Kori Ditmeyer-Moreau — December 28, 2015

In this work on mechanical behaviour of polymers, we are asked to reproduce the different figures presented in the paper of uniaxial constitutive model (called DSGZ model) [1] and we will identify the values of parameters used in the constitutive model to reproduce the experimental curves and predictions.

1 Validation of the DSGZ model

In this first part, we will write a FORTRAN code with the DSGZ constitutive model to compare the results obtained by the computational code with the experimental data obtained from the documentation. We will do this for two different glassy polymers at different temperatures and strain rate.

1.1 For polymethyl-methacrylate (PMMA)

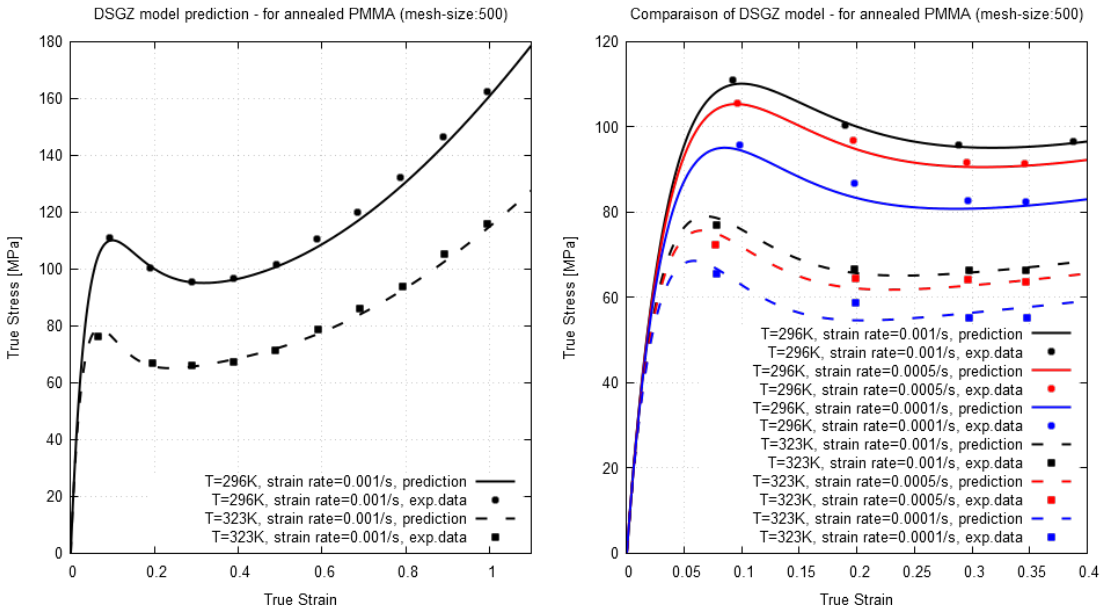
For the validation of the DSGZ constitutive model with PMMA, we have used the eight materials coefficients given by [1] and that are written in the Table 1 below:

Coefficients	C1	C2	m	a	K	C3	C4	α
Duan and AL	1.91	1.49	0.064	1191	3.9	0.0029	11	11.7

Table 1: Material coefficients for PMMA

With the FORTRAN code that we have written we have obtain prediction data that we can compare with the experimental data given by [1]. As we have not the original experimantal data used by [1], we have extract this data from the Figures of [1] using PlotDigitizer software. In the Figure 1 we can see that our FORTRAN model fits well the experimental data as in the [1] paper. Hence, we conclude that our FORTRAN code can properly predict PMMA behavior.

Figure 1: DSGZ model prediction comparison for annealed PMMA



1.2 For polycarbonate (PC)

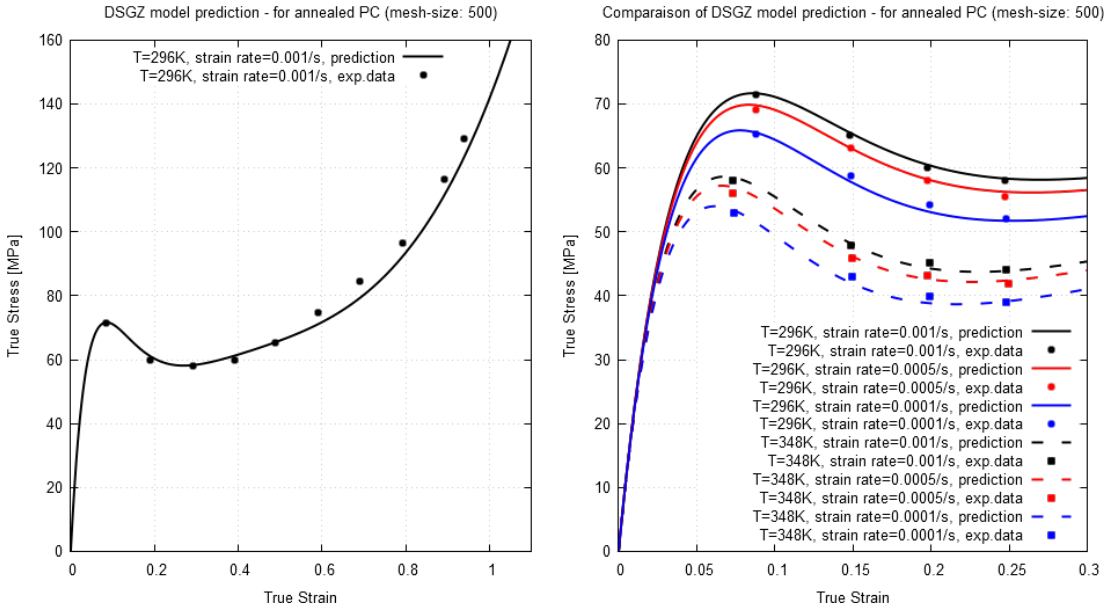
For the validation of the DSGZ constitutive model with PC, we have used the eight materials coefficients given by [1] and that are written in the Table 2 below:

Coefficients	C1	C2	m	a	K	C3	C4	α
Duan and AL	0.49	4.02	0.038	415	28.4	0.03	5.8	6.8

Table 2: Material coefficients for PC

With the FORTRAN code that we have written we have obtain prediction data that we can compare with the experimental data given by [1]. As we have not the original experimantal data used by [1], we have extract this data from the Figures of [1] using PlotDigitizer software. In the Figure 2 we can see that our FORTRAN model fits well the experimental data as in the [1] paper. Hence, we conclude that our FORTRAN code can properly predict PC behavior.

Figure 2: DSGZ model prediction for annealed PC



2 Identification of materials coefficients

The DSGZ model proposed need eight material coefficients to work. In order to calculate the coefficients we need at least three experimental stress-strain curves including variation with strain rate and temperature. The sequential calibration procedures are :

2.1 Compute C1 and C2

In order to calculate C1 and C2 we need to solve the non linear system of equations (1). We must have three large strain points on a stress-strain curve.

$$\begin{cases} \dfrac{\mathrm{e}^{-C_1\varepsilon_1}+\varepsilon_1^{C_2}}{\mathrm{e}^{-C_1\varepsilon_2}+\varepsilon_2^{C_2}} = \dfrac{\sigma_1}{\sigma_2} \\ \\ \dfrac{\mathrm{e}^{-C_1\varepsilon_1}+\varepsilon_1^{C_2}}{\mathrm{e}^{-C_1\varepsilon_3}+\varepsilon_3^{C_2}} = \dfrac{\sigma_1}{\sigma_3} \end{cases} \tag{1}$$

The resolution of this system can be done using an iterative method like Newton-Raphson method [2].Hence, first we must rewrite the system(1) to have a system

of equations equals to zero, as we can see in the system (2).

$$\begin{cases} \sigma_2 * \mathrm{e}^{-C_1\varepsilon_1} + \sigma_2 * \varepsilon_1^{C_2} - \sigma_1 * \mathrm{e}^{-C_1\varepsilon_2} - \sigma_1 * \varepsilon_2^{C_2} = 0 \\ \\ \sigma_3 * \mathrm{e}^{-C_1\varepsilon_1} + \sigma_3 * \varepsilon_1^{C_2} - \sigma_1 * \mathrm{e}^{-C_1\varepsilon_3} - \sigma_1 * \varepsilon_3^{C_2} = 0 \end{cases} \tag{2}$$

We can also define a function $F(3)$ that is equivalent to the two functions written in the system (2).

$$(2) \Leftrightarrow \begin{cases} F_1(C_1, C_2) = 0 \\ \\ F_2(C_1, C_2) = 0 \end{cases} \tag{3}$$

With an order 1 Taylor expansion we can have the equation (4) below:

$$F(C + \delta C) = F(C) + J * \delta C + O(\delta C^2). \tag{4}$$

Where J is the Jacobian matrix with $J_{ij} = \frac{\partial F_i}{\partial C_j}$. In a matrix form we have the equation (5).

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial C_1} & \frac{\partial F_1}{\partial C_2} \\ \frac{\partial F_2}{\partial C_1} & \frac{\partial F_2}{\partial C_2} \end{bmatrix} \tag{5}$$

$$\mathrm{J}= \begin{bmatrix} -\sigma_2 * \varepsilon_1 * \mathrm{e}^{-C_1\varepsilon_1} + \sigma_1 * \varepsilon_2 * \mathrm{e}^{-C_1\varepsilon_2} & \sigma_2 * \ln\varepsilon_1 * \varepsilon_1^{C_2} - \sigma_1 * \ln\varepsilon_2 * \varepsilon_2^{C_2} \\ -\sigma_3 * \varepsilon_1 * \mathrm{e}^{-C_1\varepsilon_1} + \sigma_1 * \varepsilon_3 * \mathrm{e}^{-C_1\varepsilon_3} & \sigma_3 * \ln\varepsilon_1 * \varepsilon_1^{C_2} - \sigma_1 * \ln\varepsilon_3 * \varepsilon_3^{C_2} \end{bmatrix} \tag{6}$$

Neglecting terms of order δC^2 and higher and by setting $F(C + \delta C) = 0$, the equation (4) becomes the equation (7) below:

$$F(C) = -J * \delta C \qquad \Leftrightarrow \qquad \delta C = -J^{-1} * F(C) \tag{7}$$

To solve the equation (7) we compute by hand the inverse of the Jacobian 2*2 matrix as the equation (8) below:

$$J^{-1} = \frac{1}{\det J} * \begin{bmatrix} \frac{\partial F_2}{\partial C_2} & -\frac{\partial F_1}{\partial C_2} \\ -\frac{\partial F_2}{\partial C_1} & \frac{\partial F_1}{\partial C_1} \end{bmatrix} \tag{8}$$

The corrections are then added to the solution vector:

$$C_{new} = C_{old} + \delta C \tag{9}$$

The process described above is then iterated until convergence. We have took as a converge criteria 1^{-20}.

This method present two main problems :

- the system is very sensitive to the values of the three couple strain-stress $(\varepsilon_1;\sigma 1),(\varepsilon_2;\sigma 2),(\varepsilon_3;\sigma 3)$;
- that we must initialize C1 and C2. The choice of these values have a large effect on the convergence of the Newthon-Raphson method. Indeed this method gives a very efficient means of converging, if we have a sufficiently good initial guess otherwise it diverge.

2.2 Compute m

In order to compute m, we must solve the equation (10). We must have two different curves with different strain rate from which we extract two different stresses for the same strain.

$$m = \frac{\ln(\frac{\sigma_1}{\sigma_2})}{\ln(\frac{\dot{\varepsilon_1}}{\dot{\varepsilon_2}})} \tag{10}$$

2.3 Compute a

In order to compute m, we must solve the equation (11). We must have two different curves with different temperatures from which we extract two different stresses for the same strain.

$$a = \frac{\ln(\frac{\sigma_1}{\sigma_2})}{\frac{1}{T_1} - \frac{1}{T_2}} \tag{11}$$

2.4 Compute K

K can be compute by solving the equation (12) for a large strain point.

$$K = \frac{\sigma(\varepsilon,\dot{\varepsilon},T)}{\mathrm{e}^{-C_1\varepsilon} + \varepsilon^{C_2} * h(\varepsilon,\dot{T})} \tag{12}$$

2.5 Compute C3

In order to compute C3, we must solve the equation (13). With ε_y the yielding strain corresponding to the maximum stress before softening.

$$C_3 = \frac{\varepsilon_y}{e^{\frac{a}{T}} * (\dot{\varepsilon})^m} \tag{13}$$

2.6 Fitting coefficients

The coefficients α and C4 are important to fit the initial part of the experimental stress-strain curves.

2.6.1 Compute α

In order to compute α, we must solve the equation (14). With ε corresponding to the end point of softening

$$\alpha = \frac{\ln(0.03)}{\varepsilon} \tag{14}$$

2.6.2 compute C4

In order to compute C4, we must solve the equation (15). With $coef = 7$ for glassy polymers and $coef = 200$ for the semicristalline polymers .

$$C_4 = coef + \ln(\dot{\epsilon}^m * e^{\frac{a}{T}}) \tag{15}$$

2.7 Results

With our code we have been able to compute the materials coefficients for annealed PMMA and PC. The results obtain are listed in the Table 3 and 4. We can see that for PMMA, the difference between our code and Duan and al values is close but with PC the percentage of error is really high. This can be due to the fact that the values of strain and stress that we have as input for the calculation of the coefficients are estimated values from PlotDigitizer software.

Coefficients	C1	C2	m	a	K	C3	C4	α
Duan and al	1.91	1.49	0.064	1191	3.9	0.0029	11	11.7
Our code	2.11	1.44	0.073	1201	4.1	0.0026	10.6	11.3
percentage of error (%)	10	3	14	1	5	9	3	4

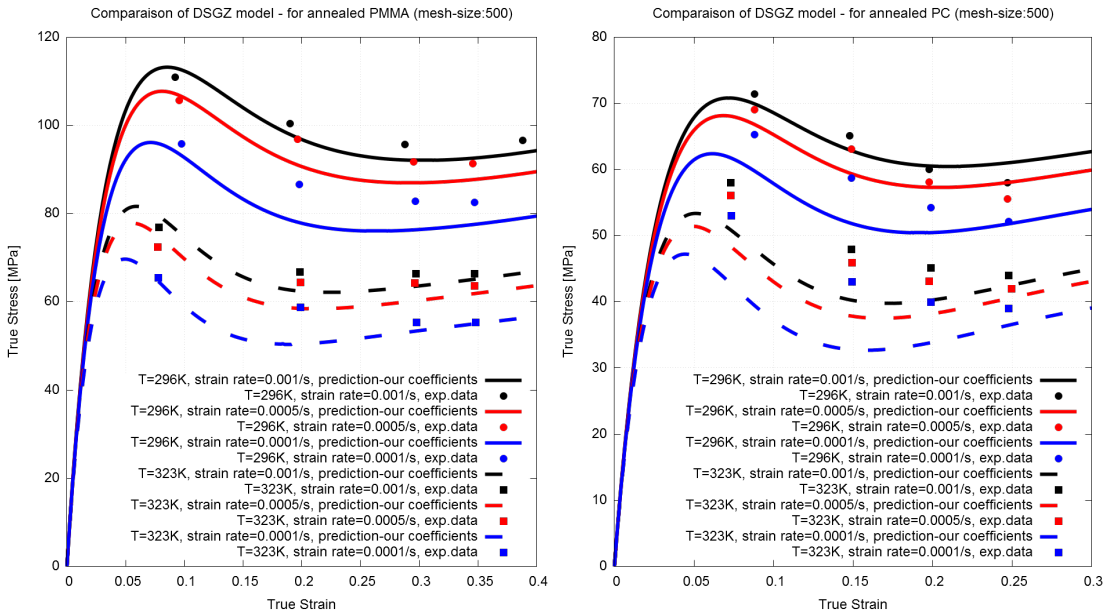
Table 3: Material coefficients for PMMA

Coefficients	C1	C2	m	a	K	C3	C4	α
Duan and al	0.49	4.02	0.038	415	28.4	0.03	5.8	6.8
Our code	0.72	3.79	0.060	617	17.8	0.02	8.7	7.0
percentage of error (%)	46	6	58	49	37	48	50	3

Table 4: Material coefficients for PC

3 Reproducing stress-strain curves with the coefficients

With the coefficients obtained in the §2.7 and with the use of the FORTRAN code writte for §1, we are able to plot a comparison of the predictions obtain with our coefficients and the experimental curves given in the [1] paper. In the Figure 3, we represent the results obtained.



(a) for annealed PMMA (b) for annealed PC

Figure 3: Comparison of stress-strain curves with material coefficients from §2.7 with experimental data

As we can see in the Figure above, the prediction values obtained with the material coefficients that we have calculate in the §2.7 don't fit well with the experimental data.This can be caused by the percentage of errors that we have had when we have calculate the materials coefficients in the §2.7 and as said in the [1] paper, stress-strain curves are very sensitive to the values of the fitting parameters, this can be one other explication of the difference obtained.

Unfortunately we can not compare the result that we have in the figure 3 with the results in the [1] paper because the authors have used the material identification to identify material coefficients from three sets of curves from G'Sell-Jonas and not for PMMA and PC.

References

[1] Duan and Al, *A Uniform Phenomenological Constitutive Model for Glassy and Semicrystalline Polymers*, POLYMER ENGINEERING AND SCIENCE, Medford, Vol.48, 2001.

[2] H. Press, A. Teukolsky, T. Vetterling, P. Flannery, *NUMERICAL RECIPES IN FORTRAN 77: THE ART OF SCIENTIFIC COMPUTING* , CAMBRIDGE UNIVERSITY PRESS, Cambridge, Volume 1 of Fortran Numerical Recipes, 1997.