

# Mesh Update Techniques for Free-Surface Flow Solvers Using Spectral Element Method

Roland Bouffanais<sup>1</sup> and Michel O. Deville<sup>1</sup>

*Received September 10, 2004; accepted (in revised form) September 16, 2005; Published online January 6, 2006*

---

This paper presents a novel mesh-update technique for unsteady free-surface Newtonian flows using spectral element method and relying on the arbitrary Lagrangian–Eulerian kinematic description for moving the grid. Selected results showing compatibility of this mesh-update technique with spectral element method are given.

---

**KEY WORDS:** Spectral element; free-surface flows; ALE; moving grid; mesh update.

## 1. INTRODUCTION

Incompressible free-surface flows are encountered in a wide range of engineering and environmental flows. In the nineties the more specific case of turbulent free-surface flows started to be investigated with numerical computation based on high-order methods [8, 9]. In our work, we aim at computing large-eddy simulation (LES) of unsteady, incompressible and Newtonian turbulent free-surface flows by using the spectral element method (SEM) [13, 14]. The choice of interface-tracking technique was made to ensure an accurate description of the free surface.

This paper highlights the computational techniques we are developing for simulating incompressible free-surface flows using the SEM. These techniques include the arbitrary Lagrangian–Eulerian (ALE) formulation [2, 7, 15], mesh update and re-meshing methods [6, 10].

This paper is organized as follows. The governing equations in the ALE framework for general free-surface flows are introduced in Sec. 2.

---

<sup>1</sup> Laboratory of Computational Engineering, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. E-mails: {Roland.Bouffanais; Michel.Deville}@epfl.ch

Then, we present the discretization methods and numerical technique in Sec. 3. Sections 4 and 5 are dedicated to the moving-grid technique and the mesh-transfer operation, respectively.

## 2. GOVERNING EQUATIONS

A moving boundary-fitted grid technique has been chosen to simulate the free surface in our computations. This choice of a *surface-tracking* technique is primarily based on accuracy requirements. With this group of techniques, the grid is configured to conform to the shape of the interface, and thus adapts continually (at each time step) to it and therefore provides an accurate description of the free surface to express the related kinematic and dynamic boundary conditions.

The free-surface incompressible Newtonian flows that we have considered are governed by the Navier–Stokes equations comprising the momentum equation and the divergence-free condition. In the arbitrary Lagrangian–Eulerian (ALE) formulation, a mixed kinematic description is employed: Lagrangian description of the free surface  $\partial\Omega_F(t)$ , Eulerian description of the fixed domain boundaries  $\partial\Omega_D$  and mixed description of the internal fluid domain  $\Omega(t)$ , subset of  $\mathbb{R}^d$  with  $d = 2, 3$  the space dimension,  $t$  referring to the time as the fluid domain is changing when its boundaries are moving. Let us denote by  $\Omega_0$  a reference configuration (for instance the domain configuration at initial time  $t = t_0$ ). The system evolution is studied in the time interval  $I = [t_0, T]$ . The position of a point in the current fluid domain  $\Omega(t)$  is denoted by  $\mathbf{x}$  (Eulerian coordinate) and in the reference frame  $\Omega_0$  by  $\mathbf{Y}$  (ALE coordinate). Let  $\mathcal{A}_t$  be a family of mappings, which at each  $t \in I$  associates a point  $\mathbf{Y} \in \Omega_0$  to a point  $\mathbf{x} \in \Omega_t$ :

$$\mathcal{A}_t : \Omega_0 \subset \mathbb{R}^d \rightarrow \Omega_t \subset \mathbb{R}^d, \quad \mathbf{x}(\mathbf{Y}, t) = \mathcal{A}_t(\mathbf{Y}). \quad (1)$$

$\mathcal{A}_t$  is assumed to be continuous and invertible on  $\overline{\Omega_0}$  and differentiable almost everywhere in  $I$ . The inverse of the mapping  $\mathcal{A}_t$  is also continuous on  $\overline{\Omega_0}$ . With these notations the set of equation reads:

$$\left. \frac{\partial \mathbf{v}}{\partial t} \right|_{\mathbf{Y}} + (\mathbf{v} - \mathbf{w}) \cdot \nabla_{\mathbf{x}} \mathbf{v} = -\nabla_{\mathbf{x}} p + 2\nu \nabla_{\mathbf{x}} \cdot \mathbf{D}_{\mathbf{x}}(\mathbf{v}) + \mathbf{f} \quad \text{in } \Omega(t), \quad (2)$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{v} = 0 \quad \text{in } \Omega(t), \quad (3)$$

with  $\mathbf{v}(\mathbf{x}, t)$  the velocity field,  $p(\mathbf{x}, t)$  the pressure field (normalized by the constant density  $\rho$ ),  $\mathbf{D}_{\mathbf{x}}(\mathbf{v}) = \frac{1}{2}(\nabla_{\mathbf{x}} \mathbf{v} + \nabla_{\mathbf{x}} \mathbf{v}^T)$  the rate-of-deformation tensor,  $\nu$  the kinematic viscosity of the fluid and  $\mathbf{f}$  the body force. The ALE mesh

velocity  $\mathbf{w}(\mathbf{x}, t)$  appearing in (2) is defined as

$$\mathbf{w}(\mathbf{x}, t) = \frac{\partial \mathbf{x}}{\partial t} \Big|_{\mathbf{Y}} = \frac{\partial \mathcal{A}_t}{\partial t} \Big|_{\mathbf{Y}}. \quad (4)$$

Surface tension effects are assumed to be negligible as we deal with turbulent flows. The associated boundary conditions are:

- the kinematic boundary condition on  $\partial\Omega_F(t)$ :

$$\mathbf{v} \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n}, \quad (5)$$

$\mathbf{n}$  being the local outward unit normal to the free surface;

- the dynamic boundary condition on  $\partial\Omega_F(t)$ :

$$-p\mathbf{n} + 2\nu\mathbf{D}_x(\mathbf{v}) \cdot \mathbf{n} = \mathbf{0}, \quad (6)$$

assuming an inviscid air and zero ambient pressure;

- homogeneous Dirichlet boundary condition on  $\partial\Omega_D$ :

$$\mathbf{v} = \mathbf{w} = \mathbf{0}. \quad (7)$$

In addition to the set of governing equations (2)–(7), the closure of this free-surface problem based on a moving-grid formulation requires one more equation governing the evolution of the mesh velocity  $\mathbf{w}$  in the internal fluid domain  $\Omega(t)$ . The boundary values of  $\mathbf{w}$  being prescribed by the equations (5) and (7) on the boundary  $\partial\Omega_F(t) \cup \partial\Omega_D$  of the fluid domain. This last governing equation for  $\mathbf{w}$  will be presented in detail in Sec. 4.

As our focus is on transient problems, proper initial conditions at time  $t = t_0$  for the fluid velocity  $\mathbf{v}$  and for the mesh velocity  $\mathbf{w}$  have to be provided. The initial fluid velocity must satisfy the divergence-free condition and the values of the initial mesh velocity have to be given together with the initial shape of the free surface.

Based on the strong formulation of this free-surface problem given above, one can derive the more appropriate weak transient ALE formulation:

Find  $(\mathbf{v}(t), p(t)) \in H_{0,D}^1(\Omega(t))^d \times L^2(\Omega(t))$  such that for almost every  $t \geq t_0$

$$\begin{aligned} & \frac{d}{dt} \int_{\Omega(t)} (\hat{\mathbf{u}} \circ \mathcal{A}_t^{-1}) \cdot \mathbf{v} \, d\Omega + \int_{\Omega(t)} (\hat{\mathbf{u}} \circ \mathcal{A}_t^{-1}) \cdot \nabla_x [\mathbf{v}\mathbf{v} - \mathbf{v}\mathbf{w}] \, d\Omega \\ &= \int_{\Omega(t)} (p \nabla_x \cdot (\hat{\mathbf{u}} \circ \mathcal{A}_t^{-1}) - 2\nu \mathbf{D}_x((\hat{\mathbf{u}} \circ \mathcal{A}_t^{-1})) : \nabla_x \mathbf{v}) \, d\Omega \\ &+ \int_{\Omega(t)} \mathbf{f} \cdot (\hat{\mathbf{u}} \circ \mathcal{A}_t^{-1}) \, d\Omega \quad \forall \hat{\mathbf{u}} \in H_{0,D}^1(\Omega_0)^d, \end{aligned} \quad (8)$$

and

$$-\int_{\Omega(t)} q \nabla_{\mathbf{x}} \cdot \mathbf{v} d\Omega = 0 \quad \forall q \in L^2(\Omega(t)). \quad (9)$$

with the functional space  $H_{0,D}^1(\Omega(t))$  defined by

$$H_{0,D}^1(\Omega(t)) = \{v \in L^2(\Omega(t)), \quad \nabla_{\mathbf{x}} v \in L^2(\Omega(t))^d, \quad v|_{\partial\Omega_D} = 0\}.$$

It is worth noting that the weak formulation (8)–(9) is only valid in our particular case where homogeneous natural and essential boundary conditions, respectively (6) and (7) are applied to the system.

### 3. NUMERICAL TECHNIQUE AND DISCRETIZATION

A classical Galerkin approximation is applied to the set of governing equations in its weak transient ALE form (8)–(9) on the flow domain  $\Omega(t)$ , in order to determine the pressure and the fluid velocity, keeping in mind that the mesh velocity is obtained by the moving-grid technique developed in the next section. The Galerkin approximation is then discretized by using the spectral element method with the classical staggered  $\mathbb{P}_N - \mathbb{P}_{N-2}$  approach to avoid the development of spurious pressure modes. Discontinuous and continuous approximations are respectively taken for the pressure and fluid velocity. The mesh velocity is discretized using the same polynomial space as the fluid velocity, namely  $\mathbb{P}_N$ , based on a Gauss–Lobatto–Legendre (GLL) grid of order  $N$ . For the discontinuous approximation of the pressure, a Gauss–Legendre (GL) grid of order  $N - 2$  is used. Consequently the ALE Navier–Stokes semi-discrete equations can be derived from (8)–(9):

$$\frac{d}{dt}(\mathbf{M}\underline{\mathbf{v}}) + \mathbf{C}(\underline{\mathbf{v}}, \underline{\mathbf{w}})\underline{\mathbf{v}} = -\mathbf{K}\underline{\mathbf{v}} + \mathbf{D}^T \underline{p} + \underline{\mathbf{F}}, \quad (10)$$

$$-\mathbf{D}\underline{\mathbf{v}} = \mathbf{0}, \quad (11)$$

$\mathbf{M}$  denoting the mass matrix,  $\mathbf{K}$  the direct stiffness matrix,  $\mathbf{D}^T$  the discrete gradient operator,  $\mathbf{D}$  the discrete divergence operator,  $\mathbf{C}(\underline{\mathbf{v}}, \underline{\mathbf{w}})$  the discrete convective operator depending both on the fluid and mesh velocities and  $\underline{\mathbf{F}}$  the discrete body force. The update of the position  $\mathbf{x}$  of the mesh points is performed by integrating the following discrete equation:

$$\frac{d\underline{\mathbf{x}}}{dt} = \underline{\mathbf{w}}. \quad (12)$$

The set of semi-discrete equations (10)–(12) is discretized in time using a decoupled approach: the linear Stokes computation (linear viscous diffusive term) is integrated based on an implicit backward differentiation formula of order 2, the nonlinear convective term is integrated based on a simple method used by Karniadakis *et al.* [11], consisting in an explicit extrapolation of order 2. Finally the update of the position of mesh points is based on an explicit and conditionally stable Adams–Bashforth of order 3 (AB3).

Lastly the treatment of the pressure relies on a generalized block LU decomposition, using a standard fractional-step method with pressure correction.

#### 4. MOVING-GRID TECHNIQUE

As already mentioned in the previous sections, our free-surface flow computations are of interface-tracking type and rely on a moving-grid technique, allowing large amplitude motions of the free surface, generating a grid conforming to the shape of the free surface for an accurate and easy application of the boundary conditions on  $\partial\Omega_F(t)$ . Moreover a description as accurate as possible of the turbulent free-surface boundary layer is essential to our work. These points justify by themselves the choice of a moving-grid technique that increases the difficulty of the marginally intractable problem of turbulent viscous flow computations.

The computation of the mesh velocity  $\mathbf{w}$  in the internal fluid domain  $\Omega(t)$  is the corner-stone of the moving-grid technique developed in the framework of the ALE formulation. The values of the mesh velocity being prescribed on the boundary  $\partial\Omega(t) = \partial\Omega_F(t) \cup \partial\Omega_D$  as expressed by equations (5) and (7), the evaluation of  $\mathbf{w}$  in  $\Omega(t)$  can be obtained as the solution of an elliptic equation:

$$\mathcal{E}_x \mathbf{w} = \mathbf{0} \quad \text{in } \Omega(t). \quad (13)$$

This elliptic equation constitutes a classical choice for calculating the mesh velocity [8]. In the present case it is desirable to impose an additional constraint to the mesh velocity problem, in order to ensure the incompressibility of the mesh by imposing a divergence-free condition to  $\mathbf{w}$ :

$$\nabla_x \cdot \mathbf{w} = 0 \quad \text{in } \Omega(t). \quad (14)$$

Our choice for the elliptic operator  $\mathcal{E}_x$  is based on the assumption that the motion of the mesh nodes is equivalent to a steady Stokes flow, corresponding physically to an incompressible and elastic motion of the mesh.

The boundary-value steady Stokes problem for the mesh velocity can be formulated as follows:

$$\mathbf{w} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n} \quad \text{on } \partial\Omega_F(t), \quad (15)$$

$$\mathbf{w} \cdot \boldsymbol{\tau} = 0 \quad \text{on } \partial\Omega_F(t), \quad (16)$$

$$\mathbf{w} = \mathbf{0} \quad \text{on } \partial\Omega_D, \quad (17)$$

where  $\boldsymbol{\tau}$  is the local unit vector directly orthogonal to  $\mathbf{n}$ , and

$$\nabla_{\mathbf{x}} \cdot \tilde{\boldsymbol{\sigma}} = \mathbf{0} \quad \text{in } \Omega(t), \quad (18)$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{w} = 0 \quad \text{in } \Omega(t), \quad (19)$$

denoting by  $\tilde{\boldsymbol{\sigma}}$  the Cauchy stress tensor of the mesh defined by:

$$\tilde{\boldsymbol{\sigma}} = -\tilde{p}\mathbf{I} + \tilde{\nu}(\nabla_{\mathbf{x}}\mathbf{w} + \nabla_{\mathbf{x}}\mathbf{w}^T) \quad (20)$$

with  $\tilde{p}$  and  $\tilde{\nu}$  being respectively the fictitious mesh pressure and the fictitious kinematic viscosity of the mesh, characterizing the elasticity of the mesh in its motion.

The choice of this boundary-value problem for the mesh velocity has several justifications. Constraining the elliptic equation by a divergence-free condition for  $\mathbf{w}$  allows to ensure the conservation of the volume of the spectral elements, condition that is helpful in practice to have rapidly convergent computations [1]. In general the global volume of the computational domain may not be conserved, e.g. with an inflow-outflow imbalance, which requires (17) to be relaxed. In addition, the mesh velocity  $\mathbf{w}$  appears in the convective part of Eqs. (2), (8) and (10), together with the divergence-free fluid velocity  $\mathbf{v}$ . Moreover it is worth remembering that the divergence-free condition imposed to  $\mathbf{w}$  leads to a conservation of the metrics (the Jacobian being constant in time) when moving the mesh. Finally the unavoidable issue of fulfilling the geometric conservation law (GCL) in the ALE framework [3–5] is automatically solved when considering a divergence-free mesh velocity as a consequence of the work of Formaggia and Nobile in [5].

From a numerical point of view, the problem corresponding to the set of Eqs. (15)–(19) is discretized using the SEM, with a staggered grid  $\mathbb{P}_N - \mathbb{P}_{N-2}$  for the couple mesh  $(\mathbf{w}, \tilde{p})$ . An Uzawa decoupling technique is employed for the treatment of the fictitious pressure.

Based on the technique described earlier, we have developed the following moving-grid algorithm:

1. Input data: mesh  $\mathcal{M}^n$  at  $t = t_n$ , with nodal coordinates  $\mathbf{x}^n$ , fluid velocity  $\mathbf{v}^n$  on  $\partial\Omega_F^n$ , mesh velocity  $\mathbf{w}^n$  in  $\Omega^n \cup \partial\Omega^n$ ;
2. Step 1: steady Stokes computation of  $\mathbf{w}^{n+1}$  by Eqs. (15)–(19);
3. Step 2: update of the nodal coordinates Eq. (12); spectral element vertices are moved according to the AB3 scheme:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{\Delta t}{12}(23\mathbf{w}^n - 16\mathbf{w}^{n-1} + 5\mathbf{w}^{n-2}); \quad (21)$$

4. Creation of the new mesh  $\mathcal{M}^{n+1}$  with the new Gauss–Lobatto and Gauss–Lobatto–Legendre grids for each new spectral element;
5. Output data: mesh  $\mathcal{M}^{n+1}$  at time-step  $t = t_{n+1}$ , with nodes coordinates  $\mathbf{x}^{n+1}$ , mesh velocity  $\mathbf{w}^{n+1}$  in  $\Omega^{n+1} \cup \partial\Omega^{n+1}$ .

Two performance tests have been carried out on a study case where one edge of a squared mesh is deformed by a sine profile. Both of these tests aimed at verifying the spectral element volume conservation that is theoretically imposed by the divergence-free condition on  $\mathbf{w}$ . The first test is dedicated to the verification of the global volume conservation, by computing the relative change of the volume of the computational domain when moving the grid from the initial square to the deformed one. For several number of spectral elements and for a polynomial interpolation order ranging from 1 up to 12, the relative change of the volume of the computational domain is found to be smaller than the machine precision. The second test is also devoted to the volume conservation but now from a local perspective and by numerically computing the  $L^2(\Omega)$ - and  $L^2(\omega)$ -norm of the divergence of the mesh velocity  $\mathbf{w}$  for a polynomial interpolation order  $N$  ranging from 5 up to 12, where  $\omega$  is interior of the computational domain made of the spectral of elements of  $\Omega$  not sharing an edge with  $\partial\Omega$ . Results are presented on Fig. 1 and it is found that these norms are exponentially decreasing with  $N$  as expected when using a spectral element method [1]. Moreover we can note that the  $L^2(\omega)$ -norm of  $\nabla \cdot \mathbf{w}$  has a faster rate of convergence than the  $L^2(\Omega)$ -norm. This is justified by the fact that the divergence-free constraint cannot easily be enforced at the grid points located in the vicinity of the boundaries of the computational domain  $\Omega$ .

## 5. MESH-TRANSFER OPERATION

In the previous section was presented the moving-grid technique used in our work to move the grid points at each time-step, generating a new mesh. Depending on the amplitude of the mesh deformation at each time-step, this technique can be applied during an important number of iterations.

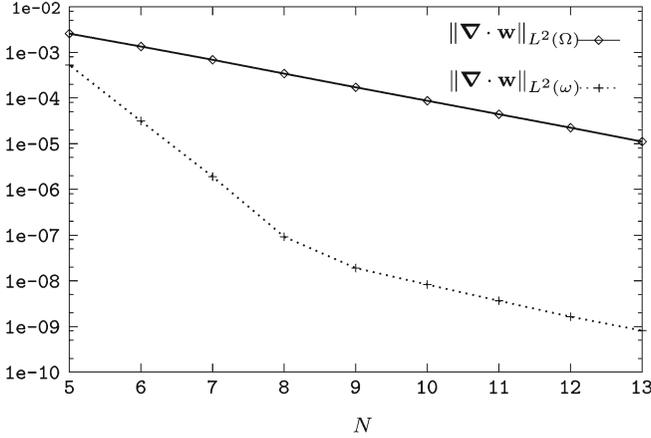


Fig. 1.  $L^2$ -norms of the divergence of the mesh velocity  $\mathbf{w}$  versus polynomial interpolation order  $N$  (Log scale).

Nevertheless the mesh obtained by moving the grid nodes can be too convoluted therefore affecting the accuracy and the convergence of the simulation. Consequently a re-meshing operation is to be called by a specific control parameter (e.g. a discrete Jacobian positiveness criterion) to provide a new mesh topology. Before starting the ALE Navier–Stokes computation at the next time-step on this newly created mesh, it is mandatory to transfer some information from the previous mesh to the new one. The main requirement imposed to this so-called mesh-transfer operation is to conserve the spectral accuracy of the SEM. The information to be transferred comprises six fields: the fluid velocities  $\mathbf{v}^n$ ,  $\mathbf{v}^{n-1}$  and the mesh velocities  $\mathbf{w}^n$ ,  $\mathbf{w}^{n-1}$ ,  $\mathbf{w}^{n-2}$  (time-integration schemes are of order 2 for  $\mathbf{v}$  and 3 for  $\mathbf{w}$ ) and also the pressure at the current time-step (use of a pressure correction technique). As written in Sec. 3, the velocities are expanded over a GLL grid and the pressure over a GL one. Therefore our mesh-transfer technique must be capable of transferring fields defined over GL and GLL grids.

Our mesh-transfer algorithm for GL grids being based on the one for GLL grids, we will start presenting in detail the latter. Let us consider two meshes  $\mathcal{M}^1$  and  $\mathcal{M}^2$  corresponding to different mesh topology of the same computational domain and the mesh-transfer operation from  $\mathcal{M}^1$  to  $\mathcal{M}^2$ . In the sequel we will assume that we have the following decompositions in terms of spectral elements:

$$\Omega_i \cup \partial\Omega_i = \bigcup_{e=1}^{E_i} \Omega^{i,e} \quad \text{for } i = 1, 2. \quad (22)$$

As the computational domain remains unchanged, for each spectral element  $\Omega^{2,e}$  of  $\mathcal{M}^2$  we have:

$$\Omega^{2,e} \subset (\Omega_1 \cup \partial\Omega_1) \quad \forall e = 1, \dots, E_2. \quad (23)$$

Due to Eq. (23) our mesh-transfer technique only requires an interpolation procedure. Let us note the physical location of the set of GLL grid points of a spectral element  $\Omega^{2,e_2}$  ( $e_2 = 1, \dots, E_2$ ) by  $\{\mathbf{x}_{ij}^{2,e_2}\}$  with ( $i = 1, \dots, N_{x,2} + 1; j = 1, \dots, N_{y,2} + 1$ ),  $N_{x,2}$  (resp.  $N_{y,2}$ ) being the order of the polynomial interpolation in the  $x$ -direction (resp.  $y$ -direction) for the mesh  $\mathcal{M}^2$  (with the same notations,  $N_{x,2}$  and  $N_{y,2}$  can be different from  $N_{x,1}$  and  $N_{y,1}$  respectively). The proposed algorithm can be summarized in three steps:

1. Find the spectral element  $\Omega^{1,e_1}$  of  $\mathcal{M}^1$  containing  $\mathbf{x}_{ij}^{2,e_2}$ ;
2. Determine the position  $\mathbf{r}^{1,e_1}$  of  $\mathbf{x}_{ij}^{2,e_2}$  within the parent element  $\hat{\Omega}^{1,e_1}$  of  $\Omega^{1,e_1}$ ;
3. Compute the value of the field at the point  $\mathbf{x}_{ij}^{2,e_2}$  given  $\mathbf{r}^{1,e_1}$ , the GLL Lagrangian interpolation basis and the values of the field at the GLL grid points of  $\Omega^{1,e_1}$ .

The first step causes no difficulty in its implementation. The second step uses a transfinite interpolation procedure in each spectral element, in order to invert the iso-parametric mapping  $\Phi$ :

$$\mathbf{r}^{1,e_1} = (r^{1,e_1}, s^{1,e_1}) = \Phi^{-1}(\mathbf{x}^{2,e_2}) \quad \text{with } \mathbf{r}^{1,e_1} \in \hat{\Omega}^{1,e_1} = [-1, 1]^2. \quad (24)$$

In practice, the inversion is carried out differently depending on the topology of the spectral element. With quadrangular spectral elements, our algorithm performs a direct analytical inversion of the affine mapping  $\Phi$  which is computationally inexpensive. With deformed spectral elements [1], the inversion of  $\Phi$  relies on the so-called ‘inverse iso-parametric mapping technique’ from Lee and Bathe [12] which is based on a Newton–Raphson type iterative procedure.

Finally in the last step, efficient routines compute the following spectral interpolation:

$$\mathbf{u}(\Phi^{-1}(\mathbf{x}_{ij}^{2,e_2})) = \mathbf{u}(r^{1,e_1}, s^{1,e_1}) = \sum_{k=0}^{N_{1,x}} \sum_{l=0}^{N_{1,y}} \mathbf{u}_{kl} \pi_k(r^{1,e_1}) \pi_l(s^{1,e_1}), \quad (25)$$

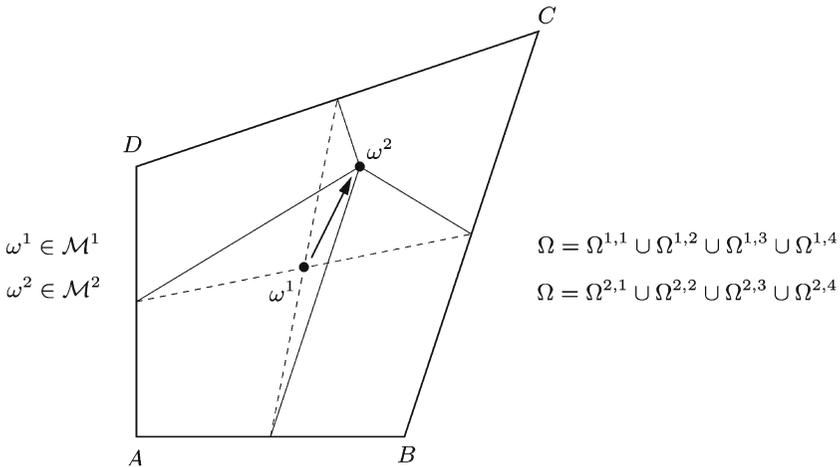
with  $\{\pi_j(\xi)\}_{j=0}^{N_{p,1}}$  and  $p = x, y$ , the one-dimensional GLL Lagrangian interpolation basis of degree  $N_{p,1}$ . As said earlier the mesh-transfer technique

for GL grids relies on the one for the GLL grids. In our simulations, the only GL-interpolated field that has to be mesh-transferred is the pressure field. Therefore, by interpolating the pressure on the GLL grid, then by applying the GLL mesh-transfer operation introduced earlier and finally by interpolating back on the GL grid, we manage to perform the requested operation. It is important to minimize the occurrence of a re-meshing as our mesh-transfer technique is computationally expensive even for quadrangular elements (affine iso-parametric mapping). A more detailed assessment of the performance of this technique is provided at the end of this section.

This mesh-transfer operation has been extensively tested in order to ensure its compatibility with the SEM, regarding its exponential rate of convergence. Tests involving the following two key parameters have been carried out: the polynomial interpolation order  $N$  and the amplitude of the change in topology of the grid when re-meshing.

The set-up is presented in Fig. 2 and is made of a mesh comprising four spectral elements. The change in topology of the mesh is prescribed by moving only the vertex  $\omega$  (see Fig. 2) common to all four spectral elements and afterwards the mesh-transfer operation is performed.

To evaluate the dependence of our technique with the interpolation order  $N$ , the central vertex is moved to produce a topological change in the mesh by a factor of approximately 10%. An analytical field  $f$  is calculated on the initial mesh and mesh-transferred onto the distorted



**Fig. 2.** Sketch of the computational domain  $\Omega$ , the two meshes  $\mathcal{M}^1$  and  $\mathcal{M}^2$  and their spectral element decompositions before and after a prescribed re-meshing operation obtained by moving the central vertex  $\omega$ .

mesh, leading to the interpolated field  $\tilde{f}$ . The interpolation error is defined by  $\varepsilon = \|f - \tilde{f}\|_{L^2(\Omega)}$  and computed values are presented in Table I, showing a conservation of the exponential rate of convergence.

**Table I.** Evolution of the Error  $\varepsilon$  with the Spectral Interpolation Order  $N$

$N$	$\varepsilon = \ f - \tilde{f}\ _{L^2(\Omega)}$	$N$	$\varepsilon = \ f - \tilde{f}\ _{L^2(\Omega)}$
3	7.232e-03	12	4.400e-12
4	1.487e-03	13	9.850e-14
5	1.367e-04	14	1.252e-14
6	2.307e-05	15	3.602e-15
7	1.457e-06	16	3.354e-15
8	2.067e-07	17	1.843e-15
9	8.382e-09	18	1.585e-15
10	1.172e-09	19	1.105e-15
11	3.383e-11	20	1.151e-15

To characterize the effect of the distortion of the mesh on our mesh-transfer operation, all possible positions of the moving vertex within the computational domain  $\Omega$  were envisaged. In particular, we present here the case where  $\omega$  is moved along the diagonal  $AC$  of the computational domain  $\Omega$  as shown in Fig. 2. Its motion is characterized by the set of coordinates  $(\alpha, \beta)$  of  $\omega$  in the parent domain  $\hat{\Omega} = [-1, 1]^2$ . The interpolation error  $\varepsilon$  was again computed for three values of  $N$  and results appearing in Table II, show that our technique is totally independent on the amplitude of topological change of the mesh due to the re-meshing operation.

Lastly, the computational expense of the mesh-transfer has been evaluated for a polynomial degree  $N=10$  in both directions ( $19^2$  grid points for this 2D grid), and as previously for a topological change in the mesh by a factor of approximately 10%, corresponding to a “small” 2D case. The results confirm the afore-mentioned cost: a complete mesh-transfer corresponds to approximately 100 Navier–Stokes solves depending on the value of the time-step.

## 6. CONCLUSION AND FUTURE STUDIES

A novel isochoric moving-grid technique and mesh-transfer technique for spectral element grids have been presented. Both of these techniques are the corner-stones of our computations of turbulent free-surface flows

**Table II.** Evolution of the Error  $\varepsilon$  when  $\omega$  Moves Along the Diagonal AC for Three Different Values of  $N$

$\alpha = \beta$	$\varepsilon = \ f^2 - \tilde{f}^2\ _{L^2(\Omega)}$		
	$N=8$	$N=10$	$N=12$
-0.9	2.100e-07	1.005e-09	3.849e-12
-0.8	2.267e-07	1.111e-09	4.448e-12
-0.7	2.000e-07	1.006e-09	3.906e-12
-0.6	1.928e-07	1.106e-09	4.073e-12
-0.5	2.289e-07	1.033e-09	3.786e-12
-0.4	1.847e-07	1.053e-09	4.199e-12
-0.3	2.326e-07	1.160e-09	4.166e-12
-0.2	2.231e-07	1.204e-09	4.332e-12
-0.1	2.067e-07	1.172e-09	4.400e-12
0.0	4.563e-16	1.199e-15	7.886e-16
0.1	2.067e-07	1.172e-09	4.400e-12
0.2	2.231e-07	1.204e-09	4.332e-12
0.3	2.326e-07	1.160e-09	4.166e-12
0.4	1.847e-07	1.053e-09	4.199e-12
0.5	2.289e-07	1.033e-09	3.786e-12
0.6	1.928e-07	1.106e-09	4.073e-12
0.7	2.000e-07	1.006e-09	3.906e-12
0.8	2.267e-07	1.111e-09	4.448e-12
0.9	2.100e-07	1.005e-09	3.849e-12

using spectral element method. Part of the work was to ensure that these two techniques have no effect on the exponential rate of convergence, the main reason of our choice of the spectral element method. We have obtained positive results all along the extensive series of tests carried out to verify the behaviour of this rate of convergence. The development of an automatized re-meshing scheme coupled to a re-meshing control parameter is still under investigation.

Our next goal is to simulate three-dimensional turbulent free-surface flows using the techniques presented in this paper with the difficult task of gaining a better insight into the physics involved in the thin turbulent boundary layer near the free surface.

## 7. ACKNOWLEDGMENTS

The authors would like to thank Professor Einar Rønquist and Professor Yvon Maday for fruitful discussions. This research is being partially funded by a Swiss National Science Foundation Grant (No. 200020-101707), whose support is gratefully acknowledged.

## REFERENCES

1. Deville, M. O., Fischer, P. F., and Mund, E. H. (2002). *High-Order Methods for Incompressible Fluid Flow*. ISBN 0-521-45309-7, Cambridge University Press, Cambridge.
2. Donea, J. (1983). Arbitrary Lagrangian–Eulerian finite element methods. In Belytschko, T. B., and Hugues, T. J. R. (eds.), *Computational Methods for Transient Analysis*, North Holland pp. 474–516.
3. Farhat, C., Geuzaine, P., and Grandmont, C. (2001). The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *J. Comput. Phys.* **174**, 669–694.
4. Farhat, C., and Geuzaine, P. (2004). Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. *Comput. Meth. Appl. Mech. Engng.* **193**, 4073–4075.
5. Formaggia, L., and Nobile, F. (2004). Stability analysis of second-order time accurate schemes for ALE–FEM. *Comput. Meth. Appl. Mech. Engng.* **193**, 4097–4116.
6. Güler, I., Behr, M., and Tezduyar, T. E. (1999). Parallel finite element computation of free-surface flows. *Comput. Mech.* **23**, 117–123.
7. Hirt, C. W., Amsden, A. A., and Cook, J. L. (1974). An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.* **14**, 227–253.
8. Ho, L. W., and Patera, A. T. (1990). A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows. *Comput. Meth. Appl. Mech. Engng.* **80**, 355–366.
9. Hodges, B. R., and Street, R. L. (1999). On simulation of turbulent nonlinear free-surface flows. *J. Comput. Phys.* **151**, 425–457.
10. Johnson, A. A., and Tezduyar, T. E. (1994). Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput. Meth. Appl. Mech. Engng.* **119**, 73–94.
11. Karniadakis, G. E., Israeli, M., and Orszag, S. A. (1991). High-order splitting methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **97**, 414–443.
12. Lee, N. S., and Bathe, K. J. (1994). Error indicators and adaptive remeshing in large deformation finite element analysis. *Finite Elements Anal. Design* **16**, 99–139.
13. Maday, Y., and Patera, A. T. (1989). Spectral element methods for the incompressible Navier–Stokes equations. *State-of-the-Art Survey on Computational Mechanics*. In Noor, A. K., and Oden, J. T. (eds.), ASME. New-York, pp. 71–142.
14. Patera, A. T. (1984). Spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.* **54**, 468–488.
15. Ramaswamy, B., and Kawahara, M. (1987). Arbitrary Lagrangian–Eulerian finite-element method for unsteady, convective, incompressible viscous free-surface fluid-flow. *Int. J. Numer. Meth. Fl.* **7**(10), 1053–1075.