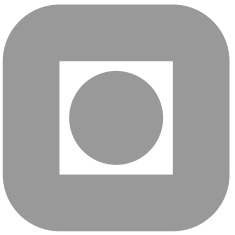
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

A Survey of Some Methods for Moving Grid and Grid Adaption

by
Bjarte Hægland and Bård Skaflestad

PREPRINT
NUMERICS NO. 2/2002



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL `http://www.math.ntnu.no/preprint/numerics/2002/N2-2002.ps`
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Symbols

$\underline{\mathbf{x}}$	Vector
$\mathbf{A}$	Matrix
$\mathbb{R}$	The real numbers
$\mathbb{R}^d$	d-dimensional space
D/Dt	Total derivative
$\|\|\|\cdot\|\|\|$	Problem dependent energy norm
$(\cdot,\cdot)$	Inner product
$\|\cdot\|$	Norm
$\operatorname{div}\underline{\mathbf{u}}$	Divergence of $\underline{\mathbf{u}}$
$\operatorname{curl}\underline{\mathbf{u}}$	Curl of $\underline{\mathbf{u}}$
$\operatorname{grad} f$	Gradient of f
$\operatorname{trace}\mathbf{A}$	Matrix trace
$\operatorname{diag} a$	Diagonal matrix with a on its diagonal
$\mathcal{L}$	Differential operator $\mathcal{L}$
$W^{m,p}(\Omega), m \geq 0, p \in [1,\infty]$	Sobolev space
$\|\cdot\|_{m,p,\Omega}$	Norm on $W^{m,p}(\Omega)$
$\|\cdot\|_{m,p,\Omega}$	Seminorm on $W^{m,p}(\Omega)$

1 Introduction

Numerical simulation of fluid flow has been studied for quite some time. Many methods have been devised and applied to various branches of flow simulation with varying results, and although the basic methods are well known, this is still field of active research. Simulating turbulent flows such as breaking waves is largely infeasible, even when using the most powerful super-computers available. Nevertheless, the insight gained from simulation has proved valuable in understanding the basic laws of nature and in designing and building structures and machinery.

Requiring large number of unknowns, simulating realistic flows and capturing the interaction between fluid and structure or free surfaces is very costly with respect to computer time and developing numerical methods which are more efficient is a desirable goal. One way of searching for that goal is to create methods which adapt to the solution of the physical problem. It has been shown that adaptive methods in which the mesh—the discretisation of the computational domain—changes, either continuously or at regular or irregular intervals throughout the computation, have greatly reduced the demand for larger number of unknowns.

In this paper we review some of the methods which have appeared in the literature and focus mainly on two aspects of adaptive methods: Error estimation and changing the mesh. We begin in section 2 by presenting the governing equations for fluid flow and continue in section 3 by showing a framework for rewriting these equations in a form independent of a particular mesh thus enabling changing meshes in time-dependent flow problems. Section 4 presents a short overview of the different classes of adaptive methods and section 5 proceeds by showing some elasticity models for mesh adaption. Adaptive methods based on moving meshes is presented in section 6 and section 7 deals with h-adaptive methods which varies the number of degrees of freedom and the size of the elements. Finally, section 8 studies various methods for estimating error in finite element applications and section 9 lists some open problems and possible future work.

2 The Navier-Stokes Equations

Fluids have properties differing greatly from those of solids. A fluid will always deform significantly when acted upon by forces but the rate at which deformation occurs is greatly dependent on atomic and molecular aspects of the substance in question [1]. On the other hand, a solid may greatly resist such deformation, at least if the forces are not of excessive magnitude. This difference has lead to some philosophical discussion as to what is the proper frame of reference for fluid mechanics and which natural processes contribute to the observable physical behaviour.

Creating mathematical models for fluid motion facilitates quantitative statements and predictions about a physical system. Thus, modelling fluid flows enables, in addition to understanding complex flows such as oceans, wind, waves and aeroplanes, prediction of interactions between fluid and structure, determination of fluid forces exerted upon an object and similar applications. Having models also enables numerical treatment in addition to experiments. With the advent of modern multi-processor electronic computers, the highly compute intensive applications arising from the models are more feasible than ever before, and on the verge of replacing experiments in some area of fluid mechanics research. But there are still important applications for which *computational fluid dynamics* is not sufficient, including, in particular, highly convection dominated viscous flows in

complex geometries.

The basic mathematical models for viscous fluid flow and have been known for more than a hundred years, yet are still relatively new compared to the models of rigid body mechanics. At the heart of these models lie three conservation principles,

- conservation of mass,
- conservation of momentum,
- conservation of energy,

all of which are reasonable and pervasive in physical models of other natural phenomena as well. When first introduced these principles were merely postulated, but experience has shown excellent correlation between the resultant models and experiments in a variety of applications.

2.1 Conservation of Mass

In words, the principle of mass conservation states: *In a given deforming material volume, $\Omega_p(t) \subset \mathbb{R}^d$, the mass $M(t)$ is constant with respect to time t.* Thus, using mathematical terms

$$\frac{\mathrm{d}}{\mathrm{d}t} M(t) = 0, \tag{1}$$

with the total mass expressed as

$$M(t) = \int_{\Omega_p(t)} \rho \, \mathrm{d}\Omega.$$

The quantity ρ is the volumetric mass of the fluid, also known as the *density*. Considering isolated fluid particles, the time rate of change of a sufficiently smooth material property, f, when following the fluid flow may be expressed as

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + (\underline{\mathbf{u}} \cdot \nabla) f, \tag{2}$$

in which $\underline{\mathbf{u}} = (u_1, \ldots, u_d)^T$ is the fluid velocity. Thus, taking into account the moving material volume $\Omega_p(t)$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_p(t)} f(\underline{\mathbf{x}}, t) \, \mathrm{d}\Omega = \int_{\Omega_p(t)} \frac{Df}{Dt} + f \operatorname{div} \underline{\mathbf{u}} \, \mathrm{d}\Omega, \tag{3}$$

a result which may also be derived from the Reynolds transport theorem [2]. Combining equations (1) and (3) whilst interpreting the material property f as the density ρ leads to

$$\int_{\Omega_p(t)} \left[\frac{D\rho}{Dt} + \rho \operatorname{div} \underline{\mathbf{u}} \right] \mathrm{d}\Omega = 0 \tag{4}$$

which states that any mass transported into $\Omega_p(t)$ is accompanied by a change of volume and possibly an increased density when the global effects are summed up. Furthermore, since equation (4) is valid for *any* material volume, the equation can be restated in an equivalent local form for all $\underline{\mathbf{x}} \in \Omega_p(t)$

$$\frac{D\rho}{Dt} + \rho \operatorname{div} \underline{\mathbf{u}} = 0. \tag{5}$$

An incompressible fluid is characterised by the assumption that $D\rho/Dt = 0$ or, in other words, that ρ is constant throughout the fluid at all times. Thus, for incompressible fluids, the velocity field $\underline{\mathbf{u}}$ must be solenoidal, $\operatorname{div} \underline{\mathbf{u}} = 0$.

2.2 Conservation of Momentum

The principle of conservation of linear momentum is a generalisation of Newton's second law of motion to continuous materials. It states that *the time rate of change of momentum of a deforming material region embedded in $\Omega_p(t)$ is equal to the forces applied to that region.* Decomposing the acting force into a body force component $\underline{\mathbf{f}}(\underline{\mathbf{x}}, t)$ and a surface tension force component $\underline{\mathbf{t}}(\underline{\mathbf{x}}, t)$, this principle may be written mathematically as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_p(t)} \rho\underline{\mathbf{u}}\,\mathrm{d}\Omega = \int_{\partial\Omega_p(t)} \underline{\mathbf{t}}\,\mathrm{d}S + \int_{\Omega_p(t)} \rho\underline{\mathbf{f}}\,\mathrm{d}\Omega. \tag{6}$$

The body force component includes effects of gravity, compressional loads and similar external forces. The surface contact forces on the other hand may not be as easily tractable in the indicated form. However, the Cauchy principle relates the surface tension to the oriented normal vector on a given surface. In particular, there exists a tensor field, $\boldsymbol{\sigma}(\underline{\mathbf{x}}, t)$, such that

$$\underline{\mathbf{t}}(\underline{\mathbf{x}}, t) = \boldsymbol{\sigma}(\underline{\mathbf{x}}, t)\underline{\mathbf{n}}(\underline{\mathbf{x}}, t) \tag{7}$$

with $\underline{\mathbf{n}}(\underline{\mathbf{x}}, t)$ denoting the oriented normal vector evaluated at the surface point $\underline{\mathbf{x}} = (x_1, \ldots, x_d)^T$ at time t.

Inserting (7) into equation (6) and applying the divergence theorem and the transport property (2) yields the local equation of fluid motion

$$\rho\frac{D\underline{\mathbf{u}}}{Dt} = \operatorname{div}\boldsymbol{\sigma} + \rho\underline{\mathbf{f}}. \tag{8}$$

This equation is valid for any fluid for which the tensor field $\boldsymbol{\sigma}$ can be derived. Furthermore, the stress tensor has to be symmetric in order to support conservation of angular momentum in addition to linear momentum.

2.3 Constitutive Equations

The equation of motion (8) is valid for any fluid. Thus, in any given application we need to derive expressions for the stress tensor field if the equation is to be practically useful. Such expressions are determined by means of "constitutive equations" which relate $\boldsymbol{\sigma}$ to the thermodynamic pressure, p, and to the rate of deformation tensor, $\boldsymbol{d}$. The latter is defined as the symmetric part of the velocity gradient and may be written as

$$\boldsymbol{d} = \frac{1}{2}\big(\nabla\underline{\mathbf{u}} + (\nabla\underline{\mathbf{u}})^T\big). \tag{9}$$

In the classical case of Newtonian fluids, the stress tensor is then written as

$$\boldsymbol{\sigma} = (-p + \lambda\operatorname{tr}\boldsymbol{d})\boldsymbol{I} + 2\mu\boldsymbol{d} \tag{10}$$

in which μ is a real parameter known as the dynamic shear viscosity and λ is the bulk viscosity. μ and λ are macroscopic and measurable representations of the molecular properties of the material in question and are often obtained from experiments. Since an incompressible fluid obeys $\operatorname{tr}\boldsymbol{d} \propto \operatorname{div}\underline{\mathbf{u}} = 0$,

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{d}$$

for incompressible Newtonian fluids.

Inserting this result into (8) establishes the incompressible Navier-Stokes (NS) equations

$$\rho\frac{D\underline{\mathbf{u}}}{Dt} + \nabla p - \nabla\cdot\left(\mu(\nabla\underline{\mathbf{u}} + (\nabla\underline{\mathbf{u}})^T)\right) = \rho\underline{\mathbf{f}}. \tag{11}$$

This form of the NS equations is known in the literature as the "stress" or "tension" formulation and incorporates the possibility of having the viscosity varying inside the fluid domain. In the special case of $\mu = \text{const}$, these equations simplify to

$$\rho\frac{D\underline{\mathbf{u}}}{Dt} + \nabla p - \mu\nabla^2\underline{\mathbf{u}} = \rho\underline{\mathbf{f}}$$

in which the Laplacian operator ∇^2 is applied to each component of $\underline{\mathbf{u}}$. This formulation is known in the literature as the "velocity" or "Laplacian" formulation of the NS equations. The final mathematical model for incompressible Newtonian fluid flow with constant viscosity is thus the following system of partial differential equations (PDEs)

$$\begin{cases} \dfrac{\partial\underline{\mathbf{u}}}{\partial t} + (\underline{\mathbf{u}}\cdot\nabla)\underline{\mathbf{u}} + \dfrac{1}{\rho}\nabla p - \nu\nabla^2\underline{\mathbf{u}} = \underline{\mathbf{f}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \operatorname{div}\underline{\mathbf{u}} = 0 \end{cases} \tag{12}$$

in which the total derivative operator D/Dt has been expanded into its constituent parts and $\nu = \mu/\rho$ is known as the kinematic viscosity.

As stated, the system (12) does not constitute a complete model for a given fluid flow. Boundary and initial conditions must be supplied in order to complete the system and provide for existence and uniqueness of solutions. Dividing the boundary of the domain into non-overlapping parts $\partial\Omega_{p_D}$ and $\partial\Omega_{p_N}$ such that $\partial\Omega_{p_D}\cup\partial\Omega_{p_N} = \partial\Omega_p$ and $\partial\Omega_{p_D}\cap\partial\Omega_{p_N} = \emptyset$, the boundary conditions may be similarly divided. A Dirichlet type boundary condition in which the velocity is prescribed is imposed on $\partial\Omega_{p_D}$, whereas prescribed surface tension may be imposed on $\partial\Omega_{p_N}$. As surface tension is related to the derivative of the velocity, this amounts to a Neumann type boundary condition. However, prescribing the surface tension requires the more general stress formulation (11) of the NS equations.

A thorough discussion of boundary conditions is beyond the scope of this introductory section, but it is important to be aware of the fact that the PDE system (12) is not completely general with respect to boundary conditions. Further details of admissible boundary conditions as well as rescalings and non-dimensionalisation of the basic equations are available in the literature in for instance [3, 2]

2.4 Mathematical Structure and Variational Formulation

For notational simplicity we use the general framework of indicial notation and the Einstein summation convention for repeated indices in this section. Written in this form, the NS equations read as

$$\begin{cases} \dfrac{\partial u_i}{\partial t} + u_j\cdot\dfrac{\partial u_i}{\partial x_j} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial x_i} - \nu\nabla^2 u_i = f_i \\ \qquad\qquad\qquad\qquad \dfrac{\partial u_i}{\partial x_i} = 0. \end{cases} \tag{13}$$

This problem is known to be difficult in general and many theoretical questions remain, as yet, unanswered—in particular for realistic flows dominated by convection. However,

when developing computer software for solving the equations numerically, there are some distinctive features to notice.

The innermost problem incorporated in the NS equations is the well known Poisson problem

$$-\nu\nabla^2 u_i = f_i$$

which controls the viscous effects. The Laplacian operator contains the highest spatial derivative of the unknown fluid velocity and thus determines regularity requirements of the functions in question. A computer programme thus has to be able to solve the Poisson equation as test case. Extending this problem by taking into account the pressure and the solenoidal velocity constraint yields the so-called Stokes problem

$$\begin{cases} \dfrac{1}{\rho}\dfrac{\partial p}{\partial x_i} - \nu\nabla^2 u_i = f_i \\ \dfrac{\partial u_i}{\partial x_i} = 0 \end{cases} \tag{14}$$

which governs highly viscous or "creeping" fluid flow. When introducing the solenoidality constraint, the system becomes an instance of a general saddle point problem. It is then necessary to introduce conditions which guarantee solvability of the resulting discrete systems. In addition any prescribed boundary conditions may have to fulfil certain compatibility conditions as well.

Introducing convection into the problem gives rise to the non-linear term $(\underline{\mathbf{u}}\cdot\nabla)\underline{\mathbf{u}}$ and the steady Navier-Stokes equations

$$\begin{cases} u_j\cdot\dfrac{\partial u_i}{\partial x_j} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial x_i} - \nu\nabla^2 u_i = f_i \\ \dfrac{\partial u_i}{\partial x_i} = 0 \end{cases} \tag{15}$$

which govern viscous fluid flow in which neither viscous nor convective effects can be neglected. Solving this problem is compounded by the non-linear term. A facility for solving non-linear systems of equations is needed and ensuring convergence of this solve is normally only possible in a vicinity of the solution to the equations. Finally, allowing temporal variation, leads to the full Navier-Stokes equations in (13) or, if needed, in (11).

As the PDE is of both parabolic and hyperbolic character in addition to constituting a saddle point problem, there is no minimisation statement from which a variational problem can be derived. However, choosing the velocity field $\underline{\mathbf{u}}$ from a space X of solenoidal—or divergence free—vector fields satisfying the imposed boundary conditions and the pressure p from a space M of scalar functions satisfying any necessary compatibility and uniqueness conditions, allows the Navier-Stokes equations to be recast in a variational form as [4]: Find $\underline{\mathbf{u}} \in X$ and $p \in M$ such that for almost all $t \in [0, T)$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\underline{\mathbf{u}}(t), \underline{\mathbf{v}}) + \nu a(\underline{\mathbf{u}}(t), \underline{\mathbf{v}}) + c(\underline{\mathbf{u}}(t); \underline{\mathbf{u}}(t), \underline{\mathbf{v}}) + \frac{1}{\rho} b(\underline{\mathbf{v}}, p(t)) = (\underline{\mathbf{f}}(t), \underline{\mathbf{v}})\ \forall \underline{\mathbf{v}} \in X \tag{16}$$

$$b(\underline{\mathbf{u}}(t), q) = 0\ \forall q \in M \tag{17}$$

$$\underline{\mathbf{u}}(0) = \underline{\mathbf{u}}_0. \tag{18}$$

Here $a(\underline{\mathbf{v}}, \underline{\mathbf{w}}) = \int_{\Omega_p(t)} \nabla\underline{\mathbf{v}} : \nabla\underline{\mathbf{w}}\,\mathrm{d}\Omega$ is a bilinear form derived from the viscous term, $b(\underline{\mathbf{w}}, q) = \int_{\Omega_p(t)} q\,\mathrm{div}\,\underline{\mathbf{w}}\,\mathrm{d}\Omega$ is the variational form of the solenoidality constraint and

$c(\underline{\mathbf{w}}; \underline{\mathbf{v}}, \underline{\mathbf{z}}) = \int_{\Omega_p(t)} \left[(\underline{\mathbf{w}} \cdot \nabla)\underline{\mathbf{v}}\right] \cdot \underline{\mathbf{z}} \, \mathrm{d}\Omega$ is a trilinear form derived from the non-linear convection term. In addition $(\underline{\mathbf{w}}, \underline{\mathbf{v}}) = \int_{\Omega_p(t)} \underline{\mathbf{w}} : \underline{\mathbf{v}} \, \mathrm{d}\Omega$ is the standard L^2 inner product for scalar or vector-valued functions defined on $\Omega_p(t)$.

3 Arbitrary Lagrangian Eulerian Formulation

3.1 Introduction

In solving computational fluid problems there are two traditional points of view to formulating the setting. On one hand we have the Eulerian approach, in which the domain of interest is kept fixed. That is our reference frame remains stationary while the fluid flows through it. The alternative way to look at the problem is to use the Lagrangian description. In this case we do not work in a domain fixed in time and space, but instead concentrate on a specific set of particles. The coordinate system is now embedded in the fluid. The domain of interest captures the set of particles and follow them and deform along with the volume made up of the particles.

Both approaches have their advantages and disadvantages. In a Lagrangian description the mesh is fixed to the material and no convective effects are present. The Lagrangian point of view may result in good resolution of details and the capability to resolve boundaries. A serious disadvantage though, become apparent when experiencing great deformation of the domain. These cases may result in great distortions of the computational grid. Problems involving flow over sharp corners may not be represented appropriately. For example, look at the somewhat artificial case of fluid flowing through a nozzle. It does not take a lot of fantasy to imagine that the grid moving into the narrowing nozzle soon becomes too skewed and unsuited for computations. The Eulerian point of view, on the other hand, can handle distortions, but possibly at the cost of resolution. The Eulerian approach is obviously not suited for representing free boundaries or moving boundaries like simulation of material deformation.

When considering problems like flow through blood vessels, wing oscillation or multibody simulation or fluid-structure interaction dynamic meshes are essential. None of the approaches mentioned above is quite capable of modelling the interaction between objects moving relatively to each other or dealing with moving, possibly deforming boundaries. New ways of attacking the problem are needed and one such approach is the Arbitrary Lagrangian Eulerian (ALE) method. The ALE approach is (somewhat) a mixture of the Eulerian and Lagrangian points of view. The ALE formulation introduces a reference computational domain that is moving arbitrarily, or in other words independently of the fluid or material. We can recover the Eulerian or Lagrangian points of view by setting the velocity of the domain equal to zero or equal to the material/fluid velocity respectively. In the ALE case the mesh does not have to remain fixed to the material in the course of deformation, thus possibly eliminating entanglement of the grid.

3.2 Geometry and Conservation Laws in ALE Formulation

In an ALE formulation the system under study occupies a moving domain Ω_t in its present configuration. As indicated by the suffix, Ω_t is time dependent, which means we have to solve our conservation laws in a non-stationary domain. To overcome this difficulty it is convenient to recast the conservation laws into a fixed reference configuration Ω_0 (usually the initial configuration). For this reason we assume there exists a time-dependent

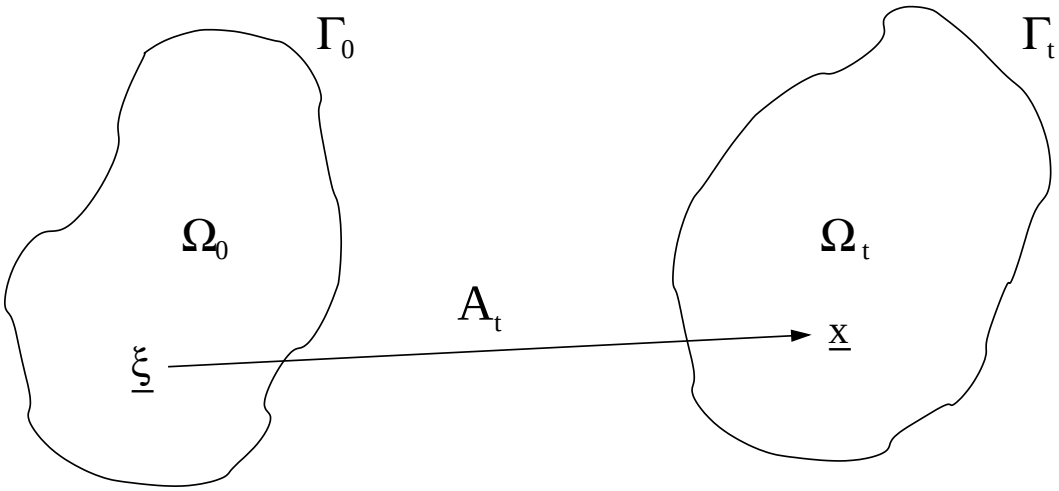


Figure 1: ALE formulation

mapping A_t, which describes the connection between the current configuration and the reference configuration.

$$\begin{aligned} A_t : \Omega_0 \subset \mathbb{R}^d &\longrightarrow \Omega_t \subset \mathbb{R}^d \\ \underline{\xi} &\longmapsto \underline{\mathbf{x}}(\underline{\xi}, t) = A_t(\underline{\xi}) \end{aligned}$$

The situation is illustrated in Figure 1. We denote the coordinates in the reference configuration Ω_0 by $\underline{\xi}$ and the coordinates in the current configuration Ω_t by $\underline{\mathbf{x}}$. Apart from small changes we will adopt the notation of Formaggia and Nobile [5] in the rest of this section. We denote by $J_{A_t} = \det \nabla A_t = \det \partial \underline{\mathbf{x}} / \partial \underline{\xi}$ the Jacobian determinant of the ALE mapping. The velocity of the particle at position $\underline{\mathbf{x}}$ in Ω_t is denoted $\underline{\mathbf{w}}(\underline{\mathbf{x}}) = \partial \underline{\mathbf{x}} / \partial t|_{\underline{\xi}}$.

3.2.1 Formulation of The Conservation Laws

Several studies involving ALE have presented to us a wealth of equations tuned for the ALE framework. We do not attempt to cover them all so we stick to a general presentation of recasting a PDE, and in special a conservation law, into the equivalent ALE format. Transporting the conservative laws, written in Eulerian format, in Ω_t into the reference configuration Ω_0 is done using two main tools: the chain rule of differentiation and the identity $\partial J_{A_t} / \partial t|_{\underline{\xi}} = J_{A_t} \nabla_{\underline{\mathbf{x}}} \cdot \underline{\mathbf{w}}$.

Given a PDE of the form

$$\left. \frac{\partial u}{\partial t} \right|_{\underline{\mathbf{x}}} + \mathcal{L}(u) = f, \tag{19}$$

we may incorporate the ALE map $\underline{\mathbf{x}} = \underline{\mathbf{x}}(\underline{\xi}, t) = A_t(\underline{\xi})$ and straight forward use of the chain rule of differentiation results in the ALE counterpart of our PDE

$$\left. \frac{\partial u}{\partial t} \right|_{\underline{\xi}} + \mathcal{L}(u) - \underline{\mathbf{w}} \cdot \nabla_{\underline{\mathbf{x}}} u = f. \tag{20}$$

Conservation laws have the general form

$$\left. \frac{\partial u}{\partial t} \right|_{\underline{\xi}} + \nabla_{\underline{\mathbf{x}}} \cdot F(u) - \underline{\mathbf{w}} \cdot \nabla_{\underline{\mathbf{x}}} u = f, \tag{21}$$

and the associated integral form of (21) can be written as [5]

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{K_t} u\,\mathrm{d}x + \int_{K_t} \nabla_{\underline{\mathbf{x}}} \cdot (F(u) - \underline{\mathbf{w}})\,\mathrm{d}x = \int_{K_t} f\,\mathrm{d}x. \tag{22}$$

Integrating by parts, (22) may be recast in flux form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{K_t} u\,\mathrm{d}x + \int_{\partial K_t} (F(u) - \underline{\mathbf{w}}) \cdot \underline{\mathbf{n}}\,\mathrm{d}s = \int_{K_t} f\,\mathrm{d}x, \tag{23}$$

which is the form normally used in the context of FVM.

3.3 Regularity of the ALE Map

A well established method for solving ALE problems is to use some variant of a finite element discretisation. One concern in implementation is to make sure the regularity of the solution is conserved as the solution evolves and the domain under study moves. In [5] Formaggia and Nobile look at conservation laws in the setting of ALE. By assuming the solution $u(\underline{\mathbf{x}}, t)$ to be in $H^1(\Omega_t)$ for all times their goal is to establish sufficient conditions on the ALE mapping to assure this assumption. They achieve this by demanding that the initial domain Ω_0 be bounded with a Lipschitz continuous boundary and preserving these properties as it evolves in time under the ALE map, $\Omega_t = A_t(\Omega_0)$. By demanding A_t to be invertible on $\overline{\Omega}_0$ and for each time $A_t \in W^{1,\infty}(\Omega_0)$ and $A_t^{-1} \in W^{1,\infty}(\Omega_t)$, then $v \in H^1(\Omega_t)$ if and only if $\hat{v} = v \circ A_t \in H^1(\Omega_0)$. This result comes in handy when looking at the finite element solution of the ALE problem as it can establish a connection between the space of test functions on the current and the initial configurations. If K_0 is an element in the discretisation of Ω_0 and $K_t = A_t(K_0)$ its corresponding element in Ω_t, where $M_k^{K_0}$ and $M_k^{K_t}$ are their homeomorphic mapping from the reference element to the elements respectively. The subscript k is the order of the parametric mapping chosen for the finite element function space of the solution u of our PDE. Formaggia and Nobile conclude that they are able to find a discretisation of the ALE map as $A_{h,t}|_{K_0} = M_k^{K_t} \circ (M_k^{K_0})^{-1}$, ($h$ indicates the discrete version), belonging to an isoparametric function space of order k. Please refer to Figure 2 for illustration. Having established this fact we have to look for ways to actually move or deform the boundary and the interior of the domain. We will cover this topic later on.

3.4 Geometric Conservation Laws

When considering problems formulated in the ALE setting, the Geometric Conservation Law (GCL) becomes integral part of the problem. In the continuous formulation, the solution of the flow problem is independent of the domain movement. In the discrete case this is not necessarily true. When solving computational flow problems on moving domains we are faced with estimating geometric quantities involving grid velocity like for example numerical fluxes in finite volume schemes. To every numerical scheme there is a Discrete Geometric Conservation Law (DGCL) corresponding to the continuous one. The DGCL equips us with guidelines on how to evaluate these quantities, by enforcing that the numerical scheme preserve a uniform flow exactly. For the method to be consistent, this is a natural requirement.

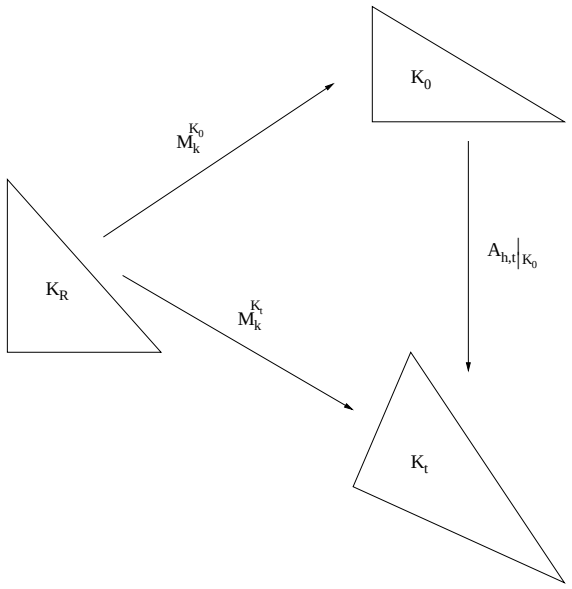


Figure 2: Discrete ALE map

The GCL can be derived from the weak form of the conservation law (23) in the absence of source terms by inserting the constant solution $u = u^*$ and noting that

$$\int_{\partial K_t} \underline{\mathbf{n}} \, \mathrm{d}s = 0. \tag{24}$$

Dividing by u^* we get identity

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{K_t} \mathrm{d}x = \int_{\partial K_t} \underline{\mathbf{w}} \cdot \underline{\mathbf{n}} \, \mathrm{d}s. \tag{25}$$

This is the GCL, and it basically states the change of volume of a cell K_t is equal to the volume swept over by the moving boundary. To find the DGCL we start with the same conservation law (23) (without the source term) and do a semi-discretisation either by a finite volume or a finite element method. The appropriate ODE solver is then applied to the resulting semi-discrete system. The DGCL for that specific solver is found by demanding the discrete scheme to exactly reproduce a constant solution. It is important to note that there is no unique DGCL, rather the DGCL is associated with a specific numerical procedure [6].

Guillard and Farhat [6] studied finite volume discretisation of the conservation law. They assume the boundaries e of the cells K to remain planar during deformation. Using consistent and conservative numerical fluxes they arrive at a DGCL of the form

$$\frac{\delta}{\delta t}|K| = \sum_{e \in \partial K} |e| \underline{\mathbf{w}}_e \cdot \underline{\mathbf{n}}_e. \tag{26}$$

in which $\delta/\delta t$ is a p'th order approximation of the time derivative. The size of a face is denoted $|e|$, $\underline{\mathbf{n}}_e$ and $\underline{\mathbf{w}}_e$ are time-averaged values of the normal vector and the velocity of the face respectively. The DGCL (26) resembles the continuous one, equation (25).

The DGCL includes position and velocity of the mesh. So we are not able to use an arbitrary routine to update the mesh if it violates the DGCL. The DGCL also provides

guidelines on how to estimate parameters as those on the right hand side of (26). Constructing a scheme which satisfies its DGCL has been shown to have several advantages.

Guillard and Farhat [6] showed that the moving mesh counterpart of a p-order time accurate fixed grid method is at least first-order accurate if it satisfies its DGCL. Satisfying the associated DGCL becomes increasingly important for higher order schemes. This result is also confirmed by experiments both in [6] and [7]. Koobus and Farhat [8] have later constructed second order time accurate schemes which obey their corresponding second order DGCL. Satisfying the DGCL will also have an impact on the stability of the numerical scheme. In [5] Formaggia and Nobile show that satisfying the corresponding first-order DGCL is sufficient for the backward Euler scheme to be unconditional stable. Nonlinear stability have been investigated by Farhat, Geuzaine and Grandmont [7]. They show that satisfying the DGCL is a necessary and sufficent condition for a numerical scheme to be nonlinearly stable in the sense of the maximum principle.

4 Adaptive Mesh Generation Methods

When solving either ODEs or PDEs numerically we need to discretise the domain of interest, and represent the solution by a set of discrete values. Classically this have been done using a fixed uniform grid on the computational domain. If one chooses to use a uniform grid the implementation will probably be of no great challenge, but in order to achieve an acceptable solution the mesh sometimes get too fine. There are several reasons for this. One obvious reason is that the geometry of the domain of the problem at hand may be so complex that you need a very fine mesh to represent it well. In a subset of the domain there might be features of the solution that demands a finer resolution than needed elsewhere. This is the case in the presence of large variations, such as shocks for example. Using an uniform mesh, we then need to refine the mesh for the whole domain, resulting in a large number of unknowns. Ideally we would like to arrive at an acceptable solution in the shortest possible time and cheapest fashion available. So the next obvious idea is to think of ways to create an adaptive mesh, with the fewest possible number of unknowns. There is a vast amount of literature on adaptive mesh methods. They are all based on various indicators to where adaption should occur. We choose to divide them into three main classes

1. r-refinement methods keeps the the total number of nodes constant and adjusts their positions in order to get the best possible approximation. These methods have been used to create optimal meshes for stationary problems or an optimal initial mesh for time-dependent problems. r-refinement have also been reported used for non-stationary problems during the time evolution, in particular for fluid-structure interaction.

2. h-refinement methods is a wide class of adaptive methods which can be subdivided into two groups. A common feature is that the mesh arrived at is a result of an iterative process. Given a mesh, a specific procedure is chosen to indicate where the mesh is too coarse or too fine. A new mesh is then produced upon the information from the previous mesh, and the process is continued until the specified specifications of the grid is met. The first variant is to keep an initial mesh as basis and either *subdivide* (refine) or coarsen the individual elements when needed, but the initial mesh is always kept as a basis. We may denote this approach as *element subdivision*.

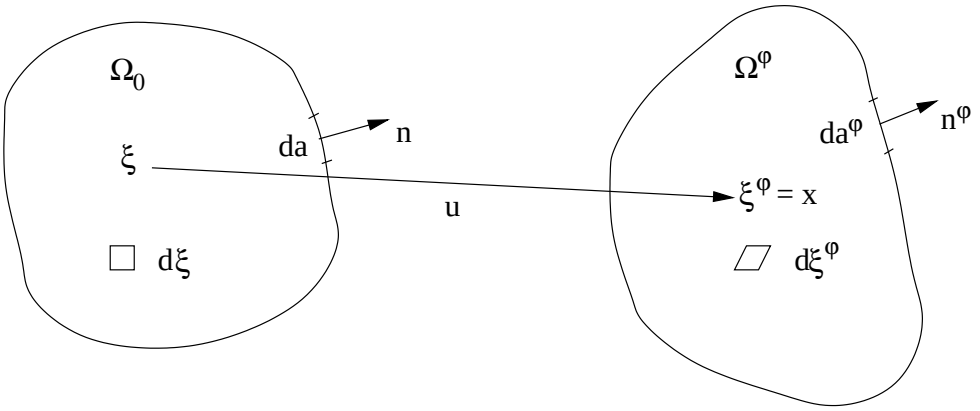


Figure 3: Deformation of a nonlinear elastic material

The second way to go is to completely *regenerate* the mesh, in which new element sizes is predicted for the whole domain and a totally new mesh is created upon this information. This approach usually results in fewer degrees of freedom compared to the first alternative. To completely regenerate the mesh is well suited for situations where severe element distortion may occur.

3. p-refinement is a method that keep the element size but instead increases the order of the polynomial used to represent the solution within each element. Again there is two alternative techniques. The first one increases the order of the polynomials uniformly for all elements on the whole domain. Secondly we could hierarchically increase the polynomial order locally.

Of course it is possible to combine two or more of the methods above to create even more specialised methods. An efficient hp-method is to first use h-refinement to arrive at a final mesh and then use p-refinement. In what follows we will concentrate on r- and h-refinement methods.

5 Elasticity Models for Mesh Adaption

When treating moving domains, a major concern is how to smoothly move the grid along, hopefully without distortions. Several techniques have been proposed for this task. One way of approaching the solution is to imagine the grid to be made up of some elastic material. Either by thinking of the grid as a discretisation of a continuous nonelastic material [9, 10], or by specifying restrictions on how the vertices can move mutually [11]. Either way, incorporating material properties into the mesh, allows movement or deformation according to certain material laws. Accompanied by a mathematical model for elasticity, this enables ensuring desirable properties of a well-behaved moving grid. For example folding can be avoided by forbidding self penetration in our model. Introductions to elasticity theory may be found in both [12] and [13].

5.1 Deformation of a Nonlinear Elastic Material

A given body made up of some material can be attacked by various body and surface forces. According to the material laws the position and shape of the body may change

due to the forces applied. One usually denotes this change as the action of the deformation $\varphi : \Omega_0 \longrightarrow \Omega^\varphi$ from the reference configuration Ω_0 to the current Ω^φ. The deformation φ is 1-to-1 and orientation preserving, that is $\det \nabla\varphi > 0$. An entity in Ω_0 is changed under the deformation map to its corresponding entity in Ω^φ indicated by the superscript φ. Two tensors play a main roles in the play of elasticity, namely the deformation gradient $F = \nabla\varphi$ and the right Cauchy-Green tensor $C = F^T F$. The right Cauchy-Green tensor is symmetric positive definite and have three invariants

$$I_1(C) = \operatorname{trace} C \tag{27}$$

$$I_2(C) = \frac{1}{2}((\operatorname{trace} C)^2 - \operatorname{trace}(C^2)) \tag{28}$$

$$I_3(C) = \det C. \tag{29}$$

The C tensor supply us with valuable information on how unit lengths $\partial\xi$, areas $n\,\mathrm{d}a$ and volumes $\mathrm{d}\xi$ changes under the deformation. Approximative values of unit length, area and volume after the deformation is respectively given as $|\partial\xi^\varphi|^2 = \partial\xi C \partial\xi$, $n^\varphi\,\mathrm{d}a^\varphi = (\det F)F^{-T}n\,\mathrm{d}a$ and $\mathrm{d}\xi^\varphi = \det F\,\mathrm{d}\xi$. See Figure 3 for reference. As pointed out in [10] the first invariant is related to length variation, the second to area changes and the third obviously to volume changes.

Under influence of external forces the material will always seek its state of equilibrium. The equilibrium equation stated in the reference configuration reads

$$\int_\Omega T : \nabla v\,\mathrm{d}\xi = \int_\Omega f \cdot v\,\mathrm{d}\xi + \int_\Gamma g \cdot v\,\mathrm{d}a, \quad \forall v : \bar{\Omega} \to \mathbb{E}, \tag{30}$$

where T is the stress tensor, f is the density of body forces, g is the density of surface traction and $\mathbb{E}$ is some Euclidean space. In the form presented above the equation does not present us with the full picture. We are still lacking one essential piece of information. What is the form of the stress tensor? Unless we know the connection between the kinematic quantities of the material and the stress tensor, we cannot accomplish much. This is were the constitutive laws enter. These are laws stating a connection between the stress tensor and for example the displacement field $u(\underline{\xi}) = \varphi(\underline{\xi}) - \underline{\xi}$ of the material. The construction of the constitutive laws are guided by experimental results, such that our mathematical model mimics real life. Various examples of constitutive laws are given in the literature [12, 13].

5.2 Mesh Adaption by Elastic Energy Minimisation

The stress tensor is usually linked to the internal elastic energy W of the material. For the case of hyper elastic materials the relationship is $T = \partial W/\partial F$. It can be shown that to minimise the total energy of the material is equivalent to deciding its state of equilibrium. This is a well known principle in physics [14].

5.2.1 An Indirect Approach

In [9] Tallec and Martin presents a method for constructing adaptive moving meshes in 2 dimensions, based on theory from continuum mechanics. They choose the physical domain Ω as their reference region, letting the deformation map τ map Ω into a square $\widehat{\Omega}$.

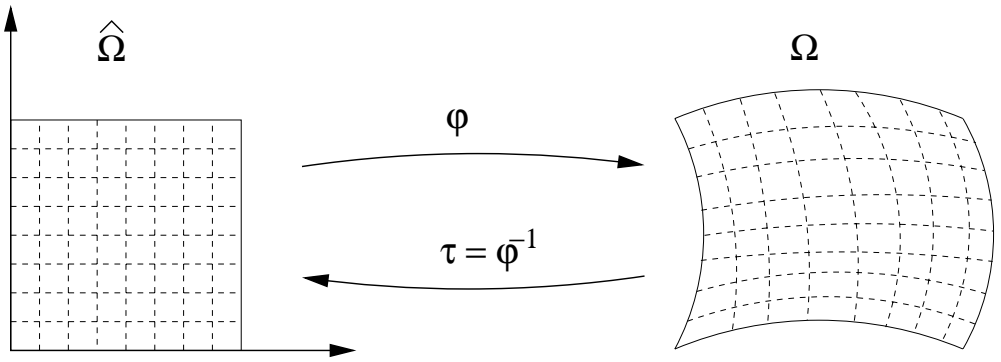


Figure 4: Elastic deformation

They construct a density function W depending on the position $\underline{\mathbf{x}}$ and the deformation map τ. Their ultimate goal is to minimise the total energy

$$\Phi(\tau) = \int_{\Omega} W(\underline{\mathbf{x}}, F)\,\mathrm{d}x$$

over the domain, for all admissible deformation maps. The elastic density function W measures the quality of the mesh and consists of four terms. They assume there the presence of some positive error estimate interpolated over the domain. Ideally one would like to adapt the mesh in such a way that the error is equally distributed over the hole domain. This results in a metric, which in a perfect world, the right Cauchy-Green tensor should be equal to. The first term of W measures the distance to this ideal deformation. The next three are penalty terms assuring positivity of W, one-to-one property of τ and the last term is included to avoid mesh distortions. Finally for unsteady calculation it is desirable to control the mesh deformation in time, so they restrict how fast the deformation gradient F may change. The resulting system may be solved using a combination of Dirichlet and Neumann boundary conditions. The Dirichlet conditions is used on the part of the boundary where one want to preserve the distribution of the points, while using Neumann conditions allow the points to slip along the boundary. The resulting variational problem is then solved using the Newton-Raphson's method. Constructing an adaptive mesh on the current configuration, Ω, is accomplished by mapping a uniform mesh on $\widehat{\Omega}$ back to Ω by the inverse of τ. This is a result of the particular choice of Ω a reference domain.

5.2.2 A More Direct Approach

A similar method to the one in the previous section is the one of Jacquotte [10]. Jacquotte define energy-like functions of the three invariants (27)–(29). To keep things consistent with our notation so far we will call these energy functions W as opposed to σ in [10]. By applying the axiom of frame indifference, the independence of orientation and the homogeneous nature of W, they argue that the energy function may be written $W = W(I_1, I_2, I_3)$. In order to ensure the existence and uniqueness of a minimum of the energy, W is imposed to be convex around the reference configuration. The choice of reference domain is now opposite to the one mentioned in the previous section. The deformation map φ is now thought to be a map from the square domain $\widehat{\Omega}$ to the current, and there is no need to invert this mapping. In this setting the energy W function measures the

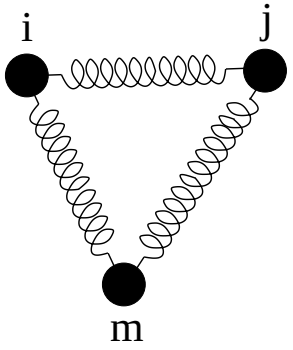


Figure 5: Imaginary springs between nodes

deformation of an optimal cubic reference element into the current cell. A proposed choice for the 3-dimensional case is

$$W = C_1(I_1 - I_3 - 2) + C_2(I_2 - 2I_3 - 1) + K(\det F - 1)^2, \quad K > 4(C_1 + C_2)/3 > 0.$$

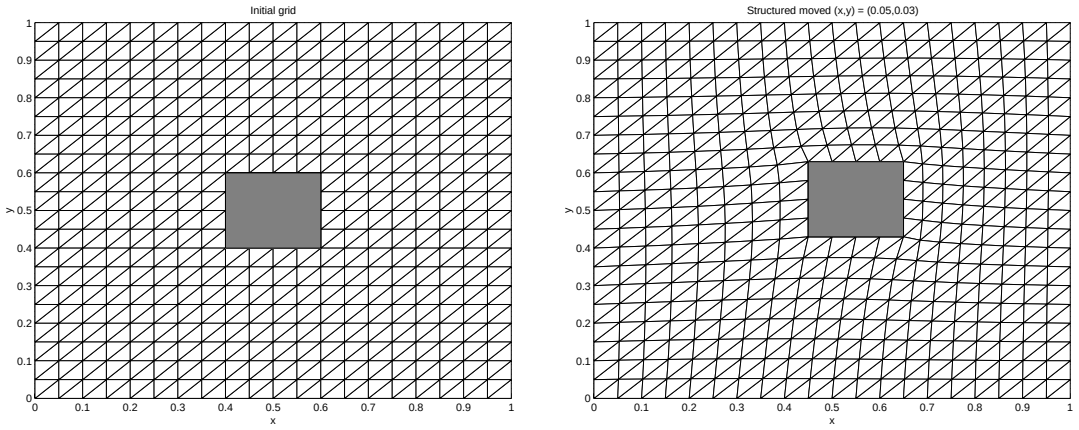
Using a finite element approximation we can construct local functions W^e of $I_i^e,\ i = 1, 2, 3$, which are polynomials of the nodal coordinates. Here e indicates an element and the total energy to be minimised is then $W = \sum_e W^e$. The Euler equations corresponding to the minimisation of W is then subjected to further study. To adapt the mesh according to a monitor function M, depending on some qualities of approximative solution of our PDE, we can replace $\det F$ with $M \det F$. This is a way to adjust the reference configuration for each cell. When it comes to boundary conditions for our mesh problem we have the freedom to choose a combination of fixing the nodes on adjacent sides and let them be free to move on the other sides.

5.3 Pseudostructural Mesh

Other ways of constructing moving grids is to to set restrictions on how the individual nodes of the grid may interact when moving the grid. This can be done for example by introducing functions measuring local smoothness and orthogonality of the grid depending on the relative position of the nodal points [10] and minimising this measure. Another way is to introduce ficticious springs between the nodes of the grid, as illustrated in Figure 5 for the 2-dimensional case. This idea was introduced by Batina in [11]. The spring stiffness for a given edge is taken to be the inverse of the length of that edge. So referring to Figure 5 the stiffness of the spring in between nodes i and j is

$$k_{ij} = \frac{1}{\|\underline{\mathbf{x}}^i - \underline{\mathbf{x}}^j\|}, \tag{31}$$

where $\underline{\mathbf{x}}^i$ indicates the coordinates of the i'th node and $\|\cdot\|$ is the usual Euclidean norm in $\mathbb{R}^2$. The method can be further improved by introducing strings for the diagonals of a quadrilateral element, or for a triangular element introduce a string from the midpoint of its longest edge to its opposit node. This enables better control of the deformation of the elements [15]. Approaches as explained above have been used on fluid-structure problems [11, 16] with unstructured grids. Grid point are held fixed on the outer boundary of the mesh and the instantaneous locations of the point on the inner boundary, around the body, are prescribed by the movement of the structure. At each step the static equilibrium



(a) Initial grid

(b) Moved grid

Figure 6: Movement of solid square

equation is solved iteratively by a predictor corrector method. Let $\underline{\delta}_i^n$ be the displacement vector of node i at time step n. Let $\underline{\tilde{\delta}}_i$ be the predicted movement of node i for the next step in time. Then the mesh updating routine reads:

- $\underline{\tilde{\delta}}_i = 2\underline{\delta}_i^n - \underline{\delta}_i^{n-1}$
- The displacements are corrected using several Jacobi iterations of the static equilibrium using

$$\underline{\delta}_i^{n+1} = \frac{\sum_{j \in V(i)} k_{ij} \underline{\tilde{\delta}}_j}{\sum_{j \in V(i)} k_{ij}},$$

  where $V(i)$ is the set of neighbour nodes to node i.
- And finally $\underline{\mathbf{x}}_i^{n+1} = \underline{\mathbf{x}}_i^n + \underline{\delta}_i^{n+1}$

What makes this method appealing is its simple construction. Given that you know the displacement of the interior grid points from the surface motion/deformation of the structure then the displacement of the other grid points can be found in usual very few iterations. A simple example for a structured grid around a solid square is illustrated in Figure 6. In the first picture we see the initial grid around a solid square. In the second picture the square is moved slightly to the right and upward and the grids adjusts to the displacement. For a fluid-structure computation one would ideally have a much better refinement of the grid nearby the structure. This will add to the quality of the solution but also avoid a too great deformation of the grid close to the structure. The better the refinement the smaller the distance in between the neighbour nodes gets and thus the stiffness of the imaginary springs get larger and will change less. This kind of pseudostructural grid has been used in fluid structure computations where the structure movement is pressure driven. The solution process of the fluid-structure problem is then described in three basic steps:

a) Advance the structural part of the system under a given pressure load

b) Adjust the fluid mesh accordingly

c) Advance the fluid system and compute a new pressure load

5.4 Harmonic Extension of Boundary Movement

Often in practice we are given the initial configuration Ω_0 and the evolution of its boundary $g : \partial\Omega_0 \times T \to \partial\Omega_t$. A simple method for finding the mesh movement in the domain from elastic dynamics was proposed in [5]. The idea is to harmonically expand the evolution of the boundary onto the whole of Ω_0.

$$\frac{\partial \underline{\mathbf{x}}}{\partial t} - \nabla_{\underline{\xi}} \cdot (\kappa \nabla_{\underline{\xi}} \underline{\mathbf{x}}) = 0, \quad \underline{\xi} \in \Omega_0, \ t \in I \tag{32}$$

$$\underline{\mathbf{x}}(\underline{\xi}, 0) = \underline{\xi}, \quad \underline{\xi} \in \Omega_0 \tag{33}$$

$$\underline{\mathbf{x}}(\underline{\xi}, 0) = g(\underline{\xi}, t), \quad \underline{\xi} \in \partial\Omega_0, \ t \in I \tag{34}$$

In practice we need only know the ALE map at discrete time levels. If we know the evolution of the boundary for to a given time, $h : \partial\Omega_0 \to \partial\Omega_T$, we are left with solving

$$\nabla_{\underline{\xi}} \cdot (\kappa \nabla_{\underline{\xi}} \underline{\mathbf{x}}) = 0, \quad \underline{\xi} \in \Omega_0 \tag{35}$$

$$\underline{\mathbf{x}}(\underline{\xi}, 0) = h(\underline{\xi}), \quad \underline{\xi} \in \partial\Omega_0, \tag{36}$$

for each time step. Here κ is a positive constant. Letting κ be a tensor function depending on the numerical solution of the problem at hand, may allow an adaptive scheme as well. Since the mesh is moved anyway, adaptivity can be achieved at very low additional cost.

6 Moving Grid Methods

6.1 Grid Deformation

The deformation method is a fairly new method for generating adaptive grids to both stationary and time-dependent problems. The deformation method looks at the overall problem and produces a coordinate mapping of the whole domain. As for elasticity problems the deformation method determines a deformation map, but rather than using knowledge from continuum mechanics, the method is based on differential geometry. It moves the nodal points of the grid in such a way that the resulting grid adapts to a prescribed physical monitor function.

The idea behind the method dates back to the results of Moser [17] and Moser and Dacorogna [18]. Assuming that the Jacobian determinant equals a strictly positive monitor function, they prove the existence of a coordinate map, ϕ, which adapts the cell volume of the resulting grid to the monitor. This result is used by Liao and coworkers to create an adaptive method based on grid movement. Through several articles [19, 20, 21, 22, 23, 24] the method was first presented for the stationary case and then later developed in the time-dependent case.

The methods are both directly based on the proofs and follow the same path towards the final coordinate mapping. The main result for the stationary case, as given in [20] and [21] is the following theorem 6.1.

THEOREM 6.1
Let Ω be a bounded open set in $\mathbb{R}^n$, B be a ball in $\mathbb{R}^n$. Suppose there exists $\phi : \Omega \to B$, an orientation preserving C^{k+3} diffeomorphism, $k \geq 1$. Let $f > 0$ be in $C^k(\bar{\Omega})$ and

$\int_\Omega 1/f(x)\,\mathrm{d}x = |\Omega|$. *Then there exists* $u \in \mathit{Diff}^k(\bar{\Omega})$ *satisfying*

$$\begin{aligned} \det \nabla u(x) &= f(u(x)), \quad x \in \Omega, \\ u(x) &= x, \quad x \in \partial\Omega. \end{aligned} \tag{37}$$

The proof of Liao and Su is based on the observation that if $w(x)$ solves the problem

$$\begin{aligned} \det \nabla w(x) &= f(x), \quad x \in \Omega \\ w(x) &= x, \quad x \in \partial\Omega, \end{aligned} \tag{38}$$

then $u(x) = w^{-1}(x)$ solves (37). This follows from the identity $w(u(x)) = x$. The problem in (38) is the one dealt with by Dacorogna and Moser in [18]. This is the form of the deformation method as it was presented in [19]. In this case the Jacobian determinant is prescribed at the old coordinates. The method was later modified to prescribe the Jacobian determinant of the coordinate mapping at the new coordinates. The proof of theorem 6.1 boils down to proving the existence of $w(x)$ in [18]. The proof is composed of two steps, and follows basically the one by Dacorogna and Moser.

1. Find $v \in C^{k+1}(\Omega, \mathbb{R}^n)$ such that v solves

$$\operatorname{div} v(x) = g(x) - 1, \quad x \in \Omega, \qquad v(x) = 0, \quad \in \partial\Omega \tag{39}$$

2. Fix $x \in \bar{\Omega}$ and solve the ODE

$$\frac{\partial}{\partial t}\phi(x,t) = \eta(\phi(x,t)), \quad t > 0, \qquad \phi(x,0) = x, \tag{40}$$

where $\eta : \bar{\Omega} \to \mathbb{R}^n$ is the deformation field

$$\eta(y) = \frac{v(y)}{t + (1-t)g(y)}, \quad \text{where } g = 1/f. \tag{41}$$

The solution $w(x) = \phi(x, 1)$ of (40) at $t = 1$ will then solve (38). This follows by proving that the quantity $H(x,t) = [t + (1-t)g(\phi(x,t))] \det \nabla\phi(x,t)$ is independent of time.

As mentioned, viewing the mapping ϕ as a coordinate map, it is clear that a volume element under the mapping ϕ will be deformed in such a way that its volume adapts to the positive monitor function. The restriction that $f > 0$ and that the mapping is bijective on the boundary of the domain ensures that the coordinate mapping is bijective on the whole domain. In theory, grid crossing should not occur. The discussion of this is included in [24] and rests on use of the rank theorem or the inverse function theorem.

To use the deformation method to create an adaptive grid one needs to prescribe a monitor function that will concentrate on the particular properties of the solution that one wants to zoom in on. The choice of monitor function may depend on the type of problem at hand. In [25] the monitor function is chosen to adapt to large pressure gradients, so their choice is

$$\frac{1}{f} = C_1(1 + C_2\nabla p).$$

The time dependent version of the grid deformation method is not very different from the stationary one. The idea is to construct a time-dependent coordinate mapping ψ, whose Jacobian determinant is given as a time varying monitor function $f(x,t)$. The deformation

method for non-stationary problems was presented in [23, 22]. Given a monitor function $f(x,t)$,

$$f(x,0) = 1, \quad x \in \Omega \tag{42}$$

$$\int_\Omega \left(\frac{1}{f} - 1\right) \mathrm{d}\Omega = 0, \quad t \in [0,T] \tag{43}$$

the time dependent deformation method constructs a mapping $\psi : \Omega_c \to \Omega_t$ such that

$$\begin{aligned} \det \nabla\psi(x,t) &= f(\psi(x,t),t), \quad x \in \Omega \\ \psi(x,t) &\in \partial\Omega, \quad x \in \partial\Omega \\ \psi(x,0) &= x, \quad x \in \Omega \end{aligned} \tag{44}$$

Ω_c is here the reference domain, and Ω_t the current configuration. Again the proof consists of two steps, with only slight modifications to the stationary version.

1. Find a vector field v such that

$$\begin{aligned} \operatorname{div} v(x) &= -\frac{\partial}{\partial t}\frac{1}{f(x,t)}, \quad x \in \Omega,\ t \in [0,T] \\ (v,n) &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{45}$$

   where n is the outward normal to the boundary $\partial\Omega$.

2. Then

$$\frac{\partial}{\partial t}\psi(x,t) = \eta(\psi(x,t),t) = v(\psi(x,t),t) f(\psi(x,t),t), \quad 0 \le t \le T, \tag{46}$$

$$\psi(x,0) = x, \tag{47}$$

Finally one has to prove the time-independence of $H(x,t) = g(\psi(x,t),t) \det \nabla\psi(x,t)$.

Liao and coworkers have also used this point of view to generate grids on surfaces and also implemented moving boundaries and grid morphing in [26].

When it comes to the implementation of the deformation method, there are several challenging issues to be faced. In a numerical experiment one is forced to construct the the monitor function based entirely on the properties of the numerical solution at hand. The monitor function should reflect the need for grid refinement. The monitor function is the link between the need and the actual implementation of the grid adaption. Several candidates have been proposed in the literature. These choices are listed in [24]:

$$\begin{aligned} f(x,t) &= \frac{C}{1 + \alpha_1 |\nabla u|^2 + \alpha_2 |u|^2} \\ f(x,t) &= \frac{C}{\sqrt{1 + \alpha |\nabla u|^2}} \\ f(x,t) &= \frac{C}{1 + |u|}. \end{aligned}$$

Another possibility is, given a positive error estimate $\delta(x,t)$ of the solution, one may choose $f(x,t) = C/\delta(x,t)$. Regardless your choice, the initial (42) and normalisation (43) conditions must be obeyed.

6.1.1 Solution Strategies and Implementation Issues

In the first step of both the static and time-dependent versions we have to find a velocity field, with known divergence. This is clearly an under determined system because a velocity field may be uniquely decomposed into a divergence free and a curl free part. In order to find a unique velocity field Liao and coworkers set $\operatorname{curl} u = 0$. The resulting system is a div-curl problem

$$\operatorname{div} u(x,t) = -\frac{\partial}{\partial t}\frac{1}{f(x,t)}, \quad \text{in } \Omega \tag{48}$$

$$\operatorname{curl} u = 0, \quad \text{in } \Omega \tag{49}$$

$$u \cdot n = 0, \quad \text{on } \partial\Omega. \tag{50}$$

There are several ways to solve this problem. In [27] Cao, Huang and Russell develop a moving grid method. They end up solving the same type of problem, in fact the deformation method is a special case of their method. One alternative is to minimise the functional

$$I[u] = \int_\Omega |\operatorname{div} u - f|^2 + |\operatorname{curl} u|^2 \, \mathrm{d}x, \tag{51}$$

and another is to solve the Euler-Lagrange equations of the functional directly. A third alternative is to use equation (49) which implies that $u = \operatorname{grad} w$. This results in the Poisson problem with homogeneous Neumann conditions.

$$\nabla^2 w = 0, \quad \text{in } \Omega \tag{52}$$

$$\frac{\partial w}{\partial n} = 0, \quad \text{on } \partial\Omega. \tag{53}$$

Getting the boundary conditions correct is a serious challenge. In the last alternative mentioned the problem is to get a continuous velocity field v. Solving the Poisson problem using a finite element method and setting $v = \operatorname{grad} w$ will not result in a continuous field. A smoothing algorithm is needed as well. Issues of the implementation of the various alternatives are discussed in [24] and [27].

Equipped with a velocity field $v(x,t)$ we have to integrate the system (46) – (47). Liao and coworkers claim that any ODE solver may be used. For example an explicit Runge-Kutta method. Still there are obstacles to overcome. Using an s-stage Runge-Kutta method, we have to evaluate the velocity field at s intermediate points. This means we have to interpolate the velocity field.

One of the advantages of the deformation method is that it is not restricted by the dimension of the problem. In theory it should produce an adaptive moving mesh, which has no grid crossing. One drawback of the method is that it does not produce orthogonal grids, and the velocity field is not necessarily rotation free. This may result in a grid which is too skewed and, thus, inappropriate for the application of the finite element method.

6.2 Moving Mesh Partial Differential Equation

The Moving Mesh Partial Differential Equation (MMPDE) method for dynamic meshes was first introduced by Huang et. al. in [28, 29]. It computes nodal positions alongside the physical solution in a tightly coupled fashion.

The method comes in various shapes depending on the dimensionality of the physical problem. In the simplest case of one space dimension, the MMPDEs are based on the

"equidistribution principle" which is also applied in the moving finite difference method by Dorfi and Drury described in [30]. This form is presented in more detail below. Early development of the method did, however, not show extendibility of this approach to problems of higher physical dimensionality. Nevertheless, the MMPDE method has, through other generalisations, been successfully applied to 2D problems in fluid mechanics such as air-foil analysis and laminar flame propagation.

6.2.1 The Equidistribution Principle

The "equidistribution principle" (EP) is, simply put, the placement of nodes, x_i, in a one-dimensional physical region, Ω_p, in order to distribute a positive monitor function, $M = M(x,t)$, evenly across the various sub intervals, $[x_i, x_{i+1}]$. Furthermore Ω_p can, through a suitable scaling of the variables, be assumed to be equal to the interval $[0,1]$. Stated in its integral form, the EP reads

$$\int_0^{x(\xi,t)} M(\tilde{x},t)\,\mathrm{d}\tilde{x} = \xi\theta(t) \tag{54}$$

in which $\theta(t) = \int_0^1 M(\tilde{x},t)\,\mathrm{d}\tilde{x}$ and $x(\xi,t)$ maps the computational coordinate $\xi \in \Omega_c = [0,1]$ at time t into physical coordinates. The effect of equation (54) is to place the physical variable x in Ω_p such that a fraction ξ of the total integral of the monitor function is accounted for. If the computational coordinate, ξ, is discretised by a uniform grid, eg. $\xi_i = i/n, i = 0,\ldots,n$, then between two consecutive nodes in the physical domain, x_i and x_{i+1}, we will have

$$\int_{x(\xi_i,t)}^{x(\xi_{i+1},t)} M(\tilde{x},t)\,\mathrm{d}\tilde{x} = \frac{1}{n}\theta(t).$$

In other words, the total monitor integral is distributed equally among the n sub intervals.

6.2.2 The MMPDE Method

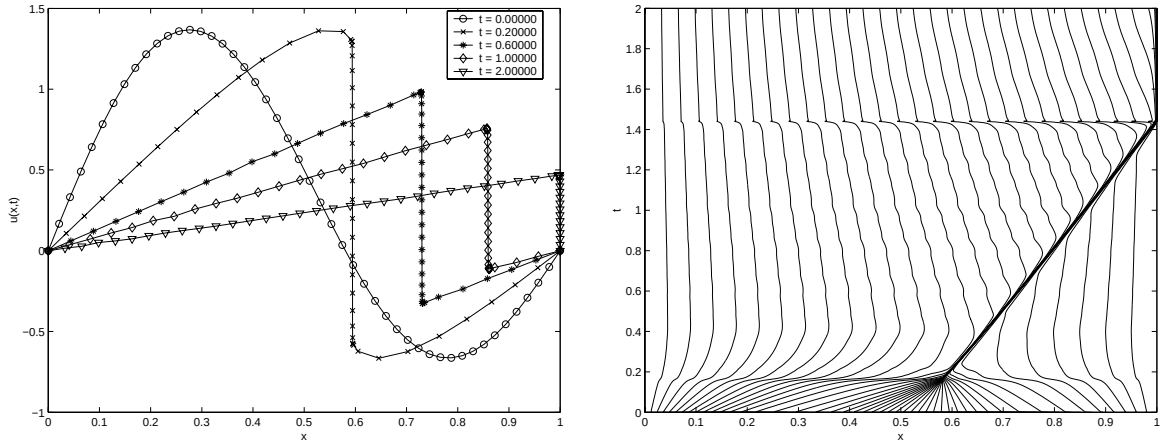
The EP in (54) can be restated in two different differential forms by differentiating with respect to ξ to give what Huang et. al. denote as "quasi-static EPs" (QSEPs)

$$M(x(\xi,t),t)\frac{\partial}{\partial\xi}x(\xi,t) = \theta(t) \tag{55}$$

and

$$\frac{\partial}{\partial\xi}\Big\{M(x(\xi,t),t)\frac{\partial}{\partial\xi}x(\xi,t)\Big\} = 0. \tag{56}$$

The concept of QSEPs is then used to formulate the MMPDEs by differentiation with respect to time and [28] describes seven different MMPDEs with varying properties and with or without temporal regularisation. The regularisation is used to decrease stiffness of the resulting ODE system in much the same way as is often seen in numerical solution of Differential Algebraic Equations (DAEs). The effect of temporal regularisation is to make the system less sensitive to changing properties of the physical solution because exact equidistribution is not required at every t. Without the regularisation the Jacobian matrix may become singular, which indicates a defective coordinate transformation. Further analysis of MMPDEs is available in [31].



(a) Numerical solution, $u(x,t)$

(b) Grid trajectories, $x(t)$

Figure 7: Numerical solution to Burgers' equation and associated grid trajectories, MMPDE case.

The MMPDEs are all continuous with respect to time and space and thus need to be discretised in order to establish a complete discrete system to be solved in conjunction with the physical problem. The discretisation most commonly suggested in the literature is a semi-discretisation in space using finite differences according to the method of lines (MOL) approach. This results in a system of ordinary differential equations (ODEs) which is usually stiff. If the MOL is applied to the physical problem as well, the end result is a system of ODEs of dimension $2n$ with n being the number of nodes. This system is ordinarily integrated using a stiff ODE solver.

Figure 7 displays the numerical solution, and the induced grid movement, to the one-dimensional boundary value problem composed of the Burgers' equation

$$u_t = \varepsilon u_{xx} - (\tfrac{1}{2}u^2)_x, \; x \in [0,1] \tag{57}$$

and the following boundary and initial conditions

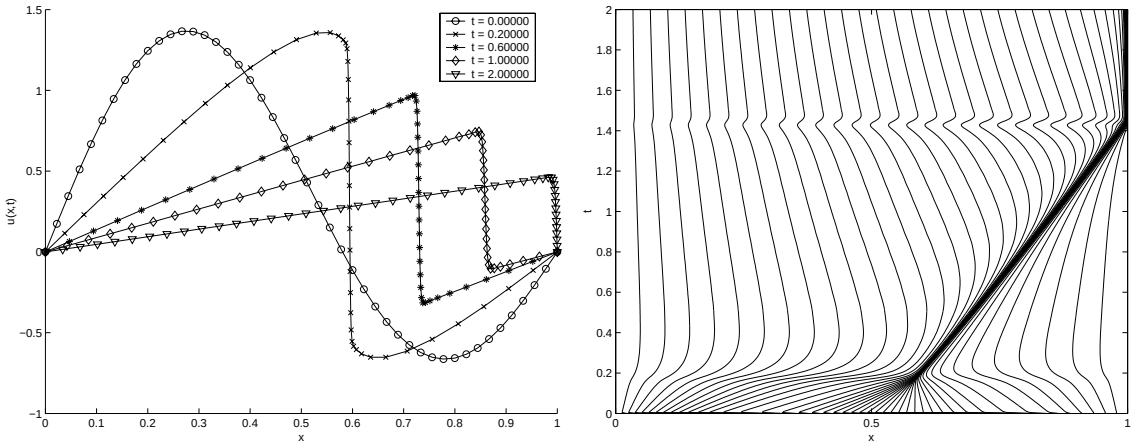
$$u(0,t) = u(1,t) = 0 \tag{58}$$

$$u(x,0) = \sin(2\pi x) + \tfrac{1}{2}\sin(\pi x) \tag{59}$$

solved using the moving grid method derived from discretising the MMPDE known as MMPDE6 in [28]

$$\frac{\partial^2 \dot{x}}{\partial \xi^2} = -\frac{1}{\tau}\frac{\partial}{\partial \xi}\big(M\frac{\partial x}{\partial \xi}\big). \tag{60}$$

The temporal regularisation parameter used in this test was $\tau = 10^{-3}$ and the plots were produced from a slightly modified version of The MathWorks' `BURGERSODE` demonstration code for `ODE15S` in MATLAB R12. We note in particular that the grid points concentrate around the shock and that grid distortions reflect from the right as the shock reaches the right wall at approximately $t = 1.42$. In comparison, figure 8 displays numerical solution and grid trajectories computed by the moving finite difference method of Dorfi



(a) Numerical solution, $u(x,t)$

(b) Grid trajectories, $x(t)$

Figure 8: Numerical solution to Burgers' equation and associated grid trajectories, moving finite difference case.

and Drury. The overall behaviour is the same as in the MMPDE case of figure 7, but the grid distortion as the shock reaches the right end of the interval is less abrupt.

An additional possibility in moving mesh methods based on MMPDEs is the incorporation of spatial smoothing. Several filters have been introduced and their properties in terms of resulting mesh movement, computational complexity and likelihood of nodal crossings are somewhat varied. A consistent conclusion, however, is that spatial smoothing, at the expense of introducing additional tunable parameters into the methods, greatly lessens sensitivity to swiftly varying or non-physical features of the physical problem. This produces more smoothly transforming meshes and controls "wiggles" sometimes seen in non-regularised methods based on MMPDEs. On the other hand, too much spatial smoothing will produce mesh movement lagging behind the important physical features, yielding a method unsuited to resolve those features.

6.2.3 Monitor Functions

Monitor functions are, in general, problem dependent. The choice of the monitor function, M, has been a matter of some debate. One approach is to choose the function based on a set of important features of the physical problem and how well these features need to be resolved in the numerical solution. Such a monitor might, for instance, attempt to place a larger number of nodes within a steep front of the solution and thus view large solution gradients and, perhaps in addition, second derivatives, as a significant metric.

A common monitor function of this kind is the "arc length" monitor

$$M(x,t) = \sqrt{1 + u_x^2(x,t)}\,, \tag{61}$$

which, u being the solution to the physical problem and $u_x = \partial u/\partial x$, aims to distribute nodes in such a way that the arc length of u in each sub interval is more or less uniform. Computational experience has shown this to be a satisfactory monitor function in many cases, but other choices are indeed possible. These may include monitors based on derived quantities, such as stresses in an elastic material.

Another possibility, not directly related to properties of the physical problem, is to choose the monitor function to be an error estimate or other error indicators of the numerical solution. Regions of high error will the be more resolved in this approach.

A conclusive study of implications of the monitor function has not been published at the time of this writing, but [32] offers an analysis of some of the monitor functions commonly used. It specifically aims at the case of two space dimensions and has a brief discussion of the three-dimensional case.

6.2.4 Extension to Several Space Dimensions

In multiple space dimensions, the situation is not as clear cut as in one space dimension. There is, as yet, no satisfactory generalisation of the one-dimensional EP to d spatial dimensions and the possibility of, for instance, nodal crossing increases when going above one space dimension. In [33], however, the authors, based upon work by Winslow and Brown, develop high dimensional MMPDEs using other monitor functions.

The basic idea is to construct the coordinate mapping, $\underline{\mathbf{x}} = \underline{\mathbf{x}}(\underline{\xi}, t)$, with $\underline{\xi} = (\xi_1, \ldots, \xi_d) \in \Omega_c = [0,1]^d$ being the computational coordinate and $\underline{\mathbf{x}} = (x_1, \ldots, x_d)$ being the coordinate in physical space, $\Omega_p \in \mathbb{R}^d$, such that a mesh adaption functional is minimised.

The mesh adaption functional

$$I[\underline{\xi}] = \tfrac{1}{2} \int_{\Omega_p} \sum_i (\nabla \xi_i)^T G_i^{-1} \nabla \xi_i \, \mathrm{d}\underline{\mathbf{x}} = \tfrac{1}{2} \int_{\Omega_p} \sum_i (\underline{\mathbf{a}}_i)^T G_i^{-1} \underline{\mathbf{a}}_i \, \mathrm{d}\underline{\mathbf{x}}, \tag{62}$$

is often used in deriving the MMPDEs in multiple space dimensions. We denote by $\underline{\mathbf{a}}^i = \nabla \xi_i$, and $\nabla = (\partial/\partial x_1, \ldots, \partial/\partial x_d)$. The matrices $G_i, i = 1, \ldots, d$ are assumed symmetric and positive definite (SPD) and are the multi dimensional equivalents of the simple monitor functions of section 6.2.3. Applying Hamilton's principle to minimise the functional in (62) yields the Euler-Lagrange equations

$$\frac{\delta I[\underline{\xi}]}{\delta \xi_i} = 0 \;\Leftrightarrow\; \nabla \cdot (G_i^{-1} \underline{\mathbf{a}}_i) = 0, \; i = 1, \ldots, d \tag{63}$$

which determine the coordinate transformation $\underline{\xi} = \underline{\xi}(\underline{\mathbf{x}}, t)$, used in static mesh generation and as a basis for dynamic adaption. The MMPDEs derived from (63) use the same theoretical basis as the ones in the original papers [28, 29], and read

$$\frac{\partial \xi_i}{\partial t} = \frac{1}{\tau} \nabla \cdot (G_i^{-1} \underline{\mathbf{a}}_i), \tag{64}$$

with $\tau > 0$ being a user defined parameter adjusting the time scale, and thus responsiveness, of the mesh equation.

Alternatively, it is possible to compute the coordinate mapping $\underline{\mathbf{x}} = \underline{\mathbf{x}}(\underline{\xi}, t)$ directly by inverting the rôle of dependent and independent variables in (63) and this is an approach often used in actual computations. The resulting MMPDEs are highly non-linear and read as

$$\frac{\partial \underline{\mathbf{x}}}{\partial t} + \sum_{i,l} \frac{\underline{\mathbf{x}}_{\xi_i}}{\tau J} \frac{\partial}{\partial \xi_l} \Big\{ \frac{1}{J g_i} [(\underline{\mathbf{x}}_{\xi_j})^T G_i \underline{\mathbf{x}}_{\xi_m} \cdot (\underline{\mathbf{x}}_{\xi_k})^T G_i \underline{\mathbf{x}}_{\xi_n} - (\underline{\mathbf{x}}_{\xi_j})^T G_i \underline{\mathbf{x}}_{\xi_n} \cdot (\underline{\mathbf{x}}_{\xi_k})^T G_i \underline{\mathbf{x}}_{\xi_m}] \Big\} = 0, \tag{65}$$

with (i,j,k) and (l,m,n) varying cyclically. Here $g_i = \det G_i$, $(\underline{\mathbf{x}}_{\xi_i})_r = \partial x_r / \partial \xi_i, r = 1, \ldots, n$ and $J = \det \partial \underline{\mathbf{x}} / \partial \underline{\xi}$. Huang et. al. [33] have additional details and give specific

simplifications for the two-dimensional case. It furthermore describes a method, known as SEAF (Spatial-Eigenvalue Approximate Factorisation), to accommodate efficient numerical integration of the discretised version of (65). Other notes on formulation and implementation of moving mesh methods based on MMPDEs may be found in [34], which also, briefly, discusses how spatial smoothing may be applied to the monitor function.

A couple of monitor functions are listed, notably the arc length-like

$$G_1 = G_2 = \frac{1}{\sqrt{1 + \|\nabla u\|_2^2}} \big(I + \nabla u (\nabla u)^T\big) \tag{66}$$

for the 2D case and the slightly more involved

$$G_1 = G_2 = G_3 = \frac{1}{\sqrt{g}}\, G \tag{67}$$

$$G(\underline{\mathbf{x}}) = I + f(\underline{\mathbf{x}}, F(\underline{\mathbf{x}})) \frac{\nabla F(\underline{\mathbf{x}}) \big(\nabla F(\underline{\mathbf{x}})\big)^T}{\|\nabla F(\underline{\mathbf{x}})\|_2^2} \tag{68}$$

in three dimensional cases. Here f depends on the distance from a point $\underline{\mathbf{x}} \in \Omega_p$ to the surface $F(\underline{\mathbf{x}}) = 0$. It is assumed that f increases as this distance decreases. There are however other choices possible and [35] studies a monitor function based on a posteriori error estimates. This work is complemented by a comparison of various a posteriori error estimate monitor functions in [36].

Both of these papers use spatial smoothing. The smoothing is accomplished by applying a low-pass filter m times to the discrete, monitor function. The low-pass filter removes some of the most sensitive modes, thus normally making the method more stable. In particular, the low-pass filtering procedure is defined as follows

$$G^{(\mu+1)}(\underline{\mathbf{x}}_p, t) = \frac{\int_{C(\underline{\xi}_p)} G^{(\mu)}(\underline{\mathbf{x}}(\underline{\xi}_p, t))\, \mathrm{d}\underline{\xi}}{\int_{C(\underline{\xi}_p)} \mathrm{d}\underline{\xi}},\ \mu = 0, \ldots, m-1$$

in which $\underline{\mathbf{x}}_p$ is a mesh point in Ω_p and $\underline{\xi}_p$ is its corresponding mesh point in Ω_c. The integration domain $C(\underline{\xi}_p) \subset \Omega_c$ is the union of neighbouring mesh cells having $\underline{\xi}_p$ as one of its vertices. The monitor function $G^{(0)}$ is the one computed from the monitor definition in for instance (66) or (67)–(68).

6.3 Moving Finite Elements (MFE)

The moving finite element method was first introduced by Miller and Miller [37, 38] as an extension to the ordinary finite element method. The method may be viewed as an r-adaptive method similar to the moving finite differences of Dorfi and Drury [30] and the moving mesh partial differential equations of Huang et. al. [28, 29], with a particular focus on finite element solution to PDEs of the form

$$u_t = \mathcal{L}u, \quad (x, t) \in \Omega \times [t_0, T). \tag{69}$$

The differential operator $\mathcal{L}$ is in general nonlinear and dependent upon spatial derivatives of the physical solution, u, in addition to u itself. It may additionally include source terms

The goal of all adaptive methods is to facilitate the computation of numerical solutions to (69) more cheaply than static or non-adaptive methods. As such, the moving finite element method aims to achieve a given accuracy, and, possibly, stability requirement by relocating nodal points to regions of the computational domain where they are most needed.

6.3.1 Standard Finite Elements for Time Dependent Problems

This section gives a short review of a basic technique for solving time dependent problems, such as (69), using standard finite elements. The approach taken here is that of spatial semi-discretisation or the method of lines which produces a system of stiff ordinary differential equations to be subsequently discretised in time. Other approaches include first discretising in time by some well-known ODE solver, eg. multi-step methods such as BDF or other temporal finite differencing, and then solving the resulting elliptic differential equation on each time step by finite elements. For details on the latter approach one might consult [39, 4, 40] or similar references.

A point of departure when formulating a numerical procedure for (69) is to restate the problem in variational form and minimising the residual error

$$\mathscr{R}(w) = \int_\Omega (w_t - \mathcal{L}w)^2 \,\mathrm{d}\Omega$$

where $w \in U$ and U is the space of admissible functions. Elements of U have to be sufficiently differentiable and satisfy the essential boundary conditions. In other words: Find $u_t \in U$ such that

$$u_t = \arg\min_{w \in U} \mathscr{R}(w). \tag{70}$$

As $\mathscr{R}(w) = (w_t - \mathcal{L}w, w_t - \mathcal{L}w)_2 = \|w_t - \mathcal{L}w\|_2^2$, this is by Pythagoras' theorem equivalent to requiring that the variational error, $u_t - \mathcal{L}u$, be orthogonal to the space U. Thus problem (70) is mathematically equivalent to the following orthogonality conditions: Find $u_t \in U$ such that

$$(u_t - \mathcal{L}u, v)_2 = 0 \tag{71}$$

for all v in U and all $t \in [t_0, T)$.

Standard finite element theory solves this problem by introducing a simplified, finite dimensional space, $U_h \subset U$. It is assumed that U_h is endowed with a basis $\{\phi_i\}_{i=1}^n$ which fully describes the resulting numerical method. The space of piecewise linear functions is often used in practice, but higher order polynomial basis functions may be applied equally well. The numerical solution is then defined as a linear combination of these basis functions with coefficients depending on time. In particular, the numerical solution is often stated as

$$u_h(x,t) = \sum_{j=1}^n u_{h,j}(t)\phi_j(x) \tag{72}$$

and thus

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^n \frac{\partial u_{h,j}}{\partial t}(t)\,\phi_j(x). \tag{73}$$

We wish to compute the numerical solution, u_h, which is closest to the exact solution, u. Closeness is here measured by the L^2-norm of the error, $\|u - u_h\|_2$, and we define the following minimisation problem: Find $\partial u_h/\partial t \in U_h$ such that

$$\frac{\partial u_h}{\partial t} = \arg\min_{w \in U_h} \|u_t - w_t\|_2.$$

Appealing once more to the Pythagorean theorem, this means $u_t - \frac{\partial u_h}{\partial t} \perp U_h$.

Inserting the numerical solution (72) into the variational problem (71) and requiring that the numerical error $u - u_h$ be orthogonal to U_h leads to the requirement that

$$(\tfrac{\partial u_h}{\partial t} - \mathcal{L}u_h, v)_2 = 0 \tag{74}$$

for all v in U_h. This is a less stringent requirement than the full variational formulation (71). On the other hand it produces a discrete system to which a solution is computable by numerical methods.

Using $v = \phi_i, i = 1, \ldots, n$ as test functions in (74) the following system of ordinary differential equations (ODEs) is derived

$$\mathbf{M}_h \dot{\underline{\mathbf{u}}}_h = \mathbf{A}_h \underline{\mathbf{u}}_h \tag{75}$$

in which $(M_h)_{ij} = \int_\Omega \phi_j \phi_i \, \mathrm{d}\Omega$, $(A_h)_{ij} = \int_\Omega \phi_i \mathcal{L}\phi_j \, \mathrm{d}\Omega$ and $(\underline{\mathbf{u}}_h)_i = u_{h,i}$. As the various entries in M_h are defined through symmetric inner products, M_h is symmetric and positive definite (SPD). $\mathbf{M}_h$ and $\mathbf{A}_h$ are respectively denoted as the "mass matrix" and the "stiffness matrix" in the literature.

6.3.2 Finite Elements with Moving Basis Functions

The basis functions, ϕ_j, in the ordinary FE method all depend implicitly on the static nodal locations $\underline{\mathbf{X}} \in \mathcal{T}$ with $\mathcal{T}$ being the set of all meshes on Ω. The moving finite element method extends the notion of a basis function to allow the nodal locations to dynamically move in time. In the following we will make the implicit dependence on nodal locations explicit and denote by $\phi_j(x, \underline{\mathbf{X}})$ the j'th basis function evaluated at $x \in \Omega$ and defined by the current mesh. Most of the MFE theory presented here is contained in the monograph [41] to which we refer for further details.

Inserting this into the numerical solution (72), we obtain the extended numerical solution

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{n} \frac{\partial u_{h,j}}{\partial t} \phi_j + \sum_{j=1}^{n} u_{h,j} \frac{\partial \phi_j}{\partial t}. \tag{76}$$

Let X_k be the k'th node in the mesh $\underline{\mathbf{X}}$ with k ranging from 1 to n. Then the final term of (76) through the chain rule becomes

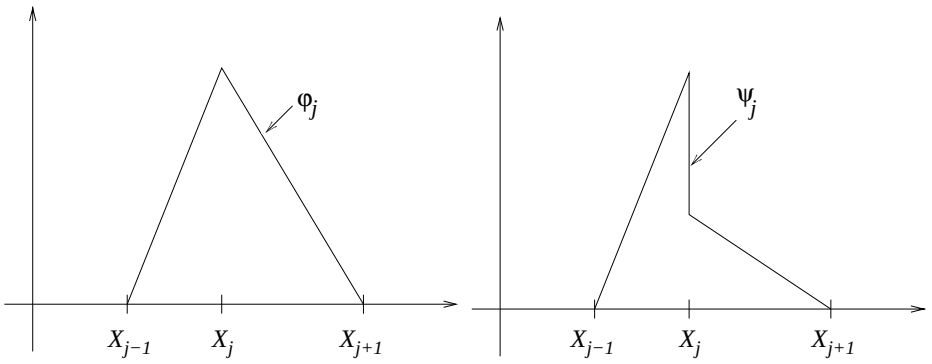
$$\sum_{j=1}^{n} u_{h,j} \sum_{k=1}^{n} (\nabla_{X_k} \phi_j)^T \frac{\partial X_k}{\partial t} = \sum_{k=1}^{n} \Big[\sum_{j=1}^{n} u_{h,j} (\nabla_{X_k} \phi_j)^T \Big] \frac{\partial X_k}{\partial t}.$$

Introducing "second-type basis functions", ψ_j, defined as $\psi_j = \sum_{k=1}^{n} u_{h,k} \nabla_{X_j} \phi_k = \nabla_{X_j} u_h$, (76) may be rewritten as

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{n} \big[\frac{\partial u_{h,j}}{\partial t} \phi_j + \psi_j^T \frac{\partial X_j}{\partial t} \big]. \tag{77}$$

In the case of ϕ_j being piecewise linear, continuous functions, ψ_j are piecewise linear and, in general, discontinuous functions. A drawing of ϕ_j and ψ_j in this special case may be found in figure 9.

The nodal velocities, $\partial X_j / \partial t$, are, as yet, undetermined and the basic mechanism by which the MFE determines these velocities is derived similarly to static finite elements.



(a) first-type basis function, ϕ_j (b) second-type basis function, ψ_j

Figure 9: First and second type basis functions, $1D$, linear ϕ_j

We wish to minimise the residual error, $\mathscr{R}(u_h)$, but now taking into account that the minimisation may be done over an extended space. This leads to the variational formulation: Find $\frac{\partial u_h}{\partial t} \in U$, $\frac{\partial \mathbf{X}}{\partial t} \in \mathcal{T}$ such that

$$(\tfrac{\partial u_h}{\partial t} - \mathcal{L}u_h, \phi_i)_2 = 0 \tag{78}$$

$$(\tfrac{\partial u_h}{\partial t} - \mathcal{L}u_h, \psi_i)_2 = 0 \tag{79}$$

for all $i = 1, \ldots, n$. Inner products are taken component wise when there is more than one component. Inserting the numerical solution (77) into the system (78)–(79) yields the following system of ODEs for all i:

$$\sum_{j=1}^{n} (\phi_i, \phi_j)_2 \dot{u}_{h,j} + \sum_{j=1}^{n} (\phi_i, \psi_j^T)_2 \dot{X}_j = (\phi_i, \mathcal{L}u_h)_2 \tag{80}$$

$$\sum_{j=1}^{n} (\psi_i, \phi_j)_2 \dot{u}_{h,j} + \sum_{j=1}^{n} (\psi_i, \psi_j^T)_2 \dot{X}_j = (\psi_i, \mathcal{L}u_h)_2. \tag{81}$$

Denoting by $\underline{\mathbf{Y}}$ the vector composed by the n sub-vectors

$$Y_i = \begin{pmatrix} u_{h,i} \\ X_i \end{pmatrix}, \ i = 1, \ldots, n,$$

the system of ODEs may be written in matrix form as $\mathbf{A}_h(\underline{\mathbf{Y}})\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}_h(\underline{\mathbf{Y}})$. The matrix $\mathbf{A}_h$ is, in the case of a scalar equation (69) in d space dimensions, an $n \times n$ block matrix in which each of the $(d+1) \times (d+1)$ blocks are

$$(A_h)_{ij} = \begin{pmatrix} (\phi_i, \phi_j)_2 & (\phi_i, \psi_j^T)_2 \\ (\psi_i, \phi_j)_2 & (\psi_i, \psi_j^T)_2 \end{pmatrix}. \tag{82}$$

The right hand side vector $\underline{\mathbf{F}}_h$ is constructed similarly to $\underline{\mathbf{Y}}$ in that

$$(F_h)_i = \begin{pmatrix} (\phi_i, \mathcal{L}u_h)_2 \\ (\psi_i, \mathcal{L}u_h)_2 \end{pmatrix}. \tag{83}$$

6.3.3 Gradient Weighted Moving Finite Elements

A weighted form of the variational formulation (78)–(79) is often recommended, in particular when the method is overly sensitive to specific features of the physical problem (eg. steep fronts). Such weighting replaces the inner products $(f, g)_2$ in equations (82)–(83) by inner products with respect to a given, positive, weight function $w : \Omega \to \mathbb{R}^+$ defined as

$$(f, g)_w = \int_\Omega f(x)g(x)w(x)\,\mathrm{d}\Omega. \tag{84}$$

One of the most studied such weighting functions give rise to what is known as the "gradient weighted" moving finite element (GWMFE) method. In this case w is defined as

$$w(x) = \frac{1}{\sqrt{1 + \|\nabla u\|_2^2}} \tag{85}$$

with u denoting the solution to (69) and ∇u the gradient with respect to the physical variable x. This weight function lessens the emphasis on regions of high solution gradients which means that the effect of minimisation in the steep parts is reduced. The effect is to limit the method's sensitivity to high gradients which, although an important feature, may introduce more stiffness into the mesh movement differential equation.

In addition to this direct effect, the GWMFE method has a more geometric interpretation than the "bare" MFE method and this is reflected in practical implementations of the method. Carlson and Miller have published papers [42, 43] which detail design and application of GWMFE codes based on this interpretation. The basic idea is to transform the equation locally by dividing both sides of (69) by the gradient weighting function (85) and recognising the term $u_t/\sqrt{1 + \|\nabla u\|_2^2}$ as the normal velocity of the graph of the function u. Thus the physical equation becomes

$$\dot{n} = \mathcal{K}u \tag{86}$$

in which $\mathcal{K}u = (\mathcal{L}u)/\sqrt{1 + \|\nabla u\|_2^2}$.

The consequences of this reformulation may not be immediately apparent, but when u is allowed to be an evolving, oriented manifold in Euclidean space some simplification are more easily realised using this version. In particular, embedding the nodal positions X_j into the numerical definition of the manifold as $\underline{\mathbf{u}}_{h,j} = (X_j, u_{h,j}), j = 1, \ldots, n$, we obtain the finite element discretisation

$$\underline{\mathbf{u}}_h = \sum_{j=1}^{n} \underline{\mathbf{u}}_{h,j}\phi_j. \tag{87}$$

The basis functions ϕ_j do not depend on time in this version of the MFE method, and this obviates the use of the second type basis functions, ψ_j, because the motion $\dot{\underline{\mathbf{u}}}_h$ may be expressed as

$$\dot{\underline{\mathbf{u}}}_h = \sum_{j=1}^{n} \dot{\underline{\mathbf{u}}}_{h,j}\phi_j.$$

Furthermore, the normal motion, $\dot{n}$, is given as $\dot{n} = \dot{\underline{\mathbf{u}}} \cdot \underline{\mathbf{n}}$ in which $\underline{\mathbf{n}}$ is the unit normal vector of the manifold.

The discrete equations derived from this method may be based on either a minimisation statement applied to the residual error or a local force balance at each node. Regardless of derivation, the equations are nevertheless expressed as

$$\int_S (\dot{\underline{\mathbf{u}}}_h \cdot \underline{\mathbf{n}})\,\underline{\mathbf{n}}\,\phi_i\,\mathrm{d}S = \int_S \mathcal{K}u\,\underline{\mathbf{n}}\,\phi_i\,\mathrm{d}S \tag{88}$$

for each node. The integrals are evaluated over the GWMFE manifold, denoted here by S. Equation (88) leads to a system of $n \cdot (d+1)$ non-linear ordinary differential equations which is usually stiff and may be written similarly to the MFE case as $\mathbf{A}_h(\underline{\mathbf{Y}})\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}_h(\underline{\mathbf{Y}})$.

When extending this method to systems of PDEs, Carlson and Miller use a single, shared mesh to represent the computational nodes for each differential equation. The rationale behind this is simplicity of implementation because the data structures involved in keeping track of which nodes belong to which mesh in a multi-mesh implementation quickly become overly complex. They also claim that the resulting system of ODEs is non-smooth when a node in one grid becomes part of a different cell or element in another grid which has severe implications for the ODE solver. The shared mesh implementation adds extra unknowns to each node and extends the local force balance to use a weighted sum of forces stemming from each of the single PDEs.

6.3.4 Regularisation of MFE and GWMFE

It should be noted that $\mathbf{A}_h$ of (82) has a non-linear dependence on $\underline{\mathbf{Y}}$ even if the PDE (69) is linear. This makes numerical solution of $A_h(\underline{\mathbf{Y}})\dot{\underline{\mathbf{Y}}} = F_h(\underline{\mathbf{Y}})$ more difficult than in the static FE case. The latter requires the solution of (75) in which the positive definite mass matrix M_h can be computed once and perhaps even factored to provide a more easily computable system. On the other hand A_h of the MFE method must be recomputed at least once every time step in an unmodified ODE procedure. Furthermore, it is only positive semi-definite which introduces additional difficulties to the solution process. Finally, the sub-matrices $(A_h)_{ij}$ may become singular if nodes coalesce.

To overcome some of the problems with respect to semi-definiteness and singularities, various types of regularisation have been proposed. A specific type of regularisation is defined by adding penalty functions to the error residual $\mathscr{R}$. In particular, adding an inter-nodal viscosity term penalising relative nodal motion (terms of the form $\dot{X}_k - \dot{X}_{k-1}$) has proved helpful. Such kinds of regularisation were deemed necessary even in the introductory reports by Miller and Miller, and the specific one-dimensional error residual was there listed as

$$\mathscr{R}(w) = \|w - \mathcal{L}w\|_2^2 + \sum_{j=2}^{n} \left[\varepsilon(\dot{X}_j - \dot{X}_{j-1})\right]^2. \tag{89}$$

This is, however, not the only regularisation scheme which has been applied to MFE.

The situation is similar in the GWMFE case. Since the method only determines how the nodes move in the direction normal to the GWMFE manifold, there is no inherent mechanism for describing motion tangential to the manifold. This causes degeneracy of the mass matrix in situations where the normal vectors of adjacent elements do no span the whole of Euclidean space. Then there exists one or more directions in which nodal movement causes no normal movement of the GWMFE manifold.

A solution proposed by Carlson and Miller in [42, 43] is to add extra inter-nodal viscosity of the form

$$\sum_{j=1}^{n} c_{ij}\dot{\underline{\mathbf{u}}}_{h,j}, \quad c_{ij} = \varepsilon \int_S \nabla\phi_i \cdot \nabla\phi_j \,\mathrm{d}S \tag{90}$$

with $\varepsilon > 0$ being a small parameter chosen in the interval $[0.25, 25] \cdot tol^2$ for a specific numerical tolerance, tol. Thus, the system $\mathbf{A}_h(\underline{\mathbf{Y}})\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}_h(\underline{\mathbf{Y}})$ becomes

$$(\mathbf{A}_h(\underline{\mathbf{Y}}) + \mathbf{C})\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}_h(\underline{\mathbf{Y}})$$

with the added bonus of the regularisation becoming negligible away from critical regions. These regions are characterised by the solution varying less than *tol* between adjacent nodes which means that the solution is numerically planar.

A further refinement to the regularisation scheme in the GWMFE case is the introduction of inter-nodal tension. This is normally not needed, but Carlson and Miller report improvement of the results for problems which are integrated over long times.

7 h-Adaptive Methods

h-refinement methods change the mesh iteratively either by regenerating or refining the mesh at each step, until the user specification are met. The most usual way of accomplishing this is to demand an error estimate or some indicator of interest to be equally distributed among the elements.

7.1 Predicting Optimal Element Size

In order for solution adaptive finite element analysis to be effective, it is important that the adapted mesh be optimal—or close to optimal—in some sense. Intuitively, an optimal mesh should guarantee that the error be less than or equal to a user prescribed tolerance while at the same time minimise the cost of obtaining the numerical solution. In [44], Coorevits, Ladeveze and Pelle propose and analyse a procedure for automating the finite element analysis of two-dimensional elasticity problems. The method is derived from a specific finite element error estimate, but can be extended to any error estimate provided the global error can be obtained through a sum of local error contributions. Such error estimates include all methods based on aggregate integrals which can be decomposed into elemental integrals on the finite element mesh. This subsection will, however, follow the general discussion of [44].

Based on a conservative elemental estimate for the error, a new mesh is constructed to cater for error in regular regions of the computational domain as well as regions of singularity. Assuming that a kinematically admissible displacement field, $\hat{U}$, and a statically admissible stress field, $\hat{\sigma}$, have been computed from finite element analysis of an elasticity model, the quantity

$$\hat{e} = \hat{\sigma} - K\epsilon(\hat{U})$$

is called the error in the constitutive relation. K is the Hooke tensor which relates the strain, ϵ, to the stress, σ, through the displacement, U, such that $\sigma = K\epsilon(U)$. $\hat{e}$ is zero whenever $(\hat{U}, \hat{\sigma})$ is an exact solution to the elasticity problem, but otherwise

$$\mathbf{e} = \|\hat{e}\|_\Omega = \|\hat{\sigma} - K\epsilon(\hat{U})\|_\Omega \tag{91}$$

may be used to measure the error of the numerical solution. The norm used here is the problem specific energy norm

$$\|v\|_\Omega = \Big[\int_\Omega v^T K^{-1} v \,\mathrm{d}\Omega\Big]^{1/2},$$

and using this definition, the relative error, ε is defined as

$$\varepsilon = \frac{\|\hat{\sigma} - K\epsilon(\hat{U})\|_\Omega}{\|\hat{\sigma} + K\epsilon(\hat{U})\|_\Omega}.$$

Within a smaller sub-region $T \subset \Omega$, we have the local contribution to this estimate defined as

$$\varepsilon_T = \frac{\|\hat{\sigma} - K\epsilon(\hat{U})\|_T}{\|\hat{\sigma} + K\epsilon(\hat{U})\|_\Omega},$$

from which we note that $\varepsilon^2 = \sum_T \varepsilon_T^2$. The authors relate this error estimate to other estimates proposed in the literature, and furthermore show that it is conservative in the sense that the true error is always less than or equal to $\mathbf{e}$.

Two conditions which have been used in assessing optimality of meshes, use slightly different definitions of optimality.

1. A mesh, $\mathcal{T}_h^*$, is *optimal* with respect to an error estimate ε, if
   - $\varepsilon^* = \varepsilon_0$, ε_0 user prescribed error tolerance.
   - The number, N^*, of elements of $\mathcal{T}_h^*$ is minimal.
2. A mesh, $\mathcal{T}_h^*$, is *optimal* with respect to an error estimate ε, if
   - $\varepsilon^* = \varepsilon_0$, ε_0 user prescribed error tolerance.
   - ε^* is uniformly distributed across $\mathcal{T}_h^*$.

The condition used in [44] is the first one, and the authors show that 1 and 2 are equivalent for problems with regular solutions for which the obtained numerical accuracy is determined by the spatial convergence rate of the finite element method only. This is, however, not the case when singularities are present in the solution.

The equivalence for regular solutions is proved by assuming that the finite element method has a convergence rate of q such that $\varepsilon = \mathcal{O}(h^q)$ when the mesh size h tends to zero. Then the local error ε_T of mesh $\mathcal{T}_h$ is related to the local error ε_T^* of mesh $\mathcal{T}_h^*$ as

$$\frac{\varepsilon_T^*}{\varepsilon_T} = \left[\frac{h_T^*}{h_T}\right]^q = r_T^q \tag{92}$$

where $r_T = h_T^*/h_T$ is the refinement coefficient for element T. h_T is the current mesh size for element T whereas h_T^* is the mesh size which must be imposed on T in order to assure mesh optimality.

Using (92), the square of the error of mesh $\mathcal{T}_h^*$ is evaluated as

$$\sum_T (\varepsilon_T^*)^2 = \sum_T r_T^{2q} \varepsilon_T^2$$

and, as r_T is the refinement factor of the mesh size in each spatial dimension of the element T, the total number, N^*, of elements in the optimal mesh $\mathcal{T}_h^*$ is

$$N^* = \sum_T \frac{1}{r_T^2}. \tag{93}$$

Thus, determining the optimal mesh $\mathcal{T}_h^*$ is equivalent to solving the constrained optimisation problem

$$\begin{cases} \text{minimise } N^* = \displaystyle\sum_T \frac{1}{r_T^2}, \quad \text{subject to} \\ \displaystyle\sum_T r_T^{2q} \varepsilon_T^2 = \varepsilon_0^2 \end{cases} \tag{94}$$

for which the explicit solution

$$r_T = \frac{\varepsilon_0^{1/q}}{\varepsilon_T^{1/q}\left[\sum_T \varepsilon_T^{2/(q+1)}\right]^{1/(2q)}},$$

may be obtained by the method of Lagrange multipliers. This solution may then be inserted into (92) to show that the contribution to the total error from an element T^* of $\mathcal{T}_h^*$ is

$$\varepsilon_{T^*}^2 = r_T^2(\varepsilon_T^*)^2 = r_T^{2q+2}\varepsilon_T^2 = \frac{\varepsilon_0^{2+2/q}}{\left[\sum_T \varepsilon_T^{2/(q+1)}\right]^{1+1/q}}$$

which is uniform across all elements of $\mathcal{T}_h^*$.

On the other hand, as stated earlier, whenever the solution has singularities this result is not true. The authors then discuss what needs to be taken into account to cater for the non-regular solution. Singularities degrade the convergence rate depending on their strength. This leads to a revised local relative error estimate of

$$\varepsilon_T = Ch^{p_T}$$

with p_T denoting the local convergence rate for the element T. p_T is determined by numerical estimates for elements connected to singular nodes and simply imposed as the order of the shape functions in all other elements. Singular nodes, defined as mesh nodes for which the numerical solution becomes singular, are detected through an automatic procedure based on element area amplification of the local errors and a theoretical model for convergence at singularities.

In this case the optimal mesh is defined through the optimisation problem

$$\begin{cases} \text{minimise } N^* = \displaystyle\sum_T \frac{1}{r_T^2}, \quad \text{subject to} \\ \displaystyle\sum_T r_T^{2p_T}\varepsilon_T^2 = \varepsilon_0 \end{cases} \tag{95}$$

which is similar to (94) except that the constraint is extended to varying convergence rates throughout the mesh. Solving (95) by means of a Lagrange multiplier A which has to satisfy

$$\sum_T (Ap_T)^{-p_T/(p_T+1)}\varepsilon_T^{2/(p_T+1)} = \varepsilon_0^2,$$

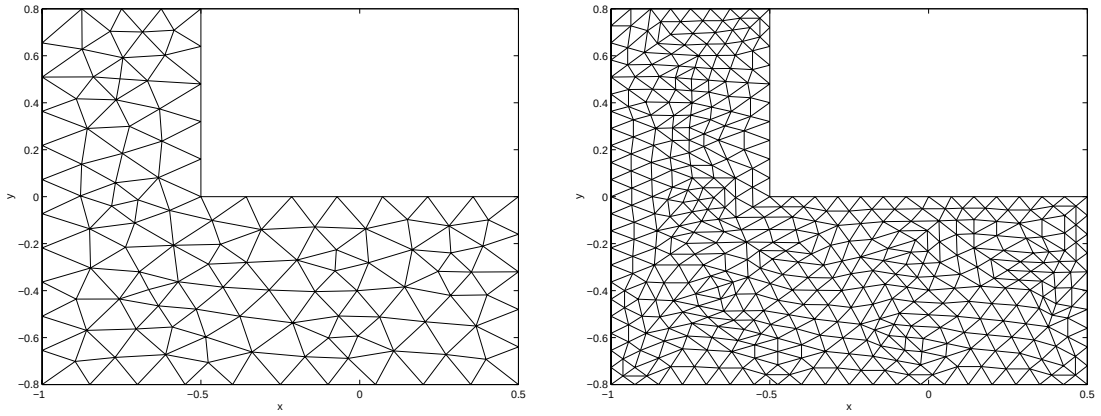
the refinement coefficients can be computed as

$$r_T = \frac{1}{(Ap_T\varepsilon_T^2)^{2(p_T+1)}}. \tag{96}$$

This leads ultimately to deduction of the local error contribution from an element T^* of the optimal mesh $\mathcal{T}_h^*$ as

$$\varepsilon_{T^*}^2 = r_T^2(\varepsilon_T^*)^2 = r_T^{2p_T+2}\varepsilon_T^2 = \frac{1}{Ap_T}$$

which is not uniform throughout $\mathcal{T}_h^*$. The products $\sqrt{p_T}\varepsilon_T$ are, however, the same throughout the whole of $\mathcal{T}_h^*$.



(a) Initial grid

(b) Refined grid

Figure 10: Uniformely refined grid

7.2 Mesh Generation

Several grid procedures for grid generation have been developed . To begin with the generators could only generate grids with triangular elements. These generators were based on the frontal method. Later generators for quadrilateral elements (in 2-dimension) and for tetrahedral and hexahedral elements (in 3-dimension) have been developed. There also exist methods based on the Delaunay triangulation.

7.3 Mesh Refinement Techniques

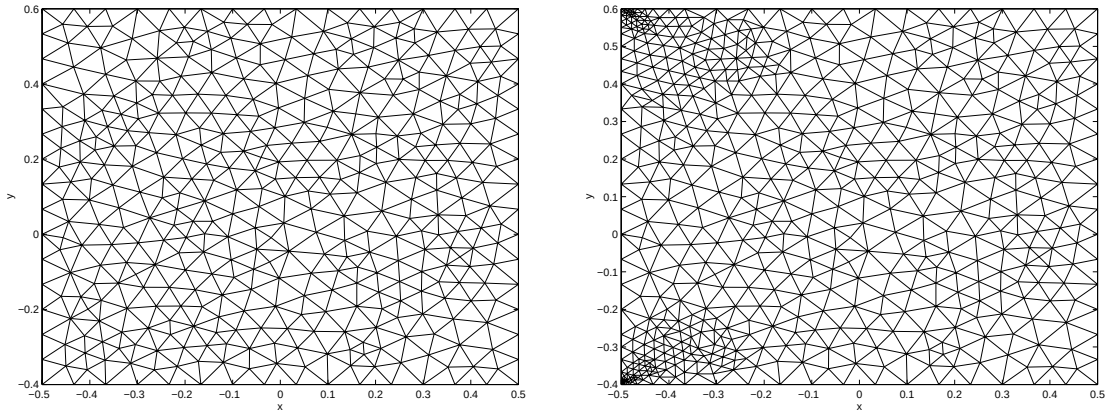
Instead of regenerating the mesh one may keep the mesh and refine it by subdividing its elements. Generally this will result in a greater number of degrees of freedom compared to a complete regeneration. For some time-dependent cases the transition in the solution is not that great, so a re-meshing might not be necessary, and refinement may only be needed in small areas. In this case a pure mesh refinement technique may be preferable.

Let us denote our finite element mesh by its collection of elements $\mathcal{T}_h$. If our goal is to distribute the estimated error evenly among the elements there are two strategies used.

1. Let $T \in \mathcal{T}_h$ be an element of the mesh, and let τ be obtained by subdividing T. If we assume the error within each element T is ch_T^p we may decide c and p with our error estimate for T and τ, and this enables us to predict the error of further subdividing the element.

2. Among all the elements let us pick the one with the greatest error and denote $\eta = \max_{T \in \mathcal{T}_h} \eta_T$. An easy technique is to refine those elements where $\eta_T \geq \gamma \cdot \eta$ for some $0 < \gamma < 1$ chosen buy the user.

When using a reasonable mesh generator it will make sure the elements are regular in shape and well suited for finite element computation. It is important to maintain this shape regularity during the refinement process. A much used strategy for the 2-dimensional case is to refine an element by joining the midpoints of the edges. Figure 10 illustrates this method for a uniform refinement technique. From a computational point of view it is



(a) Initial grid

(b) Refined grid

Figure 11: Adaptive refinement

cheaper to refine only in the areas most needed. Figure 11 shows the initial and adapted grid for a plain strain problem. A material is attached to a rigid wall at its left hand side. A force, directed downwards and slightly to the right, is then applied to its top face. Two singularities are then created at the corners of the left-hand side of the object. The final grid is refined only locally to these singularities. If we subdivide an element we introduce hanging nodes on its edges. We are then faced with additional problem. Should we accept hanging nodes or must the mesh be regular. If the latter is the case, we are forced to introduce additional rules for subdividing the neighbour elements as well. An overview of the techniques for triangular elements and their extension to the 3-dimensional case may be found in [45].

8 Error Estimates

In an ideal world we would be able to find the exact solution of any problem. Unfortunately, experience tell us this is not the case, so we have to look for approximate solutions "good enough" for our use. If we denote the exact solution of a problem by $\underline{\mathbf{u}}$ and the finite element approximation by $\underline{\mathbf{u}}_h$ the error is defined

$$\underline{\mathbf{e}} = \underline{\mathbf{u}} - \underline{\mathbf{u}}_h. \tag{97}$$

In general the quality of an approximate solution is not equal at all points in space, especially near singularities, so we usually consider some appropriate norm, $\|e\|$, of the error. To control the quality of our approximate solution we demand the error norm to be less than some specified limit. Obviously, if we knew the exact solution there would be no need of an approximation, so in general we are forced to look for ways to estimate the error. In some applications only special qualities of the solution are emphasised. In these cases we may use appropriate *indicators* to measure the solution quality instead. This section is dedicated to various posteriori error estimation methods.

8.1 Recovery Methods

The idea of recovery methods is to post process the approximate solution and in return get an even better approximation. We will cover two such techniques, namely *superconvergent patch recovery* (SPR) and *recovery by equilibration of patches* (REP). The two methods recover an improved stress field. In elasticity problems the error in the stress field may be a good indicator for adaptivity. In this subsection we will direct our attention to the following strain-displacement equations, the elastic stress-strain law and the equations of equilibrium

$$\underline{\epsilon} = \mathbb{D}\underline{\mathbf{u}} \quad \text{in } \Omega \tag{98}$$

$$\underline{\sigma} = \mathbf{C}\underline{\epsilon} \quad \text{in } \Omega \tag{99}$$

$$\mathbb{D}^T\underline{\sigma} + \underline{\mathbf{b}} = 0 \quad \text{in } \Omega. \tag{100}$$

In two dimensions, the above are ([46])

$$\underline{\sigma} = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]^T \quad \text{the vector of stress components}$$

$$\underline{\epsilon} = [\epsilon_{xx}, \epsilon_{yy}, 2\epsilon_{xy}]^T \quad \text{the vector of strain components}$$

$$\underline{\mathbf{u}} = [u, v]^T \quad \text{displacement vector}$$

$$\underline{\mathbf{b}} = [b_x, b_y]^T \quad \text{body forces}$$

$$\mathbb{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad \text{strain-displacement operator}$$

$$\mathbf{C} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} \quad \text{stress-strain matrix.}$$

E is Young's modulus and ν is Poisson's ratio. To complete the problem, we prescribe displacements and tractions on non-overlapping parts of the boundary.

$$\underline{\mathbf{u}} = \tilde{\underline{\mathbf{u}}} \quad \text{on } \partial\Omega_u \tag{101}$$

$$\mathbf{Q}\underline{\sigma} = \tilde{\underline{\mathbf{t}}} \quad \text{on } \partial\Omega_t, \tag{102}$$

where $\partial\Omega = \partial\Omega_u \cup \partial\Omega_t$. The Cauchy transformation matrix, $\mathbf{Q}$, is given as

$$\mathbf{Q} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix},$$

where $\underline{\mathbf{n}} = [n_x, n_y]^T$ is the outward unit normal at the boundary. In a variational formulation, used in FEM, the system of equations (98)–(100) is solved in terms of the displacement $\underline{\mathbf{u}}$. The result is a continuous displacement field, but the recovered stress field, $\underline{\sigma}_h = \mathbf{C}\mathbb{D}\underline{\mathbf{u}}_h$, is in general discontinuous across element boundaries. Our goal for this subsection is thus to present ways of improving the stress field.

8.1.1 Superconvergent Patch Recovery

For regular quadrilateral meshes, it is a well known fact that the solution is best approximated at the elements nodes, while solution gradients are best approximated at points inside the elements. These points correspond to the Gauss points used in obtaining the

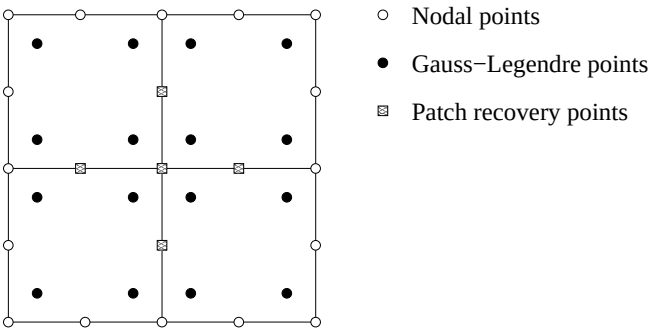


Figure 12: A patch of four 8-node elements

finite element solution itself. At these points the gradients have a convergence rate one order higher than elsewhere, and the points are called *superconvergent*. With the knowledge of superconvergent points it is only natural to take advantages of these to recover better gradients. One way of improving the solution is to extrapolate new nodal values from the superconvergent points. Another very common method is to consider a patch of elements and interpolate or "smooth" the superconvergent values within the patch by a p'th order polynomial. This is the nature of *superconvergent patch recovery* (SPR). The number of sampling points in the recovery process can be taken higher than the number of parameters in the polynomial. This way one can obtain a smoothed stress field which is superconvergent for all points within the element. Denoting by $\underline{\sigma}_h$ the finite element approximation to the true stress field, we seek a smoothed stress field, $\underline{\sigma}^*$, in which each component is a p'th order polynomial. Thus, we may write

$$\underline{\sigma}^* = \mathbf{P}\underline{\mathbf{a}}, \tag{103}$$

where $\mathbf{P} = \text{diag}(\underline{\mathbf{p}}^T)$ and $\underline{\mathbf{p}}^T = [1, x, y, \ldots, y^p]$.

The smoothed stress field is then found by minimising

$$\Pi_1 = \sum_{k=1}^{n} (\sigma_h - \mathbf{P}_k\underline{\mathbf{a}})^2, \quad \mathbf{P}_k = \mathbf{P}(\underline{\mathbf{x}}_k). \tag{104}$$

over all superconvergent points $\underline{\mathbf{x}}_k$ in the patch, see Figure 8.1.1. For many familiar types of meshes, triangular meshes being a prime example, there are no superconvergent points. Zienkiewicz and Taylor [47] nevertheless report that the SPR method has produced good results in these cases as well. An alternative approach is to, for each element, take the average value of the stress field at the element centroid point and use those values in interpolating the smoothed stress field over the patch. This approach has been taken by Kvamsdal in [48]. This method will be detailed a bit more in follwing subsections.

Although the recovered stress field $\underline{\sigma}^*$ is an improvement compared to $\underline{\sigma}_h$ from the FE solution, the quality of the recovered derivatives at the boundaries is not as good as for internal nodes. In general, the recovered stress field neither satisfies the equilibrium equation (100), nor the boundary condition (102). An obvious enhancement would be to meet these requirements as well. Unfortunately this is not easily achieved, instead one seeks an approximation which satisfies the restrictions in a least square sense. Wiberg and Abdulwahab [49] included a weighted residual of the equilibrium equation, and later Wiberg, Abdulwahab and Ziukas [50] and Blacker and Belytschko [51] also included a

weighted residual for the boundary condition. The recovered stress field is now found by minimising

$$\Pi_2 = \Pi_1 + \alpha_1 \|\mathbb{D}^T \underline{\sigma}^* + \underline{\mathbf{b}}\|^2_{\tilde{\Omega}} + \alpha_2 \|\mathbf{Q}\underline{\sigma}^* - \tilde{\underline{\mathbf{t}}}\|^2_{\partial\tilde{\Omega}}, \tag{105}$$

over each patch $\tilde{\Omega}$. The coefficients α_1 and α_2 introduced above, are penalty numbers, and can be controlled depending on what term you wish to emphasise and where in the domain Ω you want them to play a role. An alternative approach for including the interior equilibrium enhancement, is done by choosing the monomials in the $\mathbf{P}$ matrix such that the equilibrium equation is satisfied a priori. Armed with these so-called "statically admissible" polynomials we may set $\alpha_1 = 0$ in (105) and minimise the resulting functional. Extending this approach, Kvamsdal and Okstad [46] introduced the idea of combining self-equilibration of residual with statically admissible polynomials. Their method, termed *SPR+S* for "Superconvergent Patch Recovery with Statically admissible polynomials", yields a conservative error estimate and, for smooth solutions, is an improvement over SPR.

Generally an element can belong to more than one patch, thus the SPR method does not produce unique values at interior points of an element. Blacker and Belytschko [51] proposed to replace the standard nodal interpolation scheme by a cojoint polynomial. The polynomial is constructed as a weighted sum of the overlapping local patch fields, where the weighting factors are the values of the shape functions associated with the patch assembly nodes. The cojoint polynomial is defined over each element e as

$$\hat{\underline{\sigma}}_e(\underline{\mathbf{x}}) = \sum_j^q N_j(\underline{\mathbf{x}}) \underline{\sigma}^*_j(\underline{\mathbf{x}}), \tag{106}$$

where q is the number of corner nodes, $N_j(\underline{\mathbf{x}})$ are shape functions and $\underline{\sigma}^*_j(\underline{\mathbf{x}})$ is the local patch field associated with node j. Blacker and Belytschko concluded that using the co-joint polynomial makes more effective use of the overlapping patches and showed improvements for bi-linear quadrilateral elements. When using SPR+S, the interior equlibrium equation will generally not be satisfied after cojoining the local patch field. However, Kvamsdal and Okstad [46] proved that, provided each local recovered stress field is close to the average of all overlapping fields, the residual in the equilibrium equation is small.

8.1.2 Recovery by Averaging

This subsection is based on the work by Ainsworth and Craig [52], where they study the recovery of ∇u by averaging. We will mainly follow their notation, though with slight modifications. Let n denote the number of space dimensions of the problem at hand, and let N^h denote the number of elements used in its finite element approximation. Let G_h denote an abstract averaging operator with the following properties

R1 *Consistency condition.* When the true solution is a polynomial of low degree G_h should recover the true gradient. Whenever $u \in \mathcal{P}_{p+1}(\Omega)$

$$G_h(\Pi^h_p u) \equiv \nabla u \tag{107}$$

where Π^h_p is the nodal interpolation operator onto the finite element space of order p.

R2 *Localising condition.* The value of G_h at a point should be simple to evaluate, without any global computations. This condition is more of practical reasons. Define a domain, $\widehat{\Omega}_i^h$, local to the finite element Ω_i^h as

$$\widehat{\Omega}_i^h = \bigcup_{j \in \mathrm{adj}(i)} \Omega_i^h, \tag{108}$$

where adj(i) is an indexing set containing i and the numbers of those elements local to Ω_i^h. The condition on G_h then reads. For $x^* \in \Omega_i^h, G_h[v](x^*)$ depends only upon the values of ∇v on the domain $\widehat{\Omega}_i^h$. Further the size of the local domain is limited by a constant M, independent of h, meaning

$$\mathrm{card}[\mathrm{adj}(i)] \leq M.$$

R3 *Boundedness and linearity conditions.* The recovery operator G_h should be simple in the sense that it should be easy to evaluate and integrate. Furthermore, G_h need only be defined on the finite element space of test functions V_h, since it is to be applied to $u_h \in V_h$. Finally, G_h is required to be bounded and linear, so the third condition reads

$$|G_h[v]|_{0,\infty,\Omega_i^h} \leq C|v|_{1,\infty,\widehat{\Omega}_i^h}, \quad \forall\, \Omega_i^h \in \mathcal{T}^h,\ \forall\, v \in V_h. \tag{109}$$

These three conditions guarantee that when the true solution u is smooth, $G_h(\Pi_p^h u)$ is a good local approximation to ∇u.

Lemma 8.1
Suppose that G_h satisfies (R1)–(R3) and that $u \in H^{p+2}(\widehat{\Omega}_i^h),\ 2p > n$. Then for $s = 0$ or $s = 1$

$$\|\nabla u - G_h(\Pi_p^h u)\|_{s,2,\Omega} \leq C h^{p+1-s} |u|_{p+2,2,\Omega},$$

In a finite element computation we do not know the true solution, so we need to use u_h instead of $\Pi_p^h u$. The difference between u_h and $\Pi_p^h u$ is commonly known as the pollution error. The pollution error is a globally transported error independent of local approximation properties. Under certain regularity conditions regarding the partition and the regularity of the true solution we have the by now the familiar superconvergence phenomenon

$$|u_h - \Pi_p^h u|_{1,2,\Omega} \leq \mathscr{C}(u) h^{p+1}. \tag{SC}$$

When superconvergence is present, ∇u_h is a better approximation to $\nabla \Pi_p^h u$ than it is to ∇u, and the pollution error is small. Under these special circumstances we have the following important result

Lemma 8.2
Suppose $u \in H^{p+2}(\Omega),\ p > n/2$, that (SC) *is valid and that G_h satisfies (R1)–(R3). Then for $s = 0$ or $s = 1$*

$$\left|\nabla u + G_h(u_h)\right|_{s,2,\Omega} \leq C h^{p+1-s} \{|u|_{p+2,2,\Omega} + \mathscr{C}(u)\}. \tag{110}$$

We may then estimate the error in the energy norm by

$$\epsilon^2 = \sum_{i=1}^{N^h} \epsilon_i^2,$$

where

$$\epsilon_i^2 = |||G_h[u_h] - \nabla u_h|||_{\Omega_i^h}, \quad i = 1, \ldots, N^h.$$

Ainsworth and Craig show that this estimator is asymptotically exact provided that G_h satisfies (R1)–(R3) and (SC) is valid.

THEOREM 8.1
Let ϵ be the a posteriori error estimator defined above, assume that (SC) *and (R1)–(R3) hold and $2p > n$, then ϵ is an asymptotically exact estimator:*

$$|||e||| = \epsilon\{1 + Ch^\gamma\} \quad \textit{as } h \to 0$$

where $\gamma > 0$ and C are constants independent of h.

Several examples of a posteriori error estimators satisfying the above conditions are presented in [52].

8.1.3 Recovery by Equilibration of Patches

The *recovery by equilibration of patches* (REP) is a recovery method which does not assume the existence of superconvergent points, and can be used as an alternative to SPR. The method was first presented by Zienkiewicz and Boroomand in [53]. The recovery technique starts out the same way as the SPR routine, by recovering a smooth stress field $\underline{\sigma}^*$ over the patch. Consider a differential equation of equilibrium in the form

$$\mathbb{D}^T \underline{\sigma} + \underline{\mathbf{b}} = 0, \quad \text{in } \Omega, \tag{111}$$

with FE formulation

$$\mathbf{F_\Omega} \equiv \int_\Omega \mathbf{B}^T \underline{\sigma} \, \mathrm{d}\Omega - \int_\Omega \mathbf{N}^T \underline{\mathbf{b}} \, \mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{N}^T \underline{\mathbf{t}} \, \mathrm{d}\Gamma = 0. \tag{112}$$

Here Γ_t is the part of the boundary with prescribed traction, $\mathbf{N}$, $\underline{\mathbf{b}}$ and $\underline{\mathbf{t}}$ denote shape functions, body and traction forces respectively. As usual the FEM solution is written $\underline{\mathbf{u}}_h = \mathbf{N}\tilde{\underline{\mathbf{u}}}$, represented by its vector of nodal values $\tilde{\underline{\mathbf{u}}}$. Finally, $\mathbf{B} \equiv \mathbb{D}\mathbf{N}$. If we assume a continuous displacement field the FEM solution produces a discontinuous strain field $\underline{\sigma}_h$.

Considering a patch, Ω_p, in the interior of the domain it interacts with the remaining parts according to the equilibrium conditions

$$\int_{\Omega_p} \mathbf{B}^T \underline{\sigma}_h \, \mathrm{d}\Omega = \mathbf{F}_p. \tag{113}$$

F_p is the sum of external forces from the remaining domain and the body forces over the patch. The idea of the REP method is to look for a new continuous strain field $\underline{\sigma}^*$ that satisfy

$$\int_{\Omega_p} \mathbf{B}^T \underline{\sigma}^* \, \mathrm{d}\Omega = \int_{\Omega_p} \mathbf{B}^T \underline{\sigma}_h \, \mathrm{d}\Omega, \tag{114}$$

as closely as possible. The continuous gradient field is given some polynomial expansion $\underline{\sigma}^* = \mathbf{P}\underline{\mathbf{a}}$, and (114) is satisfied in a least square sense by minimising

$$\Pi = \left(\int_{\Omega_p} \mathbf{B}^T \underline{\sigma}^* \, \mathrm{d}\Omega - \int_{\Omega_p} \mathbf{B}^T \underline{\sigma}_h \, \mathrm{d}\Omega \right)^T \left(\int_{\Omega_p} \mathbf{B}^T \underline{\sigma}^* \, \mathrm{d}\Omega - \int_{\Omega_p} \mathbf{B}^T \underline{\sigma}_h \, \mathrm{d}\Omega \right). \tag{115}$$

This is the REP version as it was first presented, but as pointed out by Boroomand and Zienkiewicz [54] a more robust alternative would be to split $\underline{\sigma}^*$ into its components and satisfy the equilibrium condition for each component independently. Each component of $\underline{\sigma}^*$ is now a polynomial independent of the others. In this form REP is very similar to SPR, but with a different smoothing procedure. Finally the smoothed stress field is used to decide nodal values of the stresses and these are subsequently interpolated in the usual way by standard shape functions. The accuracy of the method is highly dependent on the size of the patch, the type of elements used and the form of the functions for recovery.

8.1.4 Error Estimates by Recovery

An important reason for doing some kind of solution recovery is the need of a posteriori error estimates to adapt the mesh, and ultimately get even better solutions. The recovered values are in general much more accurate than the direct finite element solution. To estimate the error we simply replace the exact solution with the recovered one,

$$\|e\| \approx \|\bar{e}\| = \|u^* - u_h\|, \tag{116}$$

in some kind of norm. The quality of the error estimate is often measured by the effectivity index

$$\theta = \frac{\|\bar{e}\|}{\|e\|}. \tag{117}$$

It has been concluded by Zienkiewicz and Zhu that any recovery process resulting in less error will give a reasonable error estimate. If the recovered solution converges faster than the FEM solution then the estimated error converges to the true error.

8.2 Residual Based Error Estimators

A large class of error estimators uses the residuals of the finite element approximation, commonly known as *residual error estimators*. In [55] Babuška et. al. concluded that element-residual estimators should only be used with equilibration. These are known as *equilibrated element residual estimators*, and are the most robust among the class. Our presentation of the method will mainly follow that of Zienkiewicz and Taylor in [47]. Consider a non-linear heat conduction problem in two dimensions.

$$-\nabla^T(k\nabla\psi) = Q \quad \text{in } \Omega, \tag{118}$$

with boundary conditions

$$\begin{aligned} \psi &= \bar{\psi} \quad \text{on } \Gamma_\psi \\ (-k\nabla\psi)^T\underline{\mathbf{n}} &= \bar{q} \quad \text{on } \Gamma_q. \end{aligned}$$

As usual we define the error of the FEM solution as $e = \psi - \psi_h$. The residual in the interior element i is known to be

$$r_i = \nabla^T(k\nabla\psi) + Q. \tag{119}$$

The element error indicator for the implicit element residual estimator is given as the energy norm of the error taken over the element

$$\eta_i \equiv |||e_i|||_i\,, \tag{120}$$

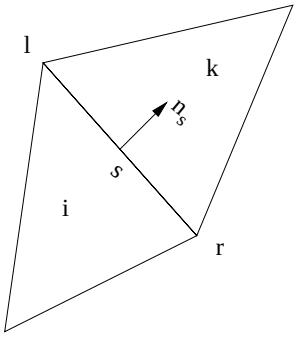


Figure 13: Two neighbouring elements sharing edge s

where

$$-\nabla^T(k\nabla e) = r_i \quad \text{in } \Omega_i, \tag{121}$$

with boundary conditions

$$-(k\nabla e)^T\underline{\mathbf{n}} = q_i - q_{h,i}. \tag{122}$$

The two terms on the right hand side of (122) are the exact normal flux and the finite element normal flux at the element edges. The exact normal flux is unknown for interior elements and must be replaced by a recovered flux q_i^*. To guarantee a solution of (121) the recovered flux must be computed such that the residual satisfy the equilibrium equation

$$\int_{\Omega_i} N_j r_i \,\mathrm{d}\Omega + \int_{\Gamma_i} N_j(q_i^* - q_{h,i}) \,\mathrm{d}\Gamma = 0. \tag{123}$$

The residual is then said to be equilibrated. In a well known flux recovery technique by Ladevéze the flux takes the form

$$q^* = \frac{1}{2}(\underline{\mathbf{q}}_{h,i} - \underline{\mathbf{q}}_{h,k})^T\underline{\mathbf{n}}_s + Z_s \tag{124}$$

where the first term on the right hand side is the average normal flux of the finite element solution from element i and its adjacent neighbour k over the edge s, see Figure 8.2. The term Z_s is a linear function defined over the edge s shared by elements i and k, with end nodes l and r

$$Z_s = L_l a_l^s + L_r a_r^s \tag{125}$$

with

$$L_l = \frac{2}{|h_s|}(2N_l^s - N_r^s) \quad L_r = \frac{2}{|h_s|}(2N_r^s - N_l^s) \tag{126}$$

where N_l^s and N_r^s are linear shape functions defined over the edge s and h_s is the length of the edge. Inserting the residual (119) into (123) results in the equation

$$\int_{\Omega_i} N_j Q \,\mathrm{d}\Omega - \int_{\Omega_i} \nabla^T(k\nabla\psi_h) \,\mathrm{d}\Omega + \int_{\Gamma_i} N_j q^* \,\mathrm{d}\Gamma = 0 \tag{127}$$

which will then decide the edge parameters. We can determine the parameters by looking at local problems involving only patches of elements connected to a node X_n, see Figure 8.2. For each patch around a node we obtain a linear system (see [47])

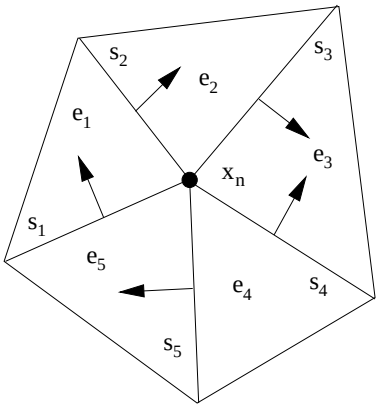


Figure 14: Patch of elements sharing node x_n

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} a_{x_n}^{s_1} \\ a_{x_n}^{s_2} \\ a_{x_n}^{s_3} \\ a_{x_n}^{s_4} \\ a_{x_n}^{s_5} \end{bmatrix} = \begin{bmatrix} -f_{e_1} \\ -f_{e_2} \\ -f_{e_3} \\ -f_{e_4} \\ -f_{e_5} \end{bmatrix}. \tag{128}$$

The equations are linearly dependent and no unique solution exists. The linear system is solved by first finding an optimal particular solution, which is then combined with a corresponding homogeneous solution to form the final solution.

When the edge parameters have been determined the local error problem is solved, for every patch, by a higher order approximation ($p+1$ or $p+2$). The solution will then supply us with the error indicator (120).

8.3 Goal Oriented Error Estimators

In many applications only specific quantities of the solution are of interest. Other quantities are of little or no interest. Examples of quantities of interest may be average value of the solution, average flux through a boundary, specific point values of both the solution and its gradients, the recovery of forces acting on a certain parts of a construction. The numerical error of the finite element approximation is now estimated in terms of these quantities rather than the classical L_2-norm or energy norm. Traditional adaption techniques are not optimal in this sense, as the will spend to many grid cells in areas not needed. The primary task of goal oriented adaption techniques is to control the error in the specific quantities. It has been confirmed by numerical experiments that such procedures greatly accelerate the attainment of local features, to a satisfactory accuracy, compared to traditional adaption methods based on energy norm estimates.

The specific quantity of interest is usually assumed to be given as a bounded linear functional Q applied to the true solution u. An examples of such a functional is the average of the solution u over a sub domain $\Omega_s \subset \Omega$,

$$Q(u) = \frac{1}{|\Omega_s|} \int_{\Omega_s} u(x)\,\mathrm{d}x, \tag{129}$$

In cases where the solution is vector-valued, the flux through $\partial\Omega_s$ might be of interest

$$Q(u) = \int_{\partial\Omega_s} \underline{\mathbf{u}} \cdot \underline{\mathbf{n}}\,\mathrm{d}\Gamma = \int_{\Omega_s} \nabla \cdot \underline{\mathbf{u}}\,\mathrm{d}\Omega. \tag{130}$$

In some cases the quantity of interest cannot be characterised by a bounded linear functional. In these cases the functional needs to be regularised [56, 57]. This is the case for estimating point values for example.

In [58] Melbø and Kvamsdal study the forces applied by a fluid on a local part of a structure. The authors use special extraction functions to express the force of interest in terms of the weak form of the Stokes problem at hand. In this approach, no regularisation is needed.

Instead of looking at the global error of the solution the goal oriented approach is to concentrate on the error of the quantity of interest. For the moment let us assume the weak form of our problem can be written as, find $u \in V$ such that

$$B(u, v) = F(v), \quad \forall v \in V, \tag{131}$$

where V is some appropriate space of test functions. Let the finite element approximation u_h of our problem be the solution of

$$B(u_h, v) = F(v), \quad \forall v \in V_h \subset V. \tag{132}$$

Here V_h is some discretisation of V. The residual are then defined as

$$R_h^u(v) = B(e, v) = F(v) - B(u_h, v), \quad \forall v \in V. \tag{133}$$

The Galerkin orthogonality property now states that

$$R_h^u(v) = 0, \forall v \in V_h. \tag{134}$$

In what follows we are interested in estimating the quantity $Q(u)$. The quantity may be estimated by $Q(u_h)$ with the error

$$Q(u) - Q(u_h) = Q(e), \tag{135}$$

where $e = u - u_h$ is the error in u_h. The main target for goal oriented adaption is to minimise the error in the functional. This is done by linking the error in the functional (135) to the residual (133). One obvious approach would be to solve the problem (133), say on a finer grid, and then by (135) give an estimate of the error in $Q(u_h)$. This is generally too costly, instead one applies the Aubin-Nitsche trick and solve the dual problem of (131). Find $w \in V$ such that

$$B(v, w) = Q(v), \quad \forall v \in V \tag{136}$$

Let w_h be its finite element approximation

$$B(v, w_h) = Q(v), \quad \forall v \in V_h. \tag{137}$$

In a similar way let us denote the error of w_h by $\epsilon = w - w_h$. A straight forward computation [56] then links the error of the functional with the error of u_h and its dual w_h as

$$Q(e) = B(e, \epsilon) \tag{138}$$

From this point on there are several ways to go about to estimate $Q(e)$. One approach is to use Cauchy-Schwartz and express $Q(e)$ in terms of local error estimates of $e = u - u_h$ and $\epsilon = w - w_h$. Let K denote an element in the discretisation V_h of V.

$$|Q(e)| = |B(e, \epsilon)| \leq \sum_K |B_K(e, \epsilon)| \leq \sum_K |||e|||_K \, |||\epsilon|||_K \tag{139}$$

Assume at this point we have local error estimates $|||e|||_K \leq \eta_K^u$ and $|||\epsilon|||_K \leq \eta_K^w$, and define $\eta^Q = \sum_K \eta_K^u \eta_K^w$ as an estimate of $Q(e)$. This is the approach used by Becker and Rannacher [57]. They look at local residuals, and propose to approximate the true dual solution w by higher order polynomial interpolation of w_h over a patch containing the element K. Denote the higher order approximation of order q by $I_h^{(q)} w_h$. Then approximate ϵ by $\left|\left|\left| I_h^{(q)} w_h - w_h \right|\right|\right|$.

Another approach, when $B(\cdot,\cdot)$ is symmetric, is to rewrite $B(e,\epsilon)$ as

$$B(e,\epsilon) = \left|\left|\left| se + s^{-1}\epsilon \right|\right|\right| + \left|\left|\left| se - s^{-1}\epsilon \right|\right|\right|, \tag{140}$$

and choose the scaling factor $s = \sqrt{|||\epsilon||| \,/\, |||e|||}$ to minimise $B(e,\epsilon)$. The approach continues to find upper and lower bounds for the two terms in (140). Based on these bounds an estimate for $Q(e)$ is derived, see [56].

In many cases, problem (132) and its dual (137) are simply the same, but with different right hand sides. Thus, the dual problem can be solved at little additional cost.

9 Open Problems and Future Work

The methods listed thus far have all been amply studied in detail when applied to the problems for which they were designed. Combining them on the other hand remains a challenge and possible future work lies in this direction.

More specifically, however, there are a couple of open problems with these methods as presented earlier. Moving mesh methods are vulnerable to crossing nodes which destroy the element topolgy. Creating methods wich, similarly to the deformation map method, guarantee elemination of this problem is highly desirable. Furthermore, the system of ODEs arising from a spatial semi-discretisation of the physical problem on the moving mesh is usually stiff. This means that to apply moving meshes in several space dimensions, accurate, efficient and stiffly stable integration methods for large systems of ODEs is a necessity.

Another challenge is to combine h-adaptive methods and moving meshes. Identifying which regions of the domain will benefit more from which adaption strategy is key to developing a method which is applicable to a large range of problems in fluid flow. In addition, carrying out the actual mesh adaption requires availability of efficient and automatic mesh generation tools.

Combining different methods applied to a single problem this way may lead to interesting topics, but another possibility is to combine already well known methods to an ensemble of related problems, such as the interaction between fluid and structure. Creating integration methods for time dependent fluid structure interaction problems may involve splitting of the spatial operators in the different sub regions of the domain. Compounding the difficulties in solving fluid structure interaction problems is the mere fact that the data are not necessarily available in the same computational meshes for each spatial differential opertor. Thus, simply transferring data between non-overlapping sub-meshes while preserving distinctive features of the solution needs to be taken into account.

Operator splitting methods for systems of ODEs have been subject to much interest of late, in particular with respect qualitative properties of the resulting integrator and how well certain features of the solution are preserved throughout the time integration.

These kinds of methods may well find new applicability in physical problems where the spatial operators are naturally split in different regions.

Finally, much of the software created for solving fluid flow problems are based on relatively low order approximations to the velocity and the pressure. Developing higher order approximations, in particular for time dependent problems, is then a challenge.

References

[1] T. Parnell. The Pitch Drop Experiment. Internet: <URL:http://www.physics.uq.edu.au/pitchdrop/pitchdrop.shtml>, 1927–. Live footage available at the web site.

[2] C. C. Lin and L. A. Segel. *Mathematics Applied to Deterministic Problems in the Natural Sciences*. Classics in Applied Mathematics. SIAM, 1995.

[3] F. M. White. *Viscous Fluid Flow*. Mechanical Engineering Series. McGraw-Hill, Inc., 1991.

[4] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer-Verlag, second corrected edition, 1997.

[5] L. Formaggia and F. Nobile. A Stability Analysis for the Arbitrary Lagrangian Eulerian Formulation with Finite Elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[6] H. Guillard and C. Farhat. On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes. *Computational Methods in Applied Mechanics and Engineering*, 190:1467–1482, 2000.

[7] C. Farhat, P. Geuzaine, and C. Grandmont. The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids. *Journal of Computational Physics*, 174:669–694, 2001.

[8] B. Koobus and C. Farhat. Second-Order Time-Accurate and Geometrically Conservative Implicit Schemes for Flow Computations on Unstructured Dynamic Meshes. *Computational Methods in Applied Mechanics and Engineering*, 170:103–129, 1999.

[9] P. Le Tallec and C. Martin. A Nonlinear Elasticity Model for Structured Mesh Adaption. In *Computational Fluid Dynamics*, pages 275–281. John Wiley & Sons Ltd., 1996.

[10] O.-P. Jacquotte. A Mechanical Model for a New Grid Generation Method in Computational Fluid Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 66:323–338, 1988.

[11] J. T. Batina. Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes. In *27th Aerospace Science Meeting, Reno 18-20 apr. 1989*, AIAA Paper 89-0115. AIAA, 1989.

[12] P. Le Tallec. *Handbook of Numerical Analysis*, volume III, chapter Numerical Methods for Nonlinear Three-dimensional Elasticity, pages 465–622. North-Holland, 1994.

[13] P. G. Ciarlet. *Mathematical Elasticity, Vol. I: Three-Dimensional Elasticity*. Number 20 in Studies in Mathematics and its Applications. North-Holland, 1988.

[14] R. A. Serway. *Physics for Scientists & Engineers with Modern Physics*. Saunders, 3 edition, 1992.

[15] P. Pegon and K. Mehr. Rezoning and Remeshing of the Fluid Domain. Technical report, Structural Mechanics Unit, Institute for Systems, Informatics and Safety, Joint Research Centre, European Commission, 1998.

[16] C. Farhat, M. Lesoinne, and N. Maman. Mixed Explicit/Implicit Time Integration of Coupled Aeroplastic Problems: Three-Field Formulation, Geometric Conservation and Distributed Solution. *International Journal for Numerical Methods in Fluids*, 21:807–835, 1995.

[17] J. Moser. On the Volume Elements on a Manifold. *Trans. A.M.S.*, 120:286–294, 1965.

[18] B. Dacorogna and J. Moser. On a Partial Differential Equation Involving the Jacobian Determinant. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 7(1):1–26, 1990.

[19] G. Liao and D. Anderson. A New Approach to Grid Generation. *Applicable Analysis*, 44:285–298, 1992.

[20] G. Liao and J. Su. Grid Generation via Deformation. *Applied Mathematics Letters*, 5(3):27–29, 1992.

[21] G. Liao, T.-W. Pan, and J. Su. Numerical Grid Generator Based on Moser's Deformation Method. *Numerical Methods for Partial Differential Equations*, 10:21–31, 1994.

[22] Z. Lei, G. Liao, G. C. de la Pena, and D. Anderson. A Moving Grid Algorithm Based on Deformation Method. Journal of publication unknown.

[23] B. Semper and G. Liao. A Moving Grid Finite-Element Method Using Grid deformation. *Numerical Methods for Partial Differential Equations*, 11:603–615, 1995.

[24] P. Bochev, G. Liao, and G. de la Pena. Analysis and Computation of Adaptive Moving Grids by Deformation. *Numerical Methods for Partial Differential Equations*, 12:489–506, 1996.

[25] F. Liu, S. Ji, and G. Liao. An Adaptive Grid Method and Its Applications to Steady Euler Flow Calculations. *SIAM Journal on Scientific Computing*, 20(3):811–825, 1998.

[26] G. Liao, G. de la Pena, and G. Liao. A Deformation Method for Moving Grid Generation. Journal of publication unknown.

[27] W. Huang W. Cao and R. D. Russel. A Moving Mesh Method Based on the Geometric Conservation Law. Journal of publication unknown.

[28] W. Huang, Y. Ren, and R. D. Russell. Moving Mesh Partial Differential Equations (MMPDEs) Based on the Equidistribution Principle. *SIAM Journal on Numerical Analysis*, 31(3):709–730, June 1994.

[29] W. Huang, Y. Ren, and R. D. Russell. Moving Mesh Methods Based on Moving Mesh Partial Differential Equations. *Journal of Computational Physics*, 113:279–290, 1994.

[30] E. A. Dorfi and L. O'C. Drury. Simple Adaptive Grids for 1-D Initial Value Problems. *Journal of Computational Physics*, 69:175–195, 1987.

[31] W. Huang and R. D. Russell. Analysis of Moving Mesh Partial Differential Equations With Spatial Smoothing. *SIAM Journal on Numerical Analysis*, 34(3):1106–1126, June 1997.

[32] W. Cao, W. Huang, and R. D. Russell. A Study of Monitor Functions for Two Dimensional Adaptive Mesh Generation. *SIAM Journal on Scientific Computing*, 20(6):1978–1994, 1999.

[33] W. Huang and R. D. Russell. A High Dimensional Moving Mesh Strategy. *Applied Numerical Mathematics*, 26:63–76, 1998.

[34] W. Huang. Practical Aspects of Formulation and Solution of Moving Mesh Partial Differential Equations. *Journal of Computational Physics*, 171:753–775, 2001.

[35] W. Cao, W. Huang, and R. D. Russell. A Two-dimensional r-Adaptive Finite Element Method Based on A Posteriori Error Estimates. *Journal of Computational Physics*, 171:871–892, 2001.

[36] W. Cao, W. Huang, and R. D. Russell. Comparison of Two-Dimensional r-Adaptive Finite Element Methods Using Various Error Indicators. *Mathematics and Computers in Simulation*, 56:127–143, 2001.

[37] K. Miller and R. N. Miller. Moving Finite Elements I. *SIAM Journal on Numerical Analysis*, 18(6):1019–1032, December 1981.

[38] K. Miller. Moving Finite Elements II. *SIAM Journal on Numerical Analysis*, 18(6):1033–1057, December 1981.

[39] H. P. Langtangen. *Computational Partial Differential Equations, Numerical Methods and Diffpack Programming*, volume 2 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 1999.

[40] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Studentlitteratur, 1987.

[41] M. J. Baines. *Moving Finite Elements*. Monographs on Numerical Analysis. Oxford University Press, 1994.

[42] N. Carlson and K. Miller. Design and Application of a Gradient-Weigthed Moving Finite Element Code, Part I, in 1-D. *SIAM Journal on Scientific Computing*, 19(3):728–765, May 1998.

[43] N. Carlson and K. Miller. Design and Application of a Gradient-Weighted Moving Finite Element Code, Part II, in 2-D. *SIAM Journal on Scientific Computing*, 19(3):766–798, May 1998.

[44] P. Coorevits, P. Ladeveze, and J.-P. Pelle. An Automatic Procedure with a Control of Accuracy for Finite Element Analysis in 2D Elasticity. *Journal of Computational Methods in Applied Mechanics and Engineering*, 121:91–120, 1995.

[45] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley Teubner, 1996.

[46] T. Kvamsdal and K. M. Okstad. Error Estimation Based on Superconvergent Patch Recovery Using Statically Admissible Stress Fields. *International Journal for Numerical Methods in Engineering*, 42:443–472, 1998.

[47] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, The Basis*, volume 1, chapter 14. Butterworth-Heinemann, 5 edition, 2000.

[48] T. Kvamsdal. Variationally Consistent Postprocessing for Adaptive Recovery of Sresses. In *ECCM '99*, 1999.

[49] N.-E. Wiberg and F. Abdulwahab. Patch Recovery Based on Superconvergent Derivaties and Equilibrium. *International Journal for Numerical Methods in Engineering*, 36:2703–2724, 1993.

[50] N.-E. Wiberg, F. Abdulwahab, and S. Ziukas. Enhanced Superconvergent Patch Recovery Incorporating Equilibrium and Boundary Conditions. *International Journal for Numerical Methods in Engineering*, 37:3417–3440, 1994.

[51] T. Blacker and T. Belytschko. Superconvergent Patch Recovery with Equilibrium and Conjoint Interpolant Echancements. *International Journal for Numerical Methods in Engineering*, 37:517–536, 1994.

[52] M. Ainsworth and A. Craig. A posteriori Error Estimators in the Finite Element Method. *Numerische Mathematik*, 60:429–463, 1992.

[53] B. Boroomand and O. C. Zienkiewicz. Recovery by Equilibrium in Patches (REP). *International Journal for Numerical Methods in Engineering*, 40(1):137–164, 1997.

[54] B. Boroomand and O. C. Zienkiewicz. An Improved REP Recovery and the Effectivity Robustness Test. *International Journal for Numerical Methods in Engineering*, 40(17):3247–3277, 1997.

[55] I. Babuška. Validation of A Posteriori Error Estimators by Numerical Approach. *International Journal for Numerical Methods in Engineering*, 37(7):1073–1123, 1994.

[56] J. T. Oden and S. Prudhomme. Goal-Oriented Error Estimation and Adaptivity for the Finite Element Method. *Computers and Mathematics with Applications*, 41:735–756, 2001.

[57] R. Becker and R. Rannacher. A Feed-Back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples. *East-West Journal of Numerical Mathematics*, 4(4):237–264, 1996.

[58] H. Melbø and T. Kvamsdal. Goal Oriented Error Estimators for Stokes Equations Based on Variationally Consistent Postprocessing. Accepted for publication and in press by the *Journal of Computational Methods in Applied Mechanics and Engineering*.

[59] A. S. Dvinsky. Adaptive Grid Generation from Harmonic Maps on Riemannian Manifolds. *Journal of Computational Physics*, 95:450–476, 1991.

[60] D. F. Griffiths, D. J. Higham, and A. B. Ross. Equidistributing Grids. Technical report, University of Dundee, 2001. `http://www.mcs.dundee.ac.uk:8080/~dfg/homepage.html`.

[61] W. Cao, W. Huang, and R. D. Russell. A Moving Mesh Method Based on the Geometric Conservation Law. Accepted for publication in "SIAM Journal on Scientific Computing".

[62] P. A. Zegeling. Moving Mesh Methods for Partial Differential Equations. Unpublished lecture notes from lecture series given at NTNU, 2001.

[63] P. A. Zegeling, J. G. Verwer, and R. M. Furzeland. A Numerical Study of Three Moving-Grid Methods for One-Dimensional Partial Differential Equations Which are Based on the Method of Lines. *Journal of Computational Physics*, 89:349–388, 1990.

[64] J. M. Coyle, J. E. Flaherty, and R. Ludwig. On the Stability of Mesh Equidstribution Strategies for Time-Dependent Partial Differential Equations. *Journal of Computational Physics*, 62:26–39, 1986.

[65] P. Zegeling and J. G. Blom. An Evaluation of the Gradient-Weighted Moving Finite Element Method in One Space Dimension. *Journal of Computational Physics*, 103:422–441, 1992.

[66] W. Huang and R. D. Russel. A Moving Collocation Method for Solving Time Dependent Partial Differential Equations. *Applied Numerical Mathematics: Transactions of IMACS*, 20(1–2):101–116, 1996.

[67] M. J. Baines. Moving Finite Element, Least Squares, and Finite Volume Approximations of Steady and Time-Dependent PDEs in Multidimensions. *Journal of Computational and Applied Mathematics*, 128:363–381, 2001.

[68] R. L. Wenbin. Moving Mesh Finite Element Methods Based on Harmonic Maps. Internet. `http://citeseer.nj.nec.com/377116.html`.

[69] P. Zegeling. *Moving-Grid Methods for Time-Dependent Partial Differential Equations*. PhD thesis, Centrum voor Wiskunde en Informatica, Amsterdam, 1992.

[70] P. Zegeling. r-refinement for Evolutionary PDEs with Finite Elements or Finite Differences. *Applied Numerical Mathematics*, 26:97–104, 1998.

[71] J. Donea. *Computational Methods for Transient Analysis*, volume 1 of *Mechanics and Mathematical Methods*, chapter 10: Arbitrary Lagrangian-Eulerian Finite Element Methods, pages 473–516. North-Holland, Elsevier, 1983.

[72] T. Kvamsdal et al., editor. *Computational Methods for Fluid-Structure Interaction*, 1999.

[73] P. Le Tallec and S. Mani. Conservation Laws for Fluid-Structure Interactions. In *Computational Methods for Fluid-Structure Interaction*, pages 61–78, 1999.

[74] C. S. Venkatasubban. A New Finite Element Formulation for ALE (Arbitrary Lagrangian Eulerian) Compressible Fluid Mechanics. *Int. J. Engng Sci*, 33(12):1743–1762, 1995.

[75] M. S. Gadala and J. Wang. ALE Formulation and Its Application in Solid Mechanics. *Comput. Methods Appl. Mehc. Engrg.*, 167:33–55, 1998.

[76] M. Lesoinne and C. Farhat. Stability Analysis of Dynamic Meshes for Transient Aerolastic Computations. In *11th AIAA Computational Fluid Dynamics Conference, Orlando*, AIAA Paper 93-3325. AIAA, 1993.

[77] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz. Analysis of the Zienkiewicz-Zhu A Posteriori Error Estimator in the Finite Element Method. *International Journal for Numerical Methods in Engineering*, 28:2161–2174, 1989.

[78] D. W. Kelly. The Self-Equilibration of Residuals and Complementary A Posteriori Error Estimates in the Finite Element Method. *International Journal for Numerical Methods in Engineering*, 20:1491–1506, 1984.

[79] H. Melbø. *A Posteriori Error Estimation for Finite Element Methods and Iterative Linear Solvers*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2001.

[80] I. Babuška and A. Miller. The Post-Processing Approach in the Finite Element Method - Part 1: Calculation of Displacements, Stresses and Other Higher Derivatives of the Displacements. *International Journal for Numerical Methods in Engineering*, 20:1085–1109, 1984.

[81] I. Babuška and A. Miller. The Post-Processing Approach in the Finite Element Method - Part 2: The Calculation of Stress Intensity Factors. *International Journal for Numerical Methods in Engineering*, 20:1111–1129, 1984.

[82] I. Babuška and A. Miller. The Post-Processing Approach in the Finite Element Method - Part 3: A Posteriori Error Estimates and Adaptive Mesh Selection. *International Journal for Numerical Methods in Engineering*, 20:2311–2324, 1984.

[83] M. Paraschivoiu, J. Peraire, and A. T. Patera. A Posteriori Finite Element Bounds for Linear-Functional Outputs of Elliptic Partial Differential Equations. *Computer methods in applied mechanics and engineering*, 150:289–312, 1997.

[84] M. Paraschivoiu and A. T. Patera. A Posteriori Bounds for Linear Functional Outputs of Crouzeix-Raviart Finite Element Discretization of the Incompressible Stokes Problem. *International Journal for Numerical Methods in Fluids*, 32:823–849, 2000.

[85] G. Liao, F. Liu, G. C. de la Pena, D. Peng, and S. Osher. Level-Set-Based Deformation Methods for Adaptive Grids. *Journal of Computational Physics*, 159:103–122, 2000.

[86] G. J. Liao and J. Z. Su. A Moving Grid Method for $(1 + 1)$ Dimension. *Applied Mathematics Letters*, 4:47–49, 1995.

[87] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, The Basis*, volume 1. Butterworth Heinemann, 2000.

[88] J. E. Flaherty et al., editors. *Adaptive Methods for Partial Differential Equations*. SIAM, 1989.

[89] P. M. Gerhart, R. J. Gross, and J. I. Hochstein. *Fundamentals of Fluid Mechanics*. Addison-Wesley Publishing Company, 1992.