Experiences with an Algebraic Multigrid Method for a 3D Biological Respiration-Diffusion Model

Abstract — **We present our experiences with the use of an algebraic multigrid method for solving the set of equations obtained after finite element discretization of a biological respiration-diffusion model. Many of the issues encountered in this study are very similar to issues that one is faced with in the simulation of electromagnetic systems: finite element discretization of two- and three-dimensional models, non-linearities, time integration methods, solution of large linear systems. We show that the application of advanced iterative solvers such as algebraic multigrid allows large problems to be solved in a very efficient way.**

I. Introduction

The research presented here is a collaboration between the Scientific Computing Group of the Department of Computer Science and the Laboratory of Postharvest Technology both from the Catholic University of Leuven in Belgium. At the Laboratory of Postharvest Technology mathematical models for moisture, gas and heat transport inside fruit are being studied and developed [8, 9]. The ultimate goal of this research is to develop good postharvest storage techniques, in order to extend the storage life of harvested fruit. One of the models that is currently being investigated is a respiration-diffusion model for the oxygen consumption and carbon dioxide production inside the 'Conference' pear, which is particular variety of pear that is grown widely in Belgium. This model is used to obtain a better understanding of the respiratory activity of fruit and the circumstances that influence the onset of certain fruit disorders such as the 'brown and hollow' or 'core breakdown' disorders (see figure 1). Such insight is potentially of a great economical importance.

To extend the postharvest storage life 'Conference' pears are stored under controlled atmospheric (CA) storage conditions, for example, in large cooling rooms. An uncontrolled atmospheric storage can change or accelerate the fruit metabolism, causing disorders that result in economical losses. The most common disorder 'brown and hollow' is characterized by a brown discoloration of the tissue and the development of cavities. This can not easily be detected from the outside, and, as such, all too often leads to an unpleasant surprise for the consumer.

In order to get some insight into the development of the 'brown and hollow' storage disorder and its dependence on the atmospheric conditions, a respiration-diffusion model was constructed. The model consists of a set of two coupled non-linear reaction diffusion equations, defined on a three-dimensional domain and completed with mixed type boundary conditions. In order to simulate that model, numerical techniques are used. Here, we present our experiences with an algebraic multigrid (AMG) method for solving the set of equations obtained after a finite element discretization of the mathematical model. Many of the issues we encountered are very similar to the ones that one is faced with in the simulation of electromagnetic devices [4, 5]. We concentrate on the use of the recent version of the AMG code developed by K. Stüben [10]. We will consider its application for solving both the steady-state problem and the time-evolution problem. For the latter, we will discuss time-discretization techniques using backward differentiation formulae (BDF) or implicit Runge-Kutta (IRK) methods. The AMG-results will be compared to the results obtained with an optimized direct method based on LU-factorization. The latter was used previously mainly for robustness reasons and ease of use.

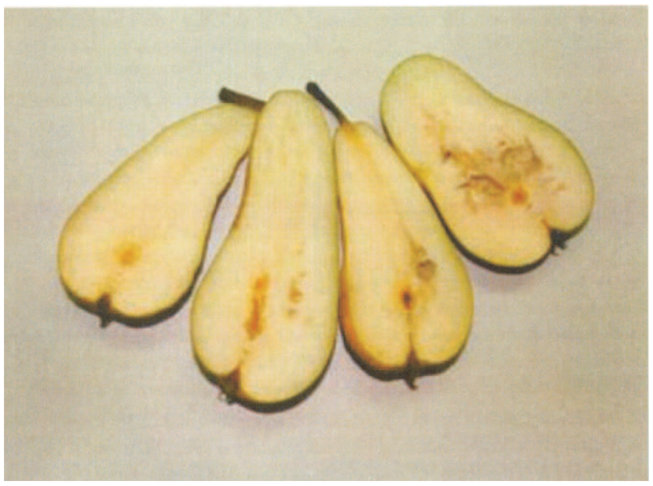
Figure 1: 'Conference' pear with, from left to right, increasing levels of 'brown and hollow' disorder (source: [8]).

II. PDE Model

The mathematical model that describes the oxygen and carbon dioxide metabolism in the pear consists of a set of two coupled reaction-diffusion equations, defined on a three-dimensional domain that represents the pear. For both oxygen and carbon dioxide a differential equation was established based on Fick's second law of diffusion describing the evolution of the gas concentration over time. Each equation consists of two terms. The first one describes the diffusive gas transport, and is characterized by a diffusion coefficient. The second term describes the oxygen consumption or carbon dioxide production. The oxygen consumption is modelled with Michaëlis-Menten kinetics including non-competitive inhibition by carbon dioxide. The carbon dioxide production is composed of a fermentative part, which is inhibited at high oxygen concentrations, and an oxidative part, almost absent at very low oxygen concentrations. The respiration-diffusion model of the Conference pear is given by the following set of two coupled non-linear partial differential equations [8]:

$$\frac{\partial C_{O_2}}{\partial t} = \nabla \cdot (D_{O_2} \nabla C_{O_2}) - V_{O_2}(C_{O_2}, C_{CO_2}), \quad (1)$$

$$\frac{\partial C_{CO_2}}{\partial t} = \nabla \cdot (D_{CO_2} \nabla C_{CO_2}) + V_{CO_2}(C_{O_2}, C_{CO_2}), \quad (2)$$

where the reaction terms are

$$V_{O_2} = \frac{V_{max} C_{O_2}}{(K_m + C_{O_2})\left(1 + \frac{C_{CO_2}}{K_{mCO_2}}\right)}, \quad (3)$$

$$V_{CO_2} = RQ_{ox} V_{O_2} + \frac{V_{mfCO_2}}{1 + \frac{C_{O_2}}{K_{mfO_2}}}, \quad (4)$$

Gas transport	D_{O_2}	$5 \cdot 10^{-9}\, \mathrm{m^2/s}$
	D_{CO_2}	$4 \cdot 10^{-8}\, \mathrm{m^2/s}$
Respiration kinetics	V_{max}	$2 \cdot 10^{-4}\, \mathrm{mol/m^3s}$
	V_{mfCO_2}	$10^{-4}\, \mathrm{mol/m^3s}$
	RQ_{OX}	0.76
	K_m	$0.05\, \mathrm{mol/m^3}$
	K_{mCO_2}	$14.51\, \mathrm{mol/m^3}$
	K_{mfO_2}	$2.83 \cdot 10^{-4}\, \mathrm{mol/m^3}$
Boundary conditions	h_{O_2}	$7 \cdot 10^{-7}\, \mathrm{m/s}$
	h_{CO_2}	$7.5 \cdot 10^{-7}\, \mathrm{m/s}$

Table 1: The constants of the respiration-diffusion model.

with C_{O_2}, C_{CO_2} the oxygen and the carbon dioxide concentration ($\mathrm{mol/m^3}$), t the time (s), D_{O_2} and D_{CO2} the gas diffusivities of O_2 and CO_2 in pear tissue ($\mathrm{m^2/s}$), V_{O_2} and V_{CO_2} the oxygen consumption and carbon dioxide production ($\mathrm{mol/m^3s}$), V_{max} and V_{mfCO_2}, respectively, the maximum aerobic oxygen consumption rate and the maximal fermentative carbon dioxide production rate ($\mathrm{mol/m^3s}$), K_m the Michaëlis-Menten constant ($\mathrm{mol/m^3}$), K_{mCO_2} the non-competitive inhibition constant of CO_2 on the O_2 consumption ($\mathrm{mol/m^3}$), K_{mfO_2} the inhibition constant of O_2 on the fermentative CO_2-production ($\mathrm{mol/m^3}$) and RQ_{ox} the respiration quotient (-). The mixed type boundary conditions are

$$D_{O_2} \frac{\partial C_{O_2}}{\partial n} = h_{O_2}(C_{O_2}^{\infty} - C_{O_2}), \tag{5}$$

$$D_{CO_2} \frac{\partial C_{CO_2}}{\partial n} = h_{CO_2}(C_{CO_2}^{\infty} - C_{CO_2}), \tag{6}$$

with h_{O_2} and h_{CO_2} the convective mass transfer coefficients of O_2 and CO_2 from the pear to the environment ($\mathrm{m/s}$), which includes the mass transfer resistance of the pear skin.

In order to obtain reliable parameter values for both the respiration and diffusion activities, extensive measurements were conducted considering the two phenomena separately. The parameters of the Michaëlis-Menten respiration kinetics were determined on intact pears, and also the effect of diffusion on the respiration parameters was determined. A novel method, based on the gas diffusion law of Fick, was used to measure the oxygen and carbon dioxide diffusivity of pear tissue and skin. Further details about the experiments and parameter measurements can be found in [8, 6, 9, 7]. The values used for the different constants of the respiration-diffusion model are listed in table 1. Some of these coefficients are temperature dependent. The values given in the table are the ones used for the simulation, as described in Section VI.

III. Discretization

Because of the complexity of the shape of the domain and the non-linearity of the respiration kinetics, there is no analytical solution for the mathematical model. Therefore, a numerical simulation procedure is used. The first part of the procedure is a geometry scan of the pear. From this scan a 3D tetrahedral mesh is generated using ANSYS. An example is given in figure 2. The mesh information is then used to discretize the PDE-model, using standard linear finite elements. The discretization results in the stiff, non-linear system of ordinary differential equations

$$C\frac{du}{dt} + Ku = f(u). \tag{7}$$

The solution vector u contains two unknowns for each node of the finite element mesh. The mass matrix C and the stiffness matrix K are large sparse matrices. To find a steady-state solution

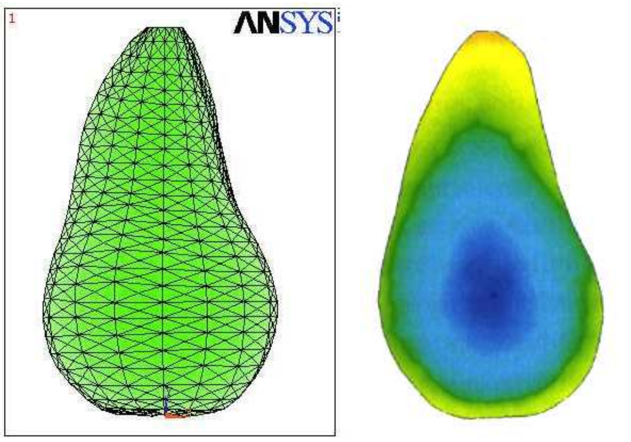

Figure 2: Finite element discretization (left) and computed O_2 concentration (right).

the system of equations

$$Ku = f(u) \tag{8}$$

has to be solved. For time-accurate simulations, the system of ordinary differential equations (7) is discretized in time using implicit time-discretization methods.

Backward differentiation formulas (BDFs) [3] use values from one or more previous time-steps to calculate a value for the current time-step. The first order BDF method is known as the implicit Euler method (BDF1) and leads to the following equations

$$C\frac{u_{n+1} - u_n}{\Delta t} + Ku_{n+1} = f(u_{n+1}). \tag{9}$$

With the seconder order BDF2-method we get the equations

$$C\frac{3u_{n+1} - 4u_n + u_{n-1}}{2\Delta t} + Ku_{n+1} = f(u_{n+1}). \tag{10}$$

These systems have the same dimensions as (8) and a very similar matrix sparsity structure.

Implicit Runge-Kutta (IRK) methods [3] use one previous value to calculate a number of so-called stage values U_i. Those are then used to find a new value at the current time-step. The update can be described by the equations

$$Cu_{n+1} = Cu_n + \Delta t \sum_{j=1}^{s} b_j(-KU_j + f(U_j)), \tag{11}$$

where the stage values U_i are calculated from the system

$$CU_i = Cu_n + \Delta t \sum_{j=1}^{s} a_{ij}(-KU_j + f(U_j)). \tag{12}$$

The latter system is s times larger than the steady-state or BDF system. The values U_i can be interpreted as approximating the solution at the points $t_{n,i} = t_n + c_i\Delta t$. An IRK method can be compactly characterized by a so-called Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}.$$

For the numerical experiments we used the popular three-stage Radau IIA method of order 5 [3], characterized by the tableau

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}.$$

Since $b_i = a_{si}$, for $s = 3$ and for all i, no extra computations are needed for determining the update (11).

IV. Newton Iteration

In this section, we briefly discuss the linearization of the nonlinear set of equations derived in the previous section. We explain how the starting values are chosen for the steady-state and for the time-dependent problem. Finally, we mention how to avoid convergence to unphysical negative solutions.

In the original procedure a standard Newton iteration was used. In order to speed up the simulation, we now apply a modified Newton iteration, that is, the Jacobian is held constant over a number of iterations. Furthermore, we replace the Newton method by an inexact or truncated Newton iteration. Instead of solving the sequence of linear systems in the modified Newton iteration exactly, only a few steps of an iterative method are performed. In our case, we typically use two or three multigrid cycles. This procedure may result in extra Newton steps, but the total computing time is usually reduced. For the implicit Runge-Kutta time-discretization, a further simplification to Newton's method is applied. All Jacobians $\partial f/\partial u(U_j)$ are replaced by the approximation $J \approx \partial f/\partial u(u_n)$, which results in the following simplified Newton step,

$$\begin{aligned}(C - \Delta t A \otimes J)\Delta U^{(k)} &= -CU^{(k)} + \Delta t(A \otimes I)F(U^{(k)}) \\ U^{(k+1)} &= U^{(k)} + \Delta U^{(k)} \qquad (13)\end{aligned}$$

with $(C - \Delta t A \otimes J)$ constant over several Newton iterations.

When computing the solutions for the coupled steady-state pear model with different temperature values and different ambient oxygen and carbon dioxide concentrations, we noticed the problem of convergence to unphysical, negative solutions, for certain parameter values. At high temperatures, for example, rapid oxygen consumption can push the iterates to negative oxygen concentrations. The mathematical model is no longer physically relevant in such a case, but a solution continues to exist (the nonlinear reactions terms can still be evaluated, but produce meaningless results). A possible remedy is to use a parameter continuation strategy. We can, for example, start with a solution at low temperature and use this solution as the starting value for a slightly higher temperature. For the time-dependent pear model, we can use the solution of the last time-step as a new starting value. It is of course also possible to use time-stepping to find steady-state solutions. If a solution can be found using only Newton iteration, however, this will generally be much more efficient. It turns out that in some cases parameter continuation is no longer sufficient. The model needs to be changed in order to force it to converge to physical solutions. For positive concentrations the reaction term remains the same, but for negative oxygen concentrations the oxygen production term is set to zero. Another approach is to perform a mathematical transformation of the form

$$\begin{aligned}C_{O_2} &= \exp(U_{O_2}) \qquad (14)\\ C_{CO_2} &= \exp(U_{CO_2}) \qquad (15)\end{aligned}$$

V. Multigrid Methods

The numerical simulation system developed at the Laboratory of Postharvest Technology is written in MATLAB and uses the highly optimized direct sparse solver provided by the MATLAB backslash operator. It is well known, however, that direct sparse solvers are often not computationally efficient and have a very high memory cost especially for 3D problems. It is therefore necessary to consider iterative methods. Such iterative methods may have to be adapted to the problem. Multigrid methods are very well suited for solving the linear systems of equations derived from diffusion problems. Unless one is dealing with a model problem, in which case the selection of the algorithmic components and parameters is well understood, the multigrid method must be made more robust by using it as a preconditioner for a Krylov method, such as the conjugate gradient (CG) algorithm. In this section, we explain the basic ideas behind the geometric multigrid method, the classical algebraic multigrid method and algebraic multigrid for systems of PDEs.

A. Geometric Multigrid

Multigrid methods are iterative methods that combine PDE discretizations on meshes of varying density, see, e.g., figure 3 for a hierarchy of meshes for a 2D pear model. High frequency error components are damped on a fine grid, whereas low frequency error components are transferred to the next coarser grid. On this next coarser grid, low frequency components appear as high frequency ones and thus the multigrid idea can be applied recursively. This recursion terminates when the cost of solving the linear system on the coarse grid becomes negligible. We describe the algorithm more formally for two grids. Extension to the general case is straightforward. Let $A^h x^h = b^h$ and $A^H x^H = b^H$ be a fine and coarse grid PDE discretization. The two-grid method uses the following steps,

- Given an approximation x^h_m for the fine grid equation, first high frequency components in the error ($e^h_m = x^h - x^h_m$) are damped by applying a few (typically one or two) steps of a stationary iterative scheme. This step is called pre-smoothing.
- The residual $r^h_m = b^h - A^h x^h_m$ is transferred to the coarser grid by the restriction operator I^H_h: $\quad r^H_m \leftarrow I^H_h r^h_m$.
- On the coarser grid the defect equation $A^H e^H_m = r^H_m$ is solved exactly.
- The coarse grid correction is transferred back to the fine grid using the interpolation operator I^h_H and added to the existing approximation: $\quad x^h_m \leftarrow x^h_m + I^h_H e^H_m$.
- As the last operation can introduce high frequency components again, a few post-smoothing steps may be performed.

If we recursively use the same iteration once to approximate the solution of the defect equation, we get the so called V-cycle multigrid iteration. If we apply two iterations, we get the W-cycle. For more information on multigrid methods we refer to [1, 11]. For a detailed analysis of geometric multigrid for the systems that arise from a BDF and IRK discretization of time dependent problems see [12].

B. Algebraic Multigrid

For difficult problems, with, e.g., complex geometries, irregular meshes, or strongly varying PDE coefficients, efficient geometric multigrid methods may be cumbersome to develop and implement. It is often not immediately clear how a sequence of coarser meshes can be constructed or which smoothers are appropriate for a given problem. Algebraic multigrid (AMG) methods offer a solution to this problem by providing the advantages of geometric multigrid techniques in a more or less black box solver. An AMG solver requires as sole input the linear system

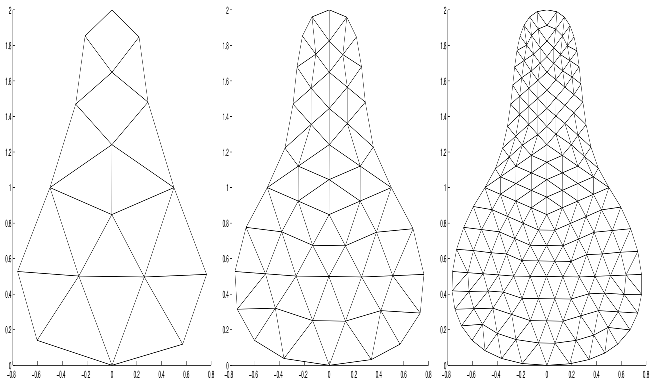

Figure 3: Grid hierarchy for a pear model.

to be solved. In the following we will describe the Brandt-Ruge-Stüben approach to AMG for real symmetric positive definite problems [11, App. A] and [10].

The AMG solution process can be divided into two phases. In a setup phase, the algorithm constructs fully automatically (i.e., without user intervention) a hierarchy of coarser meshes and the corresponding linear systems. To do so, the algorithm extracts from the fine grid matrix information about the strength of coupling between different nodes. In the solution phase, this hierarchy of discrete problems is used to solve the problem by multigrid cycling. When constructing the hierarchy of coarser discretizations, algebraic multigrid tries to balance the quality of the smoother with that of the coarse grid correction. Error components not damped by the coarse grid correction must be taken care of by the smoother and vice versa. The smoother in AMG is typically a simple point Gauss-Seidel method and a major part of the work is invested in building a coarse grid correction that makes up for the simplicity of the smoother. The coarse grid equivalent A^H of the system matrix A^h is built by using the Galerkin formula

$$A^H = I_h^H A^h I_H^h. \tag{16}$$

For symmetric problems the restriction operator is chosen to be the transpose of the prolongation operator: $I_h^H = (I_H^h)^T$. Given the smoother, the restriction and the coarse level discretization, only the selection of coarse grid points and the construction of the interpolation operator remains to be detailed. The coarse grid selection includes a partition of the fine grid variables Ω^h into two disjoint sets $\Omega^h = C^h \cup F^h$, with C^h and F^h the coarse grid and fine grid variables respectively. The next coarse grid Ω^H is then identified with C^h. Each fine grid point $i \in F^h$ is interpolated from a subset $P_i^h \subset C^h$ of the coarse grid variables, called the interpolatory variables to point i. The interpolation operator has the form:

$$e_i^h = (I_H^h e^H)_i = \begin{cases} e_i^H & \text{if } i \in C^h, \\ \sum_{j \in P_i} w_{ij}^h e_j^H & \text{if } i \in F^h. \end{cases} \tag{17}$$

The interpolation is constructed by requiring that smooth errors are accurately transferred from coarse to fine grids. The concept of smoothness can be defined purely algebraically: algebraically smooth errors satisfy

$$A_{ii} e_i + \sum_{j \in N_i} A_{ij} e_j \approx 0, \tag{18}$$

with $N_i = \{j \neq i; A_{ij} \neq 0\}$. This simple fact about smooth errors gives the basic information from which the interpolation operators are deduced. The construction of interpolation based solely on (18) results in an interpolation for which the size of the sets of interpolation variables $P_i^h, i \in F^h$, is too large, resulting in cycles that are computationally too expensive. When truncating the size of these sets, each fine grid point has to remain sufficiently connected to its set of interpolatory coarse grid points. The truncation strategy leads to heuristics for selecting the coarse grid. Once the coarse grid has been selected, the interpolation weights w_{ij}^h in (17) can be calculated.

C. Algebraic Multigrid for Systems

The original AMG methods were designed to handle the type of systems that result from the discretization of a scalar elliptic PDE. More recent algorithmic variants exist for coupled systems of PDEs. The SAMG code [10], which is used for our simulations, can handle both scalar PDEs and coupled systems of PDEs. For coupled systems, SAMG offers three basic solution strategies. We use the terminology from the SAMG user's manual to describe them. A *variable* is a solution component of the linear system. An *unknown* is a (scalar) physical function being approximated. In our case the oxygen and carbon dioxide concentration are the two unknowns. A *point* is a location in space where one or more variables are defined. In our case the points are the nodes of the finite element mesh.

In the variable-based approach, the coupled system is treated just like a scalar one. That is, SAMG's coarsening process is on the level of variables without distinguishing unknowns or points. This approach is not appropriate for the respiration-diffusion model since it only works when the coupling between the unknowns is very weak. In the unknown-based approach, coarsening is on the level of variables, but variables corresponding to different unknowns are treated independently. The coupling between the unknowns is neglected when constructing the restriction and interpolation operators. The coupling is not neglected when constructing the coarse grid Galerkin matrices, but the smoother is simplified by updating unknowns separately. In our case, for example, first all oxygen concentration variables are updated and then all carbon dioxide concentration variables. The third strategy is the point-based approach, with coarsening on the level of points. The structure of the coarse grids is the same for each unknown and can, for example, be based on the connectivity pattern of one of the unknowns. In our experiments interpolation and restriction are separate for each physical unknown. For the point-based approach a block Gauss-Seidel smoother is used that updates all unknowns at a point simultaneously by solving a small system of equations.

VI. Results

We will give some results of applying AMG as a solver in the numerical simulation of the respiration-diffusion model. First we use the code on a simplified test problem. Then we consider the use of SAMG for the steady-state pear model. The last example uses SAMG for the time-dependent pear model. For all the experiments, the MATLAB software developed by the Laboratory of Postharvest Technology is used. For the multigrid experiments, the MATLAB direct sparse solver is replaced by calls to the AMG software, which is written in Fortran.

A. Scalar Steady-State Test Problem

We simplify the pear model by considering a steady-state, scalar, linearized model problem where the oxygen concentration is the only unknown.

$$\nabla \cdot (D_{O_2} \nabla C_{O_2}) - V_{O_2} = 0 \tag{19}$$

Variables	V(1,1)	V(1,1)/CG	V(2,2)	W(1,1)
1626	0.19	0.05	0.01	0.19
2720	0.13	0.04	0.01	0.13
6514	0.15	0.04	0.01	0.15
11745	0.17	0.05	0.01	0.17

Table 2: Convergence factors of the different multigrid cycles for four different meshes, with 1626 up to 11745 variables.

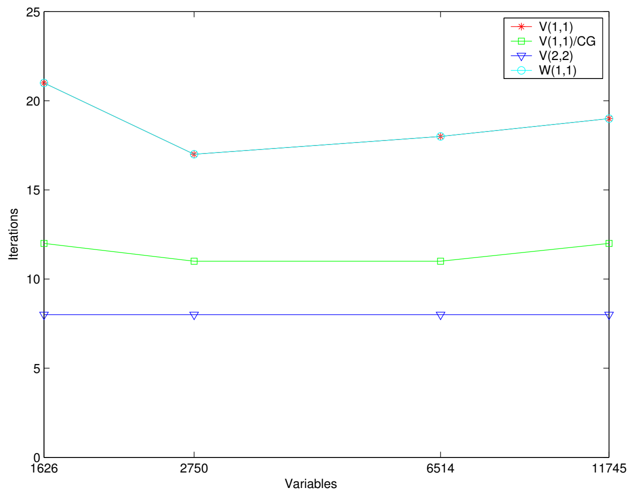

Figure 4: Number of iterations as a function of the number of variables, for different multigrid cycle types.

The oxygen consumption term V_{O_2} is a linear function; its value is proportional to the oxygen concentration C_{O_2}. This is the type of equation for which the original AMG algorithm was designed. The scalar test problem is solved using four different finite element meshes, see table 2 for the number of variables. We use Stüben's original AMG1R5 multigrid code with different multigrid cycles to solve the linear system that results from the finite element discretization. The cycles are of type V(1,1), V(2,2), or W(1,1), where the numbers between the parentheses indicate the number of pre-smoothing resp. post-smoothing steps. The V(1,1)-cycle is also used as a preconditioner for a CG algorithm.

The number of iterations needed to reduce the norm of the residual of the initial approximation by a factor 10^{12} is plotted in figure 4. This figure shows that each multigrid cycle needs about a constant number of iterations for the different grids. Table 2 shows the averaged convergence factors for the different cycle types. It illustrates that the convergence factor is more or less independent of the mesh size.

B. Steady-State Pear Model

Next, we illustrate the performance of the AMG code for the full 3D pear model, consisting of two coupled non-linear reaction diffusion equations with mixed type boundary conditions. A problem was selected with the following parameters for the environment: $T = 20°C, C^{\infty}_{O_2} = 20.8\%, C^{\infty}_{CO_2} = 0\%$. We compare the performance of the direct sparse solver with the performance of the SAMG-code. Both techniques are used as linear system solvers inside an optimized modified Newton strategy. The direct sparse solver is based on an LU factorization, which needs to be done only when the Jacobian is recomputed. The back-substitution, using the computed L and U factors is done once per Newton step. The SAMG-code employs the unknown-based approach, and uses V(2,1) multigrid cycles with CG acceleration. Only two iterations are done per Newton step. This turned out to be sufficient to avoid any increase in the number of Newton steps.

Mesh	Nodes	Unknowns	Non-zeroes	Gauss	AMG
1	1544	3088	75624	4s	0.35s
2	3447	6894	174156	44s	0.62s
3	4533	9066	229692	77s	0.85s
4	6469	12938	332636	167s	1.35s
5	11699	23398	615188		2.85s
6	13933	27866	735324		3.45s
7	20421	40842	1084316		4.51s

Table 3: Characteristics of the meshes for the 3D pear model, and execution time in seconds, per Newton step, for the direct solver (Gauss) and the iterative method (AMG).

Seven different finite element meshes are considered for this problem, the characteristics of which are summarized in table 3. The table also gives the number of nonzero elements in the stiffness matrix. In that table we report the averaged execution times per Newton step for both the Gauss method and AMG. These results were obtained on a standard 1.7GHz P4 computer with 1GB RAM. From these results we conclude that the AMG method is much more efficient. Its computational cost is approximately proportional to the size of the system. For the sparse direct solver the work is proportional to a quantity between the square and the cube of the problem size.

We would also like to comment on the memory usage. We consider only the smallest problem with a stiffness matrix containing 75,624 nonzero elements. The direct solver created very significant fill-in, which is typical for 3D problems, and needed storage for a total of 5,184,090 nonzero values. This is about 68 times the storage needed for representing the original problem! For the SAMG code, however, a storage of a mere 127,948 nonzero elements turned out sufficient. That is less than two times the storage of the stiffness matrix.

C. Time-Dependent Pear Model

To solve the time-dependent pear model, we use an implicit time discretization method and solve in each time-step the discrete problem with the SAMG solver. The implicit schemes considered in our study are the implicit Euler method (BDF1), the backward difference formula of order 2 (BDF2) and the Radau IIA implicit Runge-Kutta method of order 5 (IRK).

For the first two schemes an unknown-based approach is used with standard coarsening, variable-wise Gauss-Seidel relaxation (first all O_2, then all CO_2 concentrations), and a V(2,1)-cycle as a preconditioner for CG. The total CPU-time for BDF1 and BDF2 time-integration using the direct sparse solver and using the SAMG solver are shown in figure 5. Note that the curves for BDF1 and BDF2 cannot be distinguished. This is to be expected as the linear systems have the same size and a similar structure. It is clear that the SAMG solver is much more efficient. It can easily be used for much larger problems.

Our final results show that the multigrid approach is equally promising when IRK time discretization is used. In this case there are 6 unknowns for each grid-point: 2 concentrations and 3 stages values for each. The dimension of the linear system is therefore 6 times the number of nodes in the finite element mesh, i.e., 3 times larger than in the BDF case. Because the coupling between the stage values for each concentration is strong, the point-based approach of SAMG is used. Coarsening is done on the level of points with interpolation separate for each of the 6

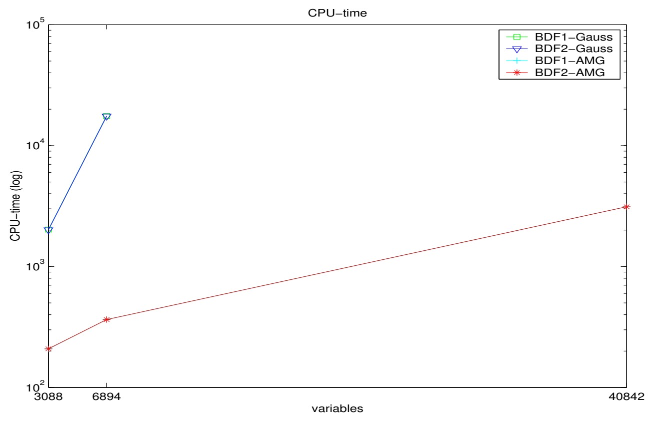

Figure 5: Timing results for BDF1 and BDF2 time integration of the 3D pear model.

	Gauss	AMG
CPU-Time(s)	148	40
# Newton iter	493	493

Table 4: Results for time integration of a 2D pear model with 332 mesh points, using the Radau IIA IRK-method.

	Gauss	AMG
CPU-Time(s)	12376	339
# Newton iter	364	423

Table 5: Results for time integration of the 3D pear model with 1544 mesh points, using the Radau IIA IRK-method.

unknowns. A block Gauss-Seidel relaxation with 6 by 6 blocks is used as smoother. A multigrid iteration with V(1,1) cycle is used as a preconditioner for CG. Table 4 shows timing results for the numerical simulation of a discretization of a 2D pear model with $2 \times 3 \times 322 = 1932$ variables. Table 5 shows timing results for the discretization of the 3D model with $2 \times 3 \times 1544 = 9264$ variables. For the 2D problem, the Gauss method is somewhat slower, but still quite acceptable. For the 3D problem, however, the SAMG solver clearly outperforms the direct method, even for this relatively small problem.

VII. Concluding Remarks

We have reported in this paper on our experiences with an algebraic multigrid code for solving a mathematical model for the respiratory activity in a Conference pear. The use of AMG has enabled us to perform both stationary and time-dependent simulations with an accuracy and computational efficiency that could not be achieved with the algorithmic components that were used before. Currently, our research activities progress in different directions. First, we try to quantify the effect on the solution values of the O_2 and CO_2 concentration of uncertainties in the model parameters (diffusion coefficients, reaction rates, etc.), characterized by a probability density function. For this, a stochastic finite element model is being developed and solved [2]. Another application deals with water transport in pear tissue which may cause shrivelling and, consequently, loss of appearance and commercial value. As pear tissue is cellular it is not really a continuum, multi-scale models for water and gas transport are currently being developed. We believe that such models may provide valuable insight into the relevant postharvest processes and eventually will lead to fruit of improved quality.

VIII. References

[1] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2000.

[2] R.G. Ghanem and P.D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.

[3] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer-Verlag, 1991.

[4] D. Lahaye, H. De Gersem, S. Vandewalle, and K. Hameyer. Algebraic multigrid for complex symmetric systems. *IEEE Transactions on Magnetics*, 36(4):1535–1538, 2000.

[5] D. Lahaye, S. Vandewalle, and K. Hameyer. An algebraic multilevel preconditioner for field-circuit coupled problems. *J. Comput. Appl. Math.*, 168(1-2):267–275, 2004.

[6] J. Lammertyn, M. Aerts, B.E. Verlinden, W. Schotmans, and B.M. Nicolaï. Logistic regression analysis of factors influencing core breakdown in ‘Conference’ pears. *Postharvest Biology and Technology*, 20:25–37, 2000.

[7] J. Lammertyn, C. Franck, B.E. Verlinden, and B.M. Nicolaï. Comparative study of the O_2, CO_2 and temperature effect on respiration between ‘Conference’ pear cell protoplasts in suspension and intact pears. *Journal of Experimental Botany*, 362:1769–1777, 2001.

[8] J. Lammertyn, N. Scheerlinck, P. Jancsók, B.E. Verlinden, and B.M. Nicolaï. A respiration-diffusion model for ‘Conference’ pears I: model development and validation. *Postharvest Biology and Technology*, 30(1):31–44, 2003.

[9] J. Lammertyn, N. Scheerlinck, P. Jancsók, B.E. Verlinden, and B.M. Nicolaï. A respiration-diffusion model for ‘Conference’ pears II: simulations and relation to core breakdown. *Postharvest Biology and Technology*, 30(1):45–57, 2003.

[10] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128:281–309, 2001.

[11] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[12] J. Van lent and S. Vandewalle. Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of parabolic PDEs. *SIAM Journal on Scientific Computing*, pages 1–25, 2005. in print.

Jan Van lent, Dominik Smits, Stefan Vandewalle*
Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A
B-3001 Leuven, Belgium

Nico Scheerlinck, Bart Nicolaï
Katholieke Universiteit Leuven
Laboratory of Postharvest Technology
de Croylaan 42
B-3001 Leuven, Belgium

*Corresponding author
`stefan.vandewalle@cs.kuleuven.ac.be`