Electronic Supplementary Information for:

Facile determination of formal transfer potentials for hydrophilic alkali metal ions at water|ionic liquid microinterfaces

Tom J. Stockmann and Zhifeng Ding[†]

Department of Chemistry, The University of Western Ontario, 1151 Richmond Street, London, Ontario, Canada, N6A 5B7

[†]To whom correspondence should be addressed. Tel: +1 519 661-2111 ext 86161; Fax: +1 519 661-3022; Email: zfding@uwo.ca

URL:http://publish.uwo.ca/~zfding/

Table of Contents

1. Theory

What follows is a brief restatement of the theory from the main paper as it relates to Comsol Model report which follows. Please note that this is not intended as a rigorous treatment but intended for those who wish to utilize/reproduce/replicate the simulation.

Simple ion transfer (IT) as detailed in equation 1, below, was simulated at a micro-ITIES housed at the tip of a micropipette.

$$i_w^{z+} \underset{k_b}{\overset{k_f}{\rightleftarrows}} i_o^{z+} \qquad \text{(S1)}$$

where ion *i* with charge z+ transfers from aqueous to RTIL. Two 'Diffusion' multiphysics were employed to represent the concentration of *i* in either phase or subdomain; they were turned off with in their opposing phase. Please note that the micropippette geometry is included in the Comsol Model report which follows and in Figure 1. Diffusion within each phase was performed using Fick's laws of diffusion and equation 2:

$$\frac{\partial c_\alpha^i(r,z,t)}{\partial t} = D_\alpha^i\left(\frac{\partial^2 c_\alpha^i(r,z,t)}{\partial r^2} + \frac{1}{r}\frac{\partial c_\alpha^i(r,z,t)}{\partial r} + \frac{\partial^2 c_\alpha^i(r,z,t)}{\partial z^2}\right) = D_\alpha^i \nabla c_\alpha^i(r,z,t) = 0 \qquad \text{(S2)}$$

such that c_α^i and D_α^i are the concentration and diffusion coefficient of species *i* in phase α; ∇, or del, is the gradient or vector operator – shown here in cylindrical coordinates. In the simulation code, the constants Caq and Corg where used to define the initial (or bulk) concentrations in the aqueous and RTIL phases with values of 5 and 0 mol·m^{-3} (or mM), respectively. Similarly, the diffusion coefficients were defined using the constants Daq and Dorg with initial values of 2.0×10^{-9} and 0.5×10^{-13} m^2·s^{-1} for the water and RTIL phases respectively. The forward and reverse rate constants k_f and k_b, respectively, were treated using Butler-Volmer kinetics through equations 3 and 4 below:

$$k_f = k^o \exp\left(-\alpha f(\Delta_{RTIL}^w \phi - \Delta_{RTIL}^w \phi^{o'})\right) \quad \text{(S4)}$$

$$k_b = k^o \exp\left((1-\alpha) f(\Delta_{RTIL}^w \phi - \Delta_{RTIL}^w \phi^{o'})\right) \quad \text{(S5)}$$

here, k^o is the standard rate constant, α is the transfer coefficient, $\Delta_o^w \phi$ is the applied Galvani potential difference across the interface, $\Delta_o^w \phi^{o'}$ is the formal transfer potential, and $f = z_i$F/RT; z_i is the charge of species *i*, F is Faraday's constant (96485.33 C·mol^{-1}), R is the gas constant (8.314 kJ·mol^{-1}·K^{-1}), and T is temperature in Kelvin (333.15 K). In the simulation the constants designations for F, R, and T remained the same, however, k^o, α, and z_i where called k0, alpha, and n1 with values of 5×10^{-6} m·s^{-1}, 0.5, and 1+, respectively. Equations 4 and 5 were included in the boundary conditions for the ITIES under 'Mass Transfer Coefficient'.

In order to fulfill the BV kinetic theory the concentration must be known at the interface, of for $c_\alpha^i(r,0,t)$. This was performed through a simple Nernst relationship denoted in the user defined equations as nernst1:

$$\frac{c_{RTIL}^i}{c_w^i} = \exp\left(\Delta_{RTIL}^w \phi - \Delta_{RTIL}^w \phi_{i^{z+}}^{o'}\right)$$

This was implemented within the interfacial boundary condition for boundary 5, c_b.

A special note concerning the formal transfer potential, variable E0, used in the simulation: a value of 0.100 V was entered, however, the CV generated possessed a calculated formal transfer potential of 0.252 V. Interestingly, when the diffusion coefficient, Dorg, is increased, the resultant half-wave potential shifts towards that of the input value: 0.100 V. With regards to the working curve utilized in the simulation, this should not be an issue since both axis were normalized, however, we are currently investigating whether this is an artifact of the simulation or a legitimate, liquid|liquid phenomenon.

Moving forward, the applied potential was a triangular wave function which can be found in the 'Scalar Expressions' and was input as a variable into equations 4 and 5.

$$\Delta^{w}_{RTIL}\phi = \Delta^{w}_{RTIL}\phi_i + \frac{2\left(\Delta^{w}_{RTIL}\phi_f - \Delta^{w}_{RTIL}\phi_i\right)}{\pi}\sinh\left\{\sin\left(\frac{\pi\nu t}{2\left(\Delta^{w}_{RTIL}\phi_f - \Delta^{w}_{RTIL}\phi_i\right)}\right)\right\} \quad \text{(S6)}$$

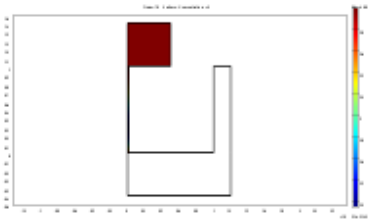
Where $\Delta^{w}_{RTIL}\phi_i$, $\Delta^{w}_{RTIL}\phi_f$, ν, and t are the initial potential, final potential, scan rate (in $V \cdot s^{-1}$), and time or Ei, Ef, nu, and t in the simulation, respectively. In this way, the potential difference was applied at the interface.

The current was integrated at the interface through a 'Integration Coupling Variable' and the following expression:

$$I = 2\pi z_i F \int (-D^i_\alpha \nabla c^i_\alpha(r,z,t))rdr \quad (6)$$

Please note that, in Comsol, $\int\left(-D^i_\alpha \nabla c^i_\alpha\right)rdr$, is represented simply as ndflux_c1_di.

2. COMSOL Model Report



1. Table of Contents

- Variables

2. Model Properties

Property	Value
Model name	
Author	
Company	
Department	
Reference	
URL	
Saved date	Jun 21, 2012 12:01:12 PM
Creation date	Jun 30, 2005 8:09:00 PM
COMSOL version	COMSOL 3.5.0.603

File name: D:\LL_EC1_UO2\Comsol35\micro-ITIES\2012\011912_wRTIL_IT\2012_0621_formatted.mph

Application modes and modules used in this model:

- Geom1 (Axial symmetry (2D))
  - Diffusion (Chemical Engineering Module)
  - Diffusion (Chemical Engineering Module)

3. Constants

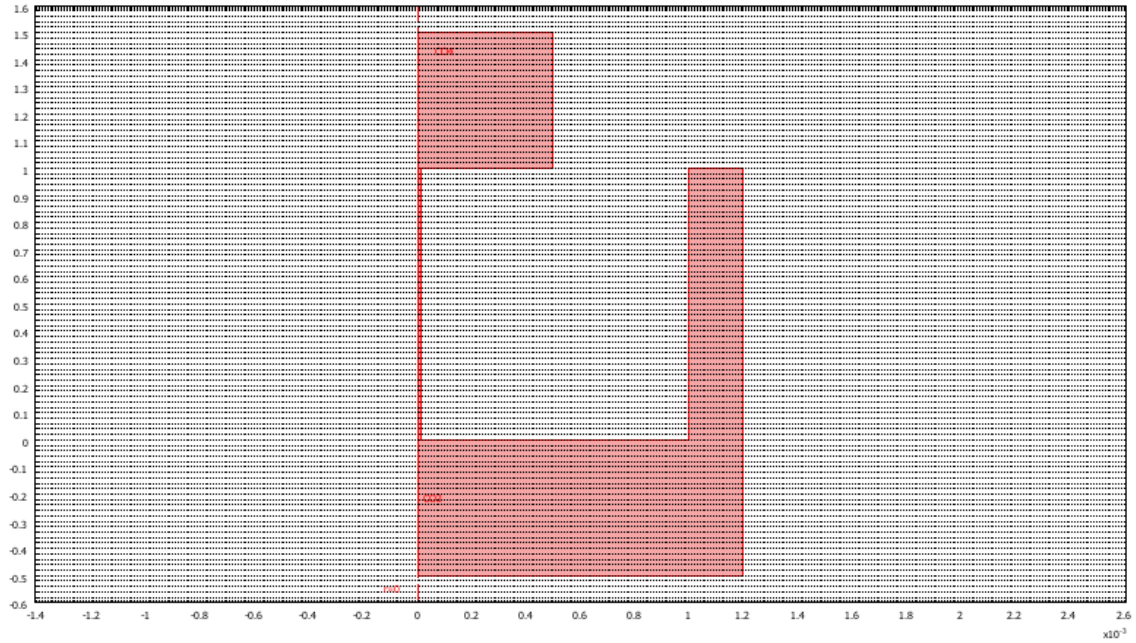
Name	Expression	Value	Description
R	8.314	8.314	gas constant
T	333.15	333.15	temperature in Kelvin
F	96485.33	96485.33	Faraday's constant
fara	F/(R*T)	34.83465	
nu	0.02	0.02	scan rate
Ei	0	0	initial potential
Ef	0.7	0.7	final potential
Daq	2e-9	2e-9	diffusion coefficient in the aqueous phase
Dorg	0.5e-13	5e-14	diffusion coefficient in the RTIL phase
Caq	5	5	[i](aq)
Corg	0	0	[i](RTIL)
n1	1	1	charge of ion i
E0	0.1	0.1	formal transfer potential

alpha	0.5	0.5	transfer coefficient
k0	5e-6	5e-6	standard rate constant

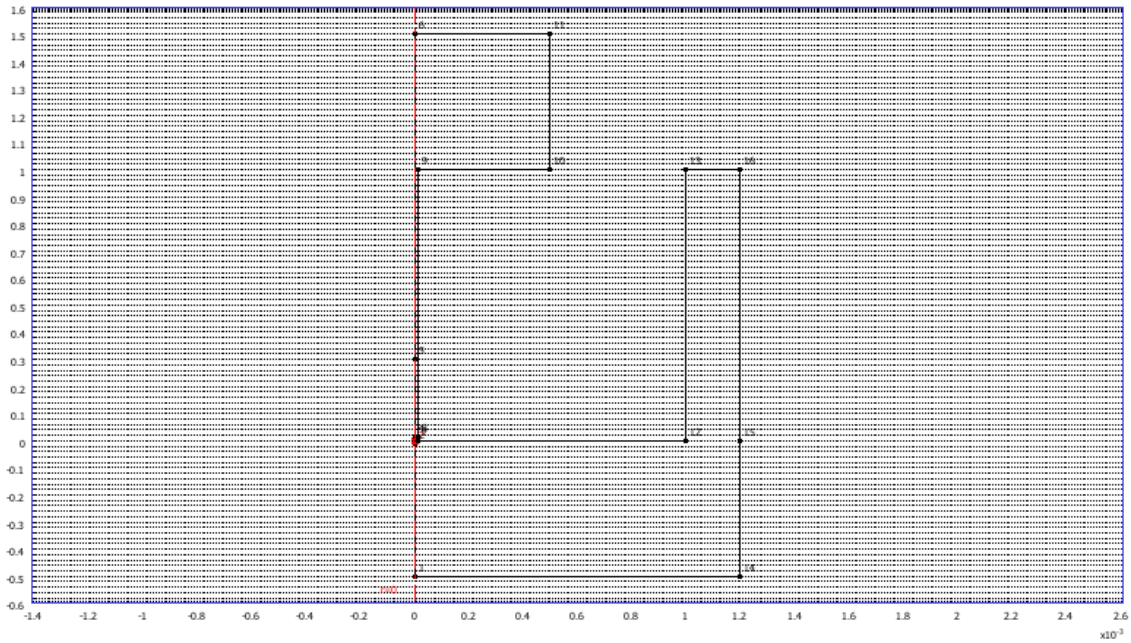
4. Geometry

Number of geometries: 1

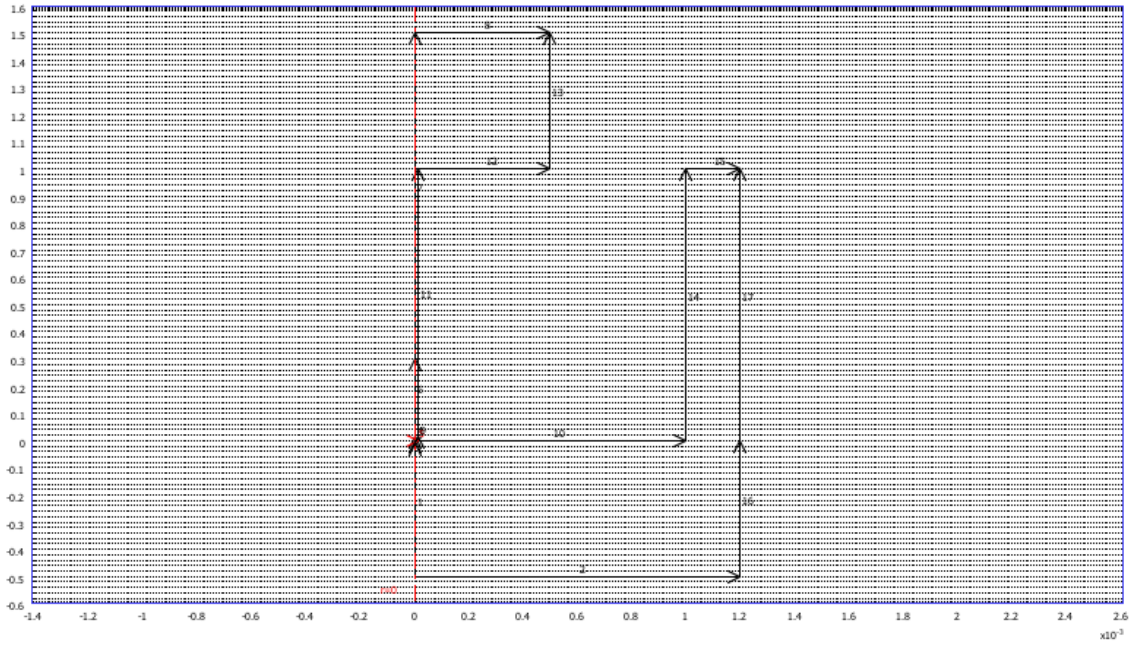
4.1. Geom1



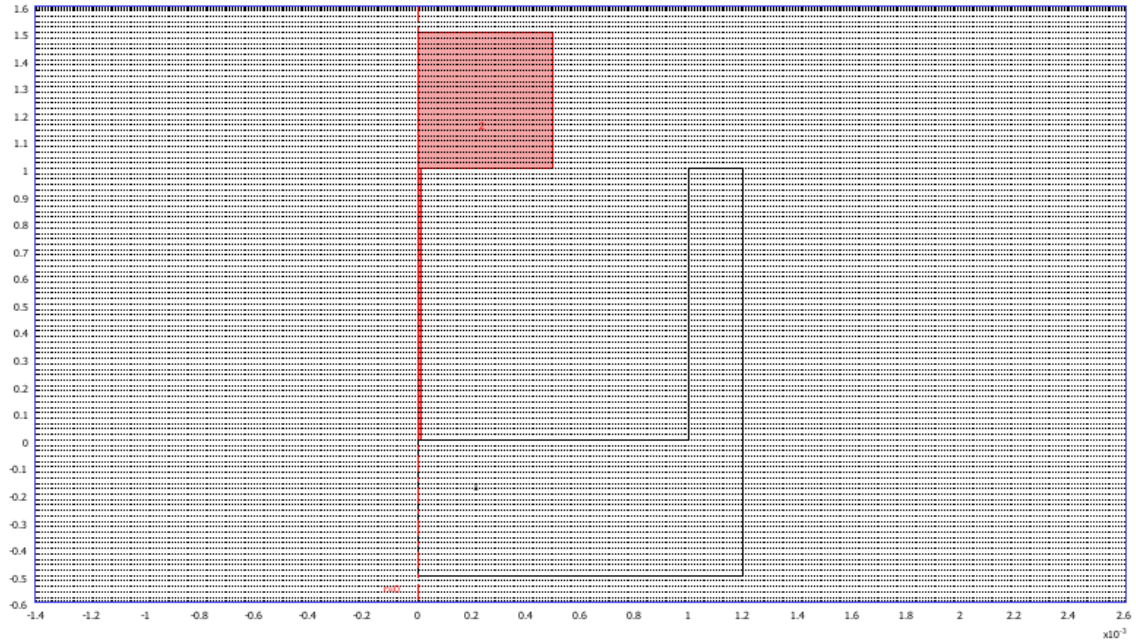
4.1.1. Point mode



4.1.2. Boundary mode



4.1.3. Subdomain mode



5. Geom1

Space dimensions: Axial symmetry (2D)

Independent variables: r, phi, z

5.1. Scalar Expressions

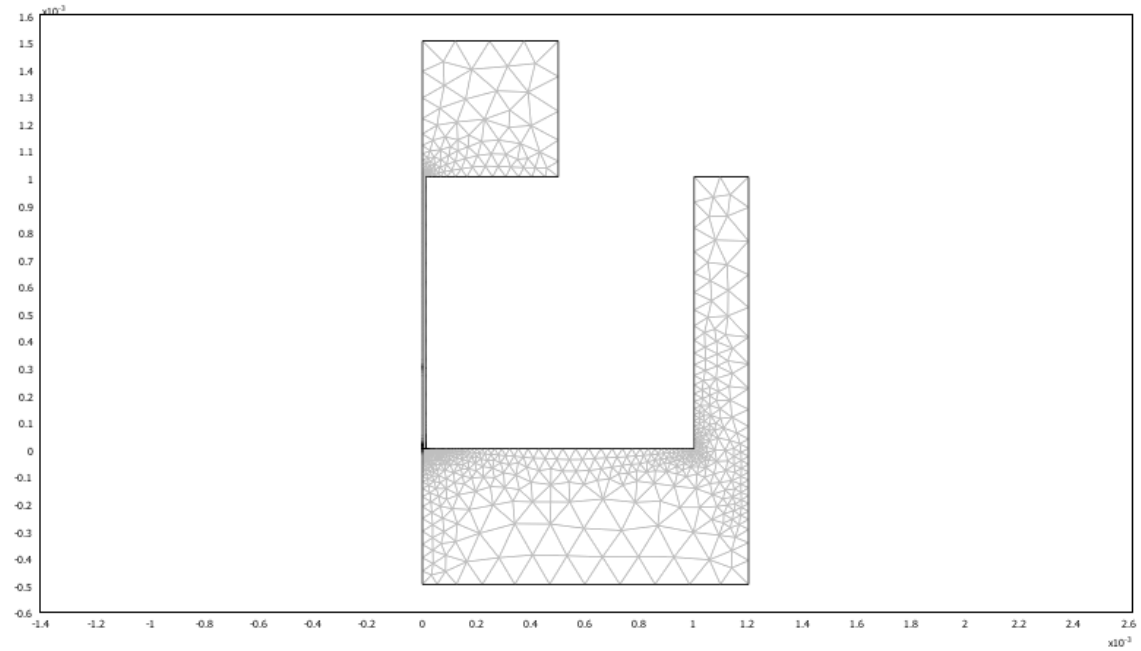
Name	Expression	Description
E_swp	Ei+2*((Ef-Ei)/pi)*asin(sin((pi*nu*t)/(2*(Ef-Ei))))	triangular wave form
kf1	k0*exp(-1*alpha*(n1*fara*(E_swp-E0)))	forward rate equation
kb1	k0*exp((1-alpha)*(n1*fara*(E_swp-E0)))	reverse rate equation
nernst1	exp(n1*fara*(E_swp-E0))	
E_swp_void	(Ei+nu*t)*(t<=25)+(Ef-1*nu*(t-25))*(t>25)*0	

5.2. Mesh

5.2.1. Mesh Statistics

Number of degrees of freedom	13617
Number of mesh points	3545
Number of elements	6328
Triangular	6328

Quadrilateral	0
Number of boundary elements	822
Number of vertex elements	16
Minimum element quality	0.76
Element area ratio	0



5.3. Application Mode: Diffusion (di2)

Application mode type: Diffusion (Chemical Engineering Module)

Application mode name: di2

5.3.1. Application Mode Properties

Property	Value
Default element type	Lagrange - Quadratic
Analysis type	Transient
Equilibrium assumption	Off
Frame	Frame (rz)
Weak constraints	Off
Constraint type	Ideal

5.3.2. Variables

Dependent variables: c2

Shape functions: shlag(2,'c2')

Interior boundaries not active

5.3.3. Boundary Settings

Boundary	4, 6-8, 13	5	9, 11-12
Type	Axial symmetry	Flux	Insulation/Symmetry
Mass transfer coefficient (kc)	0	**kb1**	0
Bulk concentration (cb)	0	**c1/nernst1**	0
Concentration (c0)	0	**c1/nernst**	0

5.3.4. Subdomain Settings

Subdomain	2
Shape functions (shape)	**shlag(2,'c2')**
Integration order (gporder)	**4**
Constraint order (cporder)	**2**
Diffusion coefficient (D)	**Daq**
Subdomain initial value	2
Concentration, c2 (c2)	Caq

5.4. Application Mode: Diffusion (di)

Application mode type: Diffusion (Chemical Engineering Module)

Application mode name: di

5.4.1. Application Mode Properties

Property	**Value**
Default element type	Lagrange - Quadratic
Analysis type	Transient
Equilibrium assumption	Off
Frame	Frame (rz)
Weak constraints	On
Constraint type	Ideal

5.4.2. Variables

Dependent variables: c1

Shape functions: shlag(2,'lm1'), shlag(2,'c1')

Interior boundaries not active

5.4.3. Boundary Settings

Boundary	1, 3	2, 15-17	5
Type	Axial symmetry	Concentration	Flux
Mass transfer coefficient (kc)	0	0	**kf1**
Bulk concentration (cb)	0	0	**c2*nernst1**
Concentration (c0)	0	**Corg**	**c2*nernst**
Integration order (wcgporder)	**4**	**4**	**4**

Boundary	10, 14
Type	Insulation/Symmetry
Mass transfer coefficient (kc)	0
Bulk concentration (cb)	0
Concentration (c0)	0
Integration order (wcgporder)	**4**

5.4.4. Subdomain Settings

Subdomain	1
Shape functions (shape)	**shlag(2,'lm1') shlag(2,'c1')**
Integration order (gporder)	**4**
Constraint order (cporder)	**2**
Diffusion coefficient (D)	**Dorg**

Subdomain initial value	1
Concentration, c1 (c1)	Corg

6. Materials/Coefficients Library

6.1. Water

Parameter	**Value**
Heat capacity at constant pressure (C)	Cp(T[1/K])[J/(kg*K)]
Speed of sound (cs)	cs(T[1/K])[m/s]
Dynamic viscosity (eta)	eta(T[1/K])[Pa*s]
Ratio of specific heats (gamma)	1.0

Thermal conductivity (k)	k(T[1/K])[W/(m*K)]
Kinematic viscosity (nu0)	nu0(T[1/K])[m^2/s]
Density (rho)	rho(T[1/K])[kg/m^3]
Electric conductivity (sigma)	5.5e-6[S/m]

6.1.1. Piecewise Analytic Functions

6.1.1.1. Function: rho(T)

Type: Polynomial

x_{start}	x_{end}	f(x)
273.15	553.75	0 8.38466135E+02 1 1.40050603E+00 2 -3.01123760E-03 3 3.71822313E-07

6.1.1.2. Function: Cp(T)

Type: Polynomial

x_{start}	x_{end}	f(x)
273.15	553.75	0 1.20101471E+04 1 -8.04072879E+01 2 3.09866854E-01 3 -5.38186884E-04 4 3.62536437E-07

6.1.1.3. Function: eta(T)

Type: Polynomial

x_{start}	x_{end}	f(x)
273.15	413.15	0 1.3799566804E+00 1 -2.1224019151E-02 2 1.3604562827E-04 3 -4.6454090319E-07 4 8.9042735735E-10 5 -9.0790692686E-13 6 3.8457331488E-16
413.15	553.75	0 4.01235783E-03 1 -2.10746715E-05 2 3.85772275E-08 3 -2.39730284E-11

6.1.1.4. Function: nu0(T)

Type: Polynomial

x_{start}	x_{end}	f(x)
273.15	413.15	0 1.3903230155E-03 1 -2.1392545737E-05 2 1.3717187260E-07 3 -4.6851166702E-10 4 8.9823339887E-13 5 -9.1602964729E-16 6 3.8807189536E-19
413.15	553.75	0 1.36222834E-06 1 -4.45388688E-09 2 4.04364450E-12

6.1.1.5. Function: k(T)

Type: Polynomial

x_{start}	x_{end}	f(x)
273.15	553.75	0 -8.69083936E-01 1 8.94880345E-03 2 -1.58366345E-05 3 7.97543259E-09

6.1.2. Interpolation Functions

6.1.2.1. Interpolation Function: cs

Interpolation method: Piecewise Cubic

Data source type: Table

x	f(x)
273	1403
278	1427
283	1447
293	1481
303	1507
313	1526
323	1541
333	1552
343	1555
353	1555
363	1550
373	1543

7. Integration Coupling Variables

7.1. Geom1

7.1.1. Source Boundary: 5

Name	Value
Variable name	Ibar2
Expression	ndflux_c2_di2*2*pi*r*n1*F
Order	4
Global	Yes

7.1.2. Source Boundary: 5

Name	Value
Variable name	Ibar1
Expression	-ndflux_c1_di*2*pi*r*n1*F
Order	4
Global	Yes

7.1.3. Source Boundary: 5

Name	Value
Variable name	ct_flux
Expression	2*pi*r*F*(n1*ndflux_c2_di2-n2*ndflux_c4_di4+n3*ndflux_c6_di6)
Order	4
Global	Yes

8. Solver Settings

Solve using a script: off

Analysis type	Transient
Auto select solver	On
Solver	Time dependent
Solution form	General
Symmetric	Off
Adaptive mesh refinement	Off
Optimization/Sensitivity	Off
Plot while solving	Off

8.1. Direct (UMFPACK)

Solver type: Linear system solver

Parameter	Value
Pivot threshold	0.1
Memory allocation factor	0.7

8.2. Time Stepping

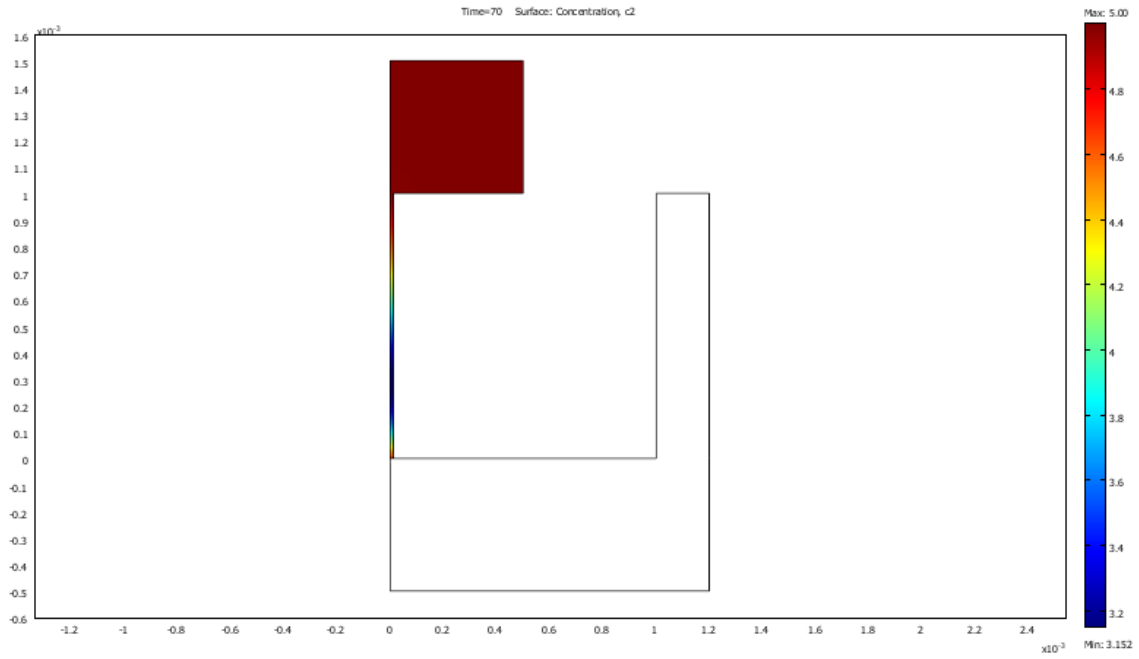
Parameter	Value

Times	range(0,0.1,70)
Relative tolerance	1e-4
Absolute tolerance	1e-4
Times to store in output	Specified times
Time steps taken by solver	Free
Maximum BDF order	5
Singular mass matrix	Maybe
Consistent initialization of DAE systems	Backward Euler
Error estimation strategy	Include algebraic
Allow complex numbers	Off

8.3. Advanced

Parameter	**Value**
Constraint handling method	Elimination
Null-space function	Automatic
Automatic assembly block size	On
Assembly block size	5000
Use Hermitian transpose of constraint matrix and in symmetry detection	On
Use complex functions with real input	Off
Stop if error due to undefined operation	On
Store solution on file	Off
Type of scaling	Automatic
Manual scaling	
Row equilibration	On
Manual control of reassembly	Off
Load constant	On
Constraint constant	On
Mass constant	On
Damping (mass) constant	On
Jacobian constant	On
Constraint Jacobian constant	On

9. Postprocessing



10. Variables

10.1. Boundary

10.1.1. Boundary 1-3, 10, 14-17

Name	Description	Expression
ndflux_c2_di2	Normal diffusive flux, c2	
ndflux_c1_di	Normal diffusive flux, c1	nr_di * dflux_c1_r_di+nz_di * dflux_c1_z_di

10.1.2. Boundary 4, 6-9, 11-13

Name	Description	Expression
ndflux_c2_di2	Normal diffusive flux, c2	nr_di2 * dflux_c2_r_di2+nz_di2 * dflux_c2_z_di2
ndflux_c1_di	Normal diffusive flux, c1	

10.1.3. Boundary 5

Name	Description	Expression
ndflux_c2_di2	Normal diffusive flux, c2	nr_di2 * dflux_c2_r_di2+nz_di2 * dflux_c2_z_di2
ndflux_c1_di	Normal diffusive flux, c1	nr_di * dflux_c1_r_di+nz_di * dflux_c1_z_di

10.2. Subdomain

10.2.1. Subdomain 1

Name	Description	Expression
grad_c2_r_di2	Concentration gradient, c2, r component	
dflux_c2_r_di2	Diffusive flux, c2, r component	
grad_c2_z_di2	Concentration gradient, c2, z component	
dflux_c2_z_di2	Diffusive flux, c2, z component	
grad_c2_di2	Concentration gradient, c2	
dflux_c2_di2	Diffusive flux, c2	
grad_c1_r_di	Concentration gradient, c1, r component	c1r
dflux_c1_r_di	Diffusive flux, c1, r component	-Drr_c1_di * c1r-Drz_c1_di * c1z
grad_c1_z_di	Concentration gradient, c1, z component	c1z
dflux_c1_z_di	Diffusive flux, c1, z component	-Dzr_c1_di * c1r-Dzz_c1_di * c1z
grad_c1_di	Concentration gradient, c1	sqrt(grad_c1_r_di^2+grad_c1_z_di^2)
dflux_c1_di	Diffusive flux, c1	sqrt(dflux_c1_r_di^2+dflux_c1_z_di^2)

10.2.2. Subdomain 2

Name	Description	Expression
grad_c2_r_di2	Concentration gradient, c2, r component	c2r
dflux_c2_r_di2	Diffusive flux, c2, r component	-Drr_c2_di2 * c2r-Drz_c2_di2 * c2z
grad_c2_z_di2	Concentration gradient, c2, z component	c2z
dflux_c2_z_di2	Diffusive flux, c2, z component	-Dzr_c2_di2 * c2r-Dzz_c2_di2 * c2z
grad_c2_di2	Concentration gradient, c2	sqrt(grad_c2_r_di2^2+grad_c2_z_di2^2)
dflux_c2_di2	Diffusive flux, c2	sqrt(dflux_c2_r_di2^2+dflux_c2_z_di2^2)
grad_c1_r_di	Concentration gradient, c1, r component	
dflux_c1_r_di	Diffusive flux, c1, r component	
grad_c1_z_di	Concentration gradient, c1, z component	
dflux_c1_z_di	Diffusive flux, c1, z component	
grad_c1_di	Concentration gradient, c1	
dflux_c1_di	Diffusive flux, c1	