ASC Report No. 18/2013

Collocation method for the modeling of membrane gas permeation systems

A. Feichtinger, A. Makaruk, E. Weinmüller, A. Friedl, M. Harasek

Institute for Analysis and Scientific Computing
Vienna University of Technology — TU Wien
www.asc.tuwien.ac.at ISBN 978-3-902627-05-6

Most recent ASC Reports

17/2013 *M. Langer, H. Woracek*
Distributional representations of $\mathcal{N}_\kappa^{(\infty)}$-functions

16/2013 *M. Feischl, T. Führer, M. Karkulik, D. Praetorius*
ZZ-type a posteriori error estimators for adaptive boundary element methods on a curve

15/2013 *K.-N. Le, M. Page, D. Praetorius, and T. Tran*
On a decoupled linear FEM integrator for Eddy-Current-LLG

14/2013 *X. Chen, A. Jüngel, and J.-G. Liu*
A note on Aubin-Lions-Dubinskii lemmas

13/2013 *G. Kitzler, J. Schöberl*
Efficient Spectral Methods for the spatially homogeneous Boltzmann equation

12/2013 *M. Miletic, A. Arnold*
An Euler-Bernoulli beam equation with boundary control: Stability and dissipative FEM

11/2013 *C. Chainais-Hillairet, A. Jüngel, and S. Schuchnigg*
Entropy-dissipative discretization of nonlinear diffusion equations and discrete Beckner inequalities

10/2013 *H. Woracek*
Entries of indefinite Nevanlinna matrices

09/2013 *L. Banas, M. Page, and D. Praetorius*
A convergent linear finite element scheme for the Maxwell-Landau-Lifshitz-Gilbert equation

08/2013 *L. Banas, M. Page, D. Praetorius, and J. Rochat*
On the Landau-Lifshitz-Gilbert equations with magnetostriction

Institute for Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8–10
1040 Wien, Austria

E-Mail: admin@asc.tuwien.ac.at
WWW: http://www.asc.tuwien.ac.at
FAX: +43-1-58801-10196

ISBN 978-3-902627-05-6

ASC TU WIEN

Collocation method for the modeling of membrane gas permeation systems

A. Feichtinger[a], A. Makaruk[b,*], E. Weinmüller[a], A. Friedl[b], M. Harasek[b]

[a] *Vienna University of Technology, Institute for Analysis and Scientific Computing, Wiedner Hauptstrasse 8, 1040 Vienna, Austria*
[b] *Vienna University of Technology, Institute of Chemical Engineering, Getreidemarkt 9/166, 1060 Vienna, Austria*

Abstract

In this work, we describe a numerical method which enables an efficient computation of membrane gas permeation processes that involve multiple membrane stages and multiple gas components. The utilized numerical approach is a collocation method equipped with a grid adaptation strategy based on a dependable error estimate of the numerical approximation. The comparison of the results provided by the collocation method with those calculated from an experimentally validated finite difference method has demonstrated, that the accuracy of both numerical approximations is practically the same. However, the current procedure is characterized by a much better computational efficiency that allows to considerably reduce the computational time. This is a crucial feature when combining computation of membrane permeation processes with optimization algorithms. In such a setting the computation of the permeation process is frequently repeated and naturally, results in long computational times when the efficiency is not adequately improved.

Keywords: membrane gas permeation, process simulation, collocation method, error estimate, grid adaptation, multicomponent separation

1. Introduction

During the recent decades, membrane gas permeation became an important process in numerous chemical engineering branches like air separations (Bhide and Stern, 1991a,b; Baker, 2002), natural gas upgrading (Baker and Lokhandwala, 2008; Bhide and Stern, 1993a,b), gas dehydration (Baker, 2002; Lin et al., 2012; Sijbesma et al., 2008), separations of Volatile Organic Components (VOC) (Feng et al., 1993; Liu et al., 2006), and gas desulfurization (Chatterjee et al., 1997; Wilks and Rezac, 2002; Makaruk et al., 2013). In addition, membrane gas permeation is often a process of choice in the production of renewable gaseous fuels. Here, the following important processes are worth mentioning: biogas upgrading and production of renewable natural gas substitute (Schell and Houston, 1983; Stern et al., 1998; Makaruk et al., 2010) or hydrogen recovery from biomass gasification gases (Mayer et al., 2010; Makaruk et al., 2012).

*Corresponding author: Aleksander Makaruk, Tel.:+43 1 58801 166279; Fax: +43 (0) 1 58801 15999

Email address: aleksander.makaruk@tuwien.ac.at (A. Makaruk)

URL: http://www.membran.at (A. Makaruk)

Efficient engineering of membrane gas permeation systems requires adequately designed tools. The gas transport phenomena in membrane gas permeation are modeled by systems of nonlinear differential equations whose solution requires dependable codes for the involved numerical simulations. In the literature, a number of different approaches to the computation of the permeation processes is available. They span from simple trial-and-error methods through integration methods to more sophisticated techniques involving domain discretization. A more detailed review of those methods can be found in (Makaruk and Harasek, 2009). The important properties of the numerical algorithm necessary for the successful and efficient simulation of the gas permeation are fast convergence and short computation time – aspects which are often disregarded (Makaruk and Harasek, 2009; Stern et al., 1984). These properties are especially crucial when the computation of the gas permeation is an auxiliary routine in an optimizing algorithm that identifies some desired process design parameters. In this case algorithm's speed and reliability are crucial, because the auxiliary routine is executed many times.

In the present article, a numerical method based on the direct discretization of the problem is used to solve

membrane gas permeation systems. The basic solver, polynomial collocation, is equipped with an error estimation procedure and a grid adaptation strategy to speed up the calculations.

The paper is organized as follows. In Section 2, we briefly introduce the background of membrane gas permeation and represent single stage and multi-stage permeation problems in the form of boundary value problems for systems of ordinary differential equations (BVP). The numerical algorithm is discussed in Section 3. Finally, results of the numerical simulation are presented and validated with the experimental data available in the literature.

2. Theory

The presented numerical method is designed to model gas separation processes that employ dense membranes in the form of hollow fibers. In these membranes, the selective separating layer is typically made of thin polymer film (in the range of μm) that is positioned on a thicker porous layer for mechanical support. A gas mixture is fed under pressure into a membrane module and is separated into the high pressure retentate stream and into the low pressure permeate stream. The gas molecule transport through thin polymer layers obeys the Fick's and Henry's laws, i.e. the gas molecules first dissolve in the polymer film on the high pressure side, then diffuse through the thin film and desorb on the low pressure permeate side (Wijmans and Baker, 1995). The laws of gas solution and diffusion can be combined to yield the following equation for the transmembrane volume flow (STP) of a single gas component i,

$$\dot{J}_i = \Pi_i A(p_{F,i} - p_{P,i}), \tag{1}$$

where Π_i is the proportionality factor (permeance), A is the membrane area, $p_{F,i}$ and $p_{P,i}$ are components' partial pressures on the feed and permeate side of the membrane, respectively. For most of the engineering calculations of membrane gas permeation systems, the following assumptions hold: 1) the flow can be assumed to be one-dimensional, 2) the concentration polarization effect, i.e. the variation of gas composition in the direction normal to the membrane surface is negligible. Hence, the change of bulk flow F of the component i on the high pressure side of the membrane can be represented for the co-current flow configuration by

$$F_i' = \Pi_i \pi s D \left(-\frac{F_i}{\sum_{j=1}^{k} F_j} p_F + \frac{P_i}{\sum_{j=1}^{k} P_j} p_P \right), \tag{2}$$

where s is the number of hollow fibers, D is the cross-sectional diameter of the selective layer, k is the number of gas components in the mixture, p_F is the pressure on the feed/retentate side, p_P is the pressure on the permeate side. Similar equation holds for the permeate side of the membrane,

$$P_i' = \Pi_i \pi s D \left(\frac{F_i}{\sum_{j=1}^{k} F_j} p_F - \frac{P_i}{\sum_{j=1}^{k} P_j} p_P \right). \tag{3}$$

In case of the counter-current flow configuration, the gas volume conservation on the permeate side is represented by

$$P_i' = \Pi_i \pi s D \left(-\frac{F_i}{\sum_{j=1}^{k} F_j} p_F + \frac{P_i}{\sum_{j=1}^{k} P_j} p_P \right). \tag{4}$$

We now rewrite equations (2)-(4) in such a way that they match the notation used in Section 3, see Equation (8). Therefore, we set $z_{2i-1} := P_i$ and $z_{2i} := F_i$. In the solution vector for k gas components, $z = (z_1, ..., z_{2k})$, z_i with odd i represent the volume flows of single gas components in the feed channel and z_i with even i represent the volume flows of single gas components in the permeate channel.

Note that z_i is a function of t, where t is the longitudinal distance along hollow fibers and $t \in [0, l]$, where l is the length of the module. To cover multistage systems with different module lengths, it is necessary to scale the interval $[0, l]$ to the normalized interval $[0, 1]$, which is done by multiplying the right hand side of the involved differential equations by l. The resulting differential equations for one gas component and co-current flow have the form

$$z_{2i-1}'(t) = \Pi_i \pi s D \left(-\frac{z_{2i-1}(t)}{\sum_{j \text{ odd}} z_j(t)} p_F + \frac{z_{2i}(t)}{\sum_{j \text{ even}} z_j(t)} p_P \right) l, \tag{5}$$

$$z_{2i}'(t) = \Pi_i \pi s D \left(\frac{z_{2i-1}(t)}{\sum_{j \text{ odd}} z_j(t)} p_F - \frac{z_{2i}(t)}{\sum_{j \text{ even}} z_j(t)} p_P \right) l. \tag{6}$$

For the counter-current flow the second equation takes the form

$$z'_{2i}(t) = \Pi_i \pi s D \left(-\frac{z_{2i-1}(t)}{\sum\limits_{j \text{ odd}} z_j(t)} p_F + \frac{z_{2i}(t)}{\sum\limits_{j \text{ even}} z_j(t)} p_P \right) l. \qquad (7)$$

Now we formulate the necessary boundary conditions closing the system. At the gas inlet on the feed side of the membrane, the related boundary conditions read $z_{2i-1}(0) = \chi_i f$, where χ_i is the volume fraction of the i-th gas component in the gas mixture and f is the total gas volume flow. In the co-current case, if no sweep gas is introduced, the boundary conditions at the inlet to the permeate side of the membrane read $z_{2i}(0) = 0$. They are used with Equation (6). The initial conditions, $z_{2i-1}(0) = \chi_i f$, $z_{2i}(0) = 0$, can be written in the form of a linear system (9), where B_0 is chosen as a $2k \times 2k$ identity matrix, B_1 as a $2k \times 2k$ zero matrix and $\beta = (\chi_1 f, 0, \chi_2 f, 0, \ldots, \chi_k f, 0)$.
In the case of the counter-current configuration, the boundary conditions for the feed side read $z_{2i-1}(0) = \chi_i f$ and for the permeate side they are $z_{2i}(1) = 0$. Now B_0 and B_1 are the following $2k \times 2k$ diagonal matrices: $B_0 = \text{diag}(1, 0, 1, 0, \ldots, 1, 0)$ and $B_1 = \text{diag}(0, 1, 0, 1, \ldots, 0, 1)$. The vector $\beta = (\chi_1 f, 0, \chi_2 f, 0, \ldots, \chi_k f, 0)$ remains unchanged.

2.1. Multistage Systems

In a multistage system, we apply Equations (5)-(7) separately for each stage. This means that for a S-stage system with k gas components, we have to solve $2Sk$ equations.
Here, it is more difficult to adapt the boundary conditions. They depend on the structure of the system and have to be specified to reflect the module configuration.

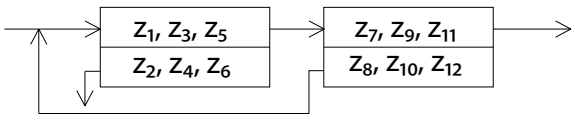

Figure 1: Two-stage counter-current system for three gas components.

For instance for a three component gas mixture permeating in the two-stage counter-current system shown in Figure 1, the boundary conditions are

$$\begin{aligned}
&z_1(0) = z_8(0) + \chi_1 f,\ z_2(1) = 0,\\
&z_3(0) = z_{10}(0) + \chi_2 f,\ z_4(1) = 0,\\
&z_5(0) = z_{12}(0) + \chi_3 f,\ z_6(1) = 0,\\
&z_7(0) = z_1(1),\ z_8(1) = 0,\\
&z_9(0) = z_3(1),\ z_{10}(1) = 0,\\
&z_{11}(0) = z_5(1),\ z_{12}(1) = 0.
\end{aligned}$$

3. Numerical Method

In order to solve the BVP in ordinary differential equations (ODEs) derived in Section 2, we apply the so called collocation method implemented in the open domain MATLAB code `bvpsuite` (Kitzhofer et al., 2010). Below, we briefly describe the basic properties of our numerical approach.

3.1. Well-Posedness of the Model

We first consider the following BVP in ODEs to describe the algorithm used for numerical simulations in Section 4. The problem consists of a nonlinear system of first order differential equations

$$z'(t) = f(t, z(t)), \quad t \in [0, 1], \qquad (8)$$

subject to boundary conditions

$$B_0 z(0) + B_1 z(1) = \beta. \qquad (9)$$

Here, we denote by z the n-dimensional real valued solution vector, whose components z_i, $i = 1, 2, \ldots, n$, are real valued scalar functions defined on the interval $[0, 1]$. The function f is also n-dimensional and we assume that it is smooth on a suitable domain. This means that higher derivatives of f exist and are bounded. The matrices B_0 and B_1 are real square $n \times n$ matrices and β is a real n-dimensional vector. The interval of integration is here normalized to $[0, 1]$, but everything can be straightforwardly extended to an arbitrary interval $[a, b]$, where $a < b$.

To enable successful numerical treatment, the above BVP has to satisfy certain structural properties which are usually referred to as the well-posedness of the model. This means that the ODE-system (8) subject to boundary conditions (9) has a unique solution and this solution depends continuously on the problem data. The well-posedness of the problem is an important property of the model which allows to express the errors in the solution in terms of the modeling errors and the data errors (all measured via appropriate norms).

Therefore, when the errors in the data become smaller due to more precise modeling or smaller measurement inaccuracies, the errors in the solution will decrease.

3.2. Solving BVPs in ODEs using `bvpsuite`

In the sequel, we discuss basic principles of the numerical approach used for the numerical experiments in Section 4. To compute the numerical solution of (8)–(9), polynomial collocation (Kitzhofer et al., 2010) is used. To enhance computational efficiency, grid is adapted to appropriately reflect the solution behavior. The necessary prerequisite for the grid adaptation strategy is a dependable error estimate of the approximation.

3.2.1. Collocation Method

Consider again the BVP (8)–(9) with the exact solution $z(t)$, $t \in [0,1]$. In order to provide the numerical approximation for z, we first introduce an equidistant partition of the interval $[0,1]$,

$$\Delta_h = \{0 = \tau_0 < \tau_1 < \ldots < \tau_i < \tau_{i+1} < \ldots < \tau_N = 1\},$$

where $h = \tau_{i+1} - \tau_i$, $i = 0, \ldots, N-1$. In each subinterval $I_i := [\tau_i, \tau_{i+1}]$, we locate m collocation points $t_{ik} = \tau_i + \rho_k h$, where $k = 1, 2, \ldots, m$ and $0 \le \rho_1 < \rho_2 < \ldots < \rho_m \le 1$. Note, that for $\rho_1 = 0$, $t_{i0} = \tau_i$ and for $\rho_m = 1$, $t_{im} = \tau_{i+1}$ which means that the first and last collocation points coincide with the grid points. The computational grid is shown in Figure 2.

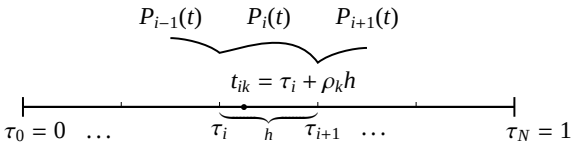

Figure 2: The computational grid.

For simplicity, let us first assume that the analytical problem is scalar, i.e. $n = 1$. The aim is to approximate z by a piecewise polynomial function $P(t) := P_i(t)$, $t \in I_i$ which is continuous in $[0,1]$. With other words, in each subinterval I_i, we approximate z by a polynomial P_i of degree m. Clearly, the number of unknowns uniquely describing a polynomial of degree m is $m+1$ and therefore, in P, the global number of unknowns is $N(m+1)$. The numerical approximation P is now computed from a nonlinear system of equations given below. First of all, we require that the differential equation (8) is exactly (up to the round off errors) satisfied in all collocation points,

$$P_i'(t_{ik}) = f(t, P_i(t_{ik})), \quad t_{ik} \in I_i, \qquad (10)$$
$$i = 0, 1, \ldots, N-1, \;\; k = 1, 2, \ldots, m$$

Moreover, the continuity conditions

$$P_i(\tau_{i+1}) = P_{i+1}(\tau_{i+1}), \quad i = 0, 1, \ldots, N-2, \qquad (11)$$

and boundary condition have to be satisfied

$$B_0 P_0(0) + B_1 P_{N-1}(1) = \beta. \qquad (12)$$

Adding the number of conditions given in (10)–(12), we obtain $Nm + N - 1 + 1 = N(m+1)$ which is equal to the number of unknowns. Consequently, the system (10)–(12) is closed. Note, that this scheme can be easily generalized to the case when the step size h is not constant.

The major question which now has to be addressed is the convergence of the scheme for $h \to 0$. This means that we are interested in the behavior of the maximal global error $\|z - P\|_\infty := \max_{0 \le t \le 1} |z(t) - P(t)|$ for $h \to 0$. In particular, it is interesting to know how fast this error decreases, or equivalently, for what $p > 0$ the following statement (a priori error estimate) holds:

$$\|z - P\|_\infty = \max_{0 \le t \le 1} |z(t) - P(t)| = c(h,z) h^p. \qquad (13)$$

Here, $c(h,z)$ depends on higher derivatives of z and $\lim_{h \to 0} c(h,z) = c > 0$. The constant p is the convergence order of the collocation scheme. Clearly, the representation (13) makes sense, when all necessary higher derivatives of z which occur in $c(h,z)$ exist and are bounded on $[0,1]$. This question of convergence addressed above, has been answered in (de Boor and Swartz, 1973) and it turns out that for an appropriately smooth problem (8)–(9) with a smooth solution z, the convergence order of the collocation scheme is $p = m$. This result means that for problems with smooth solutions it is more efficient to use high order methods than the low order ones, especially when the global error shall be small. To see this, let us assume that all solution derivatives are moderate, $c(h,z) = O(1)$. Then $\|z - P\|_\infty \approx h^p$. Further assume that we wish $\|z - P\|_\infty \approx 10^{-7}$. For a low order method, with for example $p = 1$, we have to use the stepsize $h \approx 10^{-7}$, while for a method of order $p = 7$ it is sufficient to use $h \approx 10^{-1}$. Consequently, in the first case we have to solve for around $2 \cdot 10^7$ unknowns[1],

[1] Since $m = p = 1$, we work with polynomials P_i of degree 1 and hence, each of P_i is uniquely specified by 2 unknown parameters. The stepsize $h = 10^{-7}$ means that on the interval $[0,1]$ we have to compute 10^7 polynomials. Therefore, in this case the number of unknowns is $2 \cdot 10^7$.

while in the second case the number of unknowns is around 80.

We see that collocation provides an approximation P for the solution z on a prescribed grid Δ_h, where the step size may be constant or vary. In general, a software package for solving BVPs in ODEs provides additional modules controlling the computational process. We now motivate and discuss these controlling mechanisms – error estimate and grid adaptation procedures – in some detail.

3.2.2. Error Estimates for the Global Error of the Collocation

The estimation of the error of P is necessary, because the user not only specifies the problem and expects to obtain an approximation for its solution, but also prescribes how accurate the numerical solution shall be. Using a tolerance parameter TOL, the user may wish the maximal error of the approximation to satisfy $\|z - P\|_\infty \leq TOL$. This means that we have to compute an estimate *est* for the unknown global error of the collocation polynomial, $z - P$, in order to be able to check if the requirement $\|est\|_\infty \leq TOL$ is satisfied. If this tolerance requirement was not satisfied on a grid Δ_h, we can half the step size and try the grid $\Delta_{h/2}$. Since, according to representation (13), decreasing the step size results in decreasing the global error (until the round off error level is reached), we shall be able to find h which is sufficiently small for the approximation to become appropriately precise. Clearly, the error estimate has to reflect the size of the true error correctly, at least for fine grids with small h. Error estimate satisfying this property is called asymptotically correct.

To provide an asymptotically correct estimate for the global error of the collocation solution, we propose to use the classical error estimate based on mesh halving. In this approach, we compute the collocation solution on a mesh Δ_h with the step size h and denote this approximation by $P_{\Delta_h}(t)$. Subsequently, we choose a second mesh $\Delta_{h/2}$ where in every interval of Δ_h we insert two subintervals of length $h/2$. On this mesh, we compute the numerical solution using the same collocation scheme to obtain the collocation polynomial $P_{\Delta_{h/2}}(t)$. Using these two quantities, we define

$$est(t) := 2^m(P_{\Delta_{h/2}}(t) - P_{\Delta_h}(t))/(1 - 2^m)$$

as an error estimate for the approximation $P_{\Delta_h}(t)$. This formula is executed on each subinterval I_i of Δ_h. Generally, estimates of the global error based on mesh halving

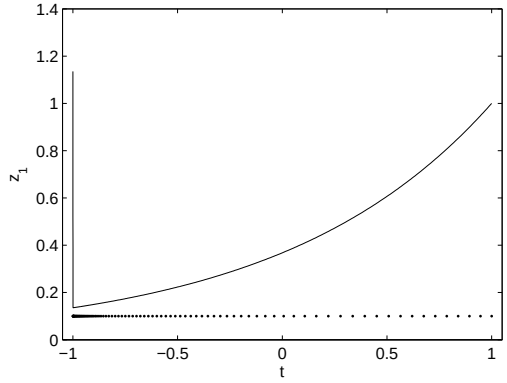

Figure 3: Numerical solution and the related adapted grid: $TOL = 10^{-6}$, number of grid points 101, $m = 8$.

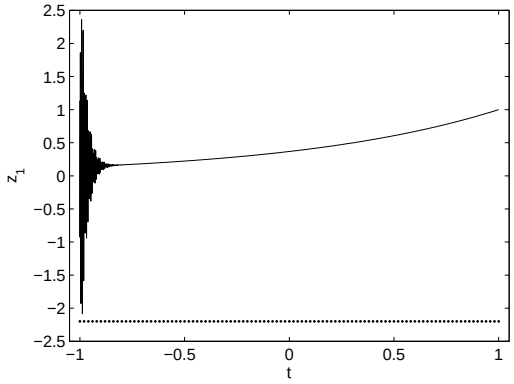

Figure 4: Numerical solution and the related uniform grid, number of grid points 101, $m = 8$.

are robust and therefore, this strategy has been implemented in `bvpsuite`. Note, that this strategy will work analogously for variable step sizes $h_i := \tau_{i+1} - \tau_i$.

3.2.3. Adaptive Mesh Selection

By decreasing the step size coherently, $h \to h/2 \to h/4 \ldots$, it will be in general, possible to satisfy the tolerance requirement but this procedure is inefficient, because it does not take into account the solution behavior and the structure of the error.

In Figures 3 and 4, the advantage of an adaptive grid is illustrated. The underlying analytical problem is a BVP for a system of two equations of the form (8) whose first solution component shows a steep layer at the left end of the interval of integration. We see, that the grid points in the adapted grid very well reflect the solution behavior. With the same number of equidistantly spaced grid points, the same effort is paid

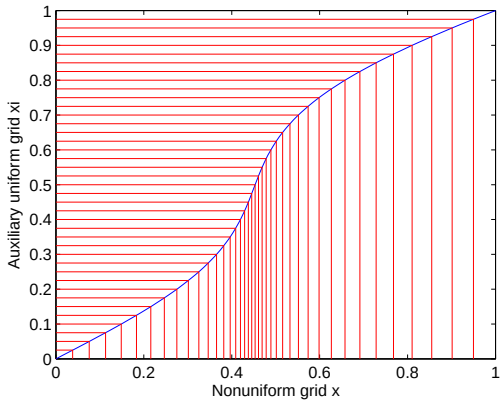

Figure 5: Uniform auxiliary grid maps to nonuniform grid, where $x_n = \Phi^{-1}(\xi_n)$.

but the obtained approximation is unacceptable.

A correct error estimate of the global error is a good indicator for the regions where the solution is difficult to approximate. These regions are usually characterized by a rapid solution change, or equivalently, by large values of its higher derivatives. This also means that the function $c(h, z)$ will be large and so will be the global error. The main idea is to locate the grid points in such a way that the global error becomes equidistributed or constant along the grid. With other words, the grid becomes finer with smaller step sizes in regions where the error is large (solution changes rapidly) and stays coarse with larger step sizes in regions where the error is small (solution changes slowly). This idea can be realized in several ways. The mesh selection strategy discussed below was proposed and investigated in (Pulverer et al., 2011). The new control algorithm consists of two phases. In the first phase carried out on the control grid with a moderate number of points, the grid points are located in such a way that they correctly reflect the solution behavior, cf. Figure 3. Most modern mesh generation techniques in two-point boundary value problems construct a smooth function mapping a uniform auxiliary grid ξ to the desired nonuniform grid x. The aim is to construct a grid deformation $\xi = \Phi(x)$ with $\Phi'(x) = \phi(x)$, cf. Figure 5. Then $\mathrm{d}\xi = \phi(x)\mathrm{d}x$ and $\Delta\xi \approx \phi(x)\Delta x$. If $\Delta\xi$ is constant then $\Delta x_{n+1/2} = \Delta\xi/\phi(x_{n+1/2})$ varies with $\phi(x)$. Note that ϕ represents the *density* of the grid points – when ϕ is small Δx is large and vice versa. Using an error estimate, a feedback control law generates a new density from the previous one.

In the second phase of the grid adaptation procedure, appropriate number of grid points is added (along the grid density function) to satisfy the tolerance. In Figure 6, it can be seen how this strategy works in practice. In the top graph the behavior of the analytical solution is shown and it is clear that the grid has to be denser in the right part of the interval. In the center graph the grid adaptation procedure is visualized. The control grid consist of 21 points and is equidistant at the beginning. Then, in three iteration steps, the proper location of the grid points in the control grid is found. Finally, on the last grid containing 97 points the tolerance has been satisfied. The bottom graph shows the procedure started on a control grid with 51 points.

We would like to mention that the scope of `bvpsuite` is not restricted to models (8)–(9). The code can cope with fully implicit ODEs of variable order posed on finite or semi-infinite intervals. Differential algebraic equations (Koch et al., 2010) and parameter-dependent problems (Kitzhofer et al., 2009) are further applications for which `bvpsuite` can be utilized .

4. Results and discussion

In Figures 7 to 9, we compare numerical results that were obtained by means of the finite difference method presented in (Makaruk and Harasek, 2009) with those calculated by the collocation method that has been specified in Section 3. The results concern an experiment to depict gas contents in retentate and permeate for different stage-cut values (ratio of the permeate volume flow to the feed volume flow) and to depict the trade-off between the gas recovery and the gas purity in retentate for both cocurrent and countercurrent flow configurations. The permeation embraces three gas components: methane, carbon dioxide and oxygen. To obtain the parameter variation, the feed gas flow has been varied to produce the dependency on the variable stage-cut while the membrane area has been kept constant. All significant process parameters for this modeling study are presented in Table 1, column A. Figures 7 to 9 demonstrate that both algorithms, the experimentally verified finite difference method and the currently presented collocation method, deliver virtually the same results. Slight differences originate from the different methods accuracies and the round-off errors.

However, the advantage of the current collocation method is its high computational efficiency, especially in cases that involve pronouncedly different permeation

rates. Let us first consider a permeation process of a gas mixture containing methane and carbon dioxide in counter-current configuration under the conditions specified in Table 1, column B. To reach the accuracy of 10^{-9} the finite difference underrelaxed method requires a grid with 1000 points and 24.7 seconds to solve the problem, while the collocation method provides the same result on a grid with 10 points and calculation time 2.2 seconds.
If a third gas component with a relatively high permeation rate is added (permeation study with H_2O defined in column C in Table 1), the performance is again decisively in favor of the current collocation method. In this scenario, the collocation method solves the problem within around 6 seconds. The finite difference method requires several minutes to accomplish the same goal. The adaption of the grid by the collocation method is performed in the vicinity of the feed inlet. Exemplary trends of the gas component partial pressures along the membrane are presented in Figure 10. On the account of the fast water permeation, high gradient of the water partial pressure appears around the feed inlet. This effect is captured very well by the collocation method. However, the first order finite difference method struggles against the numerical diffusion that has a relatively strong effect in case of readily permeating water.

5. Conclusions

In the present work, we propose a numerical procedure for the modeling of membrane gas permeation processes in hollow fiber systems. For the numerical simulation, permeation processes are formulated in form of a nonlinear system of first order ordinary differential equations solved using a collocation method. Collocation has been implemented as a basic solver in the MATLAB code `bvpsuite` also featuring a grid adaptation strategy based on an error estimate for the numerical approximation. This approach enables to substantially increase the computational efficiency, which results in the reduction of required computational times, especially in cases when optimization of the membrane gas permeation processes is involved.

The public domain code `bvpsuite` is freely available from http://www.math.tuwien.ac.at/~ewa. In the future, we plan to use this code to investigate the simulations of other process units important in chemical engineering.

Nomenclature

Symbols:

a	left end of the integration interval
A	membrane area
b	right end of the integration interval
B	matrix in the boundary conditions
$c(h,z)$	error constant
D	diameter of active layer
est	estimate for the global error of the collocation
f	total gas volume flow
$f(t,z(t))$	right-hand side of the ODE
F	volume flow on the feed side
h	step size
I	sub interval in the grid
J	gas volume flow
k	number of gas components
l	module length
m	number of collocation points
n	dimension of z or f
p	pressure
p	convergence order
P	volume flow on the permeate side
P	piecewise polynomial function
s	number of hollow fibers
S	number of stages
t	longitudinal distance along fiber
TOL	tolerance
x	nonuniform grid
z	solution vector
β	vector in the boundary conditions
Δ	equidistant partition of the interval
Π	proportionality factor (permeance)
χ	volume fraction
τ	endpoints of the grid sub intervals
ρ	distribution of the collocation points
ξ	auxiliary uniform grid
ϕ	derivative of Φ
Φ	grid deformation function

Subscripts & superscripts:

0	left integration boundary
1	right integration boundary
F	feed
h	step size
i	gas component
i	vector component
i	sub interval index
j	gas component
k	number of gas components
m	number of collocation points
N	number of sub intervals
P	permeate
Δ	equidistant partition of the interval

Table 1: Gas Permeation parameters in the modeling studies.

Experiment		A	B	C
Feed gas composition [v/v]	CH_4	0.645	0.65	0.645
	CO_2	0.345	0.35	0.345
	O_2	0.01	–	–
	H_2O	–	–	0.01
Permeances [$m^3_{(stp)}$ / (m^2 s bar)]	CH_4	1.59e-6	1.59e-6	1.59e-6
	CO_2	5.91e-5	5.91e-5	5.91e-5
	O_2	1.36e-5	–	–
	H_2O	–	–	3.2e-3
Membrane area [m^2]		0.38	0.38	0.38
Feed pressure [bar]		9.0	9.0	9.0
Permeate pressure [bar]		1.1	1.1	1.1
Feed flow [$L_{(stp)}$/min]		1 – 15	1	3
Flow configuration		both	counter	counter

References

B. D. Bhide, S. A. Stern, A new evaluation of membrane processes for the oxygen-enrichment of air. I. Identification of optimum operating conditions and process configuration, Journal of Membrane Science 62 (1) (1991a) 13 – 35, ISSN 0376-7388.

B. D. Bhide, S. A. Stern, A new evaluation of membrane processes for the oxygen-enrichment of air. II. Effects of economic parameters and membrane properties, Journal of Membrane Science 62 (1) (1991b) 37 – 58, ISSN 0376-7388.

R. W. Baker, Future Directions of Membrane Gas Separation Technology, Industrial & Engineering Chemistry Research 41 (6) (2002) 1393–1411, ISSN 0888-5885.

R. W. Baker, K. Lokhandwala, Natural Gas Processing with Membranes: An Overview, Industrial & Engineering Chemistry Research 47 (7) (2008) 2109–2121, ISSN 0888-5885.

B. D. Bhide, S. A. Stern, Membrane processes for the removal of acid gases from natural gas. I. Process configurations and optimization of operating conditions, Journal of Membrane Science 81 (3) (1993a) 209 – 237, ISSN 0376-7388.

B. D. Bhide, S. A. Stern, Membrane processes for the removal of acid gases from natural gas. II. Effects of operating conditions, economic parameters, and membrane properties, Journal of Membrane Science 81 (3) (1993b) 239 – 252, ISSN 0376-7388.

H. Lin, S. M. Thompson, A. Serbanescu-Martin, J. G. Wijmans, K. D. Amo, K. A. Lokhandwala, T. C. Merkel, Dehydration of natural gas using membranes. Part I: Composite membranes, Journal of Membrane Science 413–414 (0) (2012) 70 – 81, ISSN 0376-7388.

H. Sijbesma, K. Nymeijer, R. van Marwijk, R. Heijboer, J. Potreck, M. Wessling, Flue gas dehydration using polymer membranes, Journal of Membrane Science 313 (1–2) (2008) 263 – 276, ISSN 0376-7388.

X. Feng, S. Sourirajan, F. H. Tezel, T. Matsuura, B. A. Farnand, Separation of volatile organic compound/nitrogen mixtures by polymeric membranes, Industrial & Engineering Chemistry Research 32 (3) (1993) 533–539, ISSN 0888-5885.

Y. Liu, X. Feng, D. Lawless, Separation of gasoline vapor from nitrogen by hollow fiber composite membranes for VOC emission control, Journal of Membrane Science 271 (1-2) (2006) 114 – 124, ISSN 0376-7388.

G. Chatterjee, A. A. Houde, S. A. Stern, Poly(ether urethane) and poly(ether urethane urea) membranes with high H2S/CH4 selectivity, Journal of Membrane Science 135 (1) (1997) 99 – 106, ISSN 0376-7388.

B. Wilks, M. E. Rezac, Properties of rubbery polymers for the recovery of hydrogen sulfide from gasification gases, Journal of Applied Polymer Science 85 (11) (2002) 2436–2444.

A. Makaruk, M. Miltner, M. Harasek, Biogas desulfurization and biogas upgrading using a hybrid membrane system modeling study, Water Science & Technology 67.2 (2013) 326–332.

W. J. Schell, C. D. Houston, Use of Membranes for Biogas Treatment, Energy Progress 3 (1983) 96–100.

S. A. Stern, B. Krishnakumar, S. G. Charati, W. S. Amato, A. A. Friedman, D. J. Fuess, Performance of a bench-scale membrane pilot plant for the upgrading of biogas in a wastewater treatment plant, Journal of Membrane Science 151 (1) (1998) 63 – 74, ISSN 0376-7388.

A. Makaruk, M. Miltner, M. Harasek, Membrane biogas upgrading processes for the production of natural gas substitute, Separation and Purification Technology 74 (1) (2010) 83 – 92, ISSN 1383-5866.

T. Mayer, A. Makaruk, N. Diaz, K. Bosch, M. Miltner, M. Harasek, H. Hofbauer, Efficient Biomass Utilization by Polygeneration Processes- Production of Hydrogen, Electricity and Heat, in: Proceedings of ICPS 10 - International Conference on Polygeneration Strategies, Leipzig, 2010.

A. Makaruk, M. Miltner, M. Harasek, Membrane gas permeation in the upgrading of renewable hydrogen from biomass steam gasification gases, Applied Thermal Engineering 43 (0) (2012) 134 – 140, ISSN 1359-4311.

A. Makaruk, M. Harasek, Numerical algorithm for modelling multicomponent multipermeator systems, Journal of Membrane Science 344 (1-2) (2009) 258 – 265, ISSN 0376-7388.

S. A. Stern, J. E. Perrin, E. J. Naimon, Recycle and multimembrane permeators for gas separations, Journal of Membrane Science 20 (1) (1984) 25 – 43, ISSN 0376-7388.

J. G. Wijmans, R. W. Baker, The solution-diffusion model: a review, Journal of Membrane Science 107 (1-2) (1995) 1 – 21, ISSN 0376-7388.

G. Kitzhofer, G. Pulverer, O. Koch, C. Simon, E. Weinmüller, The new Matlab code `bvpsuite` for the solution of singular implicit BVPs, J. Numer. Anal. Indust. Appl. Math. 5 (2010) 113–134.

C. de Boor, B. Swartz, Collocation at Gaussian points, SIAM J. Numer. Anal. 10 (1973) 582–606.

G. Pulverer, G. Söderlind, E. Weinmüller, Automatic grid control in adaptive BVP solvers, Numer. Algorithms 56 (2011) 61–92.

O. Koch, R. März, D. Praetorius, E. Weinmüller, Collocation methods for index-1 DAEs with a singularity of the first kind, Math. Comp.

79 (2010) 281–304.
G. Kitzhofer, O. Koch, E. Weinmüller, Pathfollowing for essentially singular boundary value problems with application to the complex Ginzburg–Landau equation, BIT 49 (2009) 217–245.

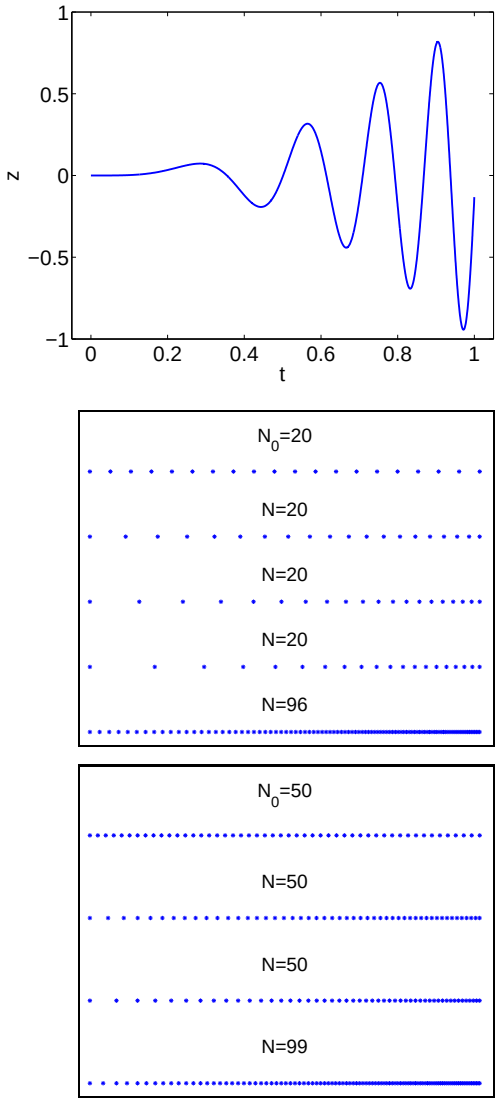

Figure 6: Exact solution (upper graph) and the grid adaptation (lower graphs) of `bvpsuite` with collocation of order $m = 4$, $TOL = 10^{-6}$.

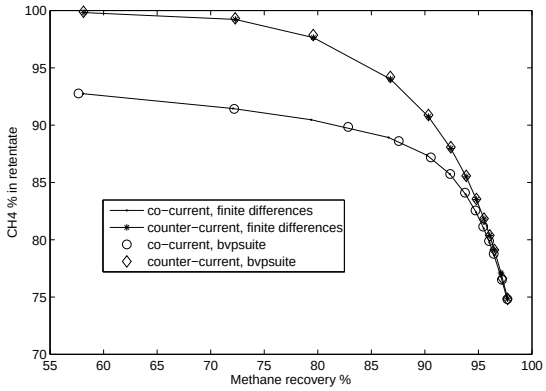

Figure 7: Methane concentration obtained from numerical simulation plotted versus methane recovery.

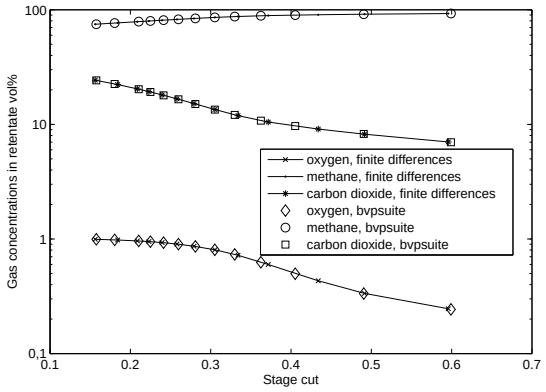

Figure 8: Gas concentration obtained from numerical simulation plotted versus stage cut for the co-current flow.

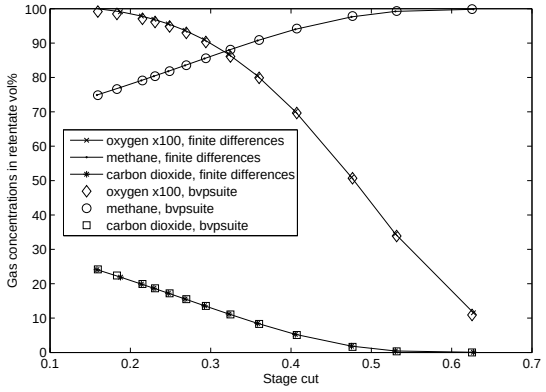

Figure 9: Gas concentration obtained from numerical simulation plotted versus stage cut for the counter-current flow.

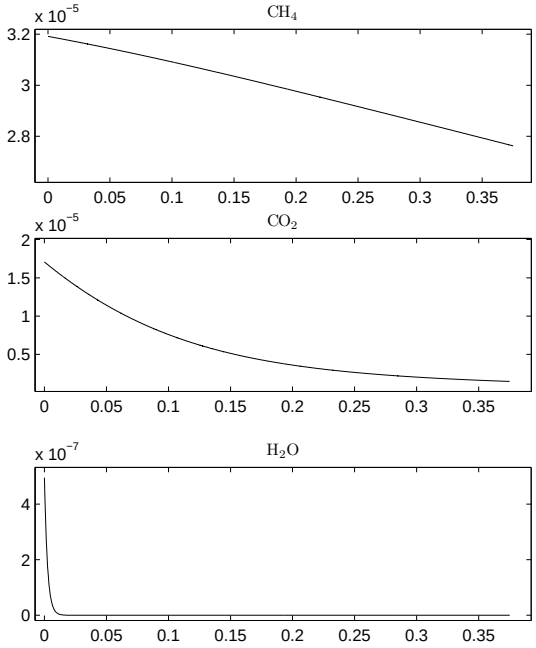

Figure 10: Gas partial pressures along the membrane for the experiment C.