

The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces

C. Vuik^{a,*}, A. Saghir^b and G. P. Boerstoel^{b,2}

^a *Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, Delft, Netherlands*

^b *TNO-TPD Institute of Applied Physics, Stieltjesweg 1, Delft, Netherlands*

SUMMARY

Numerical modeling of the melting and combustion process is an important tool in gaining understanding of the physical and chemical phenomena that occur in a gas- or oil-fired glass-melting furnace. The incompressible Navier–Stokes equations are used to model the gas flow in the furnace. The discrete Navier–Stokes equations are solved by the SIMPLE(R) pressure-correction method. In these applications, many SIMPLE(R) iterations are necessary to obtain an accurate solution. In this paper, Krylov accelerated versions are proposed: GCR-SIMPLE(R). The properties of these methods are investigated for a simple two-dimensional flow. Thereafter, the efficiencies of the methods are compared for three-dimensional flows in industrial glass-melting furnaces. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: combustion; efficiency; flow problem; Krylov acceleration; SIMPLE(R) method

1. INTRODUCTION

The increasing demand in quality, production efficiency, and environmental issues drive the glass producer to optimizing their melting furnaces. The quality demand is so high and the melting behavior so complex that a complete understanding of all important physical and chemical phenomena during the melting process is required to help us further. A very important and powerful tool in gaining this understanding is numerical modeling of the complete melting and combustion process and their interaction. At the TNO Institute of Applied Physics, a computational fluid dynamic (CFD) simulation model for gas- and oil-fired glass-melting furnaces, WISH3D-GTM, has been developed. This is a complete model for glass-melting furnaces, describing the combustion space and glass bath, and predicting the effects on melting performance and glass quality. The model is successfully used by the glass

* Correspondence to: Mekelweg 4, 2628 CD Delft, Netherlands.

¹ E-mail: c.vuik@math.tudelft.nl

² E-mail: boerstoel@tpd.tno.nl

industry and furnace manufacturers for product quality improvement, optimization of new furnace designs, and trouble-shooting.

The simulation of a complete glass-melting furnace often results in large computation times. One of the reasons for this is that the model uses the so-called SIMPLE(R) pressure-correction method to solve the incompressible Navier–Stokes equations. It is well known that the SIMPLE(R) method often needs many iterations before an accurate solution is obtained. To reduce the large computation times of the SIMPLE(R) method, a Krylov sub-space acceleration of the SIMPLE(R) method has been developed. The generalized conjugate residuals (GCR) method has been used for this purpose since this method can be applied to the non-symmetric matrices that result from the discretization of the Navier–Stokes equations. The new method presented in this paper is called GCR-SIMPLE(R).

2. DESCRIPTION OF THE MATHEMATICAL MODEL

The numerical model TNO-WISH3D takes into account all relevant phenomena in the combustion chamber of glass-melting furnaces. In the computer code, the following sub-models have been implemented: three-dimensional flow is described by the Navier–Stokes equations; turbulence is accounted for by the standard k – ε model, including wall-functions; combustion of natural gas is described by the conserved scalar approach to high temperature, non-premixed combustion; combustion of oil is described by a Lagrangian particle tracking method, including the vaporization of the oil droplets; chemistry is described with a one-step global reaction; radiative heat transfer is modeled by the discrete transfer method, where a composition dependent absorption–emission coefficient is used; a NO_x -postprocessor is used for the prediction of thermal NO-formation according to the Zel’dovich mechanism; a one-equation model predicting soot formation and oxidation has been incorporated; conduction in the furnace walls has been taken into account (conjugate heat transfer); and physical properties of the gas mixture are both composition- and temperature-dependent. The convection–diffusion equations obtained from the sub-models are discretized by the finite volume method using a hybrid scheme for the discretization of the convection terms. The pressure field is obtained by the SIMPLE or SIMPLER algorithm [1]. For a more detailed description of the sub-models and the numerical procedure used in TNO-WISH3D, the reader is referred to References [2,3].

3. THE GCR-SIMPLE(R) METHOD FOR THE STOKES PROBLEM

After discretization of the incompressible Stokes equation, the resulting linear system is symmetric and positive indefinite. Discretization of the continuity equation leads to a zero block on the main diagonal. This leads to serious problems when linear problem solvers are used. Various methods are known to overcome these difficulties: the pressure–matrix method [4], the Uzawa method [5,6], SIMPLE-type methods [1,7], the penalty method [8], the pressure-correction method [9], the PISO method [10], etc. For an overview of these methods we refer to Reference [11], Section 9.6. In CFD packages, a popular method is the SIMPLE

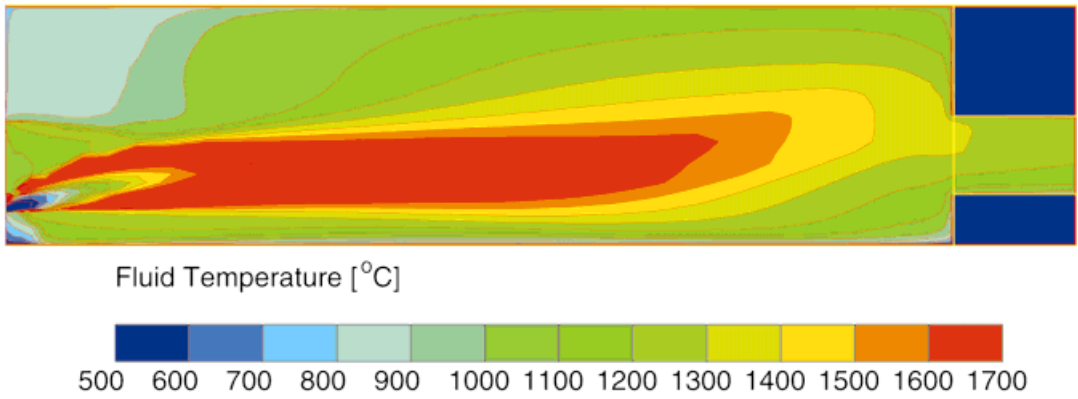


Plate 1. The temperature contours of the IFRF furnace using the GCR-SIMPLER method.

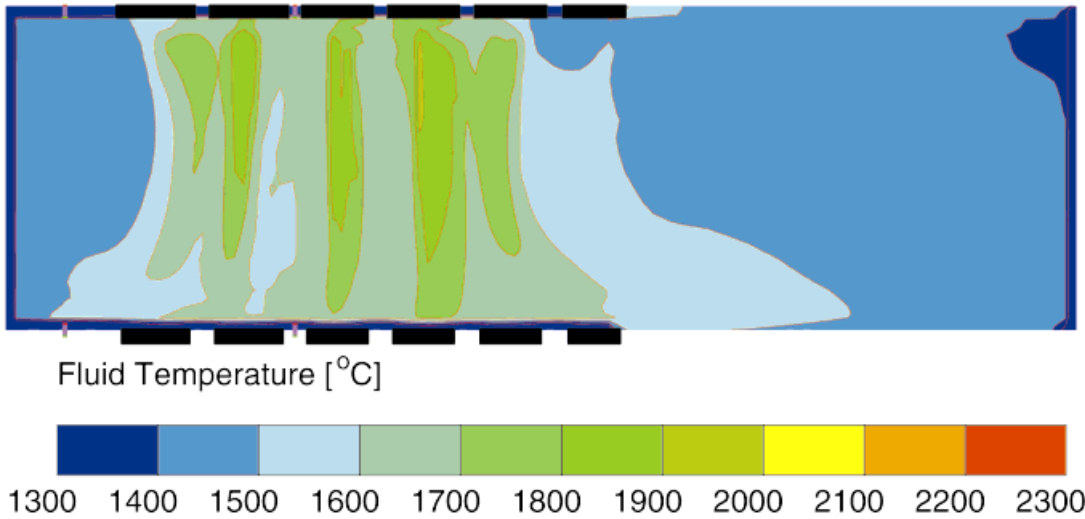


Plate 2. The temperature contours of the Ford float furnace using the GCR-SIMPLER method.

method proposed by Patankar and Spalding [12] or one of its variants: SIMPLER [1], SIMPLEST [13], or SIMPLEC [14].

In many applications, the SIMPLE method needs many iterations before an accurate solution is reached. Various authors consider a multigrid acceleration of the SIMPLE method [15–21]. In this paper we consider a Krylov sub-space acceleration of the SIMPLE(R) method [22]. The reason for this is that Krylov methods have only a small amount of overhead costs and are easy to implement in an existing CFD package. Although the discretized Stokes equation leads to a symmetric coefficient matrix, we use a Krylov sub-space method suitable for non-symmetric matrices, because we also apply the resulting method to the discrete Navier–Stokes equations, where a non-symmetric coefficient matrix occurs. For an overview of Krylov methods we refer to References [23–26].

The discretized three-dimensional incompressible Stokes equation is described by the following linear system of equations:

$$\begin{pmatrix} \mathbf{Q}_1 & 0 & 0 & \mathbf{G}_1 \\ 0 & \mathbf{Q}_2 & 0 & \mathbf{G}_2 \\ 0 & 0 & \mathbf{Q}_3 & \mathbf{G}_3 \\ \mathbf{G}_1^T & \mathbf{G}_2^T & \mathbf{G}_3^T & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ p \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \tag{1}$$

where u_i are the vectors of the velocities in the i -direction and the vector p contains the pressure unknowns. In the remainder of this section, this system is abbreviated as $\mathbf{A}x = b$.

The diagonal of the matrices \mathbf{Q}_i is denoted by \mathbf{D}_i , and $\mathbf{R} = -\sum_{i=1}^3 \mathbf{G}_i^T \mathbf{D}_i^{-1} \mathbf{G}_i$. The SIMPLE method as proposed by Patankar [1] is given by the following algorithm:

SIMPLE algorithm

1. Choose an initial estimate p^* .
2. Solve $\mathbf{Q}_i u_i^* = b_i - \mathbf{G}_i p^*$.
3. Solve $\mathbf{R} \delta p = b_4 - \sum_{i=1}^3 \mathbf{G}_i^T u_i^*$.
4. Compute $u_i = u_i^* - \mathbf{D}_i^{-1} \mathbf{G}_i \delta p$ and $p := p^* + \delta p$.
5. If not converged, take $p = p^*$ and go to step 2.

The solutions of the systems given in steps 2 and 3 are obtained by a small number of iterations with a block Gauss–Seidel method (TDMA solver [1,7]).

The SIMPLE method can also be seen as a distributive iterative method [20]. Instead of solving the system $\mathbf{A}x = b$, the system $\mathbf{A}\mathbf{B}y = b$, $x = \mathbf{B}y$ will be solved. Choosing \mathbf{B} and \mathbf{M} as

$$\mathbf{B} = \begin{pmatrix} \mathbf{I} & 0 & 0 & -\mathbf{D}_1^{-1} \mathbf{G}_1 \\ 0 & \mathbf{I} & 0 & -\mathbf{D}_2^{-1} \mathbf{G}_2 \\ 0 & 0 & \mathbf{I} & -\mathbf{D}_3^{-1} \mathbf{G}_3 \\ 0 & 0 & 0 & \mathbf{I} \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{Q}_1 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_2 & 0 & 0 \\ 0 & 0 & \mathbf{Q}_3 & 0 \\ \mathbf{G}_1^T & \mathbf{G}_2^T & \mathbf{G}_3^T & \mathbf{R} \end{pmatrix} \tag{2}$$

and using the splitting $\mathbf{AB} = \mathbf{M} - \mathbf{N}$, the following iteration is obtained (SIMPLE method):

$$x^{k+1} = x^k + \mathbf{BM}^{-1}(b - \mathbf{Ax}^k), \quad k = 1, 2, \dots, \text{niter}$$

Below, a Krylov acceleration of the SIMPLE method is derived. Many Krylov sub-space methods are known to solve non-symmetric linear systems. We choose the GCR method [27] because the method is robust, minimizes the residual, and allows a variable preconditioner [28,29]. This final property is very important, since in practice the inverse of \mathbf{M} is only computed approximately. So, the postconditioner \mathbf{BM}_k^{-1} is a different operator in every iteration.

GCR-SIMPLE algorithm

```

 $r^0 = b - \mathbf{Ax}^0$ 
for  $k = 0, 1, \dots, \text{ngcr}$ 
   $s^{k+1} = \mathbf{BM}^{-1}r^k$ 
   $v^{k+1} = \mathbf{As}^{k+1}$ 
  for  $i = 0, 1, \dots, k$ 
     $v^{k+1} = v^{k+1} - (v^{k+1}, v^i)v^i$ 
     $s^{k+1} = s^{k+1} - (v^{k+1}, v^i)s^i$ 
  end for
   $v^{k+1} = v^{k+1} / \|v^{k+1}\|_2$ 
   $s^{k+1} = s^{k+1} / \|v^{k+1}\|_2$ 
   $x^{k+1} = x^k + (r^k, v^{k+1})s^{k+1}$ 
   $r^{k+1} = r^k - (r^k, v^{k+1})v^{k+1}$ 
end for

```

Due to the modified Gram–Schmidt orthogonalization, the amount of work and memory increases when the number of iterations grows. To bound these quantities, the method is restarted after a small number of iterations. Comparing the amount of work with that of the SIMPLE method, we note that GCR-SIMPLE requires ngcr^2 vector updates and $\text{ngcr}^2/2$ inner products extra. Furthermore, an additional 2ngcr vectors should be stored in memory. When ngcr is small these costs are negligible.

In our implementation the Dirichlet boundary conditions for the velocities are incorporated in the discretized equations as follows. Suppose the condition in point P is given by $u_P = g_P$, where g_P is a given value, and c_{\max} is a large real number (of the order 10^{25}). Then c_{\max} is added to the main diagonal entry corresponding to u_P and $c_{\max}g_P$ is added to the right-hand side vector. Applying the GCR-SIMPLE method to this system leads to disappointing results: slow convergence or, in some cases, divergence of the method. Therefore, a diagonal scaling is applied to system (1) before GCR-SIMPLE is used. When \mathbf{D}_{AB} is defined by $\mathbf{D}_{AB} = \text{diag}(\mathbf{AB})$, the following adaptations should be used: $r^0 = \mathbf{D}_{AB}^{-1}(b - \mathbf{Ax}^0)$, $s^{k+1} = \mathbf{BM}_k^{-1}\mathbf{D}_{AB}r^k$, and $v^{k+1} = \mathbf{D}_{AB}^{-1}\mathbf{As}^{k+1}$. In exact arithmetic, one obtains the same iterates, however, in practice, a much better convergence is observed. The reasons for this are:

- the diagonal scaling leads to better convergence in the block Gauss–Seidel methods used in the SIMPLE method;
- the diagonal scaling leads to a better behavior with respect to rounding errors.

The GCR acceleration is also applied to the SIMPLER method. First the SIMPLER method is explained. Thereafter, the combined method GCR-SIMPLER is specified. Suppose the velocities u_i are known, then an easy calculation shows that p is a solution of the system

$$\mathbf{R}p = b_4 - \sum_{i=1}^3 \mathbf{G}_i^T \mathbf{D}_i^{-1} ((\mathbf{D}_i - \mathbf{Q}_i)u_i + b_i)$$

This idea is used in the SIMPLER method. When u_i^k is known, p^k and u_i^{k+1} are calculated as follows:

SIMPLER algorithm

1. Solve $\mathbf{R}p^k = b_4 - \sum_{i=1}^3 \mathbf{G}_i^T \mathbf{D}_i^{-1} ((\mathbf{D}_i - \mathbf{Q}_i)u_i^k + b_i)$.
2. Solve $\mathbf{Q}_i u_i^* = b_i - \mathbf{G}_i p^k$.
3. Solve $\mathbf{R}\delta p = b_4 - \sum_{i=1}^3 \mathbf{G}_i^T u_i^*$.
4. Compute $u_i^{k+1} = u_i^* - \mathbf{D}_i^{-1} \mathbf{G}_i \delta p$.

One iteration of the SIMPLER algorithm is approximately 1.3 times as expensive than one SIMPLE iteration. Steps 2–4 of both methods are comparable. This motivates us to predict p^k with step 1 of the SIMPLER method followed by a number of iterations with the GCR-SIMPLE method. This new method is called the GCR-SIMPLER method.

The SIMPLER method can also be described as a classical iterative method. Choosing \mathbf{B}_R and \mathbf{M}_R as in Equation (2), and \mathbf{B}_L and \mathbf{M}_L as

$$\mathbf{B}_L = \begin{pmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ -\mathbf{G}_1^T \mathbf{D}_1^{-1} & -\mathbf{G}_2^T \mathbf{D}_2^{-1} & -\mathbf{G}_3^T \mathbf{D}_3^{-1} & \mathbf{I} \end{pmatrix}, \quad \mathbf{M}_L = \begin{pmatrix} \mathbf{Q}_1 & 0 & 0 & \mathbf{G}_1 \\ 0 & \mathbf{Q}_2 & 0 & \mathbf{G}_2 \\ 0 & 0 & \mathbf{Q}_3 & \mathbf{G}_3 \\ 0 & 0 & 0 & \mathbf{R} \end{pmatrix} \quad (3)$$

The SIMPLER method can be given by

$$x^{k+1} = x^k + \mathbf{B}_R \mathbf{M}_R^{-1} \mathbf{B}_L^{-1} \mathbf{T} \mathbf{B}_R^{-1} \mathbf{M}_L^{-1} \mathbf{B}_L (b - \mathbf{A}x^k)$$

where \mathbf{T} is the block diagonal part of the matrix $\mathbf{M}_L + \mathbf{M}_R - \mathbf{A}$. Note that the SIMPLER method is closely related to the symmetric block Gauss–Seidel method.

In the following section, GCR-SIMPLE(R) is generalized to the incompressible Navier–Stokes equations. Thereafter, these equations are combined with a turbulent combustion model in order to predict flows in industrial furnaces.

4. THE GCR-SIMPLE(R) METHOD APPLIED TO NAVIER–STOKES

The discretization of the Navier–Stokes equations gives a non-linear system due to the convection terms. The discretization equations for the velocities can be written as follows:

$$\mathbf{Q}_i(\underline{u})u_i + \mathbf{G}_i p = b_i, \quad \text{with } \underline{u} = (u_1, u_2, u_3)^T \quad (4)$$

Various methods can be chosen to linearize \mathbf{Q}_i , like the Newton–Raphson method or the Picard iteration method. We have used the Picard iteration method, where $\mathbf{Q}_i(\underline{u}^{k+1})$ is approximated by $\mathbf{Q}_i(\underline{u}^k)$. A non-symmetric linear system is obtained with the same structure as the discretized Stokes equations. Now, the GCR-SIMPLE(R) algorithm for the Navier–Stokes equations can be summarized as follows:

```

 $x^0$  guessed value
for  $k = 0, 1, 2, \dots$ , niter
  solve  $\mathbf{A}(x^k)x^{k+1} = b$  with GCR-SIMPLE(R)
end for

```

During each iteration we do not need to solve this equation until convergence because the matrix \mathbf{A} is defined using an approximation of x^k . This has the advantage that a small value of *ngcr* can be chosen, which leads to low memory requirements. The optimal value of *ngcr* can be different for each problem.

Turbulence and combustion is described by a set of coupled second-order partial differential equations (PDEs) together with the incompressible Navier–Stokes equations. Each of these equations is of the convection–diffusion type. The radiation modeling gives rise to integral equations that require a different solution technique other than the PDEs. In order to solve the full set of coupled equations for flow, turbulence, combustion, and radiation, first the Picard iteration method is used to linearize all equations. We then proceed as follows:

1. solve u_1, u_2, u_3 , and p with GCR-SIMPLE(R).
2. Solve the turbulent quantities, temperature, and species concentrations using the block Gauss–Seidel method (TDMA).
3. Solve for radiative heat transfer using the discrete transfer method.
4. Repeat this procedure until a converged solution is obtained.

5. A NUMERICAL COMPARISON OF THE SOLUTION METHODS

In this section attention will be given to the application of the SIMPLE(R) and the GCR-SIMPLE(R) methods. We will first investigate the properties of these methods for a two-dimensional Navier–Stokes flow between two flat plates. To compare the efficiency for more realistic test problems, the IFRF furnace and the Ford Nashville float glass furnace are used.

At the outlet we distinguish two types of boundary conditions: normal velocity given or pressure given. The other boundary conditions remain the same for all methods. In the measurements the following quantities are used:

- *CPUtime*: execution time of a used method measured in seconds on an HP-735 in Section 5.1 and on an HP-J210 in Section 5.2 and 5.3.
- *residu*: absolute sum of residuals for a given variable.
- *niter*: number of iterations.

5.1. Flow between two flat plates

In this section we present some results obtained when applying the SIMPLE(R) and the GCR-SIMPLE(R) method to the flow between two flat plates with distance $D = 10$ cm and length $L = 500$ cm. For this problem an equidistant grid will be used.

To apply the SIMPLER and the GCR-SIMPLER methods we first define some default values of parameters used in these methods. For both methods the termination criterion is: stop when the sum of the absolute residuals of each variable is less than or equal to 10^{-6} . The relaxation factor for the pressure is always 1. The SIMPLER method will be used with relaxation factors equal to 0.8 for velocities u_1 and u_2 . For the GCR-SIMPLER method, ngr is taken equal to 3 and the relaxation factors for velocities u_1 and u_2 are equal to 1. The default TDMA solver is PLANE TDMA.

In order to check the dependence of the methods on the outlet boundary conditions, two different boundary conditions are used. These methods have been applied using two TDMA solvers. Several grids are used to check the grid dependence of the methods. Also, the dependence on the relaxation factors is checked.

5.1.1. Outlet boundary conditions dependence. In this section, attention will be focused on the dependence of the methods on the outlet boundary conditions. The results for the two different boundary conditions are given in Table I. The SIMPLER method using the velocity given condition gives a faster convergence than when the pressure given condition is used. Contrary to this, GCR-SIMPLER converges faster when the pressure is given at the outlet. For this reason, the SIMPLE(R) method is used with the velocity given and the GCR-SIMPLE(R) method is applied with the pressure given. Note that the pressure given at the outlet is physically correct. In the SIMPLER method the non-physical outlet boundary condition (velocity given) is chosen, because this leads to fast convergence, and for this problem the results are only slightly disturbed in the vicinity of the outlet. For other problems (e.g. with more outlets) it is hard to find the correct outlet velocity.

5.1.2. TDMA solvers dependence. Two TDMA solvers [1] will be used within the SIMPLER and the GCR-SIMPLER methods. Table II shows the influence of the choice of the TDMA solvers. For this problem, the LINE TDMA is more efficient than the PLANE TDMA solver. In general, the PLANE TDMA solver is more robust; therefore, this method is used in the remainder of this paper.

Table I. Results for two different outlet boundary conditions.

Boundary condition	SIMPLER		GCR-SIMPLER	
	niter	CPU time	niter	CPU time
Velocity given	139	16.9	966	204.5
Pressure given	218	25.6	33	9.9

Grid 40×20 .

5.1.3. Relaxation factors dependence. The optimal value of the relaxation factors is different for each problem. We only give attention to the relaxation factors of the velocities. Notice that relaxation factors for u_1 and u_2 are equal. From Table III it appears that SIMPLER does not converge when the relaxation factor is equal to 1, whereas GCR-SIMPLER is very efficient for this choice. These results motivate the default values. Note that the relaxation factors can be chosen larger for GCR-SIMPLER than for SIMPLER, which leads to less iterations and CPU time. Furthermore, GCR-SIMPLER is robust, which means that it converges for a wide range of relaxation factors.

5.1.4. Dependence of GCR-SIMPLER on the value of ngr. In this section we investigate the dependence of GCR-SIMPLER on the value of ngr. The results are given in Table IV. When ngr increases the number of GCR-SIMPLER iterations decreases, but every iteration becomes more expensive. On the 40×20 grid we see that the CPU time is more or less the same for all values of ngr. For the 40×40 grid there are larger differences. The choice $\text{ngr} = 14$ leads to a minimal amount of CPU time, however, many vectors should be stored in memory. Therefore, the value $\text{ngr} = 3$ is a good compromise. When convergence problems occur for the GCR-SIMPLER method it helps when the value of ngr is increased.

5.1.5. Grid size dependence. In Table V the results are given for various grid sizes. Both methods need more iterations when the grid size increases. For a small grid size, the CPU times are comparable, whereas for a large grid size, GCR-SIMPLER needs less CPU time than the SIMPLER method. For the 20×20 grid, the aspect ratio is equal to 50.

Table II. Results using LINE TDMA and PLANE TDMA solvers.

Method	SIMPLER		GCR-SIMPLER	
	niter	CPU time	niter	CPU time
LINE TDMA	78	7.4	33	7
PLANE TDMA	139	16.9	33	9.9

Grid 40×20 .

Table III. Results for various relaxation factors.

Relaxation factor	SIMPLER		GCR-SIMPLER	
	niter	CPU time	niter	CPU time
1.0	no convergence		33	9.9
0.9	80	10.3	78	23.3
0.8	139	16.9	130	33.9
0.7	205	24.0	162	42.0
0.6	281	32.3	220	56.4

Grid 40×20 .

Table IV. Results of the GCR-SIMPLER method for various values of *ngcr*.

<i>ngcr</i>	Grid 40×20		Grid 40×20	
	niter	CPU time	niter	CPU time
2	43	10.3	96	39.9
3	33	9.9	67	35.6
4	30	10.4	59	37.4
6	21	9.8	38	33.4
8	17	9.9	31	35
14	11	10.6	14	27.5

5.2. The IFRF furnace

In this section attention will be given to the application of the SIMPLE(R) and the GCR-SIMPLE(R) methods to the IFRF furnace as given in Figure 1 (for more details see Reference [2]). In the model for this furnace, the combustion of natural gas is described by the conserved scalar approach to high temperature, non-premixed combustion, and the chemistry is described with a one-step global reaction. In order to achieve a fair comparison of the SIMPLE(R) and the GCR-SIMPLE(R), the same relaxation factors have been used for each variable. Notice that we can increase the value of the relaxation factors when the GCR-SIMPLE(R) method is used, which makes the method more efficient. The same convergence

Table V. Results for various grid sizes

Grid size	SIMPLER		GCR-SIMPLER	
	niter	CPU time	niter	CPU time
20×20	61	5.2	29	5.9
40×20	139	16.9	33	9.9
80×20	303	68.5	80	40.2

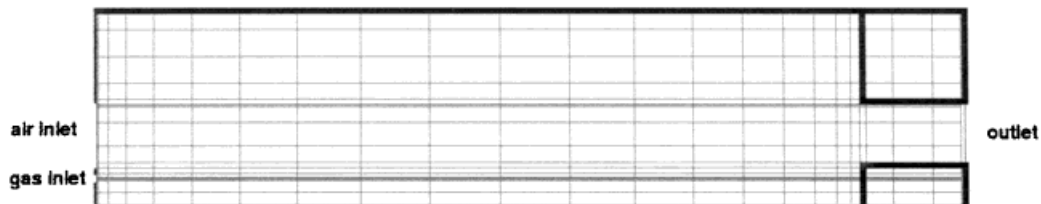
Figure 1. The symmetry plane of the furnace. Grid $24 \times 20 \times 16$.

Table VI. Results for the IFRF furnace.

Method	niter	CPU time (h)
SIMPLE	2047	4.8
SIMPLER	2415	6.9
GCR-SIMPLE	623	2.4
GCR-SIMPLER	578	2.0

Grid $24 \times 20 \times 16$.



Figure 2. The symmetry plane of the furnace. Grid $42 \times 37 \times 27$.

criterion is used for each method. The SPACE TDMA method has been used in all methods. The GCR-SIMPLE(R) method is used with $n_{gr} = 3$. In addition to this, the SIMPLE(R) method uses a given velocity at the outlet, whereas in GCR-SIMPLE(R) the pressure is prescribed.

5.2.1. Coarse grid ($24 \times 20 \times 16$). In this problem, the methods are terminated when the absolute sum of residuals of each variable is less than or equal to 10^{-3} . The finite volume grid consists of $24 \times 20 \times 16 = 7680$ points. The grid at the symmetry plane of the furnace is shown in Figure 1. The results are given in Table VI. For this problem, SIMPLE needs less iterations than SIMPLER. However, when we increase the values of the relaxation factors we obtain a better convergence for the SIMPLER method than for the SIMPLE method. The Krylov accelerated methods (GCR-SIMPLE(R)) are much more efficient. For GCR-SIMPLE, the gain is a factor of 2 in CPU time, whereas for GCR-SIMPLER the gain is a factor of 3.5.

5.2.2. Fine grid ($42 \times 37 \times 27$). We consider now a stop criterion such that the absolute sum of residuals of each variable is less than or equal to 10^{-4} . The finite volume grid consists of $42 \times 37 \times 27 = 41\,958$ points. The grid at the symmetry plane of the furnace is shown in Figure 2. High aspect ratios occur at the gas inlet ($\Delta x = 187$ mm, $\Delta y = 3$ mm, so $\Delta x/\Delta y = 62$). The convergence results are given in Table VII. Again the CPU time of GCR-SIMPLER is a factor of 3 less than that of SIMPLER. In Figure 3 the convergence behavior of the SIMPLER method is given. In this simulation, the reduction of the residual is very slow. In Figure 4 the GCR-SIMPLER residuals are plotted. The horizontal scaling of both figures is different. For the GCR-SIMPLER method, the pressure residual is between the other residuals. In these figures no convergence rates for the turbulence variables, temperatures, and species

Table VII. Results for the IFRF furnace.

Method	niter	CPU time (h)
SIMPLER	11 390	80.4
GCR-SIMPLER	3124	26.7

Grid $42 \times 37 \times 27$.

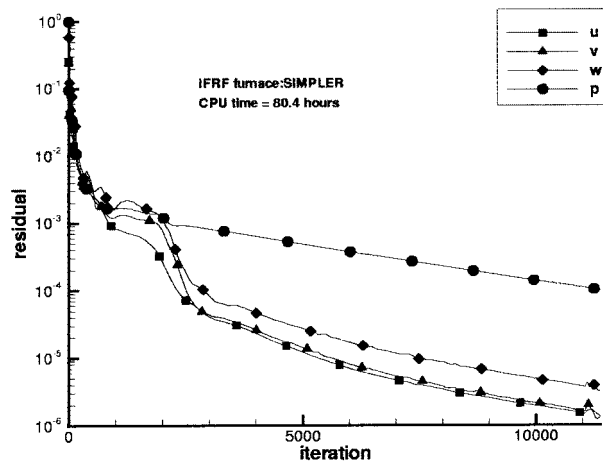


Figure 3. The absolute sum of the residuals for each variable using the SIMPLER method.

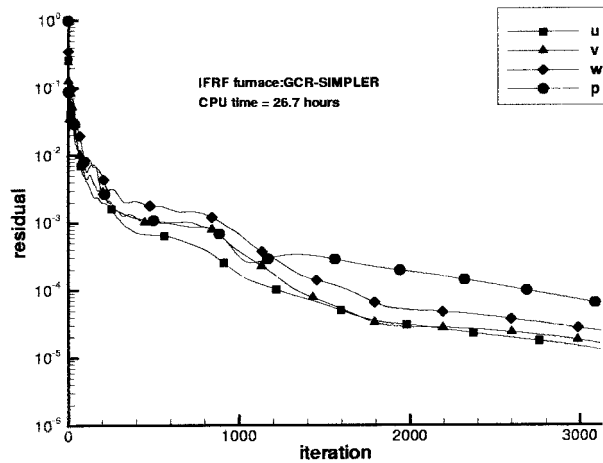


Figure 4. The absolute sum of the residuals for each variable using the GCR-SIMPLER method.

concentrations are given. The reason is that all these variables converge faster than the pressure variable for both methods. After convergence the computed velocities, temperatures, etc., are the same for both methods. Temperature contours at the symmetry plane are given in Plate 1. For a thorough validation of the computed results with measurements we refer to Reference [3].

5.3. The Ford Nashville furnace

In this section the SIMPLER and the GCR-SIMPLER methods are used to simulate the combustion chamber of the Ford furnace [30]. The combustion model is the same as the one used for the IFRF furnace. The geometry of the Ford furnace is sketched in Figure 5. The internal length \times width \times maximum height of the combustion chamber are $34.7 \times 10.1 \times 2.3$ m³. The same convergence criterion is used for each method. In this problem the iteration process is stopped when the absolute sum of the residuals of each variable is less than or equal to 10^{-4} . The finite volume grid consists of $130 \times 40 \times 40 = 208\,000$ points. The same relaxation factors are used for both methods. In the SIMPLER and the GCR-SIMPLER methods the same SPACE TDMA solver is used. The first simulation has been done using the GCR-SIMPLER method. The results are niter = 3390, CPUtime \approx 3.3 days. Using the SIMPLER method, the simulation has been stopped after 7.5 days because the maximum number of iterations has been reached. We see again a large decrease in CPU time when the Krylov acceleration is used. The temperature contours in a plane just above the glass surface are shown in Plate 2. A comparison of the computed and measured quantities for this furnace is given in Reference [30].

5.4. Memory storage

Using the GCR-SIMPLER method instead of the SIMPLER method leads to more memory storage. In Table VIII, the memory requirements are given for various problems. For a three-dimensional problem, the increase is approximately 50%. When ngrc is increased, the CPU time may decrease but the memory requirements increase.

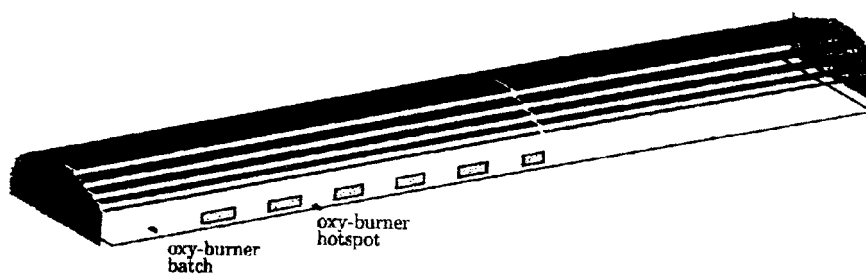


Figure 5. Geometry of the Ford float glass furnace.

Table VIII. Memory requirements for various problems measured in Mb.

Problem	SIMPLER	GCR-SIMPLER (ngcr = 3)
Flat plates (120×120)	31	39
IFRF furnace ($42 \times 37 \times 27$)	52	78
Ford furnace ($130 \times 40 \times 40$)	202	333

6. CONCLUSIONS

An efficient method to simulate glass-melting furnaces is considered. In this method, the incompressible Navier–Stokes equations are used. SIMPLE-type methods are very popular to solve the discretized incompressible Navier–Stokes equations. In this paper, SIMPLE and SIMPLER are rewritten as classical iteration methods for linear systems. Two Krylov accelerated methods are proposed: GCR-SIMPLE and GCR-SIMPLER.

The dependence of these methods on grid-size, outlet boundary condition, etc., is investigated by numerical experiments. The insights obtained from this analysis are used to propose a number of default parameters (ngcr = 3, TDMA solver, outlet boundary condition, etc.) for the GCR-SIMPLER(R) methods. Finally, the efficiency of the methods is compared using a simulation of two industrial furnaces. For these simulations, where the grid has high aspect ratios, the GCR-SIMPLER method appears to be three times as fast as the SIMPLER method. Additionally, larger relaxation factors can be used for the GCR-SIMPLER methods, which leads to a still higher efficiency. After convergence, the quality of the computed results (velocity, pressure, turbulence quantities, etc.) is comparable. The GCR-SIMPLER method requires more memory than the SIMPLER method.

REFERENCES

1. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
2. Boerstael GP. Modelling of gas-fired furnaces. PhD thesis, Delft University of Technology, Netherlands, 1997.
3. Boerstael GP, Lankhorst AM III. Modelling combustion chambers of glass melting furnaces: a validation study (invited paper), ICG703-B04-002. In *The 18th International Congress on Glass, 5–10 July, San Francisco, CA*, Choudhary MK, Drummond CH III, Huff NT (eds). The American Ceramic Society: Westerville, 1998.
4. Cahouet J, Chabard JP. Some fast 3D finite elements solvers for the generalized Stokes problem. *International Journal for Numerical Methods in Fluids* 1988; **8**: 869–895.
5. Temam R. *Navier–Stokes Equations. Theory and Numerical Analysis* (3rd edition). North-Holland: Amsterdam, 1984.
6. Vincet C, Boyer R. A preconditioned conjugate gradient Uzawa-type method for the solution of the Stokes problem by mixed Q1-P0 stabilized finite elements. *International Journal for Numerical Methods in Fluids* 1992; **14**: 289–298.
7. Ferziger JH, Peric M. *Computational Methods for Fluids Dynamics* (2nd edition). Springer: Berlin, 1999.
8. Bercovier M. Perturbation of mixed variational problem, applications to mixed finite elements methods. *RAIRO Analysis Numérique* 1978; **12**: 211–236.
9. van Kan J. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal of Scientific and Statistical Computing* 1986; **7**: 870–891.
10. Barten IE. Comparison of SIMPLE- and PISO-type algorithms for transient flows. *International Journal for Numerical Methods in Fluids* 1998; **26**: 459–483.
11. Quarteroni A, Valli A. *Numerical Approximation of Partial Differential Equations*. Springer Series in Computational Mathematics 23. Springer: Berlin, 1994.

12. Patankar SV, Spalding DB. A calculation procedure for heat and mass transfer in three-dimensional parabolic flows. *International Journal for Heat and Mass Transfer* 1972; **15**: 1787–1806.
13. Wittum G. Multi-grid methods for Stokes and Navier–Stokes equations with transforming smoothers: algorithms and numerical results. *Numerische Mathematik* 1989; **54**: 543–563.
14. van Doormaal JP, Raithby GD. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numerical Heat Transfer* 1984; **7**: 147–163.
15. Braess D, Sarazin D. An efficient smoother for the Stokes problem. *Applied Numerical Mathematics* 1997; **23**: 3–19.
16. Gjesdal T, Lossius MEH. Comparison of pressure correction smoothers for multi-grid solution of incompressible flow. *International Journal for Numerical Methods in Fluids* 1997; **25**: 393–405.
17. Lien FS, Leschziner MA. Multigrid acceleration for recirculating laminar and turbulent flows computed with a non-orthogonal, collocated finite-volume scheme. *Computer Methods in Applied Mechanical Engineering* 1994; **118**: 351–371.
18. Moukalled F, Acharya S. A local adaptive grid procedure for incompressible flows with multigridding and equidistribution concepts. *International Journal for Numerical Methods in Fluids* 1991; **13**: 1085–1111.
19. Shaw GJ, Sivaloganathan S. On the smoothing of the SIMPLE pressure correction algorithm. *International Journal for Numerical Methods in Fluids* 1988; **8**: 441–462.
20. Wesseling P. *An Introduction to Multigrid Methods*. Wiley: Chichester, 1992.
21. Zang Y, Street RL, Koseff JR. A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. *Journal of Computational Physics* 1994; **114**: 18–33.
22. Saghir A. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. FSP-RPT-980069, TNO Institute of Applied Physics, Delft, 1998.
23. Axelsson O. *Iterative Solution Methods*. Cambridge University Press: Cambridge, 1994.
24. Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, Van der Vorst H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM: Philadelphia, PA, 1994.
25. Bruaset AM. *A Survey of Preconditioned Iterative Methods*. Pitman Research Notes in Mathematics Series 328. Longman Scientific and Technical: Harlow, 1995.
26. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing: Boston, MA, 1996.
27. Eisenstat SC, Elman HC, Schultz MH. Variational iterative methods for non-symmetric systems of linear equations. *SIAM Journal of Numerical Analysis* 1983; **20**: 345–357.
28. van der Vorst HA, Vuik C. GMRESR: a family of nested GMRES methods. *Numerical Linear Algebra with Applications* 1994; **1**: 369–386.
29. Vuik C. New insights in GMRES-like methods with variable preconditioners. *Journal of Computational and Applied Mathematics* 1995; **61**: 189–204.
30. Lankhorst AM, Boerstoel GP, Muysenberg HPH, Koram KK. Complete simulation of the glass tank and combustion chamber of the former Ford Nashville float furnace including melter, refiner and working end. In *Fourth International Seminar on Mathematical Simulation in the Glass Melting*. Glass Service Limited: Vsetin, Czech Republic, 1997; 77–86.