\mathbf{u}^h$ at these degrees of freedom. This is in contrast with classical finite volume schemes or Discontinuous Galerkin (DG) schemes where average values of $\mathbf{u}$, or quantities enabling to evaluate the average of $\mathbf{u}^h$ in the case of DG schemes, are evaluated.

A residual distribution scheme for (1) writes, for an internal degree of freedom σ, as

$$\text{for all } \sigma\in K,\qquad \sum_{K\ni\sigma}\Phi^K_\sigma(\mathbf{u}^h)=0. \tag{2}$$

The residual in (2) must, following [6], satisfy the following conservation constraints

$$\text{for any } K,\sum_{\sigma\in K}\Phi^K_\sigma(\mathbf{u}^h)=\int_{\partial K}\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}dl:=\Phi^K(\mathbf{u}^h) \tag{3}$$

where $\mathbf{f}^h(\mathbf{u}^h)$ is a high order accurate approximations of the flux $\mathbf{f}(\mathbf{u})$. Natural choices are either $\mathbf{f}^h(\mathbf{u}^h)$, the Lagrange interpolant of $\mathbf{f}(\mathbf{u})$ at the degrees of freedom defining $\mathbf{u}^h$, or the true flux evaluated for $\mathbf{u}^h$. We also assume that the residuals Φ^K_σ depend continuously of their arguments. Indeed, we impose a more severe constraint: Φ^K_σ only depends on the values of $\{\mathbf{u}_{\sigma'}\}_{\sigma'\in K}$.

If Γ is any edge/face of the inflow boundary of Ω, we consider a numerical flux $\mathcal{F}$ that depends on the boundary condition $\mathbf{u}_-$, the inward normal $\vec{n}$ and the local state $\mathbf{u}^h$. In the case of wall conditions, $\mathbf{u}_-$ is defined by mirroring. Then we assume that we have boundary residuals Φ^Γ_σ which satisfy the following conservation relation

$$\text{for any } \Gamma\subset\partial\Omega,\sum_{\sigma\in\Gamma}\Phi^\Gamma_\sigma=\int_{\partial\Gamma}\left(\mathcal{F}(\mathbf{u}^h,\mathbf{u}_-,\vec{n})-\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}\right)dl:=\Phi^\Gamma, \tag{4}$$

the residuals Φ^Γ_σ are assumed to be only dependent on the $\{\mathbf{u}_{\sigma'}\}_{\sigma'\in\Gamma}$, continuous, and the relation (3) has to be red, for any boundary node σ, as

$$\text{for all } \sigma\in\partial\Omega,\qquad \sum_{K\ni\sigma}\Phi^K_\sigma+\sum_{\Gamma\subset\partial\Omega^-,\Gamma\ni\sigma}\Phi^\Gamma_\sigma=0. \tag{5}$$

Following [6], it is easy to show that if the sequence $\mathbf{u}^h$ is bounded in L^∞ when $h\to 0$, and if there exists $\mathbf{v}$ such that $\mathbf{u}^h\to\mathbf{v}$ when $h\to 0$, then $\mathbf{v}$ is a weak solution of (1). Additional constraints can be set to fulfill entropy inequalities.

Which quadrature formula should we use in practice ? This question is related to the formal accuracy of the scheme. We first explore this for the scheme (2), (5) with the conservation relations (3) and (4), and then we move to the more practical one with approximated quadrature formula (2), (5). Following again [6], have the following result:

Proposition 2.1. *If the solution* $\mathbf{u}$ *is smooth enough and the residual, applied to the* $\mathbb{P}_k$ *interpolant of* $\mathbf{u}$*, satisfy*

$$\Phi^T_\sigma(\mathbf{u}^h)=\mathcal{O}(h^{r+d}) \tag{6a}$$

and

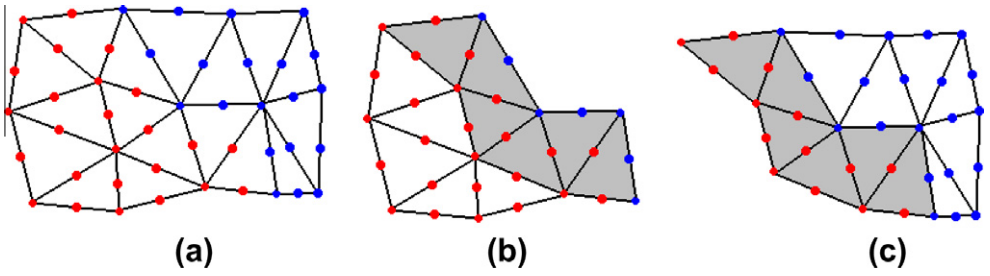

Fig. 2. 2 domains partitioning (P2 scheme), overlaps creation (P2 scheme): (a) Processor's 0 unknowns in red, Processor's 1 unknowns in blue, (b) Processor's 0 calculated unknowns in red, unknowns received from processor 1 in blue, overlap's cells in gray, (c) Processor's 1 calculated unknowns in blue, unknowns received from processor 0 in red, overlap's cells in gray. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014

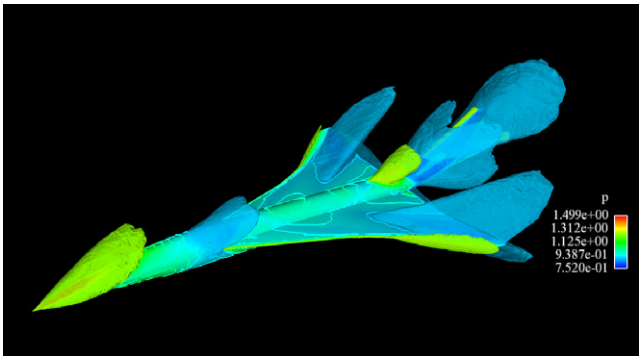

Fig. 3. Pressure isosurface and on the boundary.

$$\Phi_\sigma^\Gamma = \mathcal{O}(h^{r+d-1}), \tag{6b}$$

if moreover the approximation $\mathbf{f}^h(\mathbf{u}^h)$ *is* $r+1$*-order accurate, then the truncation error satisfies* $|\mathcal{E}(\mathbf{u}^h,\varphi^h)| \leqslant C(\varphi,\mathbf{f},\mathbf{u})h^{r+1}$. *The constant* $C(\varphi,\mathbf{u})$ *depends only on* φ *and* $\mathbf{u}$.

From the previous result, if there exists a constant (in the scalar case) or a matrix (in the system case) β_σ^T such that

$$\Phi_\sigma^K = \beta_\sigma^K\left(\int_{\partial K}\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}dl\right), \tag{7a}$$

$$\Phi_\sigma^\Gamma = \beta_\sigma^\Gamma\left(\int_{\partial K}\left(\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}-\mathcal{F}(\mathbf{u}^h,\mathbf{u}^-,\vec{n})dl\right)\right), \tag{7b}$$

then the condition (6) is fulfilled provided that β_σ^K is uniformly bounded. In practice, the conditions (7) are replaced (if $\mathbf{u}^h$ is any polynomial of degree r) by quadrature formulas ($\oint$ stands for a quadrature formula applied to $\int$)

$$\Phi_\sigma^K = \beta_\sigma^K\left(\oint_{\partial K}\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}dl\right), \tag{8a}$$

$$\Phi_\sigma^\Gamma = \beta_\sigma^\Gamma\left(\oint_{\partial K}\left(\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}-\mathcal{F}(\mathbf{u}^h,\mathbf{u}^-,\vec{n})dl\right)\right), \tag{8b}$$

It can be seen by using exactly the same arguments that the constraints on the quadrature formula are the following :

- In (8a), we must have

$$\int_{\partial K}\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}dl = \oint_{\partial K}\mathbf{f}^h(\mathbf{u}^h)\cdot\vec{n}dl + \mathcal{O}(h^{r+d}). \tag{9a}$$

- In (8b), we must have for the integrated quantity

$$\int_{\partial K} g(\mathbf{u}^h)dl = \oint_{\partial K} g(\mathbf{u}^h)dl + \mathcal{O}(h^{k+d-1}) \tag{9b}$$

In order to obtain these errors, there are two possible ways. Either the quadrature formula is exact on the approximated flux $\mathbf{f}^h(u^h)$ which basically means that $\mathbf{f}(u^h)$ is a polynomial of degree k at least since we need that $\mathbf{f}(u)-\mathbf{f}^h(u^h)=O(h^{k+1})$, or the quadrature formula is not exact but provides this error. In the paper, we have followed the first method : in each element, the flux is reconstructed by the Lagrange interpolation of the exact flux evaluated at the degrees of freedom in the element. This is similar to a quadrature free approach.

An example of such scheme is given by the Rusanov scheme rephrased in the RD context, namely

$$\Phi_\sigma^{K,Rus} = \frac{1}{N_K}\Phi^K + \alpha_K(\mathbf{u}_\sigma - \bar{\mathbf{u}}) \tag{10}$$

where N_K is the number of degree of freedom in K, $\bar{\mathbf{u}} = \frac{1}{N_K}\sum_{\sigma\in K}\mathbf{u}_\sigma$ and α_K is an upper bound to the maximum of the spectral radius of $\nabla_{\mathbf{u}}\mathbf{f}\cdot\vec{n}$ for unitary $\vec{n}$. This scheme is monotone but very dissipative.

2.1.1. Getting accuracy and monotonicity preservation

We briefly recall the method of Abgrall and Mezine [3]. All the schemes we are aware of write

$$\Phi_\sigma^K = \sum_{\sigma'\in K} c_{\sigma\sigma'}(\mathbf{u}_\sigma - \mathbf{u}_{\sigma'}) \tag{11}$$

It is well known that, in the scalar case, if all the $c_{\sigma\sigma'}$ are positive and if there is u^h that satisfies (5), then this solution satisfies a local maximum principle. It is also known that a scheme that is monotonicity preserving with constant coefficients $c_{\sigma\sigma'}$ cannot satisfy (6). This is Godunov's theorem. Thus the scheme must be nonlinear.

There is a systematic way of constructing schemes that are both monotonicity preserving and satisfy (6). We first look at the scalar case. We first start from a monotone (first order scheme) which residuals are $\Phi_\sigma^L = \sum_{\sigma'\in K} c_{\sigma\sigma'}^L(u_\sigma - u_{\sigma'})$. The coefficients $c_{\sigma\sigma'}$ are *all* positive. Since there is no ambiguity, we drop the superscript K. We assume that

$$\sum_{\sigma\in K}\Phi_\sigma^L = \Phi^K = \int_{\partial K} f^h(u^h)\cdot\vec{n}dl$$

where the integral is evaluated with the $\mathbb{P}^k(K)$ interpolant u^h. Then, $\{\Phi_\sigma^H\}$ denote high order residuals, they also satisfy the same conservation relation, and we let

$$\Phi_\sigma^H = \beta_\sigma\int_{\partial K} f^h(u^h)\cdot\vec{n}dl. \tag{12}$$

Clearly $\sum_\sigma\beta_\sigma = 1$ and this leads to introduce the parameters x_σ defined by $x_\sigma = \frac{\Phi_\sigma^H}{\Phi^K}$ for which, thanks to the conservation relation, we also have $\sum_\sigma x_\sigma = 1$. The next step is to write the formal identity $\Phi_\sigma^H = \frac{\Phi_\sigma^H}{\Phi_\sigma^L}\Phi_\sigma^L = \sum_{\sigma'}\frac{\Phi_\sigma^H}{\Phi_\sigma^L}c_{\sigma\sigma'}^L(u_\sigma - u_{\sigma'})$ and we get a monotonicity preserving constraint if for each $\sigma\in K$ we have $\frac{\Phi_\sigma^H}{\Phi_\sigma^L}\geqslant 0$ because then $c_{\sigma\sigma'}^H = \frac{\Phi_\sigma^H}{\Phi_\sigma^L}c_{\sigma\sigma'}^L \geqslant 0$.

All this can be rephrased in term of the x_σs and β_σs.

1. Conservation. We have $\sum_\sigma\beta_\sigma = 1$ and $\sum_\sigma x_\sigma = 1$.
2. Monotonicity preservation. For any $\sigma\in K$, $x_\sigma\beta_\sigma\geqslant 0$.

There are many solutions to that problem, one particularly simple one is an extension of the PSI "limiter" of Struijs [17], see [3] :

$$\beta_\ell = \frac{x_\ell^+}{\sum_{\ell'}x_{\ell'}^+}. \tag{13}$$

There is no singularity in the formula since $\sum_{\ell'}x_{\ell'}^+ = \sum_{\ell'}x_{\ell'} - \sum_{\ell'}x_{\ell'}^- \geqslant \sum_{\ell'}x_{\ell'} = 1$. Throughout the paper, we use (13).

In the case of the system (1), we need to adapt this procedure. We follow the method of [2]: for any K,

1. We evaluate $\Phi_\sigma^L := \Phi_\sigma^{R.Rus}$,
2. We estimate the average state $\bar{\mathbf{u}}$ defined by a density, velocity, speed of sound etc, where "average" means arithmetic average,
3. We choose a direction $\vec{N}$, typically the normalized averaged velocity,
4. We evaluate the left and right eigenvectors of $\nabla_{\mathbf{u}}\mathbf{f}(\bar{\mathbf{u}})\cdot\vec{N}$,
5. Project the Rusanov' residual (10) on the left eigenvectors. These are scalar quantities on which we apply (13)
6. The high order residual is then reconstructed from the "limited" scalar projection thanks to right eigenvectors. We denote this residual by $\Phi_\sigma^{K,\star} = \beta_\sigma^K\Phi^K$.

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014

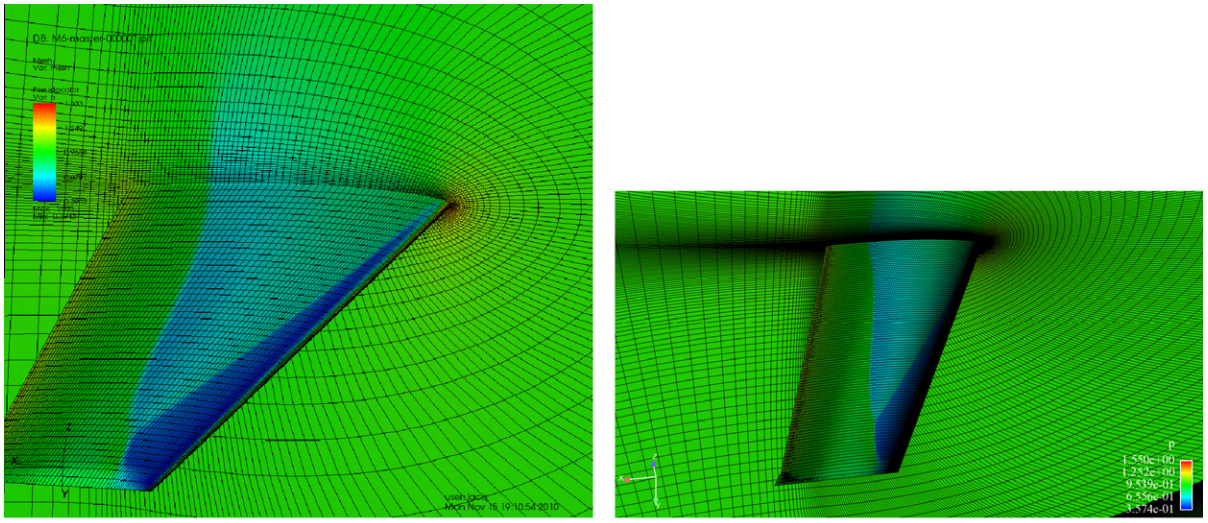

Fig. 4. Pressure on the skin.

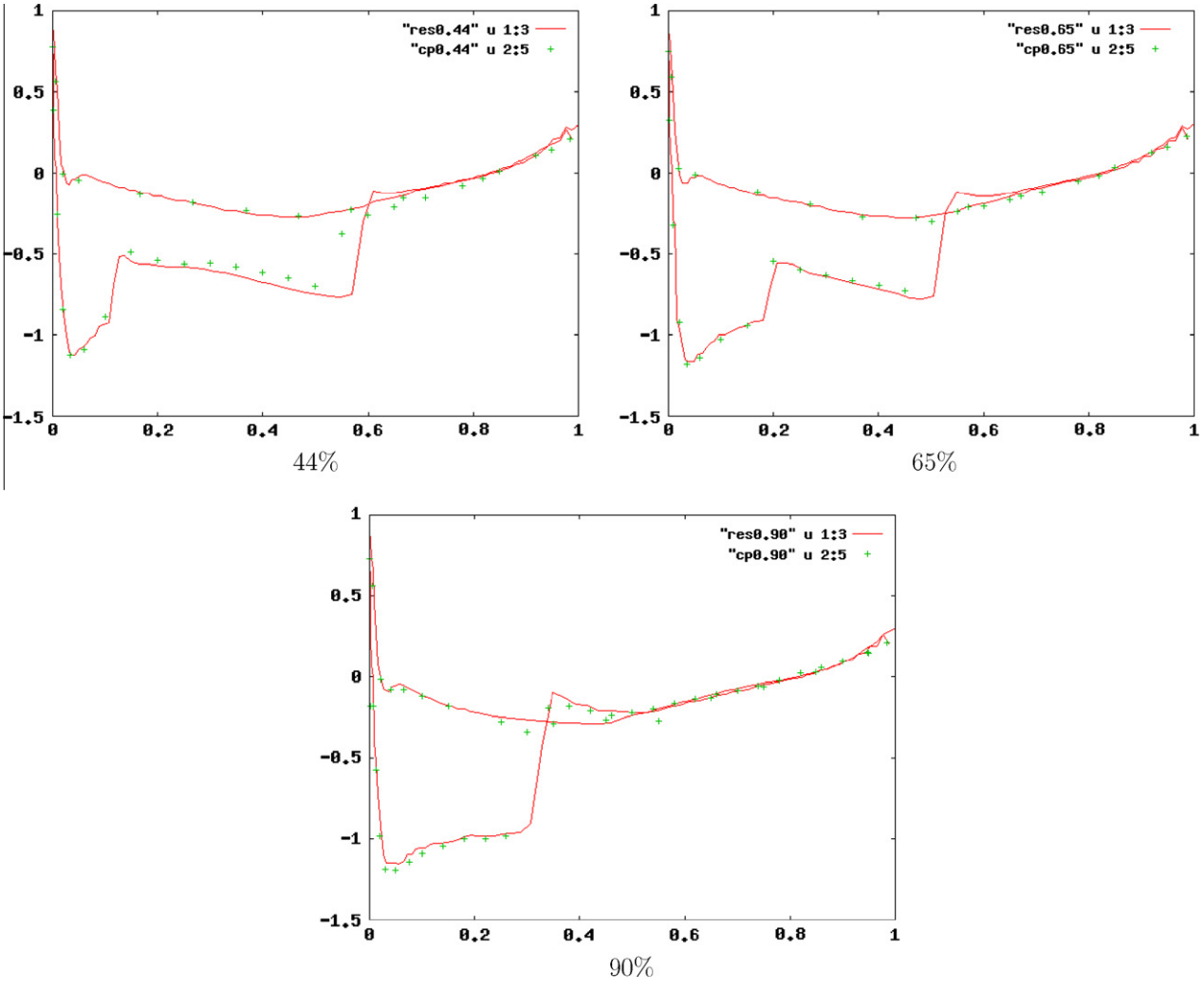

Fig. 5. Comparison of the c_p at 45%, 65% and 90% of the chord with experimental results.

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014

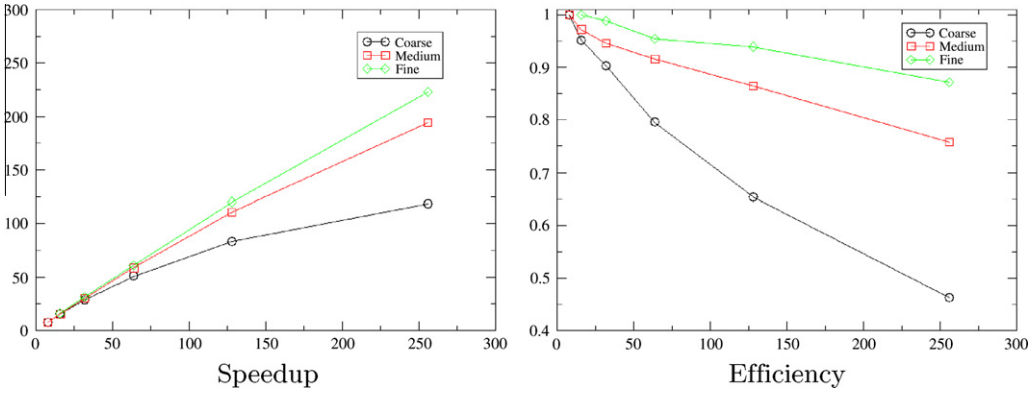

Fig. 6. Efficiency and speedup for the coarse, fine and medium meshes with 8, 16, 32, 64, 128 and 256 cores.

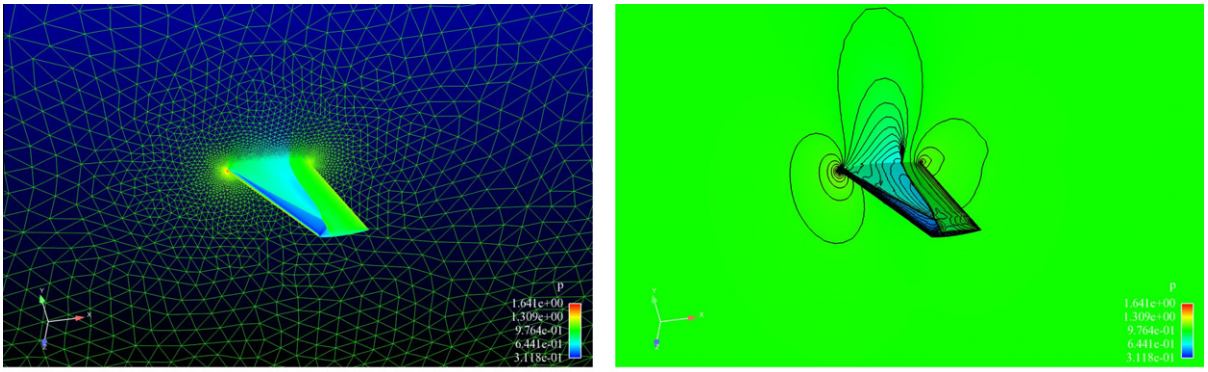

Fig. 7. Pressure and Mach isolines for the P2 scheme.

Unfortunately, this is not enough because the system of Eqs. (2) and (3) may have several solutions following a mechanism analyzed in [3]. A cure, also proposed in this reference, is to add to $\Phi_\sigma^{K,\star}$ the term $h_K \int_K \nabla_{\mathbf{u}}(\mathbf{f}(\mathbf{u}^h)\cdot\nabla\psi_\sigma)\tau(\mathbf{f}(\mathbf{u}^h)\cdot\nabla\mathbf{u}^h)$, so that the actual residual is

$$\Phi_\sigma^{K,\star\star} = \beta_\sigma^K \Phi^K + h_K \int_K \nabla_{\mathbf{u}}(\mathbf{f}(\mathbf{u}^h)\cdot\nabla\psi_\sigma)\tau(\mathbf{f}(\mathbf{u}^h)\cdot\nabla\mathbf{u}^h). \tag{14}$$

In [5] we discus ways of simplifying the integral term in (14) and we follow exactly the same procedure. Choices for τ are discussed in [2,5,3].

The implementation of boundary conditions is as follow. We use a very crude approximation, namely for $\sigma \in \Gamma \subset \partial\Omega$,

$$\Phi_\sigma^\Gamma = \omega_\sigma|\gamma|(\mathbf{f}^h(\mathbf{u}^h(\sigma))\cdot\vec{n} - \mathcal{F}(\mathbf{u}^h(\sigma),\mathbf{u}^-(\sigma),\vec{n})$$

where the weights are chosen such that $\omega > 0$ and the conservation hold. For second order accuracy, $\omega = \frac{1}{2}$, for third order accuracy, we have chosen in 2D $\omega = \frac{1}{6}, \frac{4}{6}$ and $\frac{1}{6}$ (Simpson's formula). In 3D, similar choices are done that ensure second order accuracy at the boundary (the weights are 1/12 at the vertices and 1/4 at the mid edges, remember that only tets are considered in this paper for 3D third order).

2.1.2. Implicitation of the scheme

For each degree of freedom, one has to solve (5) which constitutes a set of complex nonlinear equations. They are solved by an iterative scheme. that write

$$\mathbf{u}_\sigma^{n+1} = \mathbf{u}_\sigma^n - \omega_\sigma \sum_{K,\sigma\in K} \Phi_\sigma^K(\mathbf{u}^{n+1}).$$

An approximated linearization of the residuals $\Phi_\sigma^K(\mathbf{u}^{n+1})$ is done and we get

$$\begin{aligned}\delta\mathbf{u}_\sigma^n &= -\omega_\sigma \sum_{K,\sigma\in K} \Phi_\sigma^K(\mathbf{u}_h^{n+1}) \\ &\approx -\omega_\sigma \sum_{K,\sigma\in K} \Phi_\sigma^K(\mathbf{u}_h^n) - \omega_\sigma \sum_{K,\sigma\in K}\sum_{\sigma'\in K} \frac{\partial\Phi_\sigma^K}{\partial\mathbf{u}_{\sigma'}}\cdot\delta\mathbf{u}_{\sigma'}\end{aligned}$$

with $\delta\mathbf{u}_{\sigma'} := \mathbf{u}_{\sigma'}^{n+1} - \mathbf{u}_{\sigma'}^n$ for any σ'. The Jacobians of the residual are approximated by those of the Rusanov scheme, leading to

$$\begin{aligned}&\left(\frac{\mathrm{Id}}{\omega_\sigma} + \sum_{K,\sigma\in K} \frac{\partial\Phi_\sigma^{Rus,K}}{\partial\mathbf{u}_\sigma}\right)\delta\mathbf{u}_\sigma + \sum_{j\neq\sigma}\left(\sum_{K,\sigma\in K}\frac{\partial\Phi_\sigma^{Rus,K}}{\partial\mathbf{u}_j}\right)\delta U_j \\ &\quad = -\sum_{K,\sigma\in K}(\Phi_\sigma^K)^{\star\star}(u_h^n).\end{aligned} \tag{15}$$

The system (15) is solved by techniques such as Gauss-Seidel or GMRES with ILU(0) preconditioning.

3. Scheme parallelization

The main idea used in a parallel computation is "divide and conquer". If we have n processors then we will split our computational domain into n sub-domains by mesh partitioning. To be efficient the partitions have to correctly balance the

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014

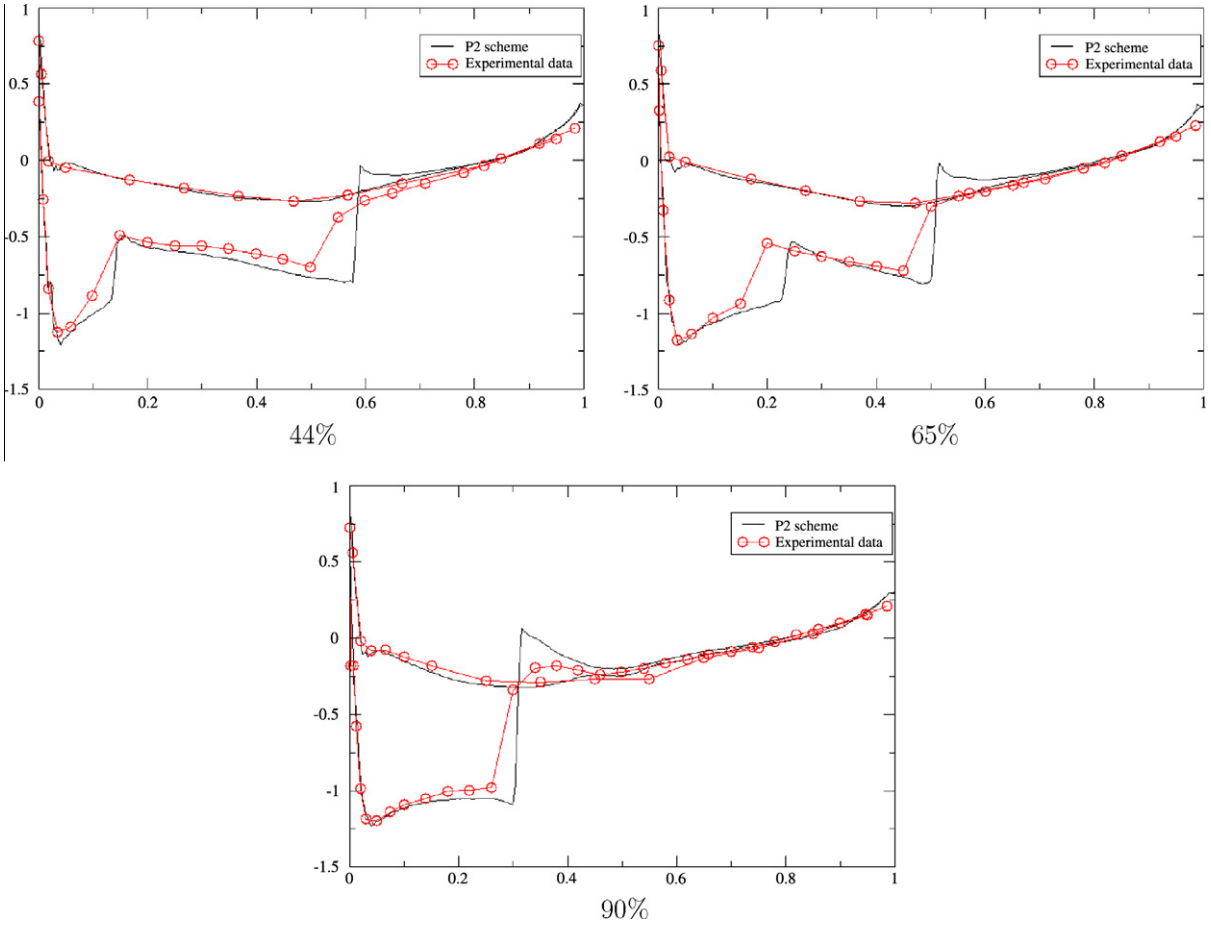

Fig. 8. Third order scheme on tet mesh: Comparison of the c_p at 44%, 65% and 90% of the chord with experimental results.

computational cost, i.e. each domain should contains the same number of unknowns.

After the partitioning step we know which processor will compute which unknown (see Fig. 1). Unfortunately, the dependency domain of some unknowns is not complete, the missing unknowns are calculated by an other processor. To solve this problem we have to build the "overlaps" by adding the missing unknowns to the domain. These unknowns are not evaluated by the current processor, they receive the missing information from the processor "owning" the unknowns (see Fig. 1a and b). There are two cases to consider: P1/Q1 approximation (second order scheme) and P2 approximation (third order). The Q2 case is still under implementation. In this case, there is one interior degree of freedom that complicates a bit the implementation of the communication scheme. It is essential, since the evaluation of the sub residuals associated to an element K is done using *only* values associated to that element, not to break this structure in the mesh partitioning.

3.1. Communication scheme for P1/Q1 scheme

Lets $\mathcal{T}_h$ be a triangulation, N be the nodes and $E \in (N,N)$ be the edges of the triangulation. Let us note P the partition, $P(i)$ with $i \in N$, contains the number of the processor computing the unknown i. The partitioning is done using the mesh partionner `scotch` [12,11]. In the second order scheme, the unknowns are located at the mesh nodes. The dependency domain D_{P1} is composed of the first neighbors, i.e. the nodes the unknown shares an edge with: $D_{P1}(i) = \{j \in N/(i,j) \in E\}$ for$i \in N$.The construction of the overlap is described by the Algorithm 1, see also Fig. 1.

Algorithm 1. Overlap creation algorithm

```
for e = (i,j) ∈ E do
  if P(i) ≠ P(j) then
    add i to the overlap of processor P(j)
    add j to the overlap of processor P(i)
  end if
end for
```

3.2. Communication scheme for P2 scheme

In the P2 case, the unknowns are located at the mesh nodes and in the middle of the edges. Let us denote $V(i)$ the neighboring elements of node i ($V(e)$ the neighboring elements of edge e). The dependency domain D_{P2} is defined as by $D_{P2}(i) = \{j \in N/\ (i,j) \in E\} \cup \{e \in E/e \in V(i)\}$for $i \in N$ and $D_{P2}(e) = \{i \in N/i \in V(e)\} \cup \{e' \in E/\ e' \in V(e)\}$for $e \in E$.

For the P2 scheme, we keep the P1/Q1 partition, the difference with the P1/Q1 communication scheme is in the overlap definition. At this point, the partition P is defined for the unknowns located at the nodes, we have to find a way to uniquely define the partition for the unknowns located at the edges. To do so we use Algorithm 2, see also Fig. 2.

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014

Algorithm 2. Modifying P2 partition

```
for e = (i,j) ∈ E do
  if P(i) ≠ P(j) then
    P(e)←min(P(i),P(j))
  else
    P(e)←P(i)
  end if
end for
```

Algorithm 3. P2 Overlap creation algorithm

```
for every element t ∈ T do
  for every edge e = (i,j) of t do
    if P(i) ≠ P(j) then
      add i to the overlap of processor P(j)
      add j to the overlap of processor P(i)
    end if
    for every node k of t do
      if P(e) ≠ P(k) then
        add e to the overlap of processor P(k)
        add k to the overlap of processor P(e)
      end if
    end for
  end for
end for
```

The construction of the P2 overlap is described by the Algorithm 3. This partition is not optimal: we have balanced the number of nodes of the mesh, which is not the number of unknowns. A more efficient way of proceeding could be to consider the graph of the third accurate scheme, to partition this graph while keeping the locality in the residual definition. This option, more complex, is available in `scotch`.

4. Numerical results

4.1. Supersonic flow

In that case we use the third order version of the scheme on a mesh using only tetrahedrons. The mesh has been provided by Alauzet (Gamma3 team, INRIA). We do not have the exact shape of the aircraft, hence the degrees of freedom, whatever the tetrahedron, are the mid-points of the edges: this means that we assume that the boundary has a polygonal shape. The inflow mach number is 2, with no incidence. The run is conducted with 128 domains. The mesh has 202,875 vertices, 1,148,326 tets i.e. 1,580,933 degrees of freedom. The pressure isosurface are displayed on Fig. 3. This example should mainly be seen as a test of the scheme robustness.

4.2. Transonic M6 wing

4.2.1. Using hybrid meshes

Here we use a series of structured (i,j,k) meshes provided by ONERA. The three meshes have: coarse grid $I = 103$, $J = 41$, $K = 22$, medium grid $I = 205$, $J = 81$, $K = 43$, fine grid $I = 409$, $J = 161$, $K = 85$. We use them with the following modifications. Instead of seeing the grid as list of points

$$(((\mathtt{x(i,j,k)}, \mathtt{y(i,j,k)}, \mathtt{z(i,j,k)}, \mathtt{i} = \mathtt{1}, \mathtt{Imax}), \mathtt{J} = \mathtt{1}, \mathtt{Jmax}), \mathtt{k} = \mathtt{1}, \mathtt{kmax})$$

we construct the hexahedrons which corners are the points of indices (i,j,k), $(i+1,j,k)$, $(i,j+1,k)$, $(i,j,k+1)$, $(i,j+1,k+1)$, $(i+1,j,k+1)$, $(i+1,j+1,k)$, $(i+1,j+1,k+1)$ where $1 \leqslant i, j, k \leqslant I_{max},J_{max},K_{max}$, the coordinates of the vertices are of course identical. This means that the connectivity of the mesh is identical as the ones provided by ONERA, as well as the physical location of the vertices, see Fig. 4. The test case is the standard M6 wing test case (Mach number of 0.84, incidence $\theta = 3.06^\circ$. The simulation is done with 64 domains (coarse grid), then 128 domains (medium grid) and 256 (fine grid), with an average of 20,000 points per domain in each case. The experimental results shown on Fig. 5 can be found on http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html. They correspond to a Mach number of 0.8395, an angle of attack of 3.06° and a Reynolds number of $11.72\ 10^6$ based on a mean aerodynamic chord of 0.64607 m. Note however that the numerical results are obtained from the Euler equations.

The Fig. 6 shows the speed up and efficiency results obtained for 500 iterations with Jacobi iterations in the implicit phase. The cluster is composed of 10 nodes. Each nodes has a 2 Quad-core Nehalem Intel Xeon X5570, with a 2.93 GHz CPU, 8 Mo of Cache L3 and 24 Go of RAM. The network is a 40 Gb/s one. It was not possible to run the code on 1 node for the fine mesh, but the code is far from being optimized (both in CPU and memory efficiency). This why there is no result for 8 cores/fine mesh. The results shown on Fig. 6 is quite good.

4.2.2. Third order results on a tet mesh

We redo the same test case. In each tetrahedron, we add the missing degrees of freedom for having a quadratic interpolation. Nothing special is done on the airfoil. The Fig. 7 presents the pressure on the wing and the Mach number isolines. It also shows the mesh. The Fig. 8 shows the pressure coefficient compared to the same experimental results. The results and the experimental data are in good agreement.

5. Conclusions

We have shown that it is possible to design effective second and higher order schemes with tets and hex elements. In particular structured block meshes can be used without any modification. Currently, we are extending these techniques in three directions : (i) Navier Stokes and turbulent flows, (ii) enlarge the type of elements that can be used with these techniques and (iii) increase the efficiency.

Acknowledgements

G. Baurin has been financed by SNECMA via a CIFRE PhD grant. R. Abgrall and P. Jacq has been financed by part by the ERC Senior Grant "ADDECCO", # 226316. R. Abgrall and M. Ricchiuto has been finance by part by the FP7-AAT-2010-RTD-1 STREP IDIHOM (#).

References

[1] Abgrall R. Toward the ultimate conservative scheme: following the quest. J Comput Phys 2001;167(2):277–315.
[2] Abgrall R. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. J Comput Phys 2006;214(2):773–808.
[3] Abgrall R, Larat A, Ricchiuto M. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. J Comput Phys 2011. doi:10.1016/j.jcp.2010.07.035.
[4] Abgrall R, Mezine M. Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. J Comput Phys 2003;188:16–55.
[5] Abgrall R, Ricchiuto M, Larat A, Tavé C. A simple construction of very high order nonoscillatory compact schemes on unstructured meshes. Comput Fluids 2009;38(7):1314–23.
[6] Abgrall R, Roe L. High-order fluctuation schemes on triangular meshes. J Sci Comput 2003;19(1–3):3–36.

[7] Bergot M, Cohen G, Duruflé M. Higher-order finite element for hybrid meshes using new nodal pyramidal elements. J Sci Comput 2010;42(3):345–81.
[8] Deconinck H, Roe PL, Struijs R. A multidimensional generalization of Roe's difference splitter for the Euler equations. Comput Fluids 1993;22(2/3):215–22.
[9] Hughes TJR, Mallet M. A new finite element formulation for CFD: III. The generalized streamline operator for multidimensional advective-diffusive systems. Comput Methods Appl Mech Eng 1986;58:305–28.
[10] Paillère H, Carette J-C, Deconinck H. Multidimentional upwind and supg methods for the solution of the compressible flow equations on unstructured grids, VKI-LS 1994-05. Comput Fluid Dyn 1994;1:1–10.
[11] François Pellegrini. PT-Scotch and libScotch 5.1 user's guide, 2008. <http://www.labri.fr/perso/pelegrin/scotch/>.
[12] François Pellegrini. Scotch and libScotch 5.1 user's guide, 2008. <http://www.labri.fr/perso/pelegrin/scotch/scotch_en.html>.
[13] Ricchiuto M, Abgrall R. Explicit Runge-kutta residual distribution schemes for time dependant problems: second order case. J Comput Phys 2010;229(16):5653–91.
[14] Ricchiuto M, Csík Á, Deconinck H. Residual distribution for general time dependent conservation laws. J Comput Phys 2005;209(1):249–89.
[15] Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. J Comput Phys 1983;43:357–72.
[16] Roe PL. Fluctuations and signals – a framework for numerical evolution problems. In: Numerical methods for fluid dynamics, Proceedings of the Conference, Reading/UK; 1982. p. 219–57.
[17] Struijs R. A Multi-dimensional upwind discretization method for the Euler Equations on unstructured grids. PhD thesis, University of Delft, Netherlands; 1994.

Please cite this article in press as: Abgrall R et al. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. Comput Fluids (2011), doi:10.1016/j.compfluid.2011.05.014