Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 371 (2020) 113286

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Modeling, simulation and machine learning for rapid process control of multiphase flowing foods

D.H. Kim[a], T.I. Zohdi[a,*], R.P. Singh[b]

[a] Department of Mechanical Engineering, 6195 Etcheverry Hall, University of California, Berkeley, CA, 94720-1740, USA
[b] Department of Biological and Agricultural Engineering, University of California, Davis, CA, USA

Available online xxxx

Abstract

Across many modern industries, as technologies have matured, the use of more complex processes involving multiphase materials has increased. In the food industry, multiphase fluids are now relatively wide-spread, in particular, because of the desire to have faster throughput for large-scale food production. In many cases involving transport, such materials consist of a fluidized binder material with embedded particles. As one increases the volume fraction of particles, a corresponding increase in effective overall viscosity occurs. Often, during the process, the material must be heated, for example, to ensure food safety, induce pasteurization, sterilization, etc. *For real-time control, this requires rapidly computable models to guide thermal processing, for example by applied electrical induction.* In the present analysis, models are developed for the required heating field (electrically induced) and pressure gradient needed in a pipe to heat a multiphase material to a target temperature and to transport the material with a prescribed flow rate.

© 2020 Elsevier B.V. All rights reserved.

Keywords: Multiphase foods; Flow; Thermal control; Machine learning

1. Introduction

Across many industries, new types of particle-laden materials are being developed and utilized. In the development of such materials, the basic philosophy is to select material combinations to produce desired aggregate responses upon deposition onto a substrate or into a mold. Oftentimes, such materials start in a fluidized form comprised of particles in a solvent or fluidized binder, forming a viscous slurry. However, because of the increasing demands for faster throughput and industrial-scale production of complex particle-laden materials, the determination of accurate pumping pressures is critical to move such fluids through delivery piping systems (Fig. 1).

There have been monumental leaps in manufacturing technologies across many industries. These technologies have the potential to drastically improve the precision in food processing efficiency, food quality, and safety [1]. Some approaches are methodical and systematic, while some are ad-hoc and haphazard. The purpose of this paper is to explore modeling and simulation themes associated with multiphase fluid flow and thermal food processing. Many foods and beverage manufacturers control their continuous fluid processes with PID algorithms, using data from downstream sensors to adjust flow and heating parameters, for example in pasteurization or sterilization processes.

[*] Corresponding author.
E-mail address: zohdi@berkeley.edu (T.I. Zohdi).

https://doi.org/10.1016/j.cma.2020.113286
0045-7825/© 2020 Elsevier B.V. All rights reserved.

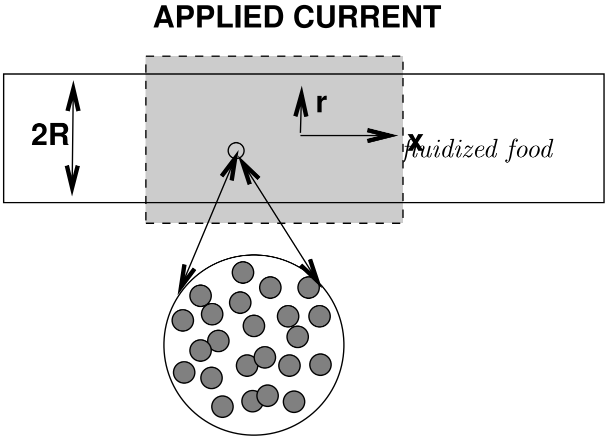


Fig. 1. Flow of a particle-laden fluid through a pipe in the presence of an applied current (heating).

As food and beverage manufacturing looks to optimize production efficiencies and energy productivity, machine learning, and other data science tools offer a chance to improve precision and predictive capabilities for real-time process optimization. In particular, new foods, such as plant-based meats, such as Beyond Meat (https://www.beyondmeat.com/) and Impossible Burger (https://impossiblefoods.com/) present new opportunities and challenges. However, for such procedures to be successful, rapidly computable models are needed to drive these technologies.

2. Technological approach

The main objective of this work is to develop a relatively simple model for the pressure gradient needed in a pipe and the reduction of a food channel width by fouling, while particle-laden fluids moving through the channel, as a function of (1) the volume fraction of added particles, (2) the pipe radius, (3) the volumetric flow rate, (4) the fluid-induced intensity of the shear stress at the pipe wall and (5) the multiphase fluid viscosity. Things overlaid on this are the induced thermal fields and associated thermal dependency of the materials in the system. This type of physical system has become increasingly more important to the 3D printing industry as well, which is attempting to rapidly print complex electrical inks (“e-inks”) of multiphase extruded materials, where the embedded particles endow the cured printed materials with overall (mechanical, electrical, thermal, magnetic, etc.) properties that the pure solvent (particle-free ink) alone does not possess. *This paper intends to adapt and to do further this analysis for food production methods.*

An overall objective of the analysis is to develop semi-analytical expressions that can help guide analysts who are designing manufacturing systems involving fluidized particle-laden foods. Theoretically speaking, one could attempt a large-scale CFD analysis, however, for accurate direct numerical simulation of particle-laden continua, the spatial discretization grids must be extremely fine, with several thousand numerical unknowns needed per particle length-scale. Furthermore, extremely fine time-discretization is required. Thus, for even a small system with several hundred–thousand particles, a proper discretization would require several billion numerical unknowns (see, for example, [2–6]). Although such simulations are possible in high-performance computing centers, their usefulness for rapid daily design analysis for real-time food processing and related processes is minimal. This is even more critical if the models are used to drive real-time control. *Therefore, in this paper we seek to develop simplified approaches.* This work presents reduced-order calculations to predict the radius change of the food channel by fouling, and pressure required to pump a suspension of rigid particles in a fluid through a pipe, under the assumption that the flow is uni-directional and fully developed. Heat is also applied as a function of an externally-applied electric field. It first arrives at a modification of Poiseulle flow through a pipe. The analysis assumes the suspension can be treated as a homogeneous fluid with an effective viscosity μ^*. This is a simplification, in order to develop useful and practical results, without having to resort to overly computationally-intensive numerical methods which seek results from detailed accounting of the micro-scale hydrodynamic interactions between particles in a suspension.

3. Fluid through a pipe of radius *R*

We consider fluid that is flowing through an idealized pipe with a circular cross-section of area $A = \pi R^2$.

$$v = v_{max}\left(1 - \left(\frac{r}{R}\right)^q\right), \tag{3.1}$$

where q is now considered a variable. For fully developed laminar flow, $q = 2$, while for increasing q one characterizes progressively turbulent flow ($q \geq 2$). Also, assuming that the overall flow rate is assumed constant Q_0, one can show that

$$v_{max} = \frac{Q_0(q+2)}{\pi R^2 q}, \tag{3.2}$$

and

$$\tau_w = \frac{\mu^* Q_0(q+2)}{\pi R^3}, \tag{3.3}$$

where μ^* represents effective viscosity of multiphase fluid.

We have the following observations: (a) Increasing μ^* or Q_o increases the stress at the wall (τ_w) and (b) Decreasing R increases the stress at the wall (τ_w).

Also, by performing a force balance in the positive x-direction, we obtain

$$-\frac{\Delta P}{\Delta x} = -\frac{\partial P}{\partial x} = \underbrace{\frac{2\mu^*(q+2)}{\pi R^4}}_{C} Q_o \overset{\text{def}}{=} C Q_o. \tag{3.4}$$

This expression allows us to correlate the pressure applied to a volume of particle-laden to allow it to move as a constant flow rate. If we fix the flow rate Q_o, the multiplier C identifies the pressure gradient needed to achieve a flow rate Q_o.

As Reynolds number increases, the velocity profile will change from a quadratic ($q = 2$) to a more blunted profile ($q > 2$). The effect of a changing profile is described by representing q as follows.

$$q = \frac{1}{2}\left((\gamma^* + c_2) + \sqrt{(\gamma^* + c_2)^2 + 8\gamma^*}\right), \tag{3.5}$$

where

$$\gamma^* = \frac{2c_1 Q_0 \rho^*}{\pi R \mu^*}. \tag{3.6}$$

ρ^* represents effective density of multiphase fluid, respectively. c_1 and c_2 are constants where $0 \leq c_1 \ll 1$ and $c_2 \approx 2$. For laminar flow ($q = 2$), $c_1 = 0$ and $c_2 = 2$. See [7] for more detail. In the remaining analysis, we will consider turbulent flow ($q > 2$) of the particle-laden fluid, in which Reynolds number is greater than 4000 [8]. Also, we will assume that the particles are not elongated and that they are well distributed within the base fluid.

4. Induced thermal fields via Joule heating

The heating process in food and beverage production facilities is typically accomplished with steam-heated heat exchangers. Most commonly, low-pressure steam heats a plate and frame or tubular heat exchanger. For thermally sensitive food products, typical industry practice is to heat food in long tubular heat exchangers using flowing hot water, heated via direct steam injection, maintained at a single static temperature. Both steam-heated (single loop) and hot water heated (double-loop) heating processes are notoriously slow to reach processing temperature or adapt the temperature to changing processing conditions. In particular, industrial-scale double-loop systems can take up to 10 min to create a response in end-target heating. Steam-powered heat exchange, in either single loop or double loop systems, is optimized for production in large facilities where processing conditions remain static and unchanging. These systems are optimized for traditional manufacturing operations and do not have the capabilities needed for the future. As this manufacturing sector moves towards shorter production runs and expanding the number of products produced in a single production line, there is an opportunity to find new production efficiencies and increase energy production to boost yields and facility operational effective efficiency (OEE). Joule heating (also known as Ohmic heating) generates heat by passing an electrical current through food which has electrical resistance [9]. Heat is generated rapidly and uniformly in the liquid matrix as well as in particulates, producing a higher quality sterile and aseptic product [10]. This heating method is best for foods containing particulates in a weak salt-containing medium due to their high resistance [11]. Joule heating adapts well to computer control and allows for instantaneous control, even though it is currently limited by the sensitivity and response time of a downstream thermocouple. It is for this reason that we select induction heating as a model process. From the first law of thermodynamics, we have the following description of Watts per unit volume:

$$\rho^* C^* \dot{\theta} = H - S \tag{4.1}$$

where θ is the temperature, C^* is specific heat capacity, H represents heating and S represents sinks. For the heating

$$H = a\frac{J^2}{\sigma^*}, \tag{4.2}$$

where a is the absorption coefficient, and S represents all of the losses (conductive, convective, radiative, refrigeration). σ^* is the effective electrical conductivity, and J is electrical current for Joule heating. Discretizing and solving yields

$$\theta(t+\Delta t) = \theta(t) + \frac{\Delta t}{\rho^* C^*}(H - S) = \theta(t) + \frac{\Delta t}{\rho^* C^*}(a\frac{J^2}{\sigma^*} - S(t)). \tag{4.3}$$

If at a given time t, one can solve for the necessary $J(t)$ when $\theta(t+\Delta t) = \theta^*$:

$$J(t) = \sqrt{\frac{\sigma^*}{a}\left(\rho^* C^* \frac{\theta^* - \theta(t)}{\Delta t} + S(t)\right)}. \tag{4.4}$$

5. Models for effective properties of particle-laden fluids

A key component of the analysis requires the characterization of the effective properties of a particle-laden fluid as a function of the volume fraction of particles and the baseline (interstitial) fluid properties. The density of the particle-laden fluid is actually an "effective density" since it actually is a mixture of materials (particles and interstitial fluid). Effective properties are defined through volume averages. For example, the effective density of the mixture is

$$\rho^* \overset{\text{def}}{=} \langle \rho(\boldsymbol{x}) \rangle_V \overset{\text{def}}{=} \frac{1}{V}\int_V \rho(\boldsymbol{x})\, dV = \frac{1}{V}\left(\int_{V_f} \rho_f\, dV + \int_{V_p} \rho_p\, dV\right) = \nu_f \rho_f + \nu_p \rho_p \tag{5.1}$$

where ν_f and ν_p are the volume fractions of the fluid and particles, respectively. The volume fractions have to sum to unity: $\nu_f + \nu_p = 1 \Rightarrow \nu_f = 1 - \nu_p$. Similar approaches can be used to calculate various types of properties, such as effective viscosity. However, to calculate it is somewhat more complicated since it requires one to estimate the interaction between the constituents. There are a number of models that provide expressions for the effective viscosity of the fluid containing particles. One of the first models for the effective viscosity of such fluids was developed in 1906 by [12]. It reads as

$$\mu^* = \mu_f(1 + 2.5\nu_p), \tag{5.2}$$

where μ^* is the effective viscosity, μ_f is the viscosity of the fluid and ν_p is the volume fraction of particles. This expression is accurate only for low volume fractions of particles. A more accurate approximation, *in fact, a strict, rigorous, lower bound* can be derived from the well-known Hashin and Shtrikman bounds [13–15] in solid mechanics. Specifically, for linearized elasticity applications, for isotropic materials with isotropic effective (mechanical) responses, the Hashin–Shtrikman bounds (for a two-phase material) are as follows for the effective bulk modulus (κ^*)

$$\kappa^{*,-} \overset{\text{def}}{=} \kappa_1 + \frac{\nu_2}{\frac{1}{\kappa_2 - \kappa_1} + \frac{3(1-\nu_2)}{3\kappa_1 + 4\mu_1}} \le \kappa^* \le \kappa_2 + \frac{1 - \nu_2}{\frac{1}{\kappa_1 - \kappa_2} + \frac{3\nu_2}{3\kappa_2 + 4\mu_2}} \overset{\text{def}}{=} \kappa^{*,+} \tag{5.3}$$

and for the effective shear modulus (G^*)

$$G^{*,-} \overset{\text{def}}{=} G_1 + \frac{\nu_2}{\frac{1}{G_2-G_1} + \frac{6(1-\nu_2)(\kappa_1+2G_1)}{5G_1(3\kappa_1+4G_1)}} \leq G^* \leq G_2 + \frac{(1-\nu_2)}{\frac{1}{G_1-G_2} + \frac{6\nu_2(\kappa_2+2G_2)}{5G_2(3\kappa_2+4G_2)}} \overset{\text{def}}{=} G^{*,+}, \tag{5.4}$$

where κ_1 (usually the matrix material) and κ_2 (usually the particulate material) are the bulk moduli and G_1 and G_2 are the shear moduli of the respective phases ($\kappa_2 \geq \kappa_1$ and $G_2 \geq G_1$), and where ν_2 is the second phase volume fraction. Such bounds are the tightest possible on isotropic effective responses, with isotropic two-phase microstructures, where only the volume fractions and phase contrasts of the constituents are known (see [15] for a discussion on the optimality of such bounds). Note that no geometric or statistical information is required for the bounds. For an authoritative review of the general theory of random heterogeneous media see [16]. One can take the limit of the particle phase becoming rigid, i.e. the bulk and shear moduli tending towards infinity, $\kappa_2 = \kappa_p \to \infty$ and $G_2 = \mu_p \to \infty$, signifying that the particles are much stiffer than the interstitial fluid, while simultaneously specifying that the interstitial fluid is incompressible, i.e. $\kappa_1/G_1 = \kappa_f/\mu_f \to \infty$ with G_1 being finite. This yields,

$$\mu^* \geq \mu^{*,-} = \mu_f \left(1 + 2.5\frac{\nu_p}{1-\nu_p}\right). \tag{5.5}$$

Eq. (5.5) represents the tightest known lower bound on the effective viscosity of a two-phase material comprised of rigid particles in a surrounding incompressible fluid. The bound recaptures the Einstein result in the $\nu_p \to 0$ limit, but is a rigorous lower bound at significant ν_p. *This rigorous lower bound* is extremely accurate up to approximately 20% volume fraction, which is sufficient for most applications of interest. These bounds have been tested in the numerical analysis literature repeatedly, for example against direct Finite Element calculations found in [6]. We refer the reader to [17] for a more in-depth analysis of the effective viscosity of particle-laden fluids. Refer to [18] for the analysis of the proper application of the non-interaction and the “dilute limit” approximations, and for detailed discussions on the isotropic and anisotropic viscosity of suspensions containing particles of diverse shapes and orientations. It is important to emphasize that [17] is accurate for up to 25%–30% in the case of spherical particles. Furthermore, [17] covers other shapes, including, importantly, mixtures of diverse shapes. Of course, one can employ formulas such as in [17] for more accuracy, however, because the Hashin–Strikman expression is a strict lower bound, $\mu^{*,-} \leq \mu^*$, we consequently generate a strict lower bound for the pressure gradient

$$-\frac{\partial P}{\partial x} \geq \underbrace{\frac{2\mu^{*,-}(q+2)}{\pi R^4}}_{C^-} Q_o \overset{\text{def}}{=} C^- Q_o. \tag{5.6}$$

6. Approximate effective thermal properties

For illustration purposes, in this model problem, a relatively simplistic definition of the heat capacity is defined through the stored energy at a point, $w = C(\theta - \theta_o)$, where θ_o is a reference temperature, for example, $\theta_o = 0$ degrees Kelvin. In this case, the effective heat capacity for a small body is given by

$$\begin{aligned}\langle w\rangle_V = \langle C\theta\rangle_V &= C^*\langle\theta\rangle_V \overset{\text{def}}{=} \frac{1}{V}\int_V C(\boldsymbol{x})\theta(\boldsymbol{x})\,dV \\ &= \frac{1}{V}\left(\int_{V_f} C_f\theta(\boldsymbol{x})\,dV + \int_{V_p} C_p\theta(\boldsymbol{x})\,dV\right) \\ &\approx (\nu_f C_f + \nu_p C_p)\theta \Rightarrow C^* = (\nu_f C_f + \nu_p C_p),\end{aligned} \tag{6.1}$$

provided θ is uniform in the (small body).

The effective density of a mixture for two-phase materials can directly be determined by

$$m = \rho^* V = (\nu_f\rho_f + \nu_p\rho_p)V, \tag{6.2}$$

while the effective thermal mass is $mC = \rho^* C^* V$.

6.1. Thermal material behavior

For illustration purposes, in this model problem, the primary properties which change by temperature are viscosity, electrical conductivity, and thermal conductivity. This could be expressed as follows.

$$\mu_f = \mu_{fo} e^{-s_1 \frac{\theta-\theta_o}{\theta_o}}, \tag{6.3}$$

and

$$\sigma_f = \sigma_{fo} e^{-s_2 \frac{\theta-\theta_o}{\theta_o}}, \tag{6.4}$$

and

$$\sigma_p = \sigma_{po} e^{-s_3 \frac{\theta-\theta_o}{\theta_o}}. \tag{6.5}$$

and

$$k_f = k_{fo} e^{-s_4 \frac{\theta-\theta_o}{\theta_o}}, \tag{6.6}$$

and

$$k_p = k_{po} e^{-s_5 \frac{\theta-\theta_o}{\theta_o}}, \tag{6.7}$$

where μ, σ, and k represent viscosity, electrical conductivity, and thermal conductivity, respectively. Subscript f represents fluid carrying particles, and subscript p represents particles. s_1, s_2, s_3, s_4, and s_5 represent thermal softening parameters.

7. Fouling model

Fouling is a cost-increasing problem for a variety of industries, including aerospace, petrochemicals and especially food [19–22]. The reduction of a food channel width, caused by deposition of thermally modified proteins and other food components, is a major issue in food processing since cleaning or removal of such deposits is crucial for quality and safety issues [23]. Also, fouling of heat exchangers (undesirable depositions on channel surfaces) is directly related to the quality of processed foods and processing cost, because it may require more energy consumption for the pressure gradients and Joule heating. In this section, we adopted the most common fouling model of a heat exchanger by Ebert and Panchal [24], to predict the deposition thickness of the food channel, as shown in Eq. (7.1):

$$\frac{dR_f}{dt} = \underbrace{c_1 Re^{-0.66} Pr^{-0.33} e^{-\frac{E_a}{RT_f}}}_{deposition} - \underbrace{c_2 \tau_w}_{suppression}, \tag{7.1}$$

where R_f is the fouling resistance (m^2 K/W), Re is Reynolds number, Pr is the Prandtl number, E_a is the activation energy (J/mol), T_f is the film temperature (K) and τ_w is the shear at the deposit surface (Pa). c_1 and c_2 are fouling parameters determined by regression of experimental data. For flow in a pipe or tube, Reynolds number is generally defined as $Re = \frac{\rho u D}{\mu}$, where ρ is the density of the fluid, u is the mean velocity of the fluid, D is the hydraulic diameter of the pipe and μ is the dynamic viscosity of the fluid. Also, the Prandtl number is defined as $Pr = \frac{C_p \mu}{k}$, where C_p is the specific heat capacity, μ is the dynamic viscosity of the fluid an k is the thermal conductivity. From fouling resistance obtained by Eq. (7.1), we can get the deposition thickness, as shown in Eq. (7.2)

$$d_f = R_f \lambda_f, \tag{7.2}$$

where d_f is the average thickness of the fouling (m), and λ_f is the thermal conductivity of the fouling (W/m K).

8. Numerical experiments

The main object of this section is to show how the pressure gradients, the electrical current needed to heat the multiphase food, and Reynolds number are dependent on the particle volume fraction in the multiphase fluid. We plotted the pressure gradient as a function of ν_p, with the following parameters[1]: (a) viscosity, $\mu_f = 0.01$ Pa-s,

[1] For reference, the viscosity of water is $\mu_f = 0.001$ Pa-s and for honey, $\mu_f = 1$ Pa-s.

(b) fluid density: $\rho_f = 1000$ kg/m^3, (c) particle density: $\rho_p = 2000$ kg/m^3, (d) flow rate: $Q_o = 0.00001\,\mathrm{m}^3/\mathrm{s}$, (e) thermal sensitivity: $s_1 = s_2 = s_3 = 1$, and (f) pipe radius: $R = 0.01$ m. The goal was to raise the temperature from 300 K to 400 K in 1 s. The plots are shown in Fig. 2. The pressure gradient steadily increases with particle volume fraction. Due to the increase in the particle volume fraction, the viscosity increases, thus Reynolds number decreases (already quite small). The point of this example was not to illustrate all the encompassing parameter set, but simply to show the explicit dependency of the pressure gradient on the presence of secondary particles.

9. Simulation algorithm

The objective of the simulation is to analyze

- the thermal behavior of food while processing,
- pressure gradient needed in a pipe to heat a multiphase material,
- the change of the radius (caused by fouling) of food channel by time.

First, we consider electric current for heating the multiphase material, as a function of the time as follows, having a factor α which represents the response rate

$$J = J_0(1 - e^{-\alpha t})sin(2\pi t). \tag{9.1}$$

We obtain the temperature of the next time step (Eq. (4.3)) using a forward Euler scheme. The thermal loss term S can account for any kind of thermal loss. In this example, we assume that the dominant mode of thermal loss to the environment is convection and that the temperature is essentially uniform within a cross-section of the pipe. Convective losses are often modeled by:

$$S = hA(\theta(t) - \theta_a),$$

where h is the convection coefficient, A is the area over which conduction occurs, $\theta(t)$ is the temperature of the object being modeled at a given time, t and θ_a is the temperature of the ambient environment. Since all of our other terms are per unit volume of the slurry, we must alter this relationship to have compatible units. We assume that we are considering an arbitrary length Δx of the pipe. The surface area A will be $2\pi R\Delta x$, and the total volume will be $\pi R^2\Delta x$. Dividing A by volume yields $2/R$. As such, the convective losses per unit volume will be:

$$S = 2h(\theta(t) - \theta_a)/R. \tag{9.2}$$

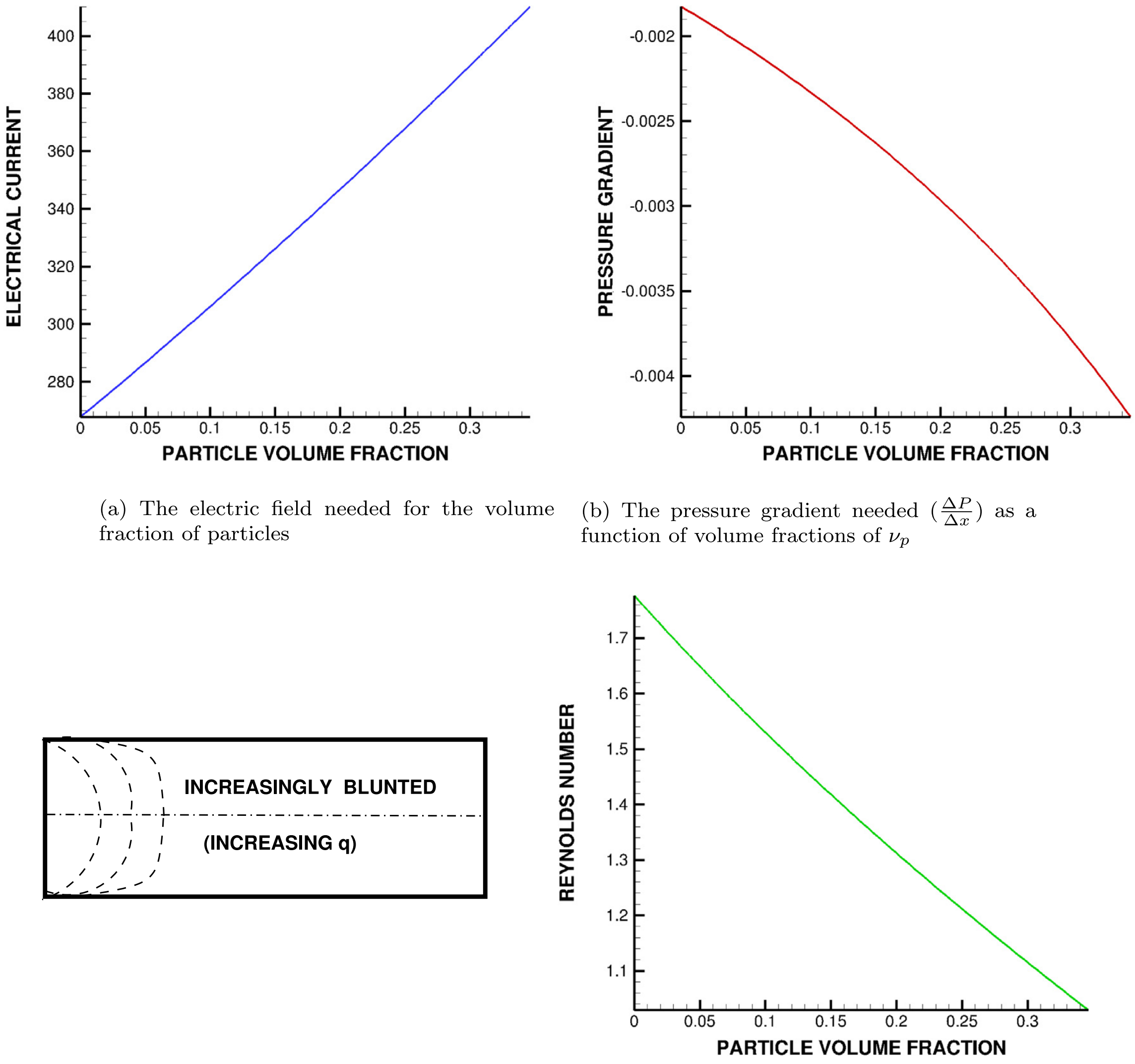
Note that $S(t)$ models the transfer of heat *out of the system*, so a positive value of S represents a loss of energy. As temperature changes, primary material properties of the mixture change(σ^*, k^* and μ^*), because those properties of fluid and particle change (Eqs. (6.3), (6.4), (6.5), (6.6), and (6.7)).

The Hashin–Shtrikman bounds for σ^* (electrical conductivity) are as follows, and σ_f and σ_p were calculated based on Eqs. (6.4), and (6.5)

$$\underbrace{\sigma_f + \frac{v_p}{\frac{1}{\sigma_p - \sigma_f} + \frac{1-v_p}{3\sigma_f}}}_{\sigma^{*,-}} \leq \sigma^* \leq \underbrace{\sigma_p + \frac{1 - v_p}{\frac{1}{\sigma_f - \sigma_p} + \frac{v_p}{3\sigma_p}}}_{\sigma^{*,+}}, \tag{9.3}$$

Similarly, the Hashin–Shtrikman bounds for k^* (thermal conductivity) are as follows, and k_f and k_p were calculated based on the Eqs. (6.6), and (6.7)

$$\underbrace{k_f + \frac{v_p}{\frac{1}{k_p - k_f} + \frac{1-v_p}{3k_f}}}_{k^{*,-}} \leq k^* \leq \underbrace{k_p + \frac{1 - v_p}{\frac{1}{k_f - k_p} + \frac{v_p}{3k_p}}}_{k^{*,+}}, \tag{9.4}$$



(a) The electric field needed for the volume fraction of particles

(b) The pressure gradient needed ($\frac{\Delta P}{\Delta x}$) as a function of volume fractions of ν_p

(c) Progressive blunting of the velocity profile with increasing q

(d) The resulting Reynolds number

Fig. 2. Electrical current, pressure gradient, and Reynolds number by particle volume fraction.

In the simulation, we take the average of the lower bound and upper bound to approximate σ^* and k^*

$$\sigma^* = \frac{\sigma^{*,-} + \sigma^{*,+}}{2}, \text{ and } k^* = \frac{k^{*,-} + k^{*,+}}{2}. \tag{9.5}$$

Also, widely used effective viscosity estimate is that of Oliver&Ward [25] which is in much better agreement (than Eq. (5.5)) with experimental data up to $v_p = 0.30$ (see [17,18] for extensive reviews). It reads as

$$\mu^* \approx \mu^{*e} = \frac{\mu_f}{1 - 2.5v_p}. \tag{9.6}$$

We could observe that the ratio of the rigorous lower bound to the Oliver&Ward [25] estimate is always less than unity for finite volume fraction [26]. In the remaining analysis, we will employ Oliver&Ward estimate for the effective viscosity of multiphase flow.

Updating σ^* affects integration of the next time step for obtaining the temperature, and updating μ^* and k^* causes the change of channel radius (Eqs. (7.1)–(7.2)) and the pressure gradient (Eq. (3.4)), because they are function of

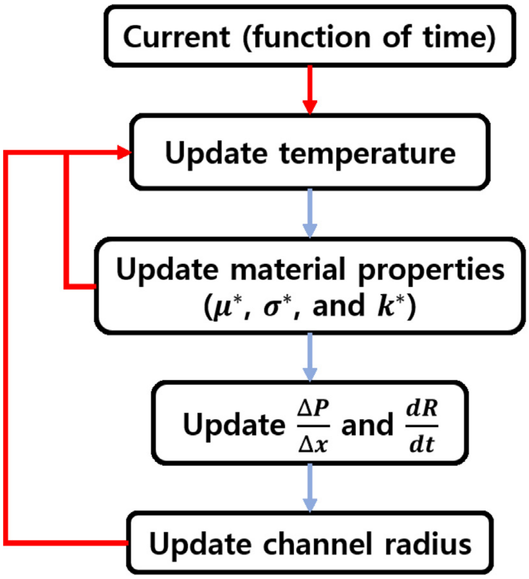

Current (function of time)
Update temperature
Update material properties (μ^*, σ^*, and k^*)
Update $\frac{\Delta P}{\Delta x}$ and $\frac{dR}{dt}$
Update channel radius

Fig. 3. Overall flowchart of the simulation.

μ^* and k^*. Based on this, we can obtain the new channel radius for the next step, which will be used for the calculation of the temperature, pressure gradient of the next time step. This iteration is done until the temperature becomes greater than or equal to the desired temperature of the food product. The overall flowchart is shown in Fig. 3, and the simulation parameters are shown in Table 1.

10. Genetic algorithm for optimization

Genetic algorithm (GA) is a type of nonderivative search and optimization tool, which works differently from classical search and gradient-optimization methods. Because of its broad applicability, ease of use, and global perspective, GA has been increasingly applied to various search and optimization problems in the recent past [27,28]. Our final objective is to find the optimal parameters we could control for food processing using GA. In order to do that, we should set the cost function we want to minimize to find a way to process food in a reasonable and efficient way. In this case, we try to minimize the fouling of the food channel after processing (because it gets larger and larger as the food flows), as well as minimize the processing time and the total energy consumption for the process. We applied a penalty if either the food processing time or energy consumption exceeded a preset limit.

The cost function we want to minimize is:

$$\boxed{\Pi = w_1\,|\eta|^2 + \hat{w}_2 \left|\frac{T - T_{limit}}{T_{limit}}\right|^2 + \hat{w}_3 \left|\frac{E_{final} - E_{limit}}{E_{limit}}\right|^2}$$

where

1. T: Total processing time,
2. T_{limit}: Limit of the total processing time,
3. E_{final}: Energy consumption for food processing,
4. E_{limit}: Limit of the total energy consumption for processing,
5. if $T \leq T_{limit}$, then $\hat{w}_2 = 0$,
6. if $T > T_{limit}$, then $\hat{w}_2 = w_2$,
7. if $E_{final} \leq E_{limit}$, then $\hat{w}_3 = 0$,
8. if $E_{final} > E_{limit}$, then $\hat{w}_3 = w_3$.

Also, η represents the fouling rate of the channel radius, as shown in Eq. (10.1), and E_{final} is calculated by numerically integrating the power by time during processing time.

$$\eta = \frac{R_0 - R}{R_0} \tag{10.1}$$

Here the design variables we want to optimize are $\boldsymbol{\Lambda} = \{\alpha, v_p, \sigma_{p0}, \rho_p, C_p, k_{p0}\}$, and their constrained ranges are $\alpha^{(-)} \leq \alpha \leq \alpha^{(+)}$, $v_p^{(-)} \leq v_p \leq v_p^{(+)}$, $\sigma_{p0}^{(-)} \leq \sigma_{p0} \leq \sigma_{p0}^{(+)}$, $\rho_p^{(-)} \leq \rho_p \leq \rho_p^{(+)}$, $C_p^{(-)} \leq C_p \leq C_p^{(+)}$ and $k_{p0}^{(-)} \leq k_{p0} \leq k_{p0}^{(+)}$.

As a consequence of the character of the objective function, we can use the following genetic algorithm:

```
Algorithm 1: Genetic Algorithm
Initialization: Randomly generate a population of $S$ initial strings, $\boldsymbol{\Lambda}^i$ $(i = 1, 2, 3, ..., S)$.
$\longrightarrow \boldsymbol{\Lambda}^i \triangleq \{\lambda_1^i, \lambda_2^i, \lambda_3^i, ..., \lambda_N^i\}$

while $min(\Pi) \geq Tolerance$ do
    STEP 1: Compute the performance of each genetic string $\Pi(\boldsymbol{\Lambda}^i)$ $(i = 1, 2, 3, ..., S)$, and rank those strings
    based on the performance values.

    STEP 2: Mate the best performing $P$ parent strings to generate $C$ offspring strings.
    $\boldsymbol{\Lambda}^{NEW} \triangleq \boldsymbol{\Phi} \odot \boldsymbol{\Lambda}^{OLD_1} + (\mathbf{1} - \boldsymbol{\Phi}) \odot \boldsymbol{\Lambda}^{OLD_2}$
    , where $\boldsymbol{\Phi} = \{\phi_1, \phi_2, \phi_3, ..., \phi_N\}$, and $0 \leq \phi_k \leq 1$ $(k = 1, 2, ..., N)$.
    $\odot$ represents component-wise multiplication.

    STEP 3: Replace the worst performing $C$ strings in the old genetic strings with the new child strings obtained
    in STEP 2.

    STEP 4:
    if Keeping parents then
        Keep the old $P$ parent strings
        Generate $S - P - C$ new strings
    else
        Remove the old $P$ parent strings
        Generate $S - C$ new strings
    end
end

(Optional) Employ gradient-based methods afterwards in the local minima, if the neighbor of the obtained
optimal point is smooth enough.
```

We remark that the definition of **fitness** of a genetic string in this algorithm indicates the value of the objective function. In other words, the most fit genetic string is simply the one with the smallest objective function. STEPS 1–6, which are associated with the genetic part of the overall algorithm, attempt to collect multiple local minima.[2]

By observing Fig. 4 one sees that if the objective functions are highly nonconvex, there exists a strong possibility that the inferior offspring will replace superior parents. Therefore, retaining the top parents is not only less computationally expensive, since these designs do not have to be reevaluated, but it is also theoretically superior. The minimization of the cost function is guaranteed to be monotone, if the top parents are retained, i.e. $\Pi(\boldsymbol{\Lambda}^{opt,I}) \geq \Pi(\boldsymbol{\Lambda}^{opt,I+1})$, where $\boldsymbol{\Lambda}^{opt,I+1}$ and $\boldsymbol{\Lambda}^{opt,I}$ are the best genetic strings from generations $I + 1$ and I respectively.

[2] It is remarked that if the function Φ is allowed to be greater than unity, one can consider the resulting convex combination (offspring) as a “mutation”.

Table 1
Simulation parameters.

Symbol	Units	Value	Description
μ_{f0}	Pa s	.005	Reference temp. viscosity of fluid (Phase 1)
σ_{f0}	Ω^{-1} m^{-1}	0.617	Reference temp. electrical conductivity of fluid (Phase 1)
ρ_f	kg/m^3	2000	Density of fluid (Phase 1)
k_{f0}	W/(m K)	0.45	Reference temp. heat conductivity of fluid (Phase 1)
C_f	J/(kg K)	1600	Specific heat capacity of fluid (Phase 1)
R_0	m	1e−1	Initial channel radius
Q_0	m^3/s	8e−3, 10e−3, 12e−3	Flow rate
J_0	A/m^2	2e4	Base electric current
h	W/(m^2 K)	10	Convective heat transfer coefficient
c_1	(m^2 K)/(W s)	5e−3	Fouling parameter
c_2	(m^2 K)/(W Pa s)	8e−8	Fouling parameter
θ_0	K	300	Initial slurry temperature
θ_a	K	300	Ambient temperature
Δt	s	0.001	Time step size
a	Unitless	0.8	Absorption coefficient
T_{limit}	s	2.5	Limit of the total processing time
E_{limit}	MJ/m^3	600	Limit of the total energy consumption for processing
s_1, s_2, s_3	Unitless	0.1, 0.12, 0.2	Thermal softening parameters
s_4, s_5	Unitless	0.01, 0.02	Thermal softening parameters
ϕ	Unitless	0.5	Hashin–Shtrikman bound combination weight
α	unitless	$1 = \alpha^{(-)} \leq \alpha \leq \alpha^{(+)} = 10$	Current response rate
v_p	unitless	$0.05 = v_p^{(-)} \leq v_p \leq v_p^{(+)} = 0.3$	Volume fraction (Phase 2)
σ_{p0}	S/m	$0.1 = \sigma_{p0}^{(-)} \leq \sigma_{p0} \leq \sigma_{p0}^{(+)} = 2.5$	Reference temp. electrical conductivity of particles (Phase 2)
ρ_p	kg/m^3	$3000 = \rho_p^{(-)} \leq \rho_p \leq \rho_p^{(+)} = 9000$	Density of particles (Phase 2)
C_p	$J/(kg \cdot K)$	$1000 = C_p^{(-)} \leq C_p \leq C_p^{(+)} = 6000$	Specific heat capacity of particles (Phase 2)
k_{p0}	$W/(m \cdot K)$	$0.11 = k_{p0}^{(-)} \leq k_{p0} \leq k_{p0}^{(+)} = 0.52$	Reference temp. heat conductivity of particles (Phase 2)
w_1, w_2, w_3	unitless	1, 2, 3	Cost function weights
N	unitless	100	The number of genetic strings per generation
-	unitless	2	The number of offspring strings per pairs

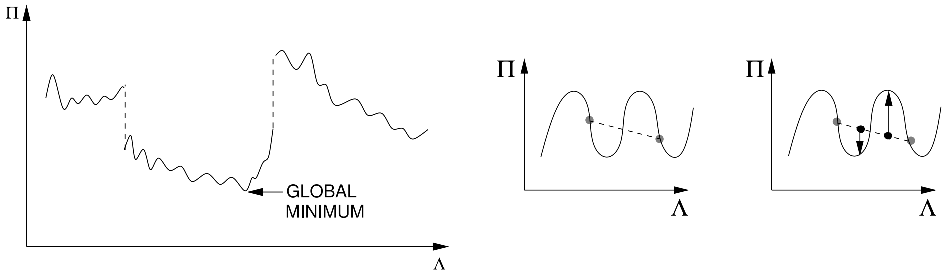
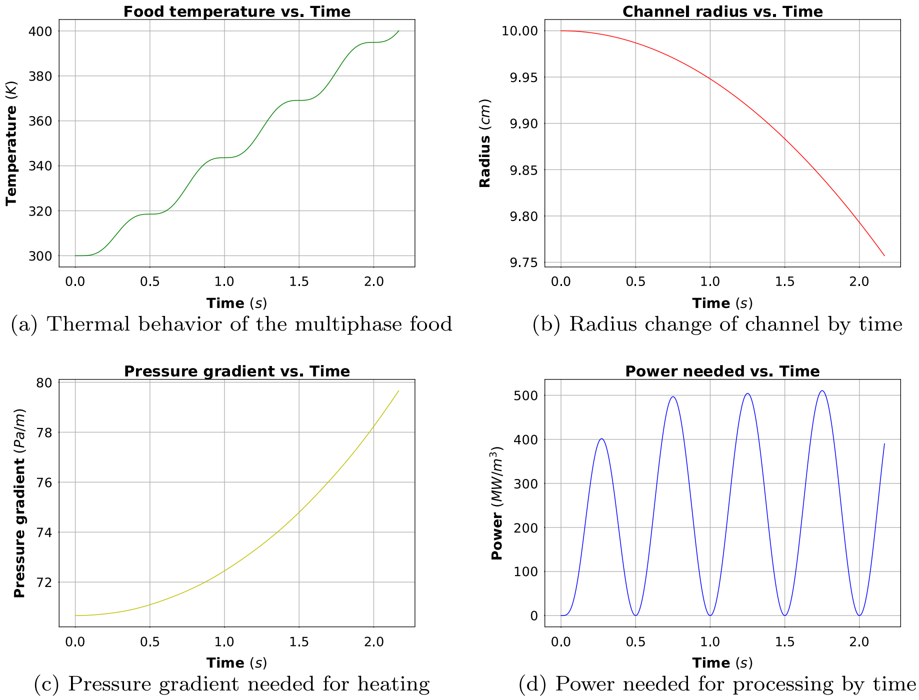


Fig. 4. LEFT: A characterization of the class of objective functions of interest. RIGHT: A loss of superior older genetic strings if the top parents are not retained.

11. Simulation results

We have developed a relatively simple model for multiphase food processing through a channel and optimized the parameters for processing with numerical simulations and the genetic algorithm. Specifically, with the parameters which were optimized using the genetic algorithm, we could obtain (a) the thermal behavior of multiphase food while processing, (b) the pressure gradient needed in a pipe to heat the multiphase material, (c) the change of the radius (by deposition fouling) of food channel by time and (d) the power needed for processing by time. The plots are shown in Figs. 5–7. Fig. 5 represents the case when $Q_0 = 0.008$ (m^3/s), Fig. 6 represents the case when $Q_0 = 0.010$ (m^3/s), and Fig. 7 represents the case when $Q_0 = 0.012$ (m^3/s).



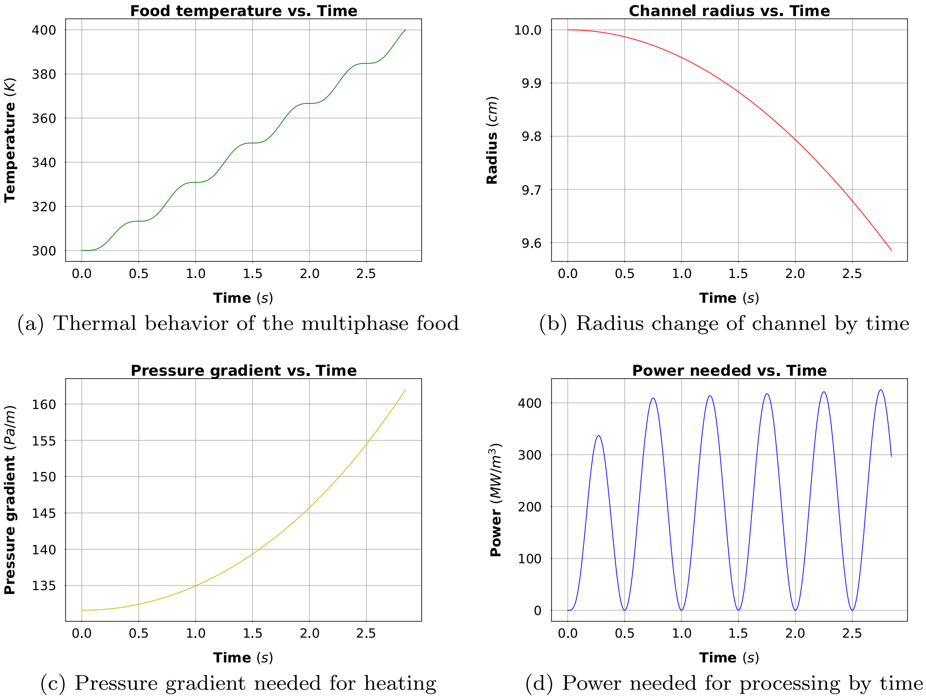
(a) Thermal behavior of the multiphase food

(b) Radius change of channel by time

(c) Pressure gradient needed for heating

(d) Power needed for processing by time

Fig. 5. CASE 1 ($Q_0 = 0.008$ (m^3/s)).



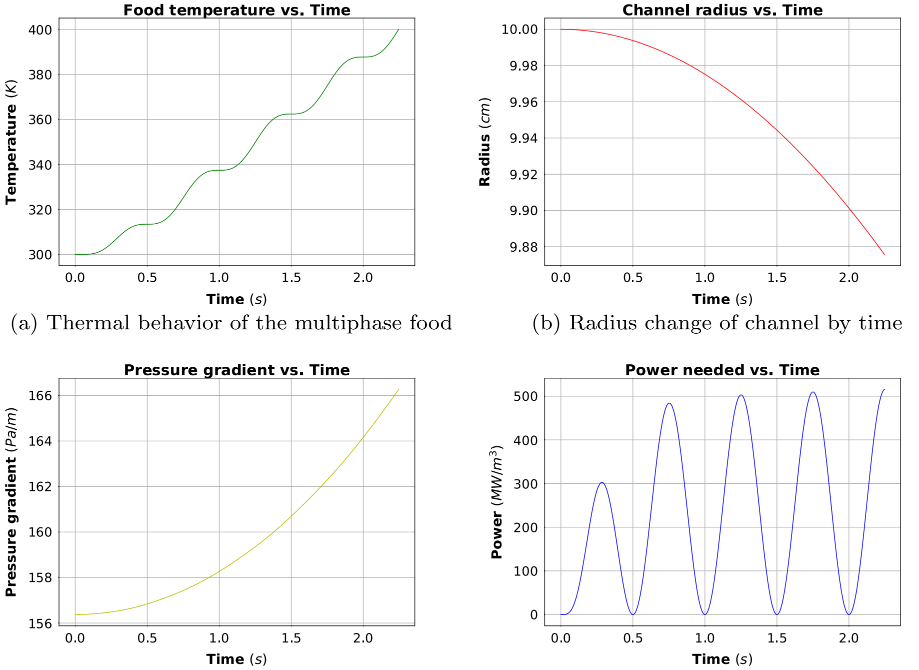
(a) Thermal behavior of the multiphase food

(b) Radius change of channel by time

(c) Pressure gradient needed for heating

(d) Power needed for processing by time

Fig. 6. CASE 2 ($Q_0 = 0.010$ (m^3/s)).



(a) Thermal behavior of the multiphase food

(b) Radius change of channel by time

(c) Pressure gradient needed for heating

(d) Power needed for processing by time

Fig. 7. CASE 3 ($Q_0 = 0.012$ (m^3/s)).

12. Prediction of deposition fouling of a channel

The unintended accumulation of the food on a channel wall while processing can cause deposition fouling. The fouling on the food delivery channel is a critical issue for processing, because it directly affects manufacturing costs [19–21]. It becomes harder to process the food precisely in a desirable way, as channel radius gets smaller and smaller. Therefore, predicting the possible changes in the radius of the food channel is important to control food quality. We applied a machine learning technique to predict the fouling rate of the food channel after final food processing time to get to the desired food temperature.

To obtain the channel fouling rate for combinations of the parameters, we still need to solve differential equations using numerical methods, as we already did in the genetic algorithm of the previous section. It might be computationally expensive when we use elaborate numerical methods (Discrete element methods, Finite element methods, etc.) and/or there are a large number of parameter sets to simulate. Even though numerical simulations are computationally expensive in general, it gives us the data that is useful for generating a predictive model.

Therefore, we applied a machine learning technique to predict the fouling rate of the channel at the final processing time with reduced computation time and desirable accuracy, when specific parameter sets are chosen while the other operation conditions (initial channel radius, electric current, ambient temperature) are fixed.

Machine learning is a subdivision of artificial intelligence, which allows computers to learn from the past data so that it could detect patterns and make predictions from noisy and complex data sets [29–33]. The ML approach deals with the design of algorithms to learn from machine-readable data and make predictions on future unknown data [34]. Also, there has been some research on generating prediction models using machine learning to solve a variety of engineering problems such as material design, computer vision, pattern recognition, and spam filtering [35–41], including those in computational mechanics [42–44].

In our approach, we used a fully connected neural network to develop the model for predicting the final fouling rate of the channel radius. The neural network is a biologically-inspired machine learning model which enables nonlinear learning process based on how neurons communicate and learn in living things. We used 9010 parameter combinations for the total data. All the parameter combinations (features) and corresponding fouling rates (labels) were normalized to have the range between 0 and 1, before training the neural network. We implemented our

Table 2
Computation time comparison.

Flow rate (m^3/s)	Training (s)	Prediction (s)	Total time (s)	Acceleration
0.008	13.74	0.0008	172.90	4.60
0.01	12.46	0.001	174.12	4.64
0.012	12.22	0.0008	171.39	4.64

Table 3
Root mean squared error of prediction.

Flow rate (m^3/s)	RMS error
0.008	0.00408
0.01	0.00318
0.012	0.00246

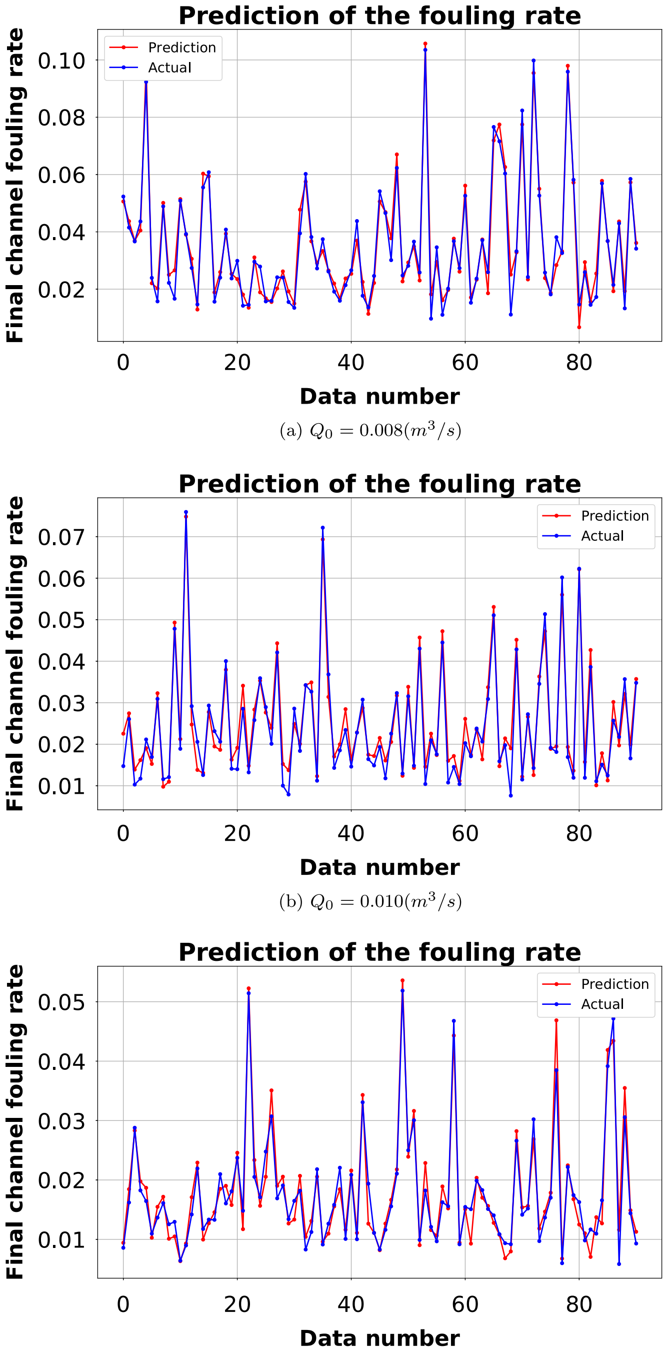
machine learning model with PyTorch (1.1.0 version), which is an efficient machine learning framework for Python and competent in both usability and speed [45]. We used batch normalization, which is one of the ways to accelerate neural network training, by reducing the internal covariance shift, allowing us to use much higher learning rates [46]. Xavier initialization was used for weight initialization in order to obtain substantially faster convergence [47]. Also, an Adam optimizer was used for the optimization, which is computationally efficient and has little memory requirements [48]. The configuration of the Machine Learning model is as follows.

- Learning rate: 0.001
- Training epochs: 100
- Batch size: 50
- The number of hidden layers: 2
- The number of nodes in hidden layers: 40, 20
- Activation function: ReLu (Rectified Linear Unit)
- Weight initialization: Xavier uniform
- Loss function: Mean Squared Error
- Optimization Method: Adam optimizer
- Division of the data: 20% for training, 79% for validation, and 1% for testing

With the trained model, we were able to desirably predict the fouling rate of the channel radius (Eq. (10.1)) for the test data set, even though we trained our model with a small portion of the total data (20% for training, 79% for validating, and 1% for testing), with considerable accuracy, as shown in Fig. 8. There are three cases with varying flow rates (Fig. 8). **Data number** represents the combinations of $\mathbf{\Lambda} = \{\alpha, v_p, \sigma_{p0}, \rho_p, C_p, k_{p0}\}$ of the test data. The parameter combinations of the test data represented by **Data number** is shown in Tables 4–6. Note that the test data is brand new data for the trained model, which means they were completely isolated from the training data set and training process.

Also, our model reduced computational cost significantly (as shown in Table 2) while ensuring the desired accuracy (as shown in Table 3). The total time represents the entire simulation time of the machine learning model, including numerical simulation for the number of training data (20% of the total data), the training time, and the testing (prediction) time. The acceleration represents how many times faster it is than direct exhaustive simulations to calculate the fouling rate for 9010 combinations of parameters.

This implies that the machine learning model could learn and detect the pattern of the fouling rate (which includes the process of solving differential equations) and predict the fouling of the channel, without having any mathematical or physical knowledge to solve differential equations and physics problems. This could illustrate the potential predictive power of machine learning with the low computational cost because we do not need to do numerical simulation for all the combinations which may be included in the genetic algorithm process.



(a) $Q_0 = 0.008(m^3/s)$

(b) $Q_0 = 0.010(m^3/s)$

(c) $Q_0 = 0.012(m^3/s)$

Fig. 8. Prediction of the fouling rate of food channel radius.

13. Overall outlook

Currently, a modest level of modern technologies has been implemented in food production. For example, sensors, cameras, telecommunications have not been widely deployed. Furthermore, the cost of specialized equipment has been prohibitive and the development of coherent, easy-to-use, rapid data fusion/management systems across

Table 4
Parameter combinations of the test data ($Q_0 = 0.008$ (m^3/s)).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
0	1.549	0.15	2.456	7759.404	3433.3	0.218
1	3.847	0.294	0.335	6105.349	4259.458	0.508
2	3.895	0.245	0.819	4181.942	5092.967	0.266
3	1.064	0.125	0.506	6140.392	5263.828	0.193
4	2.562	0.297	2.351	3775.15	3415.132	0.438
5	8.099	0.237	0.121	3727.23	2749.679	0.195
6	4.487	0.226	0.157	5545.354	3138.772	0.328
7	6.373	0.186	1.66	7784.213	5910.001	0.268
8	4.621	0.091	1.949	6439.95	4414.465	0.188
9	7.89	0.277	0.12	4903.641	1327.825	0.143
10	5.276	0.264	1.484	7035.64	3099.573	0.27
11	5.347	0.165	1.009	8632.634	5082.983	0.461
12	7.999	0.212	1.315	6683.187	1967.633	0.44
13	8.607	0.051	1.426	8422.895	3126.233	0.406
14	4.689	0.181	2.007	7584.325	5890.057	0.373
15	6.837	0.271	1.947	3265.004	4077.301	0.422
16	4.067	0.101	0.14	7968.876	3444.474	0.436
17	6.099	0.144	1.381	5948.694	4144.484	0.279
18	6.815	0.24	1.905	3973.045	3267.218	0.346
19	4.376	0.136	0.722	7860.504	2820.899	0.5
20	9.257	0.197	0.173	3092.617	2956.472	0.158
21	5.528	0.122	0.209	4989.043	5837.073	0.14
22	5.655	0.071	0.528	6727.897	5773.078	0.116
23	6.002	0.191	2.224	3335.19	3355.906	0.306
24	7.627	0.064	0.304	4873.36	1369.402	0.167
25	7.68	0.106	1.054	3243.908	4748.334	0.216
26	8.307	0.097	1.901	7089.897	2308.456	0.237
27	2.666	0.153	1.32	4057.853	1012.664	0.255
28	4.896	0.116	2.103	5193.494	5475.022	0.146
29	6.049	0.061	2.233	6832.804	4226.989	0.133
30	9.659	0.077	0.493	7077.536	3192.111	0.243
31	7.775	0.255	1.57	8566.101	1376.947	0.472
32	8.81	0.283	1.545	4493.745	5214.075	0.235
33	2.054	0.268	1.047	7779.834	1362.59	0.191
34	8.589	0.216	1.123	7107.375	3363.087	0.17
35	9.497	0.283	1.739	4630.313	1993.417	0.271
36	5.772	0.136	1.529	4635.428	4075.312	0.472
37	3.959	0.249	0.567	4696.297	1548.019	0.328
38	5.166	0.06	0.816	7087.657	5301.154	0.308
39	6.712	0.158	0.832	7776.77	2382.714	0.327
40	7.988	0.19	0.145	3634.772	1554.625	0.236
41	1.044	0.117	0.37	6138.783	4653.459	0.239
42	4.365	0.054	2.375	3587.32	5727.176	0.362
43	9.887	0.063	0.317	8819.026	5592.158	0.225
44	4.351	0.225	0.663	3033.798	5027.076	0.194
45	3.4	0.292	1.52	8851.484	2095.329	0.236
46	3.999	0.292	0.947	6391.476	4732.204	0.139
47	1.701	0.155	0.129	7762.79	5396.288	0.306
48	2.909	0.281	2.455	3551.428	2084.967	0.373
49	9.847	0.167	0.844	5712.906	4155.826	0.477
50	1.418	0.056	0.512	3089.134	5009.892	0.255
51	2.694	0.169	2.128	8402.041	2136.059	0.171
52	3.747	0.117	1.019	4391.483	4364.756	0.517
53	4.239	0.3	2.043	5246.343	4470.875	0.412

(*continued on next page*)

Table 4 (*continued*).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
54	9.979	0.077	0.273	4239.395	2739.448	0.396
55	4.874	0.261	0.328	8498.197	5401.833	0.24
56	6.398	0.056	0.301	3232.287	3332.41	0.373
57	7.663	0.078	0.154	3122.636	2411.005	0.214
58	1.514	0.14	0.238	6034.478	5289.729	0.509
59	9.996	0.19	0.883	6812.561	5578.166	0.165
60	1.607	0.251	1.41	4849.983	3990.129	0.139
61	9.718	0.12	0.854	4394.153	2442.501	0.378
62	4.92	0.099	1.503	3621.493	4953.384	0.446
63	6.286	0.195	0.854	8390.804	5191.483	0.287
64	2.437	0.089	1.775	5797.98	1149.577	0.481
65	1.329	0.191	1.449	8568.115	5527.728	0.315
66	5.432	0.257	2.285	6038.271	4253.427	0.269
67	4.804	0.278	2.287	8215.119	3030.96	0.115
68	8.814	0.184	0.157	3068.411	2843.516	0.151
69	6.651	0.144	2.44	6823.313	3277.01	0.413
70	9.965	0.28	1.788	7031.894	3760.994	0.472
71	3.354	0.086	1.932	5018.222	3000.592	0.382
72	1.169	0.235	1.463	6540.024	4345.719	0.497
73	3.062	0.275	2.028	3219.607	3852.654	0.195
74	4.217	0.249	0.704	3738.223	3115.538	0.263
75	5.827	0.074	2.087	4486.627	4704.963	0.519
76	7.767	0.244	0.141	3435.857	1243.581	0.134
77	1.346	0.154	0.224	6231.153	4732.696	0.224
78	1.27	0.241	1.288	8043.79	4691.051	0.481
79	5.676	0.24	1.278	8149.764	3998.937	0.463
80	5.215	0.081	0.316	4160.237	1821.958	0.512
81	5.495	0.149	1.72	6061.707	3496.811	0.309
82	9.217	0.153	0.451	5250.311	1459.664	0.467
83	7.969	0.069	1.825	7418.46	4914.177	0.509
84	3.173	0.183	2.44	8438.922	3141.411	0.483
85	1.882	0.141	2.28	4967.85	1073.279	0.38
86	5.487	0.156	1.35	3969.385	2644.987	0.238
87	6.69	0.281	1.166	8996.767	1632.565	0.362
88	5.602	0.092	0.341	7741.338	3433.192	0.111
89	4.442	0.284	1.206	8580.623	2926.173	0.357
90	1.573	0.082	1.427	4317.658	5299.801	0.179

different platforms is lacking. Additionally, while control systems exist, they simply are too slow to be useful in deployed mobile computing platforms in harsh environments. The long term mission of this research is to integrate and implement convergent research in the development of smart, robust, and inexpensive systems that are easy to maintain, upgrade, and deploy, incorporating state-of-the-art technologies. A key to much of this work is the transfer of advances in the fields of Advanced Manufacturing and Computational Science to food production. In particular, digital twins, which refers to a digital replica of physical systems, blend artificial intelligence, machine learning, and software analytics with data to create living digital computer models that can update and change in tandem with their physical counterparts. We will seek to enable real-time simulation of processing devices to operate in tandem with their deployed response. Updates to the digital twin are made continuously in near real-time, which necessitates rapid wireless communication, hyperspectral cameras and sensor fusion, and rapid simulation of process behavior. The digital twin concept should quickly ascertain fault behavior by utilizing the best available data. Today, there is no shortage of simulation codes; however, the fundamental limitations are real-time accuracy and deployable in-the-field use in harsh environments. A core issue across all domains of application is extreme flexibility — the ability of a system to adapt to rapid changes in the environment and system capabilities by autonomously modifying tasks and then apply various problem-solving approaches. In this context, a goal will be to make fundamental advances in several coupled autonomy-related fields to increase functional flexibility, while being constrained by

Table 5
Parameter combinations of the test data ($Q_0 = 0.010$ (m^3/s)).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
0	6.14	0.131	2.336	3580.736	1074.287	0.405
1	7.366	0.194	1.319	7920.128	3488.167	0.441
2	6.546	0.069	0.595	7439.564	1296.572	0.142
3	8.069	0.197	0.273	7197.536	2553.029	0.394
4	6.992	0.198	0.91	3704.36	5351.689	0.308
5	6.506	0.169	0.215	5782.7	5007.166	0.494
6	1.283	0.075	1.633	7819.798	4286.152	0.518
7	8.999	0.064	2.009	4702.403	2703.609	0.232
8	6.913	0.057	2.295	6070.889	3640.574	0.211
9	5.313	0.28	1.504	5434.051	4517.909	0.285
10	8.343	0.149	1.713	7696.903	3796.667	0.119
11	4.251	0.268	2.251	5347.55	5358.302	0.366
12	5.399	0.195	0.924	4955.521	5341.165	0.495
13	1.573	0.084	0.16	3933.561	1358.369	0.427
14	5.526	0.059	1.299	4417.878	5746.938	0.233
15	7.907	0.199	1.525	8347.193	5045.599	0.15
16	7.913	0.184	1.262	4081.202	5150.931	0.258
17	8.825	0.118	0.127	3679.186	3709.769	0.261
18	8.208	0.244	1.45	4764.49	4500.437	0.434
19	6.495	0.238	0.22	6225.841	4326.061	0.2
20	3.545	0.159	0.58	3533.284	1859.182	0.337
21	3.646	0.201	2.372	6478.118	3440.068	0.12
22	3.986	0.195	0.497	7797.545	1876.565	0.137
23	1.525	0.121	0.24	5680.408	5489.909	0.301
24	2.546	0.202	1.015	8919.161	3901.178	0.376
25	1.336	0.071	1.671	6119.824	2968.495	0.328
26	2.328	0.061	2.496	3945.234	4892.474	0.199
27	8.521	0.226	1.592	8861.208	4454.62	0.456
28	9.33	0.188	0.625	7597.352	1331.154	0.181
29	9.309	0.083	0.142	4597.605	3840.298	0.396
30	4.506	0.211	2.116	6237.805	1367.773	0.402
31	7.365	0.152	1.846	8430.727	1885.953	0.377
32	6.613	0.216	2.199	8504.861	2341.794	0.421
33	4.45	0.238	1.446	3544.603	4181.313	0.34
34	4.609	0.087	0.288	7393.452	1554.853	0.288
35	9.616	0.261	2.172	7403.373	5379.083	0.386
36	1.071	0.089	1.857	8608.928	1252.925	0.439
37	3.446	0.193	0.418	4291.864	3091.729	0.117
38	6.782	0.106	2.198	8922.436	3786.966	0.397
39	4.383	0.166	2.492	5042.57	1773.285	0.472
40	6.799	0.254	0.11	6127.599	1292.772	0.196
41	5.138	0.134	1.987	7621.284	2952.805	0.371
42	7.023	0.175	2.296	3309.995	4505.566	0.489
43	7.086	0.267	0.218	4201.604	2153.164	0.159
44	5.792	0.145	1.29	3216.131	1738.549	0.497
45	8.795	0.155	2.01	5458.378	2902.507	0.334
46	5.576	0.206	0.772	4151.183	1507.298	0.179
47	5.532	0.167	1.711	8743.86	2311.082	0.309
48	9.374	0.248	1.838	8682.585	2009.389	0.359
49	6.109	0.078	2.381	3636.448	2012.176	0.296
50	2.166	0.203	1.099	8702.51	3044.366	0.334
51	5.297	0.077	2.366	4051.802	3998.746	0.518
52	3.328	0.271	2.136	7263.344	2043.585	0.261

(*continued on next page*)

the multiple aspects of variable system capabilities and operation in complex environments. These methods should be benchmarked and validated against an extensive suite of experimental tests.

Table 5 (*continued*).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
53	4.575	0.101	0.154	6912.569	2418.829	0.19
54	8.081	0.151	2.457	7083.64	2776.945	0.17
55	3.908	0.177	0.548	6635.572	3127.318	0.314
56	6.428	0.264	1.733	6713.824	2894.822	0.515
57	8.357	0.123	0.296	6581.155	2697.933	0.323
58	9.003	0.214	0.194	5955.788	1192.977	0.253
59	9.802	0.055	0.202	6481.557	5289.709	0.308
60	2.432	0.103	2.243	4260.709	3420.331	0.171
61	7.717	0.12	1.827	6263.439	2399.915	0.37
62	3.475	0.131	1.634	7523.677	4489.278	0.256
63	7.867	0.129	0.271	3663.054	1972.714	0.347
64	5.743	0.271	2.386	4490.073	1391.131	0.26
65	1.977	0.193	2.254	6120.876	4469.23	0.458
66	2.757	0.152	0.42	6308.862	1341.053	0.27
67	4.742	0.183	1.68	4413.228	1015.702	0.484
68	9.889	0.096	0.274	3253.247	1898.977	0.44
69	6.368	0.265	0.808	7856.217	5156.008	0.455
70	9.919	0.095	1.355	5642.983	1646.385	0.291
71	2.827	0.221	0.558	8795.148	4688.409	0.265
72	4.534	0.184	1.4	4462.572	1254.301	0.134
73	1.227	0.166	1.265	8281.254	3170.687	0.14
74	6.591	0.297	1.538	4557.043	5111.217	0.25
75	5.312	0.147	1.817	5661.513	1222.957	0.455
76	3.909	0.118	1.692	3771.867	3305.802	0.194
77	9.697	0.269	2.317	7709.971	5539.29	0.173
78	4.905	0.12	0.937	3135.147	5807.623	0.158
79	7.611	0.111	0.419	5335.717	5490.597	0.245
80	2.27	0.282	0.806	7178.22	5990.579	0.375
81	7.148	0.119	0.147	3693.587	2416.442	0.33
82	2.208	0.291	1.094	3544.886	3700.504	0.196
83	9.935	0.068	1.739	6036.328	2055.888	0.206
84	3.848	0.103	1.569	6220.79	1088.941	0.513
85	4.786	0.068	0.35	4669.444	3201.769	0.516
86	3.297	0.186	2.295	3256.407	1832.094	0.442
87	8.355	0.163	1.026	3105.065	5858.134	0.487
88	9.78	0.23	1.363	8354.454	4339.494	0.312
89	8.423	0.21	1.314	4572.152	3396.524	0.126
90	3.571	0.208	2.002	4743.436	4432.712	0.221

Table 6
Parameter combinations of the test data ($Q_0 = 0.012$ (m^3/s)).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
0	6.4	0.067	0.522	7876.243	5294.28	0.195
1	8.78	0.24	1.116	3315.163	3814.936	0.241
2	1.915	0.132	2.017	5458.866	5279.052	0.474
3	5.979	0.146	2.487	4069.539	4539.445	0.317
4	4.064	0.115	2.412	3081.456	4120.004	0.327
5	3.761	0.085	0.371	3162.713	5162.42	0.458
6	2.31	0.069	0.104	4964.014	5683.006	0.335
7	4.44	0.151	0.919	8144.161	5741.744	0.308
8	1.221	0.209	0.376	8897.054	1056.524	0.116
9	6.439	0.21	0.847	5777.023	1814	0.321
10	8.849	0.288	0.183	7840.757	1727.002	0.258
11	7.487	0.171	0.55	8558.128	3965.964	0.121

(*continued on next page*)

Table 6 (*continued*).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
12	7.588	0.169	1.783	6494.871	2374.835	0.338
13	8.981	0.216	0.99	7794.138	4990.806	0.472
14	8.75	0.139	1.734	5713.009	2484.315	0.17
15	4.216	0.093	0.442	6046.996	5614.149	0.493
16	5.844	0.282	0.22	3935.321	4672.811	0.256
17	4.761	0.191	1.158	7752.158	3894.048	0.431
18	4.763	0.269	2.057	3853.139	1174.468	0.152
19	5.056	0.214	1.258	4200.577	2981.031	0.413
20	6.769	0.206	1.161	8467.196	5783.529	0.388
21	9.673	0.101	0.352	5057.071	2211.53	0.189
22	3.21	0.297	2.194	3280.192	4266.42	0.318
23	9.994	0.273	1.797	4081.603	2730.073	0.178
24	4.713	0.188	1.656	5510.669	3933.104	0.159
25	1.086	0.188	0.581	5109.942	1802.562	0.365
26	9.754	0.291	1.306	8539.635	4946.283	0.269
27	6.067	0.3	0.765	3245.378	1794.394	0.517
28	3.226	0.283	0.529	7603.214	4977.732	0.179
29	7.635	0.114	2.404	6106.222	4012.162	0.235
30	8.139	0.288	0.121	3781.991	1329.161	0.147
31	8.736	0.26	1.823	8281.504	1041.569	0.359
32	8.915	0.206	0.463	6300.075	2275.214	0.257
33	6.199	0.123	1.633	8890.819	2525.629	0.197
34	5.162	0.235	1.417	4398.862	4607.633	0.146
35	8.868	0.062	1	3597.699	5313.489	0.27
36	3.883	0.146	0.627	6571.866	1986.721	0.379
37	2.454	0.153	0.578	3642.078	4549.263	0.276
38	1.263	0.145	0.207	4141.563	3678.891	0.372
39	5.341	0.089	0.959	4360.097	2066.179	0.438
40	1.107	0.183	0.205	8058.297	4336.652	0.302
41	7.514	0.088	2.165	7392.249	1587.091	0.498
42	4.528	0.294	1.271	5270.921	4601.011	0.234
43	8.451	0.133	0.204	5234.141	1257.345	0.195
44	9.383	0.076	0.184	6001.8	1902.392	0.341
45	9.766	0.11	1.323	5491.346	3151.914	0.112
46	8.075	0.121	1.084	4987.955	5017.741	0.142
47	8.623	0.141	1.427	5743.662	5370.107	0.447
48	8.812	0.177	2.22	4669.98	5199.864	0.291
49	4.039	0.26	2.427	5501.073	4214.584	0.494
50	3.25	0.168	1.554	7151.826	4625.882	0.491
51	1.659	0.296	1.132	6315.573	1746.152	0.305
52	5.033	0.06	2.072	5041.384	2139.78	0.286
53	5.723	0.265	1.25	8936.12	4446.507	0.13
54	7.419	0.195	1.261	6695.9	1515.017	0.249
55	6.283	0.054	0.949	4657.896	4149.641	0.511
56	8.674	0.191	2.111	4430.844	3077.662	0.261
57	3.723	0.136	2.064	4707.604	1925.808	0.329
58	1.435	0.266	1.987	6817.481	2040.755	0.43
59	8.294	0.198	0.958	7924.672	1850.88	0.135
60	5.616	0.16	2.158	7039.122	3272.005	0.15
61	8.249	0.225	0.383	4251.902	1873.476	0.314
62	7.484	0.2	1.954	7625.207	2127.711	0.507
63	8.496	0.267	0.382	3791.214	5095.26	0.49
64	9.224	0.147	1.351	8413.807	3864.131	0.358
65	4.129	0.081	2.221	7633.587	2446.464	0.506
66	3.505	0.059	1.365	5032.053	3032.907	0.415

(*continued on next page*)

Table 6 (*continued*).

Data	α	v_p	σ_{p0} (Ω^{-1} m^{-1})	ρ_p (kg/m^3)	C_p (J/(kg K))	k_{p0} (W/(m K))
67	3.215	0.109	0.23	7915.023	1055.042	0.471
68	5.79	0.059	1.71	6376.457	2151.38	0.213
69	2.214	0.178	1.516	5894.345	4671.929	0.35
70	3.621	0.098	1.716	8745.389	4080.627	0.309
71	2.539	0.179	0.465	6227.274	3531.866	0.225
72	9.306	0.274	1.967	3629.099	5095.89	0.135
73	7.608	0.055	2.16	8401.329	4750.897	0.487
74	2.708	0.066	2.313	6694.385	3688.736	0.117
75	2.993	0.15	1.284	3730.68	4412.211	0.292
76	2.232	0.258	2.491	8767.063	4597.317	0.194
77	6.972	0.251	0.256	7734.183	1171.318	0.185
78	4.931	0.171	1.98	4670.762	4427.269	0.394
79	7.116	0.168	2.029	8174.577	3866.464	0.174
80	7.671	0.241	0.107	4285.157	1597.274	0.142
81	4.819	0.116	0.28	3542.585	1363.044	0.156
82	4.388	0.225	0.355	7705.786	3898.758	0.179
83	3.838	0.129	1.132	3071.931	3060.07	0.182
84	7.787	0.223	0.106	3666.485	1475.339	0.133
85	3.691	0.283	2.366	7904.586	3782.818	0.199
86	1.286	0.195	1.802	3339.908	5903.523	0.456
87	5.092	0.114	0.203	3557.497	1379.287	0.166
88	2.214	0.265	1.511	4430.614	3331.978	0.242
89	6.272	0.207	0.895	7031.337	3987.298	0.195
90	7.152	0.068	0.624	3125.018	3860.56	0.437

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Paul Singh, Dennis R. Heldman, Introduction to Food Engineering, Gulf Professional Publishing, 2001.

[2] Alessandro Leonardi, Falk K Wittel, Miller Mendoza, Hans J Herrmann, Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions, Comput. Part. Mech. 1 (1) (2014) 3–13.

[3] Eugenio Oñate, Miguel Angel Celigueta, Salvador Latorre, Guillermo Casas, Riccardo Rossi, Jerzy Rojek, Lagrangian analysis of multiscale particulate flows with the particle finite element method, Comput. Part. Mech. 1 (1) (2014) 85–102.

[4] B. Avci, P. Wriggers, A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows, J. Appl. Mech. 79 (1) (2012).

[5] T. Zohdi, Embedded electromagnetically sensitive particle motion in functionalized fluids, Comput. Part. Mech. 1 (1) (2014) 27–45.

[6] Tarek I. Zohdi, Peter Wriggers, An Introduction to Computational Micromechanics, Springer Science & Business Media, 2008.

[7] Tarek I. Zohdi, On simple scaling laws for pumping fluids with electrically-charged particles, Internat. J. Engrg. Sci. 123 (2018) 73–80.

[8] Osborne Reynolds, XXIX. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels, Philos. Trans. R. Soc. Lond. 174 (1883) 935–982.

[9] Hosahalli S Ramaswamy, Michele Marcotte, Sudhir Sastry, Khalid Abdelrahim, Ohmic Heating in Food Processing, CRC press, 2014.

[10] Peter John Fellows, Food Processing Technology: Principles and Practice, Elsevier, 2009.

[11] K. Shiby Varghese, M.C. Pandey, K. Radhakrishna, A.S. Bawa, Technology, applications and modelling of ohmic heating: a review, J. Food Sci. Technol. 51 (10) (2014) 2304–2317.

[12] Albert Einstein, A new determination of molecular dimensions, Ann. Phys. 19 (1906) 289–306.

[13] Z. Hashin, S. Shtrikman, On some variational principles in anisotropic and nonhomogeneous elasticity, J. Mech. Phys. Solids 10 (4) (1962) 335–342.

[14] Zvi Hashin, Shmuel Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, J. Mech. Phys. Solids 11 (2) (1963) 127–140.

[15] Zvi Hashin, Analysis of composite materials – a survey, 1983.

[16] Salvatore Torquato, H.W. Haslach Jr, Random heterogeneous materials: microstructure and macroscopic properties, Appl. Mech. Rev. 55 (4) (2002) B62–B63.

[17] Mark Kachanov, Behrouz Abedian, On the isotropic and anisotropic viscosity of suspensions containing particles of diverse shapes and orientations, Internat. J. Engrg. Sci. 94 (2015) 71–85.

[18] Igor Sevostianov, Mark Kachanov, Effective properties of heterogeneous materials: Proper application of the non-interaction and the “dilute limit” approximations, Internat. J. Engrg. Sci. 58 (2012) 124–128.
[19] C. Sandu, R.K. Singh, Energy increases in operation and cleaning due to heat-exchanger fouling in milk pasteurization, Food Technol. (Chicago) 45 (12) (1991) 84–91.
[20] J. Visser, Th J.M. Jeurnink, Fouling of heat exchangers in the dairy industry, Exp. Therm Fluid Sci. 14 (4) (1997) 407–424.
[21] H Petermeier, R Benning, A Delgado, U Kulozik, J Hinrichs, T Becker, Hybrid model of the fouling process in tubular heat exchangers for the dairy industry, J. Food Eng. 55 (1) (2002) 9–17.
[22] Hatice Ozlem Ozden, Virendra M. Puri, Computational analysis of fouling by low energy surfaces, J. Food Eng. 99 (3) (2010) 250–256.
[23] Adel Fickak, Ali Al-Raisi, Xiao Dong Chen, Effect of whey protein concentration on the fouling and cleaning of a heat transfer surface, J. Food Eng. 104 (3) (2011) 323–331.
[24] W. Ebert, C.B. Panchal, Analysis of Exxon Crude-Oil-Slip Stream Coking Data, Tech. Rep., Argonne National Lab., IL (United States), 1995.
[25] D.R. Oliver, Stacey G. Ward, Relationship between relative viscosity and volume concentration of stable suspensions of spherical particles, Nature 171 (4348) (1953) 396–397.
[26] T.I. Zohdi, An upper bound on the particle-laden dependency of shear stresses at solid–fluid interfaces, Proc. R. Soc. A 474 (2211) (2018) 20170332.
[27] Kalyanmoy Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Vol. 16, John Wiley & Sons, 2001.
[28] Mohamed G. Sahab, Vassili V. Toropov, Amir H. Gandomi, Optimum design of composite concrete floors using a hybrid genetic algorithm, in: Handbook of Neural Computation, Elsevier, 2017, pp. 581–589.
[29] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT press, 2016.
[30] R.S. Mitchell, J.G. Michalski, T.M. Carbonell, An Artificial Intelligence Approach, Springer, 2013.
[31] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT press, 2012.
[32] Pat Langley, et al., Selection of relevant features in machine learning, in: Proceedings of the AAAI Fall symposium on relevance, 184, 1994, pp. 245–271.
[33] Igor Kononenko, Matjaz Kukar, Machine Learning and Data Mining, Horwood Publishing, 2007.
[34] David J Lary, Amir H Alavi, Amir H Gandomi, Annette L Walker, Machine learning in geosciences and remote sensing, Geosci. Front. 7 (1) (2016) 3–10.
[35] Grace X. Gu, Chun-Teh Chen, Markus J. Buehler, De novo composite design based on machine learning algorithm, Extreme Mech. Lett. 18 (2018) 19–28.
[36] Chun-Teh Chen, Grace X. Gu, Machine learning for composite materials, MRS Commun. 9 (2) (2019) 556–566.
[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[38] Thiago S. Guzella, Walmir M. Caminhas, A review of machine learning approaches to spam filtering, Expert Syst. Appl. 36 (7) (2009) 10206–10222.
[39] Igor Santos, Javier Nieves, Yoseba K Penya, Pablo G Bringas, Machine-learning-based mechanical properties prediction in foundry production, in: 2009 ICCAS-SICE, IEEE, 2009, pp. 4536–4541.
[40] Arun Mannodi-Kanakkithodi, Ghanshyam Pilania, Tran Doan Huan, Turab Lookman, Rampi Ramprasad, Machine learning strategy for accelerated design of polymer dielectrics, Sci. Rep. 6 (2016) 20952.
[41] Ghanshyam Pilania, Prasanna V Balachandran, James E Gubernatis, Turab Lookman, Data-based methods for materials design and discovery: Basic ideas and general methods, Synth. Lect. Mater. Opt. 1 (1) (2020) 1–188.
[42] Masoud Sarveghadi, Amir H Gandomi, Hamed Bolandi, Amir H Alavi, Development of prediction models for shear strength of SFRCB using a machine learning approach, Neural Comput. Appl. 31 (7) (2019) 2085–2094.
[43] Florent Pled, Christophe Desceliers, Amir H Gandomi, Christian Soize, Neural network prediction of cortical bone damage using a stochastic computational mechanical model, in: 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019), 2019.
[44] Christian Soize, A probabilistic learning on manifolds as a new tool in machine learning and data science with applications in computational mechanics, in: UNCECOMP 2019, 3nd International Conference on Uncertainty Quantification in Computational Sciences and Engineering, and COMPDYN 2019, 7th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, 2019.
[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8026–8037.
[46] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, arXiv preprint arXiv:1502.03167.
[47] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[48] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.