Comparison of One-Sided and Diffusion-Limited Evaporation Models for Thin Liquid Droplets (Supplementary Material)

Christopher Larsson and Satish Kumar

Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA

Here we describe the procedure for solving the axisymmetric Laplace's equation with the hybrid spectral-finite-difference method described in §3. For ease of use, we present this method for a more general problem than that considered in §3.

We wish to numerically solve for the concentration c_g governed by

$$0 = \boldsymbol{\nabla}^2 c_g = \frac{\partial^2 c_g}{\partial z^{*2}} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial c_g}{\partial r}\right) \tag{1}$$

on the the semi-infinite domain $(r, z^*) \in (0, d) \times (0, \infty)$. The concentration c_g is subject to the symmetric boundary condition $c_g(r) = c_g(-r)$ as well as $c_g(r = d) = 0$ and $c_g \to 0$ as $z^* \to \infty$. At $z^* = 0$, it takes on the value $c_g(r, z^* = 0) = f(r)$. Since equation (1) must be solved at each time step, the goal of this numerical method is to reduce equation (1) to a matrix equation that can be quickly and efficiently solved. Note that we do not explicitly include time-dependence in this derivation because it can be encapsulated in the boundary data f.

We begin by writing c_g as an expansion of Laguerre functions L_n,

$$c_g(r, z^*) \approx \sum_{n=0}^{M-1} a_n(r) L_n(z^*), \tag{2}$$

where the coefficients $a_n(r)$ vary radially. For now, we assume that there is some set of coefficients $a_n''(r)$ such that

$$\frac{\partial^2 c_g}{\partial z^{*2}} = \sum_{n=0}^{M-1} a_n(r)\frac{\partial^2 L_n}{\partial z^{*2}} = \sum_{n=0}^{M-1} a_n''(r) L_n(z^*). \tag{3}$$

Then, substituting equation (2) into equation (1) and applying orthogonality of the Laguerre functions gives, for each n,

$$a_n''(r) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial a_n}{\partial r}\right) = 0. \tag{4}$$

We first determine a_n'', and then show how we discretize in r.

The Laguerre functions are defined as

$$L_n(x) = e^{-x/2}\ell_n(x), \tag{5}$$

where $\ell_n(x)$ is the n-th Laguerre polynomial which satisfies

$$\frac{\partial \ell_n}{\partial x} = -\sum_{k=0}^{n-1} \ell_k(x). \tag{6}$$

Thus, we have

$$\begin{aligned} \frac{\mathrm{d}L_n}{\mathrm{d}x} &= \frac{\mathrm{d}}{\mathrm{d}x}\left(e^{-x/2}\ell_n(x)\right) && (7)\\ &= -\frac{1}{2}e^{-x/2}\ell_n(x) - e^{-x/2}\sum_{k=0}^{n-1}\ell_k(x) && (8)\\ &= -\frac{1}{2}L_n(x) - \sum_{k=0}^{n-1} L_k(x). && (9) \end{aligned}$$

Defining the lower triangular matrix $\mathbf{D}$ and vector of Laguerre functions $\mathbf{L}$ as

$$(\mathbf{D})_{i,j} = \begin{cases} 0 & i \geq j \\ 1 & i < j \end{cases}, \qquad (\mathbf{L})_i = L_i(x), \tag{10}$$

we can use equation (9) to write

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}x} = \left(-\frac{1}{2}\mathbf{I} - \mathbf{D}\right)\mathbf{L}, \tag{11}$$

where $\mathbf{I}$ is the identity matrix. Thus,

$$\frac{\mathrm{d}^2\mathbf{L}}{\mathrm{d}x^2} = \left(-\frac{1}{2}\mathbf{I} - \mathbf{D}\right)^2 \mathbf{L} = \underbrace{\left(\frac{1}{4}\mathbf{I} + \mathbf{D} + \mathbf{D}^2\right)}_{\mathbf{D}_z}\mathbf{L}. \tag{12}$$

With a vector of coefficients $\mathbf{a}(r)$, we have $c_g = \mathbf{a}^T\mathbf{L}$ and thus

$$\frac{\partial^2 c_g}{\partial z^{*2}} = \mathbf{a}^T\mathbf{D}_z\mathbf{L}, \tag{13}$$

so the coefficients a_n'' are given in vector form by $\mathbf{a}'' = \left(\mathbf{a}^T\mathbf{D}_z\right)^T$.

We now discretize each coefficient $a_n(r)$ into its values at a discrete set of N nodes $\{r_j\}$ placed in the center of cells. We then apply a positivity-preserving second-order centered finite-difference method that gives

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial a_n}{\partial r}\right)\bigg|_{r_j} = \frac{2}{r_j \Delta r_j}\left[\alpha_j a_n(r_{j-1}) - (\alpha_j + \alpha_{j+1})\, a_n(r_j) + \alpha_{j+1} a_n(r_{j+1})\right], \tag{14}$$

where

$$\alpha_j = \frac{r_{j-1/2}}{\Delta r_j + \Delta r_{j-1}}, \tag{15}$$

$r_{j-1/2}$ is the position of the boundary between cells j and $j-1$, and Δr_j is the width of cell j. Note that the boundary conditions in r define the values of ghost nodes needed to evaluate the above expressions near the boundaries of the domain. Defining the $N \times M$ coefficient matrix $(\mathbf{A})_{i,j} = a_j(r_i)$ and r-derivative matrix $\mathbf{D}_r$ as

$$(\mathbf{D}_r)_{i,j} = \frac{2}{r_i \Delta r_i} \begin{cases} \alpha_i & j = i-1 \\ -(\alpha_i + \alpha_{i+1}) & j = i \\ \alpha_{i+1} & j = i+1 \\ 0 & \text{otherwise} \end{cases}, \tag{16}$$

the discrete form of the left-hand side of equation (4) is

$$\mathbf{A}\mathbf{D}_z + \mathbf{D}_r\mathbf{A}. \tag{17}$$

If a higher order-of-accuracy is desired, one can replace the second-order method here for one of higher order. Doing so simply changes the entries of $\mathbf{D}_r$ and the rest of this method is unchanged.

Unfortunately the form of equation (17) is not computationally useful since the coefficient matrix $\mathbf{A}$ is both left and right multiplied by other matrices. However, we can remedy this by stacking the matrix $\mathbf{A}$ into a vector and expressing the right multiplication as a left multiplication. Define the mapping vectors $\mathbf{I}$ and $\mathbf{J}$ and coefficient vector $\mathbf{u}$ (all with lengths NM) such that

$$(\mathbf{u})_i = a_{J_i}(r_{I_i}) = (\mathbf{A})_{I_i,J_i}. \tag{18}$$

Note that $J_i \in \{1, ..., M\}$ and $I_i \in \{1, ..., N\}$. Now, we can use the stacked z-derivative matrix

$$(\mathbf{D}_z^*)_{i,j} = \begin{cases} (\mathbf{D}_z)_{J_j,J_i} & I_i = I_j \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

To see this matrix gives the desired result (left multiplication instead of right), note that

$$(\mathbf{D}_z^*\mathbf{u})_i = \sum_{j=1}^{NM} (\mathbf{D}_z^*)_{i,j} (\mathbf{u})_j = \sum_{\substack{j \\ I_i = I_j}} (\mathbf{D}_z^*)_{i,j}\, a_{J_j}(r_{I_j}) = \sum_{\substack{j \\ I_i = I_j}} (\mathbf{D}_z)_{J_j,J_i}\, a_{J_j}(r_{I_i}). \tag{20}$$

Now, for a given index i, there are precisely M indices j such that $I_i = I_j$ (i.e., $\mathbf{I}$ contains each value in $\{1, ..., N\}$ exactly M times). These indices must be distinct, and furthermore, the corresponding J_j cover $\{1, ..., M\}$. Thus, the above sum can be replaced by

$$(\mathbf{D}_z^*\mathbf{u})_i = \sum_{j=1}^{M} (\mathbf{D}_z)_{j,J_i}\, a_j(r_{I_i}) = (\mathbf{A}\mathbf{D}_z)_{I_i,J_i}. \tag{21}$$

Thus, we have turned the right multiplication by a derivative matrix into a left multiplication.

We can similarly form the stacked r-derivative matrix

$$(\mathbf{D}_r^*)_{i,j} = \begin{cases} (\mathbf{D}_r)_{I_i,I_j} & J_i = J_j \\ 0 & \text{otherwise} \end{cases}. \tag{22}$$

By arguments similar to those above, one can show that

$$(\mathbf{D}_r^* \mathbf{u})_i = (\mathbf{D}_r \mathbf{A})_{I_i,J_i}. \tag{23}$$

Therefore, equation (4) can also be written

$$\mathbf{D}_z^* \mathbf{u} + \mathbf{D}_r^* \mathbf{u} = \underbrace{(\mathbf{D}_z^* + \mathbf{D}_r^*)}_{\mathbf{K}} \mathbf{u}. \tag{24}$$

The boundary conditions in r are encoded in the r-derivative matrix $\mathbf{D}_r$. The boundary condition $c_g \to 0$ as $z^* \to \infty$ is naturally satisfied by the Laguerre function expansion, but the boundary condition at $z^* = 0$ is not. For this, we use boundary bordering. Noting that $L_n(0) = 1$, we have

$$c_g(r_k, z^* = 0) = \sum_{n=0}^{M-1} a_n(r_k) = f(r_k) \tag{25}$$

for each k. Thus, we redefine $\mathbf{K}$ and define the boundary data vector $\mathbf{c}$ as

$$(\mathbf{K})_{i,j} = \begin{cases} 1 & J_i = M, I_i = I_j \\ 0 & J_i = M, I_i \neq I_j \\ (\mathbf{D}_z^* + \mathbf{D}_r^*)_{i,j} & \text{otherwise} \end{cases}, \qquad (\mathbf{c})_i = \begin{cases} f(r_{I_i}) & J_i = M \\ 0 & \text{otherwise} \end{cases}, \tag{26}$$

giving the linear system

$$\mathbf{K}\mathbf{u} = \mathbf{c}. \tag{27}$$

This enforces that the quantity in equation (24) vanishes except for a set of N equations that are replaced with equation (25) for each k (the choice of $J_i = M$ is heuristic; formally, we can replace any set of N equations). While $\mathbf{K}$ has $(NM)^2$ components, it is strongly banded with proper choice of the mapping vectors $\mathbf{I}$ and $\mathbf{J}$, so equation (27) can be quickly solved by a banded-system solver. While this must be done every time the boundary data $f(r)$ changes, $\mathbf{K}$ is constant and can be precomputed (and factorized using banded LU factorization).

With equation (2), the evaporative flux J is given by

$$J = -\left.\frac{\partial c_g}{\partial z^*}\right|_{z^*=0} \approx -\sum_{n=0}^{M-1} a_n'(r). \tag{28}$$

In discrete matrix form, the coefficients a_n' are (from equation (11))

$$\mathbf{A}' = \mathbf{A}\left(-\frac{1}{2}\mathbf{I} - \mathbf{D}\right). \tag{29}$$

One then computes the sum $-\sum_j (\mathbf{A}')_{i,j}$ to obtain the evaporative flux $J(r_i)$ at each node position r_i.