Three-dimensional transient finite element cooling simulation for injection molding tools

Lu Chen · Xiaowei Zhou · Zhigao Huang · Huamin Zhou

Received: 12 October 2021 / Accepted: 28 March 2022 / Published online: 5 May 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
Plastic injection molding is one of the most popular manufacturing processes for mass production, and optimizing the mold cooling system is critical for reducing the cycle time and improving the final part quality. This paper develops a three-dimensional transient cooling simulation model based on the finite element method. Comparing with the conventional cycle-averaged cooling model that uses boundary element method, this method has two major advantages: First, the transient mold temperature is more accurate than the cycle-averaged mold temperature. Second, this method allows performing cooling simulation directly on the real-world mold models without the simplifications that required by the boundary element method. To speed up the transient cooling simulation, this method uses the cycle-averaged cooling model as initial condition and applies the heat flux conservation equations at the discontinuous mesh boundaries to eliminate the interface iterations. It is shown that this method finishes the transient cooling analysis in 478 s on the real-world injection molding mold with more than 6.9 million tetrahedral elements, which is a satisfactory time for practical usages. The simulation result is validated by the actual molding experiment. It is found that the maximum temperature error is less than 4% and the average temperature error is less than 1%. The validation proves the accuracy of this simulation method.

Keywords Injection molding · Mold cooling · FEM · Transient cooling simulation

1 Introduction
Plastic injection molding is one of the most popular manufacturing processes for mass production. The typical injection molding stages are as follows: filling and packing stage, cooling stage, and mold open stage. The mold temperature within the injection molding cycle is transient in nature, which increases at the filling stage and decreases at the subsequent stages. Among all these molding stages, cooling stage is of particular importance as it takes more than 70% of the entire cycle time. Hence, the cooling system of the mold tool plays a crucial role as it has huge impact to both cycle time and part quality. Indeed, a proper cooling system design can significantly reduce the cost by shortening the cycle time and avoid the final part defects such as warpage and sink marks [1].

There are two purposes for performing the injection molding cooling simulation. First, it provides the mold designers with accurate mold cooling results to guide their design decisions before the molds are manufactured. Second, the simulated mold temperatures can be used as a more accurate thermal boundary condition for the subsequent filling and packing simulation. The earliest while most wildly used mold cooling simulation method is the boundary element method (BEM)-based cycle-averaged cooling model [2]. The BEM-based model has been the mainstream method in the past decades, and there have been extensive researches in this regard [3–8]. Besides the BEM-based model, there were also some models based on the finite element method (FEM) [9] and finite volume method (FVM) [10].

BEM reduces the mold from solid domain to the surface domain; hence, the discretization and computation are only carried out at the surfaces. This is the dominate advantage of BEM, but the cost of reducing dimension is not trivial. The BEM generates a full dense asymmetric matrix, which takes huge memory and long time to solve on large mesh. Many
efforts have been made to overcome such defects. The fast multipole method (FMM) [11–13] has been applied to BEM, which is a matrix-free method and can be highly parallelized. FMM can significantly reduce the memory usage and improve the performance. However, it is reported that the convergence is not always guaranteed [12]. Another widely adopted method is the domain decomposition method (DDM) [14–16]. DDM decomposes the initial domain into subdomains, and inside each subdomain, the solution is solved with the original governing equation. These subdomains can be solved on multiple computational nodes in parallel which reduces the memory usage on a single node. Between the adjacent subdomains, there exist internal interfaces on which the interface conditions must be satisfied. This is accomplished by iteratively solving the subdomains until converge, and such process is slow. On the contrary, FEM generates a sparse positive definite matrix which has better properties than the matrix from BEM. It uses much less memory than BEM matrix due to its sparsity, and the positive definite matrix can be solved by the conjugate gradient methods efficiently. In BEM-based injection molding cooling simulation, the mold temperature is solved with BEM, but the plastic part can be solved with one-dimensional finite difference method (FDM) for thin geometries or three-dimensional FEM for chunky geometries. Traditional BEM-based cooling method must perform an iterative process between the part and mold to achieve the part-mold interface heat flux conservation, which is a time-consuming process and introduces convergence difficulties. Zhang at el. [7] proposed a novel method to remove such iteration by integrating the analytic solution of thermal analysis of the plastic part into the boundary integral equations of the mold. In FEM, since both the plastic part and mold use three-dimensional mesh, the part and mold temperatures can be solved simultaneously by applying the interface heat flux conservation equations proposed in this work. The iterations between part and mold can be eliminated; hence, the analysis speed and convergence can be improved.

The BEM does not support simulating the real-world three-dimensional mold. As BEM only performs analysis on boundaries, it uses a box to represent the actual mold, which is a huge simplification and loses many details. Although this is acceptable from the engineering perspective in many cases, such simplifications cause difficulties on simulating advanced features such as the heat resistance between multiple mold plates, hot runner system, heater cartridge system, or special mold insert such as the 3D-printed conformal cooling channel insert.

The cycle-averaged mold temperature assumption is not always true. It assumes the mold temperature does not change significantly during the injection molding cycle, but there are exceptions. For some deep hollowed parts such as the car lights, the conventional cooling channels are difficult to cool down the mold core side, so the mold core temperature can have large variation within the cycle. In other cases such as the two-shot overmolding process [17] and rapid heating and cooling process [18], the cycle-averaged assumption no longer holds. Even for the conventional injection molds, the transient mold temperature is still more accurate than the cycle-averaged solution. The reason why using cycle-averaged model is because the transient analysis takes much longer time to run. The transient mold temperature should be used if the transient cooling analysis runs with acceptable performance.

The objective of this paper is to develop a three-dimensional transient FEM cooling algorithm for the injection molding process. This work has three major highlights: First, the performance issue of conventional BEM is improved by using FEM. FEM generates sparse positive definite matrix which can be solved more efficiently than the dense matrix from BEM, and the FEM-based heat flux conservation equations are applied at the discontinuous mesh boundaries to eliminate the part-mold iterations. Second, this work allows using three-dimensional mold mesh to describe the mold system without any compromises; the mold designers can start the cooling simulation directly from the complete CAD models of mold and plastic part; hence, a true three-dimensional simulation over both part and mold can be performed, and it allows for arbitrary number of separate mold components with different mesh sizes to be solved simultaneously. Third, a more accurate transient cooling simulation can be performed efficiently. This work employs the cycle-averaged mold temperature solution as the initial condition for the transient cooling analysis, and it converges much faster than the uniformed initial mold temperature that is used in [9].

This paper is organized as below. First the modeling assumptions are given in Sect. 2. In Sect. 3, the governing equations, boundary conditions, and the numerical implementation are discussed. In Sect. 4, an initial condition for speeding up the transient cooling analysis is given, and Sect. 5 discusses a special boundary condition treatment for the mold-mold boundary and mold-polymer boundary. In Sect. 6, the experiment and simulation setup are given for validation purpose, then the simulation results are compared against the experimental results in Sect. 7.

## 2 Modeling and basic assumption

As mentioned in Sect. 1, the typical injection molding stages are filling and packing stage, cooling stage, and mold open stage. The melt polymer is injected into the mold cavity during filling stage which heats up the mold, and the cooling system keeps pumping coolant into the mold to bring the mold temperature down. At the very beginning of the production, the mold is completely cold at the surrounding
temperature. After multiple injection cycles, there exists a heat balance such that the mold temperature curves are similar between cycles, and the production runs under such stable injection molding cycles. In Fig. 1, the transient mold temperature gradually builds up from the beginning of the injection production, after a few injection molding cycles, the temperature enters a stable cyclic state, in which the mold temperature varies within the cycle but all the injection molding cycles are the same [2]. The cycle-averaged mold temperature, the red line in the curve, is the averaged value of such stable cycles. Conventional cooling simulation calculates this cycle-averaged mold temperature, but the aim of this work is to quickly solve the transient mold temperatures under such stable cycles.

From the energy conservation point of view, the injection mold absorbs heat from part through the mold-polymer boundary, and leaks heat into the coolant and surrounding through the mold-coolant boundary and mold-air boundary as shown in Fig. 2. These effects can be regarded as heat exchange with the external system of the mold. Inside the mold system, there are also multiple mold components which are closely contacted to each other, such as the mold-mold boundary in Fig. 2. These mold platen contact boundaries usually are neglected in BEM by assuming the mold is a simple box, but the real mold is a complex assembly with many separate components. These mold components may not always be in perfect contact; there can be gaps or grease in between. Hence, the thermal conduction through the contact interface can be less efficient than the perfect contact situation.

The filling time is usually very short comparing with the packing time, cooling time, and mold open time, and the filling stage can be simplified by assuming the melt polymer is full of the cavity at the beginning of the injection molding cycle. Hence, the convection effect of the melt polymer can be ignored and only the heat conduction effect needs to be addressed, which greatly simplified the governing equation.

From the transient cooling simulation perspective, the governing equations of the whole computational domain, initial conditions, and boundary conditions are needed to solve the problem. Figure 3 shows this transient cooling simulation workflow (B) and compares against the conventional cooling simulation workflow (A). It can be seen that there are two major differences in this work. First, the part and mold coupling loop is not needed, which can significantly improve the simulation speed and accuracy and avoid the numerical convergence issues during the part and mold iterations. Apart from that, it also allows solving arbitrary number of separate mold blocks; hence, this work can solve the complex CAD mold models directly. Second, there is one extra step which solves the initial condition for this transient
cooling analysis; the aim of this step is to find a better initial value for the transient analysis to speed up the convergence.

3 Transient FEM cooling formulation

3.1 Governing equation

As the filling stage of the melt polymer has been neglected, only the transient effect and conduction effect need to be considered; hence, the part and mold share the same governing equation, which is given in Eq. (1) [9].

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T)$$  

(1)

In Eq. (1) above, $\rho$ is the density, $C_p$ is the specific heat, $k$ is the thermal conductivity, $t$ is the time, and $T$ is the temperature. The material properties $\rho$, $C_p$, and $k$ are different for polymers and mold.

The left side of Eq. (1) is the transient term, and the right side is the diffusion term. The Galerkin method is applied on Eq. (1) to execute the FEM discretization. The weak form of the diffusion term is defined in Eq. (2),

$$\int N^T \nabla \cdot (k \nabla T) dV$$

(2)

where $N$ is the element shape function vector. Integrate Eq. (2) by parts yields Eq. (3),

$$\int N_S^T n \cdot q dS - \int \nabla N^T \cdot (k \nabla T) dV$$

(3)

where the first term is the external surface integration of the element, $N_S$ is the surface shape function vector, $n$ is the unit surface normal vector, and $q$ is the heat flux applied on this surface by external system. The second term results in the symmetric definite elemental matrix which is the fundamental part of the FEM discretization.
An implicit finite element time integration scheme is applied on the transient term. The cycle time is discretized into M steps such that 0 ≤ t1 ≤ t2 ... ≤ t_{k−1} ≤ tM. The temperature of tk can be advanced by the temperature of t_{k−1} by Eq. (4),

\[
\int N^T \frac{\rho C_p}{\Delta t} T^k dV = \int N^T \frac{\rho C_p}{\Delta t} T^{k-1} dV
\] (4)

Substitute Eq. (4) into (3), the elemental discretized matrix is given by Eq. (5),

\[
\int \left( \frac{\rho C_p}{\Delta t} N^T N + kN^T \nabla N \right) dV \cdot \{ T^k \} = \int \frac{\rho C_p}{\Delta t} N^T T dV \cdot \{ T^{k-1} \} + \int N^T n \cdot q dS
\] (5)

where the \{ T^k \} is element nodal temperature unknowns at time step k. The \{ T^{k-1} \} is element nodal temperature unknowns of time step k−1 which are solved at the last time step.

Apply Eq. (5) on all part elements and mold elements, then system matrix is formed which is a sparse symmetric positive definite matrix. However, the boundary conditions on the mold-polymer boundary, mold-mold boundary, mold-coolant boundary, and mold-air boundary still need to be applied to the system matrix before solving the temperature.

### 3.2 Mold-polymer boundary condition

Conventional mold-polymer boundary condition applied to the mold side is done via an iterative approach, as shown on the left side of Fig. 3. At the beginning of the analysis, the mold is assumed at a guessed temperature, and such guessed mold temperature is used as boundary condition to solve the part temperature, then the heat flux is calculated from part side and used as boundary condition to solve the mold temperature, by running this loop until the mold temperature variation and part heat flux variation are small enough, then the part and mold temperatures are thought to be converged. This is a slow process, and the mold temperature variation or part heat flux variation may never converge on complex models. The reason is that the mold mesh and part mesh are separated meshes, they are not connected at the interfaces; hence, the heat cannot pass through from one mesh to the other, then an iterative method must be performed to reach the heat flux balance at the mold-polymer boundary.

This paper develops a method to avoid the iterative way at the mold-polymer boundary. According to [19], at the mold-polymer interface, the mold temperature is not exactly equal to part temperature; there is a heat transfer coefficient h between the part and the mold. The heat transfer coefficient has a close relationship with the contact situation of the mold-polymer interface, and it is impacted by material properties, processing parameters, and surface roughness. The heat transfer coefficient is not a constant value during the injection molding cycle; it tends to be larger at the filling and packing stages due to high melt temperature and high pressure which build up good contact area between the polymer and mold. At the cooling stage, the heat transfer coefficient decreases as the gate is frozen and polymer starts to shrink. The air gap between polymer and mold enlarges which results in larger heat resistance. Despite the complexities of the heat transfer coefficient determination, in this work, a constant heat transfer coefficient is adopted based on the recommended range from the researches [19–21], and it is straightforward to extend the existing ability to support the tabular heat transfer coefficient. The heat flux at the mold-polymer interface \( q_p \) can be defined in Eq. (6), in which \( T_m \) is the mold temperature and \( T_p \) is the part temperature.

\[
q_p = h_p(T_p - T_m)
\] (6)

In Eq. (6), both the \( T_m \) and \( T_p \) are unknown; hence, it cannot be treated as normal thermal boundary condition. Rewrite Eq. (5) into the matrix form and separate mold nodal temperatures and part nodal temperatures, then we can have the Eq. (7). \( K_p \) is the thermal system matrix over the entire part domain, \( K_m \) is the thermal system matrix over the entire mold domain, \( T_p \) is the part temperature unknowns, \( T_m \) is the mold temperature unknowns, and \( rhs_p \) and \( rhs_m \) are right hand side known values for part and mold.

\[
\begin{bmatrix}
K_p & 0 \\
0 & K_m
\end{bmatrix}
\begin{bmatrix}
T_p \\
T_m
\end{bmatrix} =
\begin{bmatrix}
 rhs_p \\
 rhs_m
\end{bmatrix}
\] (7)

By substituting Eq. (6) into (7), we can have the mold-polymer boundary constrained system Eq. (8). \( K_{pn} \) stands for the mold-polymer boundary constraints for part nodes, and \( K_{mp} \) stands for the mold-polymer boundary constraints for mold nodes. \( K_{pm} \) and \( K_{mp} \) are not necessarily equal. It depends on the mesh at the boundary and will be discussed in Sect. 5. The original \( K_p \) and \( K_m \) now become \( K'_p \) and \( K'_m \) as the Eq. (6) also adds up to coefficients to them.

\[
\begin{bmatrix}
K'_p & -K_{pm} \\
-K_{mp} & K_m
\end{bmatrix}
\begin{bmatrix}
T_p \\
T_m
\end{bmatrix} =
\begin{bmatrix}
 rhs_p \\
 rhs_m
\end{bmatrix}
\] (8)

With the Eq. (8), the mold and part temperatures can be solved simultaneously in one single system matrix without any iterative method needed.
3.3 Mold-mold boundary condition

The real injection mold is never simple; it consists of hundreds of components. Even after excluding unimportant components, there are still multiple important mold platens for simulation. Similar with Sect. 3.2, since these mold platens are separate CAD bodies and separate meshes as well, the best way to handle the mold-mold boundary condition is to apply Eq. (9) into Eq. (8). \( h_{mij} \) is the heat transfer coefficient between the mold platen \( i \) and mold platen \( j \), and \( T_{m_i} \) and \( T_{m_j} \) are the mold temperatures of the mold platen \( i \) and mold platen \( j \). \( h_{mij} \) is not trivial as the mold platens may not always be in perfect contact. There might be gap, grease, or some intentionally added insulating layer between mold platens.

\[
q_{mij} = h_{mij} \left( T_{m_i} - T_{m_j} \right) \quad (9)
\]

Similar with the treatment in Sect. 3.2, by substituting Eq. (9) into (8), we can obtain system Eq. (10) with mold-mold boundary condition handled.

\[
\begin{bmatrix}
K_p & -K_{pm_i} & -K_{pm_j} \\
-K_{pm_i} & K_{m_i} & -K_{m_i}' \\
-K_{pm_j} & -K_{m_i}' & K_{m_j}'
\end{bmatrix}
\begin{bmatrix}
T_p \\
T_{m_i} \\
T_{m_j}
\end{bmatrix} =
\begin{bmatrix}
\text{rhs}_p \\
\text{rhs}_{m_i} \\
\text{rhs}_{m_j}
\end{bmatrix} \quad (10)
\]

3.4 Mold-air boundary condition

At the mold-air boundary, the boundary condition is defined in Eq. (11), in which \( q_a \) is the heat transfer coefficient between mold and air. The heat dissipates into the air which is not major in cooling simulation and \( h_a = 10 W \cdot K^{-1} \cdot m^{-2} \) is a good engineering approximation. \( T_a \) is assumed to be a known constant.

\[
q_a = h_a (T_m - T_a) \quad (11)
\]

3.5 Mold-coolant boundary condition

At the mold-coolant boundary, the boundary condition is defined in Eq. (12), in which a heat transfer coefficient \( h_c \) between the mold and coolant [6] is defined in Eq. (13) to approximate the cooling effect of the cooling system. The \( q_c \) is the heat flux at the mold-coolant boundary, \( T_m \) is the mold temperature, \( T_c \) is the coolant temperature, \( Re \) is the Reynolds number, \( Pr \) is the Prandtl number, \( k_c \) is the thermal conductivity of the coolant, and \( D \) is the diameter of the cooling channel.

\[
q_c = h_c (T_m - T_c) \quad (12)
\]

\[
h_c = 0.023 \frac{k_c}{D} Re^{0.8} Pr^{0.4} \quad (13)
\]

4 Initial condition for transient cooling

The transient cooling simulation governed by Eq. (1) can start from an initial temperature and update temperature field step-by-step for many injection cycles, until the stable cycle reaches as shown in Fig. 1. However, starting from an arbitrary initial value can make the mold temperature to reach the stable cycle slowly. A good initial value for mold temperature can significantly improve the convergence of the simulation. This section develops the three-dimensional FEM-based cycle-averaged steady state mold temperature as the initial value for the transient cooling analysis.

The governing equation for the cycle-averaged model for the mold side is given in Eq. (14). \( k_m \) is the thermal conductivity of the mold.

\[
\nabla \cdot (k_m \nabla T) = 0 \quad (14)
\]

At the mold-polymer boundary, the cycle averaged heat flux is used as boundary condition. The cycle averaged heat flux \( \bar{q}_p \) is defined in Eq. (15), in which \( \tau_c \) is the cycle time, and \( q(t) \) is the instantaneous heat flux [6].

\[
\bar{q}_p = \frac{1}{\tau_c} \int_0^{\tau_c} q(t) dt \quad (15)
\]

At the mold-mold boundary, mold-coolant boundary, and mold-air boundary, the treatment is the same with Eqs. (9), (10), (11), (12), and (13) in transient simulation.

5 Mold-polymer and mold-mold boundary treatment

The normal FEM procedure, when discretizing Eq. (1), results in a sparse symmetric positive definite system matrix, and such kind of matrices can be easily solved
with the conjugate gradient solvers. However, in Sects. 3.2 and 3.3, when dealing with the mold-mold boundary and mold-polymer boundary, the system matrix may violate the symmetric property if the meshes at the contact region are non-conformal. Figure 4 demonstrates the conformal mesh (A) and non-conformal mesh (B) in two-dimensional space for a more intuitive understanding, and it can be easily extended to three-dimensional tetrahedral mesh of this work. In conformal mesh, the nodes at the contact region are exactly one–one match, such as the red nodes and green nodes in the conform mesh plot. However, in non-conformal mesh, there is no such property.

Recall Eq. (6) for the three-dimensional tetrahedral conformal mesh shown in Fig. 5. \( N_p \) is the FEM node on the plastic part surface, \( S_p \) is the surface area of node \( N_p \), \( N_m \) is the FEM node on the mold surface, and \( S_m \) is the surface area of node \( N_m \). As the mesh is conformal at the mold-polymer boundary, we have \( N_p \) that is coincident with \( N_m \), and \( S_p \) equals to \( S_m \). Integral Eq. (6) for node \( N_p \) yields \( h_p \cdot S_p \cdot \left( T_{N_p} - T_{N_m} \right) \); hence, \( K_{pm} = h_p \cdot S_p \), and integral Eq. (6) for node \( N_m \) yields \( K_{mp} = h_p \cdot S_m \). The system matrix in Eq. (8) is still symmetric positive definite as \( K_{pm} = K_{mp} \).

Recall Eq. (6) for the non-conformal mesh shown in Fig. 6, as the mesh is non-conformal at the contact region. The projection of polymer node \( N_p \), which is denoted as \( N_p\prime \), must be inside one surface triangle of the tetrahedral element. The three surface nodes of this surface mold tetrahedral element are \( N_{m1}, N_{m2}, \) and \( N_{m3} \). \( S_{m1}, S_{m2}, \) and \( S_{m3} \) are the areas of the triangles as indicated in Fig. 6. Integral Eq. (6) for plastic node \( N_p \) yields \( h_p \cdot S_p \cdot \left( T_{N_p} - \frac{s_{m1}}{s_{m1} + s_{m2} + s_{m3}} T_{N_{m1}} - \frac{s_{m2}}{s_{m1} + s_{m2} + s_{m3}} T_{N_{m2}} - \frac{s_{m3}}{s_{m1} + s_{m2} + s_{m3}} T_{N_{m3}} \right) \). \( K_{pm} \) no longer equals to \( K_{mp} \) under non-conformal mesh; hence, the system matrix in Eq. (8) is asymmetric.

If \( N_p\prime \) in Fig. 6 is coincident with any node of \( N_{m1}, N_{m2}, \) or \( N_{m3} \), then it becomes the case of conformal mesh. Hence, the heat flux conservation equation on non-conformal mesh is the generalization of the equation on conformal mesh, and in this work, the non-conformal mesh is always used. Applying the heat flux conservation equations on the mold-polymer and mold-mold boundaries

\[ \text{Fig. 4 Conformal mesh and non-conformal mesh. A Conformal mesh, B non-conformal mesh} \]

\[ \text{Fig. 5 Conformal mesh treatment at the contact surface} \]
sensors near the part surface could cause damages, so in the experiment, two Kistler 6190CA sensors were embedded at the mold surface of the runner system. This is a two-cavity configuration, in which a pair of plastic parts can be made in one single shoot. The whole runner system is cold runner, without any hot runner.

The plastic material for injection molding is SABIC polycarbonate OQ2720. Tables 1 and 2 list the material thermal properties and PVT properties used in transient cooling analysis. The material specific heat, thermal conductivity, and the PVT model parameters were tested by the Gottfert RG50 rheometer in our own lab.

For thermoplastic polymers, the density is a function of pressure and temperature. A commonly used equation is the Tait equation [23] which is defined in Eq. (17),

\[ V(p, T) = V_0(p, T) \left[ 1 - C \ln \left( 1 + \frac{p}{B(T)} \right) \right] + V_t(p, T) \]  

(17)

where \( C = 0.0894 \), \( T \) is temperature, \( p \) is pressure, \( V \) is specific volume, and \( V_0(p, T) \) is given by Eq. (18),

\[ V_0(p, T) = \begin{cases} \frac{b_1^s + b_3^s(T - b_5)}{b_0^s}, & \text{if} \ T \leq T_{\text{trans}} \\ \frac{b_0^m + b_3^m(T - b_5)}{b_0^m}, & \text{if} \ T > T_{\text{trans}} \end{cases} \]  

(18)

where the superscripts \( s \) and \( m \) represent the solid and melt state of the polymer, and \( B(T) \) is given by Eq. (19),

\[ B(T) = \begin{cases} b_3^s \exp \left( \frac{-b_4^s(T - b_5)}{b_0^s} \right), & \text{if} \ T \leq T_{\text{trans}} \\ b_3^m \exp \left( \frac{-b_4^m(T - b_5)}{b_0^m} \right), & \text{if} \ T > T_{\text{trans}} \end{cases} \]  

(19)

For amorphous polymers, \( V_s(p, T) = 0 \), and for semi-crystalline polymers, \( V_t(p, T) \) is given by Eq. (20),

\[ V_t(p, T) = \begin{cases} b_7 \exp(b_4(T - b_5) - b_9T), & \text{if} \ T \leq T_{\text{trans}} \\ 0, & \text{if} \ T > T_{\text{trans}} \end{cases} \]  

(20)

where \( T_{\text{trans}} \) is the transition temperature which is defined in Eq. (21).

\[ T_{\text{trans}} = b_5 + b_9T \]  

(21)

For the thermoplastic material used in this validation, Table 2 lists all the parameters of the PVT model.

The mold material is P20 and its thermal properties in Table 3 are cited from the public website. The injection molding conditions are given in Table 4; since we have not tested the actual heat transfer coefficients, we manually choose a constant value based on the recommended range from the researches [19–21].

The CAD software used in this work is FreeCAD, and mesh is generated from the open-sourced library gmsh. The FEM solver for this transient cooling simulation is self-developed, but inside the transient cooling solver the matrix is solved with the open-sourced library AMGCL.

Fig. 6 Non-conformal mesh treatment at the contact surface
The post processor for results visualization is another open-sourced software ParaView.

The CAD assembly of this injection mold has 142 components; after removing some small unimportant features, there are 12 CAD mold components used to generate three-dimensional tetrahedral mesh for transient cooling analysis as shown in Fig. 8. The mesh statistics shows that there are 6,924,353 tetrahedral meshes. Among the 6,924,353 meshes, there are 742,047 tetrahedral meshes for the plastic part, and the rest 6,182,306 elements are the mold meshes.

### Results and discussion

The experiment performed injection molding trials under three different initial melt temperatures: 320°, 305°, and

| Table 2 Material PVT model |
|----------------------------|
| Material PVT model        | Values            |
|---------------------------|-------------------|
| $b_7(K)$                  | 418.93            |
| $b_8(K\cdot Pa^{-1})$     | 3.21e-07          |
| $b_9(m^3 \cdot Kg^{-1} \cdot K^{-1})$ | 0.0008654         |
| $b_{10}(Pa)$              | 0.0008654         |
| $b_{11}(K^{-1})$          | 1.73897e+08       |
| $b_{12}(m^3 \cdot Kg^{-1} \cdot K^{-1})$ | 0.003462         |
| $b_{13}(Pa)$              | 0.003462          |
| $b_{14}(K^{-1})$          | 0.0008612         |
| $b_{15}(m^3 \cdot Kg^{-1} \cdot K^{-1})$ | 5.85e−08         |
| $b_{16}(Pa)$              | 3.4343e+08        |
| $b_{17}(K^{-1})$          | 0.00227           |
| $b_{18}(m^3 \cdot Kg^{-1})$ | 0                |
| $b_{19}(K^{-1})$          | 0                |
| $b_{20}(Pa)$              | 0                |
| $b_{21}(K^{-1})$          | 0                |
290°. For each initial melt temperature condition, three experiments were performed and the actual temperature data was collected. Figure 9 shows the simulation temperature results and the three experimental temperature results on sensors A and B for 320° melt temperature, “sim” stand for simulation, and “exp-1,” “exp-2,” and “exp-3” stand for the three experiments. The left side of Fig. 9 plots the simulated temperature and in the actual temperatures from the temperature sensors, we can see the simulation matches experiments well from both trend and magnitude perspective. The in-cycle mold temperature starts at a relative low value, when the injection + packing starts, the mold temperature increases quickly and then cools down gradually to the low temperature similar with the beginning of the cycle. The right side of Fig. 9 plots the temperature error between simulation and experiments at each time step; the experimental mold temperature used for this plot is the averaged mold temperature over the three runs; then by calculating the error in percentage against the simulation result for each time step, we can see that the maximum error is 3.79%. Figures 10, 11 plot the same results for the initial melt temperature under 305° and 290°. The maximum error and average error for this experiment are listed in Table 5.

The polymer and mold are not in perfect contact according to [19]. The mold temperature and part temperature are not equal at the mold-polymer boundary, as there is a heat resistance at the mold-polymer boundary. This work uses the heat transfer coefficient $h_p$ to model such thermal behavior. Larger $h_p$ means smaller heat resistance, then the polymer temperature and mold temperature at the mold-polymer boundary are closer to each other, and vice versa. In this section we will investigate the effect of the
heat transfer coefficient at mold-polymer boundary with \( h_p = 2500 \text{W} \cdot \text{K}^{-1} \cdot \text{m}^{-2} \) and check the temperature difference between the mold temperature and its contacted plastic part temperature. The locations to check are selected at the center of cavity (locations B and D) as shown in Fig. 12.

Similar with the treatment at the mold-polymer boundary, the heat transfer coefficient \( h_m \) is also applied at the mold-mold boundary. However, unlike the mold-polymer boundary, the heat resistance between the mold-mold boundary is smaller. In this work, the heat transfer coefficient at the mold-mold boundary is set as \( h_m = 30000 \text{W} \cdot \text{K}^{-1} \cdot \text{m}^{-2} \). Figure 12 indicates the location of the two checkpoints which are at the contact region between mold core platen and mold cavity platen (locations A and C). We use those two locations to examine the heat resistance implementation at mold-mold boundary by checking their temperature difference.

Figure 13 plots the temperature difference at the boundaries within the cycle for locations A, B, C, and D. Clearly, we can see that the temperature differences for B and D are much larger than locations A and C. This is because the locations B and D are at the mold-polymer boundary, where we set smaller heat transfer coefficient. Hence, the heat conduction from hot polymer into the mold through the contact boundary is slower due to larger interface heat resistance. The temperature differences for B and D are larger than 200° at the begin of the cycle, that is, when the initial melt temperature is 320° while the mold is around 110°. Because of this heat resistance effect, the mold temperature does not jump to the polymer temperature instantaneously, but rather the temperature difference decreases gradually, and finally, the difference is approaching zero at the end of the cycle. Such temperature gap transition means that the heat resistance effect has been correctly modeled in the simulation. The temperature difference for locations A and C is less than
0.3°, which is much smaller than locations B and D. This is because locations A and C are at the mold core and mold cavity contact interface, where the heat transfer coefficient is larger and smaller heat resistance at the thermal contact interface. It is observed from Fig. 12 that the locations A and C are symmetric; hence, it is expected that the temperatures of A and C should be close to each other, but in Fig. 13, there is a slight difference between A and C. This is because the cooling channels are not symmetric, and location C is closer to the cooling channel inlet than location A. Similarly, for the curves of locations B and D in Fig. 13, there is also a slight difference between B and D. But the value of Y axis at the first time step is large, which makes the difference between B and D not as observable as A and C in Fig. 13.

Figure 14 shows the transient mold temperatures under the stable cycle. Under the molding condition, the initial melt polymer temperature is 320°. All the plots in Fig. 14 are rendered in the open-sourced software ParaVi33333ew.

From the plots, we can see that the highest temperature appears at the mold core side which is directly in contact with the melt polymer, the mold temperature first increases at the beginning of the injection cycle, and then the temperature drops down. It should be noted that the temperature of top half mold is higher than the bottom half mold. The temperature of top half mold is slightly lower than 110°, but the bottom half mold is much lower than 110°. There are 4 cooling channels at the top half mold, but only 1 cooling channel at the bottom half, and the cooling channel inlet temperature is 110°. This means that the coolant heats up the mold and maintains the mold temperature at about 110°.

Figure 15 shows the transient plastic part temperature under the stable cycle at different time steps. From the plots, we can clearly see the part cools down from surface region to the core region at the end of the injection cycle. At the end of the injection molding cycle, the plastic part temperature drops down to around 110°, which is the coolant inlet temperature.
This work also performed two extra analyses: (1) transient cooling simulation with a uniform mold temperatures (40°) as initial condition; (2) cycle averaged cooling simulation, which is the initial condition in Sect. 4. These two extra simulations are used to compare against the transient cooling simulation presented in this work. The metrics used are the execution time and the average cavity temperature.

For the two transient simulations, the average cavity temperature is the average temperature over all the mold nodes at the mold-polymer boundary through the stable injection molding cycle, defined by Eq. (22),

\[ T_{\text{avg}} = \frac{1}{t_c} \int_0^{t_c} \frac{1}{n} \sum_{i=0}^n T_m^i \, dt \]  

(22)

where \( t_c \) is the cycle time, \( n \) is the number of mold nodes at the mold-polymer boundary, and \( T_m^i \) is the temperature of the mold node \( i \).

For the cycle averaged cooling simulation, the average cavity temperature is the average temperature over all the mold nodes at the mold-polymer boundary, defined by Eq. (23),

\[ T_{\text{avg}} = \frac{1}{n} \sum_{i=0}^n T_m^i \]  

(23)

Such definition on the average cavity temperature makes the comparison between transient simulation and steady
Fig. 12  Mold-mold/mold-polymer temperature difference check points

Fig. 13  Mold-mold/mold-polymer temperature difference plot
Fig. 14 The transient mold temperatures at the stable cycle

The International Journal of Advanced Manufacturing Technology (2022) 120:7919–7936
Fig. 15  The transient part temperatures at the stable cycle
state simulation possible. From Table 6, we can see that all these three simulations are at the same level of accuracy: (1) the error between the cycle averaged cooling simulation and the transient cooling simulation is 0.2%, which proves the correctness of this transient cooling method. As according to the definition in Fig. 1, the mold temperature of the cycle averaged model is the averaged value of the transient mold temperature at the stable cycles. But the transient cooling model is more accurate than the cycle averaged model as it can provide the temperature difference inside the cycle, which can be large as shown in Figs. 9, 10, 11 and (2) the average cavity temperature error between the two transient FEM methods is 0.15%; hence, we can conclude that the initial condition proposed in Sect. 4 has no negative impact to the simulation accuracy. When comparing the execution time, we can see that the FEM with solved initial condition is more than 14 times faster than the FEM with uniform initial condition, which is a significant speed acceleration. Although it is 4.3 times slower than the cycle averaged cooling simulation, it is still satisfactory considering it takes less than 8 min to finish.

| Method                        | Execution time (s) | Average cavity temperature (°C) |
|-------------------------------|--------------------|---------------------------------|
| FEM with solved initial condition | 478                | 113.05                          |
| FEM with uniform initial condition | 6707               | 112.88                          |
| Cycle averaged cooling simulation | 110                | 113.28                          |

8 Conclusions

Conventional cooling simulation for injection molding uses BEM to perform the cycle-averaged cooling analysis on simplified mold geometries. Using cycle-averaged temperature and simplified mold to approximate the transient temperature of real-world mold loses accuracy. This work develops a three-dimensional transient FEM cooling model to overcome the shortages of the existing methods.

(1) The transient mold temperature is more accurate than the cycle-averaged mold temperature, especially for the special molding processes such as two-shot overmolding process and rapid heating and cooling process.

(2) This work uses the three-dimensional FEM mesh for both part and mold and hence avoids the dense asymmetric system matrix generated by BEM. By applying the heat flux conservation equations at the part and mold boundary, the numerical iterations between the part and mold can be eliminated to improve the analysis speed and convergence.

(3) This work allows performing cooling analysis on real-world complex mold models. The non-conformal meshes of the multiple mold plates can be handled by the heat flux conservation equations at the discontinuous mold mesh boundaries.

(4) This work implements the cycle-averaged cooling model as initial condition of the transient cooling simulation, to further improve the transient cooling simulation speed.

This transient cooling method is validated by an actual injection molding experiment. The analysis finishes in 478 s on the model with more than 6.9 million tetrahedral elements. The simulated mold temperature is compared with the actual mold temperature; it is found that the maximum temperature error is less than 4%, and the average temperature error is less than 1%. Hence, it proves that this method can simulate the transient mold temperature with a quite satisfactory computation time and adequate accuracy.

Author contribution Zhiqiao Huang conceived the idea; Zhiqiao Huang and Lu Chen conducted the analyses; Lu Chen and Xiaowei Zhou participated in the development and testing of the prototype system; all authors contributed to the writing of the manuscript.

Funding This research was funded by the National Natural Science Foundation of China, Grant No. U20A20288.

Availability of data and material All data and models generated or used during the study appear in the submitted manuscript.

Code availability All code generated or used during the study appear in the submitted manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Sardo L, Daldoul W, Vincent M, Toulorge T (2019) Simulations of heat transfer in thermoplastic injection molds manufactured by additive techniques. Int Polym Proc 34(1):37–46. https://doi.org/10.3139/217.3594
2. Kwon TH (1988) Mold cooling system design using boundary element method. Journal of Engineering for Industry 110(4):384–394. https://doi.org/10.1115/1.3187898
3. Koo BH, Choi KI (2005) Three-dimensional mold cooling analysis for injection molding process. society of plastics engineers. Annu Tech Conf 2:103
4. Qiao H (2006) A systematic computer-aided approach to cooling system optimal design in plastic injection molding. Int J Mech Sci 48(4):430–439. https://doi.org/10.1016/j.ijmecsci.2005.11.001
5. Hioe Y, Chang KC, Zuyev K, Bhagavatula N, Castro JM (2008) A simplified approach to predict part temperature and minimum
“safe” cycle time. Polym Eng Sci 48(9):1737–1746. https://doi.org/10.1002/pen.21133
6. Zhou H, Zhang Y, Wen J, Li D (2009) An acceleration method for the BEM-based cooling simulation of injection molding. Eng Anal Boundary Elem 33(8–9):1022–1030. https://doi.org/10.1016/j.enganabound.2009.04.001
7. Zhang Y, Huang Z, Zhou H, Li D (2015) A rapid BEM-based method for cooling simulation of injection molding. Eng Anal Boundary Elem 52:110–119. https://doi.org/10.1016/j.enganabound.2014.11.020
8. Li Z, Wang X, Gu J, Ruan S, Chen C, Lyu Y, Zhao Y (2018) Topology optimization for the design of conformal cooling system in thin-wall injection molding based on BEM. The International Journal of Advanced Manufacturing Technology 94(1):1041–1059
9. Tang LQ, Pochiraju K, Chassapis C, Manoochehri S (1996) Three-dimensional transient mold cooling analysis based on galerkin finite element formulation with a matrix-free conjugate gradient technique. Int J Numer Meth Eng 39(18):3049–3064
10. Chang RY, Hsu D, Yang V, Yang W (2003) Three-dimensional computer-aided mold cooling design for injection molding. Proceedings of the annual technical conference – ANTEC, conference proceedings society of plastics engineers 1:656–660
11. Malhotra D, Biros G (2016) Algorithm 967: a distributed-memory fast multipole method for volume potentials. ACM Transactions on Mathematical Software 43(2):1–27. https://doi.org/10.1145/2898349
12. Michiels B, Fostier J, Bogaert I, De Zutter D (2015) Full-wave simulations of electromagnetic scattering problems with billions of unknowns. IEEE Trans Antennas Propag 63(2):796–799. https://doi.org/10.1109/TAP.2014.2380438
13. Adelman R, Gumerov NA, Duraiswami R (2017) FMM/GPU-accelerated boundary element method for computational magnetics and electrostatics. IEEE Trans Magn 53(12):1–11. https://doi.org/10.1109/TMAG.2017.2725951
14. Erhart K, Divo E, Kassab AJ (2006) A parallel domain decomposition boundary element method approach for the solution of large-scale transient heat conduction problems. Eng Anal Boundary Elem 30(7):553–563. https://doi.org/10.1016/j.enganabound.2005.12.004
15. Peng Z, Shao Y, Gao H, Wang S, Lin S (2017) High-fidelity, high-performance computational algorithms for intrasystem electromagnetic interference analysis of IC and electronics. IEEE Transactions on Components, Packaging and Manufacturing Technology 7:653–668. https://doi.org/10.1109/TCPMT.2016.2636296
16. Adelman R (2021) Highly parallel boundary element method for solving extremely large, wide-area power-line models. The International Journal of High Performance Computing Applications 35(2):136–153. https://doi.org/10.1177/1094342020976783
17. Aliyeva N, Sas HS, Okan BS (2021) Recent developments on the overmolding process for the fabrication of thermoset and thermoplastic composites by the integration of nano/micron-scale reinforcements. Compos A Appl Sci Manuf 149:106525. https://doi.org/10.1016/j.compama.2021.106525
18. Sánchez R, Martínez A, Mercado D, Carbonel A, Aisa J (2021) Rapid heating injection moulding: an experimental surface temperature study. Polym Testing 93:106928. https://doi.org/10.1016/j.polymertesting.2020.106928
19. Liu Y, Gehde M (2015) Evaluation of heat transfer coefficient between polymer and cavity wall for improving cooling and crystallinity results in injection moulding simulation. Appl Therm Eng 80:238–246. https://doi.org/10.1016/j.applthermaleng.2015.01.064
20. Park EM, Kim SK (2019) Effects of mold heat transfer coefficient on numerical simulation of injection molding. Trans Korean Soc Mech Eng B 43(3):201
21. Zhu L, Min L, Li X, Zhai Z, Drummer D, Jiang B (2019) Effects of process conditions on the heat transfer coefficient at the polymer-mold interface and tensile strength of thin-wall injection molding parts. J Polym Eng 39(5):493–500. https://doi.org/10.1016/j.jopoleng.2018-0369
22. Van der Vorst HA (1992) BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput 13(2):631–644. https://doi.org/10.1137/0913035
23. Kennedy P, Zheng R (2013) Flow analysis of injection molds. Carl Hansen Verlag GmbH Co KG, Germany

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.