The Application of a Black-Box Solver with Error Estimate to Different Systems of PDEs

Torsten Adolph and Willi Schönauer

Institut für Wissenschaftliches Rechnen, Forschungszentrum Karlsruhe
torsten.adolph@iwr.fzk.de
willi.schoenauer@iwr.fzk.de

Happy are those people that do not see the errors.

Summary. At first a brief overview of the Finite Difference Element Method (FDEM) is given, above all how an explicit estimate of the error is obtained. Then for some academic examples the estimated and exact error are compared showing the quality of the estimate. The PDEs for fuel cells of PEMFC and SOFC type with extremely nonlinear coefficients are solved and the error estimate shows the quality of the solution. Finally for a complicated fluid/structure interaction problem of a high pressure Diesel injection pump, where the domain of solution has 3 subdomains with different PDEs and where a nested iteration procedure is needed, the PDEs are solved and the global error estimate shows the quality of the solution. For all these examples it would be very difficult to obtain a quality control of the solution by conventional grid refinement tests.

1 Introduction

The development of the Finite Difference Element Method (FDEM) at the computer center of the University of Karlsruhe has been supported by the German Ministry of Research (BMBF). The application of FDEM to the numerical simulation of fuel cells (FCs) has been supported by the Research Alliance Fuel Cells of the state Baden-Württemberg. In this paper we present a compilation of results of these projects.

Never before such problems have been solved with error estimates. So the emphasis of this paper will be on the error estimate: together with the solution we present values or plots for the error estimates. Because of the limited accorded space of the paper we cannot present all the details of FDEM and of the examples. However, we will give the precise information where these details are in the corresponding reports. As these reports are in the Internet, the reader can immediately have a look at them at his computer.

2 The Finite Difference Element Method (FDEM)

FDEM is an unprecedented generalization of the FDM on an unstructured FEM mesh. It is a black-box solver for arbitrary nonlinear systems of 2-D and 3-D elliptic or parabolic PDEs. If the unknown solution is $u(t, x, y, z)$ the operator for PDEs and BCs (boundary conditions) is (2.4.1) and (2.4.2) in [1]:

$$Pu \equiv P(t, x, y, z, u, u_t, u_x, u_y, u_z, u_{xx}, u_{yy}, u_{zz}, u_{xy}, u_{xz}, u_{yz}) = 0 \; . \quad (1)$$

For a system of m PDEs u and Pu have m components:

$$u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \; , \quad Pu = \begin{pmatrix} P_1 u \\ \vdots \\ P_m u \end{pmatrix} \; . \quad (2)$$

Because we have a black-box solver, the PDEs and BCs and their Jacobian matrices of type (2.4.6) in [1] must be entered as Fortran code in prescribed frames.

The geometry of the domain of solution is entered as a FEM mesh with triangles in 2-D and tetrahedra in 3-D. The domain may be composed of subdomains with different PDEs and non-matching grid. From the element list and its inverted list we determine for each node more than the necessary number of nodes for difference formulas of a given consistency order q. By a sophisticated algorithm from this set the necessary number of nodes is selected, see Section 2.2 in [1]. From the difference of formulas of different consistency order we get an estimate of the discretization error. If we want e.g. the discretization error for u_x and $u_{x,d,q}$ denotes the difference formula of consistency order q, the error estimate d_x is defined by

$$d_x := u_{x,d,q+2} - u_{x,d,q} \; , \quad (3)$$

i.e. by the difference to the order $q + 2$. This has a built-in self-control: if this is not a "better" formula the error estimate shows large error.

With such an error estimate we can explicitly compute the error of the solution by the error equation (2.4.8) in [1]. The knowledge of the error estimate allows a mesh refinement and order control in space and time (for parabolic PDEs), see Section 2.5 in [1].

A special problem for a black-box solver is the efficient parallelization because the user enters his domain by the FEM mesh. We use a 1-D domain decomposition with overlap to distribute the data to the processors, see Section 2.8 in [1]. We use MPI. A detailed report on the parallelization is [2]. The resulting large and sparse linear system is solved by the LINSOL program package [3] that is also efficiently parallelized for iterative methods of CG type and (I)LU preconditioning.

3 Academic Examples

The purpose of these academic examples is to check the quality of the error estimate (which is one level higher than the usual check for the quality of the solution), see Section 2.10 in [1]. We define the global relative error for a component l of the solution and the global relative error by

$$\frac{\|\Delta u_{d,l}\|}{\|u_{d,l}\|}\ , \quad \frac{\|\Delta u_d\|}{\|u_d\|} = \max_l \frac{\|\Delta u_{d,l}\|}{\|u_{d,l}\|}\ , \tag{4}$$

where $\Delta u_{d,l}$ is computed from the error equation (component l of (2.4.8) in [1]). The norm $\|\cdot\|$ is the maximum norm. We generate from the original PDE $Pu = 0$ a "test PDE" for a given solution $\bar{u}$ by $Pu - P\bar{u} = 0$ that has $\bar{u}$ as solution. This prescription holds also for the BCs. The exact global relative error then is

$$\frac{\|\bar{u} - u_d\|}{\|u_d\|}\ . \tag{5}$$

We compute on the HP XC6000 with Itanium2 processors of 1.5 GHz (University of Karlsruhe). As exact solution $\bar{u}$ we select either a polynomial of a given order or a sugar loaf type function (2.10.16) in [1]. We solve the Navier-Stokes equations in velocity/vorticity form (2.10.13) in [1] with the unknown functions velocity components u, v and vorticity ω, and Reynolds number $Re = 1$. We solve on a circle with radius = 1 on a grid with 751 nodes, 1410 elements that has been generated by the commercial mesh generator I-DEAS. We compute with 8 processors. The given CPU time is that of the master processor 1.

Table 1. Results for the solution of the Navier-Stokes type equations on a circle with 751 nodes for different consistency orders q and test function $\bar{u}$

	order $q = 2$		order $q = 4$		order $q = 6$	
type $\bar{u}$	error exact error estim.	CPU sec.	error exact error estim.	CPU sec.	error exact error estim.	CPU sec.
pol. order 6	0.154 0.155	0.158	0.914E-02 0.367E-01	0.175	0.108E-10 0.109E-08	2.131
sugar loaf	0.694E-01 0.642E-01	0.168	0.238E-01 0.220E-01	0.184	0.457E-02 0.736*	1.853

* here the order 8 for the error estimate is overdrawn (too coarse grid)

Table 1 shows the results. Here are two remarks: for $\bar{u}$ polynomial of order 6 and consistency order $q = 6$ we should reproduce $\bar{u}$ exactly which is expressed by the small errors. For the sugar loaf function and consistency order $q = 6$ we get a large error estimate. This shows the built-in self-control: near the top of the sugar loaf the grid is too coarse for the consistency order $q + 2 = 8$

that is used for the error estimate, the order 8 is "overdrawn" (higher order may not be better).

Table 2. Results for the self-adaptation of mesh and order for sugar loaf test function for prescribed global relative error 0.25×10^{-2} (0.25%)

cycle	no. of nodes	no. of elem.	no. of nodes ref.	no. of nodes with order 2	4	6	global relat. error exact	estimated	sec. for cycle
1	751	1410	132	427	320	4	0.305E-01	0.280E-01	1.021
2	1332	2493	345	180	1144	8	0.109E-01	0.950E-02	3.604
3	2941	5469	—	360	2556	25	0.179E-02	0.174E-02	10.086

For the demonstration of the self-adaptation we solve the same problem with the sugar loaf function again with 8 processors, but now we switch on the mesh refinement and order control for a global relative error of 0.25%. The results are shown in Table 2. The requested accuracy needs 3 refinement cycles. "no. of nodes ref." is the number of refinement nodes that determine the refinement elements from which then follows the new number of nodes. Observe the excellent error estimate that results from the optimal local order. Figure 1 shows the grid after the 3^{rd} cycle, the refinement is clearly visible.

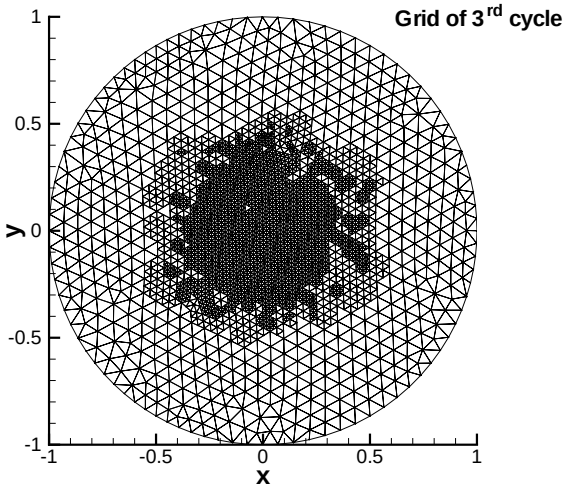

Fig. 1. Refined grid after 3^{rd} cycle of Table 2

As mentioned above we wanted to demonstrate the quality of the error estimate. The estimate is the better the smaller the error is, this is a natural

consequence of (3). What we have seen in Table 1 is also part of our test technique for each new problem: we at first create from the new problem a test PDE $Pu - P\bar{u} = 0$ and check with polynomial test solutions $\bar{u}$ the error estimate like in Table 1.

4 The Numerical Simulation of Fuel Cells (FCs)

This project was supported by the Research Alliance Fuel Cells (FABZ) of the state of Baden-Württemberg. The corresponding report [4] was written by ZSW Ulm for the equations of the PEMFCs (Part I), by the IWE of the University of Karlsruhe for the equations of the SOFCs (Part II) and by the RZ of the University of Karlsruhe for the numerical solution of the corresponding PDEs (Part III). These problems are characterized by the extreme nonlinearity of their coefficients that depend in a complicated way on the variables themselves. Quite naturally we will report here only on that part of [4] that deals with the numerical solution of the PDEs. Up to now nobody had solved these PDEs with an error estimate.

4.1 Numerical Solution of the PDEs for PEMFCs

The PEMFCs (Polymer–Electrolyte–Membrane FC or Proton Exchange Membrane FC) are the "cold" FCs, operating at about 300 K. The domain of solution for the used model is the GDL (gas diffusion layer) that is at its lower left half open to the oxygen channel and at its lower right half closed by a rib, see Fig. 2. How this is cut out of a whole cell can be seen at Fig. 1 on page III.4 in [4].

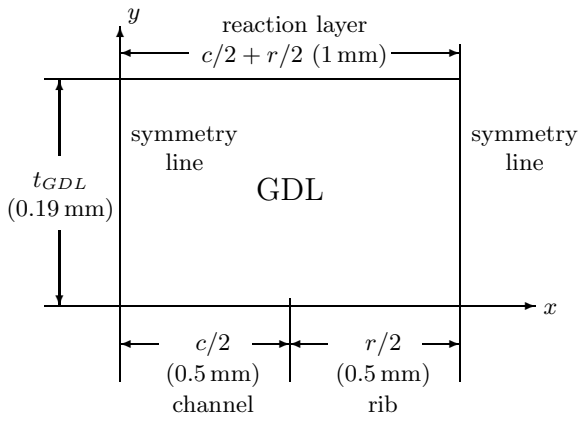

Fig. 2. Domain of solution for the PEMFC

The variables for this problem are the molar flux densities of oxygen in x-direction $\dot{n}_o^x$ and y-direction $\dot{n}_o^y$, similarly for water vapor $\dot{n}_w^x$ and $\dot{n}_w^y$, and

for nitrogen $\dot{n}_n^x$ and $\dot{n}_n^y$, the partial pressure for oxygen p_o, for water p_w and for nitrogen p_n, the total pressure p and as a special variable that has physical meaning only at the reaction layer the current density i. So we have 11 variables and need a system of 11 PDEs. These PDEs are given in Part I of [4], and in Table 1 on page III.6 is shown which equation is used for which variable position. We show below as an example a typical species transport equation, but without explanation of the notations:

$$\begin{aligned} &\frac{\dot{n}_o^x}{DKn_o} + \frac{p_w\dot{n}_o^x - p_o\dot{n}_w^x}{D_{ow}p} + \frac{p_n\dot{n}_o^x - p_o\dot{n}_n^x}{D_{on}p} + \frac{1}{RT}\frac{\partial p_o}{\partial x} - \frac{p_o}{RTp}\frac{\partial p}{\partial x} \\ &+ \left[\frac{B_o}{DKn_o} + \frac{B_o p_w}{D_{ow}p}\left(1 - \frac{B_w}{B_o}\right) + \frac{B_o p_n}{D_{on}p}\left(1 - \frac{B_n}{B_o}\right)\right]\frac{p_o}{RTp}\frac{\partial p}{\partial x} = 0\,. \end{aligned} \tag{6}$$

The effective permeabilities B_j depend in a complicated nonlinear way from the pressure p and the partial pressures p_i. We do not discuss here the BCs, they are given on pages III.9–III.11 in [4].

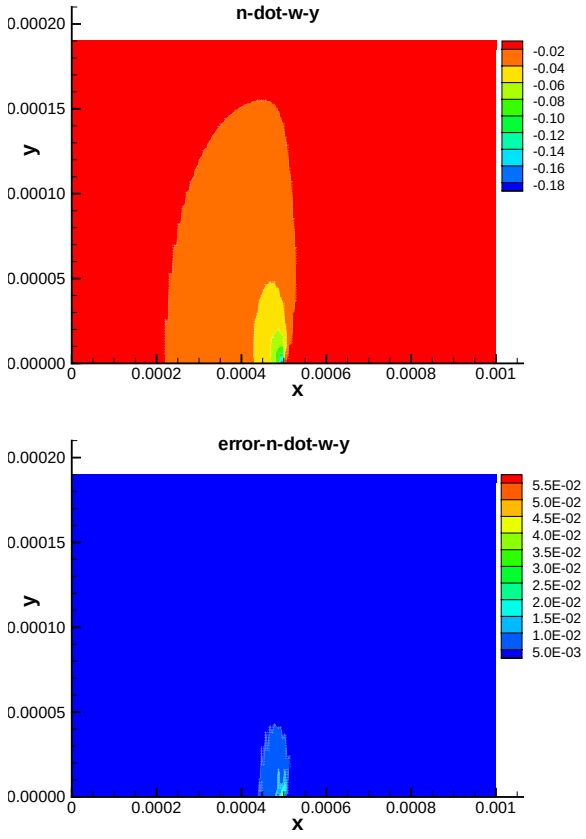

Fig. 3. Contour plot of molar flux density of water vapor in y-direction $\dot{n}_w^y$ and its error

We computed with 32 processors of the HP XC6000 with 1.5 GHz Itanium2 processors on a grid of 200×201 nodes in x,y-direction, we used consistency order $q = 4$. Because we have 11 variables per node we have 442200 unknowns. The computation time was 4123 sec on the master processor 1.

Figure 3 shows a typical result for $\dot{n}_w^y$ and its error. There is a quasi-singularity at the lower boundary where the BCs change from channel to rib. Figure 4 shows the current density i and its error along the reaction layer. This is the most interesting quantity for the engineers.

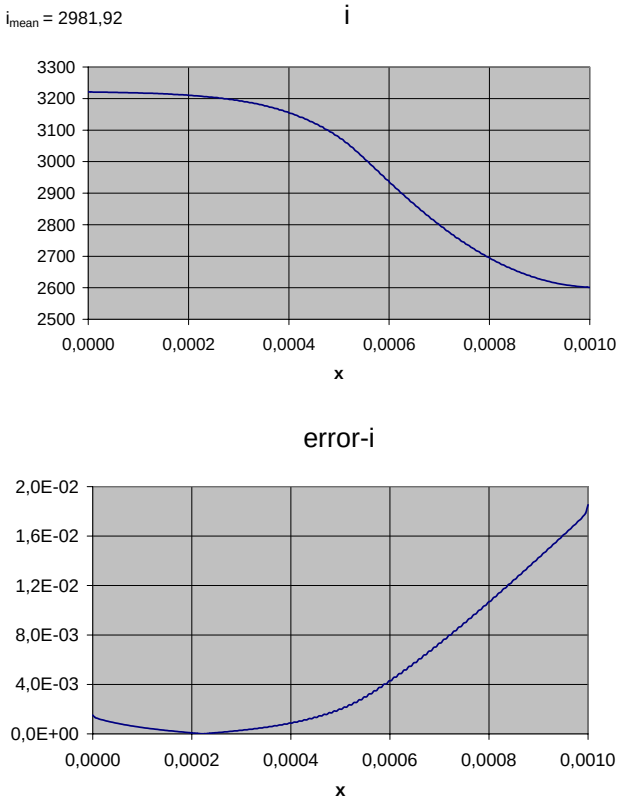

Fig. 4. Current density i along the reaction layer and its error

In [4] on pages III.15–III.24 are figures of type Fig. 3 for all variables. So the engineer can see if he can trust the solution. If there would be no error estimate he had to do grid refinement tests and observe for all variables and nodes (here 442200 values) how they change with finer grid. So the error estimate that consumes only a fraction of the total computation time is an invaluable feature in the solution process.

4.2 Numerical Solution of the PDEs for SOFCs

The SOFCs (Solid Oxide FCs) are the "hot" FCs, operating at about 1200 K. The domain of solution for the used model is the anode with flow in porous media and the gas channel with Navier-Stokes equations, see Fig. 5.

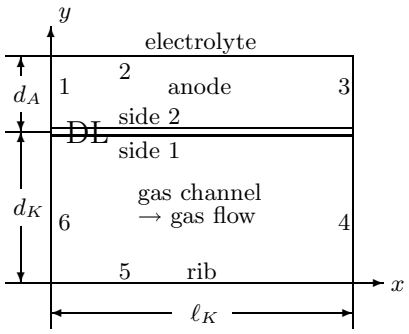

Fig. 5. Domain of solution for the SOFC and numbering of the boundaries

Here we have a solution domain that consists of two subdomains with different PDEs and a dividing line in-between for which we must prescribe coupling conditions (CCs) which are interior BCs. Because the model includes methane reforming, the variables are the flow velocities u_x, u_y in x- and y-direction, the mole fractions $Y_{CH_4} = Y_3$ for methane, $Y_{CO} = Y_4$ for carbon monoxide, $Y_{H_2} = Y_5$ for hydrogen, $Y_{CO_2} = Y_6$ for carbon dioxide, $Y_{H_2O} = Y_7$ for steam, and pressure p. So we have 8 variables and need 8 PDEs for the anode domain and 8 PDEs for the gas channel domain. These PDEs are given in Part II of [4]. In Table 12 on page III.31 the sequence of the variables and equations in the channel is given and in Table 15 on page III.36 in the anode.

In the channel we have Navier-Stokes-type equations, e.g. the y-momentum, (18) on page III.32 in [4]:

$$\begin{aligned} &\varrho u_x \tfrac{\partial u_y}{\partial x} + \varrho u_y \tfrac{\partial u_y}{\partial y} + \tfrac{\partial p}{\partial y} - \tfrac{\partial \mu}{\partial y}\left(\tfrac{4}{3}\tfrac{\partial u_y}{\partial y} - \tfrac{2}{3}\tfrac{\partial u_x}{\partial x}\right) \\ &- \mu\left(\tfrac{4}{3}\tfrac{\partial^2 u_y}{\partial y^2} + \tfrac{1}{3}\tfrac{\partial^2 u_x}{\partial x \partial y} + \tfrac{\partial^2 u_y}{\partial x^2}\right) - \tfrac{\partial \mu}{\partial x}\left(\tfrac{\partial u_x}{\partial y} + \tfrac{\partial u_y}{\partial x}\right) = 0 \,, \end{aligned} \tag{7}$$

and we have transport equations like that for methane, (19) on page III.32 in [4]:

$$\begin{aligned} &-\tfrac{\partial p}{\partial x} u_x Y_3 - \tfrac{\partial u_x}{\partial x} p Y_3 - \tfrac{\partial Y_3}{\partial x} p u_x - \tfrac{\partial p}{\partial y} u_y Y_3 - \tfrac{\partial u_y}{\partial y} p Y_3 - \tfrac{\partial Y_3}{\partial y} p u_y \\ &+ \tfrac{\partial D_{3,gas}}{\partial y}\left(\tfrac{\partial p}{\partial y} Y_3 + \tfrac{\partial Y_3}{\partial y} p\right) + D_{3,gas}\left(\tfrac{\partial^2 p}{\partial y^2} Y_3 + 2\tfrac{\partial p}{\partial y}\tfrac{\partial Y_3}{\partial y} + \tfrac{\partial^2 Y_3}{\partial y^2} p\right) = 0 \,. \end{aligned} \tag{8}$$

In the anode the Navier-Stokes equations are replaced by Darcy's law. The species transport equations have additional terms by the y-dependence of p

and by the chemical reactions. The transport equation for methane, (34) on page III.37 is now

$$\begin{aligned} &-\frac{\partial p}{\partial x}u_x Y_3 - \frac{\partial u_x}{\partial x}pY_3 - \frac{\partial Y_3}{\partial x}pu_x - \frac{\partial p}{\partial y}u_y Y_3 - \frac{\partial u_y}{\partial y}pY_3 \\ &\quad - \frac{\partial Y_3}{\partial y}pu_y + D_{3,gas}\left(\frac{\partial^2 p}{\partial y^2}Y_3 + 2\frac{\partial p}{\partial y}\frac{\partial Y_3}{\partial y} + p\frac{\partial^2 Y_3}{\partial y^2}\right) \\ &+ \frac{\partial D_{3,gas}}{\partial y}\left(\frac{\partial p}{\partial y}Y_3 + \frac{\partial Y_3}{\partial y}p\right) + D_{3,gas}\left(\frac{\partial^2 p}{\partial x^2}Y_3 + 2\frac{\partial p}{\partial x}\frac{\partial Y_3}{\partial x} + p\frac{\partial^2 Y_3}{\partial x^2}\right) \\ &\quad + \frac{\partial D_{3,gas}}{\partial x}\left(\frac{\partial p}{\partial x}Y_3 + \frac{\partial Y_3}{\partial x}p\right) - \frac{RT}{d_A}r_3 = 0\,. \end{aligned} \tag{9}$$

These equations are extremely nonlinear because μ in (7) and the $D_{i,gas}$ depend nonlinearly on the Y_j, similarly the reaction rates r_k are depending.

Here we must define the BCs for the 6 boundaries of Fig. 5 and the CCs for side 1 and side 2 of the dividing line DL. These conditions are given in Tables 17–19 on pages III.40 and III.41 in [4].

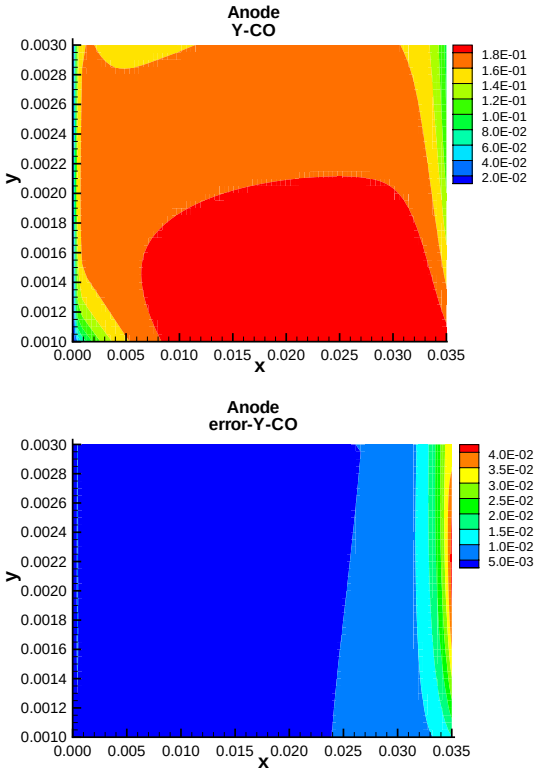

Fig. 6. Contour plot of mole fraction Y_{CO} and its error for the anode

The numerical solution of these equations was much more critical than that of the PEMFCs because of the extreme nonlinearity of the coefficients.

We computed with a grid with 80 nodes in x-direction and 41 nodes in the channel and 41 nodes in the anode in y-direction, resulting in 52480 unknowns. We used consistency order $q = 4$. The computation time on 8 processors of the HP XC6000 with processors Itanium2, 1.5 GHz, was 510 sec on the master processor 1. The colour plots of the results and error estimates for all 8 variables in the channel and anode are presented in [4], pages III.47–III.62. We present here only the mole fraction Y_{CO} and its error in Fig. 6 for the anode.

The ampleness of the generated information can be estimated only if one looks at all the nice colour plots of the report [4]. Here the engineer can immediately see the quality of the solution from the error plots, this is an unprecedented gain in information.

5 Fluid/Structure Interaction for a High Pressure Diesel Injection Pump

A detailed presentation of this problem (cooperation with Bosch) is given in Section 3.3 in [1]. In a high pressure Diesel injection pump the housing extends under the injection pressure of 2000 bar and the piston is compressed so that the lubrication gap between housing and piston changes its form and consequently the leakage flow changes. This is a fluid structure interaction problem. The problem is simplified by replacing the complicated shape of the housing by a tube or bush, see Fig. 7. The piston does not move, so we have a static configuration.

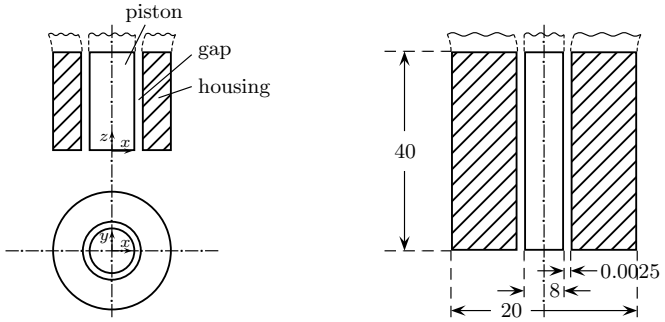

Fig. 7. Symbolic configuration and dimensions in mm. In reality the gap is extremely thin

Our domain of solution now has 3 subdomains with different PDEs: in the housing and piston we must solve the elasticity equations of steel, in-between we have the gap with the Navier-Stokes equations for Diesel. The coupling between these domains is the following: the fluid pressure p is the normal stress for housing and piston, so we have a direct interaction of the flow on

the structure. By this normal stress the housing expands and the piston is compressed, thus the form of the gap changes and this changes the flow which changes p and thus the normal stress etc. So the interaction of housing and piston on the flow is indirect and more complicated and requires an iterative procedure.

We solve the problem in axisymmetrical coordinates, then x in Fig. 7 becomes the radius r. For the elasticity equations in housing and piston the dependent variables are the displacements w and u in z- and r-direction, the stresses σ_z, σ_r, σ_φ and the shear stress τ_{rz} $(= \tau_{zr})$. Although we have rotational symmetry with $\partial/\partial\varphi = 0$, there is circumferential stress σ_φ. So we have 6 variables and need 6 PDEs. They are given in (3.3.4.1)–(3.3.4.6) on page 129 in [1]. Here we show the first and last of these equations in incremental form (index "old" is for the last solution):

$$\frac{1}{E}\left[\sigma_z - \sigma_{z,old} - \nu(\sigma_\varphi - \sigma_{\varphi,old}) - \nu(\sigma_r - \sigma_{r,old})\right] - \frac{\partial w}{\partial z} = 0\,, \tag{10}$$

$$\frac{\partial \tau_{rz}}{\partial r} + \frac{\partial \sigma_z}{\partial z} + \frac{\tau_{rz}}{r} = 0\,. \tag{11}$$

Here E is the elasticity module and ν is Poisson's ratio. Table (3.3.4.7) in [1] gives the information which equation is used in which position (for which variable) in the system of 6 equations.

In the lubrication gap we must solve the Navier-Stokes equations. The variables are the velocity components w and u in z- and r-direction, and the pressure p. So we need a system of 3 equations. Here we show the momentum equation in r-direction and the continuity equation, (3.3.4.17) and (3.3.4.19) on page 123 in [1]:

$$u\frac{\partial u}{\partial r} + w\frac{\partial u}{\partial z} + \frac{1}{\varrho}\frac{\partial p}{\partial r} - \frac{\eta}{\varrho}\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} - \frac{u}{r^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0\,, \tag{12}$$

$$\frac{\partial u}{\partial r} + \frac{u}{r} + \frac{\partial w}{\partial z} = 0\,. \tag{13}$$

Here η and ϱ are the dynamical viscosity and the density. But now we have a problem: the whole domain has 3 subdomains where two of them have 6 variables and PDEs, and one has 3 variables and PDEs. However, the code is designed for the same number of variables in the whole domain. So we add in the fluid domain 3 dummy variables with variable $= 0$ as PDEs and BCs.

Table (3.3.4.20) on page 133 in [1] shows which equation is used in which position of the system, i.e. for which preferred variable: for w we take the continuity equation, for u the r-momentum equation and for p the z-momentum equation. This is a quite natural ordering. If we want to discuss the BCs for the flow we immediately meet a serious problem: at the entry we have a prescribed pressure of 2000 bar $= 200\,\mathrm{N/mm^2}$ and at the exit of 0 bar. Because in the Navier-Stokes equations there is only $\partial p/\partial z$ we can prescribe for incompressible flow the pressure only at one position, e.g. 2000 bar at the entry. Then the

pressure at the exit is the result of the Navier-Stokes equations. At the high pressure entry we prescribe a parabolic velocity profile for w: $w(r) =$ parabola with w_{max} in the middle of the entry. The choice of w_{max} determines the pressure at the exit, i.e. the pressure drop in the gap. This introduces what we call $\boldsymbol{w_{max}}$**-iteration**: we start with an appropriate value of w_{max}, determine p_{exit} and if it is too large we increase w_{max} (gives larger pressure drop), if it is too small, we reduce w_{max}. This is made by a sophisticated iteration procedure.

The detailed discussion of the BCs for housing, piston and fluid is given in Section 3.3.4 in [1] and would be too lengthy for this paper. However, the essential part is the coupling between fluid and structure: if we have computed displacements for the pressure p given by the fluid flow, we apply in an incremental form these displacements by a shift algorithm that is presented in Section 3.3.3 in [1]. The index "old" in (10) denotes the values before the shifting (on the old grid). Then we observe if the grid still shifts. This induces what we call **grid iteration**. If the grid does no longer move we have the solution of our problem. Thus we have the nested iterations shown in Fig. 8:

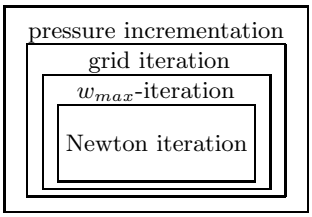

Fig. 8. The nested iterations for the solution process

The innermost iteration is the Newton iteration for the solution of the PDEs, then we must determine w_{max} for the exit pressure zero, then we must apply the displacements until the grid does no longer move. The outermost iteration gives the possibility to increase gradually the entry pressure.

When we made the first numerical experiments we had serious difficulties caused by the extreme differences in length scales: housing and piston in cm, lubrication gap in micrometer, see Fig. 7. After some trials we decided to use cm as length scale. The discretization errors of housing and piston for the used grid caused the surfaces to be "rough" in the micrometer scale. Therefore we applied a smoothing of these surfaces.

We used a grid of 401×80 in z,r-direction for the housing, 401×40 for the fluid and 401×81 for the piston. We computed with 16 processors of the HP XC6000 with 1.5 GHz Itanium2 processors. The CPU time on the master processor 1 was 3354 sec, of which 3296 sec were needed for the linear solver LINSOL with full LU preconditioning. Table 3 gives some results for 2000 bar entry pressure for housing and fluid. In Table 3.3.5.1 on page 138 of [1] further values for entry pressures of 1500 bar to 3000 bar are given. Figure 9 shows the

Table 3. Maximum value of solution component, max. relat. error and mean relat. error of solution component and volume flow through the gap for entry pressure of 2000 bar

Housing

		max. solution	max. error	mean error
w	cm	0.4143E−02	0.10E−03	0.11E−04
u	cm	0.7584E−03	0.32E−03	0.82E−04
σ_z	N/cm^2	0.2244E+05	0.30E−02	0.12E−04
σ_r	N/cm^2	0.2000E+05	0.28E−02	0.22E−04
σ_φ	N/cm^2	0.2772E+05	0.65E−03	0.67E−04
τ_{rz}	N/cm^2	0.9837E+03	0.24E−01	0.87E−04

Fluid

		max. solution	max. error	mean error	Volume [cm^3/s]
w	cm/s	0.3339E+04	0.87E+00	0.14E−01	2.40
u	cm/s	0.3810E+00	0.96E+02	0.10E+01	
p	N/cm^2	0.2000E+05	0.94E−01	0.60E−02	

form of the lubrication gap for 2000 bar entry pressure. The bold lines show the original channel. It is amazing how the high injection pressure changes the lubrication gap from the manufacturing dimension of 2.5 micrometer to up to 11.5 micrometer. The error estimates in Table 3 show the quality of the solution. For the fluid there is a maximal error of w of 87%, but the (arithmetic) mean error is 1.4%, so large errors occur only locally.

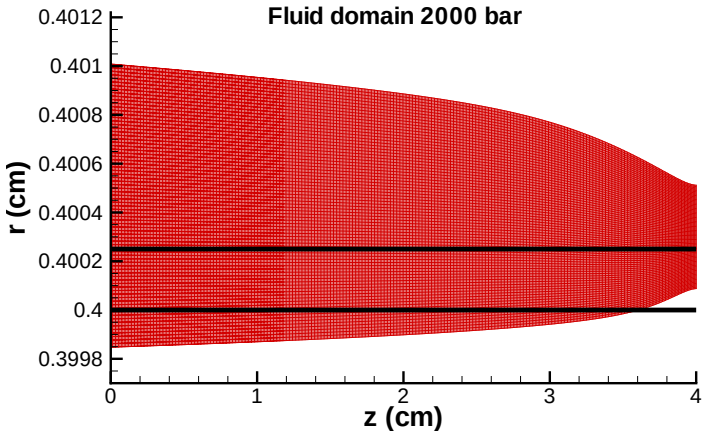

Fig. 9. Fluid domain with computational grid for 2000 bar entry pressure and original channel (*bold lines*)

Figure 10 shows the contour plot of the velocity w in z-direction which is responsible for the leakage flow that is in this case $2.40\,\mathrm{cm}^3/\mathrm{s}$. Figure 10 also shows its error plot. Here we can see that the large errors occur only locally. There a much finer grid should be used. As the mean error of w is 1.4% we can conclude that the volume flow is also accurate to this error level. Note that this is a global error estimate that includes all the errors of all the equations in the coupled domains. Here again the error estimate gives us the certainty that we can trust our solution. The reader may think how he could get this certainty by other methods for this complicated fluid/structure interaction problem.

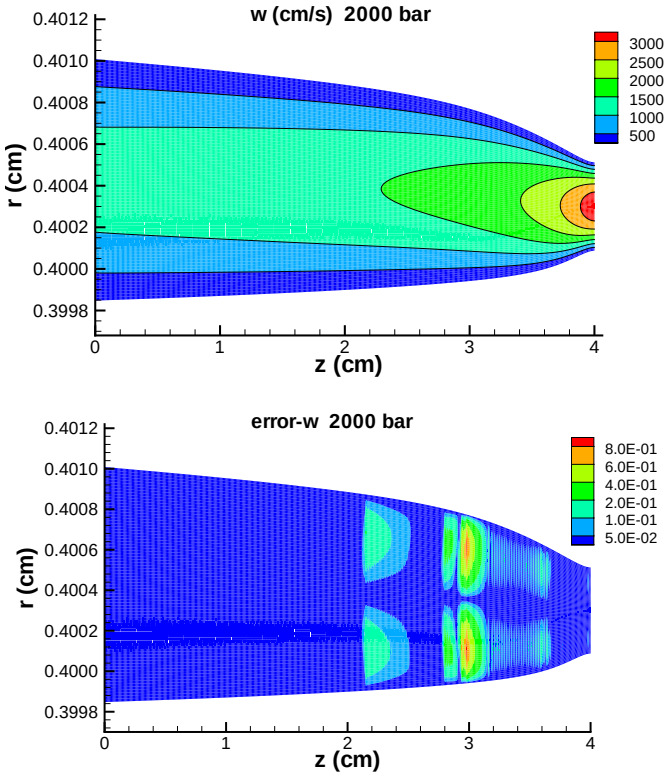

Fig. 10. Contour plot for the velocity w in z-direction for 2000 bar and its error

6 Concluding Remark

The purpose of this paper was to show that an error estimate is an invaluable advantage for a PDE solver. We have shown for some academic examples

where we constructed PDEs with known solution, that our error estimate gives an excellent approximation for the exact error. It is the better the smaller the error is. Then we solved the PDEs for the numerical simulation of fuel cells of PEMFC and SOFC type and the error estimates showed the quality of the solution for all components of the systems. Finally, we solved a fluid/structure interaction problem for a high pressure Diesel injection pump. The high pressure of 2000 bar bends up the housing that the lubrication gap widens from 2.5 micrometer up to 11.5 micrometer. The solution algorithm is a complicated nested iteration. Nevertheless we compute a global error estimate for the coupled domains of housing, piston and fluid that tells us that we can trust our solution. The conventional way would be to do a sequence of grid refinements. The reader may imagine what amount of work would be necessary to obtain a comparable information like our error estimate.

References

1. W. Schönauer, T. Adolph, FDEM: The Evolution and Application of the Finite Difference Element Method (FDEM) Program Package for the Solution of Partial Differential Equations, 2005, available at www.rz.uni-karlsruhe.de/rz/docs/FDEM/Literatur/fdem.pdf
2. T. Adolph, The Parallelization of the Mesh Refinement Algorithm in the Finite Difference Element Method, Doctoral Thesis, 2005, available at www.rz.uni-karlsruhe.de/rz/docs/FDEM/Literatur/par_mra_fdem.pdf
3. LINSOL, see www.rz.uni-karlsruhe.de/rd/linsol.php
4. The Numerical Simulation of Fuel Cells of the PEMFC and SOFC type with the Finite Difference Element Method (FDEM), collective paper by three groups, 2005, available at www.rz.uni-karlsruhe.de/rz/docs/FDEM/Literatur/fuelcells.pdf