Development of a multigrid finite difference solver for benchmark permeability analysis

Article · January 2010

CITATIONS
0

READS
213

5 authors, including:

Remko Akkerman
University of Twente
271 PUBLICATIONS **3,463** CITATIONS
SEE PROFILE

Andre de Boer
University of Twente
191 PUBLICATIONS **845** CITATIONS
SEE PROFILE

DEVELOPMENT OF A MULTIGRID FINITE DIFFERENCE SOLVER FOR BENCHMARK PERMEABILITY ANALYSIS

Richard Loendersloot [1], Wouter J.B. Grouve [2], Remko Akkerman [2], and André de Boer [2]

[1] *University of Twente, Faculty of Engineering Technology, Structural Dynamics and Acoustics, P.O box 217,7500AE Enschede, The Netherlands:*
r.loendersloot@ctw.utwente.nl

[2] *University of Twente, Faculty of Engineering Technology, Production Technology, P.O box 217,7500AE Enschede, The Netherlands.*

ABSTRACT: A finite difference solver, dedicated to flow around fibre architectures is currently being developed. The complexity of the internal geometry of textile reinforcements results in extreme computation times, or inaccurate solutions. A compromise between the two is found by implementing a multigrid algorithm and analytical solutions at the coarsest level of discretisation. Hence, the computational load of the solver is drastically reduced.
This paper discusses the main features of the 3D multigrid algorithm implemented as well as the implementation of the analytical solution in the finite difference scheme. The first tests of the solver on the permeability benchmark lithographic reference geometry are discussed.
Several tests were performed to assess the accuracy and the reduction in calculation time. The methods prove to be both accurate and efficient. However, the code is developed in Matlab© and hence is relatively slow. A C++ code is currently under development to achieve acceptable calculation times.

KEYWORDS: Multigrid, permeability, benchmark

INTRODUCTION

A few years ago, a permeability benchmark was launched. The goal of this project is to set a standard for permeability measurements and to recognise and validate numerical models to predict the permeability based on the fabric's properties. To this end, a reference geometry was designed [1]. Development of a numerical solver does not necessarily imply the design and implementation of a new code, but merely implies the identification of the relevant geometrical characteristics of textile reinforcements that must be present in the model – and subsequently how this can be achieved.
Beyond any point of discussion is that the internal architecture of woven fabrics is complicated [2]. Generally, concessions are made to the accuracy of the geometrical description to achieve reasonable calculation times. An alternative route, using a multi level discretisation, was shown to have an interesting potential in terms of maintaining geometrical accuracy, while reducing the required computational efforts [3]. The

benefits of the algorithm were shown in by a 2D solver. However, a 3D solver is essential to assess the real potential of the multigrid approach for complex textile architectures and to validate the method on the reference geometry.

3D MULTIGRID ALGORITHM AND IMPLEMENTATION

The basic flow equations that are solved are the Stokes equation describing a viscous flow of a incompressible, Newtonian flow and the continuity equation:

$$\underline{\nabla} p - \mu \underline{\underline{\nabla}}^2 \cdot \underline{u} = \underline{0} \tag{1}$$

$$\underline{\nabla} \cdot \underline{u} = 0 \tag{2}$$

With p the pressure, μ the dynamic viscosity, $\underline{u}$ the fluid velocity vector, $\underline{\nabla}$ and $\underline{\underline{\nabla}}^2$ the gradient and Laplace operator respectively. This set of equations is solved iteratively employing a finite difference technique on a staggered grid as shown in Fig. 1. The iterative solver employs a Jacobi relaxation scheme. The resulting error – or residual – compared to the, yet unknown, exact solution is used to estimate a correction for the current solution. This procedure is repeated until the numerical solution satisfies (1) and (2) within the desired limits of accuracy.

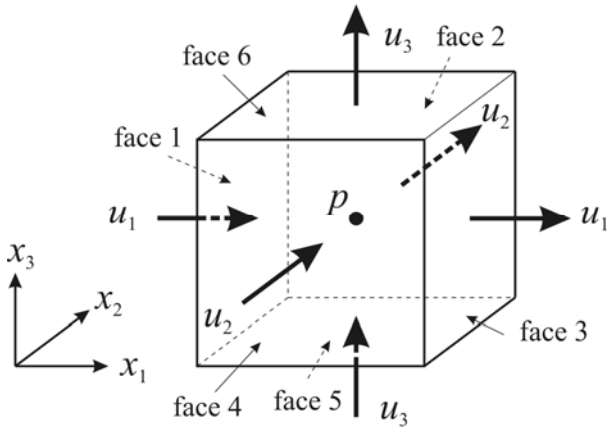

Fig. 1: *Staggered grid cell for a 3D flow domain. The velocities are defined at the cell faces, whereas the pressures are defined in the cell centre.*

The multigrid algorithm guarantees a high convergences rate combined with a high accuracy, by solving the discretised versions of (1) and (2) on grids with a different number of elements, while the solution is transferred between the different grids [4] (coarse to fine: interpolation, fine to coarse: restriction). The number of relaxations is limited rather than that the relaxation is continued till convergence is reached. Both a "Full MultiGrid" (FMG) and "V-cycle MultiGrid" (VMG) schemes were implemented, see Fig. 2. The corrections on the variables (the velocity $\underline{u}$ and pressure p) obey the same differential equation as the variables. Hence, it is possible to improve the corrections in the same way as the variables. This is visualised in Fig. 2. The circles with a solid core represent the places where the iterations are performed on the variables, whereas the open circles indicate the positions where the corrections are improved. Double lines correspond to an interpolation of the current solution to a finer grid, single lines to a restriction or interpolation of the correction. Each cycle can be repeated multiple times before continuing to the next level in FMG.

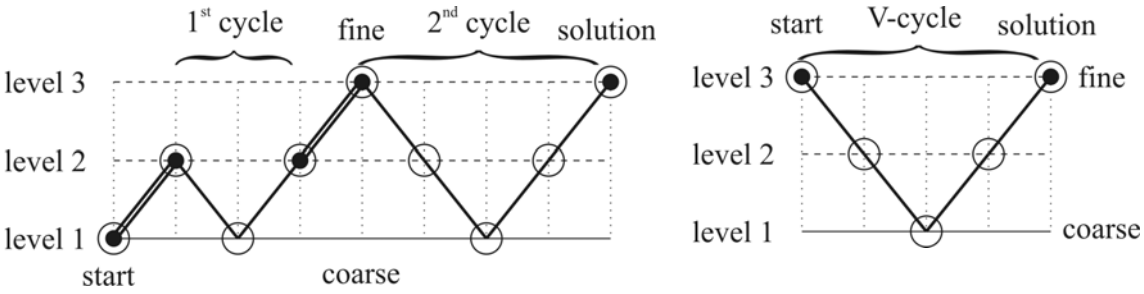

Fig. 2: *Full MultiGrid (FMG) and V-cycle MultiGrid (VMG) schemes. The circles with a solid core refer to the variables, whereas open circles refer to the corrections. The double lines indicate an interpolation of variables (the current solution), whereas single lines indicate restriction or interpolation of the correction.*

The number of degrees of freedom grows rapidly by extending a flow domain from 2D to 3D. The computational work is directly related to the number of degrees of freedom [4] and therefore it pays off to implement smart methods to limit the number of degrees of freedom or to increase the solver's efficiency.
The solver's efficiency is dramatically increased by employing the Multigrid algorithm, as was already shown for the 2D case [3]. However, the coarsest grid required two elements over the height (in x_3 direction) of the narrowest gaps. This can be reduced by inserting the analytical solution for flow between plates; The Hagen-Poisseuille equation, defining the velocity $u_1(x_3)$ in x_1-direction (see Fig. 3), reads:

$$u_1(x_3) = -\frac{1}{2\mu}\frac{\partial P}{\partial x_3}\left(\frac{h^2}{4} - x_3^2\right) \tag{3}$$

with μ the dynamic viscosity, P the pressure and h the domain height. The origin is defined in the centre of the domain (boundaries at $x_3=\pm\frac{1}{2}h$).
A test case was evaluated to check the code. A rectangular domain with a pressure gradient in the x_1 direction (see Fig. 1), symmetry boundary conditions on the domain boundaries in the x_2 direction (Fig. 1: faces 2 and 4) and no-slip boundary conditions on the domain boundaries in the x_3 direction (Fig. 1: faces 5 and 6) were defined. The dimensions of the domain, number of element, number of multigrid levels and the solver settings are listed in Table 1. Three variants were calculated, all with an equal number of elements at the finest grid:

1. Standard Gauss-Seidel relaxation on a 32×32×2 grid.
2. 3 level FMG scheme, starting at an 8×8×2 grid.
3. 4 level FMG scheme, starting at an 4×4×1 grid using the analytical solution in the Gauss-Seidel relaxation at the coarsest grid.

The convergence of the solution, based on the norm of the residuals, is shown in Fig. 3. The test cases also converged to the analytical solution presented in equation 3. The convergence rate of the standard Jacobi relaxation quickly levels out: the convergence stalls. The convergence of the two FMG schemes remains high. However, there is still a significant difference between the 3 and 4 level FMG, which is attributed to the ability to use a coarser grid.

Table 1: *Geometry and settings for the three test cases (2.5GHz CPU).*

Case	Level	Cells	L_{edge} [mm]	# Cycles	# Iterations	Runtime [s]
1	1	32×32×8	$0.3125 \cdot 10^{-4}$	n.a.	10^4 (max 10^4)	36745
2	1	8×8×2	$1.25 \cdot 10^{-4}$	n.s.	10^4 (max 10^4)	
	2	16×16×4	$0.625 \cdot 10^{-4}$	15	50	3419
	3	32×32×8	$0.3125 \cdot 10^{-4}$	15	25	
3	1	4×4×1	$2.5 \cdot 10^{-4}$	n.s.	4504 (max 10^4)	
	2	8×8×2	$1.25 \cdot 10^{-4}$	15	50	1829
	3	16×16×4	$0.625 \cdot 10^{-4}$	15	25	
	4	32×32×8	$0.3125 \cdot 10^{-4}$	15	15	

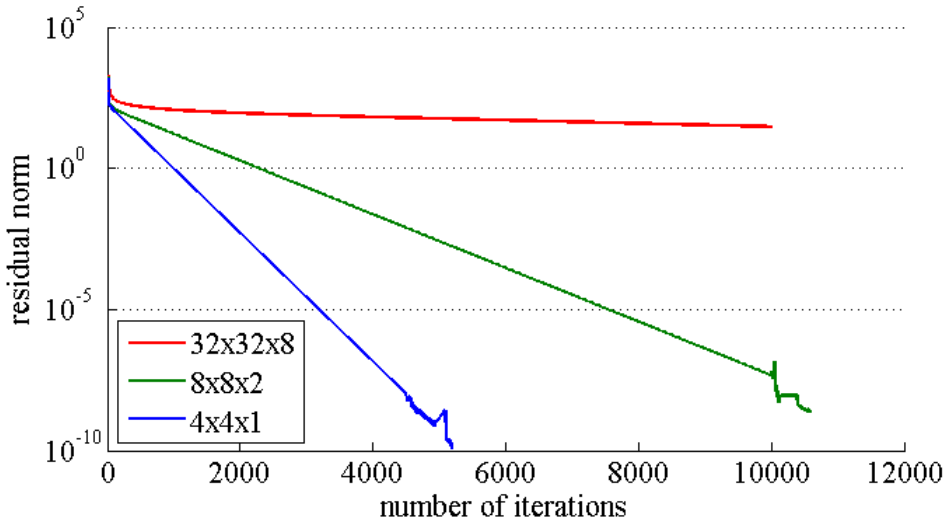

Fig. 3: *Convergence of the norm of the residual as a function of the number of iterations for the three test cases, using quadratic interpolation functions.*

BENCHMARK GEOMETRY

A reference geometry was defined by Morren *et al.* [2] to be able to validate experimental results. The reference geometry can also be employed to test numerical codes. The code developed here, is therefore first tested on the reference geometry, prior to the implementation of fabric geometries in the flow domain. The dimensions of the reference geometry are listed in Table 1.

Table 1: *Dimensions of the reference geometry.*

L_{total} [mm]	W_{total} [mm]	H_{total} [mm]	H_{layer} [mm]	d [mm]
5.5	2.5	3.0*	0.5	0.5

* A value of 2.5mm was used, to obtained cubic cells. The solver can not yet deal with non-cubic cells.

The height of the reference cell was adapted slightly, in order to obtain cubic cells with, at the coarsest level, cell edges of 0.5mm. Non-cubic cells require the use of the Jacobian of the (discretised) derivatives. This is not yet implemented. The first results are shown in Fig. 4. A 4 level V-cycle scheme is employed. Each cycle is repeated 5 times. The convergence rate during each cycle is significantly higher compared to the single level Jacobi iteration. However, the solution suffers from the interpolation error introduced due to the linear interpolation that is still employed. The test cases had

revealed a quadratic interpolation must be employed. The implementation of quadratic interpolation functions is currently ongoing. The finest grid counts 88×40×40 cells.

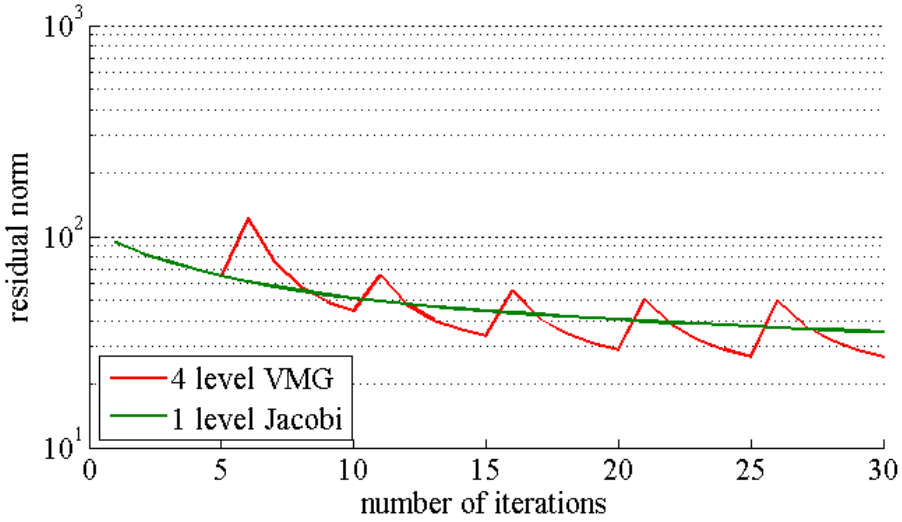

Fig. 4: *First results of the multigrid solver applied to the reference geometry, using a V-cycle and linear interpolation functions. The green line corresponds to the single level Jacobi relaxation on a grid of 88×40×40 cells.*

CONCLUSIONS AND FUTURE WORK

The 3D multigrid solver shows that both accuracy and computational efficiency can be combined. Convergence is reached approximately 10 times faster compared to the standard single level Jacobi relaxation. Using the analytical solution to enable the use of a single cell over the height of a flow channel further reduces the calculation time nearly 2 times. The first test on the benchmark geometry shows that the code is convergent, but that quadratic interpolation functions between the discretisation levels must be employed. These are currently being implemented.

REFERENCES

1 G. Morren, M. Bottiglieri, S. Bossuyt H. Sol, D. Lecompte, B. Verleye, S.V. Lomov, *Composites Part A*, 40:244-250, 2009

2. S.V. Lomov, G. Huysmans, I. Verpoest, *Textile Research Journal*, 71(6):534-543

3. R. Loendersloot, W.J.B. Grouve, R. Akkerman, S. van den Berg, Proceedings of FPCM-9, Montréal, Canada, 8 pages, 2008

4. C.H. Venner, A.A. Lubrecht, "Multilevel Methods in Lubrication", Elsevier, ISBN:0 444 50503 2, 2000

5. B. Verleye, R. Croce, M. Griebel, M. Klitz, S.V. Lomov, G. Morren, H. Sol, I. Verpoest, D. Roose, Proceedings of FPCM-9, Montréal, Canada, 8 pages, 2008