A Novel SIMPLE-Based Pressure-Enthalpy Coupling Scheme for Engine Flow Problems

Maximilian Emans^{a,b}, Zoran Žunič^c, Branislav Basara^d and Sergey Frolov^e

^{a}*Johann Radon Institute for Computational and Applied Mathematics*
 Altenbergerstr. 69, 4040 Linz, Austria
^{b}*Industrial Mathematics Competence Center (IMCC)*
 Altenbergerstr. 69, 4040 Linz, Austria
^{c}*AVL-AST d.o.o.*
 Trg Leona Stuklja 5, 2000 Maribor, Slovenia
^{d}*AVL List GmbH*
 Hans-List-Platz 1, 8020 Graz, Austria
^{e}*Semenov Institute of Chemical Physics*
 4 Kogosin street, 119991 Moscow, Russia
 E-mail(*corresp.*): `maximilian.emans@ricam.oeaw.ac.at`

Abstract. A novel method in CFD derived from the SIMPLE algorithm is presented. Instead of solving the linear equations for each variable and the pressure-correction equation separately in a so-called segregated manner, it relies on the solution of a linear system that comprises the discretisation of enthalpy and pressure-correction equation which are linked through physical coupling terms. These coupling terms reflect a more accurate approximation of the density update with respect to thermodynamics (compared to standard SIMPLE method). We show that the novel method is a reasonable extension of existing CFD techniques for variable density flows based on SIMPLE. The novel method leads to a reduction of the number of iterations of SIMPLE which translates in many – but not in all – cases to a reduction in computing time. We will therefore demonstrate practical advantages and restrictions in terms of computational efficiency for industrial CFD applications in the field of piston engine simulations.

Keywords: finite volume method, Navier–Stokes equations, numerical method.

AMS Subject Classification: 35M13; 68N30; 76G25.

1 Introduction

The simulation of flow inside piston engine cylinders and inside the attached parts of the engine is a field where the application of CFD (computational fluid

dynamics) has reached a particular maturity. Simulations are used to assist the development process of combustion engines. For many different problems, particular models have been developed. All of them have in common that they rely on a solution of the Navier–Stokes equations to obtain a velocity field, a pressure field, and a temperature distribution. Many models are coupled in two ways to this kernel of the simulation tool. On the one hand, they use these data as input, e.g., if they require the velocity field to compute the distribution of particles for the prediction of soot formation; on the other hand, they manipulate the flow field, e.g., if a combustion model significantly contributes to the sources in the energy equation.

With respect to commercial codes in the industrial development process, an important class of methods providing an appropriate approximation of the solution of the Navier–Stokes equations for this purpose masters the non-linearity of the system as well as the coupling of the momentum equations and the continuity equation by iterative algorithms derived from SIMPLE ("semi-implicit method for pressure-linked equations"). Commercial software solutions employing these methods are available such that, ideally, engineers no longer have to bother about the flow solution but can focus on the attached models. For this, the flow solution must be obtained reliably and fast; moreover, it should restrict the choice of the parameters of the simulation such as time step and spatial resolution as little as possible. Although acceptable methods exist, there is still space for improvement, and in particular a faster computation of the flow solution will always be appreciated.

Many ways have been proposed to accelerate the convergence of the SIMPLE algorithm in the genuine formulation of Patankar and Spalding [22] that tends to be unsatisfactory in some situations. Among the most important ones are SIMPLEC and PISO, see van Doormal and Raithby [10], for an overview. Whereas genuine SIMPLE requires the solution of one linear system per unknown (e.g. velocity component or pressure) and iteration, the enhanced methods require the solution of more than one linear system per unknown and iteration. The size of the systems always corresponds to the number of discretisation points or finite volumes; despite occasional convergence problems these methods are applied with great success in many important engineering applications.

Coupling equations and calculating two or more variables at the same time is an alternative to obtain accelerated convergence compared to the established approaches that are referred to as segregated. Apparently, until very recently one tends to prefer segregated approaches in CFD simulation tools for engineering applications. For the simulation of essentially incompressible flows in an industrial context, an appropriate method that couples the momentum equations and the continuity equation has been described by Chen et al. [5] not earlier than 2010. The more rapid convergence of the coupled method make these techniques in particular attractive for complex simulations where, due to the cost of physical models attached to (and interfering with) the fluid flow solution, each single iteration is expensive. The coupling of variables shifts the effort to obtain a solution of the coupled system of the Navier–Stokes equations from the non-linear or outer iteration, e.g. the iterations of SIMPLE, to the

solution of larger and more complex linear systems. In particular with regard to large-scale computations it is not sufficient only to report on rapid convergence of these methods. It is also necessary to show that appropriate methods to solve these linear systems efficiently are available such as e.g. Chen et al. [5] do. Furthermore, for high-resolution time-dependent simulations of engine flow problems, these methods should be shown to work efficiently on parallel computers.

For compressible flows the pressure-correction equation (that represents the mass conservation in SIMPLE algorithm) depends not only on the velocity field, but also on the density field, see Demirdžić et al. [9]. The segregated methods improve the velocity update with respect to the flow field, but the error introduced by the density field, calculated based on the pressure of the previous iteration, is not reduced. Since the density depends strongly on the temperature, this error can be large if the actual guess of the temperature is not yet close to the solution. As a matter of fact, the right-hand side of the pressure-correction equation strongly depends on the enthalpy changes through the density field and, at the same time, the pressure change as a result of this pressure-correction equation contributes significantly to the right-hand side of the enthalpy equation. A stronger link between enthalpy and pressure allows to employ a better approximation to the density in the pressure-correction equation. This motivated us to couple these two equations within the context of the SIMPLE algorithm. Since the coupled linear system is larger and since it has less favourable properties, one has to be aware that the gain through faster convergence does not always translate into a reduced computing time.

In this article we sketch the main ideas of a pressure-enthalpy coupling procedure extending the SIMPLE algorithm where the focus lies on the properties of the new algorithm that are relevant to predict its efficiency. Then we will present results of three benchmarks, each representing an important class of problems in engine simulation; we will discuss convergence properties and performance in terms of wall clock times.

2 SIMPLE-Based Methods for CFD

After stating the governing equations at the beginning of this section, we will deduce SIMPLE algorithm as a factorisation preconditioner for incompressible problems. The discussion of the extension for compressible flows is supposed to highlight that the pressure-enthalpy coupling is a second important step in the same direction.

2.1 Governing equations

The Navier–Stokes equations consist of the momentum equations, the continuity equation, and an energy equation. These equations read (in non-conservative formulation)

$$\rho \left(\frac{\partial \vec{v}}{\partial t} + (\vec{v} \circ \vec{\nabla})\vec{v} \right) = -\vec{\nabla}p + \vec{\nabla}\tau + \rho \vec{f_e}, \tag{2.1}$$

$$\vec{\nabla} \circ (\rho \vec{v}) = -\frac{\partial \rho}{\partial t}, \tag{2.2}$$

$$\frac{\partial(\rho h)}{\partial t} + \vec{\nabla} \circ (\rho \vec{v} h) + \vec{\nabla} \circ (\lambda \vec{\nabla}(c_p h)) = \frac{\partial p}{\partial t} - \rho \vec{v} \frac{\partial \vec{v}}{\partial t} - \frac{\rho \vec{v}}{2} \circ \vec{\nabla} \vec{v}^2$$
$$+ \vec{\nabla} \circ (\tau \circ \vec{v}) + \rho \vec{f_e} \circ \vec{v} + q_h. \tag{2.3}$$

The unknown functions are the velocity field $\vec{v}$, the pressure p, the static enthalpy h and the density ρ. Further, the thermal conductivity is denoted by λ, the heat capacity by c_p, the external body force by $\vec{f_e}$, and the external heat source by q_h; τ is the sheer strain tensor reflecting the diffusion term.

The system is completed by a relation between density and two thermodynamic variables, e.g. pressure and temperature for ideal gas $\rho = p/RT$, where R is the gas constant and $T = T(h)$ the thermodynamic temperature.

For the numerical treatment, a discretisation in space and in time is required. With conventional finite-volume techniques the discretisation of these equations leads to such a system:

$$A\vec{u} + M\vec{p} = \vec{b},$$
$$C\vec{u} = \vec{c},$$
$$G\vec{h} = \vec{g}.$$

The discrete unknown velocity field is denoted by $\vec{u}$, the pressure field by $\vec{p}$, and the enthalpy field by $\vec{h}$; the discrete density field is $\vec{\varrho}$. $A = A(\vec{u}, \vec{\varrho})$ denotes the discretised operator that acts on the velocity field in the momentum equations, i.e. it expresses convective, diffusive, and inertia components stemming from the time derivative; the part of the discretised time derivative depending on the velocity of the previous time step(s) forms, together with the discretised body force term, the vector $\vec{b}$. Note that both, A and $\vec{b}$ therefore depend on the time even if the body force is constant. M is the discretisation of the pressure term, $C = C(\vec{\varrho})$ represents the discretised continuity equation, $\vec{c}$ the corresponding right-hand side; the energy equation is represented by $G = G(\vec{u}, \vec{\varrho})$ and $\vec{g}$.

2.2 SIMPLE for incompressible flows

The SIMPLE algorithm was first employed for incompressible flow problems. In this case the problem can be expressed as

$$\begin{pmatrix} A & M \\ C & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} \vec{b} \\ \vec{c} \end{pmatrix}.$$

SIMPLE is an iterative method; the iteration count is denoted by m. The operator $A(\vec{u}, \vec{\varrho})$ is linearised in such a way that for the evaluation of its entries one uses simply the most recent values. Through $A^{(m)} := A(\vec{u}^{(m)}, \vec{\varrho}^{(m)})$ we express the operator that has been evaluated with the values calculated in iteration m. With the splitting $\vec{p}^{(m+1)} = \vec{p}^{(m)} + \vec{p}'$ the previous equation is equivalently rewritten as

$$\begin{pmatrix} A^{(m)} & M \\ C & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p}' \end{pmatrix} = \begin{pmatrix} \vec{b} - M\vec{p}^{(m)} \\ \vec{c} \end{pmatrix}.$$

The SIMPLE method relies on a factorisation of this system (the superscript of A is dropped for the moment):

$$\begin{pmatrix} A & M \\ C & 0 \end{pmatrix} = \begin{pmatrix} A & 0 \\ C & S \end{pmatrix} \cdot \begin{pmatrix} I & A^{-1}M \\ 0 & I \end{pmatrix}.$$

This is exploited to describe a forward substitution

$$\begin{pmatrix} A & 0 \\ C & S \end{pmatrix} \begin{pmatrix} \vec{u}^* \\ \vec{p}^* \end{pmatrix} = \begin{pmatrix} \vec{b} - M\vec{p}^{(m)} \\ \vec{c} \end{pmatrix}$$

and a backward substitution

$$\begin{pmatrix} I & A^{-1}M \\ 0 & I \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p}' \end{pmatrix} = \begin{pmatrix} \vec{u}^* \\ \vec{p}^* \end{pmatrix},$$

where $\vec{u}^*$ and $\vec{p}^*$ denote auxiliary variables. From this the following equations can be extracted:

$$A^{(m)}\vec{u}^* = \vec{b} - M\vec{p}^{(m)}, \tag{2.4}$$

$$S\vec{p}' = \vec{c} - C\vec{u}^*, \tag{2.5}$$

$$\vec{u}^{(m+1)} = \vec{u}^* - A^{(m)-1}M\vec{p}'. \tag{2.6}$$

In SIMPLE one uses $\tilde{A}^{(m)} := diag(A^{(m)})$ instead of $A^{(m)}$ in Eq. (2.6) and $\tilde{S}^{(m)} := -C\tilde{A}^{(m)-1}M$ instead of S in Eq. (2.5) and one employs this scheme as a preconditioner to a Richardson iteration; the resulting scheme is Algorithm 1.

Algorithm 1 SIMPLE algorithm for incompressible flows

1: initialise, $m = 0$
2: **while** <not converged> **do**
3:   solve $A^{(m)}\vec{u}^* = \vec{b} - M\vec{p}^{(m)}$
4:   solve $\tilde{S}^{(m)}\vec{p}' = \vec{c} - C\vec{u}^*$
5:   $\vec{u}^{(m+1)} = \vec{u}^* - \tilde{A}^{(m)-1}M\vec{p}'$
6:   $\vec{p}^{(m+1)} = \vec{p}^{(m)} + \vec{p}'$
7: **end while**

The iterative procedure is necessary because of two aspects. First, the coupling of the variables needs to be handled iteratively since, initially, after step 3 of Algorithm 1, the momentum equations with the pressure of the previous iteration, but not necessarily the continuity equation, are satisfied; after step 5 the continuity equation, but not the momentum equations are satisfied. In the course of the iteration the solution is driven to satisfy both constraints. Second, the system is nonlinear, i.e. the operator A depends on $\vec{u}$ such that $\vec{u}^{(m+1)} \neq \vec{u}^{(m)}$ implies $A^{(m+1)} \neq A^{(m)}$. Therefore a re-evaluation of the elements of A at the beginning of each iteration is necessary.

2.3 SIMPLE for compressible flows

For flows with a variable density that depends on pressure and temperature the system needs to be extended since the energy equation and the material

properties need to be included. With $\vec{h}^{(m+1)} = \vec{h}^{(m)} + \vec{h}'$, a naive approach would be to write

$$
\begin{pmatrix} A^{(m)} & M & 0 & 0 \\ C^{(m)} & 0 & 0 & 0 \\ 0 & 0 & G^{(m)} & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p}' \\ \vec{h}' \\ \vec{\varrho} \end{pmatrix} = \begin{pmatrix} \vec{b} - M\vec{p}^{(m)} \\ \vec{c} \\ \vec{g} - G\vec{h}^{(m)} \\ \vec{y} \end{pmatrix}, \tag{2.7}
$$

to apply a similar factorisation

$$
\begin{pmatrix} A & M & 0 & 0 \\ C & 0 & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} = \begin{pmatrix} A & 0 & 0 & 0 \\ C & S & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \cdot \begin{pmatrix} I & A^{-1}M & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}, \tag{2.8}
$$

and to deduce an algorithm analogous to Algorithm 1. A decisive improvement to this was suggested by Demirdžić et al. [9]. They substitute the dependence of the density on pressure into the continuity equation: for the time derivative of the density on the right-hand side of the continuity equation in equation (2.7) they write

$$
\frac{\partial \rho}{\partial t} = \frac{\rho' + \rho^{(m)} - \rho_0}{\Delta t}
$$

(ρ_0 is the density of the previous time step) and express the density update as multiple of the pressure update as

$$
\rho' = \frac{1}{RT}\vec{p}', \tag{2.9}
$$

since according to ideal gas law

$$
d\rho \approx \left.\frac{\partial \rho}{\partial p}\right|_T dp = \frac{1}{RT}dp.
$$

This introduces a dependency on the pressure into the continuity equation and improves the convergence significantly. The discretised system where this dependency is expressed by the linear operator S_p then reads

$$
\begin{pmatrix} A^{(m)} & M & 0 & 0 \\ C & S_p & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p}' \\ \vec{h}' \\ \vec{\varrho} \end{pmatrix} = \begin{pmatrix} \vec{b} - M\vec{p}^{(m)} \\ \vec{c} \\ \vec{g} - G\vec{h}^{(m)} \\ \vec{y} \end{pmatrix}.
$$

The method of Demirdžić et al. [9] uses this for a factorisation as in Eq. (2.8) to construct the SIMPLE algorithm for compressible flows, Algorithm 2. The advantage of the method of Demirdžić et al. is not only an improved robustness, see Demirdžić et al. [9], but also a more rapid convergence, e.g. the algorithm using (2.8) to construct the preconditioner requires 12 247 accumulated iterations in our example II with time resolution t0 (see Section 3.2) whereas with Demirdžić's method this number is only 8224. In the formulation of Algorithm 2, the enthalpy equation is solved for the enthalpy update $\vec{h}'$ in step 8

and the enthalpy is updated in step 9; it is mathematically equivalent to solve $G\vec{h}^{(m+1)} = \vec{g}$ for $\vec{h}^{(m+1)}$ in step 8 and skip step 9. In any event, an iteration is also necessary because of both, the coupling of the variables and the non-linearity. The operator G in step 8 of Algorithm 2 is evaluated based on the already updated velocity, but the density is taken from the previous time step, therefore we use the symbol $G^{(m+1/2)} := G(\vec{u}^{(m+1)}, \vec{\varrho}^{(m)})$.

Algorithm 2 SIMPLE algorithm for compressible flows

1: initialise, $m = 0$
2: **while** <not converged> **do**
3:     compute $\vec{\varrho}^{(m)} = f(\vec{p}^{(m)}, \vec{h}^{(m)})$
4:     solve $A^{(m)}\vec{u}^* = \vec{b} - M\vec{p}^{(m)}$
5:     solve $(\tilde{S}^{(m)} + S_p)\vec{p}' = \vec{c} - C\vec{u}^*$
6:     $\vec{u}^{(m+1)} = \vec{u}^* - \tilde{A}^{(m)-1}M\vec{p}'$
7:     $\vec{p}^{(m+1)} = \vec{p}^{(m)} + \vec{p}'$
8:     solve $G^{(m+1/2)}\vec{h}' = \vec{g} - G^{(m+1/2)}\vec{h}^{(m)}$
9:     $\vec{h}^{(m+1)} = \vec{h}^{(m)} + \vec{h}'$
10: **end while**

Equation (2.9) reflects only the dependence of the density on the pressure, but the density depends, as all extensive thermodynamic quantities, on two other quantities. The method we propose considers also the dependence of the density on the temperature at the point where the density on the right hand-side of the continuity is substituted. We write instead of equation (2.9)

$$\rho' = \frac{1}{RT}\left(p' + kh'\right),$$

where k is defined as $k := -p/(c_p T)$. Through this, the pressure-correction equation contains a contribution of enthalpy $\vec{h}$ in the form of a linear operator S_h. Moreover, the derivative of pressure with respect to time in the equation (2.3) can be written as

$$\frac{\partial p}{\partial t} = \frac{p' + p^{(m)} - p_0}{\Delta t},$$

where p_0 is the pressure at the previous time step. With this the pressure update is considered in the energy equation which results in a linear contribution of the pressure to the energy equation which we denote as G_p. If the enthalpy contribution in the continuity equation and the pressure contribution to the energy equation are moved to the left-hand side of these equations, the discretised system becomes

$$\begin{pmatrix} A^{(m)} & M & 0 & 0 \\ C & S_p & S_h & 0 \\ 0 & G_p & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{p}' \\ \vec{h}' \\ \vec{\varrho} \end{pmatrix} = \begin{pmatrix} \vec{b} - M\vec{p}^{(m)} \\ \vec{c} \\ \vec{g} - G\vec{h}^{(m)} \\ \vec{y} \end{pmatrix}.$$

It can be factorised as

$$\begin{pmatrix} A & M & 0 & 0 \\ C & S_p & S_h & 0 \\ 0 & G_p & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} = \begin{pmatrix} A & 0 & 0 & 0 \\ C & S+S_p & S_h & 0 \\ 0 & G_p & G & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \cdot \begin{pmatrix} I & A^{-1}M & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}.$$

The iterative algorithm exploiting this is Algorithm 3. Note that here the velocity is not yet updated when the operator G is evaluated; therefore here $G^{(m)} := G(\vec{u}^{(m)}, \vec{\varrho}^{(m)})$ is used.

Algorithm 3 SIMPLE algorithm with pressure-enthalpy coupling

1: initialise, $m = 0$
2: **while** <not converged> **do**
3:     compute $\vec{\varrho}^{(m)} = f(\vec{p}^{(m)}, \vec{h}^{(m)})$
4:     solve $A^{(m)}\vec{u}^* = \vec{b} - M\vec{p}^{(m)}$
5:     solve $\begin{pmatrix} \tilde{S}^{(m)} + S_p & S_h \\ G_p & G^{(m)} \end{pmatrix} \begin{pmatrix} \vec{p}' \\ \vec{h}' \end{pmatrix} = \begin{pmatrix} \vec{c} - C\vec{u}^* \\ \vec{g} \end{pmatrix}$
6:     $\vec{u}^{(m+1)} = \vec{u}^* - \tilde{A}^{(m)-1}M\vec{p}'$
7:     $\vec{p}^{(m+1)} = \vec{p}^{(m)} + \vec{p}'$
8:     $\vec{h}^{(m+1)} = \vec{h}^{(m)} + \vec{h}'$
9: **end while**

While in conventional SIMPLE separated linear equation systems for each of the individual variables (velocity components, pressure-correction, enthalpy) are solved, this pressure-enthalpy coupling scheme requires the solution of the coupled linear system in step 5 of Algorithm 3. This system has twice as many unknowns as the scalar systems of the segregated method and it is nonsymmetric; it has the same properties as discretised problems of mixed elliptic-hyperbolic type usually have. Its numerical treatment is more difficult than that of the scalar systems. The advantage of the pressure-enthalpy coupling is that it enhances the convergence of the iteration since it couples both variables more closely in a common linear system. Since the iteration is also due to the non-linearity of the system, a significant improvement will not be observed in every individual case: Roughly speaking, a large number of iterations is required by the non-linearity if the variables, in particular the velocity, change significantly in the course of the iteration. This is the case if large time steps are used. On the other hand, if the time step is small (or a very good initial guess for the velocity exists), then the iteration is needed for the coupling of the system. In the latter case the pressure-enthalpy coupling will be particularly efficient.

3   Benchmarks

The pressure-enthalpy coupling has been implemented in the CFD package FIRE$^{(R)}$ 2010 (distributed by AVL) that uses the SIMPLE method as the standard algorithm. The benchmarks were run on a Linux-cluster where each node is equipped with 2 quad-cores (Intel Xeon CPU X5365, 3.00 GHz, main memory 16 GB, L1-cache $2 \cdot 4 \cdot 32$ kB, L2-cache $2 \cdot 4$ MB, shared between two cores). With regard to one of our examples, namely example III, it has to be mentioned that the nodes are linked by an Infiniband interconnect with a measured bandwidth of 750 Mbit/s and a latency of 3.3 μs.

The implementation of the SIMPLE method is that of a commercial code. The employed discretisation practice was explored in other publications starting with Demirdžić and Muzaferija [8], and continuing with Ferziger and Perić [15],

Marthur and Marthy [20], Basara [3], Basara et al. [4] etc. Hence the method has been applied and proved on various applications and a comparison of the obtained results to measurements or to analytical solutions is skipped here. The parallelization of FIRE$^{(R)}$ 2010 relies on a domain decomposition based on the graph partitioning algorithm METIS, see Karypis and Kumar [17]. For the spatial finite-volume discretisation it is sufficient to provide an overlap of one layer of finite volumes between two adjacent subdomains. The data associated with the finite volumes in this layer needs to be exchanged after the values have been (re-)calculated, i.e. after vector updates and the solution of linear systems. This exchange is done through synchronous point-to-point communication using MPI. The linear solver uses the same domain decomposition as the other parts of FIRE$^{(R)}$ 2010; when it is called, it receives therefore the distributed matrix information and no redistribution of data is necessary. The parallelization strategy of the linear solver has been discussed in detail in a previous publication, see Emans [14]. There are no principal differences in the parallelization of SIMPLE and SIMPLE with pressure-enthalpy coupling.

The pressure-correction equation of the segregated SIMPLE is solved by an AMG-preconditioned conjugate gradient iterative method where the AMG is of Smoothed Aggregation type of Vaněk [28], for details about the implementation see Emans [14]. The coupled system is solved by an AMG-preconditioned GMRES that is restarted each 10 iterations; the character of the system is considered through a unknown-based approach, see Clees [6]. The preconditioning is done as v-cycle with two Gauß–Seidel sweeps for both, pre- and post-smoothing. The grid hierarchy is setup by aggregation of pairs where the selection of the pairs follows the algorithm described in detail by Notay [21]; further specific details of the algorithm and of the implementation are reported in a paper [13] dedicated to this issue. All nonsymmetric systems apart from the coupled system, but including the energy equation for the segregated SIMPLE scheme, are solved by a BiCGstab algorithm that is preconditioned by a standard ILU(0) factorisation, see Saad [24].

3.1 Gasoline engine – example I

The two benchmark cases in this example are short but representative periods taken from an unsteady simulation of a full cycle of a four cylinder gasoline engine. The variation of the position of the piston during the engine cycle as well as some characteristic results of the simulation are shown in Figure 1.

The simulation comprises the gas flow and the combustion in one of the cylinders. The stroke of the cylinder is 81.4 mm, the bore is 79.0 mm yielding a (maximum) volume of 0.4 l (per cylinder). The engine runs at 5500 rpm. Each benchmark consists of a few time-steps. As usual in CFD simulations of engines, the three-dimensional computational domain is subject to change in time: It contains the interior of the cylinder and the ports through which the air is sucked into the cylinder or hot gas is expelled from it. The piston surface is a moving boundary. A three-dimensional simulation of a full engine cycle comprises the simulation of the (compressible) flow of cold air into the cylinder while the piston is moving downward, the subsequent compression
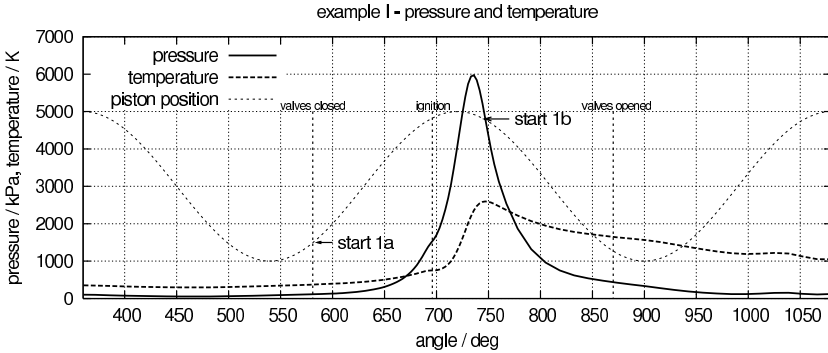
Figure 1. Scheme of the engine cycle and characteristic simulation results; the arrows point to the start of the benchmark cases; piston position schematic.

after the valves are closed, the combustion of the explosive mixture, and the discharge of the hot gas while the piston moves upward. In the simulation the modelling of the fluid flow by the above discussed system of equations is amended by the standard k-ε turbulence model by Jones and Launder [16]; the fluid properties are those of air. The model fuel that is burnt is octane; for the sake of simplicity, the eddy break-up model by Magnussen and Hjertager [19] is used to simulate the combustion process. Two and three additional transport equations are solved for the turbulence model and for the combustion model, respectively. The simulation is carried out with the FIRE$^{(R)}$ 2010 code that has been validated for this type of problem many times, see e.g. Priesching et al. [23]. Further information about the setup can be found in Emans [12]. The computations were done with eight parallel processes located on one node of the cluster. The iteration is terminated, if the normalized 1-norm of the residuals of the continuity equation has reached a tolerance of $1.0 \cdot 10^{-3}$; the tolerance of the momentum equations and the energy equation is $1.0 \cdot 10^{-2}$.

Example 1a comprises 4 degree crank angle α within the compression phase, i.e. we observe a closed system with adiabatic walls; the unstructured mesh consists of 238 000 mostly hexagonal cells. This period is simulated with four different time resolutions: $\delta\alpha_1 = 0.4^o$, $\delta\alpha_2 = 0.2^o$, $\delta\alpha_3 = 0.05^o$, and $\delta\alpha_4 = 0.01^o$ This corresponds to time steps of $\delta t_1 = 1.2 \cdot 10^{-6}s$, $\delta t_2 = 0.60 \cdot 10^{-6}s$, $\delta t_3 = 0.15 \cdot 10^{-6}s$ $\delta t_4 = 0.03 \cdot 10^{-6}s$; with these time steps we cover the range that is relevant for applications in engineering. Each calculation has been carried out once with segregated SIMPLE (SI) that uses block-pressure adjustment, see Ahmadi-Befrui [1], and with SIMPLE with pressure-enthalpy coupling (PH).

Example 1b comprises 4 degree crank angle within the combustion phase, i.e. here, additionally to fluid flow, combustion is modelled; the unstructured mesh consists of 254 000 mostly hexagonal cells. Other settings as well as the hardware are identical. The meshes used in example Ia and example Ib are shown in Figure 2.

The data shown in Figure 3 demonstrates that the pressure-enthalpy coupling improves the convergence of SIMPLE significantly in all cases. The com-
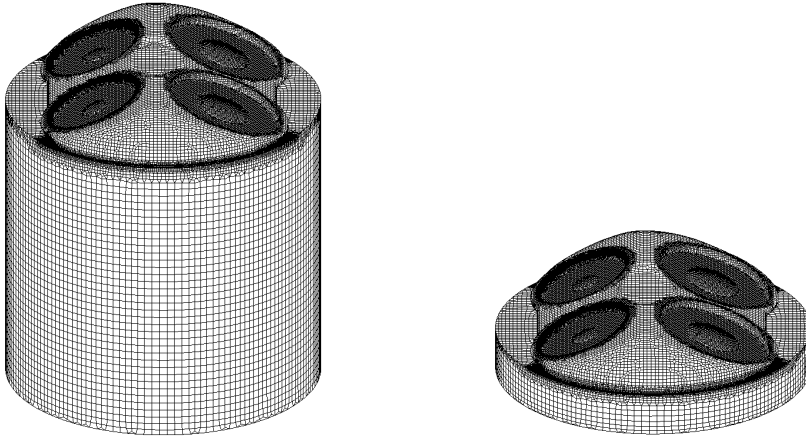
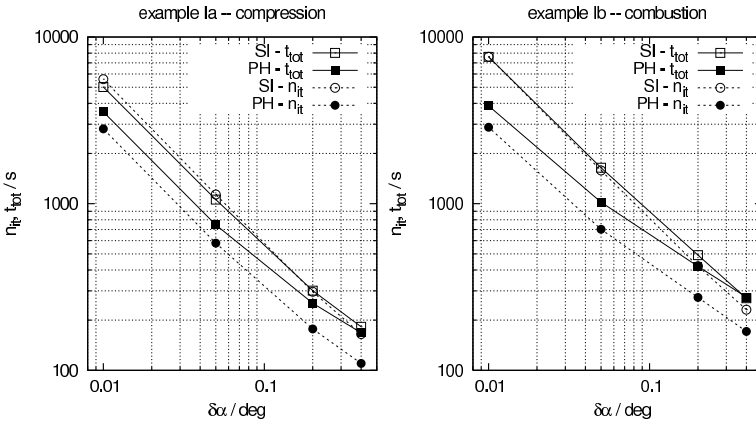**Figure 2.** Meshes of example Ia (compression) left and example Ib (combustion) right.

Figure 3. Total computing time t_{tot} and number of SIMPLE-iterations n_{it} of the benchmarks of examples Ia and Ib.

puting time is reduced to a lower extent than the number of iterations since the solution of the coupled system is more expensive than the solution of the systems in the segregated approach. For the largest part this is due to the more expensive solution of the linear system: The pressure-correction equation is a semi-definite or definite system in the case of the conventional SIMPLE algorithm, but it has no beneficial properties apart from the sparseness that can be exploited by the solver in the case of the pressure-enthalpy coupling algorithm. In most cases, however, the computing time is still reduced significantly. The exception is the coarsest time resolution of example Ib.

The savings in terms of SIMPLE iterations increase with decreased time step. The savings range from 63 % ($\delta\alpha = 0.01^o$, example Ib) to 26 % ($\delta\alpha = 0.4^o$, example Ib). The maximum saving in terms of computing time is 49 % ($\delta\alpha = 0.01^o$, example Ib).
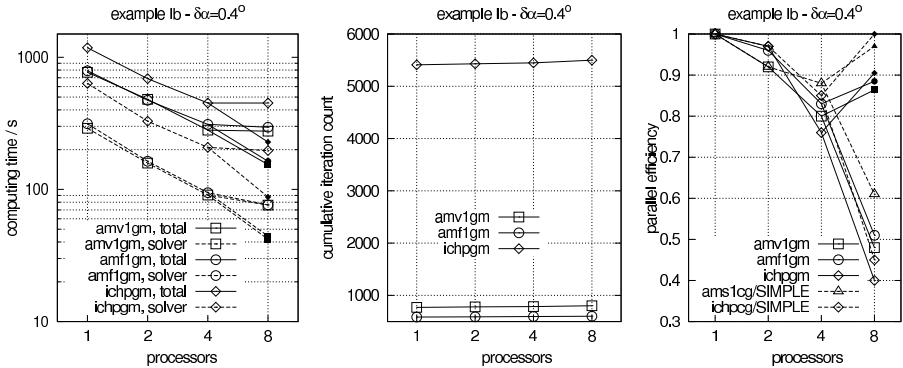
Figure 4. Characterization of the linear solver: computing times (total simulation time, solver time), cumulative iteration count, and parallel efficiency. Empty symbols refer to the calculation with eight processes on one node, filled symbols to the calculation with eight processes on two nodes, distributed as 2×4.

Since the solution of the coupled linear system is more difficult to obtain than the solution of the corresponding scalar systems, it is necessary to examine if our choice of the solver is an appropriate one with regard to algorithmic aspects. Moreover, due to memory bus saturation, in particular of the multi-core systems, the efficiency of parallel solvers tends to degrade with increased problems size, as e.g. Starikovičius et al. [25] have observed; it is therefore also necessary to check if our solver is also technically a good choice. For this we focus on the calculation of example Ib with a resolution of $\delta\alpha_1 = 0.4^o$ where 118 coupled linear systems are solved. We have repeated the calculation, using different numbers of processes and different linear solver algorithms. The computing times of the whole calculation and that of the solver part only, the iteration count, and the parallel efficiency (defined as e.g. in Emans [12] or Starikovičius et al. [26]) of the solver part are shown in Figure 4. The short amv1gm stands for the solver we use for the other calculations in this article; the solver amf1gm is a similar AMG algorithm which uses the same coarse-grid construction method as amv1gm, but employs a f-cycle, see Trottenberg et al. [27] for the definition, instead a v-cycle; algorithm ichpgm is a much simpler solver: It has an ILU(0) preconditioner, see Saad [24], instead of a multigrid method while the solver is still GMRES; this preconditioner is used e.g. by Chen et al. [5] for similar systems. We observe that the multigrid based solvers (amf1gm and amv1gm) are much faster than the method with the ILU(0) preconditioner (ichpgm). The algorithm amf1gm converges faster than amv1gm, but since one iteration (one GMRES iteration and the preconditioning using the respective cycle) is more expensive, its computing times tend to be a bit longer; furthermore, since the scalability of the work on the small grids is worse than that on the fine grids, algorithm amf1gm (doing more work on these coarse-grids) behaves slightly worse than amv1gm in particular in parallel.

The linear problems that are solved for SIMPLE-H are larger in terms of unknowns than that of SIMPLE and their matrices have more non-zero elements per row; since the memory traffic is a limiting factor for the performance as has

been pointed out e.g. in own publication [12, 14], but also e.g. in the papers of Starikovičius et al. [25, 26], it is interesting to observe possible differences in the behaviour of the parallel versions of the linear solvers of the pressure-correction equation in SIMPLE and SIMPLE-H. For this purpose we have added the parallel efficiency of our solver for SIMPLE, a conjugate gradient solver with v-cycle Smoothed Aggregation AMG as preconditioner (ams1cg), and of a standard solver for these kind of symmetric positive definite problems, a conjugate gradient solver with an incomplete Cholesky factorisation as preconditioner, see Saad [24], (ichpcg), to the diagram with the parallel efficiencies. One can observe that the parallel behaviour of the coupled solver is worse than that of the scalar solver, but it lies in the range reported in the mentioned papers (for problem where the number of unknowns is approximately the same). All solvers show only a slight increase in the number of iterations; on the particular machine, the calculation with eight parallel processes on one node has a very poor parallel efficiency: This is due to the known problem of memory bus exhaustion; it can be remedied by moving four processes to a second node. The linear solver takes between 20% and 40% of the total computing time.

3.2 Diesel engine segment – example II

An important application of CFD in engine development is the simulation of the compression and the combustion of diesel engines. The main objective is to improve the geometry and the parameters of the combustion, e.g. the positioning and the timing of the diesel injection. In practice, it is often sufficient to consider segments of the cylinders instead of the entire geometry; in such cases periodic boundary conditions are applied to map the whole volume of the cylinder into the computational domain. In this section we examine in which situations the pressure-enthalpy coupling can be employed efficiently to such kind of calculations. The bore of the cylinder is 0.086 m, the stroke is 0.098 m, yielding a volume of 0.57 l. The meshes of the largest extension of the computational domain (bottom dead centre) and that of its smallest extension (top dead centre) are shown in Figure 5. The largest mesh consists of approximately 135 000 cells, the smallest of 48 000 cells. The engine speed is 3000 rpm, the simulation starts at crank angle 610° and stops at 860°. The fuel (diesel) is injected between crank angle 713.5^o and 734.8^o. A Lagrangian model is applied to simulate the spray particles; the evaporation is modelled by the Dukowicz approach [11]. The 3-Zones ECFM (Extended Coherent Flame Model) by Colin and Benkenida [7] was employed to model the combustion. The turbulence is modelled by a standard k-ε model according to Jones and Launder [16]. The computations were done with four parallel processes located on one node of the cluster.

The time resolution is varied: Table 1 informs about the time stepping. The iteration is stopped, if the normalized 1-norm of the residuals of the continuity equation has reached a tolerance of $1.0 \cdot 10^{-5}$; the tolerance of the momentum equations and the energy equation is $1.0 \cdot 10^{-4}$.

Characteristic results of the simulation are shown in Figure 6. The range of time resolutions we examine reflects the requirements in engineering: In
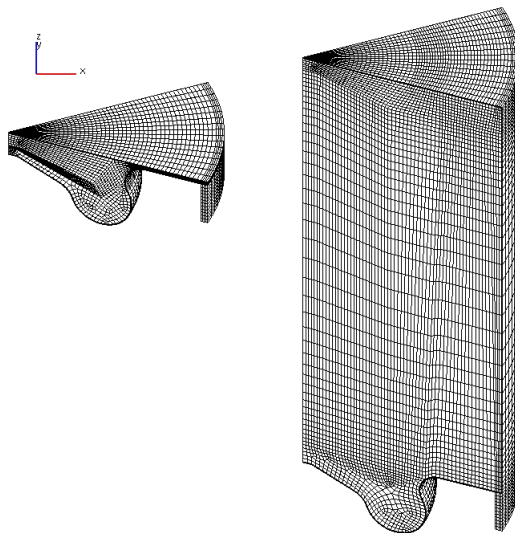
Figure 5.   Smallest and largest mesh of the benchmarks of example II.

Table 1.   Time step δt (in brackets: crank angle increment $\delta\alpha$) in benchmarks of example II.

α	$610^o \leqslant \alpha \leqslant 709^o$	$709^o < \alpha \leqslant 740^o$	$740^o < \alpha \leqslant 860^o$
t0	$1.0^o/0.556 \cdot 10^{-3}s$	$0.2^o/0.278 \cdot 10^{-4}s$	$1.0^o/0.556 \cdot 10^{-3}s$
t1	$0.5^o/0.278 \cdot 10^{-4}s$	$0.1^o/0.556 \cdot 10^{-5}s$	$0.5^o/0.278 \cdot 10^{-4}s$
t2	$0.25^o/0.139 \cdot 10^{-4}s$	$0.05^o/0.278 \cdot 10^{-6}s$	$0.25^o/0.139 \cdot 10^{-4}s$
t3	$0.05^o/0.278 \cdot 10^{-5}s$	$0.01^o/0.556 \cdot 10^{-6}s$	$0.05^o/0.278 \cdot 10^{-5}s$

practice the choice of the time step, as long as it is not dictated by numerical constraints, depends on the purpose of the simulation.

Table 2.   Impact of different time steps on results of example II: maximum velocity, maximum and minimum density at $\alpha = 725^o$, final NO and soot concentration.

	t0	t1	t2	t3
max. velocity at $\alpha = 725^o$	182.2 m/s	218.2 m/s	222.7 m/s	233.3 m/s
max. density at $\alpha = 725^o$	39.59 kg/m^3	41.23 kg/m^3	41.03 kg/m^3	40.6 kg/m^3
min. density at $\alpha = 725^o$	1.220 kg/m^3	1.202 kg/m^3	1.165 kg/m^3	1.167 kg/m^3
final NO concentration	$3.52 \cdot 10^{-4}$	$3.17 \cdot 10^{-4}$	$2.51 \cdot 10^{-4}$	$1.92 \cdot 10^{-4}$
final soot concentration	$5.95 \cdot 10^{-4}$	$5.45 \cdot 10^{-4}$	$3.76 \cdot 10^{-4}$	$2.49 \cdot 10^{-4}$

The data in Table 2 show typical examples of primary flow values that change significantly when the time resolution is changed. In order to illustrate the consequences that such differences can entail, we have added to Table 2 the output of an emission model that relies on the fluid flow solution: The NO concentration has been calculated by an extended Zeldovich approach, see

Lavoie et al. [18], and the soot formation was modelled by a kinetic model, see Wang and Frenklach [29].
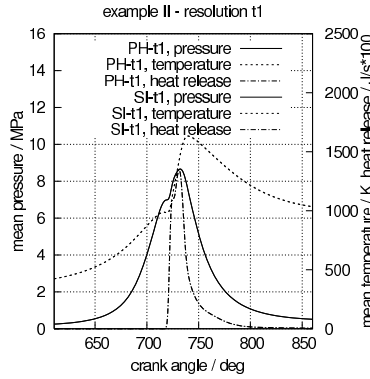

Figure 6. Results of flow simulation and heat release of example I (resolution t1).

The results plotted in Figure 6 exemplify that the prediction of the fluid flow is exactly the same for both methods; observe that both curves coincide. This means that, if both algorithms converge and the tolerances are set appropriately (which is the case here), then the flow solution obtained by the pressure-enthalpy coupling scheme is equivalent to that of conventional SIMPLE.

Table 3. Wall clock times of entire computation and computation of the solution of the pressure-correction equation (coupled system for coupled system) and accumulate iteration count for example II.

	t_{tot}	t_{sol}	n_{it}
PH-t0	$4959s$	$840s$	7392
PH-t1	$7408s$	$1309s$	$11\,865$
PH-t2	$13\,132s$	$2114s$	$20\,935$
PH-t3	$30\,322s$	$3983s$	$45\,545$

	t_{tot}	t_{sol}	n_{it}
SI-t0	$4423s$	$356s$	8224
SI-t1	$7626s$	$616s$	$14\,542$
SI-t2	$15\,545s$	$1164s$	$28\,462$
SI-t3	$45\,792s$	$2823s$	$72\,677$

The wall clock times for calculations with four parallel processes on one node of the cluster and the iteration counts in Figure 7 show that the pressure-enthalpy coupling scheme converges generally more rapidly than SIMPLE. As in the previous example the advantage of this new scheme increases with smaller time step. Although its convergence is superior, it is sometimes slower than SIMPLE. This is a consequence of the fact that a larger linear system with less favourable properties needs to be solved. Table 3 shows the total wall clock times for the entire computation and for the solution of the pressure-correction
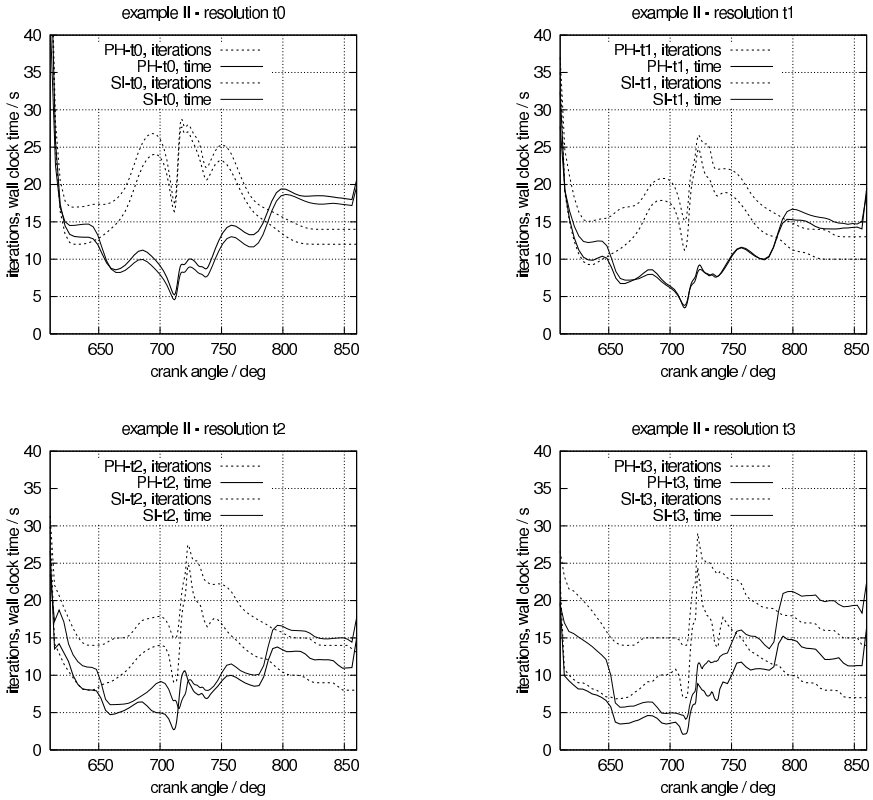
Figure 7.   Iteration count and wall clock times for benchmarks of example II.

equation. The definition of a criterion which allows in advance to choose the faster algorithm is difficult: It does not only depend on the problem and the current values of the unknowns, but also on the parameters and, in practice, the load of the used computer. The elaboration of appropriate strategies will be subject to future work.

Note that in this simulation the combustion was modelled by 3-Zones ECFM which requires the solution of 27 transport equations. The accelerating effect of the pressure-enthalpy coupling is due to the reduction of the number of SIMPLE iterations. Since in every iteration the coefficients of these transport equations need to be calculated and the equations need to be solved, a reduction of the number of iterations through the pressure-enthalpy coupling scheme is particularly beneficial if many of such transport equations are solved. Although there are simpler combustion models such as the mentioned eddy break-up model by Magnussen and Hjertager [19], advances in combustion modelling will require the solution of significantly larger numbers of transport equations, e.g. Antonelli et al. [2] track already 62 chemical species for which individual transport equations are solved. The presented results indicate that for such complex simulations the pressure-enthalpy coupling can be an attractive algorithm.

3.3   Intake port – example III

Another advanced application of CFD in engine development is the prediction of the fuel distribution in the intake port of gasoline engines. Here the fuel forms a film at the duct walls. The extension, thickness and position of this film influence the mixture formation. This wallfilm is a very complex two-phase problem for which e.g. additional two-dimensional models might be employed. Those wallfilm models interact with the prediction of the motion of the gas in the intake port. Our benchmark is the simulation of the fluid motion in the intake port during the time the inlet valve is closed. During this time, the computational domain is closed; fluid motion and a pressure wave are caused by the preceding flow of the air/fuel mixture into the cylinder. For the purpose of this study we consider only the flow of the air/fuel mixture; the wallfilm model is deactivated in order to keep the description of the case simple. The computations were done with twelve parallel processes located on two nodes of the cluster (2×6).

The computational domain is resolved spatially by a mesh of about 2.1 mio. cells which is shown in Figure 8. The uniform time step of the simulation is $1.852 \cdot 10^{-5}s$. Again the standard k-ε model according to Jones and Launder [16] is applied. The tolerances for continuity, momentum, and energy equation are $1.0 \cdot 10^{-4}$ in this case. The calculated mean physical values, iteration counts, and wall clock times of this simulation are plotted in Figure 9.
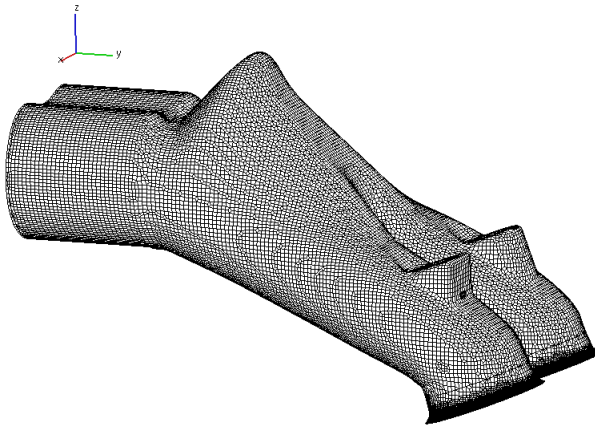

Figure 8.   Mesh of example III.

This case is another example for the pressure-enthalpy coupling scheme being an advantageous improvement to SIMPLE. The wall clock times for SIMPLE and SIMPLE with pressure-enthalpy coupling are $70\,900s$ (including $8944s$ for the solution of the coupled system) and $52\,068s$ (including $17\,955s$ for the solution of the pressure-correction equation) respectively.
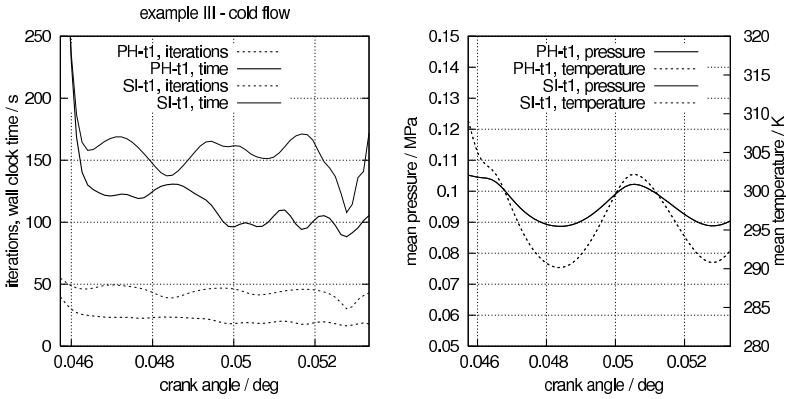
Figure 9. Computing time and iteration count (left) and results (right) of example III.

4   Conclusions

The presented pressure-enthalpy coupling scheme reduces the number of SIM-PLE iterations by up to 63% in practically relevant engine simulations. The disadvantage of the method is that the computational effort to solve linear systems for the pressure correction mechanism is increased compared to seg-regated SIMPLE. The method is particularly efficient if small time steps are chosen; for large time steps the method happens to be slower than conventional SIMPLE since in such cases the non-linearity and not the coupling of the equa-tions tends to necessitate a large portion of the iterations; in such cases the increased effort to solve the linear systems is not compensated by the reduction of the number of iterations. Nevertheless we can show that the computing time of a complex engine simulation using the pressure-enthalpy coupling scheme is reduced by 49% compared to segregated SIMPLE; the lower percentage-wise reduction of computing time compared to that of the iteration count reflects the increased computational cost to solve the linear systems. Since SIMPLE-H is not always faster than the conventional SIMPLE, it would be particularly useful for engineering application to dispose of a criterion allowing for the au-tomatic choice the faster algorithm. The development of such a criterion is, along with research for a more efficient linear solver, subject to future work.

Acknowledgement

References

[1] B. Ahmadi-Befrui. *Analysis of flow evolution in the cylinder of motored recipro-cating engines.* PhD thesis, University of London, 1985.

[2] L. Antonelli, P. Belardini, P. D'Ambra, F. Gregoretti and G. Oliva. A distributed combustion solver for engine simulations on grids. *J. Comput. Appl. Math.*, **226**:197–204, 2009. http://dx.doi.org/10.1016/j.cam.2008.08.002.

[3] B. Basara. Employment of the second-moment turbulence closure on arbitrary unstructured grids. *Internat. J. Numer. Methods Fluids*, **44**:377–407, 2004. http://dx.doi.org/10.1002/fld.646.

[4] B. Basara, A. Alajbegovic and D. Beader. Simulation of single- and two-phase flows on sliding unstructured meshes using finite volume method. *Internat. J. Numer. Methods Fluids*, **45**:1137–1159, 2004. http://dx.doi.org/10.1002/fld.733.

[5] Z.J. Chen and A.J. Przekwas. A coupled pressure-based computational method for incompressible/compressible flows. *J. Comput. Phys.*, **229**:9159–9165, 2010. http://dx.doi.org/10.1016/j.jcp.2010.08.029.

[6] T. Clees and K. Stüben. Algebraic multigrid for industrial semiconductor device simulation. In E. Bänsch(Ed.), *Challenges in Scientific Computing – CISC2003*, volume 35 of *Lect. Notes Comput. Sci. Eng.*, pp. 110–130. Springer-Verlag, 2002.

[7] O. Colin and A. Benkenida. The 3-zones extended coherent flame model (ECFM3Z) for computing premixed/diffusion combustion. *Oil & Gas Science and Technology Rev. IFP*, **59**(6):593–609, 2004.

[8] I. Demirdžić and S. Muzaferija. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Comput. Methods Appl. Mech. Engrg.*, **125**:235–255, 1995. http://dx.doi.org/10.1016/0045-7825(95)00800-G.

[9] I. Demirdžić, Ž. Lilek and M. Perić. A collocated finite volume method for predicting flow at all speeds. *Internat. J. Numer. Methods Fluids*, **16**:1029–1050, 1993. http://dx.doi.org/10.1002/fld.1650161202.

[10] J.P. Van Doormaal and G.D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, **7**:147–163, 1984. http://dx.doi.org/10.1080/01495728408961817.

[11] J.K. Dukowicz. A particle-fluid numerical model for liquid sprays. *J. Comput. Phys.*, **35**:229–253, 1980. http://dx.doi.org/10.1016/0021-9991(80)90087-X.

[12] M. Emans. AMG for linear systems in engine flow simulations. In R. Wyrzykowski, J. Dongarra, K. Karczewski and J. Wasniewski(Eds.), *PPAM2009, Part II*, volume 6068 of *Lecture Notes in Comput. Sci.*, pp. 350–359. Springer-Verlag, Berlin, Heidelberg, 2010.

[13] M. Emans. Approximate solutions of linear systems in CFD applications. *SIAM J. Sci. Comput.*, **32**:2235–2254, 2010. http://dx.doi.org/10.1137/090774860.

[14] M. Emans. Performance of parallel AMG-preconditioners on CFD-codes for weakly compressible flows. *Parallel Comput.*, **36**:326–338, 2010. http://dx.doi.org/10.1016/j.parco.2009.12.009.

[15] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag, Berin, Heidelberg, 1996.

[16] W.P. Jones and B.E. Launder. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, **15**:301–314, 1972. http://dx.doi.org/10.1016/0017-9310(72)90076-2.

[17] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, **20**:359–392, 1998. http://dx.doi.org/10.1137/S1064827595287997.

[18] G.A. Lavoie, J.B. Heywood and J.C. Keck. Experimental and theoretical investi-
     gation of the nitric oxide formation in internal combustion engines. *Combustion
     Science and Technology*, **1**:311–326, 1970.
     http://dx.doi.org/10.1080/00102206908952211.

[19] B.F. Magnussen and B.H. Hjertager. On mathematical modeling of turbulent
     combustion with special emphasis on soot formation and combustion. In *Proc.
     of the 16th Int. Symp. on Combustion 1976*, pp. 719–729. Combustion Institute
     Pittsburg, 1976.

[20] S.R. Marthur and J.Y. Murthy. A pressure based method for unstructured
     meshes. *Numerical Heat Transfer B*, **31**:195–215, 1997.
     http://dx.doi.org/10.1080/10407799708915105.

[21] Y. Notay. An aggregation-based algebraic multigrid method. *Electron. Trans.
     Numer. Anal.*, **37**:123–146, 2010.

[22] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and
     momentum transfer in three-dimensional parabolic flows. *International Journal
     Heat Mass Transfer*, **15**:1787–1806, 1972.
     http://dx.doi.org/10.1016/0017-9310(72)90054-3.

[23] P. Priesching, R. Wanker, P. Cartellieri and R. Tatschl. CFD modelling of HCCI
     engine combustion – validation and application. In *Proceedings of ICE 2003*,
     2003.

[24] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 3 edition, 2003.

[25] V. Starikovičius, R. Čiegis and O. Iliev. A parallel solver for optimization of oil
     filters. *Math. Model. Anal.*, **16**:326–342, 2011.
     http://dx.doi.org/10.3846/13926292.2011.582591.

[26] V. Starikovičius, R. Čiegis, O. Iliev and Z. Lakdawala. A parallel solver for the
     3D simulation of flows through oil filters. In R. Čiegis, D. Horty, B. Kapstrom
     and J. Žilinskas(Eds.), *Parallel Scientific Computing and Optimisation, Advances
     and Applications*, volume 27, pp. 181–192. Springer, 2009.

[27] U. Trottenberg, C. Oosterlee and Anton Schüller. *Multigrid*. Elsevier Academic
     Press Amsterdam, 2001.

[28] P. Vaněk, J. Mandel and M. Brezina. Algebraic multigrid by smoothed aggre-
     gation for second and fourth order elliptic problems. *Computing*, **56**:179–196,
     1996. http://dx.doi.org/10.1007/BF02238511.

[29] H. Wang and M. Frenklach. A detailed kinetic modeling study of aromatics
     formation in laminar premixed acetylene and ethylene flames. *Combustion and
     Flame*, **110**:173–221, 1997. http://dx.doi.org/10.1016/S0010-2180(97)00068-0.