ELSEVIER

Rapid machine-learning enabled design and control of precise next-generation cryogenic surgery in dermatology[☆]

Tarek I. Zohdi[a,*], Mona Zohdi-Mofid[b]

[a] *Department of Mechanical Engineering, University of California, Berkeley, CA, 94720-1740, USA*
[b] *Sharp Community Medical Group, 8929 University Center Ln #202, San Diego, CA 92122, USA*

Available online xxxx

Abstract

In the field of dermatology, the use of cryogenic processes, such as cryoablation, cryotherapy, etc., have grown dramatically over the last decade. This usually entails using a cryoprobe to freeze and destroy unwanted tissue, such as cancer cells. The focus of this work is to develop a digital-twin (a digital replica) of the performance of a cryogenic probe, which can be used to pre-plan and optimize surgical procedures, in order to maximize successful outcomes. Specifically, we model the optimal cryoprobe-induced cooling protocol needed to eliminate cells/tissue in specific regions, while minimizing damage to nearby tissue. The modeling approach is to develop mathematical surface point-source heat extraction kernels and then to create optimal surface patterns that the cryoprobe induces, by arranging the point-sources accordingly. Spatial and temporal control of the heat extraction is modeled. The entire subdermal thermal field is then constructed by superposing the solutions, enabling precise cryogenic treatment. Finally, a Machine Learning Algorithm (MLA) is then applied to optimize the set of parameters to deliver a precise response, making it an ideal real-time surgical tool.
© 2023 Elsevier B.V. All rights reserved.

Keywords: Cryogenics; Dermatology; Digital-twin; Machine-learning

1. Introduction

Over the last few years, cryogenic technologies have been developed in a number of fields. In particular, in the biological sciences, cryogenic dermatology, cyrosurgery, biotissue preservation and vaccine storage have become commonplace. Cryogenic processes involve temperatures below 120 K (-153 C), which is the logical dividing line to "refrigeration", since that temperature is associated with the boiling of permanent gases such as helium, neon, nitrogen, oxygen, etc. In this work, we are primarily interested in the subfield of cryodermatology. For dermatological applications, such as cryoablation and cryotherapy, heat extraction is achieved via the use of liquid nitrogen or argon gas to freeze and remove unwanted tissue, such as precancerous skin growths (actinic keratoses) and dermal cancers, such as basal cell and squamous cell carcinomas. Unwanted cells die after the frozen tissue thaws. By far the largest use is in dermatology for acne, warts, dermatofibroma, hemangioma, keloid
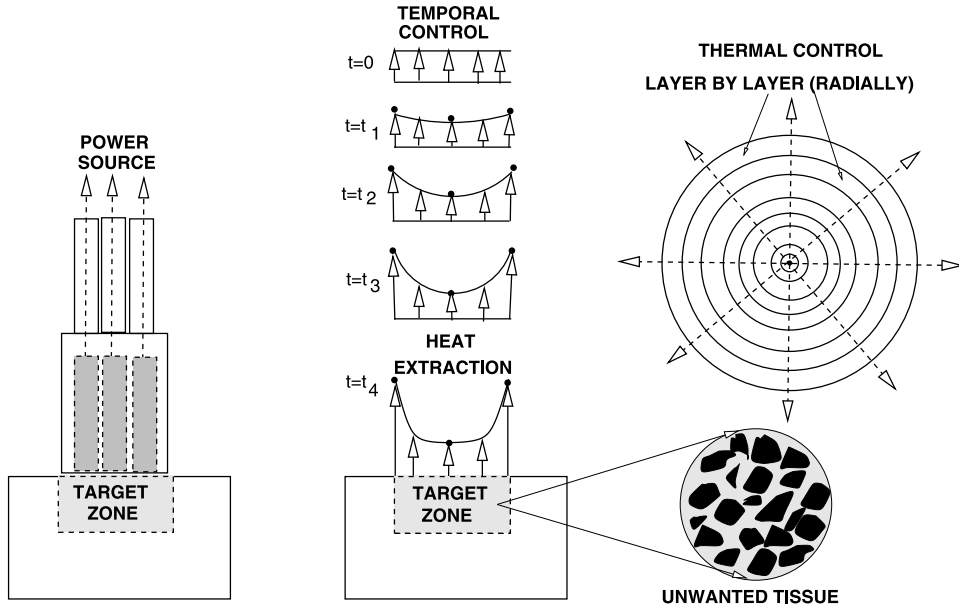
Fig. 1. The action of a precise cryoprobe with radial control of cooling and temporal modulation. This induces temporally evolving and spatially nonuniform heat extraction to induce desired subsurface cooling. The temporal changes and the spatial distribution are parameters that will be optimized by a Machine-Learning Algorithm.

(hypertrophic scar), molluscum contagiosum, myxoid cyst, pyogenic granuloma, seborrheic keratoses, skin tags, small skin cancers, etc.

In the coarsest approach, liquid nitrogen is applied with a cotton swab, while in the most refined approach is to use a cryoprobe. Recent advances in technology have allowed for the use of argon gas to drive ice formation using the Joule–Thomson effect. This gives physicians excellent control of the ice and minimizes complications using ultra-thin 17 gauge cryoneedles. Ideally, it is designed to be highly localized, freezing tissue and causing cell death, since when the temperature falls below a certain level ice-crystals form and tear the cells apart. Additionally, unwanted tissue is further harmed when the blood vessels that supply the tissue begin to freeze. The development of highly controllable cryoprobes, with liquid nitrogen or argon gas flowing through the probe have opened us a variety of possibilities. Thus, while cryosurgery can cause side effects, they are likely to be less severe than those from other local treatments, such as surgery or radiation therapy. For more details, we refer the reader to [1–8]. *However, one can still improve the method, which is the subject of this work*. The main side effect to avoid is the unintended damage of surrounding healthy tissue. The focus of this work is to develop a digital-twin (a digital replica) of the performance of a cryogenic probe, which can be used to preplan and optimize surgical procedures, in order to maximize successful outcomes. Specifically, we model the optimal cryoprobe-induced cooling protocol needed to eliminate cells/tissue in specific regions, while minimizing damage to nearby tissue. The modeling approach is to develop mathematical surface point-source heat extraction kernels and then to create optimal surface patterns that the cryoprobe should be induced by arranging the point-sources accordingly. Spatial and temporal control of the heat extraction is modeled. The entire subdermal thermal field is then constructed by superposing the solutions, enabling precise cryogenic treatment. Specifically, as shown in Fig. 1, two sets of parameters will be varied:

- **Temporal Parameters:** As shown in Fig. 1, heat extraction intensity will be varied temporally, by breaking the total simulation time into time-slabs, each with a different intensity level.
- **Spatial (Radial) Parameters:** As shown in Fig. 1, spatial profiles will be varied by breaking the domain into spatial slabs, which vary radially, each with a different intensity level.

A Machine Learning Algorithm (MLA) is then applied to optimize the set of parameters to deliver a precise response.
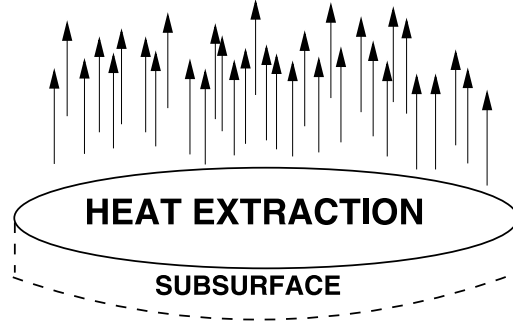
Fig. 2. Estimation of the power needed to induce a cryogenic state in a targeted pattern.

2. Qualitative system response estimation

In order to understand the fundamental behavior of the target for cryogenic processing, consider a targeted disk-shaped targeted domain in Fig. 2 of mass $m = \rho V_{tar}$, where ρ is the mass density material, V_{tar} is the targeted volume, ρC is the thermal mass density (C is the heat capacity) and S is the power extraction (Watts). Note we use the subscript "tar" throughout the work to indicate a quantity associated with the target volume.

The simplest possible estimate for the system response is given by a "lumped mass" analysis of the first law of thermodynamics, assuming the energy is entirely contained within the targeted pattern domain and that the temperature in the targeted pattern is uniform (also ignoring conductive losses)[1]:

$$mC\dot{\theta} = \rho C V_{tar}\dot{\theta} = S, \tag{2.1}$$

where θ is the temperature and $S < 0$ is the power extracted (Watts) and $\theta(t = 0) = \theta_o$, which in this simple model is assumed to equal the surrounding temperature, θ_s. The solution to this differential equation is

$$\theta(t) = \theta_s + \frac{St}{\rho C V_{tar}}. \tag{2.2}$$

Setting $\theta(t) = \theta_{tar}$ yields a simple relationship that correlates the power input, targeted pattern size, heat capacity desired temperature and processing time, which can be inverted to yield

$$t_{tar} = \frac{(\theta_{tar} - \theta_s)(\rho C)V_{tar}}{S}. \tag{2.3}$$

One can improve this estimate by including conductive losses to the surroundings, given by an overall energy balance (First Law of Thermodynamics) is

$$\dot{\theta} = \frac{\mathbb{K}}{\rho C V_{tar}}\frac{(\theta_s - \theta)}{h}A_c + \frac{S}{\rho C V_{tar}}, \tag{2.4}$$

where $\mathbb{K}$ is the conductivity, A_c is the area surrounding the targeted pattern where conduction takes place, θ_s is the average temperature surrounding the targeted pattern (assuming the simplification that $\theta_s = \theta_o$ remains unchanged) and h is a length-scale for conduction. For the case of a constant source term, this model can be solved analytically

$$\theta(t) = \theta_s + \frac{Sh}{\mathbb{K}A_c}\left(1 - e^{-\frac{\mathbb{K}A_c t}{\rho C V_{tar}h}}\right). \tag{2.5}$$

Eq. (2.5) connects the temperature, power extraction, conductivity, contact area and time. Setting the desired temperature at a desired time to be $\theta(t_{tar}) = \theta_{tar}$, the needed extraction strength is

$$S_{tar} = \frac{(\theta_{tar} - \theta_s)\mathbb{K}A_c}{h\left(1 - e^{-\frac{\mathbb{K}A_c t_{tar}}{\rho C V_{tar}h}}\right)}. \tag{2.6}$$

---

[1] A lumped mass model, assuming a uniform temperature within a targeted pattern, is dictated by the Biot number. The Biot numbers has to be significantly less than unity for such an approximation to be reasonable. This is generally not the case, however, the solution still provides qualitatively useful information.

We have the following observations:

- The cooling time for the temperature is dictated by the ratio of conduction to heat capacity, $\frac{\mathbb{K}A_c}{\rho C V_{tar}h}$.
- At steady-state, $e^{-\frac{\mathbb{K}A_c t}{(\rho C)V_{tar}h}} \to 0$, and

$$\theta(t) = \theta_s + \frac{Sh}{\mathbb{K}A_c},\tag{2.7}$$

  which indicates that the ratio of S to $\frac{\mathbb{K}A_c}{h}$ dictates the steady state temperature.
- Highly conductive surroundings, $\mathbb{K} \to \infty$, $\theta(t) = \theta_s$, yields instantaneous conductive losses-pulling the energy out of the targeted pattern.
- Poorly conducting surroundings, $\mathbb{K} \to 0$, $\theta(t) = \theta_s + \frac{St}{(\rho C)V_{tar}}$, thus trapping all of the energy.

The processing time is determined by inverting the relationship:

$$t^{des} = -\frac{(\rho C)V_{tar}h}{\mathbb{K}A_c} Ln\left(1 - (\theta^{des} - \theta_s)\frac{\mathbb{K}A_c}{S^{des}h}\right).\tag{2.8}$$

The derived lumped-mass relations provide estimates, in fact lower bounds, on the processing time, and provide a starting point for the more in depth analysis later in the presentation.

Remark 1. The ratio of the power input/extraction to the conductive flux yields

$$\frac{POWER}{CONDUCTION} = \frac{hS}{\mathbb{K}(\theta_S - \theta)A_c} \propto \frac{h\mathcal{O}(10^3)}{\mathcal{O}(1)\mathcal{O}(10^3)A_c} \propto \frac{h}{A_c},\tag{2.9}$$

where the orders of magnitude correspond to nominal values encountered in dermatology. This indicates that the length scale of the conduction gradient, which is closely related to the targeted pattern size, influences whether conduction dominates. Since $A_c \propto h^2$,

$$\frac{POWER}{CONDUCTION} \propto \frac{1}{h},\tag{2.10}$$

we see that for small targeted patterns, 0.001 m $\leq h \leq 0.01$ m, the power extraction dominates conduction, thus facilitating the temperature drop. Thus far, we have not explicitly identified the dependency of the density (ρ) and conductivity ($\mathbb{K}$). This is undertaken next.

3. Construction of power-induced thermal field with heat-extraction-kernels

The approach taken is to construct a cryo-pattern by decomposing it into individual point-load heat extraction kernels on an semi-infinite half space which are summed to yield the entire thermal field produced by the power input/extraction (Fig. 3). The first law of thermodynamics yields (ignoring the effects of stress and phase-transformations)

$$\rho\dot{w} = \rho C\frac{\partial\theta}{\partial t} = -\nabla \cdot \boldsymbol{q} + \mathcal{S}^{vol},\tag{3.1}$$

where $w = C(\theta - \theta_o)$ is the stored thermal energy, $\theta_o = 0°$ K is a reference temperature, $\boldsymbol{q} = -\mathbb{K}\nabla\theta$ is the conductive heat flux, and $\mathcal{S}^{vol} \leq 0$ is the power extraction per unit volume.

3.1. Point-source solutions

The system under consideration is shown in Fig. 3. The complete time-varying solution is obtained by superposing a series of instantaneous spatially varying heat-extraction loads and adding/integrating the solutions to represent a continuous source. The thermal field solution to a point-source can be found in the classical works of Jaeger [9] and Carslaw and Jaeger [10]. Complete, rigorous, derivations and reviews can be found in Kennedy
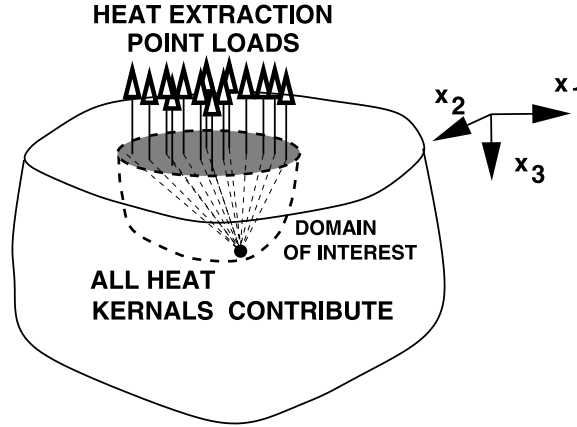
Fig. 3. Construction of a heat extraction pattern from heat-extraction point-loads (kernels).

[11] and Kachanov et al. [12]. Explicitly, the temperature increase due to a point thermal load on a half-space is[2]

$$\Delta\theta(x_1, x_2, x_3, t) = \frac{S(\boldsymbol{x}^\star)}{4(\rho C)(\pi D)^{3/2}} \int_{\tau=0}^{t} \frac{e^{-\frac{R^2}{4D(t-\tau)}}}{(t-\tau)^{3/2}}\, d\tau, \tag{3.2}$$

where $D = \frac{\mathbb{K}}{(\rho C)}$ is the thermal diffusivity, S is the *heat input rate in Watts* located at $\boldsymbol{x} = \boldsymbol{x}^\star$ and

$$R \overset{\text{def}}{=} \sqrt{(x_1^\star - x_1)^2 + (x_2^\star - x_2)^2 + (x_3^\star - x_3)^2} \tag{3.3}$$

is the distance away from the point of application $\boldsymbol{x}^\star$. The integral equation in Eq. (3.2) is solved by inserting change of variables:

$$\beta = \frac{R}{(4D(t-\tau))^{1/2}} \Rightarrow d\beta = \frac{R}{4(D)^{1/2}(t-\tau)^{3/2}}\, d\tau. \tag{3.4}$$

Thus,

$$\Delta\theta(x_1, x_2, x_3, t) = \frac{S(\boldsymbol{x}^\star)}{\rho C D \pi^{3/2} R} \int_{\beta=\frac{R}{2(Dt)^{1/2}}}^{\infty} e^{-\beta^2}\, d\beta. \tag{3.5}$$

Because the integral can be recast using the complementary error function:

$$erfc(\frac{R}{2(Dt)^{1/2}}) \overset{\text{def}}{=} \frac{2}{\pi^{1/2}} \int_{\beta=\frac{R}{2(Dt)^{1/2}}}^{\infty} e^{-\beta^2}\, d\beta, \tag{3.6}$$

the solution becomes

$$\Delta\theta(x_1, x_2, x_3, t) = \frac{S(\boldsymbol{x}^\star)}{2(\rho C)D\pi R} erfc(\frac{R}{2(Dt)^{1/2}}). \tag{3.7}$$

When $t \to \infty$, $\beta \to 0$, $erfc(0) \to 1$, and the steady-state solution becomes

$$\Delta\theta(x_1, x_2, x_3, t = \infty) = \frac{S(\boldsymbol{x}^\star)}{2(\rho C)D\pi R}. \tag{3.8}$$

3.2. Target spatial pattern construction

The point-load framework allows one to take the incremental point-load pulses and to integrate them to construct spatially-varying patterns. Spatially-varying solutions are created by superposing multiple ($i = 1, 2, \ldots, N$) point

---

[2] The top surface of system is assumed to have a zero thermal flux except at the location of where the point heat sources denoted S (*in Watts*) are applied.

loads (Fig. 3) within a pattern, thus

$$\Delta\theta^{tot-N}(x_1, x_2, x_3, t) = \sum_{i=1}^{N} \Delta\theta_i(x_1, x_2, x_3, t) \tag{3.9}$$

where

$$\Delta\theta_i(x_1, x_2, x_3, t) = \frac{S_i(\boldsymbol{x}_i)}{4(\rho C)(\pi D)^{3/2}} \int_{\tau=0}^{t} \frac{e^{-\frac{R_i^2}{4D(t-\tau)}}}{(t-\tau)^{3/2}} \, d\tau, \tag{3.10}$$

which can be integrated analytically evaluated as a complementary error function (Eq. (3.6)). At a point, the complete thermal field is computed by summing all point load contributions ($i = 1, 2, \ldots, N$, **appropriately translated/shifted according to their surface location**). Algorithmically, the procedure is (Fig. 4):

- **GENERATE THE CONFIGURATION.**
- **STEP 1: DISTRIBUTE N POINT-LOADS.**
- **STEP 2: COMPUTE THERMAL FIELD CONTRIBUTION FROM EACH LOAD, $i = 1, 2, \ldots, N$, ON THE SURFACE:** $\Delta\theta_i(x_1, x_2, x_3, t)$.
- **STEP 3: SUM CONTRIBUTIONS OF EACH LOAD, $i = 1, 2, \ldots, N$:**

$$\theta^{tot-N}(x_1, x_2, x_3, t) = \theta(t = 0) + \Delta\theta^{tot-N}(x_1, x_2, x_3, t) \tag{3.11}$$

- **STEP 4: REPEAT STEPS 2-3 FOR EACH TIME SLICE.**
- **STEP 5: COMPUTE RESPONSE STATISTICS IN TARGET PATTERN ZONE.**

Remark 2. We note that the realistic quasi-stochastic character of power input/extraction is captured in this process through the random placement of the heat extraction kernels

3.3. Heat extraction compensation within a target zone

One key advantage of this method is the easy determination of the spatial distribution of the thermal field in the target pattern and the simple post-processing of the aggregate thermal field for statistical metrics such as the normalized standard deviation in the domain of the pattern (Ω_{tar}):

$$\frac{\|\theta(\boldsymbol{x}) - \langle\theta(\boldsymbol{x})\rangle_{\Omega_{tar}}\|_{\Omega_{tar}}}{\langle\theta(\boldsymbol{x})\rangle_{\Omega_{tar}}}, \tag{3.12}$$

where

- $\|\theta(\boldsymbol{x}) - \langle\theta(\boldsymbol{x})\rangle_{\Omega_{tar}}\| \stackrel{\text{def}}{=} \left(\frac{1}{V_{tar}} \int_{\Omega_{tar}} (\theta(\boldsymbol{x}) - \langle\theta(\boldsymbol{x})\rangle_{\Omega_{tar}})^2 \, d\boldsymbol{x}\right)^{\frac{1}{2}}$ is the standard deviation and
- $\langle\theta(\boldsymbol{x})\rangle_{\Omega_{tar}} = \frac{1}{V_{tar}} \int_{\Omega_{tar}} \theta(\boldsymbol{x}) \, d\boldsymbol{x}$ is the volume average,

both of which will be useful later in the work. From Eq. (3.2), repeated again below

$$\Delta\theta(x_1, x_2, x_3, t) = \frac{S(\boldsymbol{x}^\star)}{4(\rho C)(\pi D)^{3/2}} \int_{\tau=0}^{t} \frac{e^{-\frac{R^2}{4D(t-\tau)}}}{(t-\tau)^{3/2}} \, d\tau, \tag{3.13}$$

one observes that the temperature rise will be highest in the interior of the targeted pattern, which then decreases exponentially towards the targeted pattern's edges. This motivates the pursuit of "heat-extraction-shaping" technologies that attempt to modulate the intensity of the extraction within the targeted pattern in an effort to induce a more uniform temperature field, whereby:
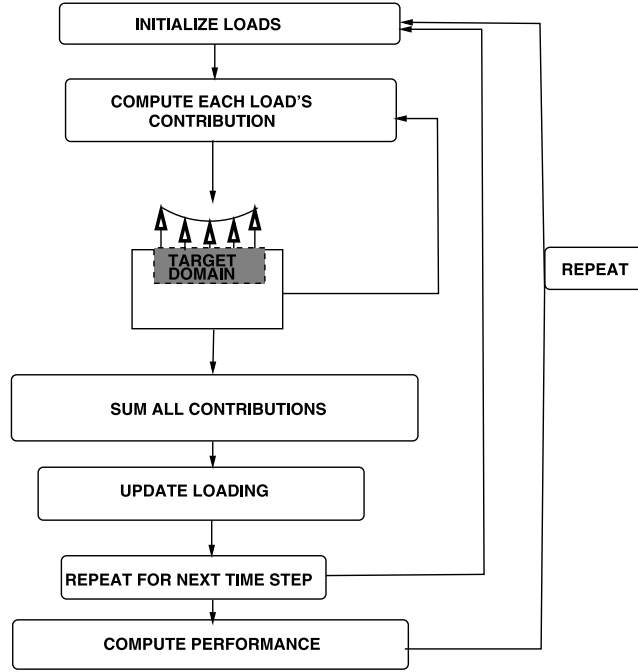
Fig. 4. Algorithm for heat-extraction kernel summation within a target pattern.

- **Temporal Parameters:** As shown in Fig. 1, heat extraction intensity will be varied temporally, by breaking the total simulation time into time-slabs, each with a different intensity level.
- **Spatial Parameters:** As shown in Fig. 1, spatial profiles will be varied by breaking the domain into spatial slabs, which vary radially, each with a different intensity level.
- The total number of Machine-Learning parameters will be a combination of the temporal and spatial parameters.
- The total wattage applied ($S^{tot}(\boldsymbol{x})$) to the targeted pattern is

$$Total\ Wattage = S^{tot} = \int_{\Omega^{tot}} \hat{S}(\boldsymbol{x})\,d\Omega. \tag{3.14}$$

4. Numerical examples

As an example, consider a probe characterized by the equation

$$\left(\frac{\|x_1 - x_{1o}\|}{R_1}\right)^{p_1} + \left(\frac{\|x_2 - x_{2o}\|}{R_2}\right)^{p_2} = 1, \tag{4.1}$$

where, for a circular probe,

- Center: $(x_{1o}, x_{2o}) = (0, 0)$,
- Shape exponents: $p_1 = p_2 = 2$,
- Radii: $R_1 = 0.01 = R_2 = 0.01$ m.

In this example,

- **Temporal Parameters**: $N_t = 10$ (ten stages of time evolution)
- **Spatial Parameters**: $N_s = 21$ (ten rings of radial variation with ten level of growth beyond a baseline value (one parameter))
- **Number of Heat Extraction Kernels**: $N_k = 10^4$

- **Total Wattage Extracted**: $S^{tot} = 5$ Watts
- **Wattage per kernel**, $\hat{S}_i$: Computed by determining (a) an unscaled initial distribution (relative magnitudes) of all of the kernels ($\hat{S}_i^o$) and then (b) computing a single scaling factor, s, that forces the sum to be equal to the total wattage via:

$$s \sum_{i=1}^{N_k} \hat{S}_i^o = S^{tot} \rightarrow s = \frac{S^{tot}}{\sum_{i=1}^{N_k} \hat{S}_i^o} \rightarrow \hat{S}_i = s\hat{S}_i^o. \tag{4.2}$$

- **Total Simulation Time**: $T = 1$ s

The following material parameter sets were used:

- $\mathbb{K} = 0.2$ W/(m-K).
- $\rho = 1000$ kg/m^3.
- $C = 5$ J/(K-kg).
- $\theta_{tar} = 120$ K.

Fig. 5 illustrates the effects of increased border/boundary heat extraction to compensate for the thermal losses at the boundary in order to process the target zone to below the target temperature and the kill the tissue. On the base of the plot, "red" indicates completely "killed tissue" (reached 120° K or below at least once), "green" indicates a transition and "blue" means undamaged. For the heat extraction arrows, "red" indicated a lower level of heat extraction (power flux), while yellow, green and green indicate increasing levels towards "blue" which is at a maximum extraction level. Additionally, the arrows have been scaled to correlate to the extraction rate of the energy. The starting temperature was set to $\theta(\boldsymbol{x}, t = 0) = 300°$ K and the simulation duration time was 1000 ms. A sampling grid of $60(x_1) \times 60(x_2) \times 5(x_3)$ and 10000 randomly-distributed point-load heat kernels were used to generate results for Fig. 5. The total wattage (4 Watts) was distributed over the 10000 heat kernels. The "mesh" of sampling points was superposed onto the target domain in order to illustrate the density of heat kernels used in Eqs. (3.9)–(3.10). We note that the realistic quasi-stochastic character of a power input/extraction is captured in this process through the random placement of the heat kernels. With the level of sampling points used (10000 points), we found no significant variation in results for increasing the number of heat-kernels. At any instant in time, following the algorithm in Fig. 4, it takes under one second to sum the entire system and to post-process. Accordingly, the code completes an entire time–history (hundreds of time-slices/evaluations) in under a minute on a Macbook Pro, making it ideal for power input/extraction-system design studies. As a practical matter, the use of this method is to test several hundred, if not thousands, processing scenarios as a system parameter optimization tool for example embedded in a Machine Learning Algorithm, which we discuss next.

5. Machine-learning optimization and automatic design

We define a design vector of the form $\boldsymbol{\Lambda}^i \overset{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \ldots, \Lambda_N^i\} = \{\text{temporal variation, radial distribution, etc.}\}$ and cost-error function of the following form

$$\Pi(\boldsymbol{\Lambda}) = \underbrace{w_1 \langle \phi(\boldsymbol{x}) \rangle_{\Omega_{tar}}}_{Term\#1} + \underbrace{w_2 \langle \gamma(\boldsymbol{x}) \rangle_{(\Omega - \Omega_{tar})}}_{Term\#2}$$
$$+ \underbrace{\frac{1}{T} \int_0^T w_3 \frac{\|\theta(\boldsymbol{x}) - \theta_{tar}\|_{\Omega_{tar}}}{\theta_{tar}} \, dt}_{Term\#3} + \underbrace{\frac{1}{T} \int_0^T w_4 \frac{\|\theta(\boldsymbol{x}) - \langle \theta(\boldsymbol{x}) \rangle_{\Omega_{tar}}\|_{\Omega_{tar}}}{\langle \theta(\boldsymbol{x}) \rangle_{\Omega_{tar}}} \, dt}_{Term\#4}, \tag{5.1}$$

where

- For and $\boldsymbol{x} \in \Omega_{tar}$, $\phi(\boldsymbol{x}) = 0$ if $\theta(\boldsymbol{x}) \leq \theta_{tar}$ at least once during the simulation and $\phi(\boldsymbol{x}) = 1$ otherwise (an "inside" indicator function) and
- For and $\boldsymbol{x} \in (\Omega - \Omega_{tar})$, $\gamma(\boldsymbol{x}) = 1$ if $\theta(\boldsymbol{x}) \leq \theta_{tar}$ at least once during the simulation and $\gamma(\boldsymbol{x}) = 0$ otherwise (an "outside" indicator function).

The terms in Eq. (5.1) have the following meanings:
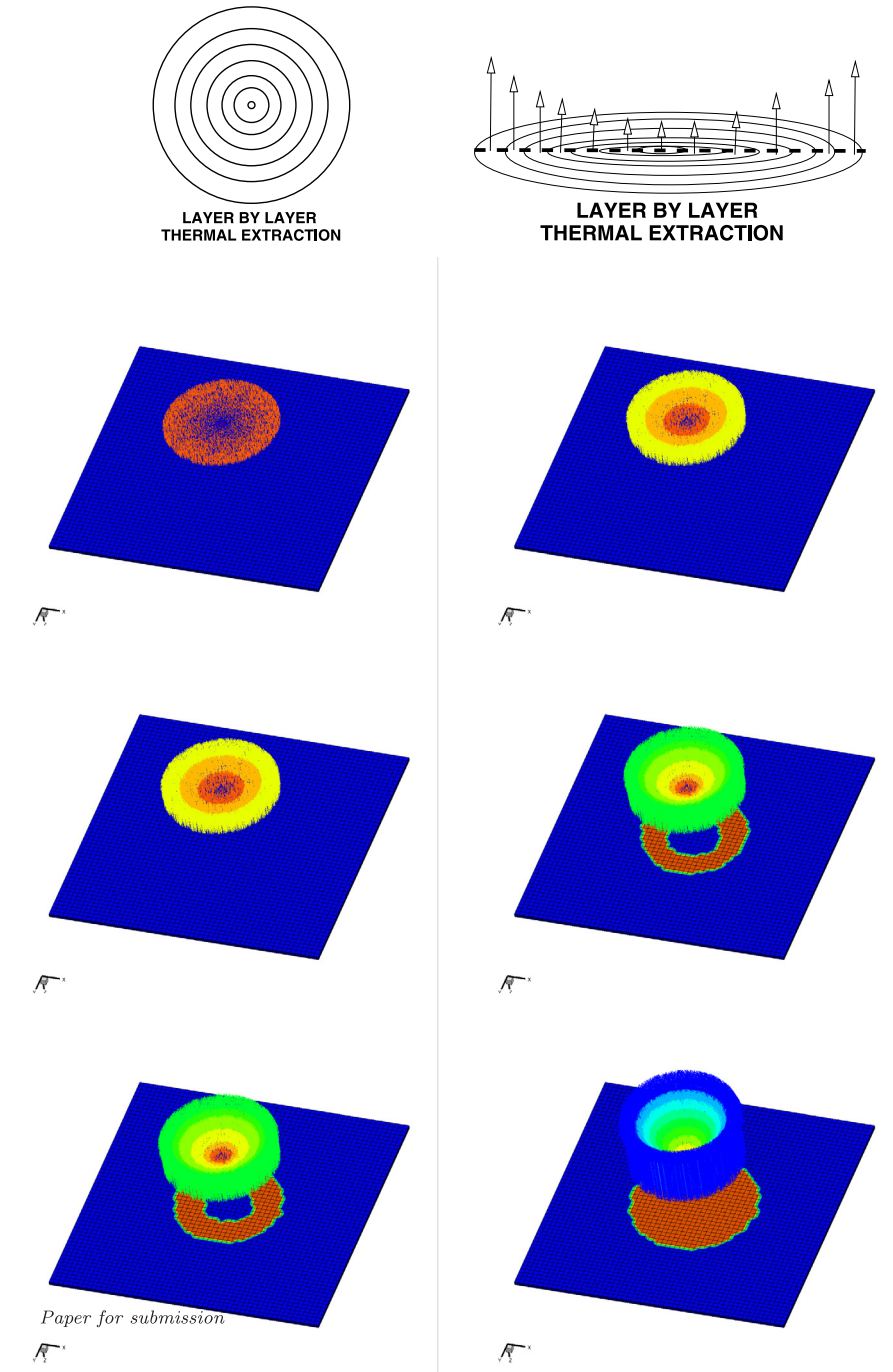
T.I. Zohdi and M. Zohdi-Mofid

Fig. 5. Increased border/boundary heat extraction is used compensate for the thermal losses at the boundary in order to process the target zone to below the target temperature and the kill the tissue. On the base of the plot, "red" indicates completely "killed tissue" (reached 120° K or below at least once), "green" indicates a transition and "blue" means undamaged/unprocessed. For the heat extraction arrows, "red" indicated a lower level of heat extraction (power flux), while yellow, green and green indicate increasing levels towards "blue" which is at a maximum extraction level. Additionally, the arrows have been scaled to correlate to the extraction rate of the energy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
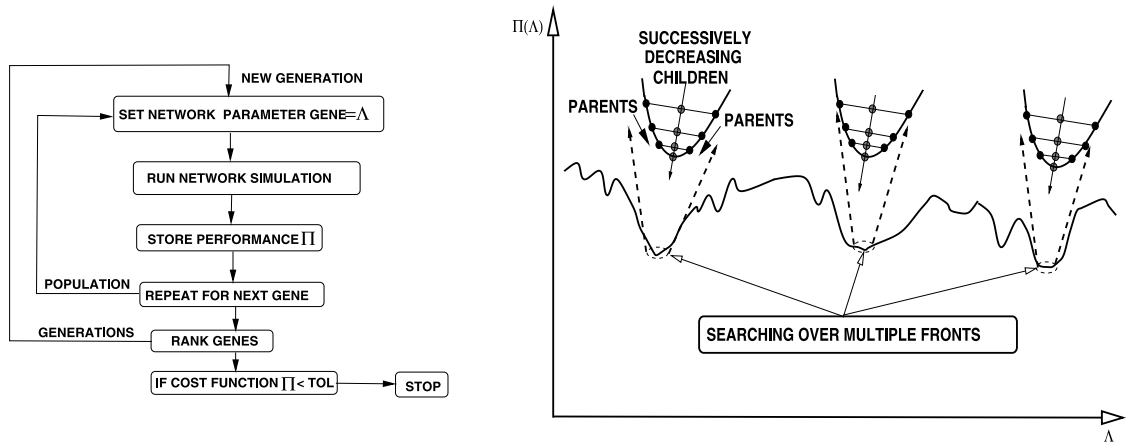
Fig. 6. The basic action of a machine-learning/genetic algorithm (Zohdi [13–18]) inducing multidirectional search.

- Term #1 represents the fraction of material inside the target zone that has *not* reached critical freezing at least once in the simulation time.
- Term #2 represents the fraction of material outside the target zone that has reached critical freezing at least once in the simulation time.
- Term #3 represents the normalized standard deviation of the temperature, relative to the desired target temperature in the target zone.
- Term #4 represents the normalized standard deviation of the temperature, relative to the average temperature in the target zone.

The design vector $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\}$ represents the spatial and temporal variations. The rapid rate at which these simulations can be completed allows the exploration of inverse problems seeking to determine what parameter combinations can deliver a desired result (Fig. 6). In order to cast the objective mathematically, we set the problem up as a machine-learning algorithm (MLA), specifically a genetic algorithm (GA) variant, which is well-suited for nonconvex optimization. Following Zohdi [13–18], we formulate the objective as a cost function minimization problem that seeks system parameters that match a desired response, in this case a minimum of $\Pi(\Lambda_1, \ldots \Lambda_N)$. We systematically minimize Eq. (5.1), $min_\Lambda \Pi$, by varying the design parameters: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\}$. The system parameter search is conducted within the constrained ranges of $\Lambda_1^{(-)} \leq \Lambda_1 \leq \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \leq \Lambda_2 \leq \Lambda_2^{(+)}$, $\Lambda_3^{(-)} \leq \Lambda_3 \leq \Lambda_3^{(+)}$, etc. These upper and lower limits are dictated by what is physically feasible.

5.1. Machine-learning algorithm (MLA)

Cost functions such as Π are nonconvex in design parameter space and often nonsmooth. Their minimization is usually difficult with direct application of gradient-based methods. This motivates nonderivative search methods, for example those found in machine-learning algorithms (MLAs). One of the most basic subsets of MLAs are so-called Genetic Algorithms (GAs). For a review of GAs, see the pioneering work of John Holland ([19,20]), as well as Goldberg [21], Davis [22], Onwubiko [23] and Goldberg and Deb [24]. A description of the algorithm will be described next, following Zohdi [13–18].

5.2. Algorithmic structure

The MLA/GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems and, broadly speaking, involves the following essential concepts (Fig. 6):

1. **POPULATION GENERATION:** Generate a parameter population of genetic strings: $\boldsymbol{\Lambda}^i$
2. **PERFORMANCE EVALUATION:** Compute performance of each genetic string: $\Pi(\boldsymbol{\Lambda}^i)$
3. **RANK STRINGS:** Rank them $\boldsymbol{\Lambda}^i$, $i = 1, ..., S$ from best to worst
4. **MATING PROCESS:** Mate pairs/produce offspring
5. **GENE ELIMINATION:** Eliminate poorly performing genetic strings
6. **POPULATION REGENERATION:** Repeat process with updated gene pool and new *random* genetic strings
7. **SOLUTION POST-PROCESSING:** Employ gradient-based methods afterwards in local "valleys"-*if smooth enough*

5.3. Specifics

Following Zohdi [13]–[18]. the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i$, $(i = 1, 2, 3, ..., S)$:

$$\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \Lambda_1^i \\ \Lambda_2^i \\ \Lambda_3^i \\ ... \\ \Lambda_N^i \end{array} \right\} \tag{5.2}$$

- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$, (i=1, ..., S)
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i$, (i=1, ..., S) from best to worst
- **STEP 4:** Mate nearest pairs and produce two offspring, (i=1, ..., S):

$$\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \boldsymbol{\Phi} \circ \boldsymbol{\Lambda}^i + (\mathbf{1} - \boldsymbol{\Phi}) \circ \boldsymbol{\Lambda}^{\mathbf{i+1}} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \phi_1 \Lambda_1^i \\ \phi_2 \Lambda_2^i \\ \phi_3 \Lambda_3^i \\ ... \\ \phi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1 - \phi_1)\Lambda_1^{i+1} \\ (1 - \phi_2)\Lambda_2^{i+1} \\ (1 - \phi_3)\Lambda_3^{i+1} \\ ... \\ (1 - \phi_N)\Lambda_N^{i+1} \end{array} \right\} \tag{5.3}$$

and

$$\boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \boldsymbol{\Psi} \circ \boldsymbol{\Lambda}^i + (\mathbf{1} - \boldsymbol{\Psi}) \circ \boldsymbol{\Lambda}^{\mathbf{i+1}} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \psi_1 \Lambda_1^i \\ \psi_2 \Lambda_2^i \\ \psi_3 \Lambda_3^i \\ ... \\ \psi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1 - \psi_1)\Lambda_1^{i+1} \\ (1 - \psi_2)\Lambda_2^{i+1} \\ (1 - \psi_3)\Lambda_3^{i+1} \\ ... \\ (1 - \psi_N)\Lambda_N^{i+1} \end{array} \right\} \tag{5.4}$$

where for this operation, the ϕ_i and ψ_i are random numbers, such that $0 \le \phi_i \le 1$, $0 \le \psi_i \le 1$, which are different for each component of each genetic string
- **STEP 5:** Eliminate the bottom M strings and keep top K parents and their K offspring (K offspring+K parents+M=S)
- **STEP 6:** Repeat STEPS 1–5 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **REFOCUS OPTION:** One can refocus search around best performing parameter set every few generations, thus concentrating the computation effort around the most promising (optimal) areas of design space. This operation is performed by re-centering the search around the best genetic string and performing the search in a new constrained local interval about that genetic string.
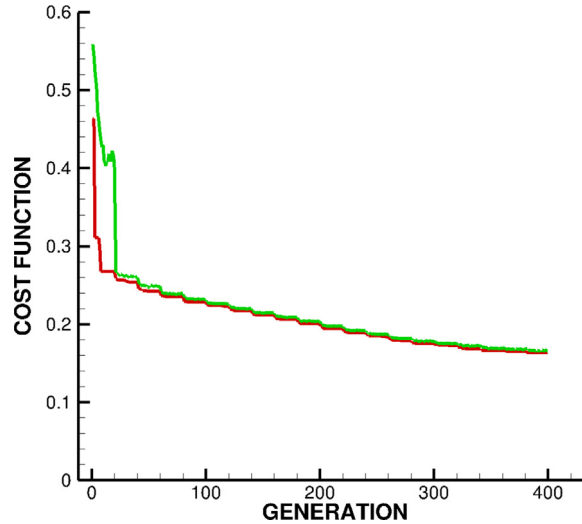
Fig. 7. Fig. 7 illustrates the results for the cost function for the best performing gene (*red*) as a function of successive generations, as well as the average performance cost function of the entire population of genes (*green*). Starting at the top left and moving to the right, we allowed the MLA/GA to readapt every 20 generations. Often, this action is more efficient than allowing the algorithm not to readapt, since it probes around the current optimum for better local alternatives. The total (average population) cost function was initially $\Pi^{total} \approx 0.5588$ and was reduced to $\Pi^{tot} \approx 0.1633$ a reduction of $\frac{0.5588-0.1633}{0.5588} \rightarrow 70.77\%$. The entire 400 generation simulation, with 24 genes per evaluation (9600 total designs) took a few minutes on a laptop, *making it ideal as a design tool*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark 3. If one selects the mating parameters $\phi's$ and $\psi's$ to be greater than one and/or less than zero, one can induce "mutations", i.e. characteristics that neither parent possesses. However, this is somewhat redundant with introduction of new random members of the population in the current algorithm. If one does not retain the parents in the algorithm above, it is possible that inferior performing offspring may replace superior parents. Thus, top parents should be kept for the next generation. This guarantees a monotone reduction in the cost function, unless one implements the refocusing option mentioned previously. Retained parents do not need to be reevaluated, making the algorithm less computationally expensive, since these parameter sets do not have to be reevaluated in the next generation's calculations. In the absence of refocusing, numerous studies of the author (Zohdi [13–18]) have shown that the advantages of parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we observe that this algorithm is easy to parallelize.

Remark 4. After application of such a global search algorithm, one can apply a gradient-based method around the best performing parameter set, 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 they are generally much more efficient for convex optimization of smooth functions. An exhaustive review of these methods can be found in the texts of Luenberger [25] and Gill, Murray and Wright [26]. *However, refocussing usually makes this extra step unnecessary, since the search eventually concentrates the computational effort locally around the best parameter set beforehand.*

5.4. Algorithmic settings

In this example, the following search parameter settings were used:

- Number of design variables: 31.
- Design weights: $w_1 = w_2 = w_3 = 0.25$.
- Population size per generation: 24.
- Number of parents to keep in each generation: 6.
- Number of children created in each generation: 6.
- Number of completely new genes created in each generation: 12.
- Number of generations for re-adaptation around a new search interval: 20.
- Number of generations: 400.

The equation for the heat load is

$$S(r, t) = S_o a(t) b(r) e^{g(r)r}, \tag{5.5}$$

where r is the radial distance from the probe center and

- $S(r, t)$ is the heat extraction load, which varies radially from the center and with time.
- $-20 \leq S_o \leq -10$ is the baseline value of the heat extraction load (Total Wattage).
- $0.1 \leq a(t) \leq 10$ is the amplitude multiplier with time (broken into 10 temporal epochs).
- $0.1 \leq b(r) \leq 10$ is the amplitude multiplier, which varies radially from the center (broken into 10 radial regions and 10 temporal epochs).
- $0.1 \leq g(r) \leq 10$ is the exponential growth, which varies radially from the center (broken into 10 radial regions).

5.5. Results

The computational gene has the following form (31 parameters):

$$\Lambda = \{\Lambda_{1-31}\} = \{S_o, a_1, a_2, \ldots, a_{10}, b_1, b_2, \ldots, b_{10}, g_1, g_2, \ldots, g_{10}\}, \tag{5.6}$$

The best gene was

- $\Lambda_1 = S_o = -1.1171$,
- $\Lambda_{2-11} = \{a_1, a_2, \ldots, a_{10}\} = \{3.847, 4.643, 1.653, 2.901, 1.486, 6.466, 0.353, 3.827, 9.547, 2.946\}$
- $\Lambda_{12-21} = \{b_1, b_2, \ldots, b_{10}\} = \{7.420, 8.894, 2.380, 8.962, 4.869, 2.373, 3.854, 5.406, 9.524, 5.222\}$
- $\Lambda_{22-31} = \{g_1, g_2, \ldots, g_{10}\} = \{3.486, 5.569, 4.343, 3.472, 2.559, 3.027, 4.790, 2.603, 2.582, 2.361\}$

Fig. 7 illustrates the results for the cost function for the best performing gene (*red*) as a function of successive generations, as well as the average performance cost function of the entire population of genes (*green*). Starting at the top left and moving to the right, we allowed the MLA/GA to readapt every 20 generations. Often, this action is more efficient than allowing the algorithm not to readapt, since it probes around the current optimum for better local alternatives. The total (average population) cost function was initially $\Pi^{total} \approx 0.5588$ and was reduced to $\Pi^{tot} \approx 0.1633$ a reduction of $\frac{0.5588 - 0.1633}{0.5588} \rightarrow 70.77\%$. The entire 400 generation simulation, with 24 genes per evaluation (9600 total designs) took a few minutes on a laptop, *making it ideal as a design tool*. We note that, for a given set of parameters, a complete simulation takes a fraction of a second, thus hundreds of thousands of parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA/GA*. The speed at which the overall process can be completed makes it a suitable digital-twin of the system that can run in real-time or faster than the actual physical system, making it suitable as either a design tool or an adaptive controller.

Remark 5. Although not considered in the present work, the role of uncertainty quantification, given the possible imprecision in experimental determination of tissue parameters, is a viable extension to pursue. Some previous uncertainty quantification results on the closely related reverse problem of laser heat input can be found in Zohdi and Castrillon [27].

6. Summary and model extensions

In summary, akin to a carpenter's hammer, the use of cryotherapy is considered one of the workhorses of dermatologists. Used to treat actinic keratoses, which are precancerous lesions that can evolve into skin cancer, it is also used to treat a variety of other skin lesions such as viral warts and molloscum, skin tags and seborrheic keratoses, as well as some superficial skin cancers. Cryotherapy is also used more invasively to treat other cancers such as cervical, prostate and liver cancer. The focus of this work was to develop a rapid computational tool, a so-called digital-twin (a digital replica) for the performance of a cryogenic process, which can be used to pre-plan and optimize surgical procedures to maximize successful outcomes. Specifically, we modeled the optimal cryoprobe-induced cooling protocol needed to eliminate cells/tissue in specific regions, without damaging nearby tissue. The modeling approach was to develop mathematical surface point-source heat extraction kernels and then to create optimal surface patterns that the cryoprobe should induce by arranging the point-sources accordingly. Radial and temporal control of the heat extraction was modeled. The entire subdermal cryogenic thermal field is then constructed by superposing the solutions, enabling precise cryogenic treatment. Finally, a Machine Learning Algorithm (MLA) is then applied to optimize the set of parameters to deliver a precise response. The approach exploited precomputed solutions for point-source heat-extraction kernels and then created patterns by arranging the point-sources in the proper configuration. The entire thermal field was then constructed by superposing the solutions. A series of three-dimensional problems were solved to illustrate the approach, which is geared to rapidly solving the response to complex patterns. The advantage of such an approach is that process designers can efficiently explore system parameter scenarios, varying quantities such as heat extraction spatial distribution and temporal duration, delaying more computationally-intensive "brute-force" coupled thermo-mechanical PDE discretization, based on Finite Element or Finite Difference methods to the end of the design process. We remark that at the end of a system design phase, in order to obtain precise values of the thermally-induced stress fields, phase-transformations and even cryoablation, etc., one will have to solve a computationally-intensive coupled thermo-mechanical initial boundary value problem with phase transformations (see Davis [28]), consisting of solving a balance of linear momentum governed by

$$\nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \rho \frac{d^2 \boldsymbol{u}}{dt^2}, \tag{6.1}$$

and the interconversions of various forms of energy (mechanical, thermal, etc.) in a system are governed by the first law of thermodynamics,

$$\rho \dot{w} - \boldsymbol{\sigma} : \nabla_x \dot{\boldsymbol{u}} + \nabla_x \cdot \boldsymbol{q} - \rho z = 0. \tag{6.2}$$

Such direct numerical multiphysics simulations, with detailed microstructural morphologies, have been constructed in Zohdi [29–32], based on Finite Element and Finite Difference Time Domain (FDTD) Methods, using immersed microstructural boundary approaches, coupled with staggering schemes. Additionally, we remark that there are other Machine Learning type paradigms that complement genetic-based approaches, such as Artificial Neural Networks (ANN). ANN have received huge attention in the scientific community over the last decade and are based on layered input–output type frameworks that are essentially adaptive nonlinear regressions of the form $\mathcal{O} = \mathcal{B}(\boldsymbol{I}, \boldsymbol{w})$, where $\mathcal{O}$ is a desired output and $\mathcal{B}$ is the ANN comprised of (1) **synapses**, which multiply inputs (I_i, $i = 1, 2, \ldots, M$) by weights (w_i, $i = 1, 2, \ldots, N$) that represent the input relevance to the desired output, (2) **neurons**, which aggregate outputs from all incoming synapses and apply activation functions to process the data and (3) **training**, which calibrates the weights to match a desired overall output. For more details, see Zohdi [33]. The blending of these computational paradigms to produce viable digital-twins for precise cryodermatological surgical procedures is under investigation by the authors.

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.

Data availability

No data was used for the research described in the article.

Acknowledgments

References

[1] K.T. Clebak, M. Mendez-Miller, J. Croad, Cutaneous cryosurgery for common skin conditions, Am. Fam. Phys. 101 (7) (2020) 399–406, Journal.

[2] W.C. Cranwell, R. Sinclair, Optimising cryosurgery technique, Aust. Fam. Phys. 46 (5) (2017) 270–274, Journal.

[3] J. Prohaska, A.H. Jan, Cryotherapy. StatPearls. Treasure Island (FL), 2022, Resource On DermNet.

[4] K.D. Timmerhaus, R. Reed, Cryogenic Engineering: Fifty Years of Progress, Springer Science+Business Media LLC, 2007.

[5] J.M. Nash, Vortex expansion devices for high temperature cryogenics, in: Proc. of the 26th Intersociety Energy Conversion Engineering Conference, vol. 4, 1991, pp. 521–525.

[6] R. Barron, Cyrogenic Systems, McGraw-Hill Book Company.

[7] Carroll Gantz, Refrigeration: A History, McFarland & Company, Inc., Jefferson, North Carolina, ISBN: 978-0-7864-7687-9, 2015, p. 227.

[8] Pfizer-BioNTech COVID-19 Vaccine Vaccination Storage & Dry Ice Safety Handling, Pfizer-BioNTech, Archived from the original on 24 January 2021. Retrieved 17 December2020.

[9] J.C. Jaeger, Moving sources of heat and the temperature of sliding contacts, Proc. R. Soc. NSW 76 (1942) 203–224.

[10] H.S. Carslaw, J.C. Jaeger, Conduction of Heat in Solids, second ed., Clarendon Press, Oxford, 1959.

[11] F.E. Kennedy, Frictional Heating and contact temperatures; in The Modern Tribology Handbook, CRC Press, 2001.

[12] M. Kachanov, B. Shafiro, I. Tsukrov, Handbook of Elasticity Solutions, Kluwer, New York, 2003.

[13] T.I. Zohdi, Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning, Comput. Methods Appl. Mech. Engrg. 331 (2018) 343–362, 1 April 2018.

[14] T.I. Zohdi, A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments, Comput. Methods Appl. Mech. Eng. 363 (2020) 112907.

[15] T.I. Zohdi, A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety, Comput. Methods Appl. Mech. Engrg. 373 (2021) 113446, 1 January 2021.

[16] T.I. Zohdi, A digital-twin and machine-learning framework for ventilation system optimization for capturing infectious disease respiratory emissions, Arch. Comput. Methods Eng. (2021) http://dx.doi.org/10.1007/s11831-021-09609-3.

[17] T.I. Zohdi, A digital-twin and machine-learning framework for the design of multiobjective agrophotovoltaic solar farms, Comput. Mech. (2021) http://dx.doi.org/10.1007/s00466-021-02035-z.

[18] T.I. Zohdi, A digital-twin and machine-learning framework for precise heat and energy management of data-centers, Comput. Mech. (2022) http://dx.doi.org/10.1007/s00466-022-02152-3.

[19] J.H. Holland, Adaptation in Natural & Artificial Systems, Ann Arbor, Mich. University of Michigan Press, 1975.

[20] J.H. Holland, J.H. Miller, Artificial adaptive agents in economic theory (PDF), Amer. Econ. Rev. 81 (2) (1991) 365–371, Archived from the original (PDF) on October 27, 2005.

[21] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.

[22] S.H. Davis, Theory of Solidification, Cambridge University Press, 2001.

[23] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.

[24] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 121–124.

[25] D. Luenberger, Introduction to Linear & Nonlinear Programming, Addison-Wesley, Menlo Park, 1974.

[26] P. Gill, W. Murray, M. Wright, Practical Optimization, Academic Press, 1995.

[27] T.I. Zohdi, N. Castrillon, Variability of targeted material thermal responses to laser-induced heating in additive manufacturing, J. Manuf. Sci. Eng., ASME (2019) http://dx.doi.org/10.1115/1.4043981.

[28] L. Davis, Handbook of Genetic Algorithms, Thompson Computer Press, 1991.

[29] 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.

[30] T.I. Zohdi, Modeling and simulation of a class of coupled thermo-chemo-mechanical processes in multiphase solids, Comput. Methods Appl. Mech. Eng. 193/6-8 (2004) 679–699.

[31] 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.

[32] T.I. Zohdi, P. Wriggers, (Book, 2005, 2008) Introduction to Computational Micromechanics. Second Reprinting, Springer-Verlag.

[33] Zohdi,T. I. (2022) 193, T.I. Zohdi, A note on rapid genetic calibration of artificial neural networks, Comput. Mech. (2022) http://dx.doi.org/10.1007/s00466-022-02216-4.