Received April 1997
Accepted September 1997

Different interface approximations in multi-domain GDQ simulation of Czochralski bulk flows

C. Shu, Y.T. Chew and Y. Liu
Department of Mechanical & Production Engineering, National University of Singapore, Singapore

Introduction

Being a widely used commercial technique in producing large, single crystals, Czochralski method has attracted the attention of many researchers. In this method, the modeling and understanding of heat and mass transfer is an important issue in the optimization of such technique to grow more uniform and better quality crystals. There are many models for the numerical simulation of flows in Czochralski crystal growth. Among them, the bulk-flow model is popularly used. There are three major forces involved in this problem, that is, the buoyancy force due to melt motion, the centrifugal pumping due to rotation, and the shear stress at the free surface due to surface tension gradients. Many solvers have been developed to simulate the Czochralski bulk flows over the last two decades. These solvers are usually based on the low order methods such as finite element, finite difference and finite volume methods[1-8]. A comprehensive review for these solvers has been given by Brown *et al.*[9]. In general, low order methods require a large number of grid points to obtain accurate numerical solutions. Subsequently, they need a lot of computational effort and virtual storage. From the point of practical application, one is always interested in an efficient numerical method which can generate accurate numerical solutions using small computational resources.

It has been shown by Shu *et al.*[10-13] that the global method of generalized differential quadrature (GDQ) is a very efficient numerical method which can obtain accurate numerical results using a considerably small number of grid points and thus requiring much small computational resources. GDQ method approximates a spatial derivative with respect to a coordinate direction at a grid point by a linear weighted sum of all the functional values in that direction. The GDQ method was developed to improve the differential quadrature (DQ) method[14] in computing the weighting coefficients of derivative approximation. Through the analysis of a high order polynomial approximation and the analysis of a linear vector space, GDQ computes the weighting coefficients of the first order derivative by a simple algebraic formulation, and the weighting coefficients of the second and higher order derivatives by a recurrence relationship. So far, the GDQ method has been successfully applied

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 8 No. 4, 1998, pp. 424–444.

to solve some incompressible flow problems[10-13] and structural problems[15-17] with high efficiency.

To further validate the prominent advantages of GDQ method, it is interesting to apply this method to simulate the Czochralski bulk flows. The benchmark problem suggested by Wheeler is chosen as a test problem for the study. It is noted that for Wheeler's benchmark problem, the change of boundary conditions at the junction point between the crystal and the free surface of the melt introduces a singularity for vorticity. Since the singular point is not at the corner of the computational domain, it may have some effects on the solution of interior region. As will be shown in the paper, when the GDQ method is applied in the whole computational domain, there will be some oscillations in the vorticity distribution due to the effect of singularity. This phenomena has also been highlighted by Raspo *et al.*[18]. To remove the effect of singularity, the multi-domain technique is proposed. The idea of a multi-domain technique is to decompose the whole domain into several subdomains in such a way that the singular points become the corner points of some subdomains. When the GDQ method is applied in each subdomain, the boundary conditions at its four corner points have no contribution to the solution of interior region. Thus, the effect of singularity is completely removed. It should be indicated that the multi-domain approach can also be applied to solve complex problems. As mentioned earlier, the GDQ method is locally applied in each subdomain for the multi-domain GDQ approach. The information exchange between neighboring subdomains is conducted through the interface. Obviously, the interface treatment is a key procedure in the multi-domain GDQ approach. In this study, four interface approximations are presented, and then compared through their applications to Wheeler's[19] benchmark problem. After numerical discretization, the resultant algebraic equations are solved by SOR iterative method.

Generalized differential quadrature

GDQ formulation

GDQ method is based on the analysis of a high order polynomial approximation and the analysis of a linear vector space. GDQ first assumes that the solution of a partial differential equation is approximated by a high order polynomial. Then it is easy to prove that the (N-1)th order polynomial constitutes an N-dimensional linear vector space. In the N-dimensional linear polynomial vector space, there exists a set of base polynomials, and any polynomial in the space can be uniquely expressed by a linear combination of all the base polynomials. Thus, if all the base polynomials satisfy a linear formulation, so does any polynomial in the space even though the exact expression of this polynomial is unknown. This property guarantees that if all the base polynomials satisfy a linear equation, so does the solution of a partial differential equation which is approximated by a high order polynomial. Through this way, the weighting coefficients of any order derivative discretization can be determined by a proper choice of base polynomials in the vector

space. The weighting coefficients of the first order derivative are calculated by a simple algebraic formulation, and the weighting coefficients of the second and higher-order derivatives are given by a recurrence relationship. The details of GDQ method is shown in reference 10. In the following, the two-dimensional results are presented. For a smooth function $f(x,y)$, GDQ discretizes its nth order derivative with respect to x, and the mth order derivative with respect to y, at the point (x_i, y_j) as

$$f_x^{(n)}(x_i, y_j) = \sum_{k=1}^{N} w_{ik}^{(n)} \cdot f(x_k, y_j), \qquad for\ n = 1, 2, \cdots, N-1 \tag{1}$$

$$f_y^{(m)}(x_i, y_j) = \sum_{k=1}^{M} \overline{w}_{jk}^{(m)} \cdot f(x_i, y_k), \qquad for\ m = 1, 2, \cdots, M-1 \tag{2}$$

for $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, M$,

where N, M are the number of grid points in the x and y direction respectively, $w_{ik}^{(n)}$, $\overline{W}_{jk}^{(m)}$ are the weighting coefficients to be determined as follows:

Weighting coefficients for the first-order derivative

$$w_{ij}^{(1)} = \frac{A^{(1)}(x_i)}{(x_i - x_j) \cdot A^{(1)}(x_j)}, \qquad i, j = 1, 2, \ldots, N, \quad but\ j \neq i \tag{3}$$

$$\overline{w}_{ij}^{(1)} = \frac{B^{(1)}(y_i)}{(y_i - y_j) \cdot B^{(1)}(y_j)}, \qquad i, j = 1, 2, \ldots, M, \quad but\ j \neq i \tag{4}$$

where

$$A^{(1)}(x_i) = \prod_{j=1, j\neq i}^{N} (x_i - x_j),$$

$$B^{(1)}(y_i) = \prod_{j=1, j\neq i}^{M} (y_i - y_j),$$

Weighting coefficients for the second- and higher-order derivatives

$$w_{ij}^{(n)} = n \cdot \left(w_{ii}^{(n-1)} \cdot w_{ij}^{(1)} - \frac{w_{ij}^{(n-1)}}{x_i - x_j} \right), \tag{5}$$

for $i, j = 1, 2, \ldots, N$, but $j \neq i$, $n = 2, 3, \ldots, N-1$,

$$\overline{w}_{ij}^{(m)} = m \cdot \left(\overline{w}_{ii}^{(m-1)} \cdot \overline{w}_{ij}^{(1)} - \frac{\overline{w}_{ij}^{(m-1)}}{y_i - y_j} \right), \tag{6}$$

for $i, j = 1, 2, \ldots, M$, but $j \neq i$, $m = 2, 3, \ldots, M-1$.

When $j = i$, the weighting coefficients are given by

$$w_{ii}^{(n)} = -\sum_{j=1, j\neq i}^{N} w_{ij}^{(n)}, \qquad i=1, 2, \ldots, N, \quad n=1, 2, \ldots, N-1, \tag{7}$$

$$\overline{w}_{ii}^{(m)} = -\sum_{j=1, j\neq i}^{M} \overline{w}_{ij}^{(m)}, \qquad i=1, 2, \ldots, M, \quad m=1, 2, \ldots, M-1. \tag{8}$$

Multi-domain GDQ scheme

It is assumed that the physical domain of a problem can be represented by Ω and the boundary by Γ. The multi-domain technique, first, decomposes the domain Ω into several subdomains. In each subdomain, the local GDQ scheme is applied in the same fashion as application of the scheme in a single domain. The solutions for interior grid points are independent for each subdomain. Globally, the information exchange between subdomains is required. This can be done across the interface of subdomains. Since any complex geometry can be transformed into a rectangular domain or a combination of rectangular subdomains, a rectangular domain is chosen for demonstration without losing generality. Basically, there are two kinds of interface topology, patched and overlapped. In the following, four ways are presented to evaluate the functional value along the interface. Three ways are in the category of patched interface while the other is for the overlapped interface.

Patched interface

The topology of a patched interface is shown in Figure 1, where Γ_{ij} is the interface between two subdomains Ω_i and Ω_j. For the patched interface, the governing equation is not applied along the interface. Instead, the continuity condition is enforced. The common choice is to keep the function and its normal derivative continuous across the interface. In other words, the function is considered as C^1 continuity across the interface. Mathematically, this continuity condition can be written as

$$f(x_N^i) = f(x_1^j) \qquad \text{on } \Gamma_{ij}, \tag{9}$$

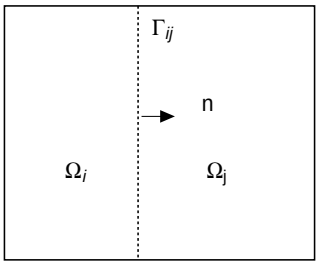

Figure 1.
Topology of a patched interface

$$f_n(x_N^i) = f_n(x_1^j) \qquad \text{on } \Gamma_{ij} \tag{10}$$

where $f(x_N^i)$, $f(x_1^j)$ represent the values of the function f at the interface of the i-subdomain and the j-subdomain, and $f_n(X_N^i)$, $f_n(X_1^j)$ represent the values of the first-order derivative of f with respect to n at the interface. Equation (9) can be easily implemented. However, the implementation of equation (10) involves the approximation of the first order derivative. Different approximations of the first order derivative would give different formulations to update the functional value along the interface. In the following, three methods are suggested to approximate the first order derivative and then generate the formulation to evaluate the functional value at the interface.

Method I

For this method, the GDQ scheme is applied to discretize the first order derivatives in equation (10). For the case selected for study, each subdomain is rectangular. Then the normal direction to the interface is parallel to one coordinate axis in the local coordinate system. For simplicity, this coordinate axis can be assumed as the x axis, and along this direction there are N grid points in the i subdomain and M grid points in the j subdomain. The weighting coefficients of the first-order derivative along the x direction are written as a_{mn}^i in the i subdomain and as a_{mn}^j in the j subdomain. Thus, using the GDQ approach, equation (10) can be approximated by

$$\sum_{k=1}^{N} a_{Nk}^i \cdot f(x_k^i) = \sum_{k=1}^{M} a_{1k}^j \cdot f(x_k^j), \tag{11}$$

Using equation (11), and setting $f(X_N^i) = f(X_1^j) = \bar{f}$, we obtain

$$\bar{f} = \frac{\sum_{k=1}^{N-1} a_{Nk}^i \cdot f(x_k^i) - \sum_{k=2}^{M} a_{1k}^j \cdot f(x_k^j),}{a_{11}^j - a_{NN}^i}, \tag{12}$$

where $\bar{f}$ is the value of the function f at the interface Γ_{ij}, which exchanges information between the two neighboring subdomains.

Method II

For this method, the first order finite difference scheme is used to discretize the first order derivatives in equation (10). It is supposed that the spacing between the interface point and its adjacent point in the subdomain Ω_i is Δx_1, and the spacing between the interface point and its adjacent point in the subdomain Ω_j is Δx_2, that is,

$$\Delta x_1 = x_N^i - x_{N-1}^i, \qquad \Delta x_2 = x_2^j - x_1^j$$

Applying the first order finite difference scheme to equation (10) gives,

$$\frac{\bar{f} - f(x_{N-1}^i)}{\Delta x_1} = \frac{f(x_2^j) - \bar{f}}{\Delta x_2} \tag{13}$$

which can be further reduced to

$$\bar{f} = \frac{\Delta x_2 \cdot f(x_{N-1}^i) + \Delta x_1 \cdot f(x_2^j)}{\Delta x_1 + \Delta x_2}, \tag{14}$$

Method III

If for simplicity, the difference between Δx_1 and Δx_2 in equation (13) is ignored, then equation (14) can be simplified as

$$\bar{f} = \frac{f(x_{N-1}^i) + f(x_2^j)}{2}, \tag{15}$$

Note that the above equation can also be derived from the simple averaging technique.

Overlapped interface. The topology of an overlapped interface is shown in Figure 2, where subdomain ABCD is overlapped with subdomain EFGH. It is noted that the right boundary of subdomain Ω_i, BC, is in the interior of subdomain Ω_j, and the left boundary of subdomain Ω_j, EH, is in the interior of subdomain Ω_i.

Method IV

For the overlapped interface, the governing equations are also applied along the interface. It is supposed that the solution in the interior of subdomains is known at an iteration level. Then the functional values along the lines of BC and EH are known. These values are then transferred into neighboring subdomains as new boundary conditions to get the solution in the interior of subdomains at the next iteration, i.e. the values along EH are transferred into subdomain Ω_j, and the values along BC are transferred into subdomain Ω_i. This process continues until converged solutions in all subdomains are obtained. For the overlapped

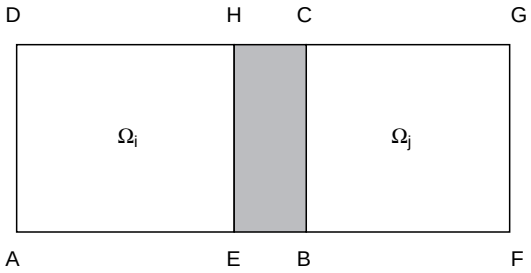

Figure 2.
Topology of an overlapped interface

topology, the functional values at the interface are given from the solution of governing equations.

Governing equations and boundary conditions

The non-dimensional governing equations can be written as[19]

$$(ru)_r / r + w_z = 0, \tag{16}$$

$$u_t + uu_r + wu_z - v^2 / r = -p_r + \Delta u - u / r^2, \tag{17}$$

$$v_t + uv_r + wv_z + uv / r = \Delta v - v / r^2, \tag{18}$$

$$w_t + uw_r + ww_z = -p_z + \Delta w + GrT, \tag{19}$$

$$T_t + uT_r + wT_z = \Delta T / \mathrm{Pr}. \tag{20}$$

where $\Delta = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{\partial^2}{\partial z^2}$.

The above equation system has been non-dimensionalised in the following manner,

$$(\bar{u}, \bar{v}, \bar{w}) = (\nu / R_C)(u, v, w), \quad (\bar{r}, \bar{z}) = R_C(r, z), \quad \bar{t} = (R_C^2 / \nu)t,$$

$$\bar{p} = -\rho_X R_C g z + (\rho_X \nu^2 / R_C^2)p, \tag{21}$$

$$\bar{T} = T_X + (T_C - T_X)T,$$

where (u,v,w) are radial, azimuthal and axial velocity components, (r,z) are polar coordinates, R is radius, t is time, p is pressure, T is temperature, υ is kinematic viscosity, g is gravitational acceleration, ρ is density of the melt, the over bar denotes a dimensional variable and a subscript denotes differentiation with respect to the subscript variable. However the subscripts x and c represent crystal and crucible respectively.

The boundary conditions are given by

$$\begin{array}{llll}
u = v = w_r = T_r = 0 & \text{for } r = 0, & 0 \le z \le \alpha, & \\
u = w = 0,\ v = Re_C,\ T = 1 & \text{for } r = 1, & 0 \le z \le \alpha, & \\
u = w = T_Z = 0,\ v = rRe_C & \text{for } 0 \le r \le 1, & z = 0, & (22) \\
u = w = T = 0,\ v = rRe_X & \text{for } 0 \le r \le \beta, & z = \alpha. & \\
u = v = w = 0,\ T = (r-\beta)/(1-\beta) & \text{for } \beta \le r \le 1, & z = \alpha. &
\end{array}$$

The nondimensional parameters that occur in equation (22) are the aspect ratios,

$$\alpha = H / R_C, \qquad \beta = R_X / R_C, \tag{23}$$

the Reynolds numbers,

$$\mathrm{Re}_X = R_C^2\Omega_X / \nu, \qquad \mathrm{Re}_C = R_C^2\Omega_C / \nu, \tag{24}$$

and the Prandtl and Grashof numbers,

$$\mathrm{Pr} = \nu / \kappa, \qquad Gr = g\overline{\beta}(T_C - T_X)R_C^3 / \nu^2, \tag{25}$$

where $\overline{\beta}$ is the coefficient of volumetric thermal expansion of the melt, R_X is the crystal radius, H is the height of melt, Ω_X and Ω_C are the angular velocities of crystal and crucible respectively, T_C is the temperature of the crucible wall, T_X is the crystal temperature. In this study, the aspect ratios and the Prandtl number are fixed at:

$\alpha = 1.0$, $\beta = 0.4$, Pr = 0.05.

To save computer resource and avoid calculating the pressure in the momentum equations, we transfer the momentum equations in the radial and axial directions into the vorticity form. It is noted that the common choice of the momentum equation for the azimuthal velocity component together with the continuity equation and the temperature equation may cause serious deviation and instability due to the presence of a strong inward component of radial flow induced by thermocapillarity. To overcome this difficulty and make computation more stable, Langlois[7] suggested using $\Omega = rv$ to replace the azimuthal component in the equation governing the azimuthal component, and using $S = \omega/r$ rather than ω as a dependent variable in the vorticity equation. This suggestion has been further validated by Kim *et al.*[8]. So the final equations can be written as

$$\frac{\partial\Omega}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(ru\Omega) + \frac{\partial}{\partial z}(w\Omega) = \frac{1}{r}\frac{\partial}{\partial r}\left[r^3\frac{\partial}{\partial r}\left(\frac{\Omega}{r^2}\right)\right] + \frac{\partial^2\Omega}{\partial z^2}, \tag{26}$$

$$\frac{\partial S}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(ruS) + \frac{\partial}{\partial z}(wS) - \frac{\partial}{\partial z}\left(\frac{\Omega^2}{r^4}\right) = -Gr\frac{1}{r}\frac{\partial T}{\partial r} + \frac{1}{r}\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}(r^2S)\right] + \frac{\partial^2 S}{\partial z^2}, \tag{27}$$

$$\frac{\partial T}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(ruT) + \frac{\partial}{\partial z}(wT) = \frac{1}{\mathrm{Pr}}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{\partial^2 T}{\partial z^2}\right), \tag{28}$$

$$\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi}{\partial r}\right) + \frac{1}{r}\frac{\partial^2\psi}{\partial z^2} = rS, \tag{29}$$

where the vorticity ω and Stokes stream function ψ are defined as

$$u = \frac{1}{r}\frac{\partial\psi}{\partial z}, \qquad w = -\frac{1}{r}\frac{\partial\psi}{\partial r}, \qquad \omega = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial r}, \tag{30}$$

The boundary conditions on four boundaries are then changed to

$$\psi = \Omega = \frac{\partial T}{\partial r} = 0, \quad S = -\frac{\partial^2 w}{\partial r^2} \tag{31}$$

on the symmetry lines,

$$\psi = T = 0, \quad S = \frac{1}{r}\frac{\partial u}{\partial z}, \quad \Omega = r^2 \, \mathrm{Re}_X, \tag{32}$$

on the growth interface,

$$\psi = S = \frac{\partial \Omega}{\partial z} = 0, \qquad \mathrm{T} = (\mathrm{r} - \beta)/(1 - \beta), \tag{33}$$

on the free surface,

$$\psi = \frac{\partial T}{\partial z} = 0, \quad S = \frac{1}{r}\frac{\partial u}{\partial z}, \quad \Omega = r^2 \, \mathrm{Re}_C, \tag{34}$$

on the crucible bottom,

$$\psi = \mathrm{S} = 0, \quad \Omega = \mathrm{r}^2 \, \mathrm{Re}_C \qquad \mathrm{T} = 1. \tag{35}$$

on the crucible wall.

Numerical algorithms

Numerical discretization

Application of GDQ method to discretize the spatial derivatives in equations (26-29) gives

$$\frac{d\Omega_{ij}}{dt} + \frac{1}{r_i}\sum_{k=1}^{N} AX_{ik}.r_k.u_{kj}.\Omega_{kj} + \sum_{k=1}^{M} AY_{jk} \cdot w_{ik}.\Omega_{ik} = \mathrm{r}_i^2.\sum_{k=1}^{N} BX_{ik}.\Omega_{kj} / r_k^2 + 3r_i.\sum_{k=1}^{N} AX_{ik}.\Omega_{kj} / r_k^2 + \sum_{k=1}^{M} \mathrm{BY}_{jk}.\Omega_{ik} \tag{36}$$

$$\frac{dS_{ij}}{dt} + \frac{1}{r_i}\sum_{k=1}^{N} AX_{ik}.r_k.u_{kj}.S_{kj} + \sum_{k=1}^{M} AY_{jk} \cdot w_{ik}.S_{ik} - \sum_{k=1}^{M} AY_{jk}.\frac{\Omega_{ik}^2}{r_i^4} = -\frac{Gr}{r_i}.\sum_{k=1}^{N} AX_{ik}.T_{kj} + \frac{1}{r_i^2}\sum_{k=1}^{N} BX_{ik}.r_k^2.S_{kj} - \frac{1}{r_i^3}\sum_{k=1}^{N} AX_{ik}.r_k^2.S_{kj} + \sum_{k=1}^{M} \mathrm{BY}_{jk}.S_{ik} \tag{37}$$

$$\frac{dT_{ij}}{dt} + \frac{1}{r_i}\sum_{k=1}^{N} AX_{ik}.r_k.u_{kj}.T_{kj} + \sum_{k=1}^{M} AY_{jk} \cdot w_{ik}.T_{ik} = \frac{1}{\mathrm{Pr}}.\left(\sum_{k=1}^{N} BX_{ik}.T_{kj} + \frac{1}{r_i}.\sum_{k=1}^{N} AX_{ik}.T_{kj} + \sum_{k=1}^{M} \mathrm{BY}_{jk}.\Omega_{ik} \right) \tag{38}$$

$$\sum_{k=1}^{N} BX_{ik}.\phi_{kj} - \frac{1}{r_i}\sum_{k=1}^{N} AX_{ik}.\phi_{kj} + \sum_{k=1}^{M} BY_{jk}.\phi_{ik} = r_i^2.S_{ij} \tag{39}$$

Where AX_{ij}, BX_{ij} are the weighting coefficients related to $\frac{\partial}{\partial r}$, and $\frac{\partial^2}{\partial r^2}$, AY_{ij}, BY_{ij} are the weighting coefficients related to $\frac{\partial}{\partial z}$ and $\frac{\partial^2}{\partial z^2}$, N, M are the number of grid points in the r and z direction respectively.

After applying the first order Euler implicit scheme to discretize the time derivative in equations (36-38), equations (36-39) can then be written as the following uniform expression

$$\sum_{k=1}^{N} P_{ik}.\phi_{kj} + \sum_{k=1}^{M} Q_{jk}.\phi_{ik} = B_{ij} \tag{40}$$

for $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots M$,

where ϕ_{ij} , P_{ik}, Q_{jk} and B_{ij} are taken as for equation (36) ,

$$\begin{aligned}
&\phi_{ij}=\Omega_{ij},\\
&P_{i,k} = \frac{r_j^2}{r_k^2}BX_{i,k} + 3.\frac{r_i}{r_k^2}AX_{i,k} - \frac{r_k}{r_i}AX_{i,k}.u_{k,j},\\
&Q_{j,k} = BY_{j,k} - AY_{j,k}.w_{i,k},\\
&B_{i,j} = 0,
\end{aligned} \tag{41}$$

for equation (37),

$$\begin{aligned}
&\phi_{ij}=S_{ij},\\
&P_{i,k} = \frac{r_k^2}{r_i^2}BX_{i,k} - \frac{r_k^2}{r_i^3}AX_{i,k} - \frac{r_k}{r_i}AX_{i,k}.u_{k,j},\\
&Q_{j,k} = BY_{j,k} - AY_{j,k}.w_{i,k},\\
&B_{i,j} = \frac{Gr}{r_i}.\sum_{k=1}^{M} AX_{i,k}.T_{k,j} - \sum_{k=1}^{M} AY_{j,k}.\frac{\Omega_{i,k}^2}{r_i^4},
\end{aligned} \tag{42}$$

for equation (38),

$\phi_{ij}=T_{ij}$

$$P_{i,k} = (BX_{i,k} + \frac{1}{r_i} AX_{i,k}) / Pr - \frac{r_k}{r_i} AX_{i,k} . u_{k,j} \quad ,$$

$$Q_{j,k} = BY_{j,k} / Pr - AY_{j,k} . w_{i,k} , \tag{43}$$

$$B_{i,j} = 0,$$

for equation (39),

$\phi_{ij}=\psi_{ij}$

$$P_{i,k} = BX_{i,k} - \frac{1}{r_i} AX_{i,k},$$

$$Q_{j,k} = BY_{j,k}, \tag{44}$$

$$B_{i,j} = r_j^2 . S_{i,j},$$

Solution of algebraic equations by SOR iterative method

The conventional SOR method for $A\phi = b$, where ϕ is a vector, can be expressed as

$$\phi_i^{n+1} = \frac{\alpha}{A_{i,i}} \left(b_i - \sum_{j=1}^{i-1} A_{i,j} . \phi_j^{n+1} - \sum_{j=i+1}^{N} A_{i,j} . \phi_j^n \right) + (1-\alpha) . \phi_i^n \tag{45}$$

The extended form for $P\phi + \phi Q = B$ can be obtained as

$$\phi_{i,j}^{n+1} = \frac{\alpha}{P_{i,i} + Q_{j,j}} \left(B_{i,j} - \left(\sum_{k=1}^{i-1} P_{i,k} . \phi_{k,j}^{n+1} + \sum_{k=i+1}^{N} P_{i,k} . \phi_{k,j}^n + \sum_{k=1}^{j-1} Q_{j,k} . \phi_{i,k}^{n+1} + \sum_{k=j+1}^{M} Q_{j,k} . \phi_{i,k}^n \right) \right) + (1-\alpha) . \phi_{i,j}^n \tag{46}$$

for $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots M$,

where n is the iteration number, ϕ = S, Ω, T, ψ respectively.

The convergence criteria are taken as

$$\sqrt{\frac{\sum_{i=2}^{N-1} \sum_{j=2}^{M-1} \left(B_{i,j} - \sum_{k=2}^{N-1} P_{i,k} . \phi_{k,j} - \sum_{k=2}^{M-1} Q_{j,k} . \phi_{i,k} \right)^2}{(N-2) \times (M-2)}} \le \delta , \tag{47}$$

for S, δ is chosen as 10^{-5}, while for Ω, T, ψ, δ is taken as 10^{-7}.

Results and discussion

To show the effect of singularity on the numerical solution in the single domain GDQ approach and validate the performance of four interface approximations in the multi-domain GDQ approach, three cases of Wheeler's problem are chosen for study. The geometry of three cases is the same as that shown in Figure 3. The conditions for these three cases are given as follows,

Case A:

$Gr = 0$, $Re_c = 0$, $Re_x = 100$

Case B:

$Gr = 0$, $Re_c = -25$, $Re_x = 100$

Case C:

$Gr = 10^5$, $Re_c = 0$, $Re_x = 0$

It is noted that for case A, the flow is generated by rotation of the crystal, while for case B, the flow is generated by opposite rotations of the crystal and the crucible. Case C is a natural convection problem.

Influence of singularity

For cases A , B, and C, the change of boundary conditions at the junction between the crystal and the melt free surface induces a singularity in the vorticity. Furthermore, the presence of the coordinate singularity at the axis $r = 0$ makes the problem more complicated. It was found that when GDQ method is applied in the whole domain, the vorticity distribution shows a high spurious oscillation. However, when the whole domain is decomposed into two subdomains and the multi-domain GDQ approach is applied, the spurious vorticity oscillation can be greatly damped. This can be observed in Figures 4, 5, and 6. Figures 4a and 4b compare the single domain vorticity distribution and the multi-domain vorticity distribution at z = 0.995 for Case A. The

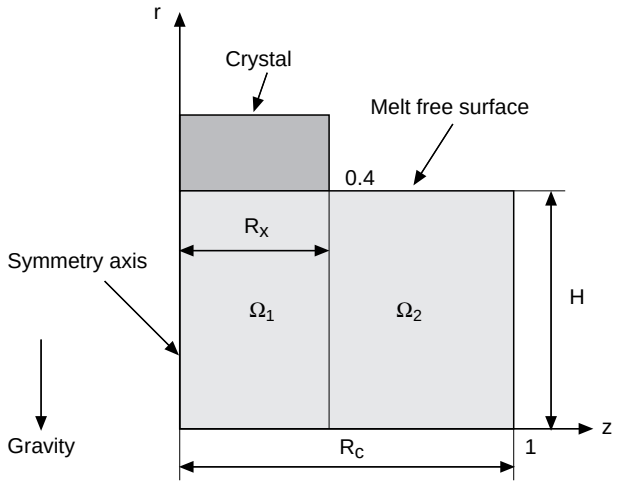

Figure 3. Configuration of Czochralski crystal growth

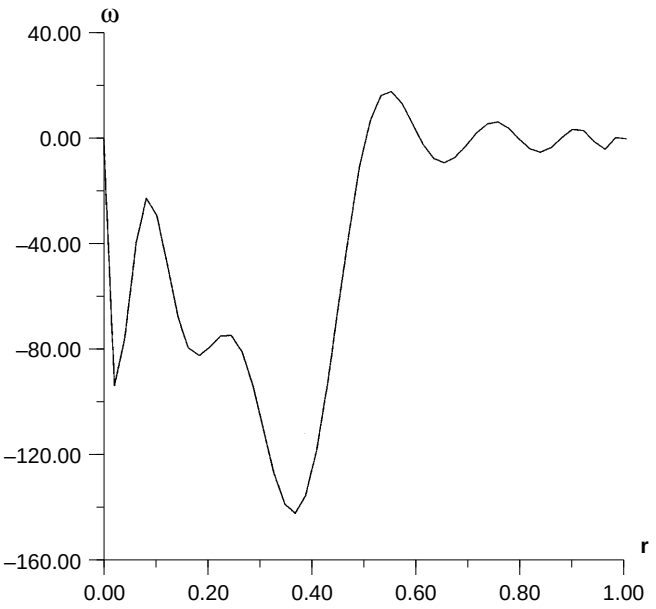

Figure 4a.
Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case A by single-domain method (mesh 15 × 15)

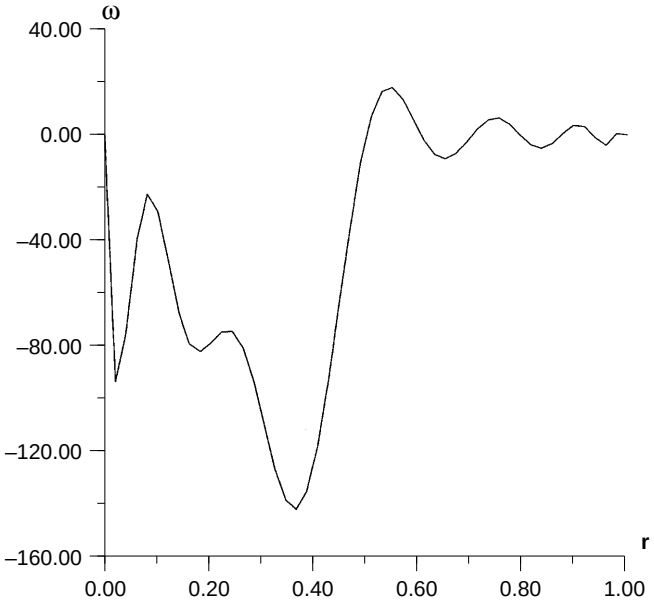

Figure 4b.
Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case A by multi-domain method (mesh 15 × 15 – 15 × 15)

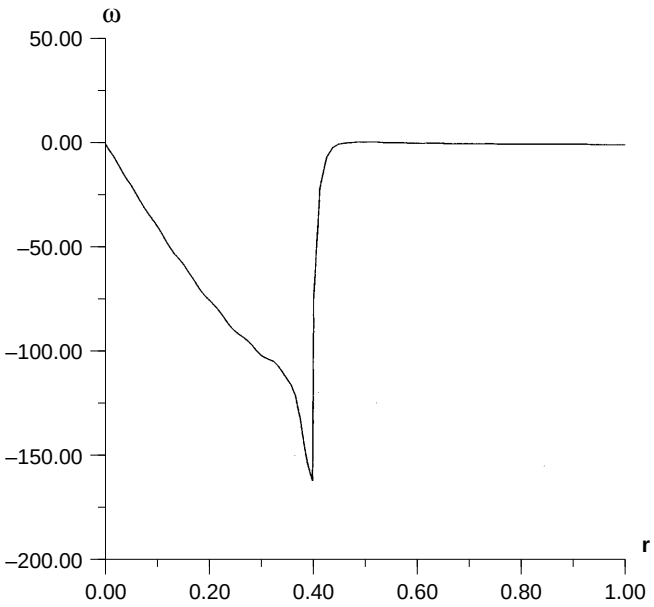

Figure 5a. Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case B by single-domain method (mesh 15×15)

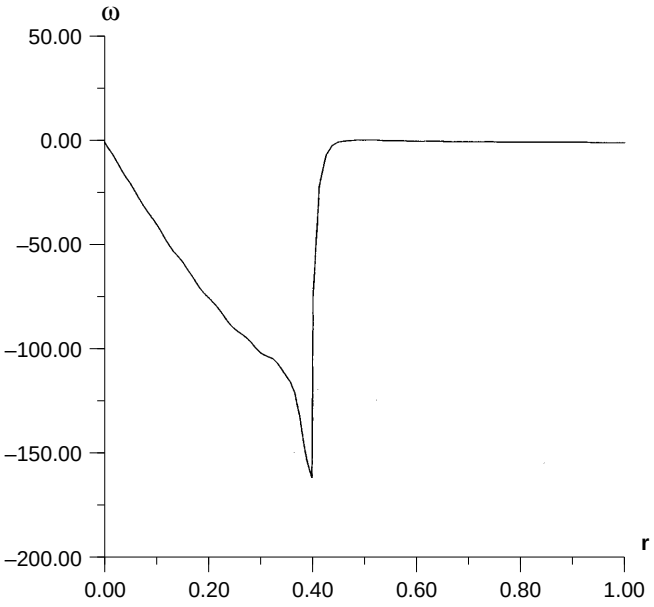

Figure 5b. Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case B by multi-domain method (mesh $15 \times 15 - 15 \times 15$)

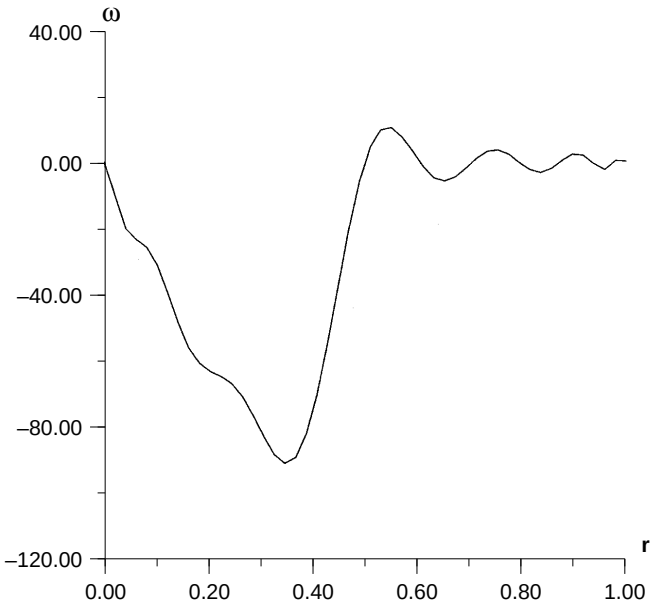

Figure 6a.
Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case C by single-domain method (mesh 15 × 15)

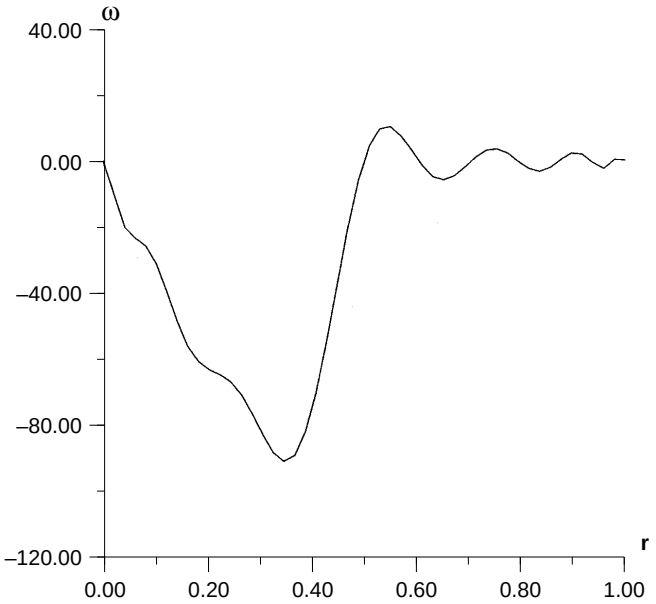

Figure 6b.
Vorticity profile below the crystal-melt free surface boundary (z = 0.995) for case C by multi-domain method (mesh 15 × 15 – 15 × 15)

respective comparison is shown in Figures 5a and 5b for Case B, and Figures 6a and 6b for Case C. It is noted that for all the cases considered, the Lagrange interpolation scheme is used to obtain the vorticity at z = 0.995. The single domain results are obtained by a mesh size of 15×15. For the multi-domain results, a mesh size of 15×15 is used in each subdomain and method IV is applied to update the functional value along the interface. It can be seen clearly from the figures that for the single domain case, the vorticity distribution exhibits great oscillations in particular near the axis and the junction point between the crystal and the melt free surface. However, for the multi-domain case, these oscillations are almost damped and the vorticity reveals a smooth distribution. The superiority of the multi-domain solution is clearly visible. By simply splitting the domain into two subdomains at the junction point of the crystal and melt free surface, the singularity is isolated.

Multi-domain results and discussion

Using the multi-domain technique, the whole computational domain is decomposed into two subdomains as shown in Figure 3, that is, Ω_1 contains the boundary of the crystal and Ω_2 contains the melt free surface. The computational process will be carried out in these two subdomains separately while the solutions are coupled through the interface by imposing different interface approximation methods. The discretized governing equations by GDQ method are solved by SOR iteration method. The minimum and maximum values of stream function computed by multi-domain GDQ method for cases A, B, and C are listed in Tables I, II, and III respectively. Also included in these Tables are the results of Buckle *et al.*[5] obtained by the second-order finite difference and finite volume method with a mesh of 65×65, and the results of

		Interface method	Ψ_{min}	Ψ_{max}
GDQ	13×13	Method II	-2.2494×10^{-1}	5.1872×10^{-6}
	13×13	Method III	-2.1110×10^{-1}	4.5781×10^{-6}
		Method IV	-2.1698×10^{-1}	6.1706×10^{-6}
GDQ	15×15	Method II	-2.3230×10^{-1}	4.9137×10^{-6}
	15×15	Method III	-2.2040×10^{-1}	4.7109×10^{-6}
		Method IV	-2.1954×10^{-1}	5.1192×10^{-6}
GDQ	17×17	Method II	-2.3903×10^{-1}	5.6694×10^{-6}
	17×17	Method III	-2.2880×10^{-1}	5.5219×10^{-6}
		Method IV	-2.2131×10^{-1}	5.0324×10^{-6}
GDQ	21×21	Method II	-2.5618×10^{-1}	5.9329×10^{-6}
	21×21	Method III	-2.4863×10^{-1}	5.8176×10^{-6}
		Method IV	-2.2176×10^{-1}	5.0794×10^{-6}
Buckle *et al.*[5]	65×65		-2.3447×10^{-1}	1.5642×10^{-6}
Xu *et al.*[6]	80×80		-2.0864×10^{-1}	3.1399×10^{-6}

Table I. Comparison of multi-domain GDQ scheme using different interface approximations for case A (Gr = 0; $Re_c = 0$, $Re_x = 100$)

		Interface method	Ψ_{min}	Ψ_{max}
GDQ	13×13	Method I	-6.4949×10^{-2}	1.2682×10^{-1}
	13×13	Method II	-4.0624×10^{-2}	1.2311×10^{-1}
		Method III	-1.9808×10^{-2}	1.5228×10^{-1}
		Method IV	-4.8835×10^{-2}	1.1157×10^{-1}
GDQ	15×15	Method I	-6.5521×10^{-2}	1.2765×10^{-1}
	15×15	Method II	-4.2016×10^{-2}	1.2405×10^{-1}
		Method III	-2.0564×10^{-2}	1.5490×10^{-1}
		Method IV	-4.8227×10^{-2}	1.1350×10^{-1}
GDQ	17×17	Method I	-6.6210×10^{-2}	1.2825×10^{-1}
	17×17	Method II	-4.3340×10^{-2}	1.2449×10^{-1}
		Method III	-2.1367×10^{-2}	1.5643×10^{-1}
		Method IV	-4.8039×10^{-2}	1.1473×10^{-1}
GDQ	21×21	Method I	-6.7663×10^{-2}	1.2904×10^{-1}
	21×21	Method II	-4.5319×10^{-2}	1.2487×10^{-1}
		Method III	-2.2594×10^{-2}	1.5811×10^{-1}
		Method IV	-4.7908×10^{-2}	1.1620×10^{-1}
Buckle *et al.*[5]				
	65×65		-5.0203×10^{-2}	1.1796×10^{-1}
Xu *et al.*[6]				
	80×80		-4.3090×10^{-2}	1.1851×10^{-1}

Table II.
Comparison of multi-domain GDQ scheme using different interface approximations for case B (Gr = 0; Re_c = 25, Re_x = 100)

		Interface method	Ψ_{min}	Ψ_{max}
GDQ	13×13	Method I	-3.3719×10^{-3}	2.8558×10^{1}
	13×13	Method II	-2.3009×10^{-3}	2.8416×10^{1}
		Method III	-9.5615×10^{-4}	2.9156×10^{1}
		Method IV	-5.5832×10^{-2}	2.8500×10^{1}
GDQ	15×15	Method I	-2.8772×10^{-4}	2.8617×10^{1}
	15×15	Method II	-3.2236×10^{-4}	2.8488×10^{1}
		Method III	-6.0718×10^{-4}	2.9152×10^{1}
		Method IV	-2.4643×10^{-2}	2.8492×10^{1}
GDQ	17×17	Method I	-1.9258×10^{-4}	2.8452×10^{1}
	17×17	Method II	-2.1634×10^{-4}	2.8428×10^{1}
		Method III	-4.0388×10^{-4}	2.9083×10^{1}
		Method IV	-1.1077×10^{-2}	2.8438×10^{1}
GDQ	21×21	Method I	-6.9638×10^{-5}	2.8376×10^{1}
	21×21	Method II	-7.9351×10^{-5}	2.8384×10^{1}
		Method III	-1.5547×10^{-4}	2.9055×10^{1}
		Method IV	-1.1760×10^{-5}	2.8416×10^{1}
Buckle *et al.*[5]				
	65×65		-1.1936×10^{-3}	2.8437×10^{1}
Xu *et al.*[6]				
	80×80		-1.1480×10^{-2}	2.8213×10^{1}

Table III.
Comparison of multi-domain GDQ scheme using different interface approximations for case C (Gr = 10^5; Re_c = 0, Re_x = 0)

Xu *et al.*[6] using the second-order upwind QUICK scheme with a mesh of 80×80. All the computational work in this paper were carried out on CRAY-J916 supercomputer using the 64-bit precision.

The multi-domain GDQ results are obtained by four interface approximations. It can be seen from the tables that the convergence trend of multi-domain GDQ solution is fairly obvious. When comparing the maximum absolute values of stream function, that is, the minimum value for case A and the maximum values for cases B and C, it was found that the difference between the results obtained from a mesh of 17×17 in each subdomain, and those from the mesh of 21×21 in each subdomain is less than 0.2 percent for case A, 1 percent for case B, and 0.1 percent for case C. However, if we compare the same results between the two low-order methods provided by Buckle *et al.*[5] and Xu *et al.*[6], it can be seen that the difference for case A is more than 10 percent, and for cases B and C, it is around 1 percent. It seems that, our results using the mesh size of 21×21 in each subdomain are more accurate. Hereafter, all results discussed are based on the mesh size of 21×21 in each subdomain. When comparing the minimum absolute values of the stream function, that is, the maximum value for case A and minimum values for cases B and C, we can find that the results given by multi-domain GDQ approach, finite difference method of Buckle and Schafer[5] and control volume method of Xu *et al.*[6] are not so consistent, and the difference is more evident in case C. This may be because the values are too small to be resolved by numerical methods irrespective of whether it is high-order or low-order. The difference may also be attributed to the process in obtaining the values of stream function. For the results of Buckle and Schafer[5] and Xu *et al.*[6], the values of stream function are obtained indirectly from the solution of primitive-variable equation system. They are given by the convergent solution of velocity field. For the multi-domain GDQ results, the values of stream function are directly resulted from the solution of vorticity-stream equation system. So, the present results may be more accurate.

When the results of the finite-difference method by Buckle and Schafer[5] using a mesh of 65×65 are taken as reference, it was found that the overlapped interface approximation (method IV) provides the most accurate results, while the results of the patched methods are less accurate. It is noted that in method IV, the functional values at the interface are given by the solution of governing equations without applying any artificial approximation. Moreover, within every iteration step, the numerical information of two subdomains is exchanged in a two-way direction in the overlapped area. This explains why the results obtained by overlapped interface approximation are more accurate than those by patched interface approximations which will be discussed later. It is worth noticing that even at a coarse mesh of 13×13 in method IV, the results are still reasonable. This further demonstrates the high accuracy and efficiency of the present multi-domain GDQ method.

Similarly, if we compare the results by three patched interface approximations (method I, II, and III) with those of Buckle and Schafer[5], we can find that

the results given by finite-difference first-order approximation (method II) are more accurate than those given by GDQ first-order approximation (method I), while the simple averaging method (method III) gives the least accurate results. This is because for patched interface, the governing equation is not applied along the interface. Instead, an enforced condition that the dependent variables and their first-order normal derivatives across the interface are continuous is applied. In other words, the function is considered as C^1 continuity across the interface. For the finite difference first-order approximation, it is exactly a linear approximation across the interface, and this is coincident with the enforced condition of interface assumption. Thus, it gives relatively more accurate results. The GDQ first-order approximation, as discussed in the earlier section, is a global method using high-order polynomial approximation. Therefore, it is overqualified when applied to a function with C^1 continuity across the interface, because the second- and higher-order terms may not cancel out fully and the remaining high-order error may accumulate and spread into the whole computational domain. This can explain why as a high-order method, GDQ first-order approximation is less accurate than the low-order method, i.e. finite difference first-order method. For method III using simple averaging, it requires an uniform mesh to obtain accurate results. However, in our test cases, the mesh used is not equally spaced and the influence of grid coordinates cannot be neglected. Thus this approximation is insufficient to give reasonable functional values at the interface; and if method III is applied to non-uniform meshes, it will give a result with huge deviation.

Figures 7-9 show the streamlines and isotherms of multi-domain GDQ results obtained by method IV using the mesh size of 21×21 in both subdomains for cases A, B, and C respectively. When the crystal rotates in case A, one vortices will be generated; while the crystal and the crucible rotate at opposite directions in case B, two vortices of different direction will be generated due to the rotation of the crystal and crucible. When both of the crystal and crucible are at rest in case C, the buoyancy force dominates the flow.

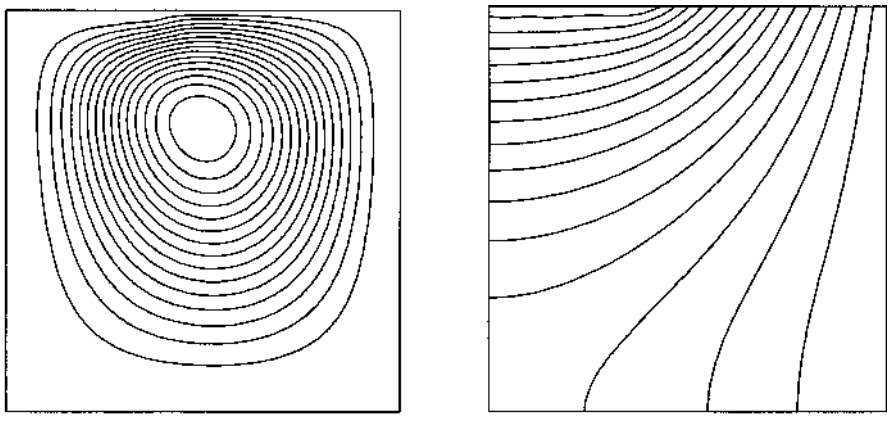

Figure 7.
Flow configuration for case A by multi-domain method (mesh $21 \times 21 - 21 \times 21$)

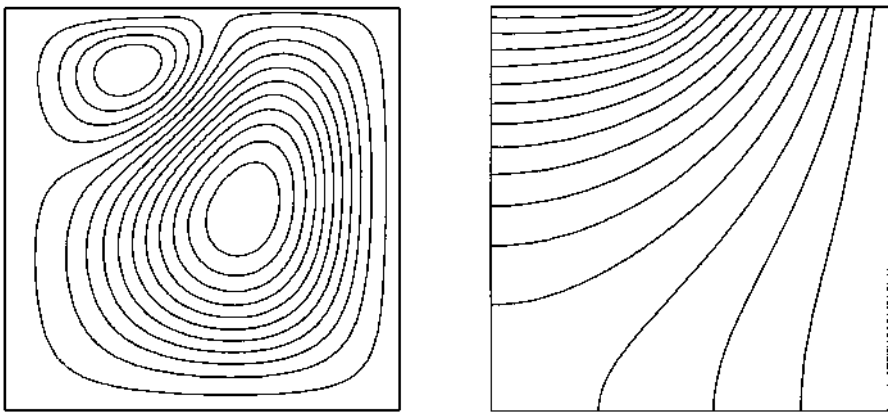

Figure 8.
Flow configuration for case B by multi-domain method (mesh $21 \times 21 - 21 \times 21$)

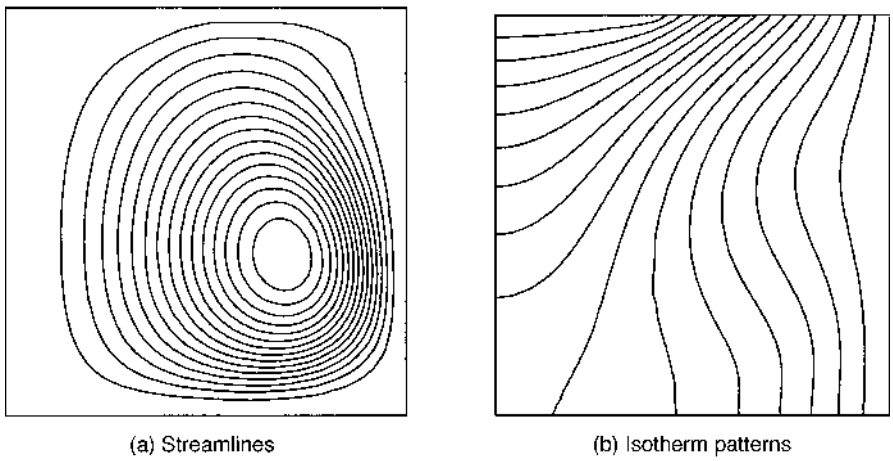

Figure 9.
Flow configuration for case C by multi-domain method (mesh $21 \times 21 - 21 \times 21$)

Conclusions

The multi-domain GDQ method can obtain accurate numerical results by using much smaller number of grid points and hence requiring very small computational resources when compared to low order methods. The overlapped interface approximation method provides the most accurate numerical solution among the four interface approximations presented in this paper. And the finite difference first-order approximation gives the best results among the three patched interface approximations. The simple averaging interface approximation method produces the least accurate results and can be treated as an inefficient method.

References

1. Crochet, M.J. and Wouters, P.J., "Finite-element simulation of Czochralski bulk flow", *J. Crystal Growth,* Vol. 65, 1983, pp. 153-65.
2. Xiao, Q. and Derby, J.J., "Bulk-flow versus thermal-capillary models for Czochralski growth of semiconductors", *J. Crystal Growth,* Vol. 129, 1993, pp. 593-609.
3. Kobayashi, N., "Computer simulation of heat, mass and fluid flows in a melt during Czochralski crystal growth", *Computer Methods in Applied Mechanics and Engineering.,* Vol. 23, 1980, pp. 21-33.
4. Kobayashi, N., "Computational simulation of the melt flow during Czochralski growth", *J. Crystal Growth,* Vol. 52, 1978, pp. 425-34.
5. Buckle, U. and Schafer, M., "Benchmark results for the numerical simulation of flow in Czochralski crystal growth", *J. Crystal Growth,* Vol. 126, 1993, pp. 682-94.
6. Xu, D., Shu, C. and Khoo, B.C., "Numerical simulation of flows in Czochralski crystal growth by second-order upwind QUICK scheme", *J. Crystal Growth,* in press, (1997).
7. Langlois, W.E., "Conservative difference procedures for rotationally symmetric flow with swirl", *Compu. Methods in Applied Mechanics and Engineering,* Vol. 25, 1981, pp. 315-33.
8. Kim, C.J. and Ro, S.T., "On the selection of prognostic equations for the rotating motion in simulating Czochralski flow", *Numerical Heat Transfer, Part B,* Vol. 28, 1995, pp. 385-99.
9. Brown, R.A., Kinney, T.A., Zhou, W. and Bornside, D.E., "Large scale simulation of the Czochralski growth of silicon crystals", *Proc. 10th Int. Heat Transfer Conf.,* Vol. 1, 1994, pp. 189-203.
10. Shu, C., "Generalized differential-integral quadrature and application to the simulation of incompressible viscous flows including parallel computation", PhD Thesis, University of Glasgow, UK, 1991.
11. Shu, C. and Richards, B.E., "Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations", *Int. J. Numer. Methods Fluids,* Vol. 15, 1992, pp. 791-8.
12. Shu, C., Chew, Y.T. and Richards, B.E., "Generalized differential-integral quadrature and their application to solve boundary layer equations", *Int. J. Numer. Methods Fluids,* Vol. 21, 1995, pp. 723-33.
13. Shu, C., Khoo, B.C. and Yeo, K.S., "Numerical solutions of incompressible Navier-Stokes equations by generalized differential quadrature", *Finite Element in Analysis and Design,* Vol. 18, 1994, pp. 83-97.
14. Bellman, R., Kashef, B.G. and Casti, J., "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations", *J. Comput. Phys.* Vol. 10, 1972.
15. Shu, C., "An efficient approach for free vibration analysis of conical shells", *Int. J. Mech. Sci.,* Vol. 38, 1996, pp. 935-49.
16. Shu, C., "Free vibration analysis of composite laminated conical shells by generalized differential quadrature", *J. Sound Vibr.,* Vol. 194, 1996, pp. 587-604.
17. Du, H., Lim, M.K. and Lin, R.M., "Application of generalized differential quadrature method to structural problems", *Inter. J. Numer. Methods Engng.,* Vol. 37, 1994, pp. 1881-96.
18. Raspo, I., Ouazzani, J. and Peyret, R., "A spectral multidomain technique for the computation of the Czochralski melt configuration", *Int. J. Num. Meth. Heat Fluid Flow,* Vol. 6, 1996, pp. 31-58.
19. Wheeler, A.A., "Four test problems for the numerical simulation of flow in Czochralski crystal growth", *J. Crystal Growth,* Vol. 102, 1990, pp. 691-5.