Strategies for handling different time-scales in CZ Silicon growth

P. Rábcke(1), A. Pursula(1) (annti.pursula@csc.fi), J. Ruokolainen(1) and O. Anttila(2)
(1) CSC-Scientific Computing Ltd., Finland, (2) Okmetic Oyj, Finland

Introduction
We present numerical methods for simulating the transient growth interface shape. Motivation:
• Time evolution of the interface is of interest in e.g. controlling the point defects [1]
• Time-averaged models (turbulence) in 2D are not always accurate enough [2]
The model tackles the following phenomena:
• Melt flow and convective heat transfer
• Required time scale less than 1 s
• Evolution of growth interface
• Required time scale \(\approx 1\) min
• Heater power control
• Required time scale 10s of mins
⇒ Multiscale model with respect to time is needed [3]
The basic idea is to use ideal algorithms for the whole range of timesteps. Methods are implemented into open source FEM software Elmer (www.csc.fi/elmer) [4]

Interface algorithms
The Stefan condition has to be satisfied on the phase change boundary:
\[
\rho L v \cdot n = k_s \frac{\partial T}{\partial n} - k_v \frac{\partial T}{\partial n},
\]
where \(\rho\) is the density of the solid, \(L\) is latent heat, \(n\) is the normal of the interface, \(T\) and \(T_i\) are the temperatures on the solid side and the liquid side of the interface, respectively, and \(k_s\) and \(k_v\) are the corresponding heat conductivities.

1) Steady state algorithm:
Set condition (1) with \(v \cdot n = V\), solve thermal environment and find the isotherm \(T = T_m\) (where \(V\) is the pull velocity)
• robust & converges in a few iterations
• suitable also for a transient simulation with time-averaged temperature (no latent heat)

2) Transient algorithm:
A derived form of Eq. (1) is used
\[
\rho L v \cdot n - \sqrt{\nu} = q_n,
\]
where \(q_n\) is the normal heat flux (rhs of Eq. (1)), \(\nu\) is the y component of the normal vector, \(\nu\) is a small diffusion parameter, and the velocity \(v\) is defined to be y-directional. Set \(T = T_m\) as BC, solve thermal environment, evaluate \(q_n\) from Eq. (2), and solve for \(v\). The displacement of the interface is then \(u = (V - \nu) dt\).
• includes latent heat ⇒ correct inertia for growth interface evolution
• heat flux computation sensitive to variations in temperature field (gradients)
• may require relaxation to converge
• similar approach often used, e.g. in [5]

3) Hybrid algorithm: Apply Robin boundary condition:
\[
q_n = L P v + g(T_m - T),
\]
where \(g = \frac{\partial T}{\partial n}\) and \(P\) is the fraction of the flux used in crystallisation.
\(T\) in Eq. (3) is treated implicitly in heat equation. After solving for temperature the flux balance is given by the penalty term in Eq. (3), and interface velocity is found from
\[
v = \frac{q}{\rho L}(T - T_m),
\]
• constant flux condition combined with a physically meaningful penalty for temperature change
• inertia of melting is accounted for
• Changing BC type by tuning parameter \(p\)

Conclusions

References: