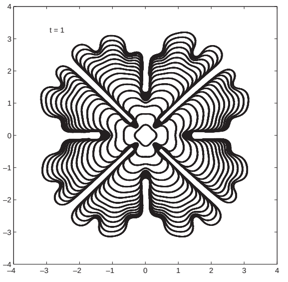

Chapter 9

The IIM for Parabolic Interface Problems

The IIM for parabolic interface problems with applications has been developed in [6, 160, 162, 172, 173, 175, 177, 181]. In this chapter, we explain the method for one-dimensional elliptic interface problems with fixed and moving interfaces, the alternative directional implicit (ADI) method for heat equations with a fixed interface, and the IIM for diffusion and advection equations with a fixed interface. The IIM for Stokes and Navier–Stokes equations with interfaces is explained in the next chapter.

9.1 The IIM for one-dimensional heat equations with fixed interfaces

Consider the model problem,

$$
\begin{gathered}
u_t(x,t) = (\beta(x,t)\,u_x)_x - \sigma(x,t)\,u(x,t) - f(x,t) + v(t)\delta(x-\alpha), \\
0 \le x \le 1, \qquad 0 < \alpha < 1, \qquad t \ge 0,
\end{gathered} \tag{9.1}
$$

with specified boundary and initial conditions. We assume that $\beta(x,t)$, $\sigma(x,t)$, and $f(x,t)$ are bounded but may have a finite discontinuity at the interface α. From the equation we can conclude that

$$
[\beta u_x] = v(t). \tag{9.2}
$$

We also specify

$$
[u] = w(t). \tag{9.3}
$$

With the IIM, the standard Crank–Nicolson scheme, which is unconditionally stable, is used at regular grid points. The finite difference scheme from time level t^n to t^{n+1} has the following generic form:

$$
\begin{aligned}
\frac{U_i^{n+1} - U_i^n}{\Delta t} = \frac{1}{2}\ \Big\{ & \gamma_{i,1}^n U_{i-1}^n + \gamma_{i,2}^n U_i^n + \gamma_{i,3}^n U_{i+1}^n - \sigma_i^n\, U_i^n - f_i^n + C_i^n \\
& + \gamma_{i,1}^{n+1} U_{i-1}^{n+1} + \gamma_{i,2}^{n+1} U_i^{n+1} + \gamma_{i,3}^{n+1} U_{i+1}^{n+1} - \sigma_i^{n+1}\, U_i^{n+1} - f_i^{n+1} + C_i^{n+1} \Big\},
\end{aligned}
$$

where Δt is the time step and the ratio $\Delta t/h$ is a constant, $\sigma_i^n = \sigma(x_i, t^n)$, and so on. At regular grid points for which $\alpha \notin (x_{i-1}, x_{i+1})$, we have the standard finite difference coefficients,

$$\begin{aligned} \gamma_{i,1}^n &= \frac{\beta_{i-\frac{1}{2}}^n}{h^2}, \qquad \gamma_{i,2}^n = -\frac{\left(\beta_{i-\frac{1}{2}}^n + \beta_{i+\frac{1}{2}}^n\right)}{h^2}, \\ \gamma_{i,3}^n &= \frac{\beta_{i+\frac{1}{2}}^n}{h^2}, \qquad\qquad C_i^n = 0, \end{aligned} \tag{9.4}$$

where $\beta_{i-1/2}^n = \beta(x_{i-1/2}, t^n)$, and so on. Since the interface is fixed, the derivation for the finite difference scheme is just slightly different from that in Chapter 2. So we omit the details and give the results directly.

Suppose $x_j \le \alpha < x_{j+1}$; then x_j and x_{j+1} are two irregular grid points. In this case the coefficients $\gamma_{j,1}^n$, $\gamma_{j,2}^n$, and $\gamma_{j,3}^n$ satisfy the following system of equations:

$$\begin{aligned} &\gamma_{j,1}^n + \gamma_{j,2}^n + \left(1 + \frac{(x_{j+1}-\alpha)^2}{2\,(\beta^+)^n}\,[\sigma]^n\right)\gamma_{j,3}^n = 0, \\ &(x_{j-1}-\alpha)\,\gamma_{j,1}^n + (x_j-\alpha)\,\gamma_{j,2}^n + \left\{ \frac{(\beta^-)^n}{(\beta^+)^n}\,\left(x_{j+1}-\alpha\right) \right. \\ &\qquad \left. + \left(\frac{(\beta_x^-)^n}{(\beta^+)^n} - \frac{(\beta^-)^n(\beta_x^+)^n}{\{(\beta^+)^n\}^2}\right)\frac{(x_{j+1}-\alpha)^2}{2}\right\}\gamma_{j,3}^n = (\beta_x^-)^n, \\ &\frac{(x_{j-1}-\alpha)^2}{2}\gamma_{j,1}^n + \frac{(x_j-\alpha)^2}{2}\gamma_{j,2}^n + \frac{(x_{j+1}-\alpha)^2\,(\beta^-)^n}{2\,(\beta^+)^n}\,\gamma_{j,3}^n = (\beta^-)^n, \end{aligned}$$

where

$$[\sigma]^n = \left[\sigma(\alpha, x^n)\right], \qquad (\beta^-)^n = \beta(\alpha^-, t^n), \qquad (\beta^+)^n = \beta(\alpha^+, t^n),$$

and so on; see (2.14) for a comparison. The correction term C_j^n is

$$\begin{aligned} C_j^n &= \gamma_{j,3}^n \left\{ w^n + (x_{j+1}-\alpha)\,\frac{v^n}{(\beta^+)^n} \right\} \\ &\quad - \gamma_{j,3}^n\,\frac{(x_{j+1}-\alpha)^2}{2}\left\{ \frac{(\beta_x^+)^n\,v^n}{\{(\beta^+)^n\}^2} - \frac{\left(\sigma^+\right)^n w^n + \dfrac{dw}{dt}\left(t^n\right) + [f]^n}{(\beta^+)^n} \right\}. \end{aligned}$$

Notice that now there is an extra term $\frac{dw(t^n)}{dt}$ compared with the correction term (2.15) in the general one-dimensional elliptic problem due to the $[u_t]$ term. Similarly, at the grid point

x_{j+1}, the coefficients $\gamma_{j+1,1}^n$, $\gamma_{j+1,2}^n$, and $\gamma_{j+1,3}^n$ satisfy the following system of equations:

$$\left(1-\frac{(x_j-\alpha)^2}{2\,(\beta^-)^n}\,[\sigma]^n\right)\gamma_{j+1,1}^n+\gamma_{j+1,2}^n+\gamma_{j+1,3}^n=0,$$

$$\left\{\frac{(\beta^+)^n}{(\beta^-)^n}\,(x_j-\alpha)+\left(\frac{(\beta_x^+)^n}{(\beta^-)^n}-\frac{(\beta_x^-)^n\,(\beta^+)^n}{\{(\beta^-)^n\}^2}\right)\frac{(x_j-\alpha)^2}{2}\right\}\gamma_{j+1,1}^n$$

$$+\,(x_{j+1}-\alpha)\,\gamma_{j+1,2}^n+(x_{j+2}-\alpha)\,\gamma_{j+1,3}^n=\left(\beta_x^+\right)^n,$$

$$\frac{(x_j-\alpha)^2}{2}\frac{(\beta^+)^n}{(\beta^-)^n}\,\gamma_{j+1,1}^n+\frac{(x_{j+1}-\alpha)^2}{2}\,\gamma_{j+1,2}^n+\frac{(x_{j+2}-\alpha)^2}{2}\,\gamma_{j+1,3}^n=(\beta^+)^n;$$

see (2.16) for a comparison. The correction term now is

$$C_{j+1}^n=\gamma_{j+1,1}^n\left\{-w^n-(x_j-\alpha)\frac{v^n}{(\beta^-)^n}\right\}$$

$$-\,\gamma_{j+1,1}^n\frac{(x_j-\alpha)^2}{2}\left\{\frac{(\beta_x^-)^n\,v^n}{\{-(\beta^-)^n\}^2}+\frac{(\sigma^-)^n\,w^n+dw/dt\left(t^n\right)+[f]^n}{(\beta^-)^n}\right\}.$$

Note that the formulas are exactly the same for the time level $n+1$. The method is unconditionally stable if $\sigma\geq 0$ regardless of the jumps, provided that $\beta(x,t)$ has the same sign across the interface.

9.2 The IIM for one-dimensional moving interface problems

In this section, we discuss the IIM for the one-dimensional moving interface problem,

$$\begin{aligned} &u_t+\lambda u u_x=(\beta u_x)_x-f(x,t), \qquad x\in(0,\,\alpha)\cup(\alpha,\,1),\\ &\frac{d\alpha}{dt}=g\left(t,\alpha;\;u^-,u^+,u_x^-,u_x^+\right), \qquad t>0, \end{aligned} \tag{9.5}$$

with an initial condition and a prescribed boundary condition at $x=0$ and $x=1$, where $\beta(x,t)>0$ and g are given functions. As before, the source term $f(x,t)$ may be discontinuous or have a delta function singularity at the interface $\alpha(t)$. It is reasonable to assume that the solution is piecewise smooth and discontinuities can occur only at the interface $\alpha(t)$.

The interface $\alpha(t)$ divides the solution domain into two parts: $0\leq x<\alpha(t)$ and $\alpha(t)<x\leq 1$. The solution in each domain $[0,\alpha(t))$ and $(\alpha(t),1]$ is assumed to be smooth, but coupled with the solution on the other side by interface conditions (or internal boundary conditions) that usually take one of the following forms.

Case 1: The solution on the interface is given. One example is the classical Stefan model for one-dimensional solidification problems. The temperature at the melting/freezing interface is given by the melting temperature, that is, $u(\alpha,t)=u_0$ is known.

Various approaches have been used to solve the Stefan problem and other *linear* free or moving interface problems numerically; see, for example, [9, 38, 59, 78, 83, 84, 114, 175, 204, 226, 271]; also see [42, 236] which use the level set method. Compared with Case 2 discussed below, the Stefan problem is easier to solve because the value of the solution on the interface is known. However, a few numerical methods are second-order accurate in the maximum norm for both the solution and the interface. Several methods involve some transformations for either the differential equations or the coordinate system, which complicates the problem in some way. The IIM proposed in [162] is simple, stable, and is second-order accurate for both the solution u and the interface $\alpha(t)$ simultaneously for more general equations.

Case 2: The jump conditions of the form

$$[u] = w(t), \qquad [\beta u_x] = v(t) \tag{9.6}$$

are given. This is a one-dimensional model for the immersed boundary method formulation with a more general equation for the motion. The problem can be written as a single equation without using the jump conditions:

$$u_t + \lambda u u_x = (\beta u_x)_x - v(t)\,\delta(x-\alpha(t)) - \hat{C}(t)\,\delta'(x-\alpha(t)) - f(x,t)$$

for some function $\hat{C}(t)$.

Beyer and LeVeque [23] studied various one-dimensional moving interface problems for the heat equation assuming a priori knowledge of the interface. In their approach a discrete delta function is carefully selected and some correction terms are added if necessary to get second-order accuracy. Wiegmann and Bube [269] applied the IIM for certain one-dimensional nonlinear problem with a *fixed* interface. However, for the moving interface problem (9.5), the interface is *unknown* and *moving*, and the discrete difference scheme is a nonlinear system of equations involving the solution and the interface.

Case 2 (with $\lambda = 0$) is also a model of the heat conduction with an interface between two different materials. In this case u is the temperature, and hence is continuous, i.e., $w(t) \equiv 0$ in (9.6). The net heat flux across the interface is $v(t)$ in the second jump condition in (9.6). Again, in this case we do not know the value of the solution on the interface but only the jump conditions.

For many classical Stefan problems, the motion of the interface is proportional to the flux across the interface,

$$\frac{d\alpha}{dt} = \sigma(t)\,[\beta u_x], \qquad u(\alpha, t) = u_0, \tag{9.7}$$

where u_0 is the known temperature at the interface. This type of problems fits both Case 1 and Case 2.

9.2.1 The modified Crank–Nicholson scheme

Given a uniform grid, set

$$x_i = ih, \qquad i = 0, 1, \dots, M, \quad x_0 = 0, \quad x_M = 1.$$

Let Δt be the time step size and let the ratio $\Delta t/h$ be a constant so that we can write $O(\Delta t)$ as $O(h)$ or vice versa. Using the Crank–Nicolson scheme, the semidiscrete difference scheme for (9.5) can be written in the following general form:

$$\begin{aligned} &\frac{U_i^{n+1}-U_i^n}{\Delta t} - Q_i^{n+\frac{1}{2}} + \frac{\lambda}{2}\left(U_i^n U_{x,i}^n + U_i^{n+1}U_{x,i}^{n+1}\right) \\ &= \frac{1}{2}\left((\beta U_x)_{x,i}^n + (\beta U_x)_{x,i}^{n+1}\right) - \frac{1}{2}\left(f_i^n + f_i^{n+1}\right), \end{aligned} \tag{9.8}$$

where $U_{x,i}^n$ and $(\beta U_x)_{x,i}^n$ are discrete analogues of u_x and $(\beta u_x)_x$ at (x_i, t^n), and $Q_i^{n+\frac{1}{2}}$ is a correction term needed when α crosses the grid line $x = x_i$ at some time between t^n and t^{n+1}. We will discuss how to determine $Q_i^{n+\frac{1}{2}}$ in the next subsection. For simplicity, we will drop the superscript n in the discussion of the spatial discretization if there is no confusion. At a grid point x_i, which is away from the interface (i.e., $\alpha \notin [x_{i-1}, x_{i+1}]$), the classic 3-point central finite difference discretizations are

$$U_{x,i} = \frac{U_{i+1} - U_{i-1}}{2h}, \tag{9.9}$$

$$(\beta U_x)_{x,i} = \frac{\beta_{i-\frac{1}{2}}U_{i-1} - \left(\beta_{i-\frac{1}{2}} + \beta_{i+\frac{1}{2}}\right)U_i + \beta_{i+\frac{1}{2}}U_{i+1}}{h^2}, \tag{9.10}$$

where $\beta_{i+\frac{1}{2}} = \beta(x_i + \frac{h}{2}, :)$. In §9.2.3 we will discuss how to discretize u_x and $(\beta u_x)_x$ when $\alpha \in [x_{j-1}, x_{j+1})$ for Cases 1 and 2.

The interface location is determined by the trapezoidal method applied to the second equation in (9.5),

$$\frac{\alpha^{n+1} - \alpha^n}{\Delta t} = \frac{1}{2}\left(g^n + g^{n+1}\right), \tag{9.11}$$

where $g^n = g(t^n, \alpha^n;\ u^{-,n},\ u^{+,n},\ u_x^{-,n},\ u_x^{+,n})$ and α^n, $u^{\pm,n}$, $u_x^{\pm,n}$ are the approximations to $\alpha(t^n)$, $u(\alpha^\pm, t^n)$, and $u_x(\alpha^\pm, t^n)$, respectively. The same is true for the time level $n+1$. The discretizations (9.8) and (9.11) are second-order accurate and fully implicit. The core of the algorithm at a time level t^n consists of the following.

- Determine $Q_j^{n+\frac{1}{2}}$ if the interface crosses the grid line $x = x_j$ from time t^n to time t^{n+1}.
- Derive the finite difference approximations for u_x and $(\beta u_x)_x$ at the two grid points closest to the interface.
- Compute the interface quantities $u^\pm$, $u_x^\pm$, $[u_t]$, etc.
- Solve the nonlinear system of equations for the approximate solution $\{U_i^{n+1}\}$ and the approximate interface location α^{n+1}.

Away from the interface, the local truncation errors for the finite difference scheme are $O(h^2)$. But at a few grid points near the interface, we allow the local truncation errors to be $O(h)$ based on the fact that the local truncation error of a finite difference scheme on a boundary can be one order lower than those of interior points without affecting global second-order accuracy.

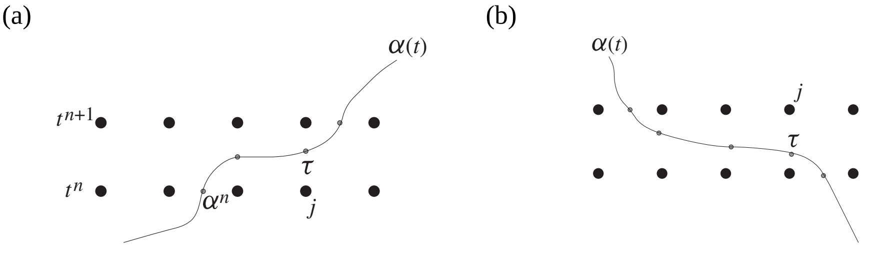

Figure 9.1. *A diagram of an interface crossing a grid line.* (a) $\alpha(t)$ *increases with time.* (b) $\alpha(t)$ *decreases with time.*

9.2.2 Dealing with grid crossing

If there is no grid crossing at a grid point x_i from time t^n to time t^{n+1}, that is, (x_i, t^n) and (x_i, t^{n+1}) are on the same side of the interface $\alpha(t)$, then $Q_i^{n+\frac{1}{2}} = 0$ and

$$\frac{u(x_i, t^{n+1}) - u(x_i, t^n)}{\Delta t} = \frac{1}{2}\Big(u_t(x_i, t^{n+1}) + u_t(x_i, t^n)\Big) + O((\Delta t)^2). \tag{9.12}$$

However, if the interface crosses the grid line $x = x_j$ at some time τ, $t^n < \tau < t^{n+1}$, such that $x_j = \alpha(\tau)$[13] (see Figure 9.1), then the time derivative of u may have a finite jump at $t = \tau$. In this case, even though we can approximate the x-derivatives of u well at each time level (see §9.2.3), the standard Crank–Nicolson scheme needs to be corrected to guarantee second-order accuracy. This is done by choosing a correction term $Q_j^{n+\frac{1}{2}}$ based on the following theorem.

Theorem 9.1. *Let* $u(x,t)$ *be the solution of* (9.5)*. Suppose that the equation* $\alpha(t) = x_j$ *has a unique solution* τ *in the interval* $t^n < t < t^{n+1}$*. If we choose*

$$Q_j^{n+\frac{1}{2}} = \frac{[u]_{;\tau}}{\Delta t} + \frac{1}{\Delta t}\left(t^{n+\frac{1}{2}} - \tau\right)[u_t]_{;\tau}, \tag{9.13}$$

then

$$\begin{aligned}\frac{u(x_j, t^{n+1}) - u(x_j, t^n)}{\Delta t} &- Q_j^{n+\frac{1}{2}} \\ &= \frac{1}{2}\left(u_t(x_j, t^{n+1}) + u_t(x_j, t^n)\right) + O(\Delta t).\end{aligned} \tag{9.14}$$

Proof: We expand $u(x_j, t^n)$ and $u(x_j, t^{n+1})$ in Taylor series at approximately time τ from each side of the interface to get

$$\begin{aligned} u(x_j, t^n) &= u(x_j, \tau^-) + (t^n - \tau)\, u_t(x_j, \tau^-) + O((\Delta t)^2), \\ u(x_j, t^{n+1}) &= u(x_j, \tau^+) + (t^{n+1} - \tau)\, u_t(x_j, \tau^+) + O(\Delta t^2) \\ &= u(x_j, \tau^-) + [u]_{;\tau} + (t^{n+1} - \tau)\, u_t(x_j, \tau^-) + (t^{n+1} - \tau)\,[u_t]_{;\tau} + O((\Delta t)^2). \end{aligned}$$

[13]The crossing time τ really depends on the grid index j as well as the time index n; see Figure 9.1. To simplify the notation, τ will be used to indicate the crossing time, without explicitly showing its dependence on j and n.

Combining the two expressions above gives

$$u(x_j, t^{n+1}) - u(x_j, t^n) = \Delta t\, u_t(x_j, \tau^-) + [u_t]_{;\tau} \\ + (t^{n+1} - \tau)\,[u_t]_{;\tau} + O\big((\Delta t)^2\big). \tag{9.15}$$

On the other hand, we also have

$$u_t(x_j, t^n) = u_t(x_j, \tau^-) + O(\Delta t),$$
$$u_t(x_j, t^{n+1}) = u_t(x_j, \tau^-) + [u_t]_{;\tau} + O(\Delta t).$$

Thus it follows that

$$\Delta t\, u_t(x_j, \tau^-) = \frac{\Delta t}{2}\big(u_t(x_j, t^n) + u_t(x_j, t^{n+1})\big) - \frac{\Delta t}{2}[u_t]_{;\tau} + O\big((\Delta t)^2\big). \tag{9.16}$$

Substituting (9.16) into (9.15) gives

$$\frac{u(x_j, t^{n+1}) - u(x_j, t^n)}{\Delta t} = \frac{1}{2}\big(u_t(x_j, t^n) + u_t(x_j, t^{n+1})\big) + \frac{[u]_{;\tau}}{\Delta t} \\ + \frac{t^{n+\frac{1}{2}} - \tau}{\Delta t}\,[u_t]_{;\tau} + O(\Delta t). \tag{9.17}$$

This is equivalent to (9.14). □

We know $[u]_{;\tau}$ from the jump conditions. However, to compute $Q_j^{n+\frac{1}{2}}$, we also need to find the crossing time τ and the jump $[u_t]_{;\tau}$. We first discuss how to find τ if it exists. We will discuss how to approximate $[u_t]_{;\tau}$ in §9.2.4. Using the Crank–Nicolson formula twice, we get

$$\frac{\alpha^\tau - \alpha^n}{\tau - t^n} = \frac{1}{2}\left(g^n + g^\tau\right),$$
$$\frac{\alpha^{n+1} - \alpha^\tau}{t^{n+1} - \tau} = \frac{1}{2}\left(g^\tau + g^{n+1}\right).$$

Eliminating g^τ from the two expressions above and using $\alpha^\tau = x_j$, we get

$$\frac{x_j - \alpha^n}{\tau - t^n} - \frac{\alpha^{n+1} - x_j}{t^{n+1} - \tau} = \frac{1}{2}\left(g^n - g^{n+1}\right). \tag{9.18}$$

This is the equation for the crossing time τ and it is coupled with (9.8), (9.11), and (9.13).

Note that the discussion above is still valid even if the interface crosses several grid points during one time step. However, it would be better to control the time step so that the interface crosses only one grid point during one time step. This will give a smaller error constant.

9.2.3 The discretizations of u_x and $(\beta u_x)_x$ near the interface

As before, only the finite difference equations at the closest grid points from the left and the right of the interface need to be modified at each time level t^n or t^{n+1}. The discretization apparently depends on interface conditions and will be discussed separately in this section.

Case 1: The solution on the interface is known. Let the solution on the interface be

$$u\,(\alpha(t), t) = r(t), \tag{9.19}$$

where $r(t)$ is a given function. Since we know the value of the solution on the interface, we could discretize u_x and u_{xx} using a one-sided interpolation. For example, if $x_j \le \alpha^n < x_{j+1}$, then

$$u^n_{x,j} \approx U^n_x(j-1,\ j,\ \alpha^n, x_j),$$

where

$$\begin{aligned} U_x(j-1,\ j,\ \alpha, x) = {} & \frac{x_j + \alpha - 2x}{(x_{j-1} - \alpha)\,h}\,U_{j-1} + \frac{x_{j-1} + \alpha - 2x}{(\alpha - x_j)\,h}\,U_j \\ & + \frac{x_{j-1} + x_j - 2x}{(x_j - \alpha)(\alpha - x_{j-1})}\,r(t). \end{aligned} \tag{9.20}$$

This is a second-order approximation to $u_x(x_j, t)$. However, notice that $\alpha(t)$ changes with time, as does $x_j - \alpha$. If $|x_j - \alpha|$ becomes too small, the magnitudes of the coefficients in the interpolation (9.20) become very large and sometimes even blow up. An intuitive fix would be

$$u^n_{x,j} \approx U^n_x(j-2,\ j-1,\ \alpha^n, x_j),$$

when $|x_j - \alpha|$ is small. A more robust fix is by the linear combination of those two above,

$$\begin{aligned} u^n_{x,j} \approx {} & \frac{\alpha^n - x_j}{h}\ U^n_x(j-1,\ j,\ \alpha^n, x_j) \\ & + \frac{x_{j+1} - \alpha^n}{h}\ U^n_x(j-2,\ j-1,\ \alpha^n, x_j). \end{aligned} \tag{9.21}$$

This approach is the simplified version of the least squares interpolation in one dimension. There are several advantages of this robust approach. First of all, the interpolation is still second-order accurate. Second, if we rewrite (9.21) as

$$u^n_{x,j} \approx \sum_{j=j-2}^{j} \gamma^n_j\, U^n_j + \gamma^n_\alpha\, r(t^n),$$

then the magnitudes of the coefficients γ^n_j and γ^n_α will be always of order $O(1/h)$. Furthermore, the truncation error in such an interpolation varies smoothly, which gives better stability.

In the same manner, we use the following interpolation to discretize $u^n_{x,j+1}$:

$$\begin{aligned} u^n_{x,j+1} \approx {} & \frac{\alpha^n - x_j}{h}\ U^n_x(j+2,\ j+3,\ \alpha^n, x_{j+1}) \\ & + \frac{x_{j+1} - \alpha^n}{h}\ U^n_x(j+1,\ j+2,\ \alpha^n, x_{j+1}). \end{aligned} \tag{9.22}$$

For the second-order derivative u_{xx}, the corresponding one-sided finite difference approximation for u_{xx} is

$$\begin{aligned} U_{xx}(j-1,\ j,\ \alpha, x) = {} & \frac{2}{(\alpha - x_{j-1})\,h}\, U_{j-1} + \frac{2}{(\alpha - x_j)\,h}\, U_j \\ & + \frac{2}{(x_{j-1}-\alpha)(x_j-\alpha)}\, r(t). \end{aligned} \tag{9.23}$$

Using the formula above, we get the following robust finite difference approximations for $u^n_{xx,j}$:

$$\begin{aligned} u^n_{xx,j} \approx {} & \frac{\alpha^n - x_j}{h}\, U^n_{xx}(j-1,\ j,\ \alpha^n, x_j) \\ & + \frac{x_{j+1}-\alpha^n}{h}\, U^n_{xx}(j-2,\ j-1,\ \alpha^n, x_j), \end{aligned} \tag{9.24}$$

where U^n_{j-2}, U^n_{j-1}, and U^n_j are used. Similarly, we have

$$\begin{aligned} u^n_{xx,j+1} \approx {} & \frac{\alpha^n - x_j}{h}\, U^n_{xx}(j+2,\ j+3,\ \alpha^n, x_{j+1}) \\ & + \frac{x_{j+1}-\alpha^n}{h}\, U^n_{xx}(j+1,\ j+2,\ \alpha^n, x_{j+1}), \end{aligned} \tag{9.25}$$

where U^n_j, U^n_{j+1}, and U^n_{j+2} are used.

With (9.21)–(9.22) and (9.24)–(9.25), we can determine the explicit terms in (9.8) that involve the first- and second-order derivatives.

At the time level $n+1$, we do not wish to increase the finite difference stencil so that we can solve the nonlinear system of equations more efficiently. Note that if x_j is close to α^{n+1}, we simply set

$$u^{n+1}_j = r(t^{n+1}) \qquad \text{if} \qquad |x_j - \alpha^{n+1}| \le h^2, \tag{9.26}$$

without affecting second-order accuracy. If $|x_j - \alpha^{n+1}| > h^2$, we use (9.23) to approximate u_{xx} and use (9.20) to approximate u_x. By doing so we will have a tridiagonal system for the linearized equations.

Case 2: The jump conditions are known. Suppose we know the jump conditions (9.6) across the interface, i.e., $[u] = w(t)$ and $[\beta u_x] = v(t)$ are given. The finite difference equations at the two irregular grid points x_j and x_{j+1} are based on the following theorems, Theorems 9.2 and 9.3, as follows.

Theorem 9.2. *Let* $u(x,t)$ *be the solution of* (9.5) *with jump conditions* (9.6). *If* $x_j \le \alpha(t) < x_{j+1}$, *then*

$$\begin{bmatrix} u(\alpha^-, t) \\ u_x(\alpha^-, t) \\ u_{xx}(\alpha^-, t) \end{bmatrix} = S \begin{bmatrix} u(x_{j-1}, t) \\ u(x_j, t) \\ u(x_{j+1}, t) \end{bmatrix} - \begin{bmatrix} C_{j,1} \\ C_{j,2} \\ C_{j,3} \end{bmatrix} + \begin{bmatrix} O(h^3) \\ O(h^2) \\ O(h) \end{bmatrix}, \tag{9.27}$$

where

$$C_{j,k} = s_{k3}\left\{w + \frac{v}{\beta^+}(x_{j+1}-\alpha) + \frac{(x_{j+1}-\alpha)^2}{2\beta^+}\left(\frac{dw}{dt} + [f] - \frac{v}{\beta^+}\left(g - \lambda u^+ + \beta_x^+\right)\right)\right\}, \tag{9.28}$$

$k = 1, 2, 3$, $S = \{s_{kj}\} = A^{-1}$, *and*

$$A^T = \{a_{kj}\}^T = \begin{bmatrix} 1 & 1 & 1 \\ x_{j-1}-\alpha & x_j-\alpha & a_{32} \\ \dfrac{(x_{j-1}-\alpha)^2}{2} & \dfrac{(x_j-\alpha)^2}{2} & \dfrac{\rho\,(x_{j+}-\alpha)^2}{2} \end{bmatrix}.$$

The element a_{32} *is given by*

$$a_{32} = \rho(x_{j+1}-\alpha) + \frac{(x_{j+1}-\alpha)^2}{2\beta^+}\left(g\,(1-\rho) + \lambda(u^+\rho - u^-) + \beta_x^- - \rho\beta_x^+\right),$$

where $\rho = \beta^-/\beta^+$.

The proof of this theorem can be found in [162]. Using this theorem we get a discretized form of $(\beta u_x)_x - \lambda u u_x$ at the grid point x_j, $x_j \le \alpha < x_{j+1}$, using the grid points from both sides of the interface,

$$\begin{aligned}(\beta u_x)_{x,j} - \lambda u_j u_{x,j} &= \beta^- u_{xx}^- + \beta_x^- u_x^- - \lambda u^- u_x^- + O(h) \\ &\approx \left(\beta^- s_{31} + s_{21}(\beta_x^- - \lambda U^-)\right) U_{j-1} + \left(\beta^- s_{32} + s_{22}(\beta_x^- - \lambda u^-)\right) U_j \\ &\quad + \left(\beta^- s_{33} + s_{23}(\beta_x^- - \lambda U^-)\right) U_{j+1} - \left(\beta^- C_{j,3} + C_{j,2}(\beta_x^- - \lambda U^-)\right).\end{aligned} \tag{9.29}$$

An important feature of the discretization above is that we can still use a 3-point stencil, and thus the discretization is valid for any location α. The theorem above also gives an interpolation formula which can be used to compute the values of u^- and u_x^- that are needed to approximate g and the frozen term $\lambda u u_x$.

For the grid point x_{j+1}, $x_j \le \alpha < x_{j+1}$, there exists a similar formula that we state as follows.

Theorem 9.3. *Let* $u(x,t)$ *be the solution of* (9.5) *with jump conditions* (9.6). *If* $x_j \le \alpha(t) < x_{j+1}$, *then*

$$\begin{bmatrix} u(\alpha^+,t) \\ u_x(\alpha^+,t) \\ u_{xx}(\alpha^+,t) \end{bmatrix} = \tilde{S}\begin{bmatrix} u(x_j,t) \\ u(x_{j+1},t) \\ u(x_{j+2},t) \end{bmatrix} - \begin{bmatrix} C_{j+1,1} \\ C_{j+1,2} \\ C_{j+1,3} \end{bmatrix} + \begin{bmatrix} O(h^3) \\ O(h^2) \\ O(h) \end{bmatrix}, \tag{9.30}$$

where

$$C_{j+1,k} = \tilde{s}_{k1}\left\{-[u] - \frac{[\beta u_x]}{\beta^-}(x_j-\alpha) - \frac{(x_j-\alpha)^2}{2\beta^-}\left(\frac{dw}{dt} + [f] - \frac{v}{\beta^-}\left(g - \lambda u^- + \beta_x^-\right)\right)\right\}, \tag{9.31}$$

$k = 1, 2, 3$, $\tilde{S} = \{\tilde{s}_{kj}\} = \tilde{A}^{-1}$, and

$$\tilde{A}^T = \{\tilde{a}_{kj}\}^T = \begin{bmatrix} 1 & 1 & 1 \\ \tilde{a}_{12} & x_{j+1} - \alpha & x_{j+2} - \alpha \\ \dfrac{(x_j - \alpha)^2}{2\,\rho} & \dfrac{(x_{j+1} - \alpha)^2}{2} & \dfrac{(x_{j+2} - \alpha)^2}{2} \end{bmatrix}.$$

The element $\tilde{a}_{12}$ is given by

$$\tilde{a}_{12} = \frac{(x_j - \alpha)}{\rho} - \frac{(x_j - \alpha)^2}{2\beta^-} \left(g\left(1 - \frac{1}{\rho}\right) + \lambda\left(\frac{u^-}{\rho} - u^+\right) - \frac{\beta_x^-}{\rho} + \beta_x^+ \right).$$

9.2.4 Computing interface quantities

As mentioned in §9.2.1, in order to compute (9.11), (9.13), (9.18), and (9.29), we need to compute the interface values such as $u^\pm$, $u_x^\pm$, $[u_t]_\tau$, etc. Again we distinguish two different cases.

Case 1: The solution on the interface $u(\alpha(t), t) = r(t)$ is known. In this case, the solution is continuous, that is, $u^- = u^+ = r(t)$. With the knowledge of the computed solution $\{U_i^n\}$, an estimate of $\{U_i^{n+1}\}$, and the solution on the interface $r(t^n)$, we use the one-sided difference (9.21) to approximate $u_x^{-,n}$ by switching the position between x_j and α^n, $x_j \le \alpha^n < x_{j+1}$. Similarly, we use (9.22) to approximate $u_x^{+,n}$ after switching the position of x_{j+1} and α^n in (9.22). The same approach is used for the next time level t^{n+1}.

If the interface crosses a grid line $x = x_i$ at some time τ, we need to compute $[u_t]_{;\tau}$ in order to get the correction term $Q_i^{n+\frac{1}{2}}$. In this case we simply use

$$[u_t]_{;\tau} \approx \frac{u_i^{n+1} - r(\tau)}{t^{n+1} - \tau} - \frac{r(\tau) - u_i^n}{\tau - t^n}. \tag{9.32}$$

Case 2: The jump conditions are known. In this case, $u^\pm$ and $u_x^\pm$ are computed using (9.27)–(9.28) and (9.30)–(9.31). In order to compute the jump $[u_t]_\tau$, we differentiate the first jump condition

$$u(\alpha^+, t) - u(\alpha^-, t) = w(t)$$

with respect to t to get

$$\big(u_x(\alpha^+, t) - u_x(\alpha^-, t)\big)\frac{d\alpha}{dt} + u_t(\alpha^+, t) - u_t(\alpha^-, t) = w'(t),$$

$$\text{i.e.,} \qquad [u_t] = w'(t) - [u_x]\, g. \tag{9.33}$$

We need to express (9.33) in terms of the quantities at either time level t^n or t^{n+1}. If $\alpha^n \ge \alpha^{n+1}$ (see Figure 9.1(b)), we have, at time $t = \tau$,

$$u^- = u(x_j, t^n) + O(h) \approx U_j^n, \qquad u^+ = u(x_j, t^{n+1}) + O(h) \approx U_j^{n+1},$$
$$u_x^- = u_x(x_j, t^n) + O(h) \approx U_{x,j}^n, \qquad u_x^+ = u_x(x_j, t^{n+1}) + O(h) \approx U_{x,j}^{n+1}.$$

Otherwise, we have

$$u^- = u(x_j, t^{n+1}) + O(h) \approx U_j^{n+1}, \qquad u^+ = u(x_j, t^n) + O(h) \approx U_j^n,$$
$$u_x^- = u_x(x_j, t^{n+1}) + O(h) \approx U_{x,j}^{n+1}, \qquad u_x^+ = u_x(x_j, t^n) + O(h) \approx U_{x,j}^n$$

for $\alpha^n < \alpha^{n+1}$. Thus we use the following scheme to compute $[u_t]_\tau$:

$$[u_t]_{;\tau} \approx \begin{cases} w'(\tau) - g(\tau, U_j^n, U_j^{n+1}, U_{x,j}^n, U_{x,j}^{n+1})(U_{x,j}^{n+1} - U_{x,j}^n) & \text{if } \alpha^n \geq \alpha^{n+1}, \\ w'(\tau) - g(\tau, U_j^{n+1}, U_j^n, U_{x,j}^{n+1}, U_{x,j}^n)(U_{x,j}^{n+1} - U_{x,j}^n) & \text{if } \alpha^n < \alpha^{n+1}. \end{cases}$$

9.2.5 Solving the resulting nonlinear system of equations

From the discussions above, we know that in order to get an approximate solution for $u(x,t)$ at time t^{n+1} using the finite difference method described here, generally we need to solve the following nonlinear system of equations:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} - Q_i^{n+\frac{1}{2}} + \frac{\lambda}{2}\left(U_i^n U_{x,i}^n + U_i^{n+1} U_{x,i}^{n+1}\right)$$
$$= \frac{1}{2}\left((\beta U_x)_{x,i}^n + (\beta U_x)_{x,i}^{n+1}\right) - \frac{1}{2}\left(f_i^n + f_i^{n+1}\right),$$

$$\frac{\alpha^{n+1} - \alpha^n}{\Delta t} = \frac{1}{2}\left(g^n + g^{n+1}\right),$$

where the quantities $U_{x,i}^n$ and $(\beta U_x)_{x,i}^n$ can be expressed as some linear combinations of U_i^n.

Since a fully implicit discretization is used, the numerical scheme should be stable. The local truncation errors are $O(h^2)$ at most grid points, but they are $O(h)$ at the two grid points that are closest to the interface from the left and the right, and at those grid points where the interface crosses. The global error in the solution is second-order accurate at all grid points as explained in §2.4 for one-dimensional elliptic interface problems.

We need to solve a nonlinear system of equations for $(U_j^{n+1}, \alpha^{n+1})$ at each time step. The difficulty is that some quantities, such as $Q_j^{n+\frac{1}{2}}$ and $C_{j,k}^{n+1}$, are not known until we know the solutions $\{U_i^{n+1}\}$ and α^{n+1}. We use an implicit discretization for the diffusion term $(\beta u_x)_x$ and a *prediction-correction* approach for $\alpha(t)$. An adaptive time step is chosen for (9.5) based on the classic stability theory,

$$\Delta t \leq \min\left\{h, \left|\frac{1}{\partial g/\partial \alpha}\right|\right\}.$$

The constraint $\Delta t \leq h$ is imposed to maintain second-order accuracy in both space and time. From the second equation in (9.5), we can get

$$\frac{d\alpha}{dt} = \frac{\partial \alpha}{\partial g}\frac{\partial g}{\partial t} = g, \qquad \text{i.e.,} \qquad \frac{\partial g}{\partial \alpha} = \frac{\partial g}{\partial t}\Big/ g.$$

That implies

$$\Delta t \le \min\left\{h,\ \left|\frac{g}{\partial g/\partial t}\right|\right\},$$

or, in the discrete form,

$$\Delta t_{new} = \min\left\{h,\ \left|\frac{g^n\,\Delta t_{old}}{g^{n+1}-g^n}\right|\right\}. \tag{9.34}$$

Below we give an outline of the iterative process. Suppose we have obtained all quantities at the time level t^n, and the current time step size is Δt (i.e., $t^{n+1} = t^n + \Delta t$). To get all corresponding quantities at the time level t^{n+1}, we follow the procedure outlined below.

- Determine j_0 such that $x_{j_0} \le \alpha^n < x_{j_0+1}$. Approximate u_x^n and $(\beta u_x)_x^n$ at x_{j_0} according to the scheme discussed in §9.2.3.
- Set

$$\alpha_1^{n+1} = \alpha^n + \Delta t\, g(t^n, \alpha^n;\ U^{-,n},\ U^{+,n},\ U_x^{-,n},\ U_x^{+,n}).$$

- Set an initial guess of the solution $\{U_{i,1}^{n+1}\} = \{U_i^n\}$ at the time level t^{n+1}.

For $m = 1, 2, \dots$, do the following:

  - (**) Determine j_m such that $x_{j_m} \le \alpha_1^{n+1} < x_{j_m+1}$. Determine the coefficients and the correction terms for u_x^{n+1} and u_{xx}^{n+1} at x_{j_m}. Substitute $U_{i,m}^{n+1}$ for U_i^{n+1} in the nonlinear term $U_i^{n+1}U_{x,i}^{n+1}$.
  - If $j_0 \ne j_m$, then for $l = j_0+1, \dots, j_m$, when $j_0 < j_m$, or for $l = j_m$, $j_m+1, \dots, j_0$, when $j_0 > j_m$, first get τ_m using (9.18), then determine the correction $Q_{l,m}^{n+\frac{1}{2}}$ to $(U_{l,m}^{n+1} - U_l^n)/\Delta t$ using the approach described in §9.2.2.
  - Solve the tridiagonal system for $U_{i,m+1}^{n+1}$.
  - Interpolate $\{U_{i,m+1}^{n+1}\}$ to get $U_{m+1}^{\pm,n+1}$ and $U_{x,m+1}^{\pm,n+1}$ if necessary (depending on the interface condition).
  - Determine

$$\alpha_{m+1}^{n+1} = \alpha^n + \frac{\Delta t}{2}\left(g^n + g_{m+1}^{n+1}\right),$$

  where

$$g_{m+1}^{n+1} = g\left(t^{n+1}, \alpha_m^{n+1}; U_{m+1}^{-,n+1}, U_{m+1}^{+,n+1}, U_{x,m+1}^{-,n+1}, U_{x,m+1}^{+,n+1}\right).$$

  - If $|\alpha_m^{n+1} - \alpha_{m+1}^{n+1}| > \epsilon$, a given tolerance, then $m = m+1$. Go to (**).

- If $|\alpha_m^{n+1} - \alpha_{m+1}^{n+1}| < \epsilon$, then set all quantities $\{\ \}_m^{n+1}$ to $\{\ \}^{n+1}$; in other words, accept the values $U_{i,m+1}^{n+1}$ and α_{m+1}^{n+1} as approximations at time level t^{n+1}. Determine the next time step size Δt,

$$\Delta t = \min\left\{ h,\ \left| \frac{g^n\,\Delta t}{g^{n+1} - g^n} \right| \right\}. \tag{9.35}$$

Go to the next time step.

9.2.6 Validation of the algorithm for a one-dimensional moving interface problem

We validate the IIM for a one-dimensional moving interface problem through a classical Stefan problem of tracking a freezing front of ice in water. The description of the problem is excerpted from [84], where Furzeland used this example to compare different methods. The thermal properties are the heat conductivity k_i, the specific heat c_i, the density ρ (assumed to be the same in each phase) and the latent heat L. The subscript $i = 1$ denotes the phase 1 (ice) in $0 < x < \alpha(t)$ and $i = 2$ denotes the phase 2 (water) in $\alpha(t) < x < 1$. Define also two constants $C_i = c_i\,\rho$ and $\sigma = L\rho$. The governing equations are

$$C_i \frac{\partial u_i}{\partial t} = k_i \frac{\partial^2 u_i}{\partial x^2}, \quad i = 1,\ 2, \quad t_0 < t,$$

$$u_1(0,t) = u^* < 0, \quad t > 0,$$

$$\left.\begin{aligned} & u_1(\alpha,t) = u_2(\alpha,t) = 0 \\ & \sigma \frac{\partial \alpha}{\partial t} = k_1 \frac{\partial u_1}{\partial x} - k_2 \frac{\partial u_2}{\partial x} \end{aligned}\right\} \quad \text{on} \quad x = \alpha(t), \quad t_0 < t,$$

where the solution u represents the temperature. This problem has the exact solution,

$$\alpha(t) = 2\,\phi\,\sqrt{\kappa_1 t}\,,$$

$$u_1(x,t) = u^* \left\{ 1 - \frac{\mathrm{erf}(x/2\sqrt{\kappa_1 t}\,)}{\mathrm{erf}\,\phi} \right\},$$

$$u_2(x,t) = u_0 \left\{ 1 - \frac{\mathrm{erfc}(x/2\sqrt{\kappa_2 t}\,)}{\mathrm{erfc}(\phi\sqrt{\kappa_1/\kappa_2})} \right\},$$

where $\kappa_i = k_i/C_i$, $\mathrm{erf}(x)$ is the error function, u^* and u_0 are two constants, and ϕ is the root of the transcendental equation

$$\frac{e^{-\phi^2}}{\mathrm{erf}\,\phi} + \frac{k_2}{k_1}\sqrt{\frac{\kappa_1}{\kappa_2}}\,\frac{u_0\,e^{-\kappa_1\phi^2/\kappa_2}}{u^*\,\mathrm{erfc}(\phi\sqrt{\kappa_1/\kappa_2})} + \frac{\phi\lambda\sqrt{\pi}}{C_1 u^*} = 0,$$

which can be easily computed, say by using the bisection method. The exact solution is used as the initial condition at time $t_0 = 0.5$, as well as the boundary condition at both ends $x = 0$ and $x = 1$. The following thermal properties are used:

$$k_1 = 2.22, \quad k_2 = 0.556, \quad C_1 = 1.762, \quad C_2 = 4.226, \quad \sigma = 338$$

with $u^* = -20$ and $u_0 = 10$, which gives $\phi = 0.2054269\ldots$.

Table 9.1. *A grid refinement analysis for the Stefan problem at $t = 1.0$.*

M	$\| E_M \|_\infty$	Ratio	$\|E_\alpha\|$	Ratio
20	4.3067×10^{-3}		1.0941×10^{-4}	
40	9.7147×10^{-4}	4.4333	2.4947×10^{-5}	4.3857
80	2.3713×10^{-4}	4.0967	5.7298×10^{-6}	4.3539
160	5.8160×10^{-5}	4.0772	1.3828×10^{-6}	4.1434
320	1.4213×10^{-5}	4.0920	3.3721×10^{-7}	4.1009

Table 9.1 shows the results of a grid refinement analysis. In the table, $\| E_M \|_\infty$ is the infinity norm of the error at the fixed time $t = 1$. E_α is the difference between the exact $\alpha(t)$ and the computed interface at the final time $t = 1$. We see clearly second-order convergence for both the temperature $u(x,t)$ and the interface location $\alpha(t)$. More examples, including nonlinear moving interface problems, can be found in [162]. In [173], the method is applied to simulate the temperature profile of an ice sheet during the process of glaciation where there are two interfaces: one is fixed, and the other is moving with time.

Multidimensional problems are discussed in Chapter 11. There we use a time splitting method and combine the IIM with evolution schemes such as the front-tracking and level set methods to evolve moving interfaces.

9.3 The modified ADI method for heat equations with discontinuities

In this section, we explain the alternating direction implicit (ADI) method for the heat equation

$$
\begin{gathered}
u_t = u_{xx} + u_{yy} - f(x,y,t), \qquad (x,y) \in \Omega \backslash \Gamma, \\
[u] = w(:,t), \qquad \left[\frac{\partial u}{\partial \mathbf{n}}\right] = v(:,t)
\end{gathered}
\tag{9.36}
$$

with a fixed interface Γ.

For parabolic equations, it is often desirable to use implicit methods because the time step restriction is severe for explicit methods. For a heat equation, the ADI method is often used to solve parabolic PDEs numerically. The ADI method is unconditionally stable with second-order accuracy in both time and space. From one time level to the next, we need only solve a sequence of tridiagonal systems of equations. We refer the reader to [33, 68, 201, 277] for an introduction to the ADI method.

The classical ADI method for the heat equation is

$$
\begin{aligned}
\frac{U_{ij}^{n+\frac{1}{2}} - U_{ij}^{n}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} + \delta_{yy} U_{ij}^{n} - f_{ij}^{n+\frac{1}{2}}, \\
\frac{U_{ij}^{n+1} - U_{ij}^{n+\frac{1}{2}}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} + \delta_{yy} U_{ij}^{n+1} - f_{ij}^{n+\frac{1}{2}},
\end{aligned}
\tag{9.37}
$$

where, for example,

$$\delta_{xx} U_{ij} = \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{h^2}. \tag{9.38}$$

The local truncation error of the above ADI method contains such a term as

$$(\Delta t)^2\, \delta_{xx}\, \delta_{yy}\, u_t \approx (\Delta t)^2 u_{txxyy}, \tag{9.39}$$

(see (9.52)) which indicates a strong regularity requirement on the solution in order to get second-order accurate results. Note that we again assume that the ratio $h/\Delta t$ is a constant.

Obviously, we cannot apply the ADI method directly to (9.36) since the solution is not even in $C(\Omega)$. It seems that we can use the IIM to add correction terms to the spatial discretization to get an ADI scheme,

$$\begin{aligned} \frac{U_{ij}^{n+\frac{1}{2}} - U_{ij}^{n}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} + \delta_{yy} U_{ij}^{n} - C_{ij}^{n+\frac{1}{2}} - f_{ij}^{n+\frac{1}{2}}, \\ \frac{U_{ij}^{n+1} - U_{ij}^{n+\frac{1}{2}}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} + \delta_{yy} U_{ij}^{n+1} - C_{ij}^{n+\frac{1}{2}} - f_{ij}^{n+\frac{1}{2}}, \end{aligned} \tag{9.40}$$

but this does not lead to a second-order solution. In fact, theoretical analysis and numerical experiments show that scheme (9.40) is only first-order accurate. The failure results from the fact that we have not split the correction term $C_{i,j}^{n+\frac{1}{2}}$ correctly.

9.3.1 The modified ADI scheme

The modified ADI method can be written as

$$\begin{aligned} \frac{U_{ij}^{n+\frac{1}{2}} - U_{ij}^{n}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^{n} - R_{ij}^{n} \\ &\quad + \delta_{yy} U_{ij}^{n} - (C_y)_{ij}^{n} - f_{ij}^{n+\frac{1}{2}}, \end{aligned} \tag{9.41}$$

$$\begin{aligned} \frac{U_{ij}^{n+1} - U_{ij}^{n+\frac{1}{2}}}{\Delta t/2} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^{n} - R_{ij}^{n} \\ &\quad + \delta_{yy} U_{ij}^{n+1} - (C_y)_{ij}^{n+1} - f_{ij}^{n+\frac{1}{2}}, \end{aligned} \tag{9.42}$$

where we add the correction terms Q_{ij}^n, R_{ij}^n, $(C_x)_{ij}^{n+\frac{1}{2}}$, $(C_y)_{ij}^n$, and $(C_y)_{ij}^{n+1}$ to get a second-order scheme. At regular grid points, that is, all the grid points in the standard ADI finite difference stencil that are on the same side of the interface, the standard ADI method is used in which

$$(C_x)_{ij}^{n+\frac{1}{2}} = Q_{ij}^n = R_{ij}^n = (C_y)_{ij}^n = Q_{ij}^n = (C_y)_{ij}^{n+1} = 0.$$

At each irregular grid point we need to determine those correction terms. This will be explained in the following subsections.

9.3.2 Determining the spatial correction terms

The local truncation error at regular grid points is $O(h^2)$. To get second-order accuracy globally, we need to determine the correction terms so that the local truncation error is of order $O(h)$ or smaller in magnitude at irregular grid points. First we explain how to approximate u_{xx} and u_{yy} by choosing the correction terms C_x and C_y. Then we explain how to choose the correction terms R_{ij}^n and Q_{ij}^n so that the local truncation error of the equivalent finite difference scheme at each irregular grid point is $O(h)$.

Let (x_i, y_j) be an irregular grid point. Without loss of generality, we assume that the interface Γ cuts the grid line $y = y_j$ at $x = x_{ij}^*$, where $x_i \le x_{ij}^* \le x_{i+1}$. Using the Taylor expansion in the x-direction at x_{ij}^*, or x^* for short, and substituting U_{ij} with the exact solution $u_{ij} = u(x_i, y_j)$, we have

$$
\begin{aligned}
\frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{h^2} &= \frac{u^- - 2u^- + u^+}{h^2} \\
&\quad + u_x^- \left(\frac{x_{i-1} - x^*}{h^2} - 2\,\frac{x_i - x^*}{h^2} \right) + u_x^+ \,\frac{x_{i+1} - x^*}{h^2} \\
&\quad + u_{xx}^- \left(\frac{(x_{i-1} - x^*)^2}{2h^2} - 2\frac{(x_i - x^*)^2}{2h^2} \right) + u_{xx}^+ \frac{(x_{i+1} - x^*)^2}{2h^2} + O(h) \\
&= u_{xx}^- + \frac{[u]}{h^2} + [u_x]\,\frac{(x_{i+1} - x^*)}{h^2} + [u_{xx}]\,\frac{(x_{i+1} - x^*)^2}{2h^2} + O(h) \\
&= u_{xx}^- + (C_x)_{ij} + O(h),
\end{aligned}
$$

where we have used the Taylor expansion

$$
u(x, y_j) = u^\pm + u_x^\pm \left(x - x_{ij}^*\right) + \frac{1}{2} u_{xx}^\pm \left(x - x_{ij}^*\right)^2 + O(h^3),
$$

and u^-, u_x^-, u_{xx}^-, and the jumps are defined at $(x_{ij}^*, y_j) \in \Gamma$.

Similarly, at the point (x_{i+1}, y_j), we have

$$
\begin{aligned}
&\frac{u(x_i, y_j) - 2u(x_{i+1}, y_j) + u(x_{i+2}, y_j)}{h^2} \\
&= u_{xx}^+ - \frac{[u]}{h^2} - [u_x]\,\frac{(x_i - x^*)}{h^2} - [u_{xx}]\,\frac{(x_i - x^*)^2}{2h^2} + O(h) \\
&= u_{xx}^+ + (C_x)_{i+1,j} + O(h).
\end{aligned}
$$

In other words, we can write

$$
\delta_{xx} u_{ij} = u_{xx}^\pm + (C_x)_{ij} + O(h) \quad \text{at irregular points,} \tag{9.43}
$$

where

$$
(C_x)_{ij} = \pm \frac{[u]}{h^2} \pm [u_x]\,\frac{(x_{i\pm 1} - x^*)}{h^2} \pm [u_{xx}]\,\frac{(x_{i\pm 1} - x^*)^2}{2h^2}. \tag{9.44}
$$

If the point (x_i, y_j) is regular, then $(C_x)_{ij} = 0$; otherwise it can be expressed in terms of the jumps of $[u]$, $[u_x]$, and $[u_{xx}]$. The sign is determined by the relative position of the interface Γ and the grid point (x_i, y_j). By the same token we can do the same in the y-direction to get

$$\delta_{yy} u_{ij} = u_{yy}^{\pm} + (C_y)_{ij} + O(h) \quad \text{at irregular points.} \tag{9.45}$$

9.3.3 Decomposing the jump condition in the coordinate directions

To determine the correction terms $(C_x)_{ij}$ and $(C_y)_{ij}$, the values of the jumps $[u_x]$, $[u_y]$, $[u_{xx}]$, $[u_{yy}]$ are needed in terms of the given information $[u]$ and $[u_{\boldsymbol{n}}]$. These interface relations can be obtained by differentiating the known jump conditions $[u]$ and $[u_{\boldsymbol{n}}]$ and using the differential equation itself.

Using the local coordinate system (1.34), we obtain $[u_\eta] = [u]_\eta = w_\eta(:, t)$ by differentiating $[u] = w$ along the interface. For the remaining second derivative jumps we can use the interface relations from (3.5), with $\beta = 1$ and f being replaced by $f + u_t$, to get

$$\begin{aligned} [u_{\xi\xi}] &= \chi''[u_\xi] - w_{\eta\eta} + [f] + w_t, \\ [u_{\eta\eta}] &= -\chi''[u_\xi] + w_{\eta\eta}, \\ [u_{\xi\eta}] &= \chi''[u_\eta] + v_\eta, \end{aligned} \tag{9.46}$$

where $[u_\xi] = [u_{\boldsymbol{n}}]$ is the given jump condition in the flux. Thus, we have expressed all the jumps in the local coordinates in terms of the known quantities $[u]$ and $[u_{\boldsymbol{n}}]$. The jump relations in the x- and y-directions then are given by the following formulas:

$$\begin{aligned} [u_x] &= [u_\xi] \cos\theta - [u_\eta] \sin\theta, \\ [u_y] &= [u_\xi] \sin\theta + [u_\eta] \cos\theta, \\ [u_{xx}] &= [u_{\xi\xi}] \cos^2\theta - 2[u_{\xi\eta}] \cos\theta \sin\theta + [u_{\eta\eta}] \sin^2\theta, \\ [u_{yy}] &= [u_{\xi\xi}] \sin^2\theta + 2[u_{\xi\eta}] \cos\theta \sin\theta + [u_{\eta\eta}] \cos^2\theta. \end{aligned} \tag{9.47}$$

With these known jumps we can compute the correction terms $(C_x)_{ij}$ and $(C_y)_{ij}$.

9.3.4 The local truncation error analysis for the ADI method

In this subsection we discuss how to determine Q_{ij} and R_{ij} through the local truncation error analysis. Now if we add (9.41) and (9.42) together, we get

$$\begin{aligned} \frac{U_{ij}^{n+1} - U_{ij}^n}{\Delta t} &= \delta_{xx} U_{ij}^{n+\frac{1}{2}} - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^n - R_{ij}^n \\ &+ \frac{1}{2}\left(\delta_{yy} U_{ij}^n + \delta_{yy} U_{ij}^{n+1}\right) - \frac{1}{2}\left((C_y)_{ij}^n + (C_y)_{ij}^{n+1}\right) - f_{ij}^{n+\frac{1}{2}}. \end{aligned} \tag{9.48}$$

If we subtract (9.42) from (9.41), we have the intermediate result

$$U_{ij}^{n+\frac{1}{2}} = \frac{U_{ij}^n + U_{ij}^{n+1}}{2} + \frac{\Delta t}{4}\left(\delta_{yy}U_{ij}^n - (C_y)_{ij}^n - \delta_{yy}U_{ij}^{n+1} + (C_y)_{ij}^{n+1}\right). \tag{9.49}$$

Plugging this into (9.48), we get

$$\begin{aligned}\frac{U_{ij}^{n+1} - U_{ij}^n}{\Delta t} &= \frac{1}{2}\delta_{xx}\left(U_{ij}^n + U_{ij}^{n+1}\right) - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^n - R_{ij}^n \\ &\quad + \frac{1}{2}\left(\delta_{yy}U_{ij}^n - (C_y)_{ij}^n + \delta_{yy}U_{ij}^{n+1} - (C_y)_{ij}^{n+1}\right) \\ &\quad + \frac{\Delta t}{4}\delta_{xx}\left(\delta_{yy}U_{ij}^n - (C_y)_{ij}^n - \delta_{yy}u_{ij}^{n+1} + (C_y)_{ij}^{n+1}\right).\end{aligned} \tag{9.50}$$

This is the *actual finite difference scheme* for the solution U_{ij}^{n+1} at time t^{n+1}. Note that the interface is fixed and all the quantities are continuous with time. We check the local truncation error of the finite difference scheme by examining each term, with U_{ij}^n being substituted for by the exact solution $u(x_i, y_j, t^n)$, to obtain the local truncation error of order $O(h)$. The left-hand side in (9.50), when substituted with the exact solution, gives

$$\frac{u(x_i, y_j, t^{n+1}) - u(x_i, y_j, t^n)}{\Delta t} = u_t(x_i, y_j, t^{n+1/2}) + O(\Delta t^2). \tag{9.51}$$

If we substitute for U_{ij} with the exact solution, the first few terms on the right-hand side can be written as

$$\begin{aligned}&\frac{1}{2}\delta_{xx}\left(u(x_i, y_j, t^n) + u(x_i, y_j, t^{n+1})\right) - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^n \\ &\quad = \frac{1}{2}\delta_{xx}u(x_i, y_j, t^{n+\frac{1}{2}}) + (\Delta t)^2\ \delta_{xx}u_{tt}(x_i, y_j, t^{n+\frac{1}{2}}) - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^n + O(\Delta t).\end{aligned}$$

When u is continuous, that is, $[u] = w = 0$, we conclude that

$$\Delta t^2\ \delta_{xx}u_{tt}(x_i, y_j, t^{n+\frac{1}{2}}) \sim \frac{(\Delta t)^2}{h} \sim h. \tag{9.52}$$

We can simply take $Q_{ij}^n \equiv 0$. If u is not continuous, we rewrite the expression above as

$$\begin{aligned}&\frac{1}{2}\delta_{xx}\left(u(x_i, y_j, t^n) + u(x_i, y_j, t^{n+1})\right) - (C_x)_{ij}^{n+\frac{1}{2}} - Q_{ij}^n \\ &\quad = \frac{1}{2}\left(\delta_{xx}u(x_i, y_j, t^n) - (C_x)_{ij}^n\right) + \frac{1}{2}\left(\delta_{xx}u(x_i, y_j, t^{n+1}) - (C_x)_{ij}^{n+1}\right) \\ &\qquad - (C_x)_{ij}^{n+\frac{1}{2}} + \frac{1}{2}\left((C_x)_{ij}^n + (C_x)_{ij}^{n+1}\right) - Q_{ij}^n \\ &\quad = \frac{1}{2}\left((u_{xx})_{ij}^n + (u_{xx})_{ij}^{n+1}\right) - (C_x)_{ij}^{n+\frac{1}{2}} + \frac{1}{2}\left((C_x)_{ij}^n + (C_x)_{ij}^{n+1}\right) - Q_{ij}^n + O(h) \\ &\quad = u_{tt}(x_i, y_j, t^{n+\frac{1}{2}}) - (C_x)_{ij}^{n+\frac{1}{2}} + \frac{1}{2}\left((C_x)_{ij}^n + (C_x)_{ij}^{n+1}\right) - Q_{ij}^n + O(h).\end{aligned}$$

It is clear now that we should take

$$Q_{ij}^{n} = \frac{1}{2}\left((C_x)_{ij}^{n} + (C_x)_{ij}^{n+1}\right) - (C_x)_{ij}^{n+\frac{1}{2}}. \tag{9.53}$$

We turn our attention to the terms remaining in (9.50). If $[u] = 0$, we have

$$\begin{aligned}
&\frac{1}{2}\left(\delta_{yy}u(x_i, y_j, t^n) - (C_y)_{ij}^{n} + \delta_{yy}u(x_i, y_j, t^{n+1}) - (C_y)_{ij}^{n+1}\right)\\
&\quad = \frac{1}{2}\left(u_{yy}(x_i, y_j, t^n) + u_{yy}(x_i, y_j, t^{n+1})\right) + O(h)\\
&\quad = u_{yy}(x_i, y_j, t^{n+\frac{1}{2}}) + O(h).
\end{aligned} \tag{9.54}$$

Thus, there is no need to correct the expression above and we set $R_{ij}^{n} = 0$. If $[u] \neq 0$, we define

$$S_{ij}^{n} = \frac{\delta_{yy}u(x_i, y_j, t^n) - (C_y)_{ij}^{n}}{h}, \tag{9.55}$$

which is an $O(1)$ quantity. Since the interface is fixed and all quantities are continuous with time, we can conclude that $|S_{ij}^{n} - S_{ij}^{n+1}| = O(h)$. Hence, from (9.45), we know that

$$\begin{aligned}
&\delta_{yy}u(x_i, y_j, t^n) - (C_y)_{ij}^{n} - \delta_{yy}u(x_i, y_j, t^{n+1}) + (C_y)_{ij}^{n+1}\\
&\quad = u_{yy}(x_i, y_j, t^n) - u_{yy}(x_i, y_j, t^{n+1}) + O(h^2).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\frac{\Delta t}{4}\delta_{xx}\left(\delta_{yy}u(x_i, y_j, t^n) - (C_y)_{ij}^{n} - \delta_{yy}u(x_i, y_j, t^{n+1}) + (C_y)_{ij}^{n+1}\right) - R_{ij}^{n}\\
&\quad = \frac{\Delta t}{4}\delta_{xx}\left(u_{yy}(x_i, y_j, t^n) - u_{yy}(x_i, y_j, t^{n+1})\right) + O(h) - R_{ij}^{n}\\
&\quad = -\frac{\Delta t^2}{4}\delta_{xx}u_{yyt}(x_i, y_j, t^n) - R_{ij}^{n} + O(h).
\end{aligned}$$

We can see that the term which needs to be offset is $(\Delta t)^2\delta_{xx}u_{yyt}(x_i, y_j, t^n)/4 \sim O(1)$. We need to approximate this term at least to first-order accuracy in order to make the right correction. Note that

$$\begin{aligned}
\frac{(\Delta t)^2}{4}\delta_{xx}u_{yyt}(x_i, y_j, t^n) = \frac{(\Delta t)^2}{4h^2}\Big(&u_{yyt}(x_{i-1}, y_j, t^n) - 2u_{yyt}(x_i, y_j, t^n)\\
&+ u_{yyt}(x_{i+1}, y_j, t^n)\Big) \pm \frac{\Delta t^2}{4h^2}[u_{yyt}](x_{ij}^{*}, y_j, t^n) + O(h),
\end{aligned} \tag{9.56}$$

where the sign is determined by the relative position of the (x_i, y_j) and the interface. The jump term $[u_{yyt}]$ above can be approximated by

$$[u_{yyt}](x_{ij}^*, y_j, t^n) = \frac{[u_{yyt}](x_{ij}^*, y_j, t^{n+1}) - [u_{yyt}](x_{ij}^*, y_j, t^n)}{\Delta t} + O(h). \tag{9.57}$$

At last we can determine the correction term R_{ij}^n as

$$R_{ij}^n = \mp \frac{[u_{yy}]^{n+1} - [u_{yy}]^n}{4\,\Delta t}, \tag{9.58}$$

where the jump is calculated at (x_i, y_{ij}^*). From the analysis above we know that if we take Q_{ij}^n and R_{ij}^n as in (9.53) and (9.58), then we can guarantee that the local truncation errors are $O(h^2)$, at regular grid points, and $O(h)$ at the irregular grid points near the interface. The finite difference scheme will still give a second-order accurate solution globally.

9.3.5 A numerical example of the modified ADI method

We consider an example in which the interface Γ is the circle $x^2 + y^2 = 1/4$. The exact solution is

$$u(x, y, t) = \begin{cases} e^{-t} J_0(r) & \text{if } r \le \dfrac{1}{2}, \\[2ex] \dfrac{e^{-t} J_0(0.5)\, Y_0(r)}{Y_0(0.5)} & \text{if } r > \dfrac{1}{2}, \end{cases} \tag{9.59}$$

where $J_0(x)$ and $Y_0(x)$ are the Bessel functions of first and second kind of order zero, respectively. In this example the solution $u(x, y, t)$ is continuous across the interface $r = \sqrt{x^2 + y^2} = 1/2$, but it has a jump in the flux which is

$$v(t) = e^{-t}\left(Y_0'(0.5)\, J_0(0.5)/Y_0(0.5) - J_0'(0.5)\right). \tag{9.60}$$

The heat equation is defined on the square $-1 \le x,\ y \le 1$. The Dirichlet boundary condition and the initial condition are taken from the exact solution.

Tables 9.2 and 9.3 show the results of grid refinement analysis at $t = 5$ with and without the correction terms $Q_{i,j}^n$ and $R_{i,j}^n$, respectively. In the tables, $\|E_N\|$ is the maximum norm of the computed solution over all the grid points, while $\|T_N\|$ is the maximum norm of the local truncation error. For a second-order method, the ratio of the errors approaches 4. We see that the method with correction terms $Q_{i,j}^n$ and $R_{i,j}^n$ behaves better. But the method without correction terms Q_{ij}^n and R_{ij}^n seems also to approach second-order accuracy. This is likely due to fortunate cancellations of errors for this particular example.

Table 9.2. *A grid refinement analysis of the example with continuous solution using the modified ADI method with correction terms $Q^n_{i,j}$ and $R^n_{i,j}$.*

N	$\| E_N \|_\infty,\ \ t=5$	Ratio 1	$\| T_N \|_\infty,\ \ t=0$	Ratio 2
20	5.39851×10^{-5}		4.04973×10^{-1}	
40	1.01368×10^{-5}	4.6345	1.885347×10^{-1}	2.1480
80	2.30004×10^{-6}	4.4072	8.94152×10^{-2}	2.1085
160	5.56747×10^{-7}	4.1312	4.33057×10^{-2}	2.0647

Table 9.3. *A grid refinement analysis of the example with continuous solution using the modified ADI method without correction terms Q^n_{ij} and R^n_{ij}.*

N	$\| E_N \|_\infty,\ \ t=5$	Ratio 1	$\| T_N \|_\infty,\ \ t=0$
20	7.70093×10^{-5}		2.32077
40	2.07288×10^{-5}	3.7151	2.60534
80	5.44382×10^{-6}	3.8078	2.73938
160	1.42010×10^{-6}	3.8334	2.80317

9.4 The IIM for diffusion and advection equations

The maximum principle preserving IIM for elliptic interface problems can be readily applied to diffusion and advection equations of the following form:

$$u_t + \mathbf{a}(\mathbf{x},t)\cdot\nabla u = \nabla\cdot(\beta\nabla u) + f, \quad \mathbf{x}\in\Omega=\Omega^+\cup\Omega^-, \quad x\notin\Gamma, \tag{9.61a}$$

$$[u]|_\Gamma = w, \quad [\beta u_{\boldsymbol{n}}]|_\Gamma = v, \tag{9.61b}$$

$$u(\mathbf{x},0)=u_0(\mathbf{x}), \quad u(\mathbf{x},t)|_{\partial\Omega}=g(t), \tag{9.61c}$$

where $\beta(\mathbf{x},t)\ge\beta_{min}>0$, $\mathbf{a}(\mathbf{x},t)=(\alpha_1,\alpha_2)$, and $f(\mathbf{x},t)$ are piecewise continuous but may have a finite jump across a fixed interface Γ. The time discretization is based on the prediction-correction Crank–Nicolson discretization,

$$\begin{aligned}\frac{U^{n+1}-U^n}{\Delta t} &+ (\mathbf{a}\cdot\nabla_h U)^{n+\frac12}\\ &= \frac12\left((\nabla_h\cdot\beta\nabla_h U)^n + (\nabla_h\cdot\beta\nabla_h U)^{n+1}\right) + f^{n+\frac12},\end{aligned} \tag{9.62}$$

where

$$(\mathbf{a}\cdot\nabla_h U)^{n+\frac12} = \frac32\,(\mathbf{a}\cdot\nabla_h U)^n - \frac12\,(\mathbf{a}\cdot\nabla_h U)^{n-1}, \tag{9.63}$$

and ∇_h is the discrete gradient operator. This discretization is second-order accurate in time. The diffusion term is discretized implicitly so that we can take large time steps, while the first-order derivative term is discretized explicitly so that second-order accuracy can be achieved without affecting the stability of the discretization of the diffusion part.

At a regular grid point (x_i, y_j), where all the points in the centered 5-point stencil are on the same side of the interface, the discretization is the standard one,

$$(\nabla_h U)_{ij} = \left(\frac{U_{i+1,j} - U_{i-1,j}}{2h}, \quad \frac{U_{i,j+1} - U_{i,j-1}}{2h} \right), \tag{9.64}$$

$$\begin{aligned}(\nabla_h \cdot (\beta \nabla_h) U)_{ij} &= \frac{1}{h} \left(\beta_{i+\frac{1}{2},j} \frac{(U_{i+1,j} - U_{ij})}{h} - \beta_{i-\frac{1}{2},j} \frac{(U_{i,j} - U_{i-1,j})}{h} \right) \\ &\quad + \frac{1}{h} \left(\beta_{i,j+\frac{1}{2}} \frac{(U_{i,j+1} - U_{i,j})}{h} - \beta_{i,j-\frac{1}{2}} \frac{(U_{i,j} - U_{i,j-1})}{h} \right),\end{aligned} \tag{9.65}$$

where we have omitted the time index for simplicity.

At an irregular grid point (x_i, y_j), where the central 5-point stencil consists of grid points from different sides of the interface Γ, the discretization is done using a method of undetermined coefficients. We let

$$(\mathbf{a} \cdot \nabla_h U)_{ij} = \sum_{k=1}^{L_2} \bar{\gamma}_k U_{i+i_k, j+j_k} - Q_{ij}, \tag{9.66}$$

$$(\nabla_h \cdot \beta \nabla_h)\, U_{ij} = \sum_{k=1}^{L_1} \gamma_k U_{i+i_k, j+j_k} - C_{ij}, \tag{9.67}$$

where L_1 and L_2 are the numbers of grid points involved in the finite difference stencil. The sum over k involves a finite number of points neighboring (x_i, y_j). So each i_k and j_k will take values in the set $\{0, \pm 1, \pm 2, \ldots\}$. The coefficients $\{\gamma_k\}$ and the indexes i_k, j_k depend on (i, j), but for simplicity of notation, the dependency has been dropped. In order to determine the coefficients, we choose a point (x_i^*, y_j^*) on the interface. Below we discuss how to find the coefficients for the diffusion and advection terms, respectively.

9.4.1 Determining the finite difference coefficients for the diffusion term

The coefficients for the approximation of the diffusion term $\nabla \cdot (\beta \nabla u)$ are almost the same as in the maximum preserving principle IIM discussed in §3.5. The modification is needed because the last interface condition at the local coordinates (1.34) centered at (x_i^*, y_j^*) in (3.5) now is

$$\begin{aligned}u^+_{\xi\xi} &= \rho u^-_{\xi\xi} + (\rho - 1)\, u^-_{\eta\eta} + \left((\rho - 1)\chi'' + \frac{\beta^-_\xi - \rho \beta^+_\xi + \rho \tilde{\alpha}^+_1 - \tilde{\alpha}^-_1}{\beta^+} \right) u^-_\xi \\ &\quad + \frac{[\tilde{\alpha}_2 - \beta_\eta]}{\beta^+}\, u^-_\eta + \frac{\tilde{\alpha}^+_2 - \beta^+_\eta}{\beta^+}\, w_\eta - w_{\eta\eta} \\ &\quad + \frac{1}{\beta^+}\, w_t + \left(\frac{\chi''}{\beta^+} + \frac{\tilde{\alpha}^+_1 - \beta_\xi{}^+}{(\beta^+)^2} \right) v - \frac{[f]}{\beta^+},\end{aligned} \tag{9.68}$$

due to the time derivative, where

$$\tilde{\alpha}_1 = \alpha_1 \cos\theta + \alpha_2 \sin\theta, \qquad \tilde{\alpha}_2 = -\alpha_1 \sin\theta + \alpha_2 \cos\theta \tag{9.69}$$

in the local coordinates (1.34). The equality constraints now are

$$
\begin{aligned}
a_1 + a_2 &= 0,\\
a_3 + \rho a_4 + a_8 \left((\rho - 1)\chi'' + \frac{\beta_\xi^- - \rho\beta_\xi^+ + \rho\tilde{\alpha}_1^+ - \tilde{\alpha}_1^-}{\beta^+} \right) &\\
+ a_{10}\,(1-\rho)\chi'' + a_{12}\,\frac{{\beta_\eta}^- - {\beta_\eta}^+\rho}{\beta^+} &= {\beta_\xi}^-,\\
a_5 + a_6 + a_8 \frac{[\tilde{\alpha}_2 - \beta_\eta]}{\beta^+} + a_{12}(1-\rho)\,\chi'' &= {\beta_\eta}^-,\\
a_7 + a_8\,\rho &= \beta^-,\\
a_9 + a_{10} + a_8\,(\rho - 1) &= \beta^-,\\
a_{11} + a_{12}\,\rho &= 0.
\end{aligned}
\tag{9.70}
$$

The sign constraints are exactly the same as in §3.5. The correction term is

$$
\begin{aligned}
C_{ij} = a_2\, w &+ \left(a_6 + a_8 \frac{\tilde{\alpha}_2^+ - {\beta_\xi}^+}{\beta^+} \right) w_\eta + (a_{10} - a_8) w_{\eta\eta} + \frac{a_8}{\beta^+}\, w_t\\
&+ \frac{1}{\beta^+} \left\{ a_4 + a_8 \left(\chi'' + \frac{\tilde{\alpha}_1^+ - {\beta_\xi}^+}{\beta^+} \right) - a_{10}\chi'' - a_{12}\frac{{\beta_\eta}^+}{\beta^+} \right\} v\\
&+ a_{12}\,\frac{1}{\beta^+}\, v_\eta - a_8 \frac{[f]}{\beta^+},
\end{aligned}
\tag{9.71}
$$

where the $\{a_i\}$'s are defined as in (3.17).

9.4.2 Determining the finite difference coefficients for the advection term

Borrowing an idea from the projection method for the Navier–Stokes equations (see [168] for example), we use an explicit discretization for the first-order derivative term since it involves only first-order derivatives of the solution. The truncation error of the finite difference approximation (9.66) to the first-order derivative term at (x_i^*, y_j^*) is

$$
T_{ij}^r = \sum_{k=1}^{L_2} \bar{\gamma}_k\, u(x_{i+i_k}, y_{j+j_k}, t) - Q_{ij} - (\mathbf{a} \cdot \nabla u)^- . \tag{9.72}
$$

As mentioned in [154, 166], we can require the truncation error to be $O(h)$ at irregular grid points without affecting the second-order accuracy as h approaches zero. Therefore, we use the standard centered 5-point stencil ($L_2 = 5$), and the linear system of equations is

$$
\begin{aligned}
\bar{a}_1 + \bar{a}_2 &= 0,\\
\bar{a}_3 + \rho\bar{a}_4 &= \tilde{\alpha}_1^-,\\
\bar{a}_5 + \bar{a}_6 &= \tilde{\alpha}_2^-,
\end{aligned}
\tag{9.73}
$$

where we have neglected higher-order terms of h. The $\{\bar{a}_k\}$'s, for example, are defined as

$$\bar{a}_1 = \sum_{k \in K^-} \bar{\gamma}_k, \qquad \bar{a}_2 = \sum_{k \in K^+} \bar{\gamma}_k, \qquad \bar{a}_3 = \sum_{k \in K^-} \xi_k \bar{\gamma}_k.$$

They are literally the same as those defined in (3.17), with $\{\gamma_k\}$ being replaced by $\bar{\gamma}_k$. The solution to the linear system of equations is also different. The system is an underdetermined system and the solution is defined as the least squares solution with the least 2-norm. The correction term then is

$$Q_{ij} = \bar{a}_2\, w + \bar{a}_6\, w_\eta + \frac{\bar{a}_4}{\beta^+}\, v. \tag{9.74}$$

Note that the algorithm described in this section is second-order accurate, that is, the global error is proportional to h^2 as h approaches zero. In practice, however, h is fixed. If the advection term is very strong ($\|\mathbf{a}\| > 1/h$), the advection may carry larger truncation errors near the interface to other parts of the domain. The global error then may be affected by such error propagation.

From the analysis of the projection method for Navier–Stokes equations (the problem here can be regarded as a special case) (see for example, [16]), we know that the time step size should satisfy

$$\Delta t \le \frac{h}{\sqrt{2}\, \|\mathbf{a}\|_2}$$

if there is no interface. For the interface problem, we suggest taking a conservative time step size,

$$\Delta t \le \frac{h}{2\, \|\mathbf{a}\|_2}. \tag{9.75}$$

That is, we reduce the time step size by a factor of $\sqrt{2}$.

Recently, Adams and Li developed a new multigrid method for solving the linear system of equations using a 9-point stencil. See [6] for the method and some numerical examples.