Comparison of linear and quadratic shape functions for a hybrid control-volume finite element method

Y. Stry
University of Applied Sciences, Nürnberg, Germany
M. Hainke
Fraunhofer Institute IIS-B, Schottkystr, Erlangen, Germany
Th. Jung
Institute for Material Science (LS VI), University of Erlangen-Nürnberg, Germany

Keywords *Finite element method, Incompressible flow, Numerical methods, Mesh*

Abstract *A control-volume based method for the numerical calculation of axisymmetric incompressible fluid flow and heat transfer is presented. The proposed method extends the staggered grid approach to unstructured triangular meshes. The velocities are stored at the vertices and the edges of a triangle, pressure and temperature are stored at the vertices. Accordingly, velocities are interpolated in a quadratic way, pressure and temperature linearly. The accuracy of the proposed method is examined for a number of different testproblems. Compared to a linear interpolation scheme implemented in the same code, more accurate solutions and smaller computation times are obtained for the proposed quadratic scheme. The method was designed for and is about to be applied to the numerical simulation of crystal growth.*

1. Introduction

Hybrid control-volume finite element methods (CVFEMs) for fluid flow and heat transfer are constructed by amalgamation and extensions of concepts that are native to finite volume methods like easy physical interpretation and the local and global conservation fullfilled even on coarse grids, and the geometric flexibility traditionally associated with finite element methods. Since the first appearence of CVFEMs, dated already to the late 60s, a lot of different methods have been proposed. The methods differ in the arrangement of the variables like colocated arrangements with special interpolation techniques (Prakash and Patankar, 1985; Rice and Schnipke, 1986) or staggered arrangements (Baliga and Patankar, 1983; Despotis and Tsangaris, 1995; Hookey and Baliga, 1988; Rida *et al.*, 1997) to avoid the well-known effect of a checkerboard pressure field. Different upwind techniques have been proposed like mass weighted

The authors gratefully acknowledge Professer Dr. G. Müller for his continuous interest in the work and support of a publication of the results. We thank the referees for the helpful comments.

International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 12 No. 8, 2002, pp. 1009-1031.

DOI 10.1108/09615530210448732

(MAW) (Schneider and Raw, 1986, 1987a,b), flow oriented (FLO) (Baliga and Patankar, 1983; Prakash and Patankar, 1985) and streamline upwind schemes (Hassan *et al.*, 1983; Raithby, 1976; Swaminathan and Voller, 1992). Furthermore, the methods have been extended to axisymmetric (Masson *et al.*, 1994; Rida *et al.*, 1997) and three-dimensional flows (Costa *et al.*, 1995; Muir and Baliga, 1986; Saabas and Baliga, 1994). A detailed overview of CVFEMs can be found in (Baliga, 1997).

In this work, a method is proposed which extends the staggered grid approach to unstructured triangular meshes. The velocities are stored at the vertices and the edges of a triangle are interpolated in a quadratic way on a triangular grid element. Pressure and temperature are interpolated only linearly on an element, thus the method relates well to the Babuška-Brezzi-condition in a finite element context. An iterative SIMPLE(R)-type algorithm (Patankar and Spalding, 1972), implemented entirely on matrix level, is used for the pressure-velocity coupling. In contrast to former works, the volume and surface integrals of the integral conservation equations are computed exactly by the use of the commercial software package MAPLE. The performance of the proposed method is examined by comparing it to results obtained using a linear interpolation for the velocity components together with the pseudo-velocity interpolation described in Prakash and Patankar (1985) to avoid the checkerboard pressure field. Both interpolation schemes are implemented in the software package CrysVUn, for which a brief description is given later.

The software package CrysVUn was especially designed for global modelling of crystal growth processes (Hainke *et al.*, 2001; Kurz, 1998; Kurz *et al.*, 1999; Metzger, 2000). It contains a physical model for crystal growing, uses the finite volume technique for the discretization of the modelling equations and works with unstructered grids. Several physical phenomena – such as non-linear heat conduction, radiative heat transfer treated with the well known method of view factors and a model for the analysis of thermoelastic stress leading to predictions concerning the quality of the crystals – are implemented. The possibility of inverse modelling allows to control the temperature in an arbitrary number of control points by adjusting the heating power of the heating elements. A graphical user interface allows an easy usage of the program.

Global modelling of 2D/axisymmetric growth facilities requires the triangulation of complex geometries, including crucible, heaters and other constructive elements, see Müller (1998) and Müller and Fischer (2001) for a more detailed discussion. Typical applications of the global modelling approach are the definition of process parameters (Backofen *et al.*, 2000; Metzger and Backofen, 2000) or the determination of boundary conditions for detailed three-dimensional calculations (Derby *et al.*, 2001; Yeckel *et al.*, 2001).

In order to keep the computational effort as small as possible, a high-order interpolation scheme for fluid flow computations may help to reduce the number of required grid elements. Therefore, in this work the performance of a quadratic interpolation scheme in a CVFE context is examined by applying the proposed method to a number of different testproblems.

2. Numerical treatment

2.1 Domain discretization and choice of control volumes

The underlying domain is discretized by an unstructured triangular mesh: First the two-dimensional cross-section of the axisymmetric domain for $\varphi = const$ is divided into a triangular grid. The basic C++ classes – vertex, edge, triangle – for the grid have their origin in the grid generator in Eichenseher (1996).

In the proposed method, velocity components are stored at the vertices and at the edges of the triangles, pressure and temperature only at the vertices. Figure 1 shows a triangular element with vertices I, R and L and edges oI, oR and oL together with the storage position of the unknown velocities u_r, u_φ, u_z and scalars p and T. In order to obtain control volumes, the mid points of the edges are joined and each of the resulting sub-triangles is subdivided by joining the midpoints with the centroid. For the discretization of the continuity equation and the temperature equation, the larger control volumes shown in Figure 2(a) are applied, whereas the discretization of the momentum equations makes use of the smaller control volumes shown in Figure 2(b). So far, the definition of control volumes is equivalent to the micro/macro element arrangement proposed in Baliga and Patankar (1983). The difference is due to the quadratic interpolation of the velocity components on a triangle. If a linear interpolation is chosen, the definition of control volumes as shown in Figure 2(b) could be interpreted as a simple grid refinement. In contrast, the quadratic

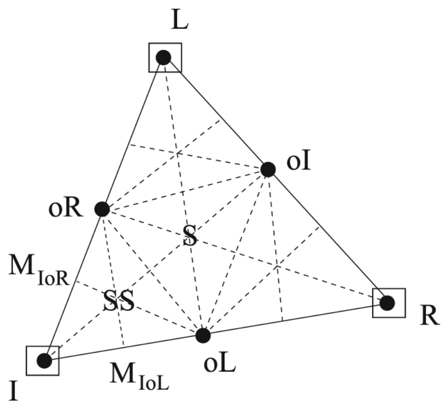


Figure 1. In the case of the quadratic interpolation, the velocity components are stored on vertices and edges (•) of a triangular element, pressure and temperature on vertices (□) only

Ansatz, discussed in Section 2.3, leads to dependencies of all six components inherent to a triangular element.

2.2 Integral conservation equations

In finite-volume methods, the starting point of the discretization is the integral formulation of the governing equations. The equations for 2D ($\alpha = 0$)/axisymetric ($\alpha = 1$) incompressible fluid flow with the Boussinesq approximation read (Baehr and Stephan, 1998; Ferzinger and Perić, 1999). *r-Momentum equation*:

$$\int_{\partial\Omega} [(\rho u_r u_r)\cdot n_r + (\rho u_z u_r)\cdot n_z]\ \mathrm{d}S - \alpha \int_{\Omega} \rho \frac{u_\varphi u_\varphi}{r}\ \mathrm{d}V$$

$$= \int_{\partial\Omega} \left[\left(\mu \frac{\partial u_r}{\partial r} \right)\cdot n_r + \left(\mu \frac{\partial u_r}{\partial z} \right)\cdot n_z \right]\ \mathrm{d}S - \alpha \int_{\Omega} \mu \frac{u_r}{r^2} \mathrm{d}V - \int_{\Omega} \frac{\partial p}{\partial r}\ \mathrm{d}V$$

$$+ \int_{\Omega} s_r\ \mathrm{d}V \tag{1}$$

φ-Momentum equation (only for the axisymmetric case):

$$\int_{\partial\Omega} [(\rho u_r u_\varphi)\cdot n_r + (\rho u_z u_\varphi)\cdot n_z]\ \mathrm{d}S - \int_{\Omega} \rho \frac{u_r u_\varphi}{r}\ \mathrm{d}V$$

$$= \int_{\partial\Omega} \left[\left(\mu \frac{\partial u_\varphi}{\partial r} \right)\cdot n_r + \left(\mu \frac{\partial u_\varphi}{\partial z} \right)\cdot n_z \right]\ \mathrm{d}S - \int_{\Omega} \mu \frac{\mathrm{u}_\varphi}{r^2} \mathrm{d}V + \int_{\Omega} s_\varphi\ \mathrm{d}V \tag{2}$$

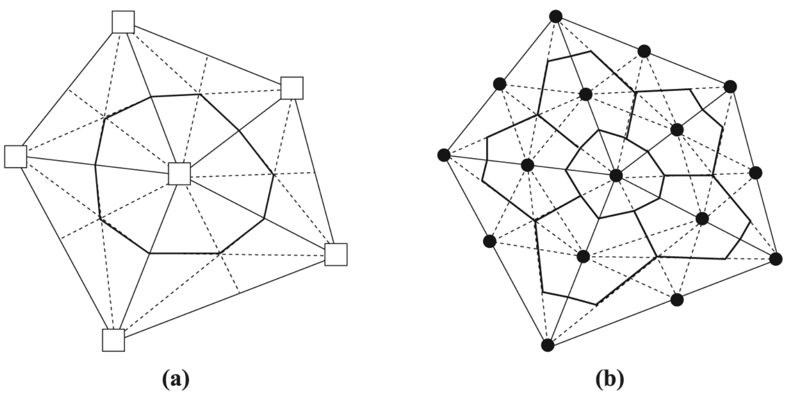


Figure 2.
(a) Large control volumes obtained by joining the centroid of the triangular element with the middle points of the edges (Donald diagram).
(b) Small control volumes constructed with a similar procedure for the sub-triangles as shown in Figure 1

z-Momentum equation:

$$\int_{\partial\Omega}[(\rho u_r u_z)\cdot n_r+(\rho u_z u_z)\cdot n_z]\ \mathrm{d}S = \int_{\partial\Omega}\left[\left(\mu\frac{\partial u_z}{\partial r}\right)\cdot n_r + \left(\mu\frac{\partial u_z}{\partial z}\right)\cdot n_z\right]\ \mathrm{d}S$$

$$-\int_{\Omega}\frac{\partial p}{\partial z}\ \mathrm{d}V - \int_{\Omega}\beta g_z\rho(T-T_{ref})\ \mathrm{d}V + \int_{\Omega}s_z\ \mathrm{d}V \tag{3}$$

Continuity equation:

$$\int_{\partial\Omega}[(\rho u_r)\cdot n_r + (\rho u_z)\cdot n_z]\ \mathrm{d}S = 0 \tag{4}$$

Temperature equation:

$$\int_{\partial\Omega}[(c_p\rho u_r T)\cdot n_r + (c_p\rho u_z T)\cdot n_z]\ \mathrm{d}S = \int_{\partial\Omega}\left[\left(\lambda\frac{\partial T}{\partial r}\right)\cdot n_r + \left(\lambda\frac{\partial T}{\partial z}\right)\cdot n_z\right]\ \mathrm{d}S$$

$$+\int_{\Omega}s_T\ \mathrm{d}V \tag{5}$$

In these equations, the following abbreviations hold: $\mathrm{d}V = (2\pi r)^{\alpha}\mathrm{d}A$ and $\mathrm{d}S = (2\pi r)^{\alpha}\mathrm{d}l$. Here, $\mathrm{d}V$ denotes the volume of the control volume, $\mathrm{d}A$ is the area of the corresponding two-dimensional cross-section, $\mathrm{d}S$ stands for the surface of the control volume and dl for its one-dimensional counterpart. The velocity components in r-, φ- and z- direction are u_r, u_φ and u_z, p denotes the pressure of the fluid, T its temperature, s_r, s_φ, s_z and s_T stand for volumetric source terms. Constant material properties are the density ρ, the dynamic viscosity μ, the acceleration due to gravity g_z, the volumetric expansion coefficient β, the specific heat c_p and the heat conductivity λ. The vector (n_r, n_z) is the outward normal. The factor 2π is omitted in all equations in the sequel.

2.3 Interpolation functions

The derivation of algebraic approximations to the integral formulation of the conservation equations requires the specification of element based interpolation functions for the dependent variables u_r, u_φ, u_z, p and T. In order to simplify the formulation of the resulting coefficients, a local (γ, δ) coordinate system is defined as stated in Figure 3.

The interpolation functions are expressed with regard to this local co-ordinate system. Quadratic interpolation for the unknown ϕ reads

$$
\begin{aligned}
\phi^{\text{quad}} &:= \phi^{\text{quad}}(\gamma, \delta): \\
&= (4\gamma - 4\gamma\delta - 4\gamma^2)\phi_{oL} + (-4\gamma\delta + 4\delta - 4\delta^2)\phi_{oR} + 4\gamma\delta\phi_{oI} \\
&\quad + (2\delta^2 - \delta)\phi_L + (2\gamma^2 - \gamma)\phi_R + (1 - 3\gamma - 3\delta + 2\gamma^2 + 4\gamma\delta \\
&\quad + 2\delta^2)\phi_I, \qquad (6)
\end{aligned}
$$

whereas linear interpolation reads

$$\phi^{\text{lin}} := \phi^{\text{lin}}(\gamma, \delta) := (1 - \gamma - \delta)\phi_I + \gamma\phi_R + \delta\phi_L. \qquad (7)$$

The partial derivatives of ϕ^{quad} and ϕ^{lin}, e.g. $\partial\phi^{\text{quad}}/\partial r$ and $\partial\phi^{\text{lin}}/\partial z$, read

$$
\begin{aligned}
\frac{\partial\phi^{\text{quad}}}{\partial r} &= [(4 - 4\delta - 8\gamma)\phi_{oL} - 4\delta\phi_{oR} + 4\delta\phi_{oI} + (4\gamma - 1)\phi_R \\
&\quad + (-3 + 4\gamma + 4\delta)\phi_I] \cdot \frac{-z_L + z_I}{\Delta V} + [-4\gamma\phi_{oL} + (-4\gamma + 4 - 8\delta)\phi_{oR} \qquad (8) \\
&\quad + 4\gamma\phi_{oI} + (-1 + 4\delta)\phi_L + (-3 + 4\gamma + 4\delta)\phi_I] \cdot \frac{z_R - z_I}{\Delta V}
\end{aligned}
$$

and

$$\frac{\partial\phi^{\text{lin}}}{\partial z} = \frac{(\phi_R - \phi_I)\cdot(r_L - r_I)}{\Delta V} + \frac{(\phi_L - \phi_I)\cdot(r_I - r_R)}{\Delta V}, \qquad (9)$$

where

$$\Delta V := (z_R - z_I)\cdot(r_L - r_I) - (z_L - z_I)\cdot(r_R - r_I) \qquad (10)$$

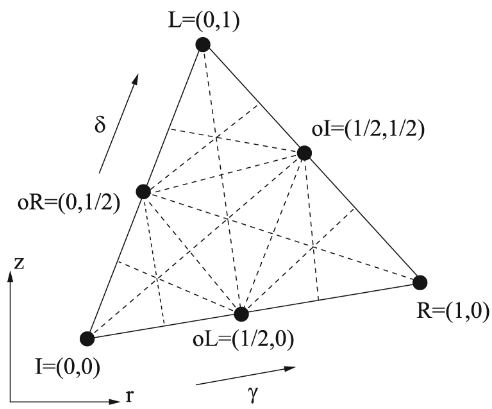


Figure 3.
Local coordinate system for the formulation of interpolation functions and the parametrization of the control volumes

denotes the negative of twice the area of the triangle with vertices I, R and L, the (r, z) co-ordinates of which being (r_I, z_I), (r_R, z_R) and (r_L, z_L) respectively.

2.4 Parametrization of control volumes

In order to obtain approximations for surface integrals in equations (1-5), lines representing control volume surfaces in the two-dimensional cross-section of the three-dimensional control volume have to be parametrized, e.g. in case of the larger control volumes the line $oL \rightarrow S$ (cf. Figure 1) can be written in the local co-ordinate system (γ, δ) with the parametrization factor τ

$$\left\{ \begin{array}{lllll} \gamma & := & \gamma(\tau) & := & \frac{1}{2} - \frac{1}{6}\tau, & \tau \in [0, 1] \\ \delta & := & \delta(\tau) & := & \frac{1}{3}\tau, & \tau \in [0, 1] \end{array} \right\}, \tag{11}$$

the calculation of the vector normal yields

$$(n_r, n_z) = \left(\frac{2z_L - z_I - z_R}{6}, -\frac{2r_L - r_I - r_R}{6} \right). \tag{12}$$

For the smaller control volumes, e.g. the line $M_{IoL} \rightarrow SS$ (cf. Figure 1) can be written in the local co-ordinate system (γ, δ)

$$\left\{ \begin{array}{llllll} \gamma & := & \gamma(\tau) & := & -\frac{1}{12}\tau + \frac{1}{4}, & \tau \in [0, 1] \\ \delta & := & \delta(\tau) & := & \frac{1}{6}\tau, & \tau \in [0, 1] \end{array} \right\}, \tag{13}$$

the calculation of the vector normal yields

$$(n_r, n_z) = \left(\frac{2z_L - z_I - z_R}{12}, -\frac{2r_L - r_I - r_R}{12} \right) \tag{14}$$

In order to obtain approximations for volume integrals in equations (1-5), areas representing volumes in the two-dimensional cross-section of the three-dimensional control volume have to be parametrized, e.g. the triangle $\triangle(I, M_{IoL}, SS)$ (cf. Figure 1) can be written in the local co-ordinate system (γ, δ) employing the parametrization factors τ and σ

$$\left\{ \begin{array}{llllll} \gamma & := & \gamma(\tau, \sigma) & := & \frac{1}{4}\tau - \frac{1}{12}\sigma & \tau \in [0, 1], \sigma \in [0, \tau] \\ \delta & := & \delta(\tau, \sigma) & := & \frac{1}{6}\sigma & \tau \in [0, 1], \sigma \in [0, \tau] \end{array} \right\}. \tag{15}$$

The functional determinant of the coordinate transformation is given by

$$FD_{\triangle(I,M_{IoL},SS)} = FD_{rz\to\gamma\delta}\cdot FD_{\gamma\delta\to\tau\gamma}$$

$$= (r_R\cdot z_L - r_R\cdot z_I - r_I\cdot z_L - r_L\cdot z_R + r_L\cdot z_I + r_I\cdot z_R)\cdot 1/24. \quad (16)$$

2.5 Approximation of volume and surface integrals

In the sequel, only the discretization of the z-momentum equation is explained, as the discretization of the other equations is similar.

As axisymmetry ($\alpha = 1$) in these equations yields an additional dependency on the radius, the radius r in the (γ, δ) co-ordinate system

$$r = r(\gamma, \delta) = r_I + \gamma(r_R - r_I) + \delta(r_L - r_I) \quad (17)$$

will occur in most of the following equations.

2.5.1 Diffusive terms. The approximation of the diffusive term of equation (3) for line $M_{IoL} \to SS$ reads

$$\int_{M_{IoL}\to SS}\left[\left(\mu\frac{\partial u_z}{\partial r}\right)\cdot n_r + \left(\mu\frac{\partial u_z}{\partial z}\right)\cdot n_z\right]\mathrm{d}S$$

$$\approx \int_{\tau=0}^{1}\mu\cdot r^{\alpha}(\gamma(\tau),\delta(\tau))\cdot\frac{\partial u_z}{\partial r}(\gamma(\tau),\delta(\tau))\cdot n_r\;\mathrm{d}\tau \quad (18)$$

$$+ \int_{\tau=0}^{1}\mu\cdot r^{\alpha}(\gamma(\tau),\delta(\tau))\cdot\frac{\partial u_z}{\partial z}(\gamma(\tau),\delta(\tau))\cdot n_z\;\mathrm{d}\tau$$

with r defined in equation (17), $\gamma(\tau)$ and $\delta(\tau)$ defined in equation (13), $\partial u_z/\partial r$ calculated in equation (8) – substitute ϕ by u_z – and $\partial u_z/\partial z$ calculated analogously, the outer normal (n_r, n_z) taken from equation (14).

2.5.2 Convective terms. A straightforward central discretization of the convective part of the z-momentum equation (3) for line $M_{IoL} \to SS$ reads

$$\rho\int_{M_{IoL}\to SS}[(u_r u_z)\cdot n_r + (u_z u_z)\cdot n_z]\;\mathrm{d}S$$

$$\approx \rho\int_{\tau=0}^{1} u_r^{\mathrm{old}}\;\; u_z(\gamma(\tau),\delta(\tau))\;\; r^{\alpha}(\gamma(\tau),\delta(\tau)\cdot n_r\;\mathrm{d}\tau \quad (19)$$

$$+ \rho\int_{\tau=0}^{1} u_z^{\mathrm{old}}\;\; u_z(\gamma(\tau),\delta(\tau))\;\; r^{\alpha}(\gamma(\tau),\delta(\tau))\cdot n_z\;\mathrm{d}\tau$$

with r defined in equation (17), $\gamma(\tau)$ and $\delta(\tau)$ defined in equation (13), u_z calculated in equation (6) – substitute ϕ by u_z –, the outer normal (n_r, n_z) taken from equation (14). The old velocities u_r^{old} and u_z^{old} can be defined by equation (6) – substitute ϕ by u_r^{old} or u_z^{old} and take the old values of u_r and u_z (cf. SIMPLE(R) algorithm, Section 3).

2.5.3 Volume integrals. The treatment of volume integrals is demonstrated for the pressure gradient in equation (3). The approximation for other volume terms is similar.

$$\int_{\triangle(I,M_{IoL},SS)} \frac{\partial p}{\partial z}\,\mathrm{d}V \approx FD_{\triangle(I,M_{IoL},SS)} \cdot \int_{\tau=0}^{1}\int_{\sigma=0}^{\tau} r^{\alpha}(\gamma(\tau,\sigma),\delta(\tau,\sigma))\frac{\partial p}{\partial z}\,\mathrm{d}\sigma\mathrm{d}\tau \quad (20)$$

with r defined in equation (17), $\gamma(\tau, \sigma)$ and $\delta(\tau, \sigma)$ defined in equation (15), $\partial p/\partial z$ calculated in equation (9) – substitute ϕ by p – $FD_{\triangle(I,\, M_{IoL},\, SS)}$ given by equation (16).

2.6 Derivation of linear equations

In order to derive linear equations between the unknowns, all integrals in the preceding section are evaluated and inserted into the integral conservation equations (1-5). For example, the z-momentum equation discretized on $M_{IoL} \rightarrow S_I$ then yields an equation between the unknowns $u_{z,I}, u_{z,R}, u_{z,L}, u_{z,oI}, u_{z,oR}, u_{z,oL}$ in the convective and the diffusive part and p_I, p_R, p_L in the pressure gradient, the convective and diffusive part reading

$$\begin{aligned}
&\int_{M_{IoL}\rightarrow S_I} \left[(\rho u_r^0 u_z)\cdot n_r + (\rho u_z^0 u_z)\cdot n_z\right]\ \mathrm{d}S \\
&\quad - \int_{M_{IoL}\rightarrow S_I} \left[\left(\mu\frac{\partial u_z}{\partial r}\right)\cdot n_r + \left(\mu\frac{\partial u_z}{\partial z}\right)\cdot n_z\right]\ \mathrm{d}S \\
&\quad \approx C_I^z \cdot u_{z,\mathrm{I}} + C_R^z \cdot u_{z,R} + C_L^z \cdot u_{z,L} + C_{oI}^z \cdot u_{z,oI} + C_{oR}^z \cdot u_{z,oR} + C_{oL}^z \cdot u_{z,oL} \quad (21)
\end{aligned}$$

The coefficients C_i^z appearing in the linear equations are technically rather complex, but not complicated. They were calculated using the mathematical software Maple. All these coefficients form entries in several matrices (cf. Sections 2.7 and 2.9).

2.7 Assemblation of discretization coefficients

For each node in the calculation domain (internal node or boundary node), where the dependent variable, u_r, u_φ, u_z, p or T, is unknown, the appropriate conservation principle has to be applied to the control volume surrounding this node. In case of e.g. node I in Figure 1, the integral conservation equations (1-3), when applied to the smaller control volumes (larger control volumes in case of the continuity equation (4) and the temperature equation (5)) can be written as follows:

$$
\begin{aligned}
&\int_{M_{IoL} \to SS} \text{surface terms } \mathrm{d}S + \int_{\triangle(I, M_{IoL}, SS)} \text{volume terms } \mathrm{d}V \\
&\quad + (\text{analogous surface term integration on line } SS \to M_{IoR}) \\
&\quad + (\text{analogous volume term integration on triangle } I, SS, M_{IoR}) \\
&\quad + (\text{similar contributions from other triangular elements associated} \\
&\qquad \text{with vertex } I) \\
&\quad + (\text{boundary conditions, if applicable}) \\
&\quad = 0
\end{aligned} \tag{22}
$$

The form of equation (22) emphasizes that the discretization coefficients can be assembled by scanning all triangles in the calculation domain element-by-element and storing the discretization coefficients of surface and volume terms in the appropriate row of the discretization matrix. The resulting discretization matrix is sparse. The assemblation is facilitated by pointer structures and degrees of freedom implemented in C++.

2.8 Boundary conditions

The treatment of Dirichlet boundary conditions is straightforward, e.g. in case of equation (21) and a fixed velocity value at vertex R, the term $C_R^z \cdot u_{z,R}$ is moved to the right hand side of the equation, instead of C_R^z being stored in the matrix. At the axis of symmetry u_r and u_φ are set to zero.

In the case of line $I \to L$ (cf. Figure 1) being part of an outflow boundary, additional surface terms for the volumetric balance of e.g. vertex I, have to be taken into account. For example, the contribution to the continuity equation along line $I \to oL$ reads

$$
\begin{aligned}
&\int_{I \to oL} [(\rho u_r) \cdot n_r + (\rho u_z) \cdot n_z] \ \mathrm{d}S \\
&\quad \approx \int_{\tau=0}^{1} \rho \ u_r(\gamma(\tau), \delta(\tau)) \ r^{\alpha}(\gamma(\tau), \delta(\tau)) \cdot n_r \ \mathrm{d}\tau \\
&\quad + \int_{\tau=0}^{1} \rho \ u_z(\gamma(\tau), \delta(\tau)) \ \mathrm{r}^{\alpha}(\gamma(\tau), \delta(\tau)) \cdot n_z \ \mathrm{d}\tau
\end{aligned} \tag{23}
$$

with a parametrization of line $I \rightarrow oL$ for γ, δ, n_r, n_z similar as described in Section 2.4.

In the case of symmetry conditions no additional fluxes appear through the specified boundaries, thus no additional terms have to be taken into account.

2.9 Resulting matrix equations

Let in the following, N denote the number of unknown velocity values in the calculation domain and M stand for the number of p-unknowns.

Momentum equations. The assemblation of all convective and diffusive terms in the momentum equations results into an $N \times N$-matrix A and an $N \times 1$-vector $rh\vec{s}A$ containing Dirichlet boundary data of the velocity components.

The assemblation of pressure gradient terms yields an $N \times M$-matrix B. The dimension N here is due to the use of the smaller control volumes for the discretization of the momentum equations, dimension M reflects that there are less p-unknowns than u_r-, u_φ- or u_z-unknowns.

With the $N \times 1$-vector $u\vec{v}$ denoting the unknown velocities at the vertices and edges of the triangular elements in the calculation domain and the $M \times 1$-vector $\vec{p}$ denoting the unknown pressure terms, the matrix equation which stands for the discretization of the momentum equations (1-3) finally reads:

$$A \cdot u\vec{v} = rh\vec{s}A + B \cdot \vec{p} \tag{24}$$

Continuity equation. Assemblation of all surface terms in the continuity equation (volume terms do not occur) results into an $M \times N$-matrix C and an $M \times 1$-vector $rh\vec{s}C$ containing Dirichlet boundary data of u_r, u_φ and u_z. The dimension M here is due to the use of the larger control volumes for the discretization of the continuity equation, dimension N reflects that the unknown velocities are located both at the vertices and the edges.

With the $N \times 1$-vector $u\vec{v}$ denoting the unknown velocities, the matrix equation standing for the discretization of the continuity equation finally reads:

$$C \cdot u\vec{v} = rh\vec{s}C \tag{25}$$

Temperature equation. The storage of the discretization coefficients of the convective terms has been adapted to the solution procedure used in Kurz, (1998) for heat conduction and is not further discussed at this point.

3. Solution methods

The solution methods for the resulting set of coupled equations (24) and (25) (and temperature equation) are based on the SIMPLE resp. SIMPLER algorithm (Patankar, 1980). This means that the equations are solved sequentially until convergence, the non-linearity and the coupling of variables being taken into account by outer iterations. For the solution of the linear equation systems the CrysVUn-user can choose from a range of solvers and preconditioners taken from the packages SuperLU and Spooles.

For the inner iterations different solvers may be used for momentum and pressure: the best performance was found for an iterative solver (BiCSTAB) for the momentum equation, whereas a direct solver (GSSV) for the pressure (-correction in the case of SIMPLE) equation is used.

3.1 Simple algorithm

The SIMPLE solution algorithm is implemented algebraically, i.e. purely on a matrix level. It starts with the solution of the momentum equation (24) with an underrelaxation according to (Patankar, 1980), taking the old or initially guessed pressure field. As the obtained velocity field $\vec{uv}^*$ does not fullfill continuity (equation 25), velocities and pressure have to be corrected

$$\vec{uv} = \vec{uv}^* + \vec{uv}_{\mathrm{corr}}, \tag{26}$$

$$\vec{p} = \vec{p}^* + \vec{p}_{\mathrm{corr}}. \tag{27}$$

By inserting the corrected values in equation (24) and subtracting the equation from the momentum equation with the preliminary velocity field, an expression for the velocity correction is obtained

$$\vec{uv}_{\mathrm{corr}} = A_D^{-1} \cdot B \cdot \vec{p}_{\mathrm{corr}}, \tag{28}$$

where only the diagonal elements of A are taken into account. Inserting the corrected velocities in equation (25) yields after re-arranging an expression for the pressure correction

$$C \cdot A_D^{-1} \cdot B \cdot \vec{p}_{\mathrm{corr}} = rh\vec{s}C - C \cdot \vec{uv}, \tag{29}$$

thus after solution of equation (29) and subsequent underrelaxation, corrected velocities and pressure are obtained according to equations (26) and (27).

3.2 SIMPLER algorithm

A possibility to compute directly the pressure field is given by the SIMPLER algorithm, which in our matrix notation can be written as follows:

We start from the momentum-equation (24), replacing the matrix A by the sum of its diagonal elements (A_D) and its off-diagonal-elements (A_{ND}):

$$(A_D + A_{ND}) \cdot \vec{uv} = rh\vec{s}A + B \cdot \vec{p} \tag{30}$$

which can be rearranged as

$$\vec{uv} = A_D^{-1} \cdot (rh\vec{s}A - A_{ND} \cdot \vec{uv}) + A_D^{-1} \cdot B \cdot \vec{p} \tag{31}$$

or

$$\widetilde{uv} = \widehat{uv} + A_D^{-1} \cdot (B \cdot \vec{p}) \quad (32)$$

Replacing $\vec{uv}$ in the continuity equation (25) by $\widetilde{uv}$ and re-arranging leads to

$$C \cdot A_D^{-1} \cdot B \cdot \vec{p} = C \cdot \vec{uv} + C \cdot A_D^{-1}(A_{ND} \cdot \vec{uv} - rh\vec{s}A) \quad (33)$$

from which the pressure can be computed directly.

3.3 Pseudo-velocity interpolation

Mainly for the sake of comparison, an equal-order scheme has been implemented, where velocities and pressure are stored on the vertices. To avoid the well-known checkerboard effect on the pressure field, the approach proposed by Prakash and Patankar (1985) has been adopted. This approach is quite similar to the earlier described SIMPLER-method. The only difference is that the continuity-matrix C is built using the pseudo-velocities $\widetilde{uv}$, which are computed at the vertices and assumed to vary linearily between them, resulting in a scheme where the pressure-contribution of the center-vertex of an element does not vanish.

4. Applications

Different testproblems for 2D and for axisymmetric fluid flow have been chosen for validating the implementation of the proposed method. The quadratic scheme is compared to a linear interpolation of the velocities regarding accuracy and computation time. The computations were done with a standard PentiumIII processor with 550 MHz.

4.1 2D testcases

4.1.1 Flow in complex geometries. The implementation of the equations for 2D buoyancy driven flow is validated by using a testcase proposed in Demirdžić *et al.* (1992). The geometry consists of a cylinder whose wall is maintained at a hot temperature enclosed by a square cavity, where the horizontal walls are assumed adiabatic, the vertical walls are kept at a low temperature. The cylinder center is slightly displaced from the cavity center. The fluid properties are chosen such that flows at $Ra = 10^6$ with $Pr = 10$ resp. $Pr = 0.1$ result. The temperature field, streamlines as predicted on the finest grid for a flow with $Pr = 10$ and parts of the used grids are shown in Figure 4. The calculated Nusselt number along the cold wall for flows at $Pr = 10$ and $Pr = 0.1$ is shown in Figure 5. All calculations were performed with the SIMPLE algorithm. The convergence to the benchmark values is obvious, thus validating the implementation.

This testcase shows that it is possible to compute flows even in complex geometries with the proposed method. It should be noted, that stable solutions were obtained even for flows with a Grashof number of 10^7 without any stabilization techniques. In order to compare the performance of the quadratic scheme, we have chosen a simpler geometry to avoid additional uncertainities due to the triangulation of the cylindrical part. This is the topic of the following section.

4.1.2 Flow in a square cavity. The performance regarding the necessary computation time to reach a certain error is studied in the following for a benchmark solution of laminar natural convection flows (Hortmann *et al.*, 1990). The geometry for the testcase consists of a square cavity with insulated top and bottom walls and the left wall at a high, the right wall at a low temperature. The temperature and flow fields for $\mathrm{Ra} = 10^5$ as predicted on the finest grid are shown in Figure 6. In order to avoid further influences, exclusivley homogeneous grids were used in the computations (with the SIMPLER algorithm) is shown in Figure 7. In order to specify the time needed to get a solution on a specific grid, the values of the benchmark properties were recorded during the iterations. The end time is reached, when no more changes in the relevant properties were found. All computations were started with a zero solution for the velocity and the temperature field.

Figure 8(a) shows the predicted percentage error of the local Nusselt number (estimated grid independent value is $\mathrm{Nu}_{\mathrm{loc}} = 7.72013$, its location is indicated in Figure 6) obtained on different grids compared to values tabulated in Hortmann *et al.* (1990). Both interpolation schemes result in a minor error for a certain number of velocity nodes compared to the benchmark values. Compared to the linear scheme, the quadratic interpolation

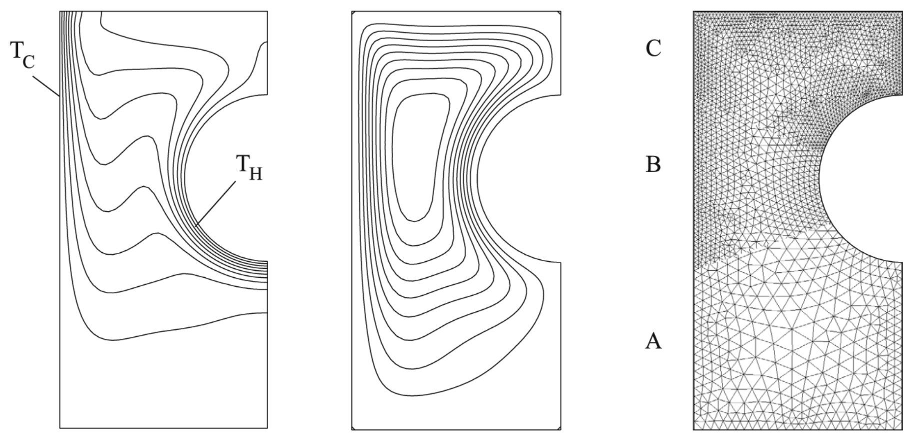


Figure 4.
Buoyancy driven flow in a complex geometry. Temperature isolines ($T_H - T_C = 1\,\mathrm{K}$, interval $0.1\,\mathrm{K}$) (left), streamlines (middle) and parts of the used grids (right)

results in more accurate results. Second-order monotonic convergence of the quadratic scheme is found, whereas a convergence one order of magnitude lower is found for the linear interpolation. The CPU time necessary to get a certain solution on a specific grid is plotted in Figure 8(b) for the various interpolation schemes. Although the coefficients resulting from the quadratic scheme are rather complex compared to the linear interpolation, the computation time to obtain a converged solution with a certain precision is smaller.

4.2 Axisymmetric testcases

4.2.1 Rotating disc flow. As an example for a driven flow we have chosen the flow in a cylinder with a height of 1 cm and a radius of 2.5 cm, whose bottom plate is rotating at a certain frequency. The density of the fluid is 1,000 kg/m^3, the viscosity $1 \cdot 10^{-3}$ kg/ms. For validation of the axisymmetric equations with the quadratic interpolation scheme, comparisons to results obtained with the commercial code FIDAP have been made. The convergence of the radial cuts of the azimuthal component to the grid independent values obtained with FIDAP for different grids is shown in Figure 9, validating the implementation. For these computations, the rotating frequency was chosen to be 0.063662 Hz.

Comparisons of the two interpolation schemes at the chosen rotation frequency did not show significant differences, thus the value was increased to 0.318 Hz, still relatively low in order to obtain a solution even on the coarsest grid. The value is resulting in a maximum azimuthal velocity of 5 cm/s. Figure 10 shows the streamfunction and azimuthal velocities for the grid with 12,500 degrees of freedom.

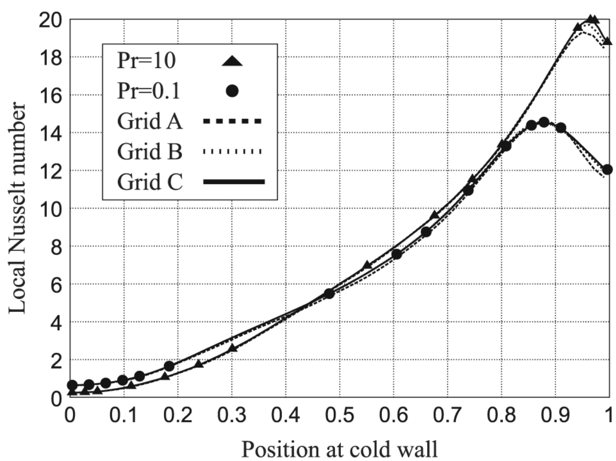


Figure 5.
Predicted local Nusselt number along the cold wall (from bottom to top) for Pr = 10 and Pr = 0.1 for different grids (A-2927, B-10111, C-15770 velocity nodes, cf. Figure 4) compared to benchmark values (symbols)

For this test case, a different approach has been chosen to compare the preformance of both schemes: Calculations (using SIMPLER) were started on the coarsest grid with zero initial velocities, computed until a given convergence criterion was reached, then interpolated to the next finer grid to be used as starting values, and so on.

Calculations were done on grids with about 700, 3,000, 12,500 and 50,000 degrees of freedom for the velocities for both methods. From the results shown in Figure 11 it is clear that the higher order interpolation not only gives better results as function of the number of degrees of freedom, but also as a function of the overall CPU-time. Nevertheless, the difference is smaller when temperature is also present, as shown in the next example.

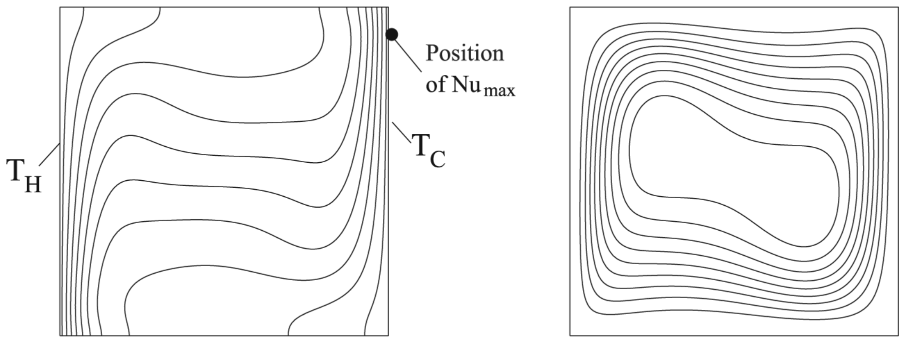


Figure 6.
Predicted isotherms ($T_H - T_C = 10\,K$, interval 1 K) and streamlines for a natural convective flow at Ra = 10^5 and Pr = 0.71

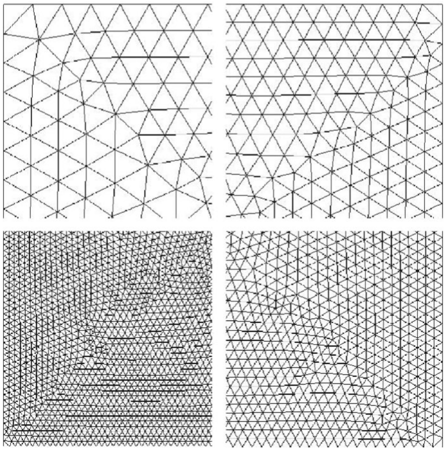


Figure 7.
Segments of the first grids as used in the computation (one quarter of each), corresponding to 263, 543, 2,255 and 5,995 temperature resp. 989, 2,081, 8,833 and 23,673 velocity nodes in case of the quadratic interpolation

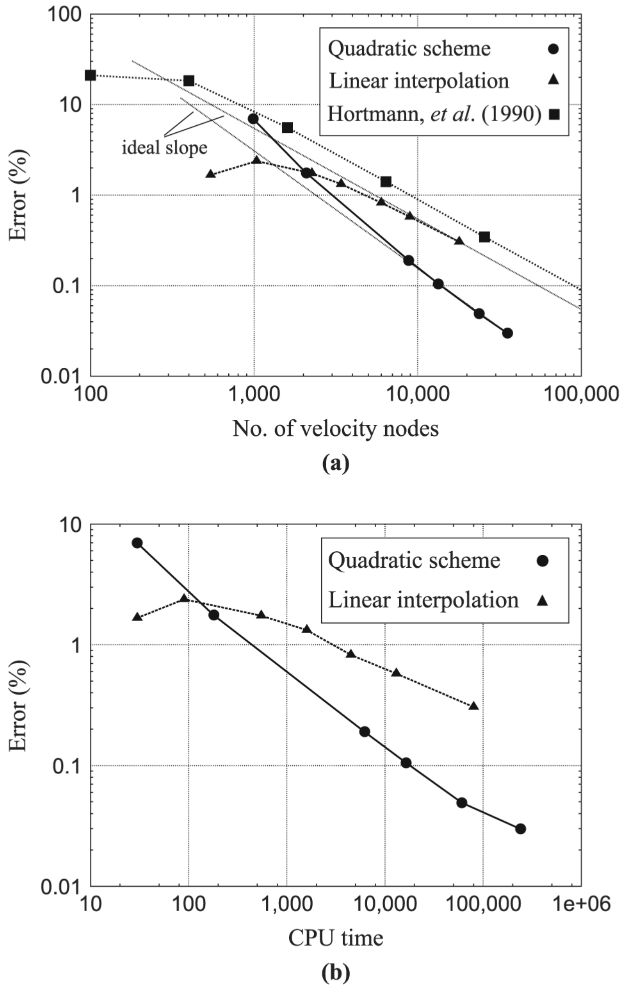


Figure 8.
(a) Predicted error of the local Nusselt number for the quadratic scheme (●), linear scheme (▲) and the values in Hortmann *et al.* (1990) (■) vs number of velocity nodes (a) and CPU time (b)

4.2.2 Axisymmetric buoyancy driven flow. As an example for axisymmetric buoyancy driven flow, a cavity with a radius and height of 1 cm has been chosen. Top and bottom are at a fixed temperature T_C at the side wall a parabolic temperature profile is applied, with $T_{\max} - T_C = 20\,\mathrm{K}$. The fluid is water with the properties $\rho = 1,000\,\mathrm{kg/m^3}$, $c_p = 4,181\,\mathrm{J/kg\,K}$, $\mu = 0.001\,\mathrm{kg/ms}$ and $\beta = 2 \cdot 10^{-4}\,\mathrm{K^{-1}}$.

The Rayleigh-Number, based on radius and maximum temperature difference, is $2.73 \cdot 10^5$, the Prandtl-Number is 6.97. As for the case of the rotating disc, calculations were carried out on grids containing about 700, 3,000, 12,500 and 50,000 degrees of freedom for the velocities, and on refining the grid interpolated solutions from the previous grid were used as starting approximations. The SIMPLE algorithm was employed this time in the calculations.

Figure 12 shows the temperature field and the streamfunction for this testcase Figure 13 shows the maximal radial velocity as a function of time with respect to the number of velocity nodes. Again, the better performance of the quadratic scheme is obvious, although not as much as in the previous case, where no temperature field had to be resolved.

5. Concluding remarks

A quadratic interpolation scheme for CVFEMs is presented. The proposed method was validated by comparison to different benchmark solutions, its

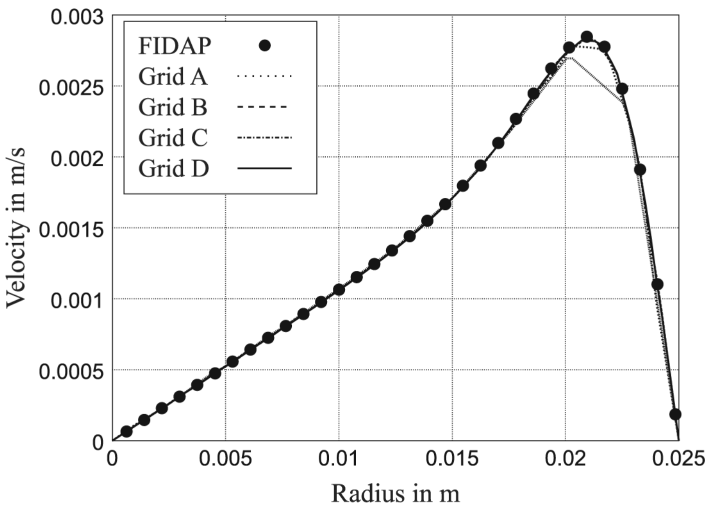


Figure 9.
Radial cuts of the azimuthal velocity at $z = H/2$ obtained on different grids with 394 (A), 1,228 (B), 3,004 (C) and 6,162 (D) velocity nodes, compared to FIDAP calculations (•). The rotating frequency is 0.063662 Hz

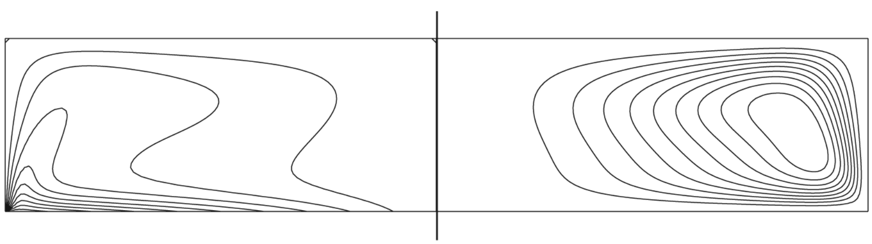


Figure 10.
Azimuthal velocities for a rotating disc flow in steps of 0.005 m/s (left) and isolines of the streamfunction (right). The maximal azimuthal velocity is 0.05 m/s, the rotating frequency is 0.318 Hz

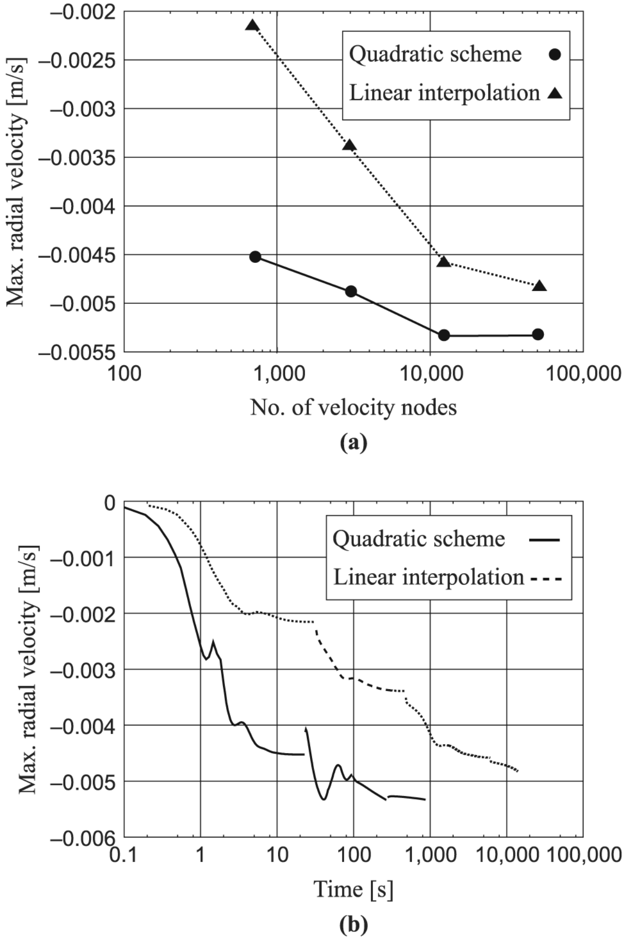


Figure 11.
Comparison of the interpolation schemes for a rotating disc flow: (a) max radial velocity vs no. of free velocity nodes, (b) max radial velocity vs. time for rotating disc flow

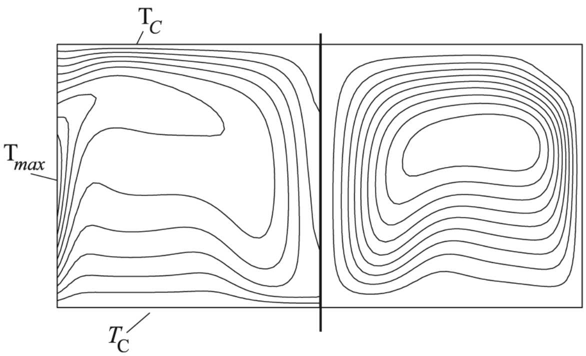
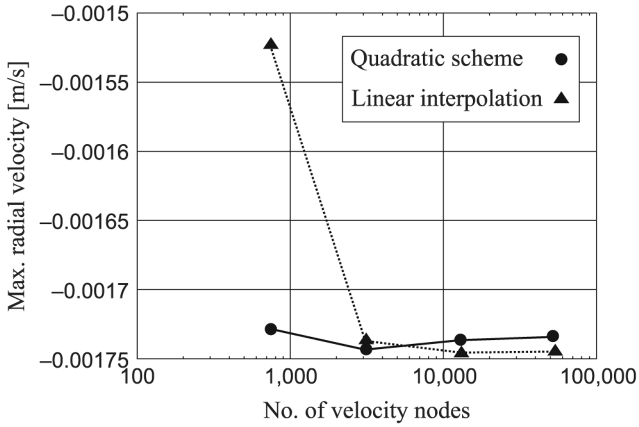


Figure 12.
Testcase for an axisymmetric buoyancy driven flow. The top and bottom walls are at a constant temperature T_C, at the side wall a parabolic temperature profile is applied with $T_{\max} - T_C = 20\,\mathrm{K}$. Plotted are temperature isolines in steps of 2 K (left) and isolines of the streamfunction (right)



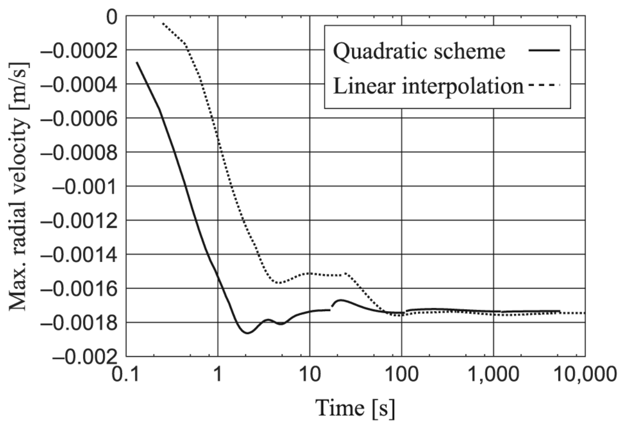


Figure 13.
Performance of the interpolation schemes: (a) max radial velocity vs no. of free velocity nodes, (b) max radial velocity vs time

performance with regard to accuracy and computation time was compared to a linear interpolation scheme implemented in the same code. Several outcomes can be stated:

(1) The benefits of the quadratic interpolation scheme compared to a linear interpolation are depending on the problem, whereby higher accuracy of the results from the quadratic interpolation was found for every testcase.

(2) Although the quadratic interpolation scheme leads to very large matrix coefficients (especially in case of axisymmetric equations), the necessary computation time to reach a certain solution error is smaller compared to the linear interpolation.

(3) Although not discussed in detail, it should be emphasized that a simple central difference scheme in the context of the proposed method leads to stable solutions even for problems with a Grashof number of 10^7.

References

Ashcraft, C., Pierce, D. and Wah, D.K. Spooles, Boeing Shared Services Group, Seattle, Washington.

Backofen, R., Kurz, M. and Müller, G. (2000), "Process modeling of the industrial VGF growth process using the software package CrysVUN++", *J. Crystal Growth*, Vol. 211, pp. 202-6.

Baehr, H.D. and Stephan, K. (1998), *Wärme- und Stoffübertragung*, 3rd ed., Springer, Berlin Heidelberg, New York.

Baliga, B.R. (1997), *Control-volume finite element methods for fluid flow and heat transfer*, Taylor and Francis, London, UK, Vol. 1, research note 3, pp. 97-135.

Baliga, B.R. and Patankar, S.V. (1983), "A control volume finite-element method for two-dimensional fluid flow and heat transfer", *Numer. Heat Transfer*, Vol. 6, pp. 245-61.

Costa, V.A.F., Oliveira, L.A. and Figueiredo, A.R. (1995), "A control-volume based finite element method for three-dimensional incompressible turbulent fluid flow, heat transfer and related phenomena", *Int. J. Numer. Meth. Fluids*, Vol. 21, pp. 591-613.

Demirdžić, I., Lilek, Ž. and Perić, M. (1992), "Fluid flow and heat transfer test problems for non-orthogonal grids: bench-mark solutions", *Int. J. Numer. Meth. Fluids*, Vol. 15, pp. 329-54.

Demmel, J.W., Gilbert, J.R. and Li, X.S. Superlu. *Computer Science Division*, University of California, Berkeley.

Derby, J.J. *et al.* (2001), "High performance computing, multi-scale models for crystal growth systems", *Lecture Notes in Computational Science and Engineering*, Springer-Verlag.

Despotis, G.K. and Tsangaris, S. (1995), "Fractional step method for solution of incompressible navier-stokes equations on unstructured triangular meshes", *Int. J. Numer. Meth. Fluids*, Vol. 20, pp. 1273-88.

Eichenseher, I. (1996), "Objektorientierte Triangulierung von zweidimensionalen gebieten", *Technical report, Report of the Institute for Applied Mathematics*, TU-München, Germany 3 June 1996.

Ferzinger, J.H. and Perić, M. (1999), *Computational Methods for Fluid Dynamics*, Springer, Vol. 2.

Fluent inc. USA. http://www.fluent.com.

Hainke, M. *et al.* (2001), "Equipment and process modelling of industrial crystal growth using the finite volume codes CrysVUn and STHAMAS", *Mathematics in Industry* Springer, (in preparation).

Hassan, Y.A., Rice, J.G. and Kim, J.H. (1983), "A stable mass-flow-weighted two-dimensional skewed upwind scheme", *Numer. Heat Transfer*, Vol. 6, pp. 395-408.

Hookey, N.A. and Baliga, B.R. (1988), "Evaluation and enhancements of some control volume finite-element methods – part 2. incompressible fluid flow problems", *Numer. Heat Transfer*, Vol. 14, pp. 273-93.

Hortmann, M., Perić, M. and Scheuerer, G. (1990), "Finite volume multigrid prediction of laminar natural convection: bench-mark solutions", *Int. J. Numer. Meth. Fluids*, Vol. 11, pp. 189-207.

Kurz, M.R.H. (1998), "Development of CrysVUN++, a software system for numerical modelling and control of industrial crystal growth processes" Ph.D. thesis, Technical Faculty of the University Erlangen-Nürnberg.

Kurz, M., Pusztai, A. and Müller, G. (1999), "Development of a new powerful computer code CrysVUN++ especially designed for fast simulation of bulk crystal growth processes", *J. Crystal Growth*, Vol. 198/199, pp. 101-6.

Maple, V. Waterloo Maple Inc., Waterloo, Ontario.

Masson, C. and Baliga, B.R. (1994), "A control-volume finite element method for dilute gas-solid particle flows", *Comp. Fluids*, Vol. 23 No. 8, pp. 1073-96.

Masson, C., Saabas, H.J. and Baliga, B.R. (1994), "Co-located equal-order control-volume finite element method for two-dimensional axisymmetric incompressible fluid flow", *Int. J. Numer. Meth. Fluids*, Vol. 18, pp. 1-26.

Metzger, M. (2000), "Optimal control of industrial crystal growth processes". Ph.D. thesis, Technical Faculty of the University Erlangen-Nürnberg.

Metzger, M. and Backofen, R. (2000), "Optimal temperature profiles for annealing of GaAs-Crystals", *J. Crystal Growth*, Vol. 220, pp. 6-15.

Muir, B.LeD. and Baliga, B.R. (1986), "Solution of three-dimensional convection-diffusion problems using tetrahedral elements and flow-oriented upwind interpolation functions", *Numer. Heat Transfer*, Vol. 9, pp. 143-62.

Müller, G. (1998), "Melt growth of semiconductors", *Mater. Sci. Forum*, Vol. 276-277, pp. 87-108.

Müller, G. and Fischer, B. (2001), "Optimization of melt growth processes by experimental analysis and computer modeling", *Adv. Crystal Growth*, pp. 167-90.

Patankar, S.V. (1980), *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation.

Patankar, S.V. and Spalding, D.B. (1972), "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", *Int. J. Heat Mass Transfer*, Vol. 15, pp. 1787-806.

Prakash, C. and Patankar, S.V. (1985), "A control volume-based finite-element method for solving the navier-stokes equations using equal-order velocity-pressure interpolation", *Numer. Heat Transfer*, Vol. 8, pp. 259-80.

Raithby, G.D. (1976), "Skew upstream differencing schemes for problems involving fluid flow", *Comp. Meths. Appl. Mech. Eng.*, Vol. 9, pp. 153-64.

Rice, J.G. and Schnipke, R.J. (1986), "An equal-order velocity-pressure formulation that does not exhibit spurious pressure modes", *Comp. Meths. Appl. Mech. Eng.*, Vol. 58, pp. 135-49.

Rida, S., McKenty, F., Meng, F.L. and Reggio, M. (1997), "A staggered control volume scheme for unstructured triangular grids", *Int. J. Numer. Meth. Fluids*, Vol. 25, pp. 697-717.

Saabas, H.J. and Baliga, B.R. (1994), "Co-located equal-order control-volume finite-element method for multidimensional, incompressible, fluid flow – part I: formulation", *Numer. Heat Transfer, Part B*, Vol. 26, pp. 381-407.

Schneider, G.E. and Raw, M.J. (1986), "A skewed, positive influence coefficient upwinding procedure for control-volume-based finite-element convection-diffusion computation", *Numer. Heat Transfer*, Vol. 9, pp. 1-26.

Schneider, G.E. and Raw, M.J. (1987a), "Control volume finite-element method for heat transfer and fluid flow using colocated variables – 1. Computational procedure", *Numer. Heat Transfer*, Vol. 11, pp. 363-90.

Schneider, G.E. and Raw, M.J. (1987b), "Control volume finite-element method for heat transfer and fluid flow using colocated variables — 2. Application and validation", *Numer. Heat Transfer*, Vol. 11, pp. 391-400.

Swaminathan, C.R. and Voller, V.R. (1992), "Streamline upwind scheme for control-volume finite elements, part I. formulations", *Numer. Heat Transfer, Part B*, Vol. 22, pp. 95-107.

Yeckel, A., Pandy, A. and Derby, J.J. (2001), "Representing realistic complexity in numerical models of crystal growth: coupling of global furnace modelling to three-dimensional flows", *Advances in Computational Heat Transfer*.