Task 10 – Model improvement
Implementation of density and viscosity effects on ionic transport of highly concentrated solutions

Etienne Grégoire, M.Sc, Eric Samson, Eng. Ph.D.
SIMCO Technologies Inc.
203-1400 Parc-Technologique, Québec, QC, G1P 4R7, Canada

CONTENTS

1. INTRODUCTION

Concrete storage structures designed to contain saltstone or other types of wasteform are potentially exposed to very high ionic concentrations. The long-term durability of those structures is strongly dependent on the alteration to their microstructure induced by ion diffusion and chemical reactions. One possible factor affecting the diffusion rates of species is the viscosity of the pore solution. Intuitively, one can see that highly concentrated solutions have a high viscosity, which is bound to reduce diffusion rates.

This document describes a model that was implemented in STADIUM® to calculate the density and viscosity of pore solution in cementitious materials and quantify the effect viscosity can have on diffusion of species. After a complete description of the model, examples are provided to assess the long-term impact of viscosity on durability. The complete computer listing is also provided at the end of the report.

1.1. DIFFUSION COEFFICIENT CORRECTION

The aim of this model is to provide a correction for the self-diffusion coefficient of ions, as first suggested by (Gordon, 1937) and further developed by (Price, Mills, & Woolf, 1996); (Leaist & Kanakos, 2000) and (Leaist & Al-Dhaher, 2000) :

$$(1) \qquad D_i^s = D_i^w \left(\frac{\eta_w}{\eta_s} \right)^{\alpha} \quad \text{with } 0 < \alpha < 1$$

where D_i^s is the diffusion coefficient of ionic specie *i* in a solution of viscosity η_s and D_i^w is the diffusion coefficient of ionic species *i* in water of viscosity η_w. This correction is based on the observation that the self-diffusion of ions in a solution is inversely correlated to the viscosity of the solution. The α parameter has been found to be variable, with a value of $\alpha=0.40 \pm 0.05$ suggested by some authors (Price, Mills, & Woolf, 1996).

1.2. THE LALIBERTÉ-COOPER MODEL

The Laliberté-Cooper model has been chosen to evaluate the viscosity of the solution. It provides a method to evaluate the density and the viscosity of solutions of multiple solutes at high concentration and over a large range of temperatures. This model has been described in (Laliberté & Cooper, 2004), (Laliberté, 2007) and (Laliberté, 2009). The model uses coefficients to fit the apparent density and viscosity of solutes at different concentrations and temperatures from experimental results.

2. DESCRIPTION OF THE MODEL IMPLEMENTATION

The functional diagram of the model implementation is shown in Figure 1. The user inputs are the molarity of the ions under study and the temperature of the solution. Currently, the model accounts for

the following ions: Na^+, K^+, Ca^{2+}, Mg^{2+}, OH^-, Cl^-, SO_4^{2-}, NO_3^-, NO_2^-, CO_3^{2-}, PO_4^{3-}, and HCO_3^-, for temperatures ranging from -20°C to 150°C in liquid water.
There are two databases that control the behavior of the model. The first database contains the rules controlling the assembly of the solutes from the ionic concentration. Currently, the model allows for the following solutes: $Ca(NO_3)_2$, $CaCl_2$, KNO_2, Na_2CO_3, Na_2SO_4, Na_3PO_4, $NaHCO_3$, NaCl, K_2CO_3, K_2SO_4, K_3PO_4, KCl, $NaNO_3$, KNO_3, KOH, $MgCl_2$, $MgSO_4$, and NaOH.[1] The second database contains the density and viscosity coefficients needed to calculate the apparent physical properties of the solutes, as well as the molar mass of the solutes needed to convert the molarity of solutes to mass fractions.

The model implementation itself contains four modules:

- the Ionic Combination module converts the ionic molarities to solutes molarities,
- the Molar2Mass module converts solutes molarities to solute mass fractions, which is iterative with the density module,
- the Density module combines three functions for the water, apparent solute, and solution densities, and
- the Viscosity module combines three functions for the water, apparent solute, and solution viscosity.

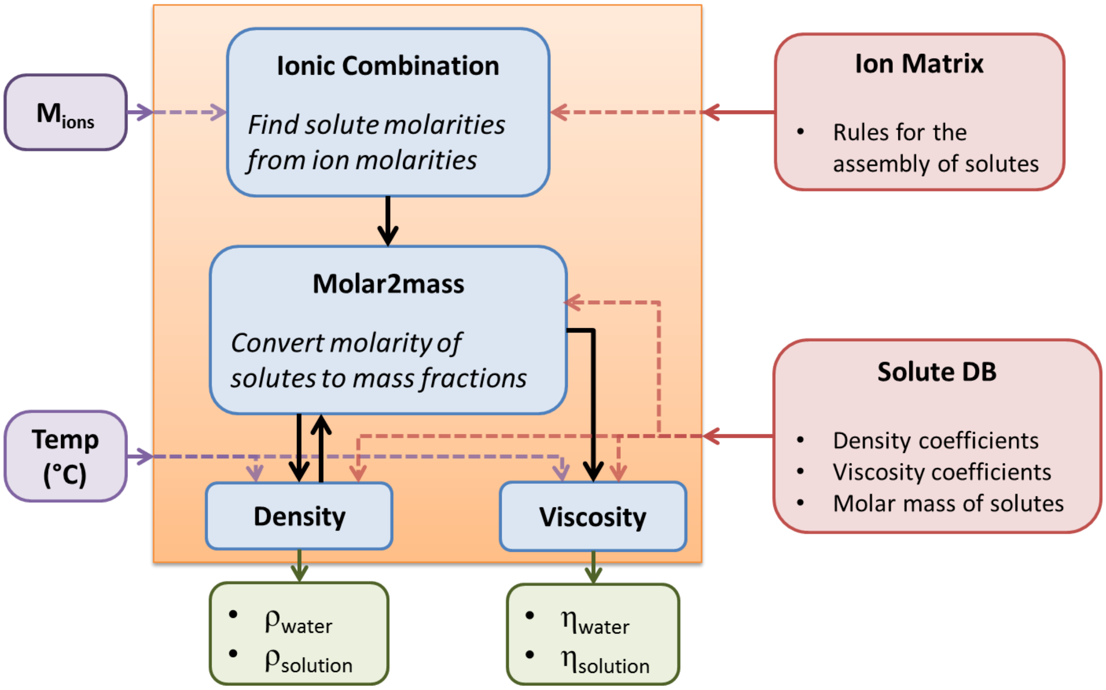

Figure 1: Functional diagram of the implementation of the Laliberté – Cooper density and viscosity model

3. FROM IONS TO SOLUTES

The Laliberté-Cooper model uses the solute concentration data for its calculation, whereas STADIUM® modeling uses ionic concentration in solution. There is need for a conversion from ionic concentration to solute concentration in order to use the Laliberté-Cooper model. To accomplish this, an ionic combination module was incorporated. For this module, the self-consistency of the model

---

[1] The Laliberté-Cooper model has data for more solutes than those currently used this implementation (around 70). The reduced number of solutes allows for an easier ionic combination.

was studied to verify the influence of the choice of solutes, when there is more than one solute combination than can account for the ionic molarity of the solution.

3.1. SELF-CONSISTENCE

The study by (Reynolds & Carter, 2008) proves that the Laliberté-Cooper is self-consistent when analyzing the density of solutions, i.e. any variation of solutes that account for the ionic concentration will yield solutions with similar densities within 0.1%. This same conclusion was not established for the viscosity of solutions. In order to test the variability induced by the choice of solutes that accounts for the ionic concentration of a solution, three tests were made. The solutions studied were NaCl+KOH with complimentary KCl+NaOH (Table 1), NaCl+NH_4NO_3 with complimentary $NaNO_3$+NH_4Cl (Table 2), and Na_2SO_4+K_2CO_3 with complimentary Na_2CO_3+K_2SO_4 (Table 3).

Table 1: Variation of the viscosity ratio for NaCl+KOH and KCl+NaOH solutions

Molarity of ions (mmol/L)				**Viscosity ratio**	**Variation**
NaCl	KCl	KOH	NaOH	(η_w / η_s)	
5000	0	5000	0	0.266389	10.9%
0	5000	0	5000	0.295415	
1000	0	1000	0	0.811897	0.6%
0	1000	0	1000	0.816951	
100	0	100	0	0.979816	2.3%
0	100	0	100	1.002274	

Table 2: Variation of the viscosity ratio for NaCl+NH_4NO_3 and $NaNO_3$+NH_4Cl solutions

Molarity of ions (mmol/L)				**Viscosity ratio**	**Variation**
NaCl	$NaNO_3$	NH_4NO_3	NH_4Cl	(η_w / η_s)	
5000	0	5000	0	0.407188	14.7%
0	5000	0	5000	0.467232	
1000	0	1000	0	0.923501	0.1%
0	1000	0	1000	0.924867	
100	0	100	0	1.000444	0.7%
0	100	0	100	0.993874	

Table 3: Variation of the viscosity ratio for Na_2SO_4+K_2CO_3 and Na_2CO_3+K_2SO_4 solutions

Molarity of ions (mmol/L)				**Viscosity ratio**	**Variation**
Na_2SO_4	Na_2CO_3	K_2CO_3	K_2SO_4	(η_w / η_s)	
2000	0	2000	0	0.179677	8.8%
0	2000	0	2000	0.165196	
1000	0	1000	0	0.455626	3.1%
0	1000	0	1000	0.441851	
100	0	100	0	0.924137	-
0	100	0	100	0.924475	

These results seem to indicate that the model has an acceptable self-consistency, with very low variation in viscosity for concentrations under 1mol/L. These variations are likely to be caused from the presence of secondary species in solution, the presence of which is not considered in the ionic molarity.

3.2. MISSING SOLUTES

Some viscosity coefficients are missing in the data from (Laliberté, 2009), most notably the data for $CaSO_4$ and $NaNO_2$. These viscosity coefficients are missing because of the lack of experimental data to fit the apparent densities of these solutes in the literature. The steps needed to provide a workaround will be discussed in section 7.2.1.

3.3. IONIC COMBINATION

The ionic combination module uses a list of the ion combination required to form the different solutes and calculate their equivalent molarities. This module uses a matrix that associates the number of ions in a given solute and minimises the residual ionic molarity after the combination. The details of this function are given in section 11.1. Only solutes for which the density and viscosity coefficients are available in the Laliberté-Cooper model are thus considered.

4. DENSITY

The density module combines three functions that calculate the water, solution, and apparent solute density. The details of these functions are given in section 11.4.

4.1. DENSITY OF WATER

The density of liquid water is evaluated from a correlation by (Kell, 1975):

$$\rho_w / kg \cdot m^{-3} = \frac{\begin{array}{c}(((((-2.8054253 \cdot 10^{-10} \cdot t/°C + 1.0556302 \cdot 10^{-7}) \cdot t/°C - \\ 4.6170461 \cdot 10^{-5}) \cdot t/°C - 7.9870401 \cdot 10^{-3}) \cdot t/°C + \\ 1.6945176 \cdot 10^{1}) \cdot t/°C + 9.9983952 \cdot 10^{2})\end{array}}{1 + 1.6879850 \cdot 10^{-2} \cdot t/°C} \tag{2}$$

where t is the temperature of water in °C.

4.2. DENSITY OF SOLUTION

The density of the solution is given by:

$$\rho_s / kg \cdot m^{-3} = \frac{1}{\dfrac{w_w}{\rho_w / kg \cdot m^{-3}} + \sum_i \dfrac{w_i}{\rho_{app,i} / kg \cdot m^{-3}}} \tag{3}$$

where w_w is the mass fraction of water, w_i is the mass fraction of solute *i*, ρ_w is the density of water and $\rho_{app,i}$ is the apparent density of solute *i*.

4.3. DENSITY OF SOLUTE

The apparent density of solute *i* is given by:

$$\rho_{app,i} / kg \cdot m^{-3} = \frac{(c_0(1-w_w)+c_1)e^{(0.000001 \cdot (t/°C + c_4)^2)}}{(1-w_w)+c_2+c_3 \cdot t/°C} \tag{4}$$

where t is the temperature of the solution and c_0 to c_4 are empirical non-dimensional constants that are evaluated from experimental data. The most up-to-date data for the constants are available in the supporting information of (Laliberté, 2009).

5. MASS FRACTION

The Laliberté-Cooper density and viscosity models use the mass fraction of solutes, while the data is available for the molarity of solutes. In order to convert the molarity to a mass fraction, the density of the solute and the solution are needed. Hence, solving for the mass fraction is an iterative process.

5.1. MASS FRACTION OF SINGLE SOLUTE

To find the mass fraction of a single electrolyte in a solution, the mass fraction is estimated, then the molarity of the solute is calculated with the following relation:

(5) $$M_i = \frac{w_i \rho_s}{W_i}$$

where M_i is the molarity, w_i is the mass fraction and W_i is the molar mass of solute *i* and ρ_s is the density of the solution that is given with equation (3). The molarity calculated with equation (5) is then compared with the molarity of the solute and corrected. The estimation is considered acceptable when the difference between the estimated and exact molarities is less than 0.001 mmol/L. The details of this function are given in section 11.2.

5.2. MASS FRACTION OF MULTIPLE SOLUTES

Since the model is implemented for multiple electrolytes, the mass fraction calculation was modified to find the mass fraction of multiple electrolytes in a solution. For all solutes *i*, the following ratio is calculated:

(6) $$R_i = \frac{w_i}{1-w_i} = \frac{w_i}{w_w}$$

where R_i is the solute:water mass fraction that is calculated from equation (5). This ratio must hold for the electrolytes to be in equilibrium with the solvent in the solution.
Since the total mass fraction of water and electrolytes is 1, then:

(7) $$1 = w_w + \sum_i w_i$$

Substituting (6) into (7), we get:

(8) $$1 = w_w + \sum_i R_i w_w$$
$$w_w = \frac{1}{1+\sum_i R_i}$$

Thus the mass fraction of water w_w is obtained, and substituting into (6), the mass fractions of individual solutes are calculated. The details of this function are given in section 11.3.

6. VISCOSITY

The viscosity module combines three functions that calculate the water, solution, and apparent solute viscosity. The details of these functions are given in section 11.5.

6.1. VISCOSITY OF WATER

The viscosity of water is evaluated by a relation from (Laliberté, 2007), which is based on data from (IAPWS, 2003):

(9) $$\eta_w / mPa \cdot s = \frac{t/{}^\circ C + 246}{(0.05594 \cdot t/{}^\circ C + 5.2842) \cdot t/{}^\circ C + 137.37}$$

where *t* is the temperature of the solution in °C

6.2. VISCOSITY OF SOLUTION

The viscosity of the solution is given by:

(10) $$\eta_s = \eta_w^{w_w} \prod_i \eta_i^{w_i}$$

where w_w is the mass fraction of water, w_i is the mass fraction of solute *i*, η_w is the viscosity of water and η_i is the viscosity of solute *i* in mPa·s.

6.3. VISCOSITY OF SOLUTE

The viscosity of solute *i* is given by:

(11) $$\eta_i / mPa \cdot s = \frac{e^{\left(\frac{v_1(1-w_w)^{v_2}+v_3}{v_4(t/°C)+1}\right)}}{v_5(1-w_w)^{v_6}+1}$$

where *t* is the temperature of the solution and v_1 to v_6 are empirical non-dimensional constants that are evaluated from experimental data. The most up-to-date data for the constants are available in the supporting information of (Laliberté, 2009).

7. EXPERIMENTAL VERIFICATION

7.1. SINGLE SOLUTES

A first verification was made to insure that the model was able to predict correctly the viscosity of single solutes at 25°C. Figure 2 shows the experimental (Expr) and predicted values (Pred) for the electrolytes KNO_2, KNO_3 and $NaNO_3$ at molarities up to 3 mol/L. Experimental values are from the supplementary information of (Laliberté, 2007). The predicted values seem to be in agreement with the experimental data for this range of molarities.[2]

7.2. MULTIPLE SOLUTES

The model verification for multiple solutes case was performed with experimental results from (Nowlan, Doan, & Sangster, 1980), which studied the density and viscosity of five mixed electrolyte solutions : (1) NaCl + KCl, (2) NaCl + NH_4NO_3, (3) NaCl + $Ca(NO_3)_2$, (4) NaCl + $CaCl_2$, and (5) NaCl + $MgSO_4$.

The density and viscosity were calculated at 25°C for three different mixing ratios of the solutes. Table 4 and Table 5 show the comparison of the predicted values from the Laliberté-Cooper model implementation and the values measured by (Nowlan, Doan, & Sangster, 1980). M_s is the total molarity of the solutes and the solute ratios are indicated at the top of the tables. The absolute difference between experimental and predicted values are also shown for each data pair.

---

[2] The KNO_3 look slightly low at molarities under 3 mol/L, but the fit is better at higher molarities. The plot only shows molarities under 3 mol/L for a better look at the molarities that are of interest in concrete problems.

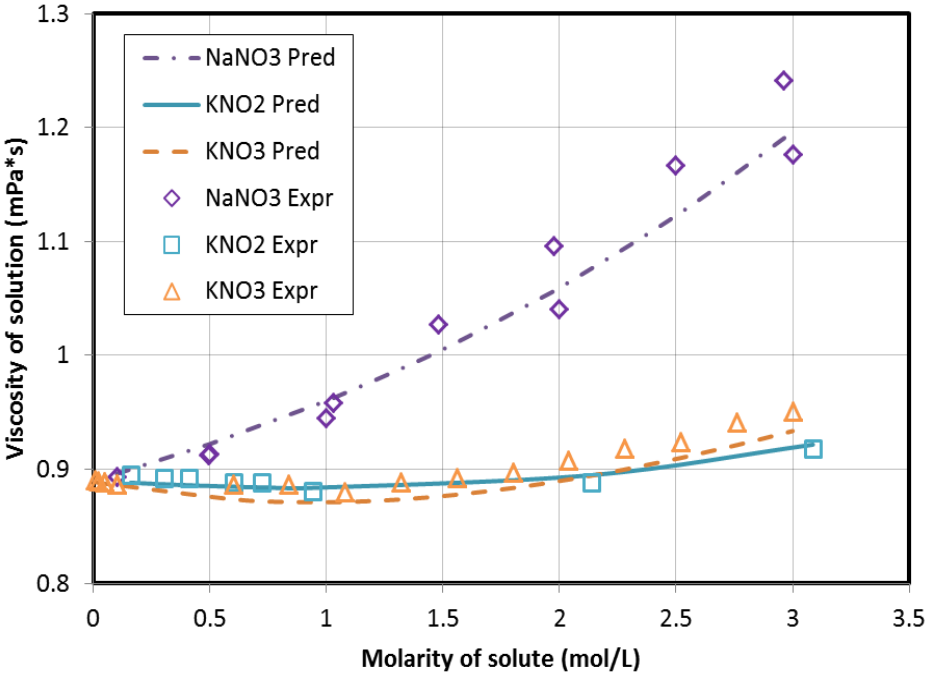

Figure 2 : Experimental (Expr) and predicted values by the model (Pred) for three different solutes at various molarities

Table 4 : Comparison of density ratios from experimental values of Nowlan et al (1980) and predicted values of the Laliberté-Cooper model implementation for mixed electrolyte solutions.

	Solute Ratios (1) : (2)								
	1 : 3			**1 : 1**			**3 : 1**		
M_s	d_r *(expr.)*	d_r *(pred.)*	$\%\Delta_{(p-e)}$	d_r *(expr.)*	d_r *(pred.)*	$\%\Delta_{(p-e)}$	d_r *(expr.)*	d_r *(pred.)*	$\%\Delta_{(p-e)}$
	NaCl (1) + KCl (2) + H_20								
0.5	0.9807	0.9782	0.26%	0.9816	0.9789	0.28%	0.9823	0.9796	0.28%
1	0.9599	0.9578	0.21%	0.9618	0.9593	0.26%	0.9626	0.9605	0.21%
1.5	0.9410	0.9388	0.23%	0.9440	0.9410	0.32%	0.9454	0.9427	0.29%
2	0.9227	0.9211	0.17%	0.9254	0.9239	0.16%	0.9278	0.9260	0.20%
2.5	0.9078	0.9044	0.37%	0.9093	0.9080	0.15%	0.9112	0.9103	0.11%
3	0.8904	0.8887	0.19%	0.8934	0.8930	0.05%	0.8953	0.8955	0.03%
3.5	0.8733	0.8740	0.08%	0.8784	0.8789	0.06%	0.8812	0.8816	0.04%
4	0.8592	0.8601	0.10%	0.8628	0.8657	0.33%	0.8663	0.8684	0.24%
	NaCl (1) + NH_4NO_3 *(2) +* H_20								
0.5	0.9862	0.9832	0.30%	0.9860	0.9823	0.38%	0.9845	0.9813	0.33%
1	0.9701	0.9676	0.26%	0.9685	0.9658	0.28%	0.9663	0.9638	0.25%
2	0.9409	0.9390	0.20%	0.9371	0.9361	0.10%	0.9337	0.9322	0.16%
4	0.8885	0.8909	0.27%	0.8828	0.8870	0.48%	0.8772	0.8794	0.25%
6	0.8436	0.8515	0.94%	0.8368	0.8481	**1.35%**	0.8290	0.8370	0.96%
8	0.8044	0.8186	**1.76%**	0.7961	0.8165	**2.57%**	-	-	-
	NaCl (1) + $Ca(NO_3)_2$ *(2) +* H_20								
0.5	0.9548	0.9525	0.25%	0.9641	0.9616	0.26%	0.9732	0.9708	0.25%
1	0.9121	0.9109	0.13%	0.9294	0.9275	0.21%	0.9464	0.9443	0.22%
1.5	0.8747	0.8741	0.07%	0.8975	0.8968	0.07%	0.9212	0.9201	0.13%
2	0.8412	0.8414	0.02%	0.8683	0.8693	0.11%	0.8975	0.8978	0.04%
3	0.7809	0.7852	0.54%	0.8163	0.8215	0.63%	0.8547	0.8585	0.44%
4	0.7319	0.7386	0.92%	0.7715	0.7814	**1.28%**	0.8170	0.8248	0.95%
	NaCl (1) + $CaCl_2$ *(2) +* H_20								
0.5	0.9660	0.9635	0.26%	0.9727	0.9690	0.37%	0.9774	0.9746	0.29%
1	0.9327	0.9304	0.25%	0.9455	0.9408	0.50%	0.9540	0.9512	0.30%
1.5	0.9020	0.9005	0.17%	0.9176	0.9150	0.28%	0.9314	0.9295	0.20%
2	0.8747	0.8732	0.17%	0.8960	0.8914	0.51%	0.9118	0.9094	0.27%
3	0.8250	0.8255	0.05%	0.8504	0.8496	0.09%	0.8728	0.8733	0.06%
4	-	-	-	0.8152	0.8140	0.14%	0.8405	0.8418	0.15%
	NaCl (1) + $MgSO_4$ *(2) +*H_20								
0.5	0.9548	0.9531	0.19%	0.9639	0.9619	0.20%	0.9734	0.9709	0.25%
1	0.9132	0.9126	0.07%	0.9292	0.9283	0.09%	0.9464	0.9446	0.19%
1.5	0.8764	0.8771	0.08%	0.8987	0.8984	0.03%	0.9217	0.9206	0.13%
2	0.8435	0.8454	0.23%	0.8699	0.8714	0.18%	0.8989	0.8985	0.04%

Table 5 : Comparison of viscosity ratios from experimental values of Nowlan et al (1980) and predicted values of the Laliberté-Cooper model implementation for mixed electrolyte solutions.

	Solute Ratios (1) : (2)								
	1 : 3			**1 : 1**			**3 : 1**		
M_s	η_r *(expr.)*	η_r *(pred.)*	$\%\Delta_{(p-e)}$	η_r *(expr.)*	η_r *(pred.)*	$\%\Delta_{(p-e)}$	η_r *(expr.)*	η_r *(pred.)*	$\%\Delta_{(p-e)}$
				NaCl (1) + KCl (2) + H_20					
0.5	0.9928	0.9920	0.08%	0.9814	0.9807	0.06%	0.9667	0.9696	0.30%
1	0.9821	0.9808	0.14%	0.9598	0.9582	0.17%	0.9347	0.9360	0.14%
1.5	0.9660	0.9664	0.04%	0.9337	0.9329	0.09%	0.8999	0.9001	0.02%
2	0.9510	0.9492	0.19%	0.9034	0.9055	0.23%	0.8631	0.8627	0.05%
2.5	0.9286	0.9293	0.08%	0.8755	0.8763	0.09%	0.8221	0.8243	0.27%
3	0.9083	0.9069	0.15%	0.8423	0.8455	0.37%	0.7829	0.7851	0.28%
3.5	0.8808	0.8821	0.15%	0.8055	0.8132	0.95%	0.7462	0.7451	0.14%
4	0.8524	0.8553	0.35%	0.7708	0.7797	**1.16%**	0.7014	0.7047	0.47%
				NaCl (1) + NH_4NO_3 *(2) +* H_20					
0.5	1.0064	1.0100	0.35%	0.9908	0.9930	0.23%	0.9769	0.9759	0.11%
1	1.0063	1.0037	0.26%	0.9751	0.9738	0.14%	0.9480	0.9439	0.44%
2	0.9900	0.9763	**1.38%**	0.9317	0.9235	0.88%	0.8778	0.8716	0.71%
4	0.8974	0.8885	0.99%	0.8042	0.8004	0.48%	0.7167	0.7142	0.36%
6	0.7633	0.7703	0.92%	0.6491	0.6620	**1.99%**	0.5456	0.5531	**1.37%**
8	0.6036	0.6388	**5.82%**	0.4893	0.5265	**7.59%**	-	-	-
				NaCl (1) + $Ca(NO_3)_2$ *(2) +* H_20					
0.5	0.9047	0.8147	**9.95%**	0.9216	0.8589	**6.80%**	0.9360	0.9068	**3.12%**
1	0.8046	0.6745	**16.17%**	0.8425	0.7430	**11.82%**	0.8754	0.8221	**6.08%**
1.5	0.7030	0.5659	**19.51%**	0.7582	0.6469	**14.67%**	0.8118	0.7458	**8.13%**
2	0.6020	0.4795	**20.34%**	0.6754	0.5662	**16.17%**	0.7479	0.6770	**9.47%**
3	0.4092	0.3492	**14.66%**	0.5076	0.4377	**13.77%**	0.6145	0.5576	**9.26%**
4	0.2595	0.2494	**3.89%**	0.3638	0.3387	**6.91%**	0.4890	0.4571	**6.53%**
				NaCl (1) + $CaCl_2$ *(2) +* H_20					
0.5	0.8869	0.9227	**4.04%**	0.9163	0.9411	**2.71%**	0.9341	0.9533	**2.06%**
1	0.7872	0.7701	**2.17%**	0.8359	0.8235	**1.49%**	0.8726	0.8744	0.21%
1.5	0.6896	0.6717	**2.60%**	0.7511	0.7289	**2.96%**	0.8081	0.7962	**1.48%**
2	0.6039	0.5977	**1.03%**	0.6836	0.6617	**3.20%**	0.7476	0.7345	**1.74%**
3	0.4397	0.4487	**2.03%**	0.5328	0.5361	0.62%	0.6219	0.6265	0.74%
4	-	-	-	0.4158	0.4130	0.68%	0.5088	0.5202	**2.25%**
				NaCl (1) + $MgSO_4$ *(2) +*H_20					
0.5	0.7772	0.7739	0.42%	0.8340	0.8306	0.42%	0.8948	0.8919	0.32%
1	0.5963	0.5988	0.43%	0.6899	0.6897	0.03%	0.7980	0.7938	0.53%
1.5	0.4491	0.4554	**1.40%**	0.5655	0.5678	0.41%	0.7071	0.7031	0.57%
2	0.3318	0.3385	**2.01%**	0.4597	0.4613	0.33%	0.6195	0.6186	0.13%

7.2.1. ANALYSIS

Table 4 shows that there is good agreement between experimental results and predicted values for the density ratios ($d_r = d_{water} / d_{solution}$) of mixed electrolyte solutions, except for the NaCl + NH_4NO_3 and NaCl + $Ca(NO_3)_2$ solutions at high (≥ 4 mol/L) molarities where the difference can reach 2.57%. Table 5 shows that the difference between experimental results and predicted values for the viscosity ratios ($\eta_r = \eta_{water} / \eta_{solution}$) of mixed electrolyte solutions is higher than that of the density. While the results of NaCl + KCl and NaCl + $MgSO_4$ show low difference, the results for NaCl + $CaCl_2$ and NaCl + $Ca(NO_3)_2$ show generalized differences across molarities and NaCl + NH_4NO_3 shows higher differences at high (8 mol/L) molarity.

Since only one combination of solutes shows discrepancies in the viscosity ratio, and only one source of data is available to show this discrepancy, we believe these results are acceptable. Furthermore, this interchanging of solutes will not occur with the most important solutes present in concrete solutions: the ionic combination module always tries to generate the most likely solutes first before any secondary species.

Measured properties of mixed electrolytes solutions are not frequently available in the literature, hence the difficulty in verifying these values. The presence higher differences in the NaCl + $Ca(NO_3)_2$ than in the other NaCl + (solute) solutions could be caused by experimental or model error. However, the viscosity data of $Ca(NO_3)_2$ from (Laliberté, 2009) proves consistent over a large range of molarities.

7.3. SRNL SIMULANT AND PERMEANTS

Complex solutions were analyzed by (Dixon, Harbour, & Phifer, 2008) for the preparation of simulated saltstone grout for hydraulic and physical testing. The molarity of the solutes are indicated in Table 6. The density and viscosity of these solutions were calculated at 20°C[3].

Table 6 : Molarity of the various used in the SRNL study

	Molarities (mol/L)				
Solution	NaOH	$NaNO_3$	$NaNO_2$	Na_2CO_3	Na_2SO_4
DDA Simulant	0.769	2.202	0.110	0.145	0.044
ARP/MCU Simulant	1.594	3.159	0.368	0.176	0.059
SWPF Simulant	2.866	1.973	0.485	0.118	0.055
DDA Permeant	0.485	2.202	0.110	0.145	0.044
ARP/MCU Permeant	1.377	3.159	0.368	0.176	0.059
SWPF Permeant	2.409	1.973	0.485	0.118	0.055

The predicted properties of these solutions were obtained first by assuming that the concentration of $NaNO_2$ was 0, due to the lack of viscosity data of this solute. The second assumption was that the viscosity properties of $NaNO_2$ are the same as those of $NaNO_3$ (i.e. use the viscosity coefficients of $NaNO_3$ for $NaNO_2$). Results of these calculations are shown in tables and Table 7 and Table 8.

7.3.1. ANALYSIS

Densities and viscosities of the SRNL simulants and permeants are generally more precise when the $NaNO_2$ is considered and that its viscosity properties are equal to those of $NaNO_3$. Using this hypothesis, the DDA simulant and permeant viscosities were overestimated by 9.5% and 9.8%,

---

[3] The temperature at which the physical properties are calculated is never stated explicitly, but page 9 of the report indicates that the dynamic viscosity of tap water is 1.002 cP which is the viscosity for water at 20°C.

respectively. Viscosity errors for the other solutions were up to 4.46% without the $NaNO_2$ solute and up to 2.26% with $NaNO_2$ solutes.

Table 7 : Predicted and experimental density of SRNL solutions

	Density (g/ml)		
	Predicted		Experimental
Solution	w/o $NaNO_2$	w/ $NaNO_2$	
DDA Simulant	1.158	1.162	1.173
ARP/MCU Simulant	1.229	1.239	1.261
SWPF Simulant	1.209	1.222	1.248
DDA Permeant	1.149	1.154	1.156
ARP/MCU Permeant	1.223	1.234	1.248
SWPF Permeant	1.196	1.210	1.224

Table 8: Predicted and experimental viscosity of SRNL solutions

	Viscosity (mPa*s)		
	Predicted		Experimental
Solution	w/o NaNO2	w/ NaNO2	
DDA Simulant	1.642	1.658	1.50
ARP/MCU Simulant	2.355	2.429	2.46
SWPF Simulant	2.713	2.821	2.78
DDA Permeant	1.526	1.541	1.39
ARP/MCU Permeant	2.231	2.302	2.25
SWPF Permeant	2.411	2.512	2.46

7.4. DATA FOR $NaNO_2$

Data for nitrites from the Laliberté-Cooper model and from literature is only available for KNO_2. The results of the single-electrolyte properties of KNO_2 and KNO_3, from Figure 2, suggest a similar behavior of nitrites and nitrates on the viscosity of solutions. The results from the SRNL data shows good agreement between experimental and predicted properties when $NaNO_3$ data is used to calculate $NaNO_2$ viscosity.

8. CONCRETE EXPOSURE SOLUTIONS

Preliminary analyses of solutions were performed to assess the correction of the ionic diffusion coefficient from the viscosity ration for eq. (1). The solutions were (1) standard seawater at 33 ppt at various temperatures, (2) pore solutions calculated by STADIUM® under chloride exposure conditions at various depths, and (3) pore solutions calculated by STADIUM® under saltstone exposure conditions at various depths.

8.1. SEAWATER

Table 9 shows the viscosity ratios of seawater at 33 ppt at temperatures from 4°C to 25°C. The solute molarities calculated with the ionic combination module for the specified ionic molarities are shown in Table 10. These results show that the viscosity of the solutions is affected by the temperature change in a different proportion than water alone.

Table 9 : Viscosity ratio of a seawater solution at different temperatures

Temp (°C)	Molarity of ions (mmol/L) Na^+	K^+	Ca^{2+}	Mg^{2+}	Cl^-	SO_4^{2-}	HCO_3^-	**Viscosity ratio** (η_w / η_s)
4	439.4	9.3	9.6	50.1	511.4	26.4	3.9	0.9453
10	439.4	9.3	9.6	50.1	511.4	26.4	3.9	0.9434
15	439.4	9.3	9.6	50.1	511.4	26.4	3.9	0.9418
20	439.4	9.3	9.6	50.1	511.4	26.4	3.9	0.9403
25	439.4	9.3	9.6	50.1	511.4	26.4	3.9	0.9388

Table 10 : Solute molarities calculated from the ionic molarities by the ionic combination module

Molarity of solutes (mmol/L) $CaCl_2$	Na_2SO_4	$NaHCO_3$	NaCl	KCl	$MgCl_2$
9.6	26.4	3.9	382.7	9.3	50.1

8.2. CHLORIDE EXPOSITION

Table 11 shows the viscosity ratios for nine pore solutions as calculated with chloride exposure simulations by STADIUM®. These results consider the pore solution at different depths from the exposed surface, and the viscosity is calculated for a temperature of 25°C.

For this exposure case, the viscosity ratio rises until the Cl^- ion concentration drops to 0, then decreases slightly over the remaining depth when the $K^+ : Na^+$ ratio increases.

Table 11 : Viscosity ratio of pore solutions of concrete under chloride exposure conditions for different ionic concentrations

Molarity of ions (mmol/L) Na^+	K^+	Ca^{2+}	OH^-	Cl^-	SO_4^{2-}	**Viscosity ratio** (η_w / η_s)
442.0	5.5	26.9	53.7	448.0	0.0	0.9614
363.0	19.5	6.2	110.0	285.0	0.2	0.9736
253.0	39.4	2.1	205.0	90.3	0.5	0.9947
191.0	61.7	1.6	241.0	13.4	0.7	0.9939
158.0	83.9	1.6	243.0	0.0	0.7	0.9923
140.0	99.9	1.6	242.0	0.0	0.7	0.9898
133.0	108.0	1.6	242.0	0.0	0.7	0.9885
130.0	112.0	1.6	243.0	0.0	0.8	0.9878
130.0	113.0	1.6	244.0	0.0	0.8	0.9876

8.3. SALTSTONE

Table 12 shows the viscosity ratios for 10 pore solution compositions as calculated with saltstone exposure simulations by STADIUM® in Task 7. These results consider the pore solution at different depths from the exposed surface, and the viscosity is calculated for a temperature of 15°C.

Exposure cases with high molarities of Na^+ and NO_2^- are somewhat problematic since, as explained in section 3.2, density and viscosity coefficients are missing from the data from (Laliberté, 2009). Currently, the only solute that can be considered with nitrites is KNO_2. Thus, for the case described in Table 12, residual concentrations of Na+ and NO_2^- are not considered for the viscosity calculation.

To account for these ions the viscosity properties of $NaNO_2$ were considered to be the same as those of $NaNO_3$, according to section 7.4. The viscosity results without and with $NaNO_2$ are shown in Table 13 for the analysis of three pore solutions from Table 12.

Table 12: Viscosity ratio of pore solutions of concrete under saltstone exposure conditions for different ionic concentrations

Molarity of ions (mmol/L)									Viscosity ratio
Na+	K+	Ca2+	OH^-	Cl^-	SO_4^{2-}	NO_3^-	NO_2^-	CO_3^{2-}	(η_w / η_s)
811	127	0.979	547	5.40	0.07	217	168	0.000	0.9089
1230	122	0.866	629	6.12	0.11	401	313	0.000	0.8573
1790	118	0.749	704	6.95	0.20	664	523	0.001	0.7994
2420	116	0.642	756	7.70	0.32	986	780	0.003	0.7427
3060	117	0.552	785	8.26	0.47	1320	1050	0.004	0.6923
3670	120	0.424	774	8.58	39.90	1630	1290	0.800	0.6420
3940	122	0.483	767	8.68	59.00	1760	1390	2.820	0.6207
4020	122	0.474	767	8.71	61.30	1800	1420	2.920	0.6152
4130	122	0.463	766	8.76	64.50	1860	1460	3.040	0.6076
4230	123	0.451	766	8.81	68.00	1920	1510	3.180	0.5993

Table 13 : Viscosity ratio for solutions without and with consideration of $NaNO_2$

Molarities (mol/L)							Viscosity ratio
$Ca(NO_3)_2$	KNO_2	Na_2CO_3	Na_2SO_4	NaCl	$NaNO_3$	NaOH	(η_w / η_s)
Original							
0.0010	0.127	0	0.0001	0.0054	0.2150	0.547	0.9134
0.0006	0.117	0	0.0005	0.0083	1.3189	0.785	0.7522
0.0005	0.123	0.0032	0.0680	0.0088	1.9191	0.766	0.6789
Residual NO2 -> NO3							
0.0010	0.127	0	0.0001	0.0054	0.2560	0.547	0.9089
0.0006	0.117	0	0.0005	0.0083	2.2519	0.785	0.6923
0.0005	0.123	0.0032	0.0680	0.0088	3.3061	0.766	0.5992

9. STADIUM® SIMULATION WITH UPDATED MODEL

The moisture transport model and ionic diffusion model inside STADIUM® were updated to reflect the new viscosity and density calculations presented here. The diffusion correction factor α in (1) was conservatively set as 0.4[4]. The viscosity and density of the solution in the moisture transport model was changed as the viscosity and density calculated with the Laliberté-Cooper model.

A 10000-year simulation was performed with a demonstration case where a 300 cm layer of saltstone topped a 20 cm layer of concrete. This corresponds to the simplified two-layer case from Task 7. Comparison between model versions with and without viscosity effect is presented in Figure 3. The figure compares the depth of penetration of the ettringite front in the concrete barrier caused by the presence of sulfate in the saltstone layer. For this particular example, the model considering viscosity

---

[4] This value was selected mainly because no better alternative was available, since no exhaustive data for the diffusion coefficient of electrolyte solutions at varying concentrations is available.

showed a reduction of the penetration depth of 5 mm after 10,000 years compared to the simulation results neglecting viscosity.

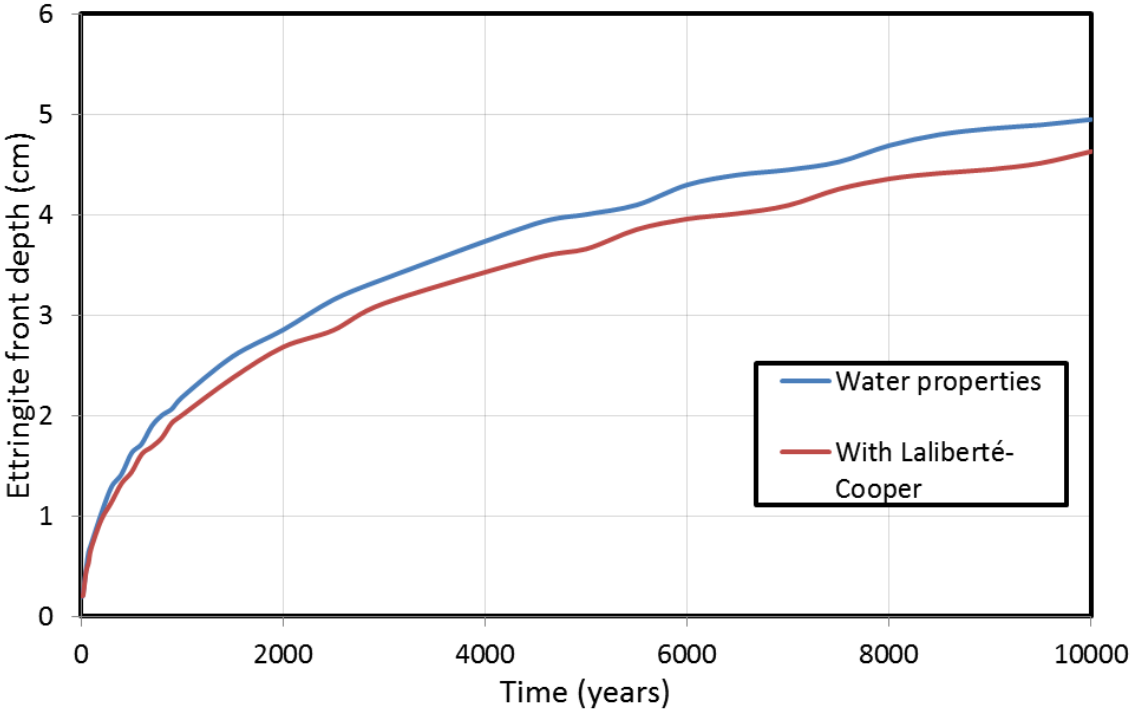

Figure 3 : Comparison of the ettringite front depth from the demonstration case when solution viscosity and density is considered (red) and when water properties are used (blue)

10. CONCLUSIONS

The preliminary results show that the viscosity ratio of solutions is affected by the high ionic molarity of pore solutions in concrete, especially is the case of the saltstone exposure shown in Table 12. When considering the diffusion coefficient correction from (1) and with a conservative α coefficient of 0.40, the ionic diffusion can be lowered by as much as 15%.

The Laliberté-Cooper density and viscosity model, having been tested as a standalone application, has been included into the current working release of STADIUM®. Viscosity results from highly concentrated solutions were compared between the standalone application and the STADIUM® calculation and were shown to be accurate to within 0.1%. Future work will look into the viscosity ratio correction factor for the diffusion factor as described by equation (1).

11. FUNCTIONS

The following functions, as programmed in MATLAB computing language, describe the Laliberté-Cooper implementation used in this study.

11.1. IONIC COMBINATION

```
function [concentration,molarity]=ionic_combination(concentration)
% Function that  uses a list of the ion combination required to form
% the different solutes to calculate their equivalent molarities
%
% Input:     1.Concentration of ions as a list, according to this order:
%      Na+,K+,Ca2+,Mg2+,OH-,Cl-,SO42-,NO3-,NO2-,CO32-,AlO2,PO43-,HCO3-
%
% Output:    1.Residual concentration of ions after solute matching, if any
%            2.Molarity of solutes as a list, according to this order:
```

```
%      NH42SO4,Al2SO43,CaNO32,CaCl2,CaSO4,H2SO4,H3PO4,HNO3,K2CO3,K2HPO4,K2SO4,
%      K3PO4,KCl,KNO2,KNO3,KOH,MgCl2,MgSO4,NaAlOH4,Na2HPO4,NaNO3,Na2CO3,Na2SO4,
%      Na3PO4,NaHCO3,NaHSO4,NaCl,NaNO2,NaOH,NH4Cl,NH4NO3
%
% Loading the ion combination matrix
load ionmatrix.mat

% Initiating the Multiplication matrix
Mult_Matrix=zeros(size(ionmatrix));

% Assume solute molarity vector to be null
molarity=zeros(1,length(ionmatrix));

j=0; %Avoid runaway with max number of iterations
% While there are still unassigned ions with molarities higher than 5mmol/L,
% with 10 maximum iterations
while (any(concentration>0.005) & j<10)

% For all solutes do...
for i=1:length(ionmatrix(:,1))
       %Multiplication of the matrix by the ionic concentration for the solute
       Mult_Matrix(i,:)=concentration .* ionmatrix(i,:);

       %If the ionic concentration is not enough to fulfil the necessary ionic
       %elements of the solute, the multiplication matrix row is set to zero
       %(i.e. the solute will not accept ions)
       if any(Mult_Matrix(i,:)./ionmatrix(i,:)==0)
             Mult_Matrix(i,:)=zeros(1,length(ionmatrix(1,:)));
       endif
endfor

% For all solutes do ...
for i=1:length(ionmatrix(:,1))
       Solute=Mult_Matrix(i,:);
       % If the solute can accept ions (i.e. the concentration of ions is
       % sufficient to form the solute)
       if any(Solute<0)
             % Extract the coefficient of the ions (i.e. remove zeroes from
             % solute row)
             Coeff=Solute(any(Solute,1));
             % Avoid negative ionic concentrations if prior solutes have
             % accounted for a portion of the ions
             if all(concentration+min(abs(Coeff))*ionmatrix(i,:)>=0)
                    % Reduce the ionic concentration by the amount needed to
                    % create the amount of solutes
                    concentration=concentration+min(abs(Coeff))*ionmatrix(i,:);
                    % Raise the solute molarity by the correct amount
                    molarity(i)=molarity(i)+min(abs(Coeff));
             endif
       endif
endfor
j=j+1;
endwhile
endfunction
```

11.2. MASS FRACTION OF SINGLE SOLUTE

```
function massfrac=molar2mass(name,temp,molar)
% Function for computing the mass fraction of a solution using
% the molarity of solutes and the temperature of the solution
%
%      Input:       1. Name of Solute
```

```
%                   2. Temperature of solution
%                   3. Molarity of solute
%
%      Output:      1. Mass fraction of solute
%      Example : molar2mass(NaCl,25,0.2)

% Load the solute database
load solutes.mat

% Extract the Solute Coefficients from DB
MW=eval([name".MW"]);

% Assign starting values for the convergence terms
low_w = 0;
low_M = 0;
resid = 0.0625;
high_w = resid;

% Test the density for a given mass fraction
Test_Density = denssol(temp, 1, name, high_w);

% Compute the Molarity associated with the mass fraction
test_M = high_w * Test_Density / MW;

% If the test Molar concentration is lower than the input molarity
% and the maximum mass fraction (1) is not reached
while ((test_M < molar) & (high_w < 1))
       % Test a higher mass fraction
    low_w = high_w;
    high_w = high_w + resid;
    Test_Density = denssol(temp, 1, name, high_w);
    test_M = high_w * Test_Density / MW;
endwhile

% If the test Molar concentration is higher than the input molarity
% and the temperature is in the acceptable bracket
if ((test_M >= molar) & (temp >= -20) & (temp <= 150))
       % While the minimal resolution (0.001 mmol/L) is not reached do...
    while (resid > 0.000001)
             % Test a new mass fraction
        test_w = (low_w + high_w) / 2;
        Test_Density = denssol(temp, 1, name, test_w);
        test_M = test_w * Test_Density / MW;
             % If the test molarity is higher than the input molarity,
             % try a lower mass fraction
        if (test_M >= molar)
            high_w = test_w;
             % If the test molarity is lower than the input molarity,
             % try a higher mass fraction
        else
            low_w = test_w;
        endif
             % Reduce the resolution by half
        resid = resid / 2;
    endwhile
    massfrac = test_w;
% If the temperature is not in the acceptable bracket, the mass
% fraction is not determined.
else
    massfrac = NaN;
endif
endfunction
```

11.3. MASS FRACTION OF MULTIPLE SOLUTES

```
function massfrac=molar2massMultiple(temp,nbsolutes,solutes,molarity)
% Function for computing the mass fraction of a solution using
% the molarity of solutes and the temperature of the solution
%
%      Input:       1. Temperature of Solution
%                   2. Number of solutes
%                   3. Vector containing the names of the solutes, as strings
%                   4. Molarity of the solutes, as strings
%
%      Output:      1. Mass fraction of solutes
%      Exemple : molar2massMultiple(25,2,[NaCl NaOH],[1 2])

% Load the solute database
load solutes.mat


% Calculate, for all solutes, the mass fraction for the individual
% solute, as well as the solute to water mass fraction
for i=1:nbsolutes
      if molarity(i)>0 %Only calculate the mass fraction if the molarity is
higher than 0
            temp_massfrac(i)=molar2mass(solutes(i,:),temp,molarity(i));
      else
            temp_massfrac(i)=0;
      endif
      %Calculate the solute to water ratio
      ratio(i)=temp_massfrac(i)/(1-temp_massfrac(i));
end

% Since the sum of the mass fractions of the water and solutes needs
% to be 1, then 1=wwater+sum(ratio*wwater) for all i
wwater=1/(1+sum(ratio));

% Calculate the final mass fraction for all solutes based on the new
% wwater and ratios calculated earlier.
for i=1:nbsolutes
      massfrac(i)=ratio(i)*wwater;
end
```

11.4. DENSITY

11.4.1. DENSITY OF WATER

```
function rhow=densitywater(temp)
% This function takes the temperature, in °C, as an input and
% returns the density of water based on the equation proposed by Laliberté 2004
%
%      Input:       1. Temperature of water
%
%      Output:      1. Density of water
rhow = (((((-2.8054253E-10 * temp + 0.00000010556302) * temp - ...
            0.000046170461) * temp - 0.0079870401) * temp + ...
            16.945176) * temp + 999.83952) / (1 + 0.01687985 * temp);

Endfunction
```

11.4.2. DENSITY OF SOLUTE

```
function DensSolute=density(name,temp,wwater)
% Function for computing the density of a solute using the mass fraction of
% solutes and the temperature of the solution
%      Input:       1. Name of Solute
```

```
%                   2. Temperature of solution
%                   3. wwater -> This version uses the 2009 formula which gives
%                   the apparent density as a function of the sum of the mass
%                   fractions of solutes, hence (1-wwater)
%
%      Output:      1. Apparent density of solute i
%      Exemple : density(NaCl,25,0.95)

%Load the solute database
load solutes.mat

%Extract the density Coefficients from DB
SC=eval([name".c"]);

% Calculate the Density of the solute with Laliberté 2009
% Remember that the equation uses c from 0 to 4, but indices are 1 to 5
DensSolute=((SC(0+1)*(1-wwater) + SC(1 + 1))*exp(0.000001*(temp + SC(4 +
1))^2))/((1-wwater) + SC(2 + 1) + SC(3 + 1)*temp);

endfunction
```

11.4.3. DENSITY OF SOLUTION

```
function DensitySolution=denssol(temp,nbsolutes,solutes,massfrac)
% Function for computing the density of a solution using the mass
% fraction of solutes and the temperature of the solution as inputs
%
%      Input:       1. Temperature of the solution
%                   2. Number of solutes
%                   3. Vector containing the names of the solutes, as strings
%                   4. Mass Fractions of the solutes, as strings
%
%      Output: 1. Density of the solution
%      Exemple : denssol(25,2,["NaCl" "NaOH"],[0.1 0.2])

%      Load solute db
load solutes.mat

if nbsolutes(1)==0
       % If no solutes are present, the solution density is that of water
       DensitySolution=densitywater(temp);
else
       % The mass fraction of water is given by:
       wwater = 1;
       for i=1:nbsolutes
             wwater = wwater - massfrac(i);
       end

       %Sum of fractions
       sumFractions = 0;
       for i=1:nbsolutes
             sumFractions = sumFractions +
massfrac(i)/density(solutes(i,:),temp,wwater);
       end

       % The density of the solution is given by:
       DensitySolution = 1/(wwater/densitywater(temp) + sumFractions);
endif

endfunction
```

11.5. VISOCSITY

11.5.1. VISCOSITY OF WATER

```
function etaw=viscositywater(temp)
% This function takes the temperature, in °C, as an input and returns the
% viscosity of water based on the equation proposed by Laliberté 2007
%
%      Input:       1. Temperature of water
%
%      Output:      1. Viscosity of water
    etaw = (temp + 245.9996) / ((0.055943 * temp + 5.284245) *...
              temp + 137.3664);

endfunction
```

11.5.2. VISCOSITY OF SOLUTE

```
function ViscSolute=viscosity(name,temp,wwater)
% Function for computing the viscosity of a solute using the mass
% fraction of solutes and the temperature of the solution
%
%      Input:       1. Name of Solute
%                   2. Temperature of solution
%                   3. Mass fraction of solute
%
%      Output: 1. Apparent viscosity of solute i
%      Exemple : viscosity(NaCl,25,0.2)

%Load the solute DB
load solutes.mat

%Extract the viscosity coefficients for the solute from the DB
VC=eval([name".v"]);

%Calculate the Viscosity of the solute with Laliberté 2007
ViscSolute=exp((VC(1)*(1-wwater)^(VC(2))+VC(3))/(VC(4)*temp+1))/...
             (VC(5)*(1-wwater)^VC(6)+1);

Endfunction
```

11.5.3. VISCOSITY OF SOLUTION

```
function ViscositySolution=viscsol(temp,nbsolutes,solutes,massfrac)
% Function for computing the viscosity of a solution using the
% mass fraction of solutes and the temperature of the solution as inputs
%
%      Input:       1. Temperature of the solution
%                   2. Number of solutes
%                   3. Vector containing the names of the solutes, as strings
%                   4. Mass Fractions of the solutes, as strings
%
%      Output: 1. Viscosity of the solution
%      Exemple : viscsol(25,2,["NaCl" "NaOH"],[0.1 0.2])

%      Load solute db
load solutes.mat

if nbsolutes(1)==0
       % If no solutes are present, the solution viscosity is that of water
       ViscositySolution=viscositywater(temp);
else
       %The mass fraction of water is given by:
       wwater = 1;
```

```
	for i=1:nbsolutes
		wwater = wwater - massfrac(i);
	end

	% Exponential values of RHS term
	RHS=0;
	for i=1:nbsolutes
		RHS(i) = viscosity(solutes(i,:),temp,wwater)^massfrac(i);
	end

	% The viscosity of the solution is given by:
	ViscositySolution = viscositywater(temp)^wwater * prod(RHS);
endif

endfunction
```

11.6. LALIBERTE

```
function laliberte(temp,ionic_molarity)
% This function takes, as input, the ionic concentration of a series of
% ions and calculates the molarity of the most probable solutes,
% calculates the mass fraction of the different solutes and gives the
% density and viscosity of the solutes based on the Laliberté-Cooper
% model with the 2009 data.
%
%	The first argument must be the temperature and the second must be
%	the ionic concentration, in this order:
%		Na+,K+,Ca2+,Mg2+,OH-,Cl-,SO42-,NO3-,NO2-,CO32-,AlO2,PO43-,HCO3-
% Etienne Gregoire 02/11/2011
%
%
%Load the ionmatrix for the solutes assembly
load ionmatrix
ionic_concentration_test=ionic_molarity;

%Forms the most probable solutes
for i=1:length(ionic_concentration_test(:,1))
	[concentration(i,:),molarity(i,:)]=…
		ionic_combination(ionic_concentration_test(i,:));
endfor

solutes=["NH42SO4";"Al2SO43";"CaNO32";"CaCl2";"CaSO4";"H2SO4";"H3PO4";"HNO3";"K
NO2";"NaAlOH4";"Na2HPO4";"Na2CO3";"Na2SO4";"Na3PO4";"NaHCO3";"NaHSO4";"NaCl";"N
aNO2";"K2CO3";"K2HPO4";"K2SO4";"K3PO4";"KCl";"NaNO3";"KNO3";"KOH";"MgCl2";"MgSO
4";"NaOH";"NH4Cl";"NH4NO3"];

massfrac=zeros(size(molarity));
nbsolutes=length(solutes);

nbsolutions=length(molarity(:,1));

DensitySolution=zeros(nbsolutions,1);
DensityWater=zeros(nbsolutions,1);
DensityRatio=zeros(nbsolutions,1);

ViscositySolution=zeros(nbsolutions,1);
ViscosityWater=zeros(nbsolutions,1);
ViscosityRatio=zeros(nbsolutions,1);


for i=1:nbsolutions
	massfrac(i,:)=molar2massMultiple(temp(i),nbsolutes,solutes,molarity(i,:);
```

```
       DensitySolution(i)  =denssol(temp(i),nbsolutes,solutes,massfrac(i,:));
       DensityWater(i)     =densitywater(temp(i));
       %Ratio of water density / solution density
       DensityRatio(i)     =DensityWater(i) / DensitySolution(i) ;

       ViscositySolution(i)=viscsol(temp(i),nbsolutes,solutes,massfrac(i,:));
       ViscosityWater(i)   =viscositywater(temp(i));
       %Ratio of water viscosity / solution viscosity
       ViscosityRatio(i)   =ViscosityWater(i) / ViscositySolution(i);

       str = fprintf('Calculation for solution %d/%d is completed.\n\n', i,...
               nbsolutions);
endfor
endfunction
```

REFERENCES

Aboott, A. P., Hope, E. G., & Palmer, D. J. (2005). Effect of Electrolyte Concentration on the Viscosity and Voltammetry of Supercritical Solutions. *Analytical Chemistry*, 6702-6708.

Breslau, B. R., & Miller, I. F. (1970). On the Viscosity of Concentrated Aqueous Electrolyte Solutions. *The Journal of Physical chemistry*, 1056-1061.

Clegg, S. L., & Wexler, A. S. (2011). Densities and Apparent Molar Volumes of Atmospherically Important Electrolyte Solutions. 1. The Solutes H2SO4, HNO3, HCl, Na2SO4, NaNO3, NaCl, (NH4)2SO4, NH4NO3, and NH4Cl from 0 to 50°C, Including Extrapolations to Very Low Temperature and [...]. *The Journal of Physical Chemistry A*, 3393-3460.

Dixon, K., Harbour, J., & Phifer, M. (2008). *Hydraulic and Physical Properties of Saltstone Grouts and Vault Concretes.* SRNL-STI-2008-00421: SRNL.

Gordon, A. R. (1937). The Diffusion Constant of an Electrolyte, and Its Relation to Concentration. *Journal of Chemical Physics*, 522-526.

Hu, Y.-F., Chu, H.-D., Li, J.-G., Liu, Z.-C., Peng, X.-M., Ling, S., et al. (2011). Extension of the Simple Equations for Prediction of the Properties of Mixed Electrolyte Solutions to the Mixed Ionic Liquid Solutions. *Industrial & Engineering Chemistry Research*, 4161-4165.

IAPWS. (2003). *Revised Release on the IAPS Formulation 1985 for the Viscosity of Ordinary Water.* Vejle, Denmark.

IAPWS. (2008). *Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance.* Berlin, Germany.

Jiang, J., & Sandler, S. I. (2003). A New Model for the Viscosity of Electrolyte Solutions. *Ind. Eng. Chem. Res.*, 6267-6272.

Kell, G. S. (1975). Density, Thermal Expansivity, and Compressibility of Liquid Water from 0° to 150°C: Correlations and Tables for Atmospheric Pressure and Saturation Reviewed and Expressed on 1968 Temperature Scale. *Journal of Chemical and Engineering Data*, 97-105.

Lakshamanan, S., & Nagaraja Rao, K. (1969). Studies on the influence of non-electrolytes on the viscosity of an electrolyte solution. *Electrochimica Acta*, 1173-1181.

Laliberté, M. (2007). Correction to Model for Calculating the Viscosity of Aqueous Solutions. *Journal of Chemical and Engineering Data*, 1507-1508.

Laliberté, M. (2007). Model for Calculating the Viscosity of Aqueous Solutions. *Journal of Chemical and Engineering Data*, 321-335.

Laliberté, M. (2009). A Model for Calculating the Heat Capacity of Aqueous Solutions, with Updated Density and Viscosity Data. *Journal of Chemical & Engineering Data*, 1725-1760.

Laliberté, M., & Cooper, W. E. (2004). Model for Calculating the Density of Aqueous Electrolyte Solutions. *Journal of Chemical and Engineering Data*, 1141-1151.

Lam, E. J., Alvarez, M. N., Galvez, M. E., & Alvarez, E. B. (2008). A Model for Calculating the Density of Aqueous Multicompoment Electrolyte Solutions. *Journal of the Chilean Chemical Society*, 1404-1409.

Leaist, D. G., & Al-Dhaher, F. F. (2000). Predicting the Diffusion Coefficients of Concentrated Mixed Electrolyte Solutions from Binary Solution Data. NaCl + MgCl2 + H2O and NaCl + SrCl2 + H2O at 25 °C. *Journal of Chemical and Engineering Data*, 308-314.

Leaist, D. G., & Kanakos, M. A. (2000). Measured and predicted ternary diffusion coefficients for concentrated aqueous LiCl + KCl solutions over a wide range of compositions. *Phys. Chem. Chem. Phys.*, 1015-1021.

Lencka, M. M., Anderko, A., Sanders, S. J., & Young, R. D. (1998). Modeling Viscosity of Multicomponent Electrolyte Solutions. *International Journal of Thermophysics*, 367-378.

Li, C., & Lee, H. (2000). Density calculation of electrolyte solutions with the solution osmotic pressure. *Chemical Engineering Science*, 655-665.

Nowlan, M.-F., Doan, T. H., & Sangster, J. (1980). Prediction of the viscosity of Mixed Electrolyte Solutions From Single-Salt Data. *The Canadian Journal opf CHemical Engineering*, 637-642.

Pazuki, G. R., Taghikhani, V., & Vossoughi, M. (2010). A New Hydration Model in Correlating the Mean Ionic Activity Coefficient and Density of Aqueous Electrolyte Solutions. *Journal of Dispersion Science and Technology*, 641-649.

Price, W. E., Mills, R., & Woolf, L. A. (1996). Use of Experimental Diffusion Coefficients To Probe Solute-Solute and Solute-Solvent Interactions in Electrolyte Solutions. *Journal of Physical Chemistry*, 1406-1410.

Renolds, J. G., & Carter, R. (2007). Density model for sodium hydroxide–sodium aluminate solutions. *Hydrometallurgy*, 233-241.

Reynolds, J. G., & Carter, R. (2008). Reconciliation of Solute Concentration Data with Water Contents and Densities of Multi-Component Electrolyte Solutions. *Journal Solution Chemistry*, 1113-1125.

Reynolds, J. G., & Carter, R. (2008). The Laliberté–Cooper density model: Self-consistency and a new method of parameterization. *Fluid Phase Equilibria*, 14-20.

Roth, R., Wolf, U., & Wolf, G. (1997). A simple method for Modelling and Prediction of the Specific Heat and Density of Aqueous Electrolyte Solutions. *Computer Coupling of Phase Diagrams and Thermochemistry*, 475-481.

Sherwood, J. D. (2011). Non-Newtonian Stress in an Electrolyte. *The Journal of Physical Chemistry B*, 1084-1088.

Van Damme, S., Dufreche, J.-F., & Deconinck, J. (2006). Influence of Ion Properties on the Equilibrium and Transport Properties of Electrolyte Solutions. *Journal of Physical Chemistry B*, 1015-1019.

Zafarani-Moattar, M. T., & Majdan-Cegincara, R. (2009). Viscosity Modeling and Prediction of Aqueous Mixed Electrolyte Solutions. *Ind. Eng. Chem. Res.*, 5833-5844.