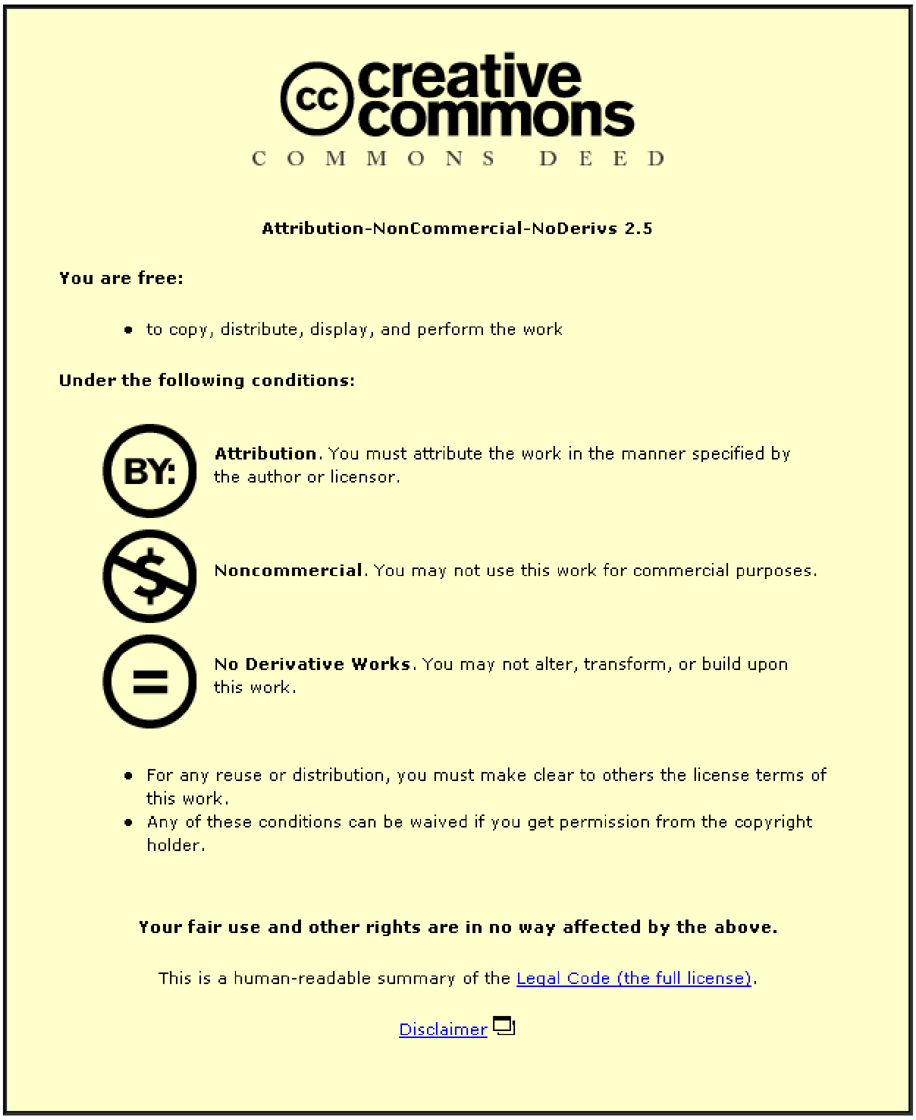

DYNAMIC SIMULATION OF HYBRID DIFFERENTIAL ALGEBRAIC SYSTEMS USING GPROMS: CASE STUDY IN EMULSION POLYMERIZATION

PAULEN RADOSLAV[a], BENYAHIA BRAHIM[b], LATIFI ABDERRAZAK[c], FIKAR MIROSLAV[a]
[a] FCFT, STU in Bratislava, Slovakia, e-mail: {radoslav.paulen, miroslav.fikar}@stuba.sk
[b] Department of Chemical Engineering, MIT, Cambridge, USA, e-mail: b.benyahia@mit.edu
[c] LRPG, CNRS–ENSIC–INPL, Nancy, France, e-mail: abderrazak.latifi@ensic.inpl-nancy.fr

Abstract: This paper is deals with dynamic simulation of a semi-batch polymerization reactor for emulsion copolymerization of styrene and butyl acrylate in the presence of n-dodecyl mercaptan as chain transfer agent (CTA). Mathematical model of the polymerization reaction involved is used to predict the glass transition temperature of produced polymer, global monomer conversion, the number and weight average molecular weights, the particle size distribution and the amount of residual monomers. This model, a set of hybrid differential algebraic equations (DAEs), is implemented into gPROMS environment for modeling, simulation, and optimization.

Keywords: Dynamic Simulation, Hybrid Systems, Differential Algebraic Systems, Emulsion Polymerization

1 INTRODUCTION

We witness an enormous rise of computational power of modern computer workstations. This is supported by improved computer hardware performance and by exploiting advanced techniques such as parallel and distributed computing. On the other hand, in scientific and engineering computing, more and more new efficient and competitive software packages are continuously developed. This enables scientists and engineers to use advanced computation techniques and to study more complex phenomena and systems.

In chemical and process systems engineering, such a progress is always welcome by those who deal with problems like equilibrium modeling and/or its characterization for chemical compounds and reactions, computational fluid dynamics, mathematical modeling, simulation, process design, parameter estimation, mixed-integer and/or non-linear optimization of complex systems. These complex systems include hybrid systems, stiff and differential algebraic systems, and parameter-distributed and multi-scale systems. In this paper we look closer at the problem of dynamic simulation of process which model is described by a set of hybrid differential algebraic equations (DAEs). More concretely, we aim to implement and discuss the implementation of such models into gPROMS environment developed by Barton and Pantelides [1993]. As a case study of hybrid DAE system we choose an emulsion copolymerization reaction taking place in fed-batch reactor.

Emulsion polymerization is an important industrial process used to produce a great variety of polymers of multiple uses (e.g. paints, adhesives, coatings, varnishes, etc.). Moreover, it has significant advantages over bulk and solution polymerization processes. These advantages result mostly from the multiphase and compartmentalized nature of the emulsion polymerization which allows for production of polymers of high molecular weights at high polymerization rates at the same time, delivering a high versatility to product qualities. Another distinct advantage of employing emulsion type of polymerization process is a very good controllability of process (reactor) temperature and thus very good controllability of the overall process. This is again caused by the multiphase nature of the process and by the excellent heat conducting properties of water being typically present in relatively large amount in the reactor.

However, any emulsion polymerization reactor represents a rather complex system both from modeling and simulation point of view. This complexity, which arises from the presence of non-linear and discontinuous behavior and is typically supported by a presence of algebraic equations, which need to be satisfied at all time points of dynamic simulation, makes the reliable simulation of polymerization reactions very challenging task. This paper deals with simulation of a tendency model for the emulsion copolymerization of styrene and butyl acrylate in the presence of n-dodecyl mercaptan as chain transfer agent. The objective for the model is to predict the global conversion, the number and weight average molecular weights, the average diameter of polymer particles and the residual amounts of monomers.

In the next sections, we give description of hybrid DAE systems, we provide main characteristics of the process model, we deal with issues connected to implementation of process of considered polymerization reaction into gPROMS environment and, finally, we present and discuss obtained results.

2 MODELS OF DYNAMIC SYSTEMS

The aim of this section is to present basic features of different types of mathematical models which find their applications in representation of properties (characteristics) of dynamic systems and are be used for simulation of processes behavior.

2.1 Models Based on Ordinary Differential Equations

Ordinary differential equations (ODEs) as well as their analysis and solution methods (analytical or numerical) represent a cornerstone in study of dynamics of any process. Dynamic model represented by a set of ODEs may be written as follows

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \forall t \in [t_0, t_\mathrm{f}], \tag{1}$$

where dot over a variable stands for its time derivative, $\boldsymbol{x}$ denotes vector of n_x state variables, which represent inner dynamical properties of the studied system, and $\boldsymbol{u}$ is vector of n_u control variables, the external dynamic actions which influence process behavior. Vector function $\boldsymbol{f}(\cdot)$ is then such that $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. Variable t denotes independent variable, time, and t_0 and t_f are time instants in which we commence and terminate observation of the process, respectively. There are only few cases, special forms of Eq. (1), in which we are able to deduce a solution to this set ODEs, $\boldsymbol{x}(t)$, $\forall t \in [t_0, t_\mathrm{f}]$. In general, however, we have to use numerical techniques in order to find solution which additionally satisfies initial conditions $\boldsymbol{x}_0$ such that

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{2}$$

Equations (1) and (2) form together an initial value problem (IVP). To resolve IVPs, one has many possibilities to choose between various free or commercial software packages which implement a great variety of methods (such as explicit or implicit Euler, Runge-Kutta, backward differentiation formulae, Adams-Moulton method, ...) and are written in various programming languages including C, C++, Python, FORTRAN, MATLAB, etc.

2.2 Models Based on Differential Algebraic Equations

A set of differential algebraic equations is represented by following mathematical form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), \qquad \forall t \in [t_0, t_\mathrm{f}], \tag{3a}$$

$$0 = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), \qquad \forall t \in [t_0, t_\mathrm{f}], \tag{3b}$$

where $\boldsymbol{y}(t)$ are n_y so-called algebraic variables which means that they are time-dependent but their differential form (time derivative) is not present in Eqs. (3). Set of n_e equations, given by the functional form of $\boldsymbol{g}(\cdot)$, which is such that $\boldsymbol{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_e}$, has to be satisfied for all time instants between t_0 and t_f. A classification of DAEs is usually made using an *index* concept [Brenan et al., 1989]. This index represents a number of consecutive differentiations which are needed to be performed in order to transform the set of algebraic equations into differential form.

Again, there are various methods and many software packages capable of providing solution to DAEs. These include FORTRAN package DASSL written by Petzold [1982], C++ packages DASOLV by Jarvis and Pantelides [1992] and SUNDIALS by Hindmarsh et al. [2005], and so on.

2.3 Hybrid Differential Algebraic Models

Hybrid process models keep both character of continuous and discrete process models. They are composed of the set of modes and transition rules and transition functions between these modes. This stands for discrete features of this type of process models. Each of this modes is, however, represented by continuous dynamics of the process itself. Here, we adopt notation introduced by [Feehery, 1998]. Dealing with the hybrid process model means dealing with system of a form

$$\text{mode } S_i\text{:}\quad \dot{\boldsymbol{x}}^{(i)} = \boldsymbol{f}^{(i)}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{u}^{(i)}), \qquad \forall t \in [t_0^{(i)}, t_\mathrm{f}^{(i)}],\ S_i \in \bigcup_{k=1}^{n_k} S_k, \tag{4}$$

$$0 = \boldsymbol{g}^{(i)}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{u}^{(i)}), \qquad \forall t \in [t_0^{(i)}, t_\mathrm{f}^{(i)}],\ S_i \in \bigcup_{k=1}^{n_k} S_k, \tag{5}$$

where $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n_x^{(i)}}$, $\boldsymbol{y}^{(i)} \in \mathbb{R}^{n_y^{(i)}}$, $\boldsymbol{u}^{(i)} \in \mathbb{R}^{n_u^{(i)}}$, and n_k denotes number of modes. Function $\boldsymbol{f}^{(i)}$ is such that $\boldsymbol{f}^{(i)} : \mathbb{R}^{n_x^{(i)}} \times \mathbb{R}^{n_y^{(i)}} \times \mathbb{R}^{n_u^{(i)}} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x^{(i)}}$. Switching from one stage (mode S_k) to another (mode S_j) occurs when a unique transition condition $L_j^{(k)}(\dot{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{u}^{(k)}, t) = 0,\ S_j \in P^{(k)}$ is satisfied. Set $P^{(k)}$ contains all possible successive stages of mode S_k. Transition function $T_j^{(k)}(\dot{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{u}^{(k)}, \dot{\boldsymbol{x}}^{(j)}, \boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)}, \boldsymbol{u}^{(j)}, t)$ determines possible discontinuity in variables participating in model. There is a special case of transition function defined at initial mode and initial time $t = t_0^{(k)}$ where $T_k^{(0)}(\dot{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{u}^{(k)}, t_0) = 0 \Rightarrow \boldsymbol{x}^{(k)}(t_0^{(k)}) = \boldsymbol{x}_0^{(k)}$.

Solving systems of differential algebraic equations in the presence of changing dynamic and possible discontinuities is a difficult computational task even for small-scale systems. Package gPROMS stands for modeling, simulation and optimization environment. It contains DASOLV package for treating of hybrid DAE problems. We aim to use this package as a tool for dynamic simulation of emulsion copolymerization reaction taking place in fed-batch reactor.

3 EMULSION POLYMERIZATION MODEL

This section introduces some basic features of the model of fed-batch emulsion polymerization reactor where copolymerization reaction takes place between styrene and butyl-acrylate in the presence of chain transfer agent (CTA). We present differential equations which model the material balance of the system together with main features which contribute to complexity of the model (switching behavior and algebraic equations), namely surfactant partition, glass and gel effects, and species partition. Rest of the section explain algebraic relations using which we can evaluate properties of the produced polymer.

Emulsion polymerization is driven by radical mechanisms, where the monomers are mainly located in droplets dispersed in an aqueous phase. These droplets are stabilized by a surfactant. The initiator is soluble in the water phase which contains an excess of surfactant mainly in its micellar form. The initiator decomposes in the aqueous phase and generates primary radicals. The establishment of the model and its experimental validation were done by Benyahia et al. [2010].

3.1 Kinetic Scheme

As mentioned above, emulsion polymerization involves mostly a radical reactions. These are present in both in aqueous and organic (droplets) phases.

- Aqueous phase

$$
\begin{array}{rrcl}
\text{Initiation} & I_2 & \overset{k_d}{\rightarrow} & 2R^{\bullet}_{aq} \\
\text{Inhibition} & R^{\bullet}_{aq} + Z_{aq} & \overset{k_{Zaq}}{\rightarrow} & P + Z^{\bullet}_{aq} \\
\text{Nucleation} & R^{\bullet}_{aq} + \text{micelle} & \overset{k_N}{\rightarrow} & \text{particle} + R^{\bullet} \\
\text{Radical absorption} & R^{\bullet}_{aq} & \overset{k_{cp}}{\rightarrow} & R^{\bullet}
\end{array}
$$

where I_2 represents a molecule of initiator, k_d is initiator decomposition rate constant, $R^{\bullet}_{aq}$ denotes free radical in aqueous phase, Z_{aq} is molecule of inhibitor in aqueous phase, P represents a molecule of polymer, k_{Zaq} stands for inhibition rate constant, $R^{\bullet}$ denotes free radical in organic phase (at his level, we do not need to distinguish between radicals ended by ith or jth monomer), k_N is nucleation rate constant, and k_{cp} stands for capture rate constant.

- Organic phase (particles)

$$
\begin{array}{rrcl}
\text{Propagation} & R^{\bullet}_i + M_j & \overset{k_{pij}}{\rightarrow} & R^{\bullet}_j \\
\text{Termination by combination} & R^{\bullet}_i + R^{\bullet}_j & \overset{k_{tcij}}{\rightarrow} & P \\
\text{Termination by disproportionation} & R^{\bullet}_i + R^{\bullet}_j & \overset{k_{tdij}}{\rightarrow} & 2P \\
\text{Inhibition} & R^{\bullet}_i + Z_p & \overset{k_{Zp}}{\rightarrow} & P + Z^{\bullet}_p \\
\text{Transfer to monomers} & R^{\bullet}_i + M_j & \overset{k_{trmij}}{\rightarrow} & P + R^{\bullet}_j \\
\text{Transfer to transfer agent} & R^{\bullet}_i + CTA_p & \overset{k_{CTAp}}{\rightarrow} & P + CTA^{\bullet}_p \\
\text{Radical desorption} & R^{\bullet} & \overset{k_{des}}{\rightarrow} & R^{\bullet}_{aq}
\end{array}
$$

where M_j represents jth monomer, k_{pij} is propagation rate coefficient between the jth monomer and a free radical ended by the ith monomer, k_{tcij} stands for termination rate constant by combination of free radicals ended by ith and jth monomer, k_{tdij} represents termination rate constant by disproportionation of free radicals ended by ith and jth monomer, Z_p is molecule of inhibitor in organic phase and $Z^{\bullet}_p$ its respective radical, k_{Zp} stands for inhibition rate constant in organic phase, k_{trmij} represents transfer rate constant between the jth monomer and the growing radical ended by ith monomer, CTA_p is molecule of the transfer agent in the particles and $CTA^{\bullet}_p$ stands for its respective radical, k_{CTAp} denotes transfer agent rate constant in particles, and k_{des} desorption rate constant.

3.2 Material Balance

The material balances are presented in a general form for semi-batch process using the reaction rates defined in further details in Benyahia et al. [2010]. We note that these equations could be easily simplified for the case of a batch process.

$$\frac{\mathrm{d}V_R}{\mathrm{d}t} = Q_f + Q_{If} + \sum_{i=1,2} \left(\frac{1}{\rho_{pi}} - \frac{1}{\rho_i} \right) M_M^i (\mathcal{R}_{pi} + \mathcal{R}_{trmi}), \tag{6a}$$

$$\frac{\mathrm{d}M_i}{\mathrm{d}t} = -\mathcal{R}_{pi} - \mathcal{R}_{trmi} + Q_f [M_i]_f, \qquad \forall i = 1,2 \tag{6b}$$

$$\frac{\mathrm{d}M_{Ti}}{\mathrm{d}t} = Q_f [M_i]_f, \qquad \forall i = 1,2 \tag{6c}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = -\mathcal{R}_d + Q_{If}[I]_f, \tag{6d}$$

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = -(\mathcal{R}_{Zp1} + \mathcal{R}_{Zp2}) + Q_f [Z]_f, \tag{6e}$$

$$\frac{\mathrm{d}CTA}{\mathrm{d}t} = -\mathcal{R}_{CTAp1} - \mathcal{R}_{CTAp2} + Q_f [CTA]_f, \tag{6f}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = Q_f [S]_f, \tag{6g}$$

$$\frac{\mathrm{d}N_p}{\mathrm{d}t} = \mathcal{R}_N, \tag{6h}$$

$$\begin{aligned} \frac{\mathrm{d}R_{p1}}{\mathrm{d}t} &= (\mathcal{R}_N + \mathcal{R}_{cp}) f_{aq1} - \mathcal{R}_{p12} + \mathcal{R}_{p21} - \mathcal{R}_{trm12} \\ &+ \mathcal{R}_{trm21} - \mathcal{R}_{Zp1} - \mathcal{R}_{des1} - (\mathcal{R}_{T11} + \mathcal{R}_{T12}), \end{aligned} \tag{6i}$$

$$\begin{aligned} \frac{\mathrm{d}R_{p2}}{\mathrm{d}t} &= (\mathcal{R}_N + \mathcal{R}_{cp}) f_{aq2} - \mathcal{R}_{p21} + \mathcal{R}_{p12} - \mathcal{R}_{trm21} \\ &+ \mathcal{R}_{trm12} - \mathcal{R}_{Zp2} - \mathcal{R}_{des2} - (\mathcal{R}_{T22} + \mathcal{R}_{T21}), \end{aligned} \tag{6j}$$

$$\frac{\mathrm{d}(N_p \tilde{n})}{\mathrm{d}t} = 2\mathcal{R}_{cp}\bar{n} - \left(\frac{2\tilde{\tilde{n}}}{\tilde{n}} + 1 \right) \mathcal{R}_T - 2\frac{\tilde{n}}{\bar{n}} (\mathcal{R}_{Zp} + \mathcal{R}_{des}), \tag{6k}$$

$$\frac{\mathrm{d}(N_p \bar{n} \lambda_1)}{\mathrm{d}t} = \mathcal{R}_N + \mathcal{R}_{cp} - \mathcal{R}_{des} + \mathcal{R}_p - (\mathcal{R}_{Zp} + \mathcal{R}_T)\lambda_1, \tag{6l}$$

$$\begin{aligned} \frac{\mathrm{d}(N_p \bar{n} \lambda_2)}{\mathrm{d}t} &= \mathcal{R}_N + \mathcal{R}_{cp} - \mathcal{R}_{des} + \mathcal{R}_p (1 + 2\lambda_1) \\ &+ (\mathcal{R}_{trm} + \mathcal{R}_{CTAp})(1 - \lambda_2) - (\mathcal{R}_{Zp} + \mathcal{R}_T)\lambda_2, \end{aligned} \tag{6m}$$

$$\frac{\mathrm{d}N_m}{\mathrm{d}t} = \mathcal{R}_{Zp} + \mathcal{R}_{trm} + \mathcal{R}_{TD} + \mathcal{R}_{CTAp} + \frac{\mathcal{R}_{TC}}{2}, \tag{6n}$$

$$\frac{\mathrm{d}(N_m L_1)}{\mathrm{d}t} = \lambda_1 (\mathcal{R}_{Zp} + \mathcal{R}_{trm} + \mathcal{R}_{TD} + \mathcal{R}_{CTAp} + \mathcal{R}_{TC}), \tag{6o}$$

$$\frac{\mathrm{d}(N_m L_2)}{\mathrm{d}t} = \lambda_2 (\mathcal{R}_{Zp} + \mathcal{R}_{trm} + \mathcal{R}_{TD} + \mathcal{R}_{CTAp}) + \mathcal{R}_{TC}(\lambda_2 + \lambda_1^2), \tag{6p}$$

where V_R stands for the total volume in the reactor, Q_f is the feed rate of pre-emulsion (monomers, inhibitor, CTA and surfactant), Q_{If} represents the feed rate of initiator, ρ_i and ρ_{pi} denote the density of the ith monomer and the density of the homopolymer of ith monomer, respectively, M_M^i is molecular weight of the ith monomer, $\mathcal{R}_{pi}$ and $\mathcal{R}_{trmi}$ stand for propagation rate of the ith monomer and the transfer to monomer rate for the ith monomer, respectively, M_i represents number of moles of the ith monomer, $[M_i]_f$, $[I]_f$,$[Z]_f$, $[S]_f$, $[CTA]_f$ are the concentrations of ith monomer, initiator, inhibitor, surfactant and chain transfer agent in the feed, respectively, M_{Ti} is the total amount in moles of the ith monomer introduced into the reactor throughout the reaction, I, Z, CTA, and S stand for number of moles of initiator, inhibitor, chain transfer agent and surfactant, respectively, initiator decomposition rate is denoted by $\mathcal{R}_d$, $\mathcal{R}_{Zpi}$ represents inhibitor consumption rate with the ith monomer, $\mathcal{R}_{CTApi}$ is transfer agent consumption rate by the ith monomer, $\mathcal{R}_N$ stands for the micellar nucleation rate, R_{pi} is the total number of moles of radicals ended by the ith monomer in the particles, $\mathcal{R}_{cp}$ is radical capture rate, $\mathcal{R}_{des}$ ($\mathcal{R}_{desi}$) denotes (the ith monomer) desorption rate, f_{aqi} stands for the fraction of the ith radical in the aqueous phase, R_{pij} is the propagation rate of the jth monomer with the ith radical, R_{trmij} transfer to monomer rate (radicals ended by the ith monomer to the jth monomer), $\mathcal{R}_{Tij}$ represents the termination rate between a radical ended by the ith monomer and the jth radical, λ_i stands for normalized ith moments of the macroradicals, N_p is total number of moles of particles, N_m represents total number of macromolecules, $\bar{n}$ stands for average number of radicals per particle, $\tilde{n}$ stands for the average number of pairs of radicals in a particle.

These equations represent dynamic part (represented by differential equations) of the model of the polymerization process in semi-batch reactor. A set of initial conditions reads as follows

$$V_{R0} = V_{w0} + \frac{m_{10}}{\rho_1} + \frac{m_{20}}{\rho_2} + \frac{m_{S0}}{\rho_S}, \tag{7a}$$

$$M_{i0} = \frac{m_{i0}}{M_M^i}, \qquad \forall i = 1, 2 \tag{7b}$$

$$M_{Ti0} = \frac{m_{i0}}{M_M^i}, \qquad \forall i = 1, 2 \tag{7c}$$

$$I_0 = \frac{m_{I0}}{M_M^I}, \tag{7d}$$

$$Z_0 = \left(\frac{m_{10}}{M_M^1} + \frac{m_{20}}{M_M^2} \right) 15 \times 10^{-6}, \tag{7e}$$

$$CTA_0 = \frac{m_{CTA0}}{M_M^{CTA}}, \tag{7f}$$

$$S_0 = \frac{m_{S0}}{M_M^S}, \tag{7g}$$

$$N_{p0} = R_{p10} = R_{p20} = \tilde{n}_0 = \lambda_{10} = 0, \tag{7h}$$

$$\lambda_{20} = N_{m0} = L_{10} = L_{20} = 0. \tag{7i}$$

3.3 Surfactant Partition

The surfactant is distributed between the particles, droplets, aqueous phase and micelles. The total number of moles of surfactant in the reactor is given by

$$S = S_{aq} + S_{mic} + S_p + S_d = S_{aq}^* + S_p + S_d, \tag{8}$$

where S_p, S_{mic}, S_d, and S_{aq} are the number of moles of surfactant in the particles, in the micelles, in the droplets, and dissolved in the aqueous phase respectively

$$S_{mic} = N_{mic} n_s, \quad S_p = \frac{6V_p}{d_p a_s}, \quad S_d = \frac{6V_d}{d_d a_s}, \tag{9}$$

where N_{mic} is the total number of moles of micelles, n_s denotes the number of surfactant molecules per micelle, a_s represents the surface covered by one mole of surfactant, d_d is the average droplets diameter.

The micelles disappear once the concentration of surfactant in the aqueous phase go down the critical micelle concentration. Thus

if $S_{aq}^* < \text{CMC}\, V_{aq}$

$$N_{mic} = 0, \tag{10a}$$

$$S_{aq} = S_{aq}^*, \tag{10b}$$

else

$$S_{aq} = \text{CMC}\, V_{aq}, \tag{11a}$$

$$N_{mic} = \frac{S - (S_d + S_p + \text{CMC}\, V_{aq})}{n_s}, \tag{11b}$$

where CMC is critical micelle concentration. We may observe that Eq. (8) represents an algebraic equation(s) which has to be satisfied for all time instants of the process while Eqs. (10) and (11) contribute to hybrid behavior of the model itself.

3.4 Glass and Gel Effects

According to the fraction of the polymer in the particles, the glass and the gel effects equations used for propagation and termination rate coefficients are given as follows Nomura et al. [1994]

$$k_{pij} = \begin{cases} k_{pij}^0 & \text{if } W_p \leq 0.7, \\ k_{pij}^0 \exp\left[-a_{ij}^{Gl}(W_p - 0.7)\right] & \text{if } W_p > 0.7, \end{cases} \tag{12}$$

$$k_{Tij} = \begin{cases} k_{Tij}^0 & \text{if } W_p \leq 0.32, \\ k_{Tij}^0 \exp\left[-b^{Ge}(W_p - 0.32)\right] & \text{if } 0.32 < W_p \leq 0.8, \\ k_{Tij}^0 \exp\left[-b^{Ge}(0.8 - 0.32) - b^{Gl}(W_p - 0.8)\right] & \text{if } W_p > 0.8, \end{cases} \tag{13}$$

where a_{ij}^{Gl} is the glass coefficient of propagation reaction of jth monomer with ith radical, b^{Gl} and b^{Ge} are, respectively, gel and glass coefficients of termination reaction respectively and W_p is the mass fraction of polymer in the particles defined by

$$W_p = \frac{\sum\limits_{i=1,2} (M_{Ti} - M_i - R_{aq} f_{aqi}) M_M^i}{\sum\limits_{i=1,2} (M_{pi} + M_{Ti} - M_i - R_{aq} f_{aqi}) M_M^i}, \tag{14}$$

where M_{Ti} is the total number of moles of ith monomer put in the reactor at time t, M_i is the total number of moles of ith monomer left in the reactor at time t, f_{aqi} the molar fraction of ith monomer in the aqueous phase, and M_M^i is the molecular weight of the ith monomer. Again we observe that Eqs. (12) and (13) directly imply hybrid characteristics of the studied model of the studied polymerization process.

3.5 Species Partition

Finally, equations presented in this section show another source of high complexity of the process model since they stand for algebraic equations which should be satisfied along the simulation horizon but their structure can be changed when certain switching conditions are satisfied.

The partition of the different species between the aqueous phase, particles and droplets is needed for evaluation of reaction rates. This can be done using the method of the partition coefficients under the thermodynamic equilibrium developed by Gugliotta et al. [1995]. Lets consider the total volume engaged in the reactor, V_R, expressed by means of the total volumes of each specie and different phases:

$$\begin{aligned} V_R &= V_1 + V_2 + V_Z + V_{CTA} + V_{pol} + V_{aq}, \\ &= V_{aq} + V_p + V_d, \end{aligned} \tag{15}$$

where V_1, V_2, V_Z, V_Z are the total volumes of the first monomer, the second monomer, inhibitor and CTA in the reactor respectively and V_{pol} is the polymer volume. Particles volume and the polymer volume are related to each other by

$$V_p = \left(\frac{\sigma}{\sigma - 1}\right) V_{pol}, \tag{16}$$

where σ is a coefficient related to the saturation degree of the particle and V_{pol} represents the total polymer volume (in the particles) such that

$$V_{pol} = \sum_{i=1,2} (M_{Ti} - M_i - R_{aq} f_{aqi}) \frac{M_M^i}{\rho_{pi}} \tag{17}$$

Since the solubility of each specie in the aqueous phase is very low, the volume of the aqueous phase is equivalent to the water volume (V_w),

$$V_{aq} \cong V_w. \tag{18}$$

Hence, from equations (15) and (16) we get the droplets volume

$$V_d = V_1 + V_2 + V_Z + V_{CTA} - \frac{V_{pol}}{1-\sigma}. \tag{19}$$

When the droplets disappear, equations (15) and (16) lead to the expression for the particles volume V_p

$$V_p = V_1 + V_2 + V_Z + V_{CTA} + V_{pol}. \tag{20}$$

Finally, we can summarize the volume partition problem as follows

if the droplets are available

$$V_d = V_1 + V_2 + V_Z + V_{CTA} - \frac{V_{pol}}{1-\sigma}, \tag{21a}$$

$$V_p = \left(\frac{\sigma}{\sigma-1}\right) V_{pol}, \tag{21b}$$

otherwise

$$V_d = 0, \tag{22a}$$

$$V_p = V_1 + V_2 + V_Z + V_{CTA} + V_{pol}. \tag{22b}$$

It should be emphasized that in all cases the aqueous volume is determined from the volume balance

$$V_{aq} = V_R - V_d - V_p. \tag{23}$$

3.6 Evaluation of Polymer Properties

3.6.1 Molecular Weight Distribution

The physical and mechanical properties of polymers depend strongly on their MWDs. The number and weight average molecular weights, $\bar{M}_n$ and $\bar{M}_n$, can be then easily calculated using the following equations

$$\bar{M}_n = \bar{M} L_1, \tag{24}$$

$$\bar{M}_w = \bar{M} \frac{L_2}{L_1}, \tag{25}$$

where $\bar{M}$ is the average molecular weight of the monomeric unit given by

$$\bar{M} = \frac{\sum\limits_{i=1,2} (M_{Ti} - M_i - R_{aq} f_{aqi}) M_M^i}{\sum\limits_{i=1,2} (M_{Ti} - M_i - R_{aq} f_{aqi})}. \tag{26}$$

3.6.2 Glass Transition Temperature

The control of the product quality during emulsion polymerization requires the monitoring of a large set of parameters related to end-use properties. Some of these properties are often described through distributed characteristics such as MWD and/or the copolymer composition distribution (CCD). The latest is the case of the glass transition temperature, T_g, which depends strongly on the microstructure of the macromolecules, particularly for copolymers. To complete the model, the glass transition temperature is evaluated according to Fox's equation [Fox and Flory, 1950].

$$\left[T_g + \frac{a}{\bar{M}_n}\right]^{-1} = \sum_{i=1,2} \frac{W_i}{T_{gi}}, \tag{27}$$

where a is an adjustment parameter (Fox and Flory constant), T_{gi} is the glass transition temperature of the homopolymer of ith monomer and W_i stands for the mass fractions of ith monomer in the copolymer defined by

$$W_i = \frac{(M_{Ti} - M_i - R_{aq} f_{aqi}) M_M^i}{\sum\limits_{i=1,2} (M_{Ti} - M_i - R_{aq} f_{aqi}) M_M^i}. \tag{28}$$

Another possibility is to use so-called instantaneous glass transition temperature, T_g^{ins}, defined by

$$\frac{1}{T_g^{\text{ins}}} = \sum_{i=1,2} \frac{W_i^{\text{ins}}}{T_{gi}}, \tag{29}$$

where W_i^{ins} is the instantaneous mass fraction of the ith monomer in copolymer such as

$$W_i^{\text{ins}} = \frac{R_{pi} M_M^i}{R_{p2} M_M^2 + R_{p2} M_M^2}. \tag{30}$$

3.6.3 Conversion and Copolymer Composition

The global mass conversion is given by

$$X = \frac{\sum\limits_{i=1,2} (M_{Ti} - M_i) M_M^i}{\sum\limits_{i=1,2} M_{Ti} M_M^i}. \tag{31}$$

The residual mass fraction of ith monomer is defined as

$$F_i = \frac{M_i M_M^i}{\sum\limits_{i=1,2} M_i M_M^i}. \tag{32}$$

3.7 Discussion

Hence that, presented model of copolymerization process involves 18 ordinary differential equations. It involves algebraic equations (surfactant and volume partition equations) as well. Moreover, eqs. (10) to (13), (21) and (22) show hybrid (switching) nature of this model. Finally, we can state that the copolymerization process is modeled as a set of hybrid differential algebraic equations (DAEs). These properties make dynamic simulation of this process very challenging task since this activity requires numerical integration and continuous resolution of stiff system of model equations. To cope with the stiffness of the process model we use normalization of state variables such that normalized ith

Table 1 – Values of normalization constants.

norm$_i$	state	value
10	N_p	1×10^{-9}
11	R_{p1}	1×10^{-9}
12	R_{p2}	1×10^{-10}
13	$\tilde{n}$	1×10^{-5}
14	λ_1	1×10^{-7}
15	λ_2	1×10^{-3}
16	N_m	1×10^{-6}
17	L_1	1×10^{-3}
18	L_2	10

state variable $\tilde{x}_i = x_i/\text{norm}_i$, where x_i represents former state variable and norm$_i$ means normalization constant. We use initial conditions values as the normalization constants for the states described by equations (6a) – (6g). Table 1 gives overview of the normalization constants used for the rest of state variables.

The resulting process model is implemented into gPROMS environment. Next section introduces basic features of gPROMS environment and discusses implementation issues and the ways of avoiding them.

4 SIMULATION IN GPROMS ENVIRONMENT

gPROMS is modeling, simulation and optimization standalone toolbox which is capable of performing large-scale dynamic simulations and optimizations of complex processes. Its features include solving of systems of DAEs, automatic root finding of switching functions, when considered process model is of hybrid nature, as well as automatic sensitivity functions generation and evaluation which can be fairly useful for the purposes of process optimization.

There are various *entities* provided by gPROMS which can be exploited in order to run simulation of a dynamic or steady-state system, to perform process optimization and optimal experiment design, to estimate parameters of the system from measured data and so on. Entities used for simulation include MODEL and PROCESS.

4.1 The MODEL Entity

In this entity we specify model equations. These are to be input into EQUATION environment, while values of time-invariant parameters involved in these equations are to be declared in PARAMETER environment and their values should be introduced in the SET environment. The time-dependent variables, previously denoted as $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}$, are to be declared in VARIABLE section of the PROCESS entity code.

Defining model equations can be, in many cases, done just by inputting process equations (1) or (3) as they are, using intrinsic functions. In case of hybrid process behavior, one can exploit IF or CASE conditionals to explicitly define the switching (hybrid) nature of the process. Provided that there are two modes between which the process is switching, the IF conditional statement can be used such that

```
    IF $L_2^{(1)}(\dot{\boldsymbol{x}}^{(1)}, \boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)}, t) = 0$
        $\dot{\boldsymbol{x}}^{(2)} = \boldsymbol{f}^{(2)}(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}, \boldsymbol{u}^{(2)})$
        $0 = \boldsymbol{g}^{(2)}(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}, \boldsymbol{u}^{(2)})$
ELSE
        $\dot{\boldsymbol{x}}^{(1)} = \boldsymbol{f}^{(1)}(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)})$
        $0 = \boldsymbol{g}^{(1)}(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)})$
 END
```

However, using the CASE statement, the switching between two modes can be may be formulated as

```
CASE OF mode
 WHEN 1st-mode : $\dot{\boldsymbol{x}}^{(1)} = \boldsymbol{f}^{(1)}(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)})$
                 $0 = \boldsymbol{g}^{(1)}(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)})$
 SWITCH TO 2nd-mode IF $L_2^{(1)}(\dot{\boldsymbol{x}}^{(1)}, \boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \boldsymbol{u}^{(1)}, t) = 0$
 WHEN 2nd-mode : $\dot{\boldsymbol{x}}^{(2)} = \boldsymbol{f}^{(2)}(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}, \boldsymbol{u}^{(2)})$
                 $0 = \boldsymbol{g}^{(2)}(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}, \boldsymbol{u}^{(2)})$
 SWITCH TO 1st-mode IF $L_1^{(2)}(\dot{\boldsymbol{x}}^{(2)}, \boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}, \boldsymbol{u}^{(2)}, t) = 0$
END
```

Here “mode” represents so-called SELECTOR variable type which is allowed to attain only two values, either “1st-mode” or “2nd-mode”. The advantage of using the latter formulation for describing the switching behavior is that one can directly and much more efficiently control (manage) the initialization procedure of simulation algorithm.

4.2 The PROCESS Entity

In this entity user specifies parameters of the simulation and any sort of additional information which is required/accepted by the simulation algorithm. Required information is represented by a set of initial conditions of the form (2). This can be done in the INITIAL environment. Optional, but potentially very useful information can be introduced with the regard of initialization of the values of algebraic variables and it can be done using PRESET environment. Another possibility is to introduce a REINITIALIZATION procedure which defines transition functions $T_j^{(k)}(\dot{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{u}^{(k)}, \dot{\boldsymbol{x}}^{(j)}, \boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)}, \boldsymbol{u}^{(j)}, t)$ at time of the switching. Within PROCESS entity, one can set all desired numerical SOLUTIONPARAMETERS such as relative and absolute tolerances, tolerances to the switching functions, and so on.

5 RESULTS AND DISCUSSION

We consider two types of simulation: simulation in batch mode ($Q_f = 0$) and simulation in fed-batch mode ($Q_f = Q_f(t)$). Table 2 shows composition of initial loads (for batch and fed-batch mode) and pre-emulsioned feed stream of polymerization reaction which we use in our calculations. These are the same as ones used in Benyahia et al. [2010] and Benyahia et al. [2011] for model validation and process optimization respectively. We consider to simulate 6000 seconds of batch process run.

Our implementation of the process model into gPROMS environment considers using CASE conditionals mainly to express switching behavior of the model and to assure non-negativity of time-dependent variables $\boldsymbol{x}$ and $\boldsymbol{y}$ such as concentrations, volumes and so on. Moreover we use PRESET environment to initialize all algebraic variables $\boldsymbol{y}$. We found that simulation is numerically more stable

Table 2 – Composition of the initial loads and the feed stream of the reactor for batch and fed-batch mode simulation.

	Batch simulation	Fed-batch simulation	
Component	Initial load [g]	Initial load [g]	Feed [g]
Water	570	171	399
Butyl acrylate	60	12	48
Styrene	60	12	48
Initiator	1	1	0
Surfactant	15	3	12
CTA	0.6	0.12	0.48

when initial guesses for algebraic variables are specified. To generate a consistent initialization values for them we implemented considered process model into FORTRAN and MATLAB. Unlike gPROMS, using these programming languages, model equations are treated sequentially and thus may provide additional information about algebraic variables/equations and model switching behavior.

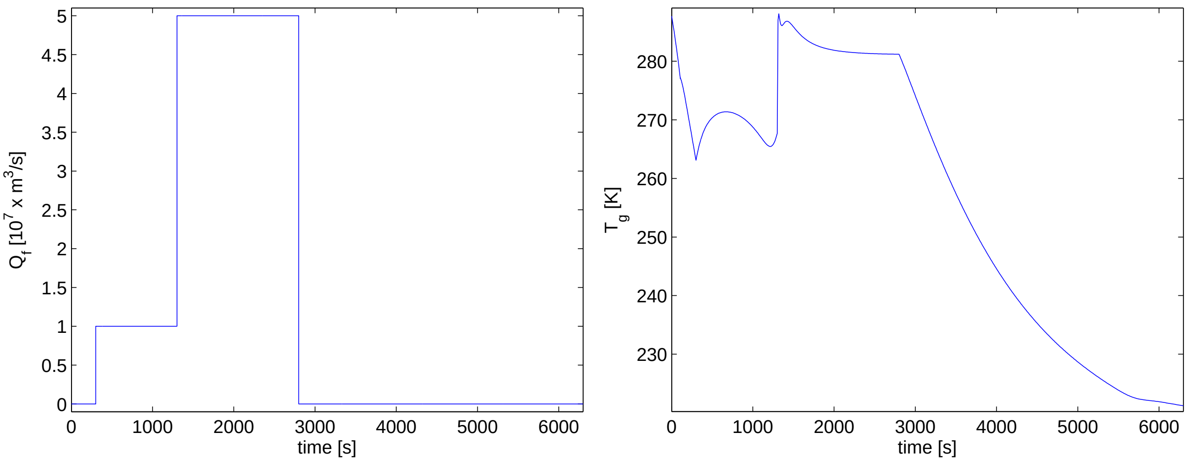

Figure 1 – Time-varying flowrate of the feed stream Q_f (left plot) and the corresponding trajectory of glass transition temperature T_g (right plot).

Figure 1 presents simulation results for fed-batch operation graphically. It shows considered time-dependent flowrate of the pre-emulsioned feed stream, Q_f and corresponding trajectory of glass transition temperature, T_g as the one of the qualitative properties which are to be predicted by the model. We may observe that numerical simulation solver is able to handle (and eventually recover from) discontinuities occurring during the simulation run.

As an extension of this work, we plan to numerically derive an optimal control of the process which will allow us at the same time produce polymer of desired quality and quantity. In well-established procedures, such as direct single shooting (control vector parametrization), it is very often that we need to repeatedly solve the system of process equations (i.e. simulate the process) with some actual guess of the control structure. That is why the computational time needed for simulation is one of the features of simulation we should have regard to. We compared computational time needed for simulation of batch and fed-batch alternative of the process.

Table 3 compares computational times of batch and fed-batch simulations in gPROMS (with DA-

Table 3 – Comparison of running times of the simulations for batch and fed-batch mode simulation using different simulation packages.

Simulation package	Batch simulation, CPU [s]	Fed-batch simulation, CPU [s]
gPROMS (C++)	0.78	0.62
DASSL (FORTRAN)	0.06	0.10
SUNDIALS (C++/MATLAB)	1.40	2.80
ode15s (MATLAB)	31.58	38.25

SOLV numerical integrator), DASSL (numerical integrator written in FORTRAN), SUNDIALS package (written in C++ with MATLAB interface available), and ode15s (MATLAB). Simulations compared therein were carried out using the initial conditions from Table 2, using normalization constants from Table 1 and using the same relative and absolute integration error tolerances.

Results indicate that MATLAB package ode15s, the one which is supposed to deal with stiff and DAE systems, took enormously large portion of time in comparison with other packages. Simulations using SUNDIALS package with its MATLAB interface turned out to be much more time-effective. FORTRAN DASSL package showed to be the fastest hybrid DAE simulator among others. Although gPROMS is not capable of the fastest simulation, it is still considerably fast for many applications. Moreover there are other features which make it very useful. One of these features is automatic sensitivity functions derivation and evaluation. Sensitivity information is essential when one considers to find optimal control of the process numerically since it is directly connected to the problem of evaluation of gradient information for time-varying variables.

6 CONCLUSIONS

In this work, a dynamic simulation of hybrid differential algebraic systems was studied. Our goal was to implement and discuss simulation of process model consisted of hybrid DAEs into gPROMS environment. We considered a case study in simulation of batch and fed-batch copolymerization reactor with reaction of styrene and butyl acrylate in the presence of chain transfer agent. Obtained results were discussed and some points were raised with regard to future work on dynamic optimization of this process. Although gPROMS environment showed to be not the most efficient simulator in comparison with other available packages it is still reliable hybrid DAE equation solver. If process optimization is considered, its additional features may outbalance this.

7 ACKNOWLEDGMENTS

The first and the fourth author acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11. We are also pleased to acknowledge the financial support of the Slovak Research and Development Agency for the mobility project SK-FR-0004-11.

References

BARTON, P. I.; PANTELIDES, C. C. 1993. gPROMS – a combined discrete/continuous modelling environment for chemical processing systems. *Simulation Series*, 25, 3, 25–34.

BENYAHIA, B.; LATIFI, M.; FONTEIX, C.; PLA, F. 2011. Multicriteria dynamic optimization of an emulsion copolymerization reactor. *Computers & Chemical Engineering*, 35, 12, 2886 – 2895. ISSN 0098-1354. doi:10.1016/j.compchemeng.2011.05.014.

BENYAHIA, B.; LATIFI, M.; FONTEIX, C.; PLA, F.; NACEF, S. 2010. Emulsion copolymerization of styrene and butyl acrylate in the presence of a chain transfer agent. part 1: Modelling and experimentation of batch and fedbatch processes. *Chemical Engineering Science*, 65, 2, 850–869. ISSN

0009-2509. doi:10.1016/j.ces.2009.09.036.

BRENAN, K. E.; CAMPBELL, S. E.; PETZOLD, L. R. 1989. *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*. New York: North-Holland.

FEEHERY, W. F. 1998. *Dynamic Optimization with Path Constraints*. Ph.D. thesis, MIT, Cambridge, USA.

FOX, T. G.; FLORY, P. J. 1950. Second-order transition temperatures and related properties of polystyrene. *Journal of Applied Physics*, 21, 581–591. doi:10.1063/1.1699711.

GUGLIOTTA, L. M.; ARZAMENDI, G.; ASUA, J. M. 1995. Choice of monomer partition model in mathematical modeling of emulsion copolymerization systems. *Journal of Applied Polymer Science*, 55, 7, 1017–1039. ISSN 1097-4628. doi:10.1002/app.1995.070550706.

HINDMARSH, A. C.; BROWN, P. N.; GRANT, K. E.; LEE, S. L.; SERBAN, R.; SHUMAKER, D. E.; WOODWARD, C. S. 2005. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31, 3, 363–396.

JARVIS, R. B.; PANTELIDES, C. C. 1992. DASOLV – a differential-algebraic equation solver. Technical report.

NOMURA, M.; SUZUKI, H.; TOKUNAGA, H.; FUJITA, K. 1994. Mass transfer effects in emulsion polymerization systems. I. diffusional behaviour of chain transfer agents in the emulsion polymerization of styrene. *Journal of Applied Polymer Science*, 51, 21–23.

PETZOLD, L. R. 1982. A description of DASSL : A differential/algebraic system solver. In *Proceedings IMACS World Congress, Montreal, Canada*, 65–68.