Centrum voor Wiskunde en Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

A Finite-Element Solver for the 2D Heat Equation with Convection

J. Wackers

REPORT MAS-E0408 JULY 2004

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

A Finite-Element Solver for the 2D Heat Equation with Convection

ABSTRACT

A finite-element method is developed for the two-dimensional advection-diffusion heat equation. The method features up to cubic triangular elements with Lagrange polynomial basis functions and isoparametric elements for curved boundaries. First, test problems show that the error of the method agrees with theoretical estimates. Then the method is used to compute the temperature in the flow around an internally cooled airfoil and in the airfoil itself.

2000 Mathematics Subject Classification: 65N30 80A20
Keywords and Phrases: Finite-element method; convection-diffusion equation; isoparametric elements; cooled airfoil
Note: This research was supported by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (http://www.bsik-bricks.nl), theme MSV1, and was carried out under CWI project MAS2.1 `Computational Fluid Dynamics and Computational Electromagnetics'.

A Finite-Element Solver for the 2D Heat Equation with Convection

Jeroen Wackers
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
A finite-element method is developed for the two-dimensional advection-diffusion heat equation. The method features up to cubic triangular elements with Lagrange polynomial basis functions and isoparametric elements for curved boundaries.

First, test problems show that the error of the method agrees with theoretical estimates. Then the method is used to compute the temperature in the flow around an internally cooled airfoil and in the airfoil itself.

2000 Mathematics Subject Classification: 65N30, 80A20
Keywords and Phrases: Finite-element method, convection-diffusion equation, isoparametric elements, cooled airfoil.
Note: This research was supported by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (http://www.bsik-bricks.nl), theme MSV1, and was carried out under CWI project MAS2.1 'Computational Fluid Dynamics and Computational Electromagnetics'.

1. Introduction

This report describes a finite-element method for the heat equation with convection. It describes both the solver itself and the results for a realistic physical problem, obtained with the solver. The report is written as an assignment for a course studying the notes by Hemker [1]. It concentrates on those aspects of the finite-element method that are needed for the present method, but not treated in-depth in these notes. The new points are:

1. High-order triangular elements and accurate quadrature in 2D,
2. Computing integrals over a master element, instead of over the real elements,
3. Isoparametric elements, with curved edges.

The basics of the finite-element method and the theoretical error estimates for this method are assumed to be known to the reader. A theoretical basis for the finite-element method is found in [2].

The physical problem studied is the heat distribution in the fluid flow past an airfoil, which is cooled from a duct within, and in the airfoil itself. The problem is a model for the flow in a gas turbine, where hot air from the combustion chamber passes the internally cooled first stages of the turbine.

2. Heat equation

In this paper, we seek the temperature distribution in an incompressible fluid flow. For these flows, the velocity field is independent of the temperature. The temperature is given by the heat equation with convection, which does depend on the flow field. This equation is:

$$-k\Delta T + \rho C_p \mathbf{u}(\mathbf{x}) \cdot \nabla T = 0. \tag{2.1}$$

The heat conduction coefficient k, the density ρ and specific heat C_p are constant, the velocity $\mathbf{u}$ varies through the domain. So we may write, for a closed domain Ω with boundary $\partial\Omega$,

$$\begin{aligned} -\epsilon\Delta T + \mathbf{u}(\mathbf{x}) \cdot \nabla T = 0 \qquad & \text{in } \Omega, \\ B(T) = 0 \qquad & \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

with $\epsilon = \frac{k}{\rho C_p}$. For the problems studied here, the boundary condition B is a combination of inhomogeneous Dirichlet conditions,

$$T(\mathbf{x}) = T_B(\mathbf{x}), \tag{2.3a}$$

and homogeneous Neumann conditions,

$$\frac{\partial T}{\partial n} = 0. \tag{2.3b}$$

The domain may consist of two subdomains Ω_1 and Ω_2 with different ϵ. Then the solution is given by equation (2.2) in each of the domains, with the appropriate boundary conditions and an inhomogeneous Dirichlet condition on the boundary between the domains:

$$T_{\Omega_1} = T_{\Omega_2} \qquad \text{on } \partial\Omega_1 \cap \partial\Omega_2. \tag{2.3c}$$

3. Galerkin formulation

Multiply equation (2.2) with a test function $\psi \in H_0^1(\bar{\Omega})$ and integrate over Ω. After partial integration, we find:

$$\epsilon \iint_\Omega T_x \psi_x \, d\Omega + \epsilon \iint_\Omega T_y \psi_y \, d\Omega + \iint_\Omega u T_x \psi \, d\Omega + \iint_\Omega v T_y \psi \, d\Omega = 0. \tag{3.1}$$

Now divide Ω in m elements Ω_k, $k \in [1, m]$ and define n basis functions ϕ_i, $i \in [1, n]$. Then T is approximated with a linear combination of these basis functions,

$$T_h = \sum_{i=1}^{n} T_i \phi_i. \tag{3.2}$$

The T_i are the values of T in n control points, distributed over the elements. If we use the same functions ϕ as test functions, then equation (3.1) becomes, $\forall j \in [1, n]$:

$$\sum_{i=1}^{n} T_i \sum_{k=1}^{m} \left(\epsilon \iint_{\Omega_k} \phi_{i_x} \phi_{j_x} \, d\Omega + \epsilon \iint_{\Omega_k} \phi_{i_y} \phi_{j_y} \, d\Omega + \iint_{\Omega_k} u \phi_{i_x} \phi_j \, d\Omega + \iint_{\Omega_k} v \phi_{i_y} \phi_j \, d\Omega \right) = 0. \tag{3.3}$$

This is a system of n equations

$$A\mathbf{T}_h = 0, \tag{3.4}$$

which is solved for $\mathbf{T}_h$, the vector containing T_i, $i = 1, \cdots, n$. The integrals are first computed per element and grouped in element matrices A_e, which are then summed to form A.

To incorporate the boundary conditions (2.3), some control points must be placed on the boundary. In case of Dirichlet boundary conditions, the T_i at the boundary must get the prescribed value, so the rows in A for these T_i contain only a 1 on the main diagonal and the value T_B is placed in the right hand side vector. For Neumann boundary conditions, an extra boundary integral appears in equation (3.1), as the result of the partial integration of the first terms. But for homogeneous Neumann boundary conditions, this boundary integral has the value 0. Or, in other words, homogeneous Neumann boundary conditions are applied by doing absolutely nothing, they are natural boundary conditions.

The internal boundary condition (2.3c) is implemented by placing some degrees of freedom on the internal boundary and by computing equations for these degrees of freedom in the normal way. Only, these equations have contributions from elements on both sides of the boundary. The diffusion parameter ϵ is constant in each element, so the element-wise integration gives no special problems. When summing the contributions from elements in two different domains, it is important to realise that we really solve equation (2.1), instead of (2.2). The domains may have different ρ and C_p, so we must sum $(\rho C_p)_1$ times the contribution from the elements in domain 1, plus $(\rho C_p)_2$ times the contributions from domain 2.

4. Elements, basis functions and quadrature

For convenience and to simplify theoretical error estimation, we use elements that are equivalent to a single 'master' element. Thus, each element has the same degrees of freedom and the basis functions on each element are the same. The elements are triangular, so the master element is the triangle (0,0) (1,0) (0,1).

A number of control points $\mathbf{x}_i$ is defined on each element. The basis functions are the Lagrangian polynomials, that have the value 1 in one of the control points and 0 in all the other points. In order to span the 2D polynomial space of degree p, a total of $n_e = (p+1)(p+2)/2$ basis functions are needed. Thus, an element with linear functions has 3 basis functions and 3 control points per element, a quadratic element has 6 basis functions, etc. The linear, quadratic and cubic master elements used in this paper are drawn in figure 1, with their respective control points. Full properties of the elements are found in appendix A.

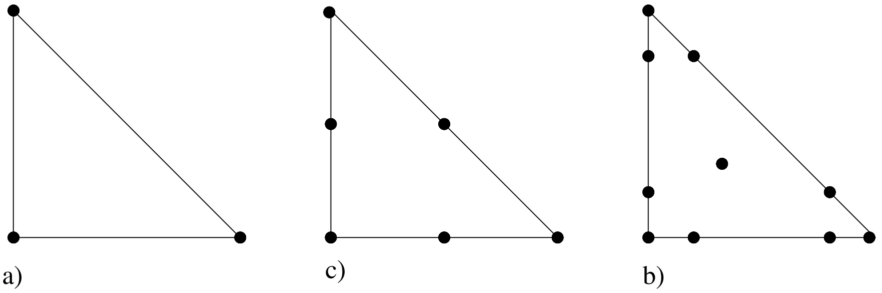


Figure 1: Linear (a), quadratic (b) and cubic (c) element.

To evaluate an integral of a function f over Ω_k approximately, a quadrature rule is used: the function is approximated by the basis functions on Ω_k and this approximation is integrated. For $f \in \mathcal{C}^{p+1}(\Omega_k)$, we have

$$f(\mathbf{x}) = \sum_{i=1}^{n_e} f(\mathbf{x}_i)\phi_i(\mathbf{x}) + \mathcal{O}(h^{p+1}).$$

Integrating this expression, we get

$$\iint_{\Omega_k} f(\mathbf{x})\, d\Omega = \sum_{i=1}^{n_e} f(\mathbf{x}_i) \iint_{\Omega_k} \phi_i \, d\Omega + \mathcal{O}(h^2)\cdot\mathcal{O}(h^{t+1}), \quad \text{with} \quad t \geq p.$$

So the quadrature rule becomes

$$\iint_{\Omega_k} f(\mathbf{x})\, d\Omega = \sum_{i=1}^{n_e} w_i f(\mathbf{x}_i) + \mathcal{O}(h^2)\cdot\mathcal{O}(h^{t+1}), \qquad \text{with } w_i = \iint_{\Omega_k} \phi_i \, d\Omega, \tag{4.1}$$

the weight functions are the integrals of the basis functions.

The order of accuracy of the quadrature rule $t+1$ is at least as good as the order $p+1$ of the basis function polynomials, meaning that the quadrature rule integrates polynomials of degree p exactly. But, by choosing the locations of the control points properly, even polynomials of a higher degree can be integrated exactly.

It is proved in [1] for the 1D case, that the order of the quadrature rule must satisfy $t+1 \geq 2p-1$, to preserve the order of accuracy of the finite-element method. This required order is not higher than $p+1$ for the linear and quadratic elements, that can thus have arbitrary locations for the control points. But the cubic element can have only one control point distribution, to get a fifth-order accuracy for the quadrature rule. This distribution is non-equidistant, see figure 1. A full description is given in appendix A.

5. Transformation to the master element

As all elements are topologically equivalent, the element-wise integrals are not computed directly. Instead, the integrands are transformed to the master element and integrated there.

Usually, the elements are triangles. In that case, the transformation is linear and the integration is rather simple. But if the elements have a more complex shape, then an isoparametric transformation must be used. Both are explained here.

5.1 Linear transformation

Consider an element Ω, with coordinates x and y and basis functions ϕ_i (the subscripts k are dropped for convenience). This element has a master element $\hat{\Omega}$, with coordinates $\hat{x}$ and $\hat{y}$ and basis functions $\hat{\phi}_i$. We define the correspondence of a function on Ω and $\hat{\Omega}$ as:

$$f(\mathbf{x}) = \hat{f}(\hat{\mathbf{x}}), \tag{5.1}$$

for any function f. If both elements are triangular, then there is an affine transformation F:

$$\mathbf{x} = F(\hat{\mathbf{x}}) = B\hat{\mathbf{x}} + \mathbf{b}, \tag{5.2}$$

with constant B and $\mathbf{b}$, such that $\Omega = F(\hat{\Omega})$. If we call the leftmost corner of the elements 1, the lowest of the remaining two corners 2 and the last corner 3 (figure 2), then the transformation becomes:

$$\mathbf{b} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \qquad B = \begin{pmatrix} \Delta x_2 & \Delta x_3 \\ \Delta y_2 & \Delta y_3 \end{pmatrix}. \tag{5.3}$$

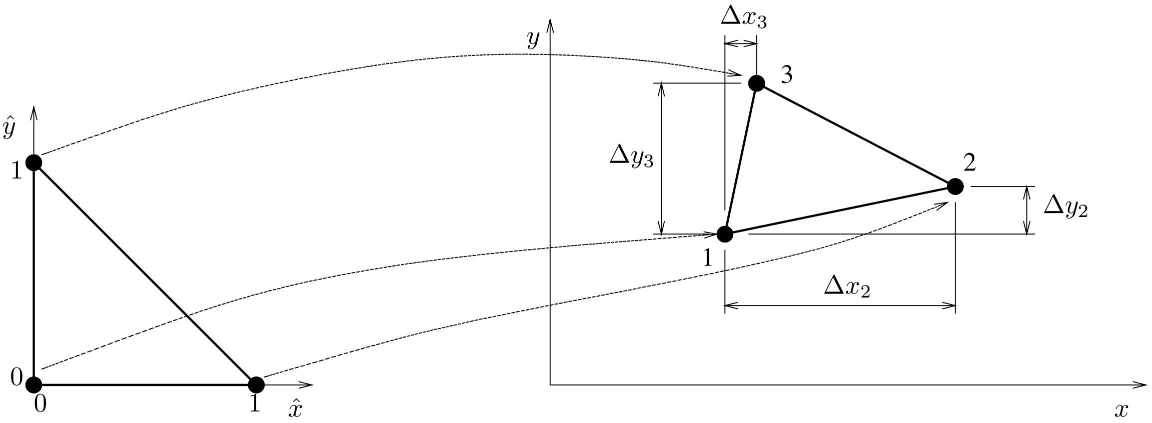


Figure 2: Linear transformation, from master element $\bar{\Omega}$ to Ω.

The determinant of B gives the ratio of the areas of Ω and $\hat{\Omega}$:

$$D = \Delta x_2 \Delta y_3 - \Delta x_3 \Delta y_2. \tag{5.4}$$

Thus $d\Omega = D\, d\hat{\Omega}$ and

$$\iint_{\Omega} f\, d\Omega = D \iint_{\hat{\Omega}} \hat{f}\, d\hat{\Omega}. \tag{5.5}$$

The derivatives of f follow from the chain rule, applied to equation (5.1):

$$\frac{\partial f}{\partial x} = \frac{\partial \hat{f}}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial \hat{f}}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x},$$
$$\frac{\partial f}{\partial y} = \frac{\partial \hat{f}}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial \hat{f}}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y}.$$

And as

$$d\hat{\mathbf{x}} = B^{-1} d\mathbf{x} = \frac{1}{D} \begin{pmatrix} \Delta y_3 & -\Delta x_3 \\ -\Delta y_2 & \Delta x_2 \end{pmatrix} d\mathbf{x},$$

we find that

$$\frac{\partial \hat{x}}{\partial x} = \frac{\Delta y_3}{D}, \qquad \frac{\partial \hat{y}}{\partial x} = -\frac{\Delta y_2}{D}, \qquad \frac{\partial \hat{x}}{\partial y} = -\frac{\Delta x_3}{D}, \qquad \frac{\partial \hat{y}}{\partial y} = \frac{\Delta x_2}{D}.$$

So

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{1}{D} \left(\Delta y_3 \frac{\partial \hat{f}}{\partial \hat{x}} - \Delta y_2 \frac{\partial \hat{f}}{\partial \hat{y}} \right), \\ \frac{\partial f}{\partial y} &= \frac{1}{D} \left(-\Delta x_3 \frac{\partial \hat{f}}{\partial \hat{x}} + \Delta x_2 \frac{\partial \hat{f}}{\partial \hat{y}} \right). \end{aligned} \tag{5.6}$$

Combining equations (5.1), (5.5) and (5.6), we can write the finite-element equation (3.3) as integrals over the master element. The entry ij in the element matrix A_k then becomes:

$$\begin{gathered} \epsilon \iint_{\Omega_k} \phi_{i_x} \phi_{j_x} \, d\Omega + \epsilon \iint_{\Omega_k} \phi_{i_y} \phi_{j_y} \, d\Omega + \iint_{\Omega_k} u \phi_{i_x} \phi_j \, d\Omega + \iint_{\Omega_k} v \phi_{i_y} \phi_j \, d\Omega = \\ \frac{\epsilon}{D} \left((\Delta x_3^2 + \Delta y_3^2) \iint_{\hat{\Omega}} \hat{\phi}_{i_{\hat{x}}} \hat{\phi}_{j_{\hat{x}}} \, d\hat{\Omega} - (\Delta x_2 \Delta x_3 + \Delta y_2 \Delta y_3) \iint_{\hat{\Omega}} \left(\hat{\phi}_{i_{\hat{x}}} \hat{\phi}_{j_{\hat{y}}} + \hat{\phi}_{i_{\hat{y}}} \hat{\phi}_{j_{\hat{x}}} \right) d\hat{\Omega} + \right. \\ \left. (\Delta x_2^2 + \Delta y_2^2) \iint_{\hat{\Omega}} \hat{\phi}_{i_{\hat{y}}} \hat{\phi}_{j_{\hat{y}}} \, d\hat{\Omega} \right) + \\ \Delta y_3 \iint_{\hat{\Omega}} \hat{u} \hat{\phi}_{i_{\hat{x}}} \hat{\phi}_j \, d\hat{\Omega} - \Delta y_2 \iint_{\hat{\Omega}} \hat{u} \hat{\phi}_{i_{\hat{y}}} \hat{\phi}_j \, d\hat{\Omega} - \Delta x_3 \iint_{\hat{\Omega}} \hat{v} \hat{\phi}_{i_{\hat{x}}} \hat{\phi}_j \, d\hat{\Omega} + \Delta y_3 \iint_{\hat{\Omega}} \hat{v} \hat{\phi}_{i_{\hat{y}}} \hat{\phi}_j \, d\hat{\Omega}. \end{gathered} \tag{5.7}$$

The first three integrals are the same for every element. Therefore, they are computed only once and multiplied with the correct $(\Delta x_3^2 + \Delta y_3^2)$ etc. for each element, to get the element matrices A_k. The second group of integrals is not the same for each element, but simplifies greatly when a quadrature rule is used, since $\hat{\phi}_j$ is only nonzero in its own control point. Therefore

$$\iint_{\hat{\Omega}} \hat{u} \hat{\phi}_{i_{\hat{x}}} \hat{\phi}_j \, d\hat{\Omega} \approx u(\mathbf{x}_j) \hat{w}_j \hat{\phi}_{i_{\hat{x}}}(\hat{\mathbf{x}}_j), \tag{5.8}$$

etc. All but the velocity in this expression has to be computed only once.

5.2 Isoparametric transformation

The linear transformation can only be used for triangular elements. This means that the control point locations in each element are fully determined by the location of the triangle corner points. Thus, the control points on the element edges are always on a straight line. In some cases, like on curved boundaries, this is undesirable.

Other distributions of the control points can be obtained with a more complex transformation. A polynomial transformation of the same degree p as the element makes it possible to place all points in arbitrary positions, since the number of element points equals the number of parameters in those polynomials. So the transformation F is:

$$\mathbf{x} = F(\hat{\mathbf{x}}) = \sum_{i,j} \mathbf{b}_{i,j} \hat{x}^i \hat{y}^j, \quad i, j \geq 0, \, i + j \leq p. \tag{5.9}$$

The most convenient way to compute this transformation polynomial is to express it in terms of the element basis functions:

$$\mathbf{x} = \sum_{i=1}^{n_e} \mathbf{x}_i \hat{\phi}_i(\hat{\mathbf{x}}). \tag{5.10}$$

The basis polynomials are 1 in their own control points and 0 in all other control points, so this transformation gives the desired position for the control points (see figure 3). As both the solution and the elements are based on the same polynomial basis, we call these elements isoparametric.

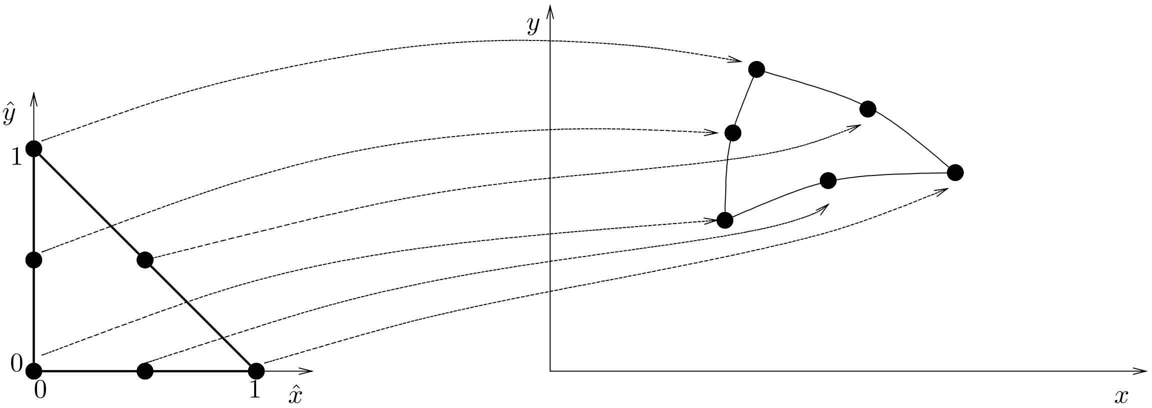


Figure 3: Isoparametric transformation, from master element $\bar{\Omega}$ to Ω.

Integration over isoparametric elements is more involved than over normal elements. However, we can follow the same steps as in the previous section. Writing out equation (5.10) gives:

$$x = \sum_{i=1}^{n_e} x_i \hat{\phi}_i(\hat{x}, \hat{y}), \qquad y = \sum_{i=1}^{n_e} y_i \hat{\phi}_i(\hat{x}, \hat{y}).$$

Differentiating these expressions with respect to x and y, we get:

$$\begin{aligned}
1 &= \left(\sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{x}}}\right) \frac{\partial \hat{x}}{\partial x} + \left(\sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{y}}}\right) \frac{\partial \hat{y}}{\partial x},\\
0 &= \left(\sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{x}}}\right) \frac{\partial \hat{x}}{\partial y} + \left(\sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{y}}}\right) \frac{\partial \hat{y}}{\partial y},\\
0 &= \left(\sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{x}}}\right) \frac{\partial \hat{x}}{\partial x} + \left(\sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{y}}}\right) \frac{\partial \hat{y}}{\partial x},\\
1 &= \left(\sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{x}}}\right) \frac{\partial \hat{x}}{\partial y} + \left(\sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{y}}}\right) \frac{\partial \hat{y}}{\partial y}.
\end{aligned}$$

Solving this system gives:

$$\frac{\partial \hat{x}}{\partial x} = \frac{S_{y\hat{y}}}{D}, \qquad \frac{\partial \hat{y}}{\partial x} = -\frac{S_{y\hat{x}}}{D}, \qquad \frac{\partial \hat{x}}{\partial y} = -\frac{S_{x\hat{y}}}{D}, \qquad \frac{\partial \hat{y}}{\partial y} = \frac{S_{x\hat{x}}}{D}, \tag{5.11a}$$

with

$$S_{x\hat{x}} = \sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{x}}}, \quad S_{x\hat{y}} = \sum_{i=1}^{n_e} x_i \hat{\phi}_{i_{\hat{y}}}, \quad S_{y\hat{x}} = \sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{x}}}, \quad S_{y\hat{y}} = \sum_{i=1}^{n_e} y_i \hat{\phi}_{i_{\hat{y}}}, \tag{5.11b}$$

and

$$D = S_{x\hat{x}} S_{y\hat{y}} - S_{x\hat{y}} S_{y\hat{x}}. \tag{5.11c}$$

This gives an equivalent for equation (5.6):

$$\begin{aligned}\frac{\partial f}{\partial x} &= \frac{1}{D}\left(S_{y\hat{y}}\frac{\partial \hat{f}}{\partial \hat{x}} - S_{y\hat{x}}\frac{\partial \hat{f}}{\partial \hat{y}}\right),\\ \frac{\partial f}{\partial y} &= \frac{1}{D}\left(-S_{x\hat{y}}\frac{\partial \hat{f}}{\partial \hat{x}} + S_{x\hat{x}}\frac{\partial \hat{f}}{\partial \hat{y}}\right).\end{aligned} \tag{5.12}$$

Linearising the transformation (5.10) around (x, y),

$$\begin{aligned}x + dx &= x + S_{x\hat{x}}d\hat{x} + S_{x\hat{y}}d\hat{y},\\ y + dy &= y + S_{y\hat{x}}d\hat{x} + S_{y\hat{y}}d\hat{y},\end{aligned}$$

shows that the determinant D from equation (5.11c) is indeed the area ratio between Ω and $\hat{\Omega}$. But now, this ratio depends on the location. Writing out the integrals in equation (3.3), we obtain an equation like (5.7),

$$\begin{gathered}\epsilon\iint_{\Omega_k}\phi_{i_x}\phi_{j_x}\,d\Omega + \epsilon\iint_{\Omega_k}\phi_{i_y}\phi_{j_y}\,d\Omega + \iint_{\Omega_k}u\phi_{i_x}\phi_j\,d\Omega + \iint_{\Omega_k}v\phi_{i_y}\phi_j\,d\Omega =\\ \epsilon\left(\iint_{\hat{\Omega}}\frac{S_{y\hat{y}}^2}{D}\hat{\phi}_{i_{\hat{x}}}\hat{\phi}_{j_{\hat{x}}}\,d\hat{\Omega} - \iint_{\hat{\Omega}}\frac{S_{y\hat{y}}S_{y\hat{x}} + S_{x\hat{x}}S_{x\hat{y}}}{D}\left(\hat{\phi}_{i_{\hat{x}}}\hat{\phi}_{j_{\hat{y}}} + \hat{\phi}_{i_{\hat{y}}}\hat{\phi}_{j_{\hat{x}}}\right)d\hat{\Omega} + \iint_{\hat{\Omega}}\frac{S_{x\hat{x}}^2}{D}\hat{\phi}_{i_{\hat{y}}}\hat{\phi}_{j_{\hat{y}}}\,d\hat{\Omega}\right)\\ + \iint_{\hat{\Omega}}\hat{u}S_{y\hat{y}}\hat{\phi}_{i_{\hat{x}}}\hat{\phi}_j\,d\hat{\Omega} - \iint_{\hat{\Omega}}\hat{u}S_{y\hat{x}}\hat{\phi}_{i_{\hat{y}}}\hat{\phi}_j\,d\hat{\Omega} - \iint_{\hat{\Omega}}\hat{v}S_{x\hat{y}}\hat{\phi}_{i_{\hat{x}}}\hat{\phi}_j\,d\hat{\Omega} + \iint_{\hat{\Omega}}\hat{v}S_{x\hat{x}}\hat{\phi}_{i_{\hat{y}}}\hat{\phi}_j\,d\hat{\Omega}.\end{gathered} \tag{5.13}$$

These integrals can be evaluated by quadrature. The requirements on the accuracy of the quadrature rule are given in the next section.

In practice, linear and isoparametric elements can be mixed, since the linear transformation itself gives exactly the same results as the isoparametric transformation, applied to element control points that are distributed as for the linear transformation.

And finally, an interesting remark: the linear transformation, applied to elements with linear basis functions, is an isoparametric transformation. Indeed, the formulas given above simplify to the linear formulas when the basis functions have constant derivatives.

6. Accuracy of the method

In order to study the accuracy of the finite-element method, different ways to measure the error are possible, and of interest. This section defines the error norms used here, recalls the theoretical accuracy results found in [1, 2] and extends this analysis to isoparametric elements.

6.1 Error norms

All error norms are based on the local error:

$$e = T - T_h. \tag{6.1}$$

Two integral norms for this error are studied. The first is the Sobolev H^1-norm $\|\cdot\|_{1,\Omega}$, a natural norm for second-order elliptic problems, the second is the L^2-norm $\|\cdot\|_{L^2(\Omega)}$. For test problems, we usually know the exact solution only in the control points. In that case, the integrals in the error norms are approximated with the interpolation based on the known basis functions on the elements, using

$$e \approx \sum e_i\phi_i. \tag{6.2}$$

Then the error norms are approximated as:

$$\|e\|_{L^2(\Omega)} \approx \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} e_i e_j \iint_{\Omega_k} \phi_i \phi_j \, d\Omega \right)^{\frac{1}{2}}, \tag{6.3}$$

and

$$\|e\|_{1,\Omega} \approx \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} e_i e_j \left(\iint_{\Omega_k} \phi_i \phi_j \, d\Omega + \iint_{\Omega_k} \phi_{i_x} \phi_{j_x} \, d\Omega + \iint_{\Omega_k} \phi_{i_y} \phi_{j_y} \, d\Omega \right) \right)^{\frac{1}{2}}. \tag{6.4}$$

The integrals in these expressions can either be computed exactly or by quadrature. The last two integrals in equation (6.4) also appear in (3.3).

6.2 Accuracy with linear transformation

When the elements are triangles and equation (5.7) is used to compute the integrals over the elements, then the order of the solution error can be estimated using the analysis in [1]. In this case, the transformation to the master element is not taken into account for the error estimate: even with the transformation, the basis functions on the elements ϕ_i are polynomials of degree p and the integrals over the elements themselves or over the master element differ only by constant factors. Thus, the error estimate can be made as if polynomial basis functions were integrated over the elements themselves.

The results of this error analysis are, that if the basis and test functions are polynomials of degree p, then:

$$\|e\|_{1,\Omega} \leq Ch^p \|T\|_{p+1,\Omega}, \tag{6.5}$$

with C a constant. Under mild further assumptions, the L^2-error is an order of accuracy better:

$$\|e\|_{L^2(\Omega)} \leq Ch^{p+1} \|T\|_{p+1,\Omega}. \tag{6.6}$$

These results are derived assuming exact evaluation of the integrals.

When a quadrature rule is used, an extra error is made. Suppose a quadrature rule of order $t+1$ is used. If we denote the solution with quadrature by $\tilde{T}_h$, the bilinear functional by B_h and the right-hand side functional by f_h, then it can be found that:

$$|f(\psi_h) - f_h(\psi_h)| \leq Ch^{t+1} \|\psi_h\|_{p,\Delta h}, \tag{6.7a}$$

$$|B(\tilde{T}_h, \psi_h) - B_h(\tilde{T}_h, \psi_h)| \leq Ch^{t+1} \|\tilde{T}_h\|_{p,\Delta h} \|\psi_h\|_{p,\Delta h}. \tag{6.7b}$$

From this result, it is found that:

$$\|T_h - \tilde{T}_h\|_{1,\Omega} = \mathcal{O}(h^p) \qquad \text{when } t \geq 2p-2. \tag{6.8}$$

See also the remarks at the end of section 4.

These results are summarized in table 1. It gives the expected order of the error and the requirement on the accuracy of the quadrature, for the three elements that are used here.

6.3 Accuracy with isoparametric transformation

Isoparametric elements are mostly used to model curved boundaries. This implies that the transformation (5.10) is not different for each element: it is a global transformation that follows the curved boundary and does not change when the grid is refined. We can also say that the isoparametric transformation transforms the problem on the real domain to an equivalent problem on a computational domain with straight boundaries, which is then discretised and solved with normal elements. Thus, the conclusions drawn for the linear transformation in the

Degree of basis functions	Order of the error H^1	 L^2	Req. on quadrature Min. order	 Min. degree
1	1	2	1	0
2	2	3	3	2
3	3	4	5	4

Table 1: Order of the errors and quadrature requirement for three types of elements. The degree of the quadrature rule is the degree of the polynomials that it can integrate exactly.

previous section apply to the isoparametric transformation too, the error estimates and the requirements on the quadrature are the same.

It is important to consider what happens with the shape of the cells when the grid size is reduced. When the cells become smaller, the influence of the higher-order terms in the transformation is reduced, so the cells resemble normal, triangular cells more and more when they become smaller. It is this reduction of the influence of the higher-order terms in the isoparametric transformation that causes the error norms to converge in the same way as with the linear transformation.

On the other hand, in the hypothetical case where the transformation is different for each cell, the accuracy of the solution is greatly reduced. It can be shown that, in this case, the H^1-error is of $\mathcal{O}(h)$, irrespective of the degree of the elements and of the quadrature rule.

In practice, it appears that the error for the isoparametric transform is reduced much when a very accurate quadrature rule is used (although the order of convergence is not changed). Therefore, all integrals on isoparametric elements are computed here with a quadrature rule based on 6th-degree polynomials and on Fekete points [3], that are optimised to approximate the locations for minimal integration errors. This quadrature rule is also given in appendix A.

7. Initial test cases

Before using the method on the main test case, two simpler test problems with known exact solutions are studied. The first is a pure diffusion problem, the second a 1D convection-diffusion problem with a boundary layer.

7.1 Diffusion on a square

This problem is the solution of the 2D Laplace equation on the unit square, with both Dirichlet and Neumann boundary conditions.

$$\begin{aligned} \Delta T &= 0, \qquad x \in [0,1],\, y \in [0,1], \\ T &= -\cos(2\pi x), \qquad y = 0, \\ T &= \cos(2\pi x), \qquad y = 1, \\ T_x &= 0, \qquad x = 0 \text{ and } x = 1. \end{aligned} \tag{7.1}$$

This problem has a known solution,

$$T = \frac{\cos\left(2\pi(x-\frac{1}{2})\right)\sinh\left(2\pi(y-\frac{1}{2})\right)}{\sinh(2\pi)}, \tag{7.2}$$

all very smooth and friendly. The solution is plotted in figure 4.

The problem is solved on two different types of meshes (see figure 5): regular, structured triangular meshes and the same type of meshes with a small, random displacement of the triangle corner points (max. 0.2 times the average point distance). The size of the mesh h is the average distance between two control points lying next to each other in x- or y-direction. So meshes with the same size have the same number of DOF. Thus, a mesh with the same size as another mesh, but with higher-order elements, has less elements.

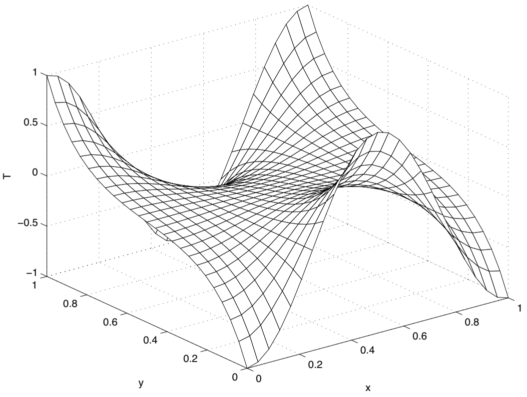


Figure 4: Solution of the diffusion problem (7.1).

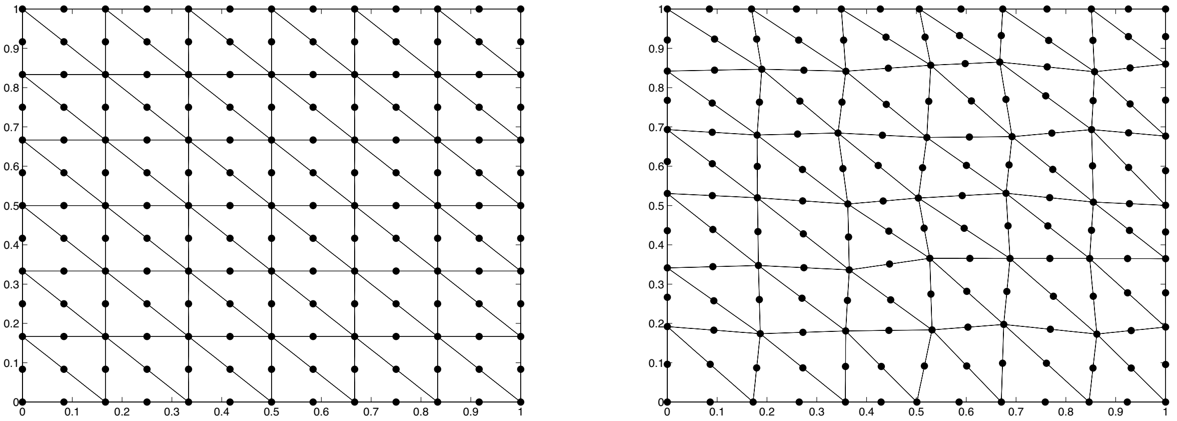


Figure 5: Diffusion problem: an example of a fully structured (left) and a semi-unstructured (right) grid, with quadratic elements.

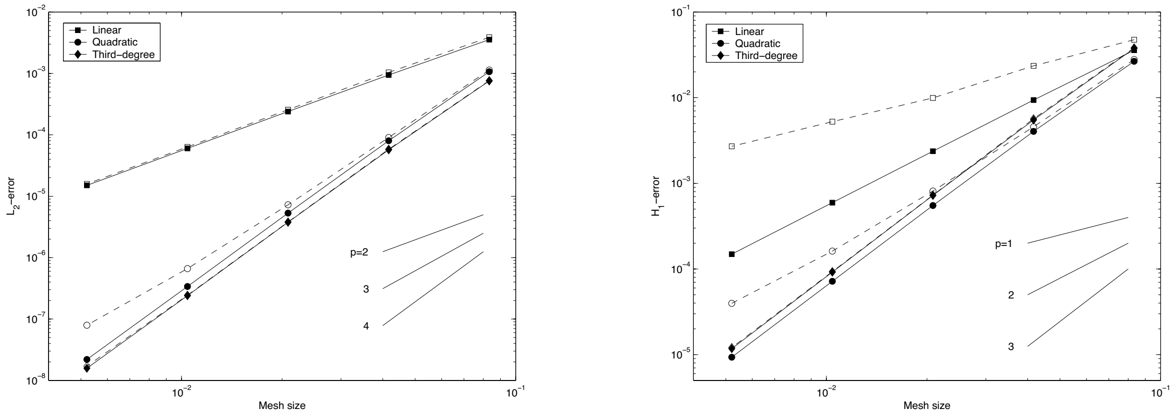


Figure 6: Diffusion problem: L^2- and H^1-error for three types of elements, on fully structured (—) and semi-structured (- - -) grids.

Plots of the errors for five different grid sizes are given in figure 6. These errors were determined with exact evaluation of the integrals in (6.4) and (6.3). Most of the orders of convergence agree with those given in table 1. Only, the H^1-error for linear regular elements and both errors for quadratic elements are one order of h too small (bad, isn't it). But luckily, for fine irregular grids, the quadratic elements show the proper convergence rate. Generally, the difference in the error on structured and unstructured grids is small.

The effect of quadrature is investigated with cubic elements. The error for the fifth-order Lobatto quadrature elements is compared with the error for cubic elements with equidistributed points, in figure 7. These last elements have only fourth-order accurate quadrature, an order less than required. Thus, their errors are an order of h larger than those for Lobatto elements.

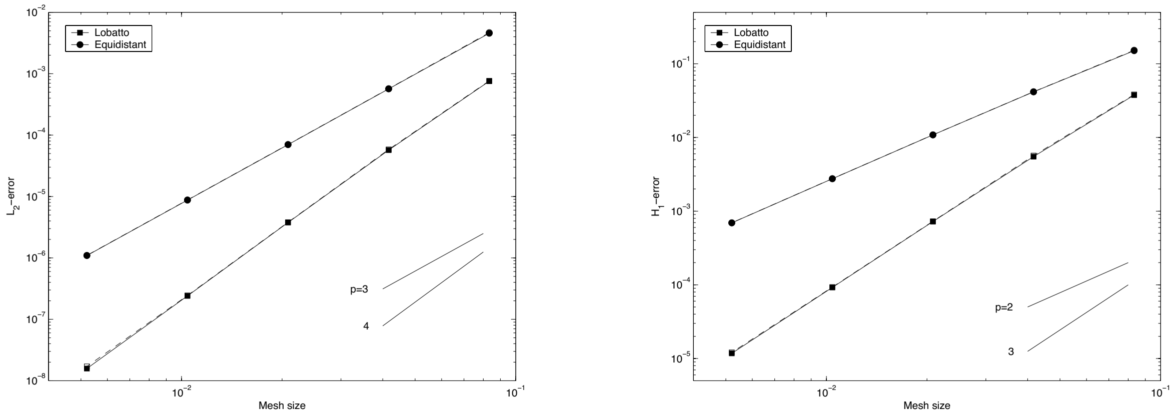


Figure 7: Diffusion problem: L^2- and H^1-error for cubic elements with equidistributed and Lobatto control points.

Isoparametric elements were investigated on two types of grids (see figure 8. The first type is the random-displacement grid that was used before, but now the internal points get a random displacement too (of course, this type of grid will never be used in practice). The second grid type has a global deformation that is independent of the mesh size:

$$\begin{aligned} x_i &= x_{i,\mathrm{reg}} + 0.2\sin(3x_{i,\mathrm{reg}})\sin(y_{i,\mathrm{reg}}), \\ y_i &= y_{i,\mathrm{reg}} + 0.2\sin(x_{i,\mathrm{reg}})\sin(3y_{i,\mathrm{reg}}), \end{aligned}$$

with $x_{i,\mathrm{reg}}$ and $y_{i,\mathrm{reg}}$ the control points for the fully structured grid. This grid is a model for a grid with a curved boundary. As predicted in the previous section, the H^1-errors on the random irregular grids are first-order, independent of the order of the elements (see figure 9). But on the sine-perturbed grids, the accuracy of the regular grids is almost fully recovered. Note that the errors on the regular grids differ a bit from those in figure 6, since all the errors for isoparametric elements are computed with quadrature.

7.2 1D convection-diffusion with boundary layer

The second test problem is used to test the convection part of the method and its behaviour for solutions with boundary layers. This is important because the airfoil problem in the next section also has a boundary layer structure. The problem is one-dimensional, so the solution on a 2D mesh must be constant in y-direction. The velocity field is constant.

$$\begin{aligned} -\epsilon T_{xx} + UT_x &= 0, && x \in [0,1],\ y \in [0,1], \\ T &= T_0, && x = 0, \\ T &= T_1, && x = 1, \\ T_y &= 0, && y = 0 \text{ and } y = 1. \end{aligned} \tag{7.3}$$

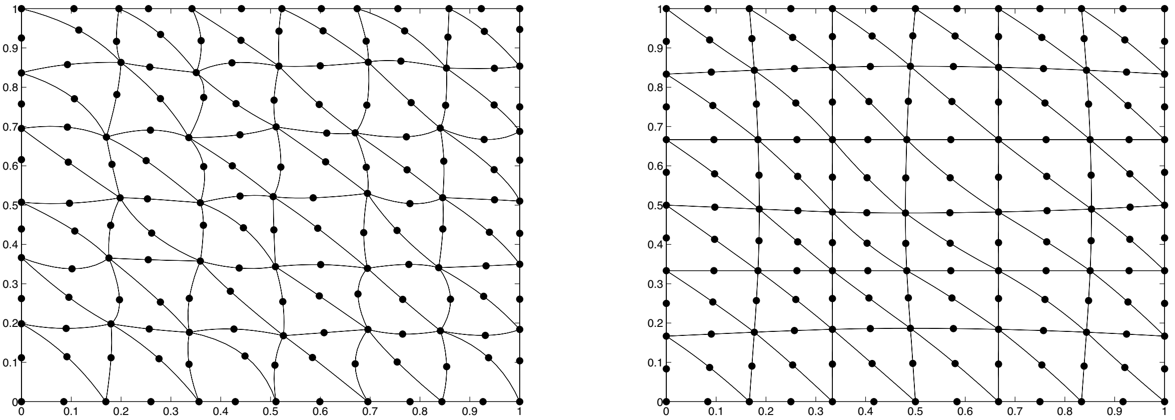


Figure 8: Diffusion problem: isoparametric grids. Semi-structured (left) and sine-perturbed (right) grid with quadratic elements.

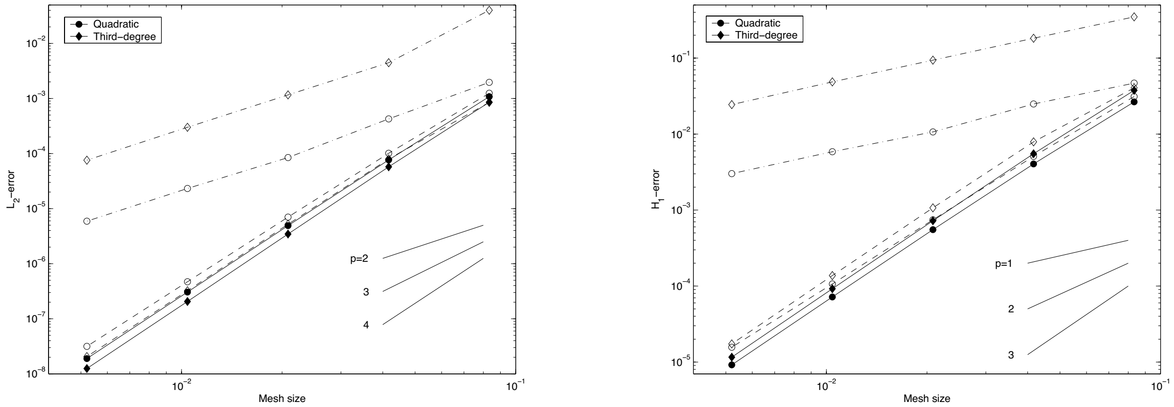


Figure 9: Diffusion problem: L^2- and H^1-error for isoparametric elements, on fully structured (—), semi-structured (-·-) and sine-perturbed (- - -) grids.

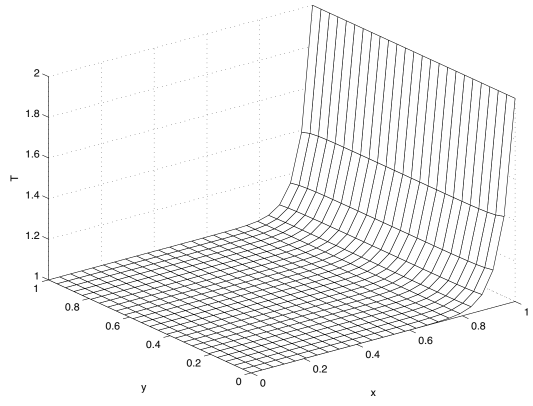


Figure 10: Solution of the diffusion problem (7.3).

The analytical solution is

$$T = \frac{T_0 e^{\frac{U}{\epsilon}} - T_1}{e^{\frac{U}{\epsilon}} - 1} + \frac{T_1 - T_0}{e^{\frac{U}{\epsilon}} - 1} e^{\frac{U}{\epsilon} x}. \tag{7.4}$$

The thickness of the boundary layer decreases with U/ϵ. Here we choose $\epsilon = \frac{1}{20}$, $U = 1$, $T_0 = 1$ and $T_1 = 2$. The solution is given in figure 10.

For fine grids, the error behaviour is the same as for the previous diffusion-only problem, both for normal elements (figure 11) and for isoparametric elements (figure 12). For very coarse grids, some small irregularities appear. These are caused by wiggles in the solution, that appear when the control points are placed so far away from each other that they cannot capture the boundary layer. So for a smooth solotion in a boundary layer, there must be enough degrees of freedom in the layer. It appeared that this number does not depend much on the order of the elements.

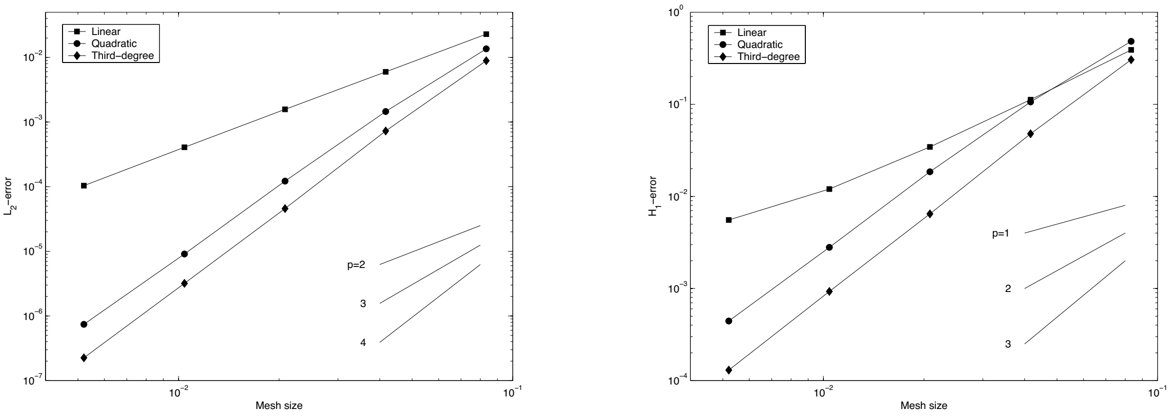


Figure 11: Convection-diffusion problem: L^2- and H^1-error for three types of elements, on fully structured (—) and semi-structured (- - -) grids.

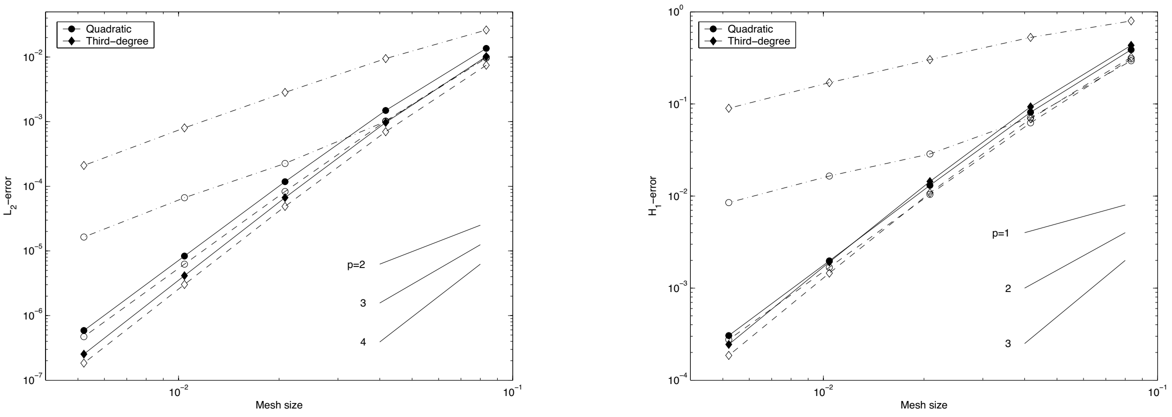


Figure 12: Convection-diffusion problem: L^2- and H^1-error for isoparametric elements, on fully structured (—), semi-structured (-·-) and sine-perturbed (- - -) grids.

8. Flow past a cooled airfoil

In gas turbines very hot air from the combustion chamber is led to the turbine, which drives the compressor in front of the combustor and, possibly, a shaft which leads to the external device powered by the engine. The turbine is spinning very fast, so the loads on the turbine blades are high. The blade material loses its strength at the high temperature of the exhaust gas (about 1700 K at the turbine inlet), so the blades are cooled with internal channels and by blowing cool air out over the blade surface.

A simple model for such a flow is studied here. As said in section 1, we assume incompressible, irrotational flow. This means that the velocity field is given by the potential equation and the temperature by the convection-diffusion heat equation. The turbine is modeled in 2D by an infinite row of airfoils placed above each other. Thus, the problem is symmetric and the flow over half an airfoil only is studied (see figure 13). The top and bottom of the domain are symmetry planes, hence the Neumann boundary conditions. The cooling in the blade is modeled by a single canal with a constant wall temperature. Cooling air blown out is not modeled. The airfoil is a NACA 0012 profile.

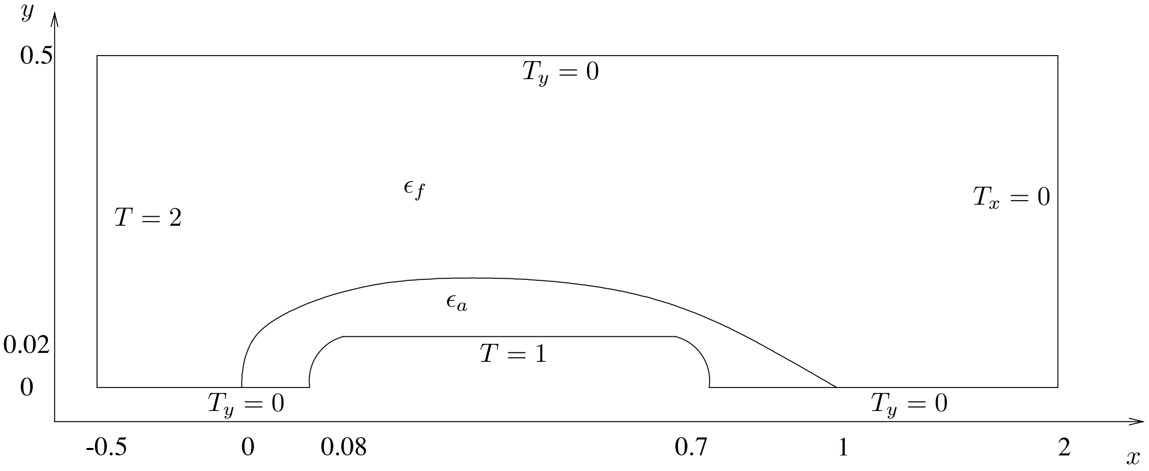


Figure 13: Domain and boundary conditions for cooled-airfoil problem (not to scale).

The problem has two domains: the flow and the airfoil, which are linked with a boundary condition (2.3c). The velocity in the airfoil is zero, so the heat equation (2.2) there reduces to $\Delta T = 0$. Therefore, the effect of different ρC_p in the flow and the airfoil (see section (3)) can be taken into account by using a corrected diffusion coefficient in the airfoil: $\epsilon_a = \frac{k_a}{(\rho C_p)_f}$. So from a mathematical point of view, we just solve equation (2.2) with different values of ϵ in the two domains.

The flow field is computed with a dipole distribution on the airfoil centerline, to make it independent of the mesh. The meshes used are H-type meshes that follow the streamlines and equipotential lines of the flow field. The velocity computation and mesh generation are described in appendix B. The basic mesh has 47690 degrees of freedom in 10432 cubic cells. This mesh can be coarsened up to eight times for a convergence study. The four times coarsened mesh is shown in figure 14, the flow field in figure 15.

An estimate of the convergence behaviour is found in table 2. The exact solution is not known, so we cannot compute the local error exactly. Instead, as a very rough indicator for the convergence of the error, the convergence of the norm of the *solution* is studied. Convergence of this norm does not indicate the order of convergence of the error, but it gives a rough indication that the solution does, in fact, converge.

Now, the physical behaviour of the temperature is studied. The temperature field depends on two parameters only. The problem is linear, so the shape of the temperature distribution depends neither on the inflow temperature T_i, nor on the difference between T_i and the temperature T_p in the cooling duct. The only parameters of interest are:

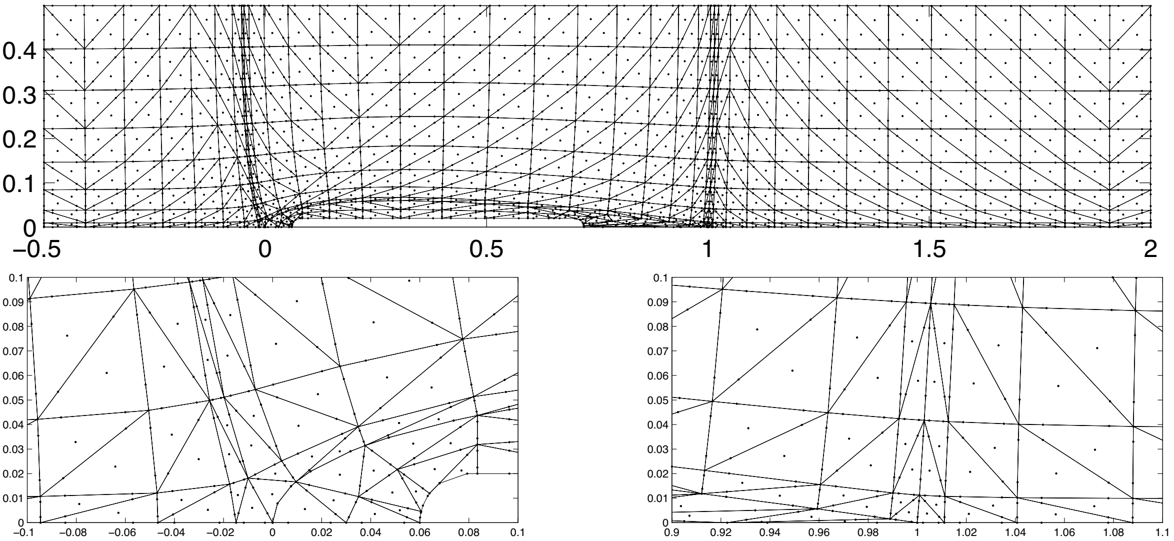
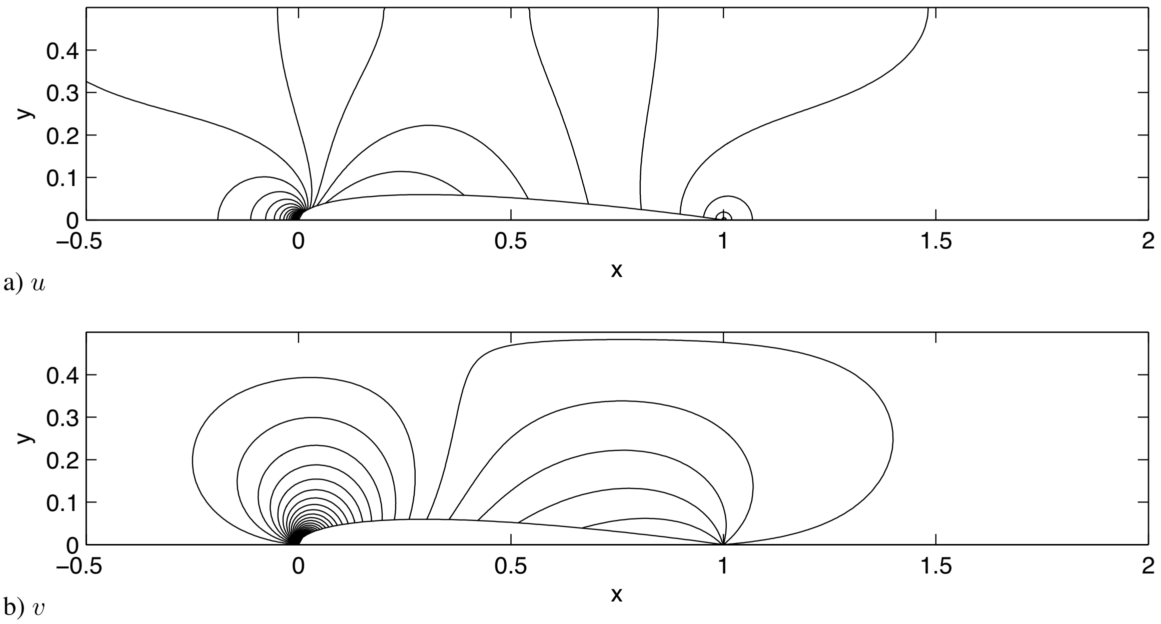


Figure 14: Grid for cooled-airfoil problem (coarsening factor 4), with close-ups of the leading edge and trailing edge.



a) u

b) v

Figure 15: Velocity components u and v. u has minima near the leading and trailing edge and a maximum above the airfoil. Far away from the airfoil, it has the undisturbed value 1. v is positive near the leading edge, negative near the trailing edge and zero far away from the airfoil.

Grids	ΔL^2	ΔH^1
8 – 4	$3.073 \cdot 10^{-4}$	$1.265 \cdot 10^{-2}$
4 – 2	$-2.256 \cdot 10^{-5}$	$4.550 \cdot 10^{-3}$
2 – 1	$-3.518 \cdot 10^{-6}$	$-3.086 \cdot 10^{-4}$

Table 2: Convergence of the difference in L^2- and H^1-norms of the solution on consecutive grids, with coarsening factors from 8 to 1.

1. The ratio of convection and diffusion velocity in the flow,
2. The ratio of diffusion in the flow and in the airfoil.

The first ratio is expressed in the Péclet number[1]:

$$Pe = \frac{U_0 c}{\epsilon_f}, \tag{8.1}$$

with U_0 the undisturbed velocity and c the airfoil chord. For a typical turbine blade, Pe is of the order 10^5 – 10^6. The second ratio is ϵ_a/ϵ_f. For a typical metal airfoil, this is of the order 10^3 – 10^4 (see [4]).

Three properties of the temperature field are of interest:

1. The highest temperature in the airfoil, important for the material strength,
2. The heat flow from the cooling duct, the total cooling power required, which is the integral over the duct boundary of the heat flux $\epsilon_a \frac{\partial T}{\partial n}$,
3. The thickness of the temperature boundary layer, defined here as the place where $T - T_a = 0.95(T_i - T_a)$, at $x = 1$.

In the plots that give the dependence of the tempreature on the Péclet number (figure 16 and 18), we see that the boundary layer thickness decreases with the square root of Pe, as expected. The heat flow from the channel decreases more or less in the same way. This is explicable: for high values of ϵ_a/ϵ_f, the temperature of the airfoil is almost 1. So the temperature gradient in the flow at the airfoil surface is more or less proportional to the boundary layer thickness. And the heat flow from the cooling duct equals the heat flow into the flow domain, that depends on this gradient.

The maximum temperature in the airfoil appears at the leading edge. As Pe increases, the temperature gradient seems to move into the airfoil, so the maximum temperature in the airfoil increases.

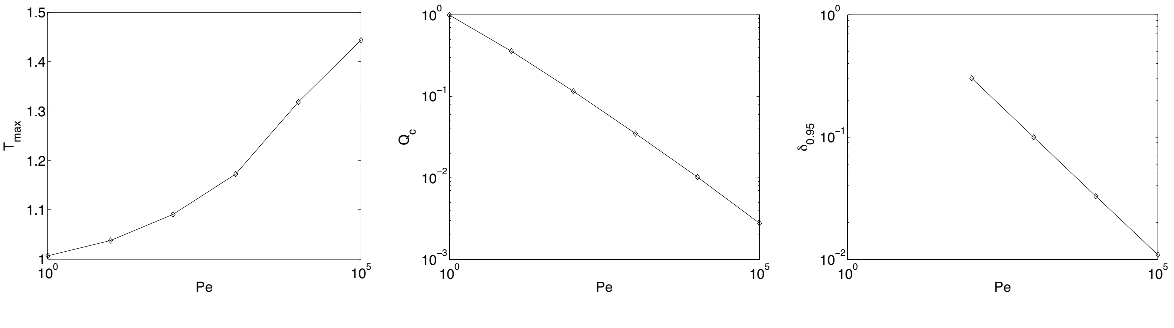


Figure 16: Cooled airfoil: dependence of T_{max} on the airfoil, heat flow Q_c and boundary layer thickness δ on the Peclet number for $\epsilon_a/\epsilon_f = 100$.

[1] For the fluid dynamicists, $Pe = Re\,Pr$.

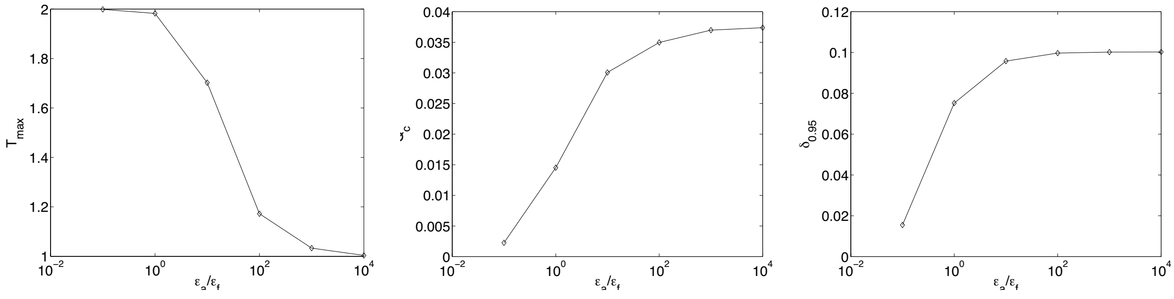


Figure 17: Cooled airfoil: dependence of T_{max} on the airfoil, heat flow Q_c and boundary layer thickness δ on the ratio ϵ_a/ϵ_f for $Pe = 1000$.

Fgures 17 and 19 give the dependence of the temperature field on ϵ_a/ϵ_f. There are two regions: for low ϵ_a/ϵ_f, the temperature gradient region lies mostly in the airfoil, the boundary layer is thin and its thickness varies strongly with ϵ_a/ϵ_f. But when ϵ_a/ϵ_f is high (above 100), the temperature in the airfoil is more or less constant, so the temperature in the flow does not depend much on ϵ_a/ϵ_f. This is the region where the boundary layer depends only on Pe and varies as the square root of Pe.

So, for a real turbine blade ($\epsilon_a/\epsilon_f = 1000$, $Pe = 100.000$), the temperature in the airfoil is mostly constant and the heat flux and boundary layer do not depend much on the airfoil material or the place of the cooling. However, there may be a hot region near the leading edge that requires attention. The boundary layer is thin, much less than the thickness of the airfoil itself.

9. Conclusion

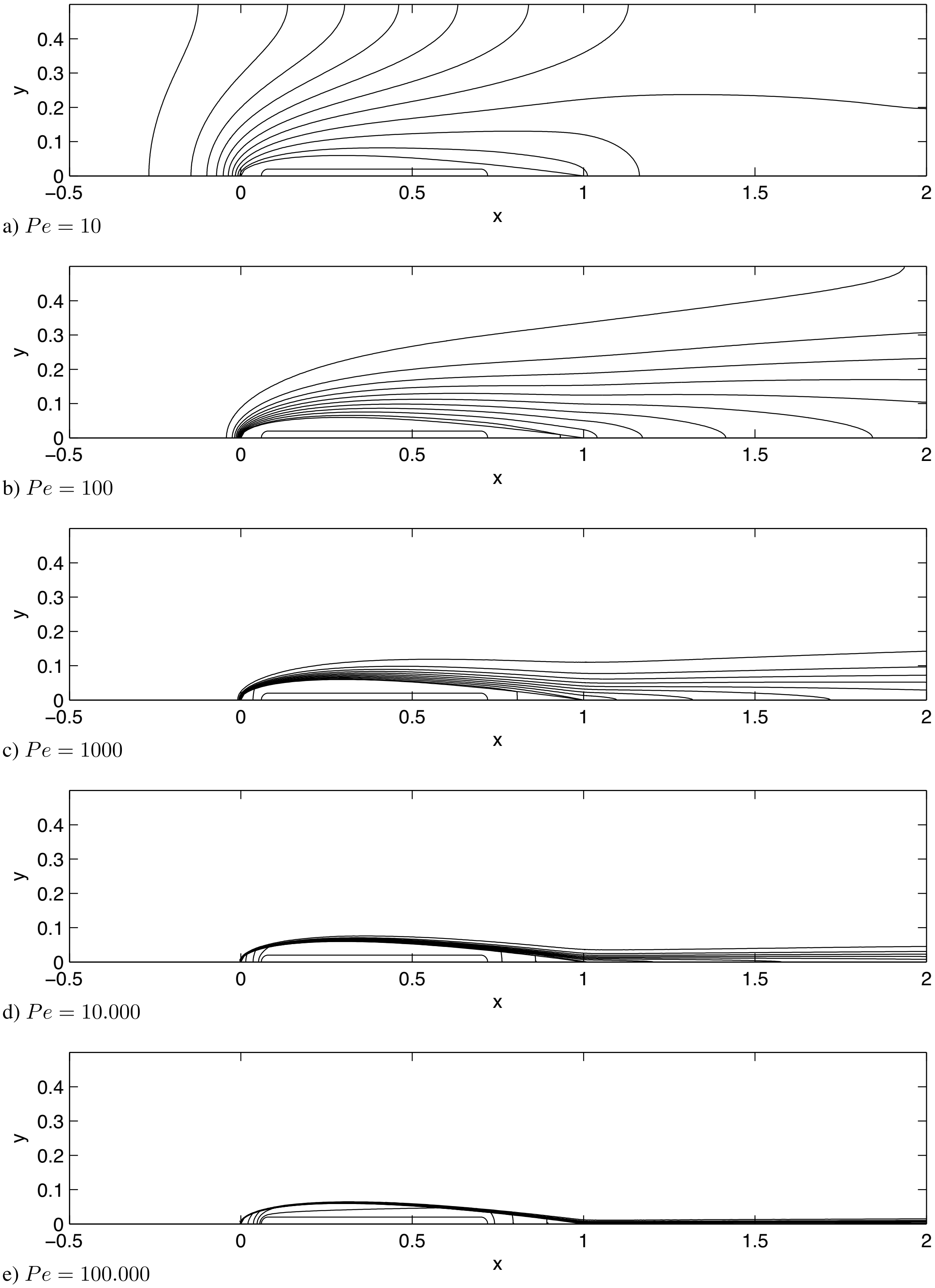
A finite-element method is developed for the 2D convection-diffusion heat equation. For high-order elements, the accuracy of the quadrature rule is important: accurate quadrature rules are presented for triangular elements with up to cubic basis functions. Curved boundaries are handled with isoparametric elements. It is shown that, for smooth curves, the isoparametric elements do not spoil the order of convergence of the error.

Two tests, using simple test problems, show that the norms of the errors converge on grid refinement with the theoretically obtained order for each of the elements. In practice, quadratic elements often show an order of convergence that is even better than the order predicted by theory.

The computation of the temperature around an internally cooled airfoil shows the use of the method for practical problems. For normal airfoils, the temperature gradient is the largest in a thin boundary layer outside the airfoil. However, some regions in the airfoil may still get hot.

References

1. P.W. Hemker. *Lecture Notes Scientific Computing, advanced, 1998. Discretisation of PDEs, Finite-Element Method.* Amsterdam, 1998. http://homepages.cwi.nl/~pieth/LECTURES/lectures04.ps.
2. J.L. Lions (ed.), P.G. Ciarlet. *Handbook of Numerical Analysis, Vol. II: Finite Element Methods, part I.*
3. M.A. Taylor, B.A. Wingate, and R.E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM J. Num. Anal.* **38**, 5 (2000), pp. 1707–1720.
4. G. Verkerk, J.B. Broens, W. Kranendonk, F.J. v.d. Puijl, J.L. Sikkema, C.W. Stam. *Binas, Informatieboek vwo/havo voor het onderwijs in de natuurwetenschappen.* Wolters-Noordhoff, Groningen 1986.



a) $Pe = 10$

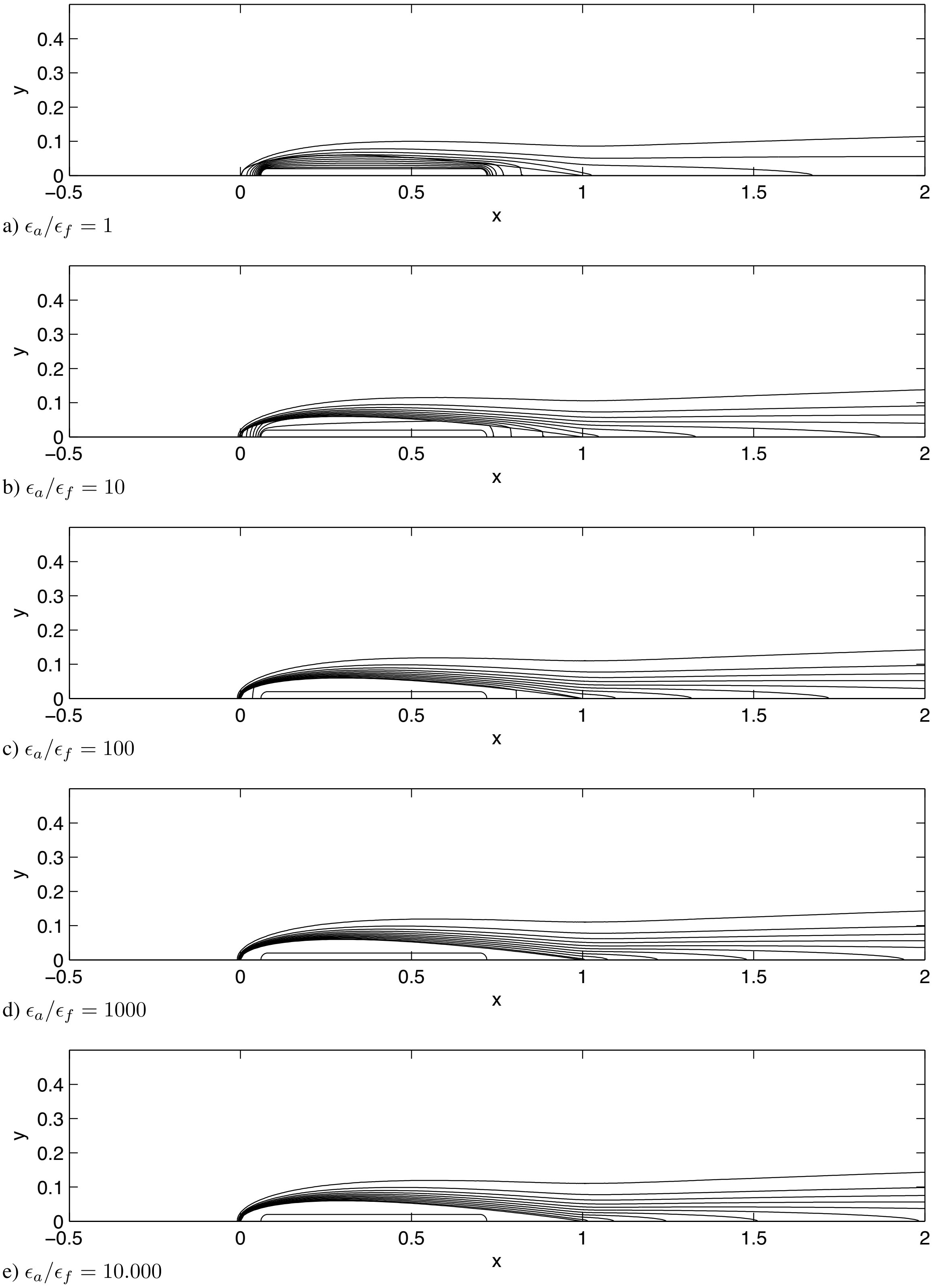
b) $Pe = 100$

c) $Pe = 1000$

d) $Pe = 10.000$

e) $Pe = 100.000$

Figure 18: Cooled airfoil: variation of temperature profile with Péclet number. Constant $\epsilon_a/\epsilon_f = 100$.



a) $\epsilon_a/\epsilon_f = 1$

b) $\epsilon_a/\epsilon_f = 10$

c) $\epsilon_a/\epsilon_f = 100$

d) $\epsilon_a/\epsilon_f = 1000$

e) $\epsilon_a/\epsilon_f = 10.000$

Figure 19: Cooled airfoil: variation of temperature profile with Péclet number. Constant $\epsilon_a/\epsilon_f = 100$.

A. The elements

This appendix gives data for the three types of elements that were used. For each element the control points, the quadrature weights, the basis functions and the derivatives of these basis functions are given. Figures show the shape of the basis functions.

A.1 Linear element

point	x_i	y_i	w_i	basis function
1	0	0	$\frac{1}{6}$	$\phi_1 = -(x + y - 1)$
2	1	0	$\frac{1}{6}$	$\phi_2 = x$
3	0	1	$\frac{1}{6}$	$\phi_3 = y$

Table 3: Control point coordinates, quadrature weights w_i and the basis functions ϕ_i corresponding to each control point.

point	ϕ_{1_x}	ϕ_{2_x}	ϕ_{3_x}	ϕ_{1_y}	ϕ_{2_y}	ϕ_{3_y}
1	-1	1	0	-1	0	1
2	-1	1	0	-1	0	1
3	-1	1	0	-1	0	1

Table 4: The values of the basis function derivatives in each control point.

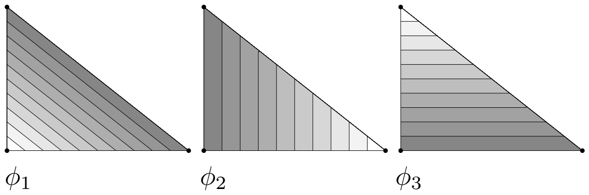


Figure 20: The basis functions.

A.2 Quadratic element

The quadratic basis functions are products of the linear basis functions.

point	x_i	y_i	w_i	basis function
1	0	0	0	$\phi_1 = 2(x+y-1)(x+y-\frac{1}{2})$
2	$\frac{1}{2}$	0	$\frac{1}{6}$	$\phi_2 = -4x(x+y-1)$
3	1	0	0	$\phi_3 = 2x(x-\frac{1}{2})$
4	0	$\frac{1}{2}$	$\frac{1}{6}$	$\phi_4 = -4y(x+y-1)$
5	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	$\phi_5 = 4xy$
6	0	1	0	$\phi_6 = 2y(y-\frac{1}{2})$

Table 5: Control point coordinates, quadrature weights w_i and the basis functions ϕ_i corresponding to each control point.

point	ϕ_{1_x}	ϕ_{2_x}	ϕ_{3_x}	ϕ_{4_x}	ϕ_{5_x}	ϕ_{6_x}	ϕ_{1_y}	ϕ_{2_y}	ϕ_{3_y}	ϕ_{4_y}	ϕ_{5_y}	ϕ_{6_y}
1	-3	4	-1	0	0	0	-3	0	0	4	0	-1
2	-1	0	1	0	0	0	-1	-2	0	2	2	-1
3	1	-4	3	0	0	0	1	-4	0	0	4	-1
4	-1	2	-1	-2	2	0	-1	0	0	0	0	1
5	1	-2	1	-2	2	0	1	-2	0	-2	2	1
6	1	0	-1	-4	4	0	1	0	0	-4	0	3

Table 6: The values of the basis function derivatives in each control point.

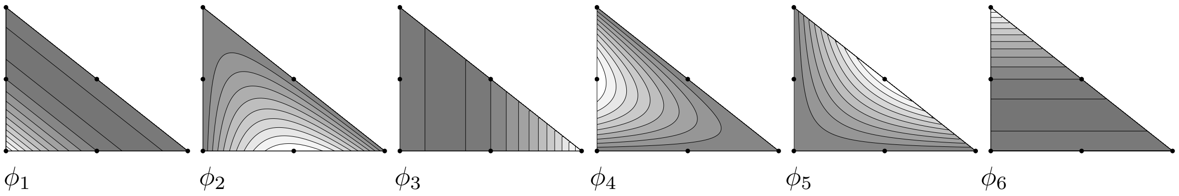


Figure 21: The basis functions.

A.3 Cubic 'Lobatto' element

In this element, the control points are not equidistant. Therefore, not all basis functions are products of linear functions. The derivatives in each control point were not computed analytically and are not given here.

point	x_i	y_i	w_i	basis function
1	0	0	$-\frac{1}{120}$	$\phi_1 = (-6x^2 - 15xy - 6y^2 + 6x + 6y - 1)(x + y - 1)$
2	$\frac{1}{2} - \frac{1}{6}\sqrt{3}$	0	$\frac{1}{20}$	$\phi_2 = 6\sqrt{3}x\left(x + (\frac{1}{2} + \frac{1}{2}\sqrt{3})y - (\frac{1}{2} + \frac{1}{6}\sqrt{3})\right)(x + y - 1)$
3	$\frac{1}{2} + \frac{1}{6}\sqrt{3}$	0	$\frac{1}{20}$	$\phi_3 = -6\sqrt{3}x\left(x - (\frac{1}{2} - \frac{1}{2}\sqrt{3})y - (\frac{1}{2} - \frac{1}{6}\sqrt{3})\right)(x + y - 1)$
4	1	0	$-\frac{1}{120}$	$\phi_4 = x(6x^2 - 3xy - 3y^2 - 6x + 3y + 1)$
5	0	$\frac{1}{2} - \frac{1}{6}\sqrt{3}$	$\frac{1}{20}$	$\phi_5 = 6\sqrt{3}y\left((\frac{1}{2} + \frac{1}{2}\sqrt{3})x + y - (\frac{1}{2} + \frac{1}{6}\sqrt{3})\right)(x + y - 1)$
6	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{9}{40}$	$\phi_6 = -27xy(x + y - 1)$
7	$\frac{1}{2} + \frac{1}{6}\sqrt{3}$	$\frac{1}{2} - \frac{1}{6}\sqrt{3}$	$\frac{1}{20}$	$\phi_7 = (9 + 3\sqrt{3})xy\left(x + (2 - \sqrt{3})y - (1 - \frac{1}{3}\sqrt{3})\right)$
8	0	$\frac{1}{2} + \frac{1}{6}\sqrt{3}$	$\frac{1}{20}$	$\phi_8 = -6\sqrt{3}y\left(-(\frac{1}{2} - \frac{1}{2}\sqrt{3})x + y - (\frac{1}{2} - \frac{1}{6}\sqrt{3})\right)(x + y - 1)$
9	$\frac{1}{2} - \frac{1}{6}\sqrt{3}$	$\frac{1}{2} + \frac{1}{6}\sqrt{3}$	$\frac{1}{20}$	$\phi_9 = (9 + 3\sqrt{3})xy\left((2 - \sqrt{3})x + y - (1 - \frac{1}{3}\sqrt{3})\right)$
10	0	1	$-\frac{1}{120}$	$\phi_{10} = y(-3x^2 - 3xy + 6y^2 + 3x - 6y + 1)$

Table 7: Control point coordinates, quadrature weights w_i and the basis functions ϕ_i corresponding to each control point.

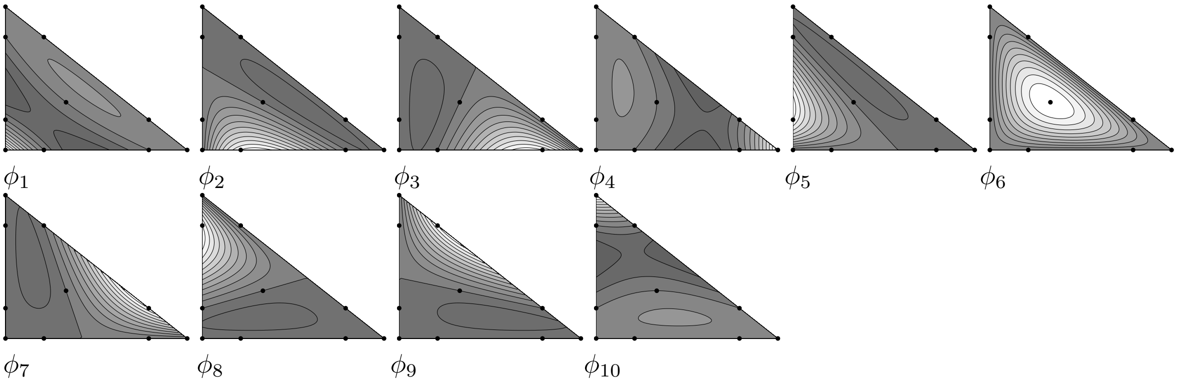


Figure 22: The basis functions.

A.4 Sixth-degree Fekete quadrature points

This high-order quadrature rule, based on sixth-degree polynomials and with 28 control points, is used for isoparametric elements. It is taken from [3].

point	x_i	y_i	w_i
1	0.000000	0.000000	0.000101
2	0.084885	0.000000	0.004824
3	0.265565	0.000000	0.006809
4	0.500000	0.000000	0.008973
5	0.734435	0.000000	0.006809
6	0.915115	0.000000	0.004824
7	1.000000	0.000000	0.000101
8	0.000000	0.084885	0.004824
9	0.106335	0.106335	0.027605
10	0.316270	0.117181	0.044284
11	0.566549	0.117181	0.044284
12	0.787329	0.106335	0.027605
13	0.915115	0.084885	0.004824
14	0.000000	0.265565	0.006809
15	0.117181	0.316270	0.044284
16	0.333333	0.333333	0.054464
17	0.566549	0.316270	0.044284
18	0.734435	0.265565	0.006809
19	0.000000	0.500000	0.008973
20	0.117181	0.566549	0.044284
21	0.316270	0.566549	0.044284
22	0.500000	0.500000	0.008974
23	0.000000	0.734435	0.006809
24	0.106335	0.787329	0.027605
25	0.265565	0.734435	0.006809
26	0.000000	0.915115	0.004824
27	0.084885	0.915115	0.004824
28	0.000000	1.000000	0.000101

Table 8: Control point coordinates and quadrature weights w_i.

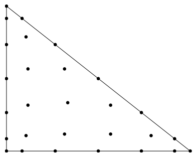


Figure 23: Control points for Fekete quadrature.

B. Flow field and grid generation

The flow field $\mathbf{u}$ in the problem from section 8 is a solution of the potential equation $\Delta\phi = 0$, with $u = \phi_x$ and $v = \phi_y$. So, in principle, the velocity field can be computed with the same finite-element method as the temperature field. However, to avoid non-linear effects in the temperature error, it is desirable to have a velocity field that does not depend on the mesh. Therefore, the velocity field is solved with a boundary integral method: the airfoil is represented by a dipole distribition on its centerline, the strength of these dipoles is computed by specifying that the velocity in control points on the wing surface is tangential to the wing.

The flow field here is computed using 150 dipole panels of equal length and constant dipole strength. The 150 control points are placed above the middles of the panels. Symmetry around the lines $y = 0$ and $y = 0.5$ is obtained (approximately) by mirroring the dipole distribution four times in these lines. Thus, the velocity in the control points depends on the airfoil itself and eight others.

To set up the equations for the dipole strength, we need to know the ecffect of a single panel. One dipole panel of constant strength μ, with length Δx, creates in a point (x_c, y_c) a potential ϕ, a stream function value ψ and velocities u and v:

$$\phi = -\tfrac{1}{2}\mu \ln\left(\frac{y_c^2 + (x_c - x_b - \Delta x)^2}{y_c^2 + (x_c - x_b)^2}\right), \tag{B.1}$$

$$\psi = \mu\left(\arctan\left(\frac{x_c - x_b - \Delta x}{y_c}\right) - \arctan\left(\frac{x_c - x_b}{y_c}\right)\right), \tag{B.2}$$

$$u = \frac{\partial\phi}{\partial x_c} = \mu\left(\frac{x_c - x_b}{(x_c - x_b)^2 + y_c^2} - \frac{x_c - x_b - \Delta x}{(x_c - x_b - \Delta x)^2 + y_c^2}\right), \tag{B.3}$$

$$v = \frac{\partial\phi}{\partial y_c} = -\mu\left(\frac{y_c}{(x_c - x_b)^2 + y_c^2} - \frac{y_c}{(x_c - x_b - \Delta x)^2 + y_c^2}\right). \tag{B.4}$$

The total value of any parameter in a point is found by summing the contributions of all panels. The equations for the dipole strengths μ are found by requiring that $\psi = 0$ in each control point, which means that the flow is tangential to the airfoil.

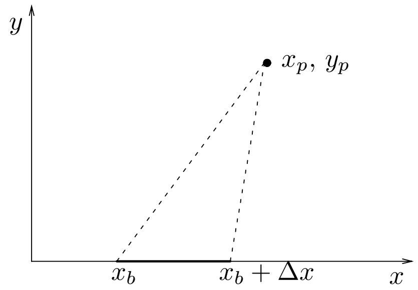


Figure 24: Dipole panel and control point.

Once the potential is found, it can be used in a convenient way to construct a finite-element grid: we can take streamlines and equipotential lines as grid lines. These lines divide the flow domain in a rectangular H-type grid, which is not distorted much, since the streamlines and equipotential ines are orthogonal to each other.

This grid generating procedure proved to be very successful. Only very close to the leading and trailing edge (about 0.001 away) does the potential solution display small wiggles, requiring some smoothing of the grid.