One-Dimensional Stefan Problem

Tracy Backes

May 5, 2007

1 Introduction

Working with systems that involve moving boundaries can be a very difficult task. Not only do we have to solve the equations describing the system, but we also have to find the region the system occupies at each step. One of the common moving-boundary classes, Stefan problems are systems of diffusion or heat-conduction where the boundaries between the different phases in the system change over time. For example, the solidification of water into ice can be formulated as a Stefan problem. Unfortunately, because Stefan problems can become so difficult to handle, there is often no way to analytically solve the system. Therefore, mathematicians have spent a lot of time exploring which numerical methods are most practical for working with these problems.

As we shall soon see, even the case of freezing water can quickly become tricky to deal with. Therefore, for my project I focused on the very simplified case of a freezing liquid. I decided to focus on a very symmetric problem so that I could essentially describe the system one-dimensionally. In order to solve this system I based my work off of a paper by researchers Cauldwell and Chiu.

2 Background

Consider a solid pipe of radius a running through an infinite volume of liquid. If the liquid is initially at freezing temperature T_f and the pipe is maintained at some colder temperature T_s, the liquid will begin to solidify around the pipe. A freezing front moves progressively through the liquid such that behind the front the material is in the solid phase, while ahead of the front the material remains in the liquid phase. We further assume that the solid pipe maintains constant temperature T_s, while the material ahead of the front maintains constant temperature T_f; however, we allow Newton heat loss at the freezing front. As illustrated in Figure 1, the system is radially symmetric, so the distance to the freezing front can be described as $R(t)$, the radial distance from the center of the cylinder. For the purposes of this paper, we are interested in determining the motion of this front.

The moving-boundary problem for this system becomes

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial r^2} + \frac{\kappa}{r}\frac{\partial T}{\partial r}, \qquad a < r < R(t), \quad t > 0 \tag{1}$$

$$T(r,t) = T_f, \qquad r \geq R(t), \quad t > 0 \tag{2}$$

$$T(r,t) = T_s, \qquad r = a, \quad t \geq 0 \tag{3}$$

$$K\left(\frac{\partial T}{\partial r}\right)_{R(t)} = L\rho \frac{dR(t)}{dt} \tag{4}$$

where $T(r,t)$ is the temperature of the solid layer at depth r and time t, κ is the thermal diffusivity, K is the conductivity, ρ is the density and L is the latent heat of freezing for the liquid. To make the numerical methods easier, we non-dimensionalize the variables using the following substitutions

$$z = r/a, \qquad tau = \kappa t/a^2, \qquad \alpha = L/c(T_f - T_s), \tag{5}$$

$$U = (T - T_s)/(T_f - T_s), \tag{6}$$

where α is what is commonly referred to as the Stefan number. Using (5) and (6), our new boundary value problem becomes

$$\frac{\partial U}{\partial \tau} = \frac{\partial^2 U}{\partial z^2} + \frac{1}{z}\frac{\partial U}{\partial z}, \qquad 1 < z < Z(t), \quad \tau > 0 \tag{7}$$

$$U(z,\tau) = 1, \qquad z \geq Z(\tau), \quad \tau > 0 \tag{8}$$

$$U(z,\tau) = 0, \qquad z = 1, \quad \tau \geq 0 \tag{9}$$

$$\left(\frac{\partial U}{\partial z}\right)_{Z(\tau)} = \alpha \frac{dZ(\tau)}{d\tau}, \quad Z(0) = 1. \tag{10}$$

In order to determine the behavior of the freezing front, $Z(\tau)$, we must solve this moving-boundary system.

3 Choosing a Method

In order to solve the moving boundary system (7)-(10) I turned to literature on the cylindrical freezing problem. In 1958 Goodman established a method based on satisfying the integral form of the heat conduction equation (Hill, 106). This heat balance integral method (HBIM) has since been widely adopted in literature. Although it has some hangups (especially when it comes to approximating a temperature profile for the system), it is overall a very useful approach. While researching the HBIM I came across a paper by Caldwell and Chiu that extended Goodman's original method to also include spatial sub-divisions. I decided to explore the approach they outlined in their paper because it seemed like I could apply many of the concepts we had seen earlier in class.

4 Extended Heat Balance Integral Method

The basic idea behind the HBIM is to integrate the heat flow equation and substitute in an approximate temperature profile. Solving this system yields an estimate for the

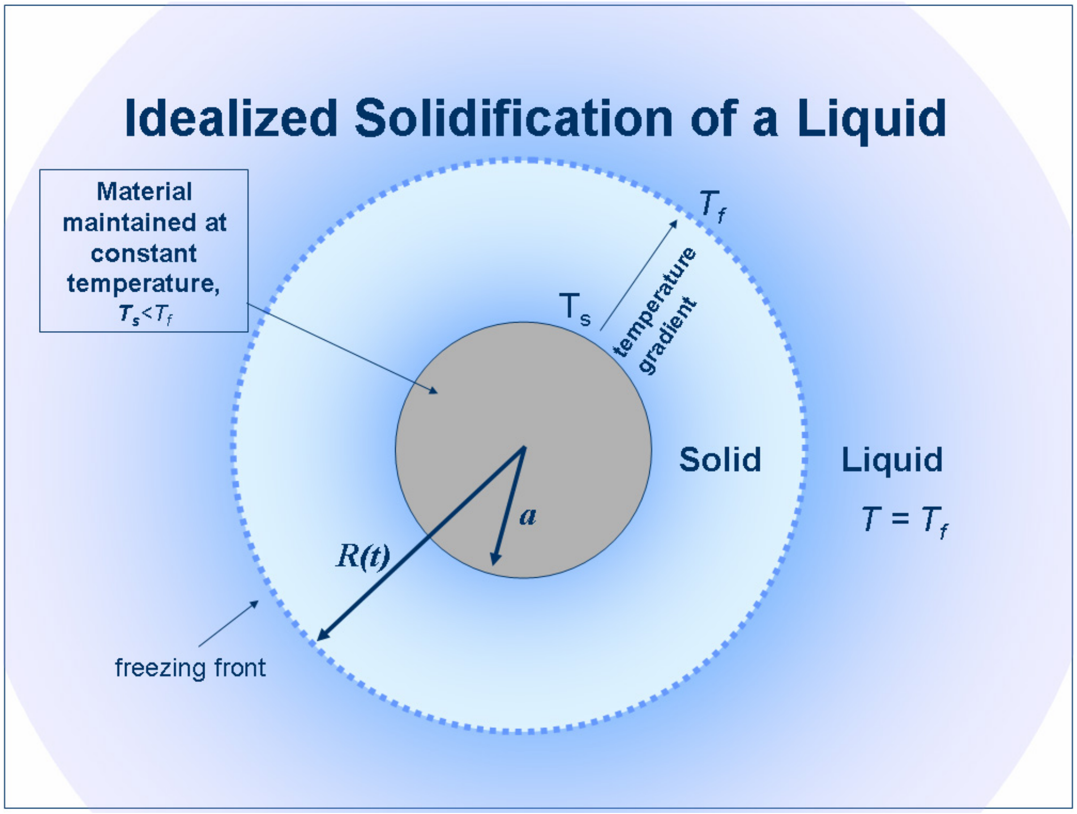
Idealized Solidification of a Liquid
Material maintained at constant temperature, $T_s<T_f$
T_f
T_s
temperature gradient
Solid
Liquid
$T = T_f$
a
$R(t)$
freezing front

Figure 1: Diagram of Cylindrical System

motion of the freezing front. However, as I mentioned earlier, choosing an appropriate temperature profile can be a little tricky. After reviewing extensions to the HBIM by several other groups, Caldwell and Chiu propose performing HBIM with a linear temperature profile while also including spatial subdivisions. Upon applying spatial subdivisions, the problem becomes a system of first-order, non-linear differential equations. Solving this new system eventually yields the location of our freezing front $Z(\tau)$.

In order to implement these spatial divisions, we first divide the temperature into n equal intervals and assume that

$$U = \frac{i}{n} + \frac{z - Z_i}{n(Z_{i+1} - Z_i)}, \qquad Z_i < U < Z_{i+1}. \tag{11}$$

Then, following the general idea of the HBIM, we integrate over each subrange $[Z_i, Z_{i+1}]$, and replace U with the temperature profile we assumed in (11). For the cylindrical case this results in the following n heat balance equations,

$$\begin{aligned} \frac{d}{d\tau}\left[\frac{Z_{i+1}^3 - Z_i^3}{Z_{i+1} - Z_i}\right] &= 6\left[\frac{Z_i}{Z_{i+1} - Z_i} - \frac{Z_{i+1}}{Z_{i+2} - Z_{i+1}}\right], \quad i = 0, 1, \ldots, n-2 \\ \frac{d}{d\tau}\left[\frac{Z_n^3 - Z_{n-1}^3}{Z_n - Z_{n-1}} + 3\alpha n Z_n^2\right] &= 6\frac{Z_{n-1}}{Z_n - Z_{n-1}}, \end{aligned}$$

where $Z_0(\tau) = 1$ for all τ. This system can in turn be rearranged to get a system of first-order differential equations which describe the depth of the freezing front, Z_i

$$\dot{Z}_1 = \frac{1}{2Z_1 + 1}\left(\frac{6}{Z_1 - 1} - \frac{6Z_1}{Z_2 - Z_1}\right) \tag{12}$$

$$\dot{Z}_i = \frac{1}{2Z_i + Z_{i-1}}\left(\frac{6Z_{i-1}}{Z_i - Z_{i-1}} - \frac{6Z_i}{Z_{i+1} - Z_i} - (Z_i + 2Z_{i-1})\dot{Z}_i - 1\right), \quad i = 2, 3, \ldots, n-1 \tag{13}$$

$$\dot{Z}_n = \frac{1}{2(1 + 3\alpha n)Z_n + Z_{n-1}}\left(\frac{6Z_{n-1}}{Z_n - Z_{n-1}} - (Z_n + 2Z_{n-1})\dot{Z}_{n-1}\right). \tag{14}$$

This system of differential equations can be easily solved using Euler's method, which is how I implemented it in my own program. The Caldwell and Chiu paper mentions using Euler's method, but they only publish the results they found using a fourth-order Runge-Kutta method. I decided to first try using Euler's method and then compare my results to theirs to see how well my program matched up.

5 Small Time Approximation

In order to use Euler's Method (or even a Runge-Kutta method), it is necessary to have some starting approximation of Z_i. However, there is a singularity at $\tau = 0$, so we cannot start at exactly $Z(0)$ and must instead approximate Z at some time τ close to 0. I decided to follow the same approximations used by Caldwell and Chiu to begin started on my own implementation. For their small time approximation, they used the first four terms from a solution first given by Poots in 1962

$$Z_i(\tau) \approx 1 + \mu_{i,0}\tau^{1/2} + \mu_{i,1}\tau + \mu_{i,2}\tau^{3/2}. \tag{15}$$

However, Caldwell and Chiu advocate calculating each $\mu_{i,j}$ by their new method instead of the derivation published by Poots. In order to do this, we must plug the small time approximation (15) into the system of equations (12)-(14). From here we can simplify to obtain a system of n non-linear equations for each set of the coefficients $\mu_{i,0}$, $\mu_{i,1}$ and $\mu_{i,2}$. These resulting systems of equations follow in Appendix A.

Fortunately, we can exploit the nature of this system of non-linear equations to help us solve for the coefficients. It turns out that the Jacobian matrix J of each set of equations is tridiagonal

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial \mu_{1,j}} & \frac{\partial f_1}{\partial \mu_{2,j}} & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{\partial f_2}{\partial \mu_{1,j}} & \frac{\partial f_2}{\partial \mu_{2,j}} & \frac{\partial f_2}{\partial \mu_{3,j}} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{\partial f_3}{\partial \mu_{2,j}} & \frac{\partial f_3}{\partial \mu_{3,j}} & \ddots & 0 & \cdots & \cdots \\ \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots \\ \cdots & \cdots & 0 & \ddots & \frac{\partial f_{n-2}}{\partial \mu_{n-2,j}} & \frac{\partial f_{n-2}}{\partial \mu_{n-1,j}} & 0 \\ \cdots & \cdots & \cdots & 0 & \frac{\partial f_{n-1}}{\partial \mu_{n-2,j}} & \frac{\partial f_{n-1}}{\partial \mu_{n-1,j}} & \frac{\partial f_{n-1}}{\partial \mu_{n,j}} \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{\partial f_n}{\partial \mu_{n-1,j}} & \frac{\partial f_n}{\partial \mu_{n,j}} \end{pmatrix}. \tag{16}$$

Therefore, it makes sense to use Newton's method to solve our non-linear systems. We rewrite our system in terms of the linear system

$$J(\mu^{(k)})[\mu^{(k+1)} - \mu^{(k)}] = -f(\mu^{(k)}) \tag{17}$$

and repeatedly solve the system until the approximate solution is as accurate as desired; in other words, we iterate until $\mu^{(k+1)} - \mu^{(k)}$ is less than some ϵ. In order to use Newton's method, we need to actually find the Jacobian, which we can approximate using a finite forward-difference formula of order $\mathbf{O}(h)$. Furthermore, since the Jacobian is tridiagonal, we can solve (17) quickly using LU-factorization.

In my implementation, I used Matlab's built in function to factor J into the product of lower- and upper-triangle matrices L and U. The benefit of doing this is that (17) can then be quickly solved through forward and backward substitution. First using left division by L, we solve for $U\mu$

$$LU\mu = -f, \tag{18}$$
$$U\mu = y, \tag{19}$$
$$Ly = -f \tag{20}$$

which Matlab does by forward substitution since L is a lower-triangular matrix. Next we use left division to solve equation (19) for μ. Matlab uses backwards substitution since U is an upper-diagonal matrix.

Unfortunately, the systems of equations for each set of coefficients $\mu_{i,0}$, $\mu_{i,1}$ and $\mu_{i,2}$ are dependent on eachother. Therefore we actually need to solve for $\mu_{i,0}$ before solving for $\mu_{i,1}$ and we need to solve for $\mu_{i,2}$ before solving for $\mu_{i,1}$. Therefore, we have to implement Newton's method and LU-factorization three separate times in order to calculate all of the coefficients.

6 Results

The first part of the code that I ran was my calculation of the coefficients for the small time approximation. I used the same value for α as the Cauldwell and Chiu paper so that I could compare results. My coefficients agreed up to the eighth decimal place with those that Cauldwell and Chiu published. My coefficients for n = 1 through 8 division follow in Table 1.

Table 1: The coefficients I calculated for the small time equation, where n is the number of subdivisions.

i	$\mu_{i,0}$	$\mu_{i,1}$	$\mu_{i,2}$
		$(n = 1)$	
1	1.1547 0054	-0.3950 6173	0.3379 0962
		$(n = 2)$	
1	0.5671 1246	-0.2768 0377	0.2320 4798
2	1.1838 0996	-0.2866 4148	0.2180 5279
		$(n = 4)$	
1	0.2796 2749	-0.1592 4187	0.1357 0464
2	0.5648 3008	-0.2505 6275	0.1983 6550
3	0.8683 0502	-0.2831 9427	0.2154 3866
4	1.2088 0228	-0.2451 3437	0.1823 9656
		$(n = 8)$	
1	0.1386 1349	-0.0850 7646	0.0737 1180
2	0.2778 9601	-0.1524 3282	0.1264 3258
3	0.4192 2830	-0.2043 1228	0.1639 1618
4	0.5641 2973	-0.2418 3497	0.1892 2724
5	0.7143 8357	-0.2650 2322	0.2034 8651
6	0.8722 1745	-0.2725 8662	0.2061 4177
7	1.0405 9242	-0.2613 2254	0.1948 7497
8	1.2237 1166	-0.2247 2875	0.1651 9636

The second part of the code that I ran was for determining the location of the freezing front. Because I implemented Euler's method instead of a fourth-order Runge-Kutta I expected a few more discrepancies between my results and the results published by Cauldwell and Chiu. However, my implementation yielded surprisingly similar results. As can be seem in Table 2, I found that my values agreed with their results up until about the fifth significant digit.

Table 2: $Z(\tau)$ for $n = 8$, $\alpha = 1$, and a stepsize of 0.000025.

τ	Cauldwell and Chiu	Euler's Method (Our Implementation)	Error
0.010	1.1202 8907	1.1202 8907	0.0000 0000
0.015	1.1467 8814	1.1468 0076	0.0000 1262
0.020	1.1689 8837	1.1690 0700	0.0000 1863
0.025	1.1884 4630	1.1884 6827	0.0000 2197
0.030	1.2059 5868	1.2059 8267	0.0000 2399
0.035	1.2219 9850	1.2220 2376	0.0000 2526
0.040	1.2368 7378	1.2368 9987	0.0000 2609
0.045	1.2507 9833	1.2508 2497	0.0000 2664
0.050	1.2639 2779	1.2639 5477	0.0000 2698
0.055	1.2763 7954	1.2764 0674	0.0000 2720
0.060	1.2882 4476	1.2882 7208	0.0000 2732
0.065	1.2995 9591	1.2996 2329	0.0000 2738
0.070	1.3104 9172	1.3105 1910	0.0000 2738
0.075	1.3209 8055	1.3210 0791	0.0000 2736
0.080	1.3311 0282	1.3311 3011	0.0000 2729
0.085	1.3408 2680	1.3409 1989	0.0000 9309
0.090	1.3503 7933	1.3504 0644	0.0000 2711
0.095	1.3595 8793	1.3596 1493	0.0000 2700
0.100	1.3685 4038	1.3685 6726	0.0000 2688

7 Conclusion

Overall, I was satisfied with the way my model turned out. I began by reading about some of the most basic Stefan problems and the difficulties associated with them. It was really interesting to then go to a recent paper and read how Cauldwell and Chiu had handled the specific case of cylindrical freezing. I also found it really rewarding that I was able to engage with their paper on a deep level. Before I took this class, I would not have understood several of the finer points of the paper, and I doubt that I would have been able to implement my program. I also liked that this problem involved several different concepts we had learned in class since numerical methods were needed for both the overall solution and the calculation of the small time coefficients.

8 Appendix A: Small Time Coefficients

In order to find the system of equations, we must plug the small time approximation (15) into the system of equations (12)-(14). From here we can simplify to obtain a system of n non-linear equations for each set of the coefficients $\mu_{i,0}$, $\mu_{i,1}$ and $\mu_{i,2}$. These equations follow below. I've written them in the form $f_i = f(\mu_{i,j}) = 0$, (and $f(\mu_{i,j})$ is the function we take the Jacobian of in the small time approximation section).

In the case where $n = 1$

$$\begin{aligned}
f_0 &= \frac{2}{\sqrt{1+2\alpha}} - \mu_{1,0} \\
f_1 &= -\frac{8}{9}\frac{(1+3\alpha)}{(1+2\alpha)^2} - \mu_{1,1} \\
f_2 &= \frac{80}{81}\frac{(1+3\alpha)^2}{(1+2\alpha)^{7/2}} - \mu_{1,3}
\end{aligned}$$

When $n > 1$, there are 3 separate systems. The coefficient $\mu_{i,0}$ is defined by

$$f_0(i) = \begin{cases} \frac{4}{\mu_{1,0}} - \frac{4}{\mu_{2,0}-\mu_{1,0}} - \mu_{1,0}, & i = 1 \\ \frac{4}{\mu_{i,0}-\mu_{i-1,0}} - \frac{4}{\mu_{i+1,0}-\mu_{i,0}} - \mu_{i-1,0} - \mu_{i-1,0}, & i = 2, \ldots, n-1 \\ \frac{4}{\mu_{n,0}-\mu_{n-1,0}} - \mu_{n-1,0} - (1+2\alpha n)\mu_{n,0}, & i = n \end{cases}$$

the coefficient $\mu_{i,1}$ is defined by

$$\begin{aligned}
f_1(1) &= 6\left[\frac{\mu_{2,1}-\mu_{1,1}}{(\mu_{2,0}-\mu_{1,0})^2} - \frac{\mu_{1,0}}{\mu_{2,0}-\mu_{1,0}} - \frac{\mu_{1,1}}{\mu_{1,0}}\right] - \mu_{1,0}^2 - 3\mu_{1,1} \\
f_1(i) &= \left\{ \begin{array}{l} 12\left[\frac{\mu_{i-1,0}}{\mu_{i,0}-\mu_{i-1,0}} - \frac{\mu_{i,1}-\mu_{i-1,1}}{(\mu_{i,0}-\mu_{i-1,0})^2} - \frac{\mu_{i,0}}{\mu_{i+1,0}-\mu_{i,0}} + \frac{\mu_{i+1,1}-\mu_{i,1}}{(\mu_{i+1,0}-\mu_{i,0})^2}\right] - \ldots \\ \quad 6(\mu_{i,1}+\mu_{i-1,1}) - \mu_{i,0}(2\mu_{i,0} + \mu i - 1, 0) + \mu_{i-1}(\mu_{i,0} + 2\mu_{i-1,0}) \end{array} \right. \\
f_1(n) &= \left\{ \begin{array}{l} 12\left[\frac{\mu_{n-1,0}}{\mu_{n,0}-\mu_{n-1,0}} - \frac{\mu_{n,1}-\mu_{n-1,1}}{(\mu_{n,0}-\mu_{n-1,0})^2}\right] - (6+12\alpha n)\mu_{n,1} - 6\mu_{n-1,1} - \ldots \\ \mu_{n,0}[(2+6\alpha n)\mu_{n,0} + \mu_{n-1,0}] - \mu_{n-1,1}(\mu_{n,0} + 2\mu_{n-1,0}) \end{array} \right.
\end{aligned}$$

and the coefficient $\mu_{i,2}$ is defined by

$$f_2(1) \quad = \quad \begin{cases} 4\left[\frac{\mu_{2,2}-\mu_{1,2}}{(\mu_{2,0}-\mu_{1,0})^2} + \frac{\mu_{1,0}(\mu_{2,1}-\mu_{1,1})}{(\mu_{2,0}-\mu_{1,0})^2} - \frac{(\mu_{2,1}-\mu_{1,1})^2}{(\mu_{2,0}-\mu_{1,0})^3} - \frac{\mu_{1,1}}{\mu_{2,0}-\mu_{1,0}} + \frac{\mu_{1,1}^2}{\mu_[1,0}^3 - \frac{\mu_{1,2}}{\mu_{1,0}^2}\right] - \dots \\ 2\mu_{1,0}\mu_{1,1} - 3\mu_{1,2} \end{cases}$$

$$f_2(i) \quad = \quad \begin{cases} 12\left[\frac{\mu_{i-1,1}}{\mu_{i,0}-\mu_{i-1,0}} - \frac{\mu_{i-1,0}(\mu_{i,1}-\mu_{i-1,1})}{(\mu_{i,0}-\mu_{i-1,0})^2} + \frac{(\mu_{i,1}-\mu_{i-1,1})^2}{(\mu_{i,0}-\mu_{i-1,0})^3} - \frac{\mu_{i,2}-\mu_{i-1,2}}{(\mu_{i,0}-\mu_{i-1,0})^2} - \frac{\mu_{i,1}}{\mu_{i+1}-\mu_{i,0}} + \dots\right. \\ \frac{\mu_{i,0}(\mu_{i+1,1}-\mu_{i,1})}{(\mu_{i+1}-\mu_{i,0})^2} - \frac{(\mu_{i+1,1}-\mu_{i,1})^2}{(\mu_{i+1}-\mu_{i,0})^3} + \frac{\mu_{i+1,2}-\mu_{i,2}}{(\mu_{i+1}-\mu_{i,0})^2} - \dots \\ 9(\mu_{i,2}+\mu_{i-1,2}) - \mu_{i,0}(2\mu_{i,1}+\mu_{i-1,1}) - \mu_{i-1,1}(\mu_{i,1}+2\mu_{i-1,1}) - \dots \\ 2\mu_{i-1,1}(2\mu_{i,0}+\mu_{i-1,0}) - 2\mu_{i-1,1}(\mu_{i,0}+2\mu_{i-1,0}) \end{cases}$$

$$f_2(n) \quad = \quad \begin{cases} 12\left[\frac{\mu_{n-1,0}}{\mu_{n,0}-\mu_{n-1,0}} - \frac{\mu_{n-1,0}(\mu_{n,1}-\mu_{n-1,1})}{(\mu_{n,0}-\mu_{n-1,0})^2} + \frac{(\mu_{n,1}-\mu_{n-1,1})^2}{(\mu_{n,0}-\mu_{n-1,0})^3} - \frac{\mu_{n,2}-\mu_{n-1,2}}{(\mu_{n,0}-\mu_{n-1,0})^2}\right] - \dots \\ (9+18\alpha n)\mu_{n,2} - 9\mu_{n-1,2} - 2\mu_{n,0}[(2+6\alpha n)\mu_{n,0}+\mu_{n-1,0}] - 2\mu_{n-1,1}(\mu_{n,0}+2\mu_{n-1,0}) - \dots \\ \mu_{n,0}[(2+6\alpha n)\mu_{n,1}+\mu_{n-1,1}] - \mu_{n-1,0}(\mu_{n,1}+2\mu_{n-1,1}) \end{cases}$$

where $i = 2, \dots, n-1$

References

[C] Caldwell J, Chiu CK. "Numerical solution of one-phase Stefan problems by the heat balance integral method, Part I–cylindrical and spherical geometries." *Communications in Numerical Methods In Engineering* 2000; **16**:569-583.

[C] Caldwell J, Chiu CK. "Numerical solution of one-phase Stefan problems by the heat balance integral method, Part II–special small time starting procedure." *Communications in Numerical Methods In Engineering* 2000; **16**:585-593.

[H] Heath MT. *Scientific Computing: an Introductory Survey*, WCB/McGraw-Hill, 1997.

[H] Hill JM. *One-dimensional Stefan Problems: an Introduction*, Longman: London, 1987.