Environmental Science & Technology

Supporting Information for

Numerical solution of the Polanyi-DR isotherm in linear driving force models

David M. Lorenzetti* and Michael D. Sohn

Indoor Environment Department

Lawrence Berkeley National Laboratory, Berkeley CA, USA

E-mail: dmlorenzetti@lbl.gov

Phone: +1 (510)486-4562. Fax: +1 (510)486-6658

The supporting information includes mathematical and implementation details of the solution methods tested. Equations from the main paper are referenced as, for example, Eq. (M5).

Seven pages. No figures. No tables.

This document was prepared on 29 September 2011.

Derivatives

Many techniques for solving Eq. (M5) require the derivative

$$r' \equiv \frac{dr}{dc} = k_c + k_q \frac{df}{dc} \,. \tag{1}$$

Since $df/dc \geq 0$, it follows $r' > 0$. Thus if a trial solution c_+ has a residual $r_+ > 0$, then c_+ is more positive than c_s. Conversely, $r_+ < 0$ means c_+ is too negative.

Given the c_s that sets $r = 0$, Eq. (M4) gives $w = k_c(c_b - c_s)$. In addition to w, the transport solver may require the partial derivatives

$$\frac{\partial w}{\partial c_b} = k_c \left(1 - \frac{\partial c_s}{\partial c_b}\right) \qquad \text{and} \qquad \frac{\partial w}{\partial q_b} = -k_c \frac{\partial c_s}{\partial q_b} \,. \tag{2}$$

Differentiating Eq. (M5) implicitly (i.e., holding $r = 0$) gives

$$\frac{\partial c_s}{\partial c_b} = \frac{k_c}{r'} \qquad \text{and} \qquad \frac{\partial c_s}{\partial q_b} = \frac{k_q}{r'} \,. \tag{3}$$

These results hold for any isotherm f used in the linear driving force model.

Regula falsi (rf)

The regula falsi, or false position, method places c_+ where the chord connecting the bracket's bounding points crosses $r = 0$ (*1*):

$$c_+ = c_\ell - \frac{r_\ell (c_r - c_\ell)}{r_r - r_\ell} \,. \tag{4}$$

Regula falsi converges from the right on this problem. The Polanyi-DR isotherm makes $d^2r/dc^2 < 0$, which ensures that the chord connecting the bounds lies beneath the residual curve, and hence crosses $r = 0$ to the right of the solution.

To combat this one-sided convergence, variation **rfa** applies Aitken's delta-squared method (*1*).

Suppose regula falsi produces a sequence c_n, c_{n+1}, c_{n+2} of estimated solutions. Aitken, assuming the errors follow a geometric sequence, replaces c_{n+2} with

$$c_+ = c_n - \frac{(c_{n+1} - c_n)^2}{c_{n+2} - 2c_{n+1} + c_n} \,. \tag{5}$$

When converging from the right, a geometric progression should produce a positive denominator; if not, we set $c_+ = c_{n+2}$. The right-hand bound of the resulting bracket becomes c_n for the next Aitken extrapolation.

Newton–Raphson (nr)

This method starts at a known point on the residual curve, then follows the tangent line to where it crosses $r = 0$ (*1*):

$$c_+ = c_y - \frac{r_y}{r'_y} \,, \tag{6}$$

where c_y is one of c_ℓ or c_r, and r_y the residual evaluated there.

Newton–Raphson converges from the left on this problem. Suppose c_y lies to the left of the solution. Then $r_y < 0$, and the method follows the tangent toward increasing c. However, since the slope of the residual curve becomes less positive as c increases, the tangent crosses $r = 0$ to the left of the desired solution. Conversely, if $c_y = c_r$, the method follows a too-shallow tangent line, and again places c_+ to the left of c_s.

Because of this behavior, for simplicity we always take $c_y = c_\ell$ in Eq. (6). In addition, variation **nra** applies Aitken acceleration. Converging from the left implies a negative denominator in Eq. (5); otherwise, we take $c_+ = c_{n+2}$.

Quadratic fit (qf$*$)

Fit the quadratic model $\widehat{r} = A\Delta c^2 + B\Delta c + C$, where $\Delta c = c - c_y$, to points (c_y, r_y) and (c_z, r_z), and to the slope at c_y. Then

$$A = \frac{\Delta r_z}{\Delta c_z^2} - \frac{r_y'}{\Delta c_z} \qquad \text{and} \qquad B = r_y' \qquad \text{and} \qquad C = r_y\,. \tag{7}$$

The quadratic formula gives $c_+ - c_y = (-B \pm \sqrt{B^2 - 4AC})/(2A)$. Let c_y and c_z be the bracket points, in any order. Then $A < 0$ and $B^2 - 4AC > 0$. Furthermore, adding the square root gives the smaller-magnitude root of $\widehat{r} = 0$, placing c_+ in the bracket. However, finite-precision effects make this formulation unstable. Therefore we first find the larger-magnitude root, then use the fact that the product of the two roots is C/A. This gives

$$c_+ = c_y - \frac{2 r_y \Delta c_z}{B\Delta c_z + \text{sign}\{\Delta c_z\} \sqrt{(B\Delta c_z)^2 - 4(r_y \Delta c_z)(A\Delta c_z)}}\,. \tag{8}$$

We tested several variations on this method: **qfl** always picks $c_y = c_\ell$; **qfr** uses $c_y = c_r$; **qfs** uses the bound with the smaller residual magnitude; and **qfu** takes c_y as the bound that was most recently updated.

Inverse quadratic (iq$*$)

This family of methods models c as a quadratic function of r, then takes c_+ as the concentration where $r = 0$.

Method **iq3** fits to the left and right bounds, and to the most recently replaced bound, c_p, giving

$$c_+ = \frac{c_\ell (r_p r_r)(r_p - r_r) - c_r (r_p r_\ell)(r_p - r_\ell) + c_p (r_r r_\ell)(r_r - r_\ell)}{(r_p - r_r)(r_p - r_\ell)(r_r - r_\ell)}\,. \tag{9}$$

At the first iteration (i.e., before a c_p exists), or if finite-precision effects produce a zero in the denominator, we make c_+ the midpoint from Eq. (M11).

Method **iqs**$*$ fits to the slope at c_y, as well as to bounds c_y and c_z:

$$c_+ = c_y - \left[\frac{r_z}{r_y'} - \Delta c_z \left(\frac{r_y}{\Delta r_z}\right)\right]\left(\frac{r_y}{\Delta r_z}\right), \tag{10}$$

where $\Delta r_z = r_z - r_y$. Variations **iqsl**, **iqsr**, **iqss**, and **iqsu** pick c_y following the same notation as used for the quadratic fits. Note that taking $c_y = c_r$ may place c_+ to the left of the bracket, in which case we apply bisection.

Inverse cubic (icu$*$)

Consider a cubic model $\widehat{c} = A\Delta r^3 + B\Delta r^2 + C\Delta r + D$, where $\Delta r = r - r_\ell$. Fitting this model to the points and slopes at c_ℓ and c_r, then evaluating at $r = 0$, gives method **icub**:

$$c_+ = c_\ell - \frac{r_\ell}{r_\ell'} + \left(\frac{r_\ell}{\Delta r_r}\right)^2 \left(\Delta c_r \left[\frac{3r_r - r_\ell}{\Delta r_r}\right] - \frac{2r_r - r_\ell}{r_\ell'} - \frac{r_r}{r_r'}\right). \tag{11}$$

Interpreting the rightmost term as a correction to Eq. (6), and recalling that Newton–Raphson converges from the left, we require a positive correction. Otherwise, we take the Newton–Raphson step.

Nonpositive corrections frequently result from a nonpositive coefficient B in the cubic fit making the model $\widehat{c}$ bend to the left of c_ℓ. The reduced inverse cubic method, **icur**, prevents this by forcing $B = 0$. Fitting to the two bounds, and to the slope at c_ℓ, gives

$$c_+ = c_\ell - r_\ell \left(\frac{1-\alpha}{r_\ell'} + \alpha\frac{\Delta c_r}{\Delta r_r}\right) \qquad \text{where} \qquad \alpha = \left(\frac{r_\ell}{\Delta r_r}\right)^2. \tag{12}$$

The result is a weighted average of the Newton–Raphson and regula falsi steps, favoring the former as the magnitude of r_ℓ falls relative to r_r. To avoid slow convergence from the right, we force $\alpha \le 0.9$.

Rational function (rat∗)

Model the residuals as a rational function

$$\widehat{r} = r_y + \frac{F_1 \Delta c}{1 + F_2 \Delta c}, \tag{13}$$

where $\Delta c = c - c_y$. Fitting to the bracket points, and to the slope at c_y, gives $F_1 = r_y'$ and

$$F_2 = \frac{r_y' \Delta c_z - \Delta r_z}{\Delta c_z \Delta r_z}. \tag{14}$$

In exact arithmetic, $F_2 > 0$. Setting $\widehat{r} = 0$ gives

$$c_+ = c_y - \frac{r_y}{r_y'} \left(\frac{\Delta c_z \Delta r_z}{\Delta c_z \Delta r_z + r_y \Delta c_z - (r_y / r_y') \Delta r_z} \right), \tag{15}$$

i.e., a scaled Newton–Raphson step. Variations **ratl**, **ratr**, **ratu**, and **rats** choose c_y following the same notation as used for the quadratic fits.

Ridder's method (rid)

Designed to factor out exponential behavior from a residual function, Ridder's method (*1*), applied to Eq. (M5), sets

$$c_+ = c_m - (c_m - c_\ell) \frac{r_m}{\sqrt{r_m^2 - r_\ell r_r}}, \tag{16}$$

where c_m is the bracket midpoint of Eq. (M11). Note this method requires two residual evaluations per iteration.

Precomputed values

For speed, the implementations precompute $1/c_{\max}$, then calculate $\ln\{c_{\max}/c\}$ as $-\ln\{c \cdot (1/c_{\max})\}$. This replaces a relatively expensive division with a multiplication, at a slight cost in accuracy (on average, about one decimal digit of the machine's precision). Numerical tests show that precomputing $\ln\{c_{\max}\}$, for use in the alternate expression $\ln\{c_{\max}\} - \ln\{c\}$, induces average errors slightly larger (by a factor of about 1.4) than those due to inverting $c_{\max}$.

Some algorithms also cache values such as $1/r'_\ell$, for re-use when a bound does not change between iterations. We made these decisions case-by-case, based on numerical testing.

To avoid loss of precision when finding small differences between large numbers, we do not precompute the constant term $k_c c_b + k_q q_b$ in Eq. (M5).

References

(1) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes: The Art of Scientific Computing*, 3rd ed.; Cambridge University Press, 2007.