Development of a 2D/3D Computational Fluid Dynamic Code for Pore-Level Analysis of Cryocooler Regenerators

Ali Ghavami[1], Carl Kirkconnell[2] and S Mostafa Ghiaasiaan[1].

[1]Georgia Institute of Technology,
G.W. Woodruff School of Mechanical Engineering
Atlanta, GA, USA 30332
[2]West Coast Solutions, Huntington Beach, CA, USA 92647

ABSTRACT

Pore-level CFD analysis of periodic flow in representative porous structures is a useful technique for the estimation of solid-fluid momentum and thermal interaction parameters in cryocooler regenerators. Two and three-dimensional simulations are prohibitively computation intensive, however, and are impractical with commercial general-purpose CFD tools. A fast-running, flexible and efficient 2D/3D computational fluid dynamic (CFD) code is under development at Georgia Tech, specifically for the pore-level analysis of regenerators. Pore-level simulations can be carried out with this code for arbitrary and complex regenerator filler geometries, leading to about a four-time reduction of computation in comparison with commercial CFD codes. The computations are minimized by focusing only on critically important parameters. The theoretical and computational characteristics of this code are discussed, and typical code predictions representing friction and heat transfer are presented and compared with the predictions of the ANSYS Fluent CFD code.

INTRODUCTION

A regenerator is a key component of any Stirling and pulse tube cryocoolers which needs to provide high heat capacity, low friction factor in the main flow direction, and adequate heat transfer between the porous zone matrix and the working fluid. Powders composed of spheres, wire woven mesh, and cylindrical and square bundles of rod are common regenerator filler types. However, the 3D printing and additive manufacturing techniques are now capable of making more complicated forms of regenerators.

Many computational fluid dynamic (CFD) studies have been performed on different aspects of different regenerators in the past. Pathak [1] investigated the friction factor and heat transfer coefficient of porous media composed of rectangular cylinders under oscillating flow. Tao and Ghavami [2, 3] carried out pore level analyses addressing entropy generation in cylindrical rod and woven mesh regenerators under steady and oscillating flow. Ghavami [4] also investigated a novel revolute regenerator, made by 3D printing technology, in steady flow by defining and analyzing a single unit cell. Unit cell is the smallest portion of a complex regenerator which can be repeated in the flow direction to

Cryocoolers 21, edited by R.G. Ross, Jr., J.R. Raab and S.D. Miller

represent the entire regenerator. CFD analysis of cryocooler regenerators by the aid of commercial software including ANSYS Fluent [5] or COMSOL [6] often needs significant computation time, however. Other commercial tools like Sage [7] and REGEN [8] are 1-D software and cannot capture the 2/3D hydrodynamics and heat transfer aspects of regenerator.

This paper reports on the development of a 2D/3D C++ [9] code for the pore-level simulation of cryocooler regenerators of arbitrary geometric configuration. It is assumed that the fluid is laminar, non-Newtonian, and has constant properties. Different boundary conditions including wall, symmetry, and oscillating velocity can be handled for the solution of the momentum equations. Both Dirichlet and Neuman boundary conditions are implemented for the energy equation, however, at this time the energy equation is only solved for the fluid, and the capability for conjugate heat transfer analysis whereby conduction in the solid part of the regenerator is also modeled will be added in the future. The finite volume method is used for the discretization of the conservation equations. For numerical solution of the conservation equations, the projection method with staggered grids is applied, which ensures the stability of solution. Post processing and visualization of the solution data is performed by MATLAB [10] and Tecplot [11]. The results of the C++ code will be compared with the CFD simulation results of Pathak [1] which was performed using Fluent [5].

COMPUTATIONAL METHOD

Governing Equations

The continuity, momentum and energy conservation equations are non-dimensionalized by writing:

$$u_i' = \frac{u_i}{U},\ x_i' = \frac{x_i}{W},\ P_i' = \frac{W}{\rho\nu U}P,\ t' = \frac{\nu}{W^2}t,\ T' = \frac{T}{T_\infty}, \mathrm{Re} = \frac{\rho UW}{\mu}, \mathrm{Pr} = \frac{\mu c_p}{k} \tag{1}$$

where i is the Cartesian coordinate component index. After substituting these non-dimensional variables in conservation equations, performing some manipulation, and dropping primes for simplicity, we will have:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2}$$

$$\frac{\partial u_i}{\partial t} + \mathrm{Re}\frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{3}$$

$$\frac{\partial T}{\partial t} + \mathrm{Re}\left(\frac{\partial (uT)}{\partial x} + \frac{\partial (vT)}{\partial y} + \frac{\partial (wT)}{\partial z}\right) = \frac{1}{\mathrm{Pr}}\left(\frac{\partial^2 T}{\partial x \partial x} + \frac{\partial^2 T}{\partial y \partial y} + \frac{\partial^2 T}{\partial z \partial z}\right)$$
$$+ \frac{U^2}{c_p T_\infty}\left\{2\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right) + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2\right\} \tag{4}$$

The second term on the right-hand side of energy equation represents viscous dissipation, φ_d, and can be calculated from the velocity field solution prior to the solution of energy equation. Note that Einstein's rule for summation has been used in Equation 3.

Solution Methodology

Equations 2-4 are the final form of equations which need to be solved. Since it is assumed that the fluid properties are constant, Equations 2 and 3 are solved first, and once the velocity field is known, Equation 4 can be solved. Equations 2 and 3 are discretized and solved using explicit projection method [12, 13]. Implementation of projection method involves the following three steps:

Step 1: Use the explicit scheme to solve for an intermediate velocity, u_i^*. This intermediate velocity does not satisfy the continuity equation as the effect of pressure gradient is not included in this step:

$$\frac{\partial u_i}{\partial t} + \mathrm{Re}\frac{\partial u_i u_j}{\partial x_j} = \frac{\partial^2 u_i}{\partial x_j \partial x_j} \rightarrow \frac{u_i^* - u_i^n}{\Delta t} = -\mathrm{Re}\frac{\partial u_i u_j}{\partial x_j} + \frac{\partial^2 u_i}{\partial x_j \partial x_j} \rightarrow u_i^* = u_i^n + \Delta t\left(-\mathrm{Re}\frac{\partial u_i u_j}{\partial x_j} + \frac{\partial^2 u_i}{\partial x_j \partial x_j}\right) \tag{5}$$

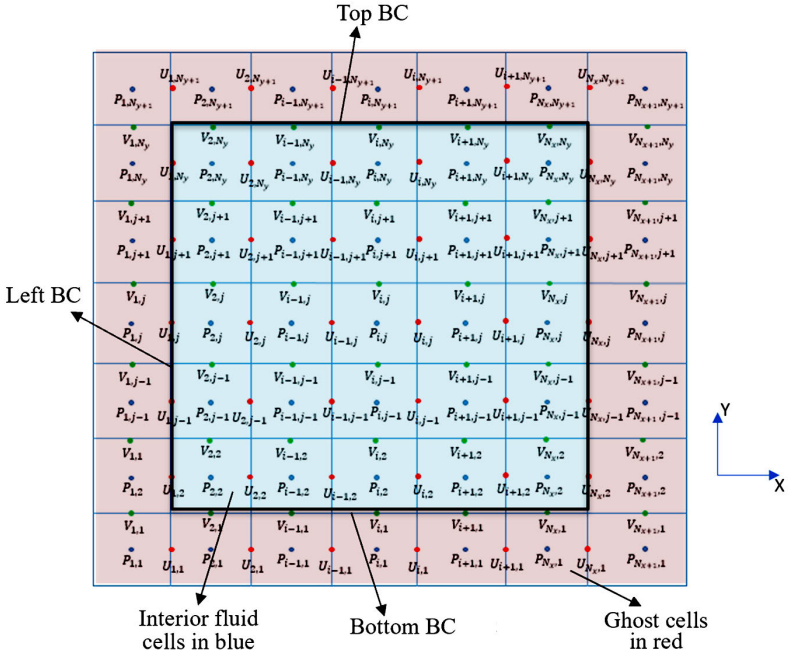

Figure 1. Staggered grids for interior nodes and outer boundary nodes

Step 2: Solve for the pressure by writing:

$$\frac{\partial u_i}{\partial t} = -\frac{\partial P}{\partial x_i} \quad \rightarrow \quad \frac{u_i^{n+1} - u_i^*}{\Delta t} = -\left(\frac{\partial P}{\partial x_i}\right)^{n+1} \tag{6}$$

Applying the divergence operation on the Equation 6 and utilizing the continuity equation, Poisson's equation for pressure is obtained and is case as:

$$\frac{1}{\Delta t}\nabla.\left(u_i^{n+1} - u_i^*\right) = -(\nabla^2 P)^{n+1} \rightarrow -\frac{1}{\Delta t}\nabla.\left(u^*\right) = -(\nabla^2 P)^{n+1} \rightarrow \left(\frac{\partial^2 P}{\partial x_i^2}\right)^{n+1} = \frac{1}{\Delta t}\left(\frac{\partial u_i^*}{\partial x_i}\right) \tag{7}$$

Step 3: Knowing P^{n+1} from step 2, calculate the corrected velocity, u^{n+1}, from:

$$u_i^{n+1} = u_i^* - \Delta t\left(\frac{\partial P}{\partial x_i}\right)^{n+1} \tag{8}$$

From Equation 8, the velocity field is calculated in the new time step and then we can solve for energy equation. Again, i and j subscription in Equations 5 through 8 corresponds to index notation.

Computational domain and equations discretization

For better understanding, discretization of both computational domain and equations are first described in 2D, and the final derivations in 3D come afterwards. Wherever the governing equations can be solved sequentially, it is advantageous to write the finite volume formulation suited for a so-called staggered grid rather than collocated grid. Staggered grid allows coupling of variables and consequently improves the stability. Figure 1 shows the staggered grid discretization for interior fluid computational domain excluding the interior solid part of regenerator. In staggered grid discretization, scalar and vector variables are assigned to different grid points called primary and secondary grids, respectively. The pressure is defined on the primary grid whereas the velocity components are defined on the cell faces of the secondary grid. More precisely, the x component of velocity, u, is assigned on the one-half grid line in the x direction at the same y location on the primary grid. The y component of the velocity, v, is defined in a similar fashion but in y direction. Temperature cells are exactly the same as pressure cells. These primary and secondary cells are shown in Figure 2.

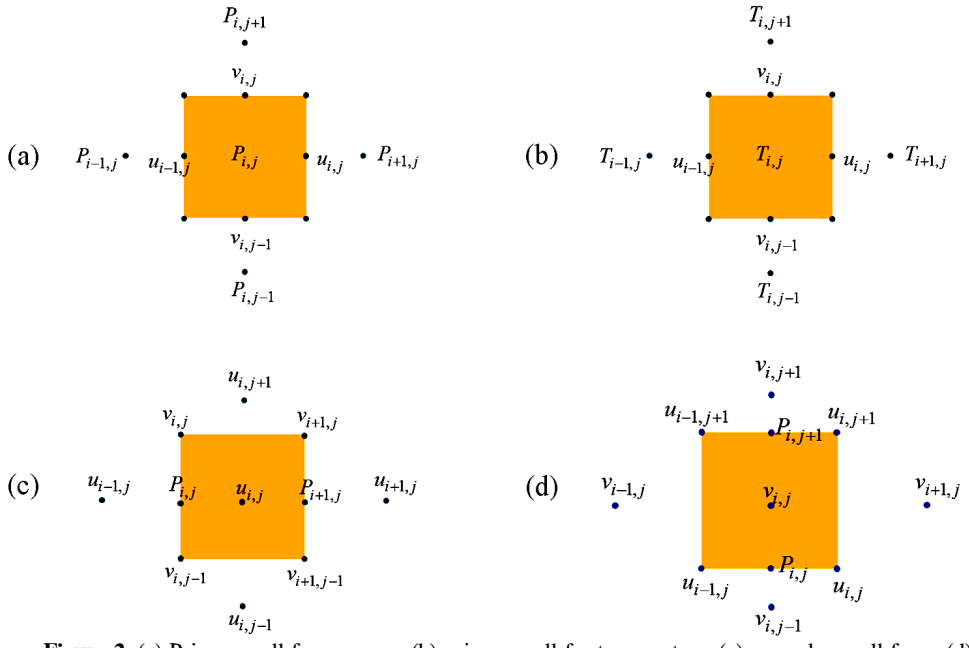

Figure 2. (a) Primary cell for pressure, (b) primary cell for temperature, (c) secondary cell for u, (d) secondary cell for v.

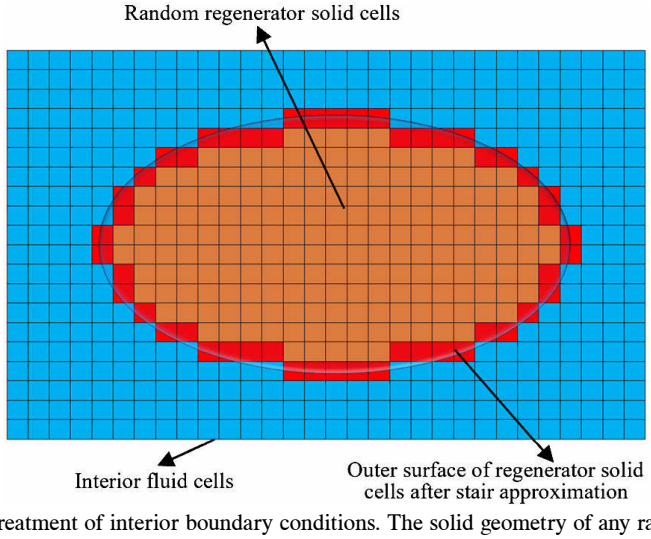

Figure 3. Treatment of interior boundary conditions. The solid geometry of any random regenerator is approximated by the boundary cells. The blue cell represents the fluid computational domain.

Staggered cells are applied, as shown in Figures 2 and 3. The cells representing vectors (velocity) are referred to as secondary cells. To derive the finite-volume equations all the terms in the conservation equations must be integrated over the volume of a relevant cell. The divergence theorem is then applied to convert volume to surface integrals wherever needed. Thus, considering the secondary cell for u, Figure 2.c, the finite-volume forms of the convective, viscous, and unsteady terms of Equation 5 give, respectively:

$$\oint u(\vec{v}\cdot d\vec{s}) = u^2_{i+\frac{1}{2},j}\Delta y - u^2_{i-\frac{1}{2},j}\Delta y + u_{i,j+\frac{1}{2}}v_{i+\frac{1}{2},j}\Delta x - u_{i,j-\frac{1}{2}}v_{i+\frac{1}{2},j-1}\Delta x = \frac{1}{4}(u_{i+1,j}+u_{i,j})^2\Delta y - \frac{1}{4}(u_{i-1,j}+u_{i,j})^2\Delta y + \frac{1}{4}(u_{i,j}+u_{i,j+1})(v_{i,j}+v_{i+1,j})\Delta x - \frac{1}{4}(u_{i,j}+u_{i,j-1})(v_{i,j-1}+v_{i+1,j-1})\Delta x \quad (9)$$

$$\oint \nabla u \cdot d\vec{s} = \left.\frac{\partial u}{\partial x}\right|_{i+\frac{1}{2},j} \Delta y - \left.\frac{\partial u}{\partial x}\right|_{i-\frac{1}{2},j} \Delta y + \left.\frac{\partial u}{\partial y}\right|_{i,j+\frac{1}{2}} \Delta x - \left.\frac{\partial u}{\partial y}\right|_{i,j-\frac{1}{2}} \Delta x \tag{10}$$

$$= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x} \Delta y + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y} \Delta x$$

$$\frac{u_i^* - u_i^n}{\Delta t} = -Re \frac{\partial u_i u_j}{\partial x_j} + \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{11}$$

Substituting Equations 9-11 into Equation 5 (where the terms in the latter equation have been integrated over the volume of the cell) leads to the following final form of equation of motion in x direction for step 1:

$$u_{i,j}^* = u_{i,j}^n + \Delta t \left(-Re \left(\frac{0.25(u_{i+1,j} + u_{i,j})^2 - 0.25(u_{i-1,j} + u_{i,j})^2}{\Delta x} \right) \right.$$
$$- Re \left(\frac{0.25(u_{i,j} + u_{i,j+1})(v_{i,j} + v_{i+1,j}) - 0.25(u_{i,j-1} + u_{i,j})(v_{i,j-1} + v_{i+1,j-1})}{\Delta y} \right)$$
$$\left. + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} \right) \tag{12}$$

Following the same procedure for the secondary cell that represents v, Figure 2.d, we get for velocity in y direction for step 1:

$$v_{i,j}^* = v_{i,j}^n + \Delta t \left(-Re \left(\frac{0.25(v_{i,j+1} + v_{i,j})^2 - 0.25(v_{i,j-1} + v_{i,j})^2}{\Delta y} \right) \right.$$
$$- Re \left(\frac{0.25(u_{i,j} + u_{i,j+1})(v_{i,j} + v_{i+1,j}) - 0.25(u_{i-1,j} + u_{i-1,j+1})(v_{i,j} + v_{i-1,j})}{\Delta x} \right)$$
$$\left. + \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{\Delta x^2} + \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{\Delta y^2} \right) \tag{13}$$

In accordance with step 2, the finite volume method is applied to the primary cell, Figure3.a, and thereby:

$$\iint \nabla \cdot \nabla P \, d\forall = \frac{1}{\Delta t} \iint \nabla \cdot \vec{V} \, d\forall \tag{14}$$

$$\oint \nabla P \, d\vec{s} = \frac{1}{\Delta t} \oint \vec{V} \, d\vec{s} \tag{15}$$

$$\left.\frac{\partial P}{\partial x}\right|_{i+\frac{1}{2},j} \Delta y - \left.\frac{\partial P}{\partial x}\right|_{i-\frac{1}{2},j} \Delta y + \left.\frac{\partial P}{\partial y}\right|_{i,j+\frac{1}{2}} \Delta x - \left.\frac{\partial P}{\partial y}\right|_{i,j-\frac{1}{2}} \Delta y = \frac{1}{\Delta t} \left(u_{i,j}^* \Delta y - u_{i-1,j}^* \Delta y + v_{i,j}^* \Delta x - v_{i,j-1}^* \Delta x \right) \tag{16}$$

$$\frac{P_{i-1,j} - 2P_{i,j} + P_{i+1,j}}{\Delta x^2} + \frac{P_{i,j-1} - 2P_{i,j} + P_{i,j+1}}{\Delta y^2} = \frac{1}{\Delta t} \left(\frac{u_{i,j}^* - u_{i-1,j}^*}{\Delta x} \right) + \frac{1}{\Delta t} \left(\frac{v_{i,j}^* - v_{i,j-1}^*}{\Delta y} \right) \tag{17}$$

Assuming $\Delta x = \Delta y = h$, and using Gauss-Seidel successive under relaxation (SOR) method, the final equation in step 2 is derived as:

$$P_{i,j}^{n+1} = C\alpha \left(\left(P_{i-1,j}^* + P_{i+1,j}^* \right) + \left(P_{i,j-1}^* + P_{i,j+1}^* \right) - \frac{h}{\Delta t} \left(u_{i,j}^* - u_{i-1,j}^* \right) - \frac{h}{\Delta t} \left(v_{i,j}^* - v_{i,j-1}^* \right) \right) + (1 - \alpha) P_{i,j}^n \tag{18}$$

where α=12 represents the underrelaxation factor and C=0.25.

Discretizing Equation 8, in accordance with step 3, results in:

$$u_{i,j}^{n+1} = u_{i,j}^* - \left(\frac{\Delta t}{\Delta x} \right) \left(P_{i+1,j}^{n+1} - P_{i,j}^{n+1} \right) \tag{19}$$

After solving the velocity-pressure domain, energy equation can be discretized by applying the finite volume formulation on the temperature cell, Figure 3.b:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta V + Re \oint \vec{V} T \cdot d\vec{s} = \frac{1}{Pr} \oint \nabla T \, d\vec{s} + \oint \varphi_{dissipation} \, d\vec{s} \tag{20}$$

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta V + Re \cdot \left(u_{i,j} T_{i+\frac{1}{2},j} \Delta y - u_{i-1,j} T_{i-\frac{1}{2},j} \Delta y + v_{i,j} T_{i,j+\frac{1}{2}} \Delta x - v_{i,j-1} T_{i,j-\frac{1}{2}} \Delta z \right)$$
$$= \frac{1}{Pr} \left(\left.\frac{\partial T}{\partial x}\right|_{i+\frac{1}{2},j} \Delta y - \left.\frac{\partial T}{\partial x}\right|_{i-\frac{1}{2},j} \Delta y + \left.\frac{\partial T}{\partial y}\right|_{i,j+\frac{1}{2}} \Delta x - \left.\frac{\partial T}{\partial y}\right|_{i,j-\frac{1}{2}} \Delta x \right) + \oint \varphi_{dissipation} \, d\vec{s} \tag{21}$$

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \Delta t\left(-\mathrm{Re}\left(\frac{u_{i,j}^{n}(T_{i+1,j}^{n} + T_{i,j}^{n}) - u_{i-1,j}^{n}(T_{i-1,j}^{n} + T_{i,j}^{n})}{2\Delta x} + \frac{v_{i,j}^{n}(T_{i,j+1}^{n} + T_{i,j}^{n}) - v_{i,j-1}^{n}(T_{i,j-1}^{n} + T_{i,j}^{n})}{2\Delta y}\right)\right.$$
$$+\frac{1}{\mathrm{Pr}}\left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2}\right)$$
$$+\frac{U^2}{c_p T_\infty}\left(2\left(\left(\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x}\right)^2 + \left(\frac{v_{i,j}^{n} - v_{i,j-1}^{n}}{\Delta y}\right)^2\right)\right.$$
$$\left.\left.+\left(\frac{u_{i,j+1}^{n} + u_{i-1,j+1}^{n} - u_{i,j-1}^{n} - u_{i-1,j-1}^{n}}{4\Delta y} + \frac{v_{i+1,j}^{n} + v_{i+1,j-1}^{n} - v_{i-1,j}^{n} - v_{i-1,j-1}^{n}}{4\Delta x}\right)^2\right)\right) \tag{22}$$

Equations 12, 13, 18, 19, and 22 should be solved sequentially in 2D to calculate velocity, pressure and temperature fields.

The derivation of finite-volume equations in 3D follows a similar process and result in the forthcoming equations:

$$u_{i,j,k}^{*} = u_{i,j,k}^{n} + \Delta t\left(-\mathrm{Re}\left(\frac{0.25(u_{i+1,j,k} + u_{i,j,k})^2 - 0.25(u_{i-1,j,k} + u_{i,j,k})^2}{\Delta x}\right)\right. \tag{23}$$
$$-\mathrm{Re}\left(\frac{0.25(u_{i,j,k} + u_{i,j+1,k})(v_{i,j,k} + v_{i+1,j,k}) - 0.25(u_{i,j-1,k} + u_{i,j,k})(v_{i,j-1,k} + v_{i+1,j-1,k})}{\Delta y}\right)$$
$$-\mathrm{Re}\left(\frac{0.25(u_{i,j,k} + u_{i,j,k+1})(w_{i,j,k} + w_{i+1,j,k}) - 0.25(u_{i,j,k-1} + u_{i,j,k})(w_{i,j,k-1} + w_{i+1,j,k-1})}{\Delta y}\right)$$
$$\left.+\frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{\Delta x^2} + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{\Delta y^2} + \frac{u_{i,j,k+1} - 2u_{i,j} + u_{i,j,k-1}}{\Delta z^2}\right)$$

$$v_{i,j,k}^{*} = v_{i,j,k}^{n} + \Delta t\left(-\mathrm{Re}\left(\frac{0.25(v_{i,j+1,k} + u_{i,j,k})^2 - 0.25(v_{i,j+1,k} + u_{i,j,k})^2}{\Delta x}\right)\right. \tag{24}$$
$$-\mathrm{Re}\left(\frac{0.25(u_{i,j,k} + u_{i,j+1,k})(v_{i,j,k} + v_{i+1,j,k}) - 0.25(u_{i-1,j,k} + u_{i-1,j+1,k})(v_{i,j,k} + v_{i-1,j,k})}{\Delta y}\right)$$
$$-\mathrm{Re}\left(\frac{0.25(v_{i,j,k} + v_{i,j,k+1})(w_{i,j,k} + w_{i,j+1,k}) - 0.25(v_{i,j,k-1} + v_{i,j,k})(w_{i,j,k-1} + w_{i,j+1,k-1})}{\Delta y}\right)$$
$$\left.+\frac{v_{i+1,j,k} - 2v_{i,j,k} + v_{i-1,j,k}}{\Delta x^2} + \frac{v_{i,j+1,k} - 2v_{i,j,k} + v_{i,j-1,k}}{\Delta y^2} + \frac{v_{i,j,k+1} - 2v_{i,j} + v_{i,j,k-1}}{\Delta z^2}\right)$$

$$w_{i,j,k}^{*} = w_{i,j,k}^{n} + \Delta t\left(-\mathrm{Re}\left(\frac{0.25(w_{i,j,k+1} + w_{i,j,k})^2 - 0.25(w_{i,j,k+1} + w_{i,j,k})^2}{\Delta x}\right)\right. \tag{25}$$
$$-\mathrm{Re}\left(\frac{0.25(w_{i,j,k} + w_{i,j+1,k})(v_{i,j,k} + v_{i,j,k+1}) - 0.25(v_{i,j-1,k+1} + v_{i,j-1,k})(w_{i,j,k} + w_{i,j-1,k})}{\Delta y}\right)$$
$$-\mathrm{Re}\left(\frac{0.25(u_{i,j,k} + u_{i,j,k+1})(w_{i+1,j,k} + w_{i,j,k}) - 0.25(u_{i-1,j,k} + u_{i-1,j,k+1})(w_{i-1,j,k} + w_{i,j,k})}{\Delta y}\right)$$
$$\left.+\frac{w_{i+1,j,k} - 2w_{i,j,k} + w_{i-1,j,k}}{\Delta x^2} + \frac{w_{i,j+1,k} - 2w_{i,j,k} + w_{i,j-1,k}}{\Delta y^2} + \frac{w_{i,j,k+1} - 2w_{i,j} + w_{i,j,k-1}}{\Delta z^2}\right)$$

$$P_{i,j,k} = C\alpha\left(\left(P_{i-1,j,k} + P_{i+1,j,k}\right) + \left(P_{i,j-1,k} + P_{i,j+1,k}\right) + \left(P_{i,j,k-1} + P_{i,j+1,k-1}\right) - \frac{h}{\Delta t}\left(u_{i,j,k}^{*} - u_{i-1,j,k}^{*}\right)\right. \tag{26}$$
$$\left.-\frac{h}{\Delta t}\left(v_{i,j,k}^{*} - v_{i,j-1,k}^{*}\right) + \frac{h}{\Delta t}\left(w_{i,j,k}^{*} - w_{i,j,k-1}^{*}\right)\right) + (1-\alpha)P_{i,j,k}$$

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{*} - \left(\frac{\Delta t}{\Delta x}\right)\left(P_{i+1,j,k}^{n+1} - P_{i,j,k}^{n+1}\right) \tag{27}$$

$$v_{i,j,k}^{n+1} = v_{i,j,k}^{*} - \left(\frac{\Delta t}{\Delta y}\right)\left(P_{i,j+1,k}^{n+1} - P_{i,j,k}^{n+1}\right) \tag{28}$$

$$w_{i,j,k}^{n+1} = w_{i,j,k}^{*} - \left(\frac{\Delta t}{\Delta z}\right)\left(P_{i,j,k+1}^{n+1} - P_{i,j,k}^{n+1}\right) \tag{29}$$

$$T_{i,j,k}^{n+1} = T_{i,j,k}^{n} + \Delta t\Bigg(-\mathrm{Re}\left(\frac{u_{i+1,j,k}^{n}T_{i+1,j,k}^{n} - u_{i-1,j,k}^{n}T_{i-1,j,k}^{n}}{2\Delta x} + \frac{v_{i,j+1,k}^{n}T_{i,j+1,k}^{n} - v_{i,j-1,k}^{n}T_{i,j-1,k}^{n}}{2\Delta y} + \frac{w_{i,j,k+1}^{n}T_{i,j,k+1}^{n} - w_{i,j,k-1}^{n}T_{i,j,k-1}^{n}}{2\Delta z}\right)$$
$$+\frac{1}{\mathrm{Pr}}\left(\frac{T_{i+1,j,k}^{n} - 2T_{i,j,k}^{n} + T_{i-1,j,k}^{n}}{\Delta x^2} + \frac{T_{i,j+1,k}^{n} - 2T_{i,j,k}^{n} + T_{i,j-1,k}^{n}}{\Delta y^2} + \frac{T_{i,j,k+1}^{n} - 2T_{i,j,k}^{n} + T_{i,j,k-1}^{n}}{\Delta z^2}\right)$$
$$+\frac{U^2}{c_p T_\infty}\left(2\left(\left(\frac{u_{i,j,k}^{n} - u_{i-1,j,k}^{n}}{\Delta x}\right)^2 + \left(\frac{v_{i,j,k}^{n} - v_{i,j-1,k}^{n}}{\Delta y}\right)^2 + \left(\frac{w_{i,j,k}^{n} - w_{i,1,k-1}^{n}}{\Delta y}\right)^2\right)\right.$$
$$+\left(\frac{u_{i,j+1,k}^{n} + u_{i-1,j+1,k}^{n} - u_{i,j-1,k}^{n} - u_{i-1,j-1,k}^{n}}{4\Delta y} + \frac{v_{i+1,j,k}^{n} + v_{i+1,j-1,k}^{n} - v_{i-1,j,k}^{n} - v_{i-1,j-1,k}^{n}}{4\Delta x}\right)^2$$
$$+\left(\frac{u_{i,j,k+1}^{n} + u_{i-1,j,k+1}^{n} - u_{i,j,k-1}^{n} - u_{i-1,j,k-1}^{n}}{4\Delta z} + \frac{w_{i+1,j,k}^{n} + w_{i+1,j,k-1}^{n} - w_{i-1,j,k}^{n} - w_{i-1,j,k-1}^{n}}{4\Delta x}\right)^2$$
$$\left.+\left(\frac{v_{i,j,k+1}^{n} + v_{i,j-1,k+1}^{n} - v_{i,j,k-1}^{n} - v_{i,j-1,k-1}^{n}}{4\Delta z} + \frac{w_{i,j+1,k}^{n} + w_{i,j+1,k-1}^{n} - w_{i,j-1,k}^{n} - w_{i,j-1,k-1}^{n}}{4\Delta y}\right)^2\right)\Bigg) \tag{30}$$

The value of parameter C in Equations 26 is equal to 1/6 for the interior nodes in 3D. Since the algorithm is explicit, for numerical stability the time step is limited to [13]:

$$\Delta t \leq 0.25\mathrm{Re}(\min(\Delta x_i, \Delta y_i))^2 \tag{31}$$

Boundary conditions

For ease of discussion, consider 2D flow in rectangular domain, as shown in Fig. 1 where the computational domain has top, right, bottom, and left outer boundaries which are shown with heavy black lines in the figure. As noted in the figure, ghost cells (also depicted in Figure 2) are added all around the outer boundaries of the computational domain. These ghost cells add extra nodes to the computational domain, but facilitate the implementation of the boundary conditions.

The type of these boundaries varies based on the type of modeling. The regenerator can be modeled as a whole, or only a section of the regenerator can be addressed by using the unit cell concept. Top and bottom boundaries can thus be set to "wall" or "symmetry", while the left and right boundaries can be set to oscillating inlet/outlet velocities. These are formulated as:

$$\text{Top and bottom BCs are wall: } v_{i,1} = v_{i,N_y+1} = 0 \ , \ u_{i,1} = -u_{i,2} \ , \ u_{i,N_y+2} = -u_{i,N_y+1} \tag{32}$$

$$\text{Top and bottom BCs are symmetry: } v_{i,1} = v_{i,N_y+1} = 0 \ , \ u_{i,1} = u_{i,2} \ , \ u_{i,N_y+2} = u_{i,N_y+1} \tag{33}$$

$$\text{Left BC is oscillating inlet velocity: } u_{1,j} = \sin(2\pi ft + \emptyset_1), \ v_{1,j} = -v_{2,j} \tag{34}$$

$$\text{Right BC is oscillating outlet velocity: } u_{Nx+1,j} = \sin(2\pi ft + \emptyset_2), \ v_{N_x+2,j} = -v_{N_x+1,j} \tag{35}$$

Note that oscillating velocity in Equations 34 and 35 is nondimensionalized by U which is the amplitude of inlet velocity. No slip assumption is used for wall BC.

For the pressure boundary condition, in accordance with the continuity equation, Equations 18 and 26 can be used for boundary nodes as well. However, the coefficient C is different for different nodes at the boundary. The values of C are summarized in Table 1.

For the temperature, both Dirichlet and Neuman boundary conditions are embedded in the code as options. Dirichlet BC specifies the value of temperature at the boundary while Neuman BC specifies the normal derivative of temperature at the boundary. Adiabatic or zero heat flux is a special case of Neuman BC, which can be used wherever symmetry BC is chosen or at the wall of regenerator.

We now consider the treatment of inner boundaries, which apply to the interface between the fluid and interior regenerator solid bodies. For these interior boundaries we make a first order

Table 1. Value of C in Equations 23 and 31.

	Interior nodes	Nodes at the face boundary	Nodes at the edge boundary	Nodes at the corner boundary
2D	1/4	NA	1/3	1/2
3D	1/6	1/5	1/4	1/3

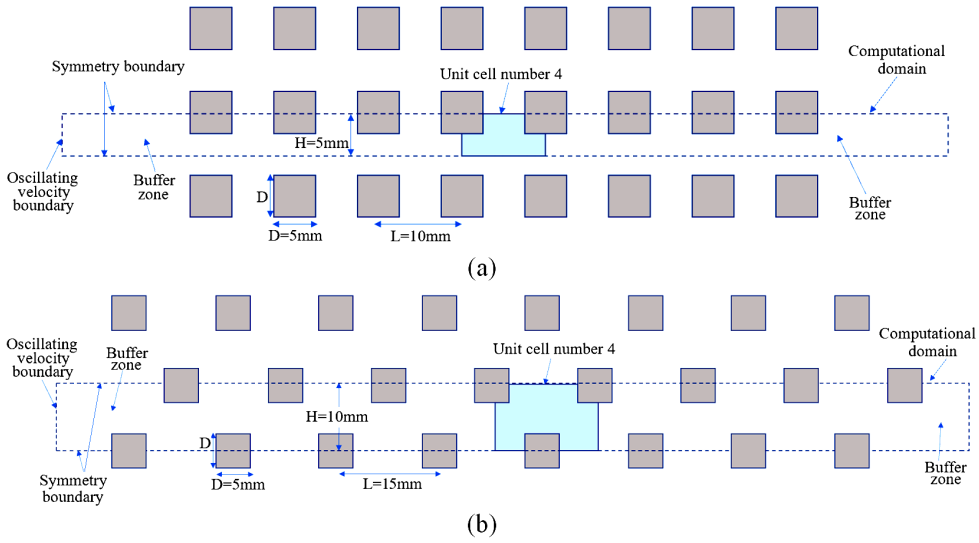

Figure 4. Computational domain of test cases, (a) collocated square bundle, (b) staggered square bundle.

approximation to the interior solid body geometry and use staircase staggered grids. A 2D example of the implementation of the boundary nodes for interior cells was shown earlier in Figure 3.

RESULTS AND DISCUSSION

Test Cases

Figure 4 shows the computational domain that is used as a test case for 2D simulations. Collocated and staggered square bundles are chosen as two different regenerator filler configurations. The computational domain consists of seven unit cells configured in a row. All dimensions and working condition are borrowed from the work of Pathak [1] whose simulations were performed using Fluent [5]. All code inputs are the same as those used by Pathak [1] to make the comparison between the Fluent and developed code possible.

The working fluid is air with constant properties at room temperature and atmospheric pressure. Reynolds numbers at the inlet are set to 280, 560, and 980. The range of frequency is 20-80 Hz. The regenerator is assumed isothermal with surface temperature of $T_s = 300$ K. The temperature of inlet and outlet oscillating flow is set to 200 K. The buffer zone is added at both ends to ensure fully developed flow into the computational domain. The computational domain is discretized to 10000×1000 elements. The dimensionless time step is equal to 10^{-3} which satisfies the condition in Equation 31. In grid-independence tests, using finer mesh changed the results only by less than 0.5% which confirms the grid independence for the forthcoming simulation results.

Cycle average Darcy friction facture and Nusselt number are calculate for the middle unit cell (unit cell number 4) by the following equations, respectively:

$$C_D = \frac{\frac{\Delta P}{\Delta x} D}{\frac{1}{2}\rho U^2}, \qquad \langle C_D \rangle = \frac{1}{2T}\int_0^{2T} C_D dt \tag{36}$$

$$Nu = \frac{DQ}{kA(T_s - T_{inlet})}, \qquad \langle Nu \rangle = \frac{1}{2T}\int_0^{2T} Nu dt \tag{37}$$

Results

Figure 5 shows the velocity field for unit cell number 4 (in the black box) and its surrounding unit cells (cells number 3 and 5) for the staggered square bundle configuration. The cells beyond the

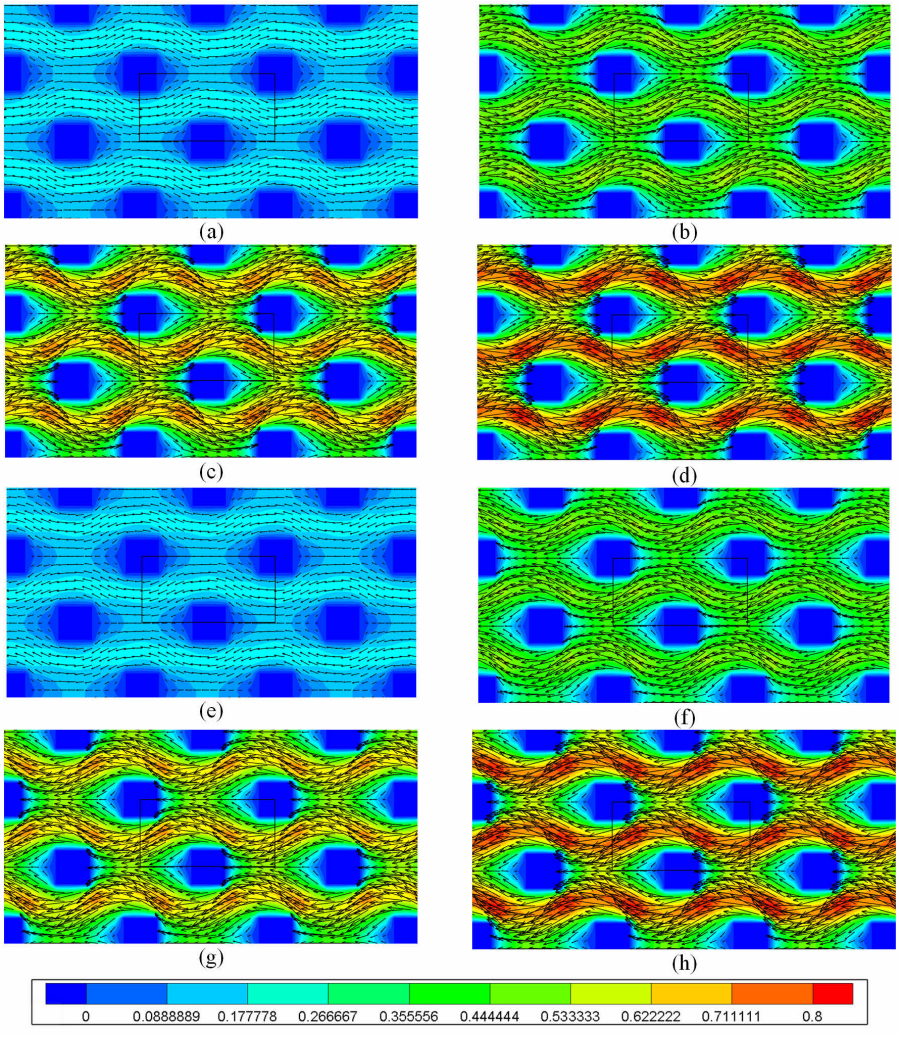

Figure 5. Velocity contours and velocity vectors at different cycle times: (a) 1/16 cycle, (b) 2/16 cycle, (c) 3/16 cycle, (d) 4/16 cycle, (e) 9/16 cycle, (f) 10/16 cycle, (g) 11/16 cycle, (h) 12/16 cycle.

symmetry boundary conditions and velocity vectors are also visualized to prove the correct implementation of symmetry boundary conditions. Figure 6 shows temperature contours for all the seven unit cells. The surface temperature of regenerator is set to 300 K and the inlet-outlet oscillating flow have a constant temperature of 200 K.

To remove the entrance effect on the solution data, the cycle average friction factor and Nu over the middle unit cell (unit cell number 4) is calculated. Figure 7a and 7b summarize the cycle average friction factor and Nu number results, respectively, for the collocated square bundle. The 2D/3D code results are compared to the results reported by Pathak [1]. The 3D results are only presented for f = 20 Hz.

As can be seen in Figure 7a, the friction factor in the collocated square bundles is decreased as Re increased as expected. It also can be seen that the friction factor is more sensitive to frequency at lower Reynolds numbers. Furthermore, the difference between the C++ code and Fluent results is less than 10%. This difference may at least partly come from the difference in the number of grids and the assumption of constant properties for the working fluid.

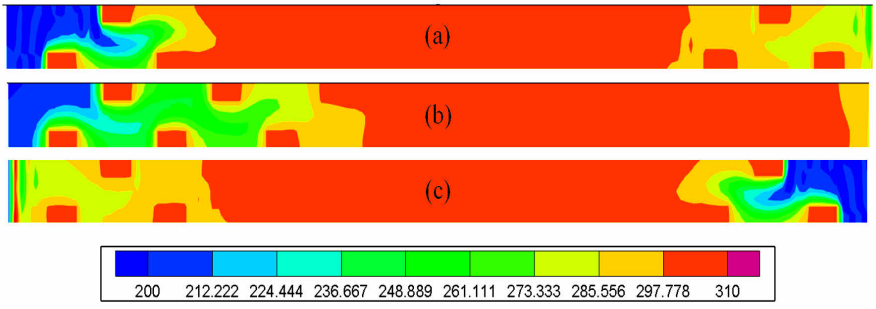

Figure 6. Temperature contours in Kelvin at different cycle times: (a) 4/16 cycle, (b) 8/16 cycle, (c) 12/16 cycle.

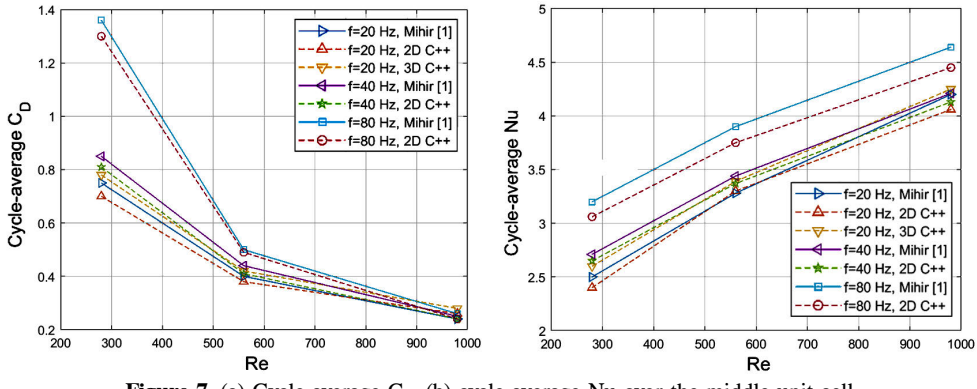

Figure 7. (a) Cycle average C_D, (b) cycle average Nu over the middle unit cell.

Figure 7b shows the dependence of Nu on both Re and frequency. Increasing either frequency or Re number will result in increasing Nu. Also, Nu appears to be more sensitive to the frequency in comparison to friction factor. Again, the present code predictions and Fluent results are within 10% of each other which shows the usability of developed code.

With CFD codes such as Fluent [5] such simulations typically take about a month or longer to reach steady-periodic conditions on Microsoft Surface Studio (Intel core i7-1065G7 CPU at 1.30 GHz, 32 GB RAM). In comparison, the new code reaches steady-periodic conditions in a little more than a week.

CONCLUSION AND FUTURE WORK

A 2D/3D code is under development in C++ to simulate the flow behavior in a random regenerator filler. The finite volume scheme and projection method is used to discretize and solve the conservation equations. A staggered grid methodology is used for discretizing the computational domain to enhance the stability of the pressure-velocity solutions. The fluid is assumed to be non-Newtonian but laminar with constant properties. Test studies were performed for flow through square bundles of tubes under oscillating flow. The cycle average Darcy friction factor and Nu number were calculated for the square bundle of tubes and the results match withing 10% in comparison to previous CFD analysis with Fluent. The code can be run for any generic regenerator geometry.

Different boundary condition for momentum and energy equations including wall, symmetry, oscillating velocity, Dirichlet and Neuman boundary condition are already embedded into the code. However, the code can be further enhanced by adding the capability to handle more boundary conditions. Temperature dependent properties will be another enhancement of the code. The code solves the energy equation only for the fluid region at this point. Thus, another major enhancement can be achieved by including the solid structure in conjugated heat transfer analysis.

NOMENCLATURE

u_i	Index notation for velocity	x_i	Index notation for coordinate system
u	Velocity in x direction	t	Time
v	Velocity in y direction	φ	Viscous dissipation
w	Velocity in w direction	n	Time step counter
ρ	Density	Nx	Total number of elements in x direction
μ	Dynamic viscousity	Ny	Total number of elements in y direction
ν	Kinematic viscousity	T_S	Surface temperature of regenerator
P	Pressure	D	Hydraulic diameter
Cp	Heat capacity	A	Surface area
k	Thermal conductivity	C	Coefficient ic pressure equation
U	Inlet amplitude velocity	C_D	Darcy friction factor
W	Width of computational domain	f	frequency
T	Temperature	$\vec{S}$	Surface area vector
T_∞	Reference temperature	′	Non dimensional variable
Re	Reynolds number	*	Intermediate variable
Nu	Nusselt number	< >	Cycle average
Pr	Prandtle number	i, j	Element counter in x and y direction

ACKNOWLEDGMENT

This work was funded by West Coast Solutions (wecoso.com) through the WCS-GT Cryocooler Research Fund.

REFERENCES

1. M. G. Pathak, T. Mulcahey, and S. M. Ghiaasiaan, "Hydrodynamic and thermal effects of drag and heat transfer coefficients under laminar unsteady flow conditions in porous media," *AIP Conference Proceedings*, 2012, vol. 1434, no. 1: American Institute of Physics, pp. 1875-1882.
2. T. Fang, A. Ghavami, and S. M. Ghiaasiaan, "A Second Law Study of the Regenerators in Cryocoolers based on Pore-level Analysis of Entropy Generation," *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 755, no. 1: IOP Publishing, p. 012049.
3. A. Ghavami, T. Fang, and S. M. Ghiaasiaan, "Entropy generation in the woven mesh regenerator filler of cryocoolers," *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 755, no. 1: IOP Publishing, p. 012064.
4. A. Ghavami, C. Kirkconnell, S. Ghiaasiaan, W. Chen, and M. Zagarola, "Analysis of a Novel Micro Structured Regenerator Filler," *Cryocoolers 20*, ICC Press, Boulder, CO (2018), pp. 235-244.
5. A. Fluent, "14.5 user's guide," *Fluent Inc., Lebanon, NH,* 2011.
6. C. Multiphysics, "Comsol multiphysics user guide (version 4.3 a)," *COMSOL, AB,* vol. 39, 2012.
7. D. Gedeon, "Sage user's guide, Stirling, pulse-tube and low-T cooler model classes," *Gedeon Associates,* 2014.
8. G. John and R. R. O'Gallagher Abbie, "Regen3. 3 user manual," ed: NIST, 2008.
9. H. Schildt, *C++: The complete reference*. McGraw-Hill/Osborne, 2003.
10. D. J. Higham and N. J. Higham, *MATLAB guide*. SIAM, 2016.
11. W. Bellevue, "Tecplot. 360 2010 User's Manual," ed: Tecplot, Inc, 2010.
12. K. A. Hoffman and S. T. Chiang, "Computational fluid dynamics," *Engineering Education System,* vol. 2, 2000.
13. S. Patankar, *Numerical heat transfer and fluid flow*, Taylor & Francis, 2018.