Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 331 (2018) 343–362
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning

T.I. Zohdi

Department of Mechanical Engineering, University of California, Berkeley, CA, 94720-1740, USA

Received 19 September 2017; received in revised form 21 November 2017; accepted 22 November 2017
Available online 2 December 2017

Abstract

Rapid free-form printing of heated polymeric materials, whereby an extruder is attached to a robotic arm, is a growing field. In order to properly plan a complex tool path, one needs to determine the dynamic thermomechanical response of the extruded filament material as the robot arm is moving in free-space. Accordingly, the analysis in this work involves:

- Kinematics of the robot arm and the extrusion-printer head,
- Dynamics of the extruded polymer filament,
- Thermal conduction along the length of the heated filament,
- Convective and radiative cooling of the filament through the surroundings and
- Curing and solidification of the filament.

Numerical simulations are undertaken to illustrate the basic model and a Machine Learning Algorithm is developed to optimize the robotic system to deliver a prespecified structure. Detailed extensions are also discussed.

Keywords: 3D printing; Machine learning; Dynamics; Thermodynamics

1. Introduction

Within the last decade, several industrialized countries have stressed the importance of advanced manufacturing to their economies. Many of these plans have highlighted the development of additive manufacturing techniques, such as 3D printing which, as of 2017, are still evolving. The objective is to develop superior products, produced at lower overall operational costs. For these goals to be realized, a deep understanding of the essential ingredients comprising the materials involved in additive manufacturing is needed. The combination of rigorous material modeling theories, coupled with the dramatic increase of computational power can potentially play a significant role in the analysis, control, and design of many emerging additive manufacturing processes. Specialized materials and the precise design

E-mail address: zohdi@berkeley.edu.

https://doi.org/10.1016/j.cma.2017.11.030
0045-7825/

of their properties are key factors in the processes. One such process is extrusion-based deposition. The goal of these processes is primarily to build freeform structures that are extremely difficult to construct using classical manufacturing methods.

Specifically, Additive Manufacturing (AM) is usually defined as the process of free-form deposition of materials to build up structures from 3D model data, as opposed to subtractive manufacturing methodologies, which remove material (American Society for Testing and Materials, ASTM). We refer the reader to the recent overview article by Huang et al. [1] on the wide array of activities in the manufacturing community in this area. One subclass of AM, so-called 3D Printing (3DP), has received a great deal of attention over the last few years. Typically such a process takes CAD drawings and slices them into layers, printing layer by layer. 3DP was a 2.2 billion dollar industry in 2016, with applications ranging from motor vehicles, consumer products, medical devices, military hardware and the arts. One subclass of additive manufacturing are heated filament based processes (historically for prototyping) which are comprised of thermoplastics, which is the focus of this paper.

Recently, free-form printing of heated polymeric materials, whereby an extruder is attached to a robotic arm, has gain notoriety. For example, see https://www.youtube.com/watch?v=umcSJ2A_KuU. In order to properly plan a complex tool path, one needs to determine the dynamic thermomechanical response of the extruded filament material as the robot arm is moving in free-space. Accordingly, the analysis in this work involves:

- Kinematics of the robot arm and the extrusion-printer head,
- Dynamics of the extruded polymer filament,
- Thermal conduction along the length of the heated filament,
- Convective and radiative cooling of the filament through the surroundings and
- Curing and solidification of the filament.

Following a framework found in Zohdi [2–20], the objective of this work is to develop a computational framework for a manufacturing process design tool which captures the following, strongly-coupled physical events:

- *extruded filament dynamics*, which primarily entails: (a) the dynamics of the heated semi-solid material out of a nozzle, (b) the binding forces of the extruded material to the rest of the solidifying structure, (c) the material stiffness-dependency on the material temperature and (d) the forces of gravitation.
- *extruded filament heat-transfer*, which primarily entails: (a) heat transfer from the extruder to the filament, (b) conduction along the filament, (c) convective cooling of the filament to the surroundings and (d) radiative cooling of the filament to the surroundings.

2. A model problem: kinematics of a free-form robotic printer

As a model problem, (Fig. 1), we employ a simple vector loop representation for the kinematics of an extruder placed onto the end of a robot arm. Drawing on methods used in the robotics literature (for example, see Hunt [21], Hartenberg and Denavit [22], Howell [23], McCarthy [24,25], Reuleaux [26], Sandor and Erdman [27], Slocum [28], Suh and Radcliffe [29] and Uicker et al. [30]), we consider the idealization of robotic printer illustrated in Fig. 1 as a linkage. Links one and two are x–y planar, while link three is x–z planar. The angles and angular velocities are all controlled. The position vector ($\boldsymbol{r}^e$) to the extruder head is given by

$$\boldsymbol{r}^e = \boldsymbol{r}^r_1 + \boldsymbol{r}^r_2 + \boldsymbol{r}^r_3. \tag{1}$$

Differentiating, a velocity vector loop is generated

$$\boldsymbol{v}^e = \dot{\boldsymbol{r}}^r_1 + \dot{\boldsymbol{r}}^r_2 + \dot{\boldsymbol{r}}^r_3 = \dot{\boldsymbol{r}}^e. \tag{2}$$

In component form, for the x components of position

$$r^e_x = r^r_1 cos\theta_1 + r^r_2 cos\theta_2 + r^r_3 sin\theta_3, \tag{3}$$

for the y components of position

$$r^e_y = r^r_1 sin\theta_1 + r^r_2 sin\theta_2 \tag{4}$$

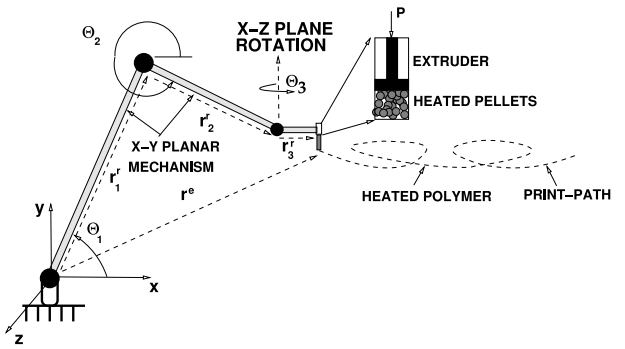

Fig. 1. Model problem of a robot with a extruding printer head.

and for the z components of position

$$r_z^e = r_3^r cos\theta_3, \tag{5}$$

where the link lengths are given by $r_i = \|\boldsymbol{r}_i^r\| = \sqrt{\boldsymbol{r}_i^r \cdot \boldsymbol{r}_i^r}$ (for $i = 1, 2, 3$) and the planar (x–y) angles are measured counter-clockwise from horizontal right (Fig. 1). The velocities can subsequently found from differentiating the component equations of Eqs. (3)–(5), yielding, for the x components of velocity

$$v_x^e = -r_1^r \dot{\theta}_1 sin\theta_1 - r_2^r \dot{\theta}_2 sin\theta_2 + r_3^r \dot{\theta}_3 cos\theta_3, \tag{6}$$

for the y components of velocity,

$$v_y^e = r_1^r \dot{\theta}_1 cos\theta_1 + r_2^r \dot{\theta}_2 cos\theta_2 \tag{7}$$

and for the z components of velocity

$$v_z^e = -r_3^r \dot{\theta}_3 sin\theta_3. \tag{8}$$

The total velocity of the filament coming out of the extruder is the velocity of the extruder plus the relative filament extrusion velocity ($\Delta \boldsymbol{v}^{f-e}$):

$$\boldsymbol{v}^f = \boldsymbol{v}^e + (\boldsymbol{v}^f - \boldsymbol{v}^e) = \boldsymbol{v}^e + \Delta \boldsymbol{v}^{f-e}. \tag{9}$$

3. Dynamics of an extruded heated filament

The increments of extruded filament, which grow as the robotic system proceeds (Fig. 2), are each represented by a continuously growing number of "control points" (or "control particles"). Accordingly, at a snapshot in time, consider a collection of N_{cp} control points. For an arbitrary ith control point in the system, acted upon by inter-control point binding forces, the dynamics are governed by

$$m_i \ddot{\boldsymbol{r}}_i = \boldsymbol{\Psi}_i^b + m\boldsymbol{g} + other\ external\ forces \stackrel{\text{def}}{=} \boldsymbol{\Psi}_i^{tot}(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{N_{cp}}), \tag{10}$$

where $\boldsymbol{r}_i$ is the position vector of the ith control point and $\boldsymbol{\Psi}_i^{tot}$ is the total of the forces acting on the ith control point.

4. Interaction (binding) forces

Consider the ith control point which is bound to the two adjacent control points (Fig. 2). We account for two forces, generated by the relative velocity of control point i and neighboring control point j:

- Normal force (here the superscripts denote the normal components of velocity):

$$\boldsymbol{\Psi}_i^{b,n} = K^n(\boldsymbol{v}_j^n - \boldsymbol{v}_i^n)\Delta t A_{ij}^c \tag{11}$$

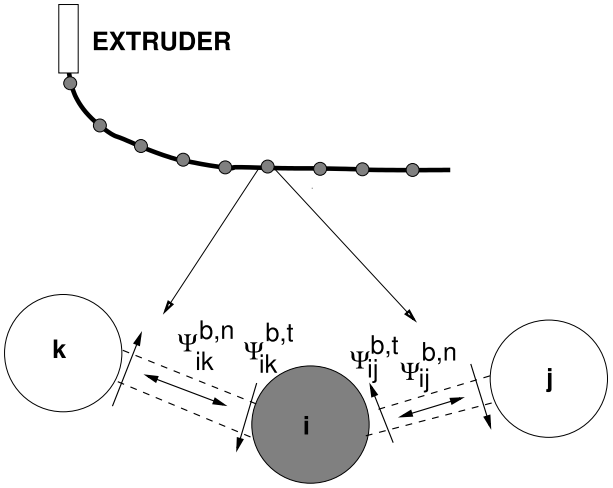

Fig. 2. Forces acting on a control point ("control particle").

where

$$v_i^n = \boldsymbol{v}_i \cdot \boldsymbol{n}_{ij}, \tag{12}$$

and

$$v_j^n = \boldsymbol{v}_j \cdot \boldsymbol{n}_{ij}, \tag{13}$$

where

$$\boldsymbol{n}_{ij} = -\frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|} = \frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}. \tag{14}$$

Here, Δt is a time-increment and A^c_{ij} is the cross-sectional area of the filament.

- Tangential force (here the subscripts denote the tangential components of velocity):

$$\boldsymbol{\Psi}^{b,t} = K^t(\boldsymbol{v}_j^t - \boldsymbol{v}_i^t)\Delta t A^c_{ij} \tag{15}$$

where, subtracting away the normal component of the velocity $\boldsymbol{v}^t = \boldsymbol{v} - (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n}$, yields the tangential velocity

$$\boldsymbol{v}_i^t = \boldsymbol{v}_i - (\boldsymbol{v}_i \cdot \boldsymbol{n}_{ij})\boldsymbol{n}_{ij}, \tag{16}$$

and

$$\boldsymbol{v}_j^t = \boldsymbol{v}_j - (\boldsymbol{v}_j \cdot \boldsymbol{n}_{ij})\boldsymbol{n}_{ij}. \tag{17}$$

The total binding force for each control point i is

$$\boldsymbol{\Psi}_i^b = \boldsymbol{\Psi}_i^{b,n} + \boldsymbol{\Psi}_i^{b,t}. \tag{18}$$

With the governing equations established, one can then integrate Eq. (10) to obtain the velocity for the ith control point with a simple Forward-Euler integration[1]

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i(t) + \frac{1}{m_i}\int_t^{t+\Delta t} \boldsymbol{\Psi}_i^{tot}(t)\, dt \approx \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{\Psi}_i^{tot}(t), \tag{19}$$

and the position for the ith control point by applying the integration process again:

$$\boldsymbol{r}_i(t + \Delta t) \approx \boldsymbol{r}_i(t) + \Delta t \boldsymbol{v}_i(t), \tag{20}$$

[1] More sophisticated implicit solution methods for strongly interacting multi-component systems can be found in Zohdi [2–20].

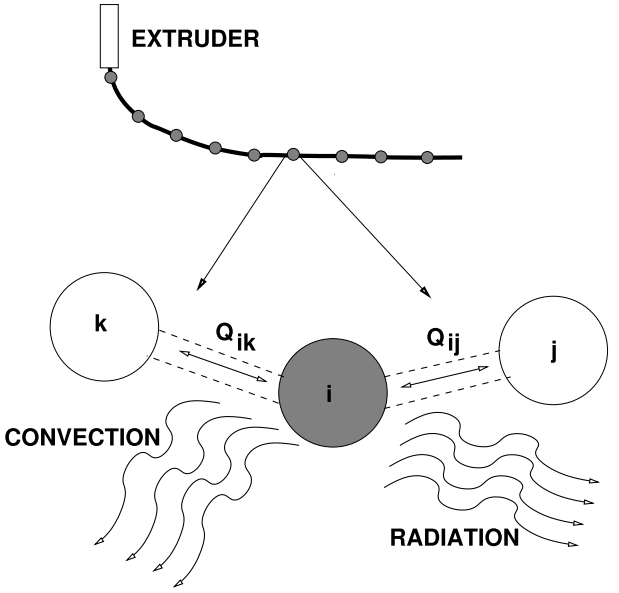

Fig. 3. Conductive exchange for a control point with neighboring elements (through control points), as well as convective and radiative losses to the surroundings.

which provides an updating scheme for the set equations for $i = 1, 2, 3, \ldots, N_{cp}$ control points. Other types of forces, such as electromagnetic actuation forces, are discussed in the summary section of this paper.

5. Thermal interaction between the filament increments, robot and surroundings

For the thermodynamics of lumped mass control points, we consider conduction, convection and radiation (Fig. 3). We assume that strain-rate effects are negligible for the element's thermodynamics. Thus, for each control point $(i = 1, 2, \ldots, N_{cp})$,

$$m_i C_i \dot{\theta}_i = \mathcal{Q}_i + \mathcal{H}_{ci} + \mathcal{H}_{ri}, \tag{21}$$

where $\mathcal{Q}_i$ represents the conductive contribution from surrounding control points, the robotic extruder and support structures (if any), $\mathcal{H}_{ci}$ represents surrounding convection and $\mathcal{H}_{ri}$ represents the (infrared) radiation. It is assumed that the temperature fields are uniform within the (small) elements surrounding the control points. This assumption is justified, i.e. a lumped thermal model (associates with the control points), ignoring temperature gradients and assuming a uniform temperature when the Biot number is small. The Biot number for a cylindrical element scales with the ratio of the element volume (V) to the element surface area (A^s), $\frac{V}{A^s} = \frac{\pi R^2 \Delta L}{2\pi R \Delta L} = \frac{R}{2}$ (R is the radius of the filament), which indicates that a uniform temperature distribution is appropriate, since the elements, by definition, are small. Assuming that the fields are uniform in each element allows for the following (for the $i = 1, 2, \ldots, N_{cp}$ control points)

$$\mathcal{Q} = -\int_{\partial\omega} \boldsymbol{Q} \cdot \boldsymbol{n}\, dA \approx \sum_{j=1}^{N_{cp}} I\!K_{ij} \frac{\theta_j - \theta_i}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|} A^c_{ij}, \tag{22}$$

where the summation extends over all elements $j = 1, 2, 3, \ldots, N_{cp}$ that are in contact with element i (Fig. 3). $A^c_{ij} = \pi R_i^2$ is the contact area associated with the control point/element pair (ij).[2] This yields, including standard

[2] We keep the mass of the ligaments, $m_i = 2\rho\pi R_i \Delta L_i$, between the control points constant (ΔL_i is the distance between control points), thus the radius is adjusted.

convection and radiation terms

$$m_i C_i \frac{d\theta_i}{dt} = \underbrace{\sum_{j=1}^{N_{cp}} I\!K_{ij} \frac{\theta_j - \theta_i}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|} A_{ij}^c}_{\mathcal{Q}_i} \underbrace{- h A_i^s \|\boldsymbol{v}_i - \boldsymbol{v}_s\|(\theta_i - \theta_s)}_{\mathcal{H}_{ci}} \underbrace{- \epsilon \beta A_i^s (\theta_i^4 - \theta_s^4)}_{\mathcal{H}_{ri}}, \tag{23}$$

where h is the convection coefficient, $A^s \approx \frac{1}{2}(2\pi R^{left} \Delta L^{left} + 2\pi R^{right} \Delta L^{right})$ is the surface area (averaged from the two ligaments connected to the control point), $\|\boldsymbol{v}_i - \boldsymbol{v}_s\|$ is the norm of the relative velocity between the control point and the surroundings $0 \leq \epsilon \leq 1$ is the radiative efficiency and $\beta = 5.67 \times 10^{-8}$ W/(m^2 − K^4) is the Stefan–Boltzmann constant.

5.1. Numerical integration

Explicit numerical integration of the energy equation yields, for each control point ($i = 1, 2, \ldots, N_{cp}$) yields

$$\begin{aligned} \theta_i(t + \Delta t) &= \theta_i(t) + \frac{1}{m_i C_i} \left(\int_t^{t+\Delta t} \mathcal{Q}_i \, dt + \int_t^{t+\Delta t} \mathcal{H}_{ci} \, dt + \int_t^{t+\Delta t} \mathcal{H}_{ri} \, dt \right) \\ &\approx \theta_i(t) + \frac{\Delta t}{m_i C_i} \left(\mathcal{Q}_i(t) + \mathcal{H}_{ci}(t) + \mathcal{H}_{ri}(t) \right). \end{aligned} \tag{24}$$

5.2. Overall solution algorithm

The algorithm is as follows:

(1) COMPUTE ROBOT SYSTEM POSITION TIME = t :

(2) SET i = 1 :

(3) IF i > N_{cp} THEN GO TO (5)

(4) IF i ≤ N_{cp} THEN : (FOR CONTROL POINT i)

- (a) COMPUTE POSITION :$\boldsymbol{r}_i(t + \Delta t)$
- (b) COMPUTE TEMPERATURE :$\theta_i(t + \Delta t)$
- (c) GO TO (3) AND NEXT CONTROL POINT (i = i + 1)

(5)INCREMENT TIME : $t = t + \Delta t$ (25)

Remark. Bending and torsional stresses are of secondary importance for the heated polymeric applications here, however, one could include such effects, in addition to extensional and shear phenomena, by computing these element by element, and applying the resultant to the control points.

6. Numerical example

In order to motivate the example, we consider a model problem comprised of three links, as discussed at the outset of this work. Two of the links move in-plane and the third rotates out of plane. The material is extruded as a given relative velocity to linkage extrusion head. The angular velocities are held constant in this simple illustrative example. Of course, in a real robotic system these velocities would be constantly changing according to a control algorithm. The temperatures of the polymeric are relatively generic, and are not meant to represent and particular polymer.

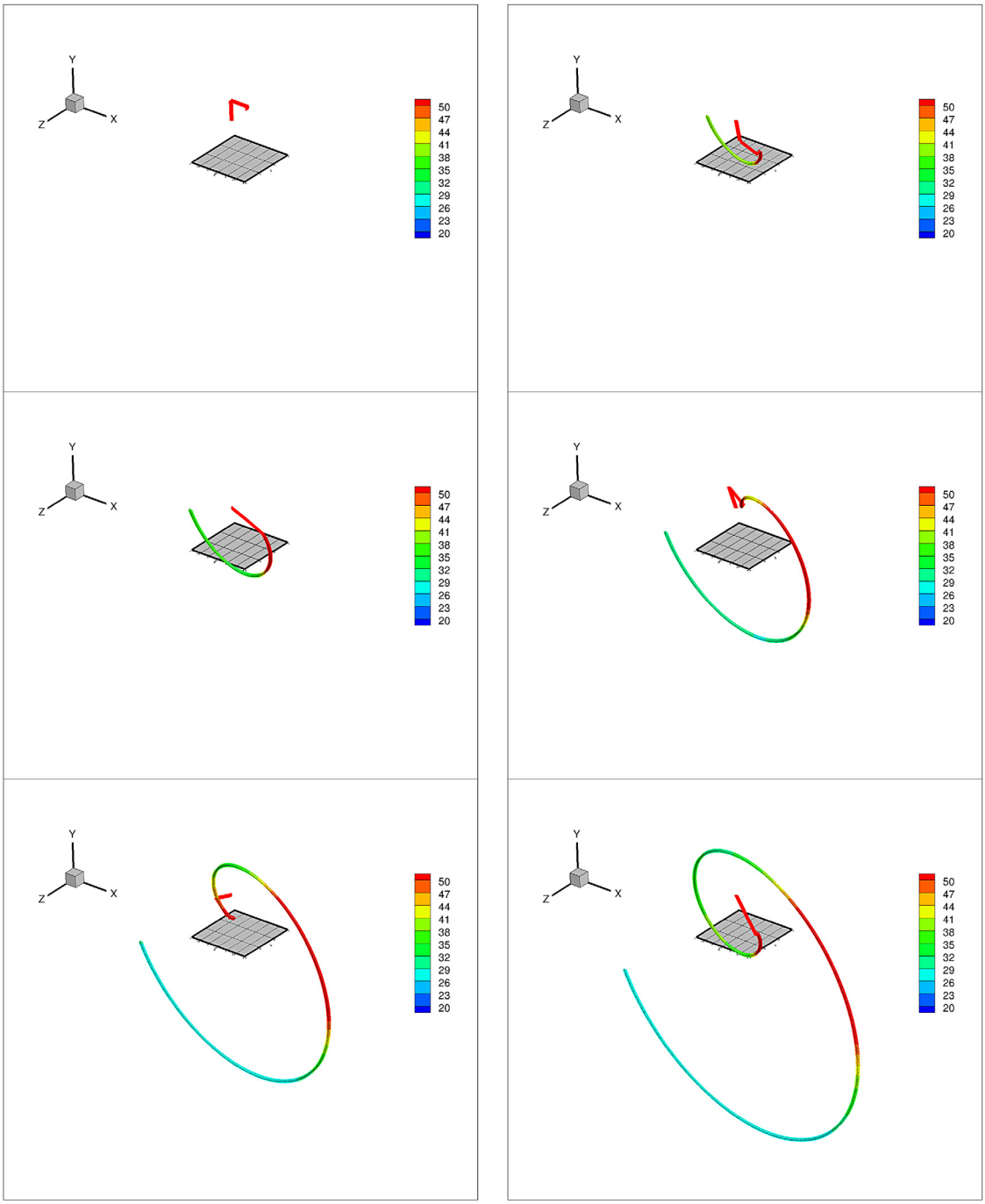

Fig. 4. A polymeric "swirl": left to right and top to bottom. The colors indicate the temperature. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The parameters chosen are shown in Table 1. The results are shown in Fig. 4. Also shown in Fig. 5, is the energy expended by the robot during this process. This was achieved by first computing the kinetic energy associated with the components in the system (ignoring the energy expended by the extruder and the stored thermo-mechanical energy in

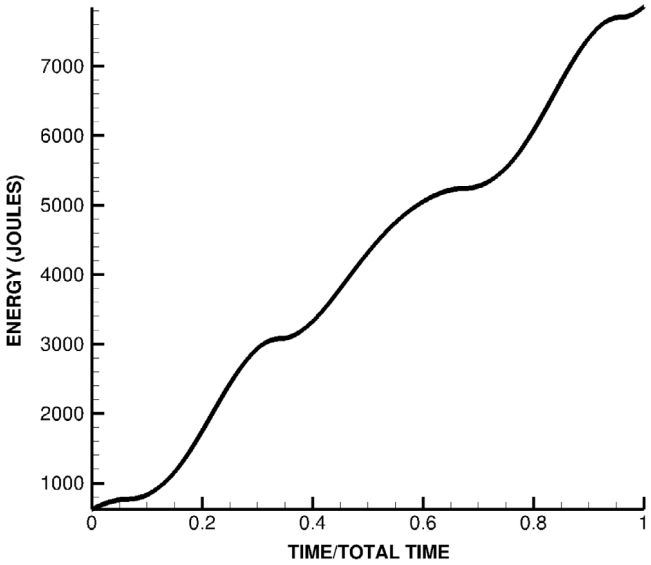

Fig. 5. Energy used by the robot: $\int_0^T \Delta W_{t\to t+\Delta t} \approx \sum_{i=1}^{N_t} \Delta W_{t\to t+\Delta t}$.

Table 1
Material parameters used in the example.

Parameter	Value
Link one angular velocity	$\dot{\theta}_1 = -10$ rad/s
Link two angular velocity	$\dot{\theta}_2 = -1$ rad/s
Link three angular velocity	$\dot{\theta}_3 = 10$ rad/s
Extrusion velocity	$\Delta v^{f-e} = 1$ m/s
Link one mass	$m_1 = \rho_1 \pi R_1^2 L_1$, $\rho_1 = 7000$ kg/m^3, $R_1 = 0.05$ m, $L_1 = 1$ m
Link two mass	$m_2 = \rho_2 \pi R_2^2 L_2$, $\rho_2 = 7000$ kg/m^3, $R_2 = 0.05$ m, $L_2 = 1$ m
Link three mass	$m_3 = \rho_3 \pi R_3^2 L_3$, $\rho_3 = 7000$ kg/m^3, $R_3 = 0.05$ m, $L_3 = 1$ m
Surrounding cooling fluid velocity	$\boldsymbol{v}_f = (10, 0, 0)$ m/s
Surrounding convective cooling coefficient	$h = 100$ J-m/(m^2-K)
Filament mass density	$\rho_f = 2000$ kg/m^3
Temperature of the robot	100 ° C
Heat capacity of the filament	$C = 100$ J/(kg-K)
Ambient temperature	$\theta_o = 25$ °C
Robot temperature	$\theta_R = 100$ °C
Normal thermal dependence of the material	$K_n(\theta = 0) = 10^4$, $K_n(\theta) = K_n(\theta_R) e^{\frac{-10(\theta-\theta_R)}{\theta_R}}$
Tangential thermal dependence of the material	$K_t(\theta = 0) = 10^4$, $K_n(\theta) = K_t(\theta_R) e^{\frac{-10(\theta-\theta_R)}{\theta_R}}$
Filament radius	$R_f = 0.001$ m
Filament thermal conductivity	$\mathbb{K} = 100$ W/(m-K)

the material):

$$T(t) = \sum_{i=1}^{mass\ centers} \frac{1}{2} m_i \boldsymbol{v}_i^{cm}(t) \cdot \boldsymbol{v}_i^{cm}(t) + \sum_{i=1}^{links} \frac{1}{2} I_i^{cm} \dot{\theta}_i^2(t), \tag{26}$$

where, in this specific case, for the translational energy

$$\sum_{i=1}^{mass\ centers} \frac{1}{2} m_i \boldsymbol{v}_i^{cm}(t) \cdot \boldsymbol{v}_i^{cm}(t) = \frac{1}{2} \left(m_1 \dot{\boldsymbol{r}}_1^{cm} \cdot \dot{\boldsymbol{r}}_1^{cm} + m_2 \dot{\boldsymbol{r}}_2^{cm} \cdot \dot{\boldsymbol{r}}_2^{cm} + m_3 \dot{\boldsymbol{r}}_3^{cm} \cdot \dot{\boldsymbol{r}}_3^{cm} \right), \tag{27}$$

and for the rotational energy

$$\sum_{i=1}^{links} \frac{1}{2} I_i^{cm} \dot{\theta}_i^2(t) = \frac{1}{2} \left(I_1^{cm} \dot{\theta}_1^2 + I_2^{cm} \dot{\theta}_2^2 + I_3^{cm} \dot{\theta}_3^2 \right). \tag{28}$$

The potential energy due to gravity is

$$U(t) = \sum_{i=1}^{mass\ centers} m_i g h_i(t) = m_1 g h_1(t) + m_2 g h_2(t) + m_3 g h_3(t), \tag{29}$$

where g is gravity and h_i is a given reference datum height. From standard work–energy principles, we equate the sum of the kinetic and potential energy in an arbitrary "configurational state 1" (at time $= t$), plus the work done from state 1 to a later (incremental) state, "configurational state 2" (time $= t + \Delta t$, configuration 2) to yield

$$T(t) + U(t) + \Delta W_{t \to t+\Delta t} = T(t + \Delta t) + U(t + \Delta t), \tag{30}$$

where $\Delta W_{t \to t+\Delta t}$ is the incremental work done in moving the system. Rearranging, one can immediately compute the incremental work

$$\Delta W_{t \to t+\Delta t} = (T(t + \Delta t) + U(t + \Delta t)) - (T(t) + U(t)). \tag{31}$$

The total amount of energy expended by the robotic system is

$$\int_{t=0}^{t=T} \Delta W_{t \to t+\Delta t} \approx \sum_{i=1}^{N_t} \Delta W_{t \to t+\Delta t}, \tag{32}$$

where N_t is the number of time intervals. The technological issue of how the power is supplied by the actuators, their location and control, is outside the scope of the current paper.

7. Machine learning for process planning

An important part of any modern robotic system is the use of Machine Learning algorithms for process planning. In order to illustrate a basic Machine Learning algorithm, consider the following objective function (at time $t = T$ where $\boldsymbol{s}$ is a path variable along the structure)

$$\Pi = \sqrt{\frac{\int_\Omega (\boldsymbol{r}^*(\boldsymbol{s}, T) - \boldsymbol{r}^{gen}(\boldsymbol{s}, T)) \cdot (\boldsymbol{r}^*(\boldsymbol{s}, T) - \boldsymbol{r}^{gen}(\boldsymbol{s}, T))\, d\boldsymbol{s}}{\int_\Omega \boldsymbol{r}^*(\boldsymbol{s}, T) \cdot \boldsymbol{r}^*(\boldsymbol{s}, T)\, d\boldsymbol{s}}}, \tag{33}$$

where $\boldsymbol{r}^*(\boldsymbol{s}, T)$ is a desired extruded filament structure at time T and $\boldsymbol{r}^{gen}(\boldsymbol{s}, T)$ is a generated extruded filament structure at time T. The system design vector ($\boldsymbol{\Lambda}$) and design component parameters are, for example, $\boldsymbol{\Lambda} = \{\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \Delta \boldsymbol{v}^{f-e}\}$, with constrained ranges of $\dot{\theta}_1^{(-)} \leq \dot{\theta}_1 \leq \dot{\theta}_1^{(+)}$, $\dot{\theta}_2^{(-)} \leq \dot{\theta}_2 \leq \dot{\theta}_2^{(+)}$, $\dot{\theta}_3^{(-)} \leq \dot{\theta}_3 \leq \dot{\theta}_3^{(+)}$ and $\Delta \boldsymbol{v}^{f-e(-)} \leq \Delta \boldsymbol{v}^{f-e} \leq \Delta \boldsymbol{v}^{f-e(+)}$. There are two characteristics of such a cost-function that make direct standard gradient type minimization schemes, such as Newton's method, difficult:

- (I) the incorporation of design search space restrictions, which render the objective function not continuously differentiable in design space and
- (II) the objective function is highly nonconvex, i.e. the system Hessian is not positive definite (invertible) throughout design space.

One way to minimize such objective functions is by following a two stage approach whereby (1) one determines promising optimal regions in parameter space using (non-derivative) "machine-learning" algorithms (such are evolutionary "genetic" algorithms, simulated annealing, etc.) and then (2) applies classical gradient-based schemes in locally convex regions, if the objective functions are smooth, since they are generally extremely efficient for the minimization of smooth convex functions. As indicated, the search for convex "pockets" of Π can be achieved by using "genetic" algorithms (GA), before applying classical gradient-based schemes.[3]

[3] An exhaustive review of these methods can be found in the texts of Luenberger [31] and Gill, Murray and Wright [32], while a state of the art can be found in Papadrakakis et al. [33].

Genetic algorithms, which can be considered as types of Machine Learning algorithms, are search methods based on the principles of natural selection, employing concepts of species evolution, such as reproduction, mutation and crossover. Implementation typically involves a randomly generated population of elemental strings (usually of fixed-length vectors) and "genetic" information, each of which represents a specific choice of system parameters. The population of individuals undergo "mating sequences" and other biologically-inspired events in order to find promising regions of the search space. In Zohdi [2–20], a genetic algorithm has been developed to treat a wide variety of nonconvex inverse problems involving various aspects of multiphysics/multiscale phenomena, and we refer the interested reader to those works. Specifically, the central idea is that the system parameters form a genetic string and a survival of the fittest algorithm is applied to a population of such strings. Such methods can be traced back, at least, to the work of John Holland (Holland [34]). For reviews of such methods, see, for example, Goldberg [35], Davis [36], Onwubiko [37], Kennedy and Eberhart [38] Lagaros et al. [39], Papadrakakis et al. [40–43] and Goldberg and Deb [44].

The overall process is: (a) a population (S) of different parameter sets are generated at random within the parameter space, each represented by a ("genetic") string of the system (N) parameters, (b) the performance of each parameter set is tested, (c) the parameter sets are ranked from top to bottom according to their performance, (d) the best parameter sets (parents) are mated pairwise producing two offspring (children), i.e. each best pair exchanges information by taking random convex combinations of the parameter set components of the parents' genetic strings and (e) the worst performing genetic strings are eliminated, then new replacement parameter sets (genetic strings) are introduced into the remaining population of best performing genetic strings and the process (a–e) is then repeated.

The term "fitness" of a genetic string is used to indicate the value of the objective function. The most fit genetic string is the one with the smallest objective function. The retention of the top fit genetic strings from a previous generation (parents) is critical, since if the objective functions are highly nonconvex (the present case), there exists a clear possibility that the inferior offspring will replace superior parents. The minimization of the cost function is guaranteed to be monotone (guaranteed improvement) with increasing generations, when the top parents are retained. Although not retaining the parents allows for an influx of new "genetic" information, there is no assurance of successive improvement, generation after generation. In the scientific literature, numerical studies imply that, for sufficiently large populations, the benefits of parent retention outweigh this advantage and any disadvantages of "inbreeding", i.e. a stagnant population. An implementation of such ideas is as follows (Zohdi [2–20]):

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i, (i = 1, 2, 3, \ldots, S)$: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \Lambda_4^i, \ldots, \Lambda_N^i\} \stackrel{\text{def}}{=} \{\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \Delta \boldsymbol{v}^{f-e}\}$
- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i), (i = 1, \ldots, S)$
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i, (i = 1, \ldots, S)$
- **STEP 4:** Mate nearest pairs and produce two offspring, $(i = 1, \ldots, S)$ $\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \Phi^{(I)} \boldsymbol{\Lambda}^i + (1 - \Phi^{(I)}) \boldsymbol{\Lambda}^{i+1}$, $\boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \Phi^{(II)} \boldsymbol{\Lambda}^i + (1 - \Phi^{(II)}) \boldsymbol{\Lambda}^{i+1}$
- **NOTE:** $\Phi^{(I)}$ and $\Phi^{(II)}$ are random numbers, such that $0 \leq \Phi^{(I)} \leq 1$, $0 \leq \Phi^{(II)} \leq 1$, which are different for each component of each genetic string
- **STEP 5:** Eliminate the bottom $M < S$ strings and keep top $K < N$ parents and top K offspring (K offspring $+K$ parents $+M = S$)
- **STEP 6:** Repeat STEPS 1–6 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **Option:** Rescale and restart search around best performing parameter set every few generations

Remark 1. For more details on this so-called "inheritance property" see Davis [36] or Kennedy and Eberhart [38]. In the presented algorithm, inbreeding is mitigated since, with each new generation, new parameter sets, selected at random within the parameter space, are added to the population. Previous numerical studies of the author (Zohdi [2–20]) have indicated that not retaining the parents is suboptimal due to the possibility that inferior offspring will replace superior parents. Additionally, parent retention is computationally less expensive, since these parameter sets do not have to be reevaluated (or ranked) in the next generation.

Table 2
Using the baseline objective function (Π). The results for the best performing gene (design parameter set) for the top 4 performers. Note that *alternate* designs were found that achieve nearly the same specified structure.

Design	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	Δv^{f-e}	Π
1	−10.11459492	−0.57032291	7.76821050	0.7066189934	0.0227394806
2	−9.68024579	−0.87110943	8.36491522	0.8740600133	0.0271929587
3	−10.29412361	−0.76018938	5.42563235	0.7255430364	0.0271930857
4	−7.27486780	−0.89944714	7.95193676	0.7870301992	0.0271931748

Remark 2. STEPS 1–6, which are associated with the genetic part of the overall algorithm, attempt to collect multiple local minima.[4] At first glance, it seems somewhat superfluous to retain even the top parents in such an algorithm. However, many studies have shown that the retention of the top old fit genetic strings is critical for proper convergence. With parent retention, the minimization of the cost function is guaranteed to be monotone with increasing generations, i.e. $\Pi(\mathbf{\Lambda}^{opt,I}) \geq \Pi(\mathbf{\Lambda}^{opt,I+1})$, where $\mathbf{\Lambda}^{opt,I+1}$ and $\mathbf{\Lambda}^{opt,I}$ are the best genetic strings from generations $I+1$ and I respectively. There is no such guarantee if the top parents are not retained. While the nonretention of parents allows more newer genetic strings to be evaluated in the next generation, numerical studies conducted thus far imply, for sufficiently large populations, that the benefits of parent retention outweigh this advantage, as well as any disadvantages of "inbreeding", i.e. a stagnant population. The case of inbreeding is circumvented in the current algorithm due to the fact that, with each new generation, new material designs, selected at random within the design space, are introduced into the population.

7.1. A machine learning/genetic algorithm numerical example

Consider the following Machine Learning parameters:

- The number of design variables: 4,
- The search domain for $\dot{\theta}_1$: $\dot{\theta}_1^- = -5 \leq \dot{\theta}_1 \leq -30 = \dot{\theta}_1^+$,
- The search domain for $\dot{\theta}_2$: $\dot{\theta}_2^- = -0.5 \leq \dot{\theta}_2 \leq -3 = \dot{\theta}_2^+$,
- The search domain for $\dot{\theta}_3$: $\dot{\theta}_3^- = 5 \leq \dot{\theta}_3 \leq 30 = \dot{\theta}_3^+$,
- The search domain for Δv^e: $\Delta v^{e,-} = 0.5 \leq \dot{\theta}_3 \leq 3.5 = \Delta v^{e,+}$,
- The population size per generation: 20,
- The number of parents to keep in each generation: 4,
- The number of children created in each generation: 4 and
- The number of completely new genes created in each generation: 12.

As a test structure, we generated an optimal structure, by using the one generated in the previous section at time $t = T$. In other words, we seek a set of robot system parameters that will deliver the result that we generated before, however by first starting with a random selection, and allowing the Machine Learning algorithm to find them. The optimal parameters that generated the structure were:

- $\dot{\theta}_1 = -10$ rad/s,
- $\dot{\theta}_2 = -1$ rad/s,
- $\dot{\theta}_3 = 10$ rad/s and
- $\Delta v^e = 1$ m/s.

Fig. 6 illustrates the results for the best performing gene (design parameter set) as a function of successive generations, as well as the average performance of the population of genes (designs). Table 2 shows the designs generated. Note that *alternate* designs were found that achieve nearly the same specified structure. We automatically restarted search around best performing parameter set every *10 generations*. We remark in passing that this algorithm is trivially parallelizable.

[4] 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 3
Using the enhanced objective function (Π_2): The results for the best performing gene (design parameter set) for the top 4 performers. Note that *alternate* designs were found that achieve nearly the same specified structure.

Design	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	Δv^{f-e}	Π_2
1	−9.32617373	−1.22102129	9.17809824	0.7920092700	0.0407938048
2	−12.19490694	−1.75739157	11.84792102	1.2028333827	0.0415696693
3	−11.09104121	−1.66922686	10.12596901	0.7729491688	0.0424840845
4	−11.42242661	−1.81467667	11.87834369	0.8770790982	0.0440106166

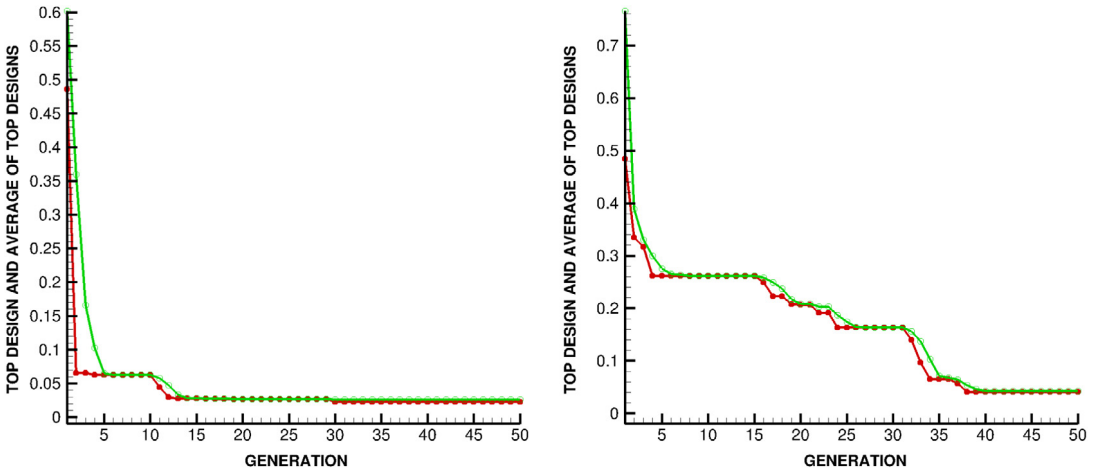

Fig. 6. Left—using the objective function (Π). Machine learning output, generation after generation. Shown is the best performing gene (design parameter set, in **red**) as a function of successive generations, as well as the average performance of the population of genes (designs, in **green**). Right—using the enhanced objective function (Π_2). Machine learning output, generation after generation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Remark. Without any difficulty, one could add more desired features to the objective function, such as a specified mass of the object (M^*):

$$\Pi_2 = \sqrt{\frac{1}{w_1 + w_2}\left(w_1 \frac{\int_\Omega (\boldsymbol{r}^*(\boldsymbol{s}, T) - \boldsymbol{r}^{gen}(\boldsymbol{s}, T)) \cdot (\boldsymbol{r}^*(\boldsymbol{s}, T) - \boldsymbol{r}^{gen}(\boldsymbol{s}, T))\, d\boldsymbol{s}}{\int_\Omega \boldsymbol{r}^*(\boldsymbol{s}, T) \cdot \boldsymbol{r}^*(\boldsymbol{s}, T)\, d\boldsymbol{s}} + w_2 \left(\frac{M^* - M^{gen}}{M^*} \right)^2 \right)}, \tag{34}$$

where $M^{gen} = \int_\Omega \rho A(\boldsymbol{s})\, d\boldsymbol{s}$ is the generated mass and w_1 and w_2 are design weights. Fig. 6 illustrates the results for the best performing gene (design parameter set). Table 3 shows the designs achieved with $w_1 = 1$ and $w_2 = 1$, and M^* being the mass generated by the base design $\dot{\theta}_1 = -10$ rad/s, $\dot{\theta}_2 = -1$ rad/s, $\dot{\theta}_3 = 10$ rad/s and $\Delta v^e = 1$ m/s. Note that *alternate* designs were found that achieve nearly the same specified structure.

8. Summary and extensions

There are quite a few directions in which such research can be extended. However, two issues standout: (1) enhancement of the Machine Learning algorithm, (2) enhanced material modeling and (3) enhanced extrusion control. We briefly discuss both issues.

8.1. Enhancement of the machine learning algorithm

After application of such a global search Machine Learning algorithm, one can apply a gradient-based method, if the objective function is sufficiently smooth in that region of the parameter space. In other words, if one has located a convex portion of the parameter space with a global genetic search, one can employ gradient-based procedures locally to minimize the objective function further, since the are generally much more efficient for convex optimization of

smooth functions. In such procedures, in order to obtain a new directional step for $\boldsymbol{\Lambda}$, one must solve the following system

$$[\boldsymbol{H}]\{\Delta\boldsymbol{\Lambda}\} = -\{\boldsymbol{g}\}, \tag{35}$$

where $[\boldsymbol{H}]$ is the Hessian matrix ($N \times N$), where $\{\Delta\boldsymbol{\Lambda}\}$ is the parameter increment ($N \times 1$), and $\{\boldsymbol{g}\}$ is the gradient ($N \times 1$). We shall not employ this second (post-genetic) stage in this work. Specifically, this is determined by forcing the gradient of $\nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda}) = \mathbf{0}$. Expanding (linearizing) around a first guess $\boldsymbol{\Lambda}^i$ yields

$$\nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda}^{i+1}) \approx \nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda}^i) + \nabla \left(\nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda}^i) \right) \cdot (\boldsymbol{\Lambda}^{i+1} - \boldsymbol{\Lambda}^i) + higher - order - terms \approx \mathbf{0} \tag{36}$$

or in more streamlined matrix notation, defining the Hessian, $[\boldsymbol{H}] = \nabla \left(\nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda}) \right)$ and $\{\boldsymbol{g}\} = \nabla_{\boldsymbol{\Lambda}} \Pi(\boldsymbol{\Lambda})$, thus

$$[\boldsymbol{H}]\{\Delta\boldsymbol{\Lambda}\} + \{\boldsymbol{g}\} = \mathbf{0}. \tag{37}$$

Following a standard Newton-type multivariate search, a new design increment is computed,

$$\Delta = (\Delta\Lambda_1, \Delta\Lambda_2, \ldots, \Delta\Lambda_N), \tag{38}$$

for a design vector, $\boldsymbol{\Lambda}$, by solving the following system, $[\boldsymbol{H}]\{\Delta\boldsymbol{\Lambda}\} = -\{\boldsymbol{g}\}$, where $[\boldsymbol{H}]$ is the Hessian matrix ($N \times N$), with components

$$H_{ij} = \frac{\partial^2 \Pi(\boldsymbol{\Lambda})}{\partial \Lambda_i \partial \Lambda_j}, \tag{39}$$

$\{\boldsymbol{g}\}$ is the gradient ($N \times 1$), with components

$$g_i = \frac{\partial \Pi(\boldsymbol{\Lambda})}{\partial \Lambda_i} \tag{40}$$

and where $\{\Delta\boldsymbol{\Lambda}\}$ is the design increment ($N \times 1$), with components $\Delta\Lambda_i$. After the design increment has been solved for, one then forms an updated design vector, $\boldsymbol{\Lambda}^{new} = \boldsymbol{\Lambda}^{old} + \Delta\boldsymbol{\Lambda}$, and the process is repeated until $\|\Pi\| \leq TOL$. Explicitly, the incremental system is

$$\begin{bmatrix}
\frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_1 \partial \Lambda_1} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_1 \partial \Lambda_2} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_1 \partial \Lambda_3} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_1 \partial \Lambda_4} & \ldots\ldots \\
\frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_2 \partial \Lambda_1} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_2 \partial \Lambda_2} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_2 \partial \Lambda_3} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_2 \partial \Lambda_4} & \ldots\ldots \\
\frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_3 \partial \Lambda_1} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_3 \partial \Lambda_2} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_3 \partial \Lambda_3} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_3 \partial \Lambda_4} & \ldots\ldots \\
\frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_4 \partial \Lambda_1} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_4 \partial \Lambda_2} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_4 \partial \Lambda_3} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_4 \partial \Lambda_4} & \ldots\ldots \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\
\frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_N \partial \Lambda_1} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_N \partial \Lambda_2} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_N \partial \Lambda_3} & \frac{\partial^2 \Pi(\Lambda)}{\partial \Lambda_N \partial \Lambda_4} & \ldots\ldots
\end{bmatrix}
\begin{Bmatrix}
\Delta\Lambda_1 \\ \Delta\Lambda_2 \\ \Delta\Lambda_3 \\ \Delta\Lambda_4 \\ \ldots\ldots \\ \ldots\ldots \\ \Delta\Lambda_N
\end{Bmatrix}
= -
\begin{Bmatrix}
\frac{\partial \Pi(\Lambda)}{\partial \Lambda_1} \\
\frac{\partial \Pi(\Lambda)}{\partial \Lambda_2} \\
\frac{\partial \Pi(\Lambda)}{\partial \Lambda_3} \\
\frac{\partial \Pi(\Lambda)}{\partial \Lambda_4} \\
\ldots\ldots \\ \ldots\ldots \\
\frac{\partial \Pi(\Lambda)}{\partial \Lambda_N}
\end{Bmatrix}. \tag{41}$$

The derivatives must often be computed numerically:

- For the first derivative of Π at $(\Lambda_1, \Lambda_2, \Lambda_3)$:

$$\frac{\partial \Pi}{\partial \Lambda_1} \approx \frac{\Pi(\Lambda_1 + \Delta\Lambda_1, \Lambda_2, \Lambda_3) - \Pi(\Lambda_1 - \Delta\Lambda_1, \Lambda_2, \Lambda_3)}{2\Delta\Lambda_1} \tag{42}$$

- For the second derivative at $(\Lambda_1, \Lambda_2, \Lambda_3)$:

$$\begin{aligned}
\frac{\partial}{\partial \Lambda_1} \left(\frac{\partial \Pi}{\partial \Lambda_1} \right) &\approx \frac{\left(\frac{\partial \Pi}{\partial \Lambda_1} \right) |_{\Lambda_1 + \frac{\Delta\Lambda_1}{2}, \Lambda_2, \Lambda_3} - \left(\frac{\partial \Pi}{\partial \Lambda_1} \right) |_{\Lambda_1 - \frac{\Delta\Lambda_1}{2}, \Lambda_2, \Lambda_3}}{\Delta\Lambda_1} \\
&= \frac{1}{\Delta\Lambda_1} \{ \left(\frac{\Pi(\Lambda_1 + \Delta\Lambda_1, \Lambda_2, \Lambda_3) - \Pi(\Lambda_1, \Lambda_2, \Lambda_3)}{\Delta\Lambda_1} \right) \\
&\quad - \left(\frac{\Pi(\Lambda_1, \Lambda_2, \Lambda_3) - \Pi(\Lambda_1 - \Delta\Lambda_1, \Lambda_2, \Lambda_3)}{\Delta\Lambda_1} \right) \}.
\end{aligned} \tag{43}$$

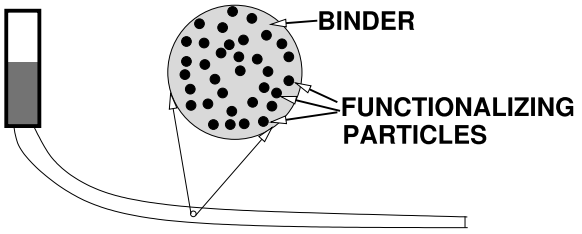

Fig. 7. A zoom on an extruded functionalized material.

- For the cross-derivative at (Λ_1, Λ_2):

$$\frac{\partial}{\partial \Lambda_2}\left(\frac{\partial \Pi}{\partial \Lambda_1}\right) \approx \frac{\partial}{\partial \Lambda_2}\left(\frac{\Pi(\Lambda_1+\Delta\Lambda_1, \Lambda_2, \Lambda_3)-\Pi(\Lambda_1-\Delta\Lambda_1, \Lambda_2, \Lambda_3)}{2\Delta\Lambda_1}\right)$$
$$\approx \frac{1}{4\Delta\Lambda_1\Delta\Lambda_2}\{(\Pi(\Lambda_1+\Delta\Lambda_1, \Lambda_2+\Delta\Lambda_2, \Lambda_3)-\Pi(\Lambda_1-\Delta\Lambda_1, \Lambda_2+\Delta\Lambda_2, \Lambda_3))$$
$$-(\Pi(\Lambda_1+\Delta\Lambda_1, \Lambda_2-\Delta\Lambda_2, \Lambda_3)-\Pi(\Lambda_1-\Delta\Lambda_1, \Lambda_2-\Delta\Lambda_2, \Lambda_3))\}. \tag{44}$$

An exhaustive review of these methods can be found in the texts of Luenberger [31] and Gill, Murray and Wright [32], while a state of the art can be found in Papadrakakis et al. [33] (see Fig. 7).

8.2. Enhanced material modeling

Often, to extend polymeric materials to applications beyond prototyping, second-phase particles are added to the heated mixture which solidify (cure) to form the overall material properties comprised of particles in a binding matrix when deposited on a substrate or building of a free-form 3D structure. The particles are used to "tune" the binding matrix properties to the desired overall state. Specifically, much of the commercial additive manufacturing processes are polymer-based, with second-phase particles added to enhance the properties of the binder, which is typically either (1) Polylactic acid or polylactide (PLA), which is a biodegradable thermoplastic aliphatic polyester or (2) Acrylonitrile butadiene styrene (ABS) which is a common thermoplastic polymer. In 2015, PLA had the second highest consumption volume of any bioplastic of the world. PLA is derived from renewable resources, such as plants (corn starch, sugarcane, etc.). ABS is a terpolymer that is significantly stronger than PLA. It is made by polymerizing styrene and acrylonitrile in the presence of polybutadiene. The styrene gives the plastic a reflective surface while the rubbery polybutadiene endows toughness. The overall properties are created by rubber toughening, where fine particles of elastomer are distributed throughout the rigid matrix. Typically, metal and ceramic particles are also added to endow specific mechanical, thermal, electrical and magnetic effective overall properties. Methods to estimate the overall macroscopic properties of heterogeneous materials dates back at least to the 1800's by the pioneering works of Maxwell [45,46] and Lord Rayleigh [47], and the seminal contribution of Hashin and Shtrikman in the form of effective property bounds during the 1960's (Hashin and Shtrikman [48,49], Hashin [50]). The so-called Hashin–Shtrikman bounds are the tightest possible bounds on isotropic effective responses, generated from isotropic microstructures, where the volumetric data and phase contrasts of the constituents are the only data known. For example, for the elastic bulk modulus:

$$\kappa^{*,-} \stackrel{\text{def}}{=} \kappa_1 + \frac{v_2}{\frac{1}{\kappa_2-\kappa_1}+\frac{3(1-v_2)}{3\kappa_1+4\mu_1}} \leq \kappa^* \leq \kappa_2 + \frac{1-v_2}{\frac{1}{\kappa_1-\kappa_2}+\frac{3v_2}{3\kappa_2+4\mu_2}} \stackrel{\text{def}}{=} \kappa^{*,+}, \tag{45}$$

and for the elastic shear modulus

$$\mu^{*,-} \stackrel{\text{def}}{=} \mu_1 + \frac{v_2}{\frac{1}{\mu_2-\mu_1}+\frac{6(1-v_2)(\kappa_1+2\mu_1)}{5\mu_1(3\kappa_1+4\mu_1)}} \leq \mu^* \leq \mu_2 + \frac{(1-v_2)}{\frac{1}{\mu_1-\mu_2}+\frac{6v_2(\kappa_2+2\mu_2)}{5\mu_2(3\kappa_2+4\mu_2)}} \stackrel{\text{def}}{=} \mu^{*,+}, \tag{46}$$

where κ_2 and κ_1 are the bulk moduli and μ_2 and μ_1 are the shear moduli of the respective phases ($\kappa_2 \geq \kappa_1$ and $\mu_2 \geq \mu_1$), and where v_2 is the second phase volume fraction. Phase 2 is the stiffer of the two constituents (which

usually corresponds to the particles).[5] One can form estimates for the effective properties by forming a convex combination of them, such as

$$\kappa^* \approx \phi\kappa^{*,+} + (1-\phi)\kappa^{*,-} \tag{47}$$

and

$$\mu^* \approx \phi\mu^{*,+} + (1-\phi)\mu^{*,-}, \tag{48}$$

where $0 \leq \phi \leq 1$ is a parameter such that:

- If $\phi = 0$ we have the lower bound,
- If $\phi = 1$ we have the upper bound and
- If $\phi = 1/2$ we have the average of the bounds.

Although the usual approach is to select $\phi = 0.5$, a more detailed discussion on the sharper selection of ϕ is provided in Zohdi [51] and can be calibrated from relatively simple experiments. One can post-process the effective bulk and shear modulus to obtain the effective Young's modulus $E^* = 2\mu^*(1+\nu^*) = 3\kappa^*(1-2\nu^*)$ and the effective Poisson ratio $\nu^* = \frac{3\kappa^*-2\mu^*}{2(3\kappa^*+\mu^*)}$. One can refine the simulation parameters, K^n and K^t by using E^* and μ^* by first estimating the normal (longitudinal) strain as

$$\epsilon^n \approx \frac{(\boldsymbol{v}_j^n - \boldsymbol{v}_i^n)\Delta t}{\Delta L} \tag{49}$$

where ΔL is the nominal unstretched distance between control points ($\|\boldsymbol{r}_i - \boldsymbol{r}_j\|$) and the normal (longitudinal) stress as

$$\sigma^n \approx E^*\epsilon^n \approx E^*\frac{(\boldsymbol{v}_j^n - \boldsymbol{v}_i^n)\Delta t}{\Delta L}, \tag{50}$$

leading to (multiplying stress times area)

$$\boldsymbol{\Psi}_i^{b,n} = K^n(\boldsymbol{v}_j^n - \boldsymbol{v}_i^n)\Delta t\, A_{ij}^c = E^*\frac{(\boldsymbol{v}_j^n - \boldsymbol{v}_i^n)\Delta t}{\Delta L}A_{ij}^c. \tag{51}$$

Thus, K^n is related to E^* through

$$K^n = \frac{E^*}{\Delta L}. \tag{52}$$

Following the same approach for the shear strain yields

$$\epsilon^s \approx \frac{(\boldsymbol{v}_j^t - \boldsymbol{v}_i^t)\Delta t}{\Delta L} \tag{53}$$

and

$$\sigma^t \approx 2\mu^*\epsilon^t \approx 2\mu^*\frac{(\boldsymbol{v}_j^t - \boldsymbol{v}_i^t)\Delta t}{\Delta L}. \tag{54}$$

Thus,

$$\boldsymbol{\Psi}^{b,t} = K^t(\boldsymbol{v}_j^t - \boldsymbol{v}_i^t)\Delta t\, A_{ij}^c = 2\mu^*\frac{(\boldsymbol{v}_j^t - \boldsymbol{v}_i^t)\Delta t}{\Delta L}A_{ij}^c. \tag{55}$$

Therefore, K^t is related to μ^* through

$$K^t = \frac{2\mu^*}{\Delta L}. \tag{56}$$

The material parameters E^* and μ^* are more physically tangible than K^n and K^t, and also have the caveat that they can be changed according to the increments of extruded material composition, which can changed as the material

[5] Note that no geometric or statistical information is required for the bounds.

is being extruded. Thus, in future more detailed analysis, the relations in Eqs. (51) and (55) are preferred. Similar expressions exist for estimates for the effective conductivity:

$$\underbrace{I\!K_1 + \frac{v_2}{\frac{1}{I\!K_2 - I\!K_1} + \frac{1-v_2}{3I\!K_1}}}_{I\!K^{*,-}} \leq I\!K^* \leq \underbrace{I\!K_2 + \frac{1-v_2}{\frac{1}{I\!K_1 - I\!K_2} + \frac{v_2}{3I\!K_2}}}_{I\!K^{*,+}}, \tag{57}$$

where the conductivity of phase 2 (with volume fraction v_2) is larger than phase 1 ($I\!K_2 \geq I\!K_1$). Usually, v_2 corresponds to the particle material, although there can be applications where the matrix is more conductive than the particles.[6] With respect to effective properties, we refer the reader to Torquato [52], Jikov et al. [53], Hashin [50], Mura [54], Markov [55] for theoretical aspects and for more computationally-oriented approaches, Ghosh [56], Ghosh and Dimiduk [57], Zohdi and Wriggers [19], Zohdi [10] and recently, Matous et al. [58] for a review of the state-of-the-art in multiscale methods for nonlinear heterogeneous materials. The determination of effective properties of such extruded mixtures is currently being pursued by the author. Numerical approaches for the determination of the effective properties are more accurate, although computationally expensive, in particular if one were to discretize the entire particle-laden heterogeneous domain with a very fine mesh. Thus, multiscale models, whereby effective property estimates are used for the majority of domain calculations and fine-scale microstructural models are used in selected critical regions are often employed. For example, see Zohdi and Wriggers [19] for reviews of such multiscale methods.

In other regimes and related manufacturing processes the particle-mixture may be fluid-like (as opposed to solid-like) and comprised of a highly saturated with an interstitial fluid or embedded within a fluid medium. In such cases, the fluid dynamics interaction with the particles can play a major role on the system behavior. This can also involve the evolution of heat. A spatio-temporal discretization can be used to solve coupled fluid particle systems, by employing Finite Element, Finite Difference or Finite Volume methods coupled to Discrete Elements (Oñate et al. [59–61], Avci and Wriggers [62], Leonardi et al. [63], Oñate et al. [61], Bolintineanu et al. [64] and Zohdi [7,15]). Examples of such particle–fluid systems are colloidal Ferrofluids consisting of ferromagnetic nanoscale particles embedded in an organic solvent.[7] Such fluids have a long history (Winslow [65,66]) in industrial applications as electrorheological fluids involving hydraulic valves, brakes, actuators and computer screens. Typically, the volume fraction of the particles is under five percent, and they are coated with a surfactant (oleic acid, citric acid, tetramethylammonium hydroxide, soy lecithin, etc.), in order to avoid agglomeration by mitigating near-field interaction effects between particles. We refer the reader to Albrecht et al. [67], Andelman and Rosensweig [68] and Berger et al. [69] for reviews. Extensions to this type of analysis could include parameter studies on the changes in dynamics due to strong magnetic fields. Although magnetic fields are somewhat less used than electric fields for deposition, they have certain unique utility by being able to "bend" sprays and depositions (see Martin [70,71] for the state of the art).

8.3. Extruded electromagnetic control

Accordingly, electromagnetic control of such processes is emerging as a viable technique for a wide variety of metal-infused filaments. To extend the work, the dynamics are governed by

$$m_i \ddot{\boldsymbol{r}}_i = \boldsymbol{\Psi}_i^b + \boldsymbol{\Psi}_i^{e+m} \stackrel{\text{def}}{=} \boldsymbol{\Psi}_i^{tot}(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{N_{cp}}). \tag{60}$$

[6] The effective density of the mixture is

$$\rho^* \stackrel{\text{def}}{=} \langle \rho(\boldsymbol{x}) \rangle_V \stackrel{\text{def}}{=} \frac{1}{V}\int_V \rho(\boldsymbol{x})\, dV = \frac{1}{V}\left(\int_{V_f} \rho_m\, dV + \int_{V_p} \rho_p\, dV\right) = v_m \rho_m + v_p \rho_p \tag{58}$$

where v_m and v_p are the volume fractions of the matrix and particles, respectively. The volume fractions have to sum to unity:

$$v_m + v_p = 1 \Rightarrow v_m = 1 - v_p. \tag{59}$$

[7] The particles are usually an iron-based compound, for example hematite or magnetite.

We decompose the electromagnetic forces into two parts: (1) Lorentz forces (for charged/electrified materials) and (2) magnetic forces (for magnetic materials). In mathematical form,

$$\boldsymbol{\Psi}_i^{e+m} = \boldsymbol{\Psi}_i^{lor,e+m} + \boldsymbol{\Psi}_i^{mag} = \underbrace{q_i(\boldsymbol{E}^{ext} + \boldsymbol{v}_i \times \boldsymbol{B}^{ext})}_{\boldsymbol{\Psi}_i^{lor,e+m}} + \boldsymbol{\Psi}_i^{mag}, \tag{61}$$

where Lorentz-induced forces from independent external fields are $\boldsymbol{E}^{ext}$ and $\boldsymbol{B}^{ext}$ is $\boldsymbol{\Psi}_i^{lor,e+m}$. The terms $\boldsymbol{E}^{ext}$ and $\boldsymbol{B}^{ext}$ are considered to be externally controlled and uncoupled from one another.[8] We note that intrinsic magnetic (non-electromagnetic) forces can be described by $\boldsymbol{\Psi}^{mag} = \nabla(\gamma \boldsymbol{B}^{ext} \cdot \boldsymbol{B}^{ext})$ (independently of the Lorentz forces), where γ is a material parameter that is related to the magnetization (magnetic dipole properties, susceptibility, permeability, moment density, etc.) of the element (see Feynman et al. [73], Cullity and Graham [74], Boyer [75] or Jackson [72]).

Both of the mentioned issues above are actively being pursued by the author. In closing, we raise a final issue which can affect manufacturing quality-fluid instabilities. Breakup of long slender fluid columns is observed in a variety of applications. In the analysis of standard fluids, the breakup of a long column of fluid with slight perturbations (longitudinal waviness) was first investigated experimentally by Plateau in 1873, who found that a vertically falling stream of water will break up into drops if its wavelength is greater than approximately 3.13–3.18 times its diameter. Subsequently, Rayleigh analytically proved that a wavy falling column of non-viscous liquid (with circular cross-section) should break up into drops if its wavelength exceeded its circumference. This type of instability is driven by surface tension, which forces fluids to minimize their surface area. See Papageorgiou [76] and Eggers [77] for more details. In the case of an extruded viscous medium, the degree of fluidity/viscosity influences surface tension. Thus, depending on processing parameters, such instabilities can be of concern. Strong adoption of computational methods in additive manufacturing and 3D printing has the potential to bring a level of systematic analysis that can make it a reliable large-scale manufacturing process (see Avila et al. [78], Dornfeld et al. [79], Vijayaraghavan et al. [80], Vijayaraghavan and Dornfeld [81], Dornfeld and Lee [82] and Dornfeld [83]). From the view of practical additive manufacturing simulation tools, there are several hurtles. Additive manufacturing, which is based on deposition, by itself is inadequate for high-precision applications and needs to be combined with classical manufacturing processes, as high-precision surface milling, as outlined in a recent US National Academies Report (Zohdi and Dornfeld [20]). Thus, it is important to realize that guided by simulation, additive and subtractive processes should be combined to model deposition and removal of material, while simultaneously controlling the tolerances, surface finish and, ultimately, product quality. The development of models and simulation tools for additive and subtractive processes is of current actively being investigated by the author.

Generally, new types of advanced manufacturing processes are rapidly evolving. The need for computational methods in manufacturing industry has grown monumentally in the last decade. Unfortunately, the outreach from the computational mechanics community does not reflect this trend adequately. In particular, an important topic that is crucial to this maximizing this field's potential in manufacturing is the concept of a digital twin, which refers to a digital replica of physical systems. Digital twins blend artificial intelligence, machine learning and software analytics with data to create living digital computer models that update and change in tandem with their physical counterparts. Updates to the digital twin are made continuously in near real-time-which necessitates rapid computation and sensor fusion. Such a paradigm also integrates historical data from past machine usage to factor into its digital model in order to optimize the operation and maintenance of systems and manufacturing processes. For example, when sensors collect data from a connected device, the sensor data can be used to update a "digital twin" copy of the device's state in real time. Often the phrase "device shadow" is also used and can be combined with historical data and human expertise to improve the outcome of prognostics. Examples in other segments of industry include power systems, such as aircraft engines, wind turbines, HVAC control systems, etc. Thus, one critical focus of computational manufacturing research should be to use data analytics and machine learning to control manufacturing processes via simultaneous numerical simulations. Waiting decades for codes to slowly mature, as it did for PDE-discretization type codes (Finite Element Methods), is suboptimal. One cannot expect industry to have the resources to achieve this on their own. As industry moves towards becoming a high-tech, digitized industry, the workforce needs to build substantially different data and modeling skills that focus on operations (e.g., understanding of uncertainty quantification and error propagation over multiple stages in an industrial system), rather than just information management technology. Unfortunately, the

[8] For the velocity ranges in the present applications (significantly below the speed of light), self-induced magnetic fields between elements are unimportant (Jackson [72]).

skills needed to integrate disparate systems together are not being developed quickly or robustly enough in industry alone nor have they, until very recently, been addressed properly in the university environment, which is an obvious bottleneck for industrial implementation. However, several industry-academia partnerships have recently started to evolve, such as the huge Manufacturing USA network initiative in the United States, which signals the sincerity of governments to rectify this problem. The author is current working on research and methods towards these goals, and the presented work is a step in this direction.

References

[1] Y. Huang, M.C. Leu, J. Mazumdar, A. Donmez, Additive manufacturing: current state, future potential, gaps and needs, and recommendation, J. Manuf. Sci. Eng. 137 (2015) 014001-1.
[2] T.I. Zohdi, An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids, Internat. J. Numer. Methods Engrg. 53 (2002) 1511–1532.
[3] T.I. Zohdi, Modeling and simulation of a class of coupled thermo-chemo-mechanical processes in multiphase solids, Comput. Methods Appl. Mech. Engrg. 193 (6–8) (2004) 679–699.
[4] T.I. Zohdi, Modeling and direct simulation of near-field granular flows, Internat. J. Solids Structures 42 (2) (2004) 539–564.
[5] T.I. Zohdi, A computational framework for agglomeration in thermo-chemically reacting granular flows, Proc. R. Soc. 460 (2052) (2004) 3421–3445.
[6] T.I. Zohdi, Charge-induced clustering in multifield particulate flow, Internat. J. Numer. Methods Engrg. 62 (7) (2005) 870–898.
[7] T.I. Zohdi, Computation of strongly coupled multifield interaction in particle-fluid systems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 3927–3950.
[8] T.I. Zohdi, On the computation of the coupled thermo-electromagnetic response of continua with particulate microstructure, Internat. J. Numer. Methods Engrg. 76 (2008) 1250–1279.
[9] T.I. Zohdi, On the dynamics of charged electromagnetic particulate jets, Arch. Comput. Methods Eng. 17 (2) (2010) 109–135.
[10] T.I. Zohdi, Simulation of coupled microscale multiphysical-fields in particulate-doped dielectrics with staggered adaptive FDTD, Comput. Methods Appl. Mech. Engrg. 199 (2010) 79–101.
[11] T.I. Zohdi, Numerical simulation of charged particulate cluster-droplet impact on electrified surfaces, J. Comput. Phys. 233 (2013) 509–526.
[12] T.I. Zohdi, Rapid simulation of laser processing of discrete particulate materials, Arch. Comput. Methods Eng. 20 (2013) 309–325.
[13] T.I. Zohdi, A direct particle-based computational framework for electrically-enhanced thermo-mechanical sintering of powdered materials, Math. Mech. Solids 19 (1) (2014) 93–113.
[14] T.I. Zohdi, Additive particle deposition and selective laser processing-a computational manufacturing framework, Comput. Mech. 54 (2014) 171–191.
[15] T.I. Zohdi, Embedded electromagnetically sensitive particle motion in functionalized fluids, Comput. Part. Mech. 1 (2014) 27–45.
[16] T.I. Zohdi, Modeling and simulation of cooling-induced residual stresses in heated particulate mixture depositions, Comput. Mech. 56 (2015) 613–630.
[17] T.I. Zohdi, Modeling and efficient simulation of the deposition of particulate flows onto compliant substrates, Internat. J. Engrg. Sci. 99 (2015) 74–91. http://dx.doi.org/10.1016/j.ijengsci.2015.10.012.
[18] T.I. Zohdi, Modeling and simulation of laser processing of particulate-functionalized materials, Arch. Comput. Methods Eng. (2015) 1–25. http://dx.doi.org/10.1007/s11831-015-9160-1.
[19] T.I. Zohdi, P. Wriggers, Introduction to Computational Micromechanics, Springer-Verlag, 2005 (Book, 2008) Second Reprinting (Peer Reviewed).
[20] T.I. Zohdi, D.A. Dornfeld, Future Synergy between Computational Mechanics and Advanced Additive Manufacturing. 2015. US National Academies Report: http://sites.nationalacademies.org/cs/groups/pgasite/documents/webpage/pga_166813.pdf.
[21] K.H. Hunt, Kinematic Geometry of Mechanisms, in: Oxford Engineering Science Series, 1979.
[22] R.S. Hartenberg, J. Denavit, Kinematic Synthesis of Linkages, McGraw-Hill, New York, 1964.
[23] L.L. Howell, Compliant Mechanisms, John Wiley & Sons, 2001.
[24] J.M. McCarthy, G.S. Soh, Geometric Design of Linkages, Springer, New York, 2010.
[25] J.M. McCarthy, Introduction to Theoretical Kinematics, MIT Press, Cambridge MA, 1990.
[26] F. Reuleaux, The Kinematics of Machinery, (trans. and annotated by A. B. W. Kennedy), reprinted by Dover, New York, 1876 (1963).
[27] G.N. Sandor, A.G. Erdman, Advanced Mechanism Design: Analysis and Synthesis, Vol. 2, Prentice-Hall, Englewood Cliffs, NJ, 1984.
[28] A. Slocum, Precision Machine Design, SME, 1992.
[29] C.H. Suh, C.W. Radcliffe, Kinematics and Mechanism Design, John Wiley and Sons, New York, 1978.
[30] J.J. Uicker, G.R. Pennock, J.E. Shigley, Theory of Machines and Mechanisms, Oxford University Press, New York, 2003.
[31] D. Luenberger, Introduction to Linear & Nonlinear Programming, Addison-Wesley, Menlo Park, 1974.
[32] P. Gill, W. Murray, M. Wright, Practical Optimization, Academic Press, 1995.
[33] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, V. Plevris, Large scale structural optimization: Computational methods and optimization algorithms, Arch. Comput. Methods Eng. State Art Rev. 8 (3) (2001) 239–301.
[34] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Mich, 1975.
[35] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
[36] S.H. Davis, Theory of Solidification, Cambridge University Press, 2001.

[37] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.
[38] J. Kennedy, R. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
[39] N. Lagaros, M. Papadrakakis, G. Kokossalakis, Structural optimization using evolutionary algorithms, Comput. Struct. 80 (2002) 571–589.
[40] M. Papadrakakis, N. Lagaros, G. Thierauf, J. Cai, Advanced solution methods in structural optimisation using evolution strategies, Eng. Comput. J. 15 (1) (1998) 12–34.
[41] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Structural optimization using evolution strategies and neutral networks, Comput. Methods Appl. Mech. Engrg. 156 (1) (1998) 309–335.
[42] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Optimization of large-scale 3D trusses using evolution strategies and neural networks, Int. J. Space Struct. 14 (3) (1999) 211–223.
[43] M. Papadrakakis, J. Tsompanakis, N. Lagaros, Structural shape optimisation using evolution strategies, Eng. Optim. 31 (1999) 515–540.
[44] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 121–124.
[45] J.C. Maxwell, On the dynamical theory of gases, Philos. Trans. Soc. London 157 (1867) 49.
[46] J.C. Maxwell, A Treatise on Electricity and Magnetism, third ed., Clarendon Press, Oxford, 1873.
[47] J.W. Rayleigh, On the influence of obstacles arranged in rectangular order upon properties of a medium, Phil. Mag. 32 (1892) 481–491.
[48] Z. Hashin, S. Shtrikman, On some variational principles in anisotropic and nonhomogeneous elasticity, J. Mech. Phys. Solids 10 (1962) 335–342.
[49] Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, J. Mech. Phys. Solids 11 (1963) 127–140.
[50] Z. Hashin, Analysis of composite materials: a survey, ASME J. Appl. Mech. 50 (1983) 481–505.
[51] T.I. Zohdi, An explicit macro-micro phase-averaged stress correlation for particle-enhanced composite materials in loaded structures, Internat. J. Engrg. Sci. (2016). http://dx.doi.org/10.1016/j.ijengsci.2016.09.005.
[52] S. Torquato, Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Springer-Verlag, New York, 2002.
[53] V.V. Jikov, S.M. Kozlov, O.A. Olenik, Homogenization of Differential Operators and Integral Functionals, Springer-Verlag, 1994.
[54] T. Mura, Micromechanics of Defects in Solids, second ed., Kluwer Academic Publishers, 1993.
[55] K.Z. Markov, Elementary micromechanics of heterogeneous media, in: K.Z. Markov, L. Preziozi (Eds.), Heterogeneous Media: Micromechanics Modeling Methods and Simulations, Birkhauser, Boston, 2000, p. 1162.
[56] S. Ghosh, Micromechanical Analysis and Multi-Scale Modeling Using the Voronoi Cell Finite Element Method, CRC Press/Taylor & Francis, 2011.
[57] S. Ghosh, D. Dimiduk, Computational Methods for Microstructure-Property Relations, Springer, NY, 2011.
[58] K. Matous, M.G.D. Geers, V. Kouznetsova, A. Gillman, A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials, J. Comput. Phys. 330 (2017) 192–220.
[59] E. Oñate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Engrg. 197 (19–20) (2008) 1777–1800.
[60] E. Oñate, M.A. Celigueta, S.R. Idelsohn, F. Salazar, B. Suárez, Possibilities of the particle finite element method for fluid-soil–structure interaction problems, Comput. Mech. 48 (2011) 307–318.
[61] E. Oñate, M.A. Celigueta, S. Latorre, G. Casas, R. Rossi, J. Rojek, Lagrangian analysis of multiscale particulate flows with the particle finite element method, Comput. Part. Mech. 1 (1) (2014) 85–102.
[62] B. Avci, P. Wriggers, A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows, J. Appl. Mech. 79 (2012) 010901-(1-7).
[63] A. Leonardi, F.K. Wittel, M. Mendoza, H.J. Herrmann, Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions, Comput. Part. Mech. 1 (1) (2014) 3–13.
[64] D.S. Bolintineanu, G.S. Grest, J.B. Lechman, F. Pierce, S.J. Plimpton, P.R. Schunk, Particle dynamics modeling methods for colloid suspensions, Comput. Part. Mech. 1 (3) (2014) 321–356.
[65] W.M. Winslow, Method and means for translating electrical impulses into mechanical force. U.S. Patent 2,417,850, 1947.
[66] W.M. Winslow, Induced fibration of suspensions, J. Appl. Phys. 20 (12) (1949) 1137–1140.
[67] T. Albrecht, C. Bührer, M. Fahnle, K. Maier, D. Platzek, J. Reske, First observation of ferromagnetism and ferromagnetic domains in a liquid metal, Appl. Phys. A 65 (2) (1997) 215.
[68] D. Andelman, R.E. Rosensweig, The Phenomenology of modulated phases: From magnetic solids and fluids to organic films and polymers, in: Yoav Tsori, Ullrich Steiner (Eds.), Polymers, Liquids and Colloids in Electric Fields: Interfacial Instabilities, Orientation and Phase Transitions, World Scientific, ISBN: 978-981-4271-68-4, 2009, p. 156.
[69] P. Berger, N.B. Adelman, K.J. Beckman, D.J. Campbell, A.B. Ellis, G.C. Lisensky, Preparation and properties of an aqueous ferrofluid, J. Chem. Educ. (ISSN: 0021-9584) 76 (7) (1999) 943–948. http://dx.doi.org/10.1021/ed076p943.
[70] P. Martin, Handbook of Deposition Technologies for Films and Coatings, third ed., Elsevier, 2009.
[71] P. Martin, Introduction to Surface Engineering and Functionally Engineered Materials, Scrivener and Elsevier, 2011.
[72] J.D. Jackson, Classical Electrodynamics, third ed., Wiley, 1998.
[73] Richard P. Feynman, Robert B. Leighton, Matthew Sands, The Feynman Lectures on Physics 2, ISBN: 0-8053-9045-6, 2006.
[74] Cullity, C.D. Graham, Introduction to Magnetic Materials, second ed., Wiley-IEEE Press, ISBN: 0-471-47741-9, 2008, p. 103.
[75] Timothy H. Boyer, The force on a magnetic dipole, Amer. J. Phys. 56 (8) (1988) 688–692. http://dx.doi.org/10.1119/1.15501. Bibcode:1988AmJPh.56688B.
[76] D.T. Papageorgiou, On the breakup of viscous liquid threads, Phys. Fluids 7 (7) (1995) 1521–1529.
[77] J. Eggers, Nonlinear dynamics and breakup of free-surface flows, Rev. Modern Phys. 69 (3) (1997) 865.

[78] M. Avila, J. Gardner, C. Reich-Weiser, . Tripathi, A. Vijayaraghavan, D.A. Dornfeld, Strategies for Burr Minimization and Cleanability in Aerospace and Automotive Manufacturing. SAE Technical Paper 2005-01-3327, 2005, 2005. http://dx.doi.org/10.4271/2005-01-3327.
[79] D. Dornfeld, P. Wright, A. Vijayaraghavan, M. Helu, Enabling manufacturing research through interoperability, Trans. North Am. Manuf. Res. Inst. SME 2009 (37) (2009) 443–450.
[80] A. Vijayaraghavan, L. Huet, D. Dornfeld, W. Sobel, B. Blomquist, M. Conley, Addressing process planning and verification issues with mtconnect, Trans. North Am. Manuf. Res. Inst. SME 2010 (38) (2010) 557–564.
[81] . Vijayaraghavan, D. Dornfeld, Automated energy monitoring of machine tools, CIRP Ann. 59 (1) (2010) 21–24.
[82] D.A. Dornfeld, D.E. Lee, Precision Manufacturing, Springer-Verlag, New York, 2008.
[83] D. Dornfeld, Green Manufacturing –Fundamentals and Applications, Springer, New York, 2013.