Computational Mechanics
https://doi.org/10.1007/s00466-022-02152-3

ORIGINAL PAPER

A digital-twin and machine-learning framework for precise heat and energy management of data-centers

T. I. Zohdi[1]

Received: 20 November 2021 / Accepted: 29 January 2022

Abstract
The massive growth in data-centers has led to increased interest and regulations for management of waste heat and its utilization. This work seeks to develop a combined Digital-Twin and Machine-Learning framework to optimize such systems by controlling both the ventilation and the cooling of the bases of data units/processors in the system. This framework ascertains optimal cooling strategies to deliver a target temperature in the system using a minimum amount of energy. A model problem is constructed for a data-center, where the design variables are the flow rates and air-cooling at multiple ventilation ports and ground-level conduction-based base-cooling of processors. A thermo-fluid model, based on the Navier–Stokes equations and the first law of thermodynamics, for the data-center is constructed and a rapid, stencil-based, iterative solution method is developed. This is then combined with a genomic-based machine-learning algorithm to develop a digital-twin (digital-replica) 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. Numerical examples are provided to illustrate the framework.

Keywords Data-centers · Heat management · Digital-twins · Machine-learning

1 Introduction

1.1 Motivation

Massive increases in internet users worldwide has led to significant demand for 'data-center' services, and subsequent energy use. Here we define 'data-centers' as spaces within building dedicated to housing computer systems comprised of data handling units, telecommunications, high-performance computing devices and associated equipment. Between 2010 and 2018, the global quantity of data traversing the internet increased more than ten-fold, while global data-center storage capacity increased by a factor of 25 in parallel (Masanet et al. [1]). At the largest industrial-scale, the energy usage of such systems is huge, requiring large-scale cooling and air conditioning. Such systems started in the 1940s with the advent of the first computers and have grown with the rise of industrial-scale computation, military installations, research labs, banks, etc. The heat produced by such systems is immense, thus warranting sophisticated cooling systems. While the analysis of the energy trends are hotly debated, one point of agreement is that the volume of data-centers is consistently increasing, year by year. The reader is referred to [2–75] for a wide swath of the literature on this topic. All data points to trends that costs of such systems is huge and growing. The basic trends on energy consumption by data-centers can be found in the extensive report of Shehabi et al. [76]. Therein, the authors have made accurate estimates of data-center energy consumption from 2000 to 2016, relying on previous studies, historical data and forecasted consumption. That report states that in 2014, data-centers in the U.S. consumed an estimated 70 billion kWh, representing about 1.8% of total U.S. electricity consumption. Their analysis also indicates data-center electricity consumption increased by about 4% from 2010 to 2014, a large shift from the 24% increase estimated from 2005 to 2010 and the nearly 90% increase estimated from 2000 to 2005. The trends of approximately 1% increase each year have been consistent over the last decade. In 2017, US based data-centers alone used up more than 90 billion kilowatt-hours of electricity and consumed around 205 terawatt-hours (TWh) in 2018, or one percent of global electricity use

✉ T. I. Zohdi
zohdi@me.berkeley.edu

[1] Department of Mechanical Engineering, 6195 Etcheverry Hall, University of California, Berkeley, CA 94720-1740, USA

Published online: 10 March 2022

(Masanet et al. [1]), and continues to grow, even through the era of pandemic.

1.2 Restrictions

Recent governmental restrictions on energy waste for such installations has led to interest in developing systems that cool data-centers efficiently. A key aspect is the modularity of such systems and the ability to flexibly and quickly reconfigure data-centers. Accordingly, cooling systems need to be optimized to meet requirements specified by data-center managers. This is especially important since oftentimes older buildings or structures are retrofitted to become data-centers. Generally, there are two main types of cooling: (1) air conditioning and ambient air flow and ventilation and (2) fluid-jacket contact cooling surrounding the units, typically at the base. Although advancements in low heat wattage data storage and low voltage cable routing have reduced heating and methods of separating hot and cold airstreams, for example hot-cold aisle containment and in-row cooling units, adaptive cooling is a necessity. We remark that cold aisle containment is achieved by opening the back of equipment racks and enclosing the fronts of the servers with doors and covers. There are a variety of methods in this field, and as these systems become more complex, one must rely on adaptive methods. The total draw of power can range from a few kilowatts for a small set of units to megawatts for a large-scale operation. For very energy intensive data-centers, electricity can account for over 10% of the cost of ownership. There are a variety of metrics used, such as the power usage effectiveness ratio (PUE), which is the ratio of the total power entering the data-center (IT+Overhead) divided by the power used by IT equipment. It is an indicator of the overhead power, such as cooling, lighting, etc. Typical data-centers have a PUE of approximately 2, while the state of the art is $PUE = 1.2$. In 2014, the California Code of Regulations mandated energy efficiency regulations, in particular on airflow. In 2015, the United States enacted the Energy Efficiency Improvement Act, which requires efficient operation of federal facilities, including data-centers. Worldwide, in particular throughout the EU, there have been a series of similar legislation. However, even if one puts legislation aside, the sheer cost of running a data-center approaches the construction costs. *It has now become critical to develop simulation based models to guide operations.*

1.3 Objectives

The present work seeks to develop a combined Digital-Twin and Machine-Learning framework to optimize such systems by controlling both the ventilation and the cooling of the bases of data units/processors in the system. This framework ascertains optimal cooling strategies to deliver a target temperature in the system using a minimum amount of energy. A model problem (Fig. 1) is constructed for a data-center, where the design variables are the flow rates and air-cooling at multiple ventilation ports and ground-level conduction-based base-cooling of processors. A thermo-fluid model, based on the Navier-Stokes Equations and the first law of thermodynamics, for the data-center is constructed and a rapid, stencil-based, iterative solution method is developed. This is then combined with a genomic-based machine-learning algorithm to develop a digital-twin (digital-replica) 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. Numerical examples are provided to illustrate the framework.

2 Governing equations

We start from first principles, proceeding by developing a coupled thermo-fluid model for the air surrounding the processors and the heat-generated by their operation.

2.1 Fluid flow model

For a hydrostatic fluid the stress can be written as

$$\boldsymbol{\sigma} = -P_o \mathbf{1}, \tag{2.1}$$

where $P_o = \frac{tr\boldsymbol{\sigma}}{3}$ is the hydrostatic pressure. In other words, there are no shear stresses in a fluid at rest. In the dynamic case, the pressure, denoted the "thermodynamic pressure", is related to the temperature and the fluid density by an equation of state

$$\mathcal{Z}(P, \rho, \theta) = 0. \tag{2.2}$$

For a fluid in motion

$$\boldsymbol{\sigma} = -P\mathbf{1} + \boldsymbol{\tau}^{vs} \tag{2.3}$$

where $\boldsymbol{\tau}^{vs}$ is a so-called viscous stress tensor, needed in a balance of linear momentum:[1]

$$\nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \rho \frac{d\boldsymbol{v}}{dt}, \tag{2.4}$$

where $\boldsymbol{v}$ is the fluid velocity at point $\boldsymbol{x}$ and $\boldsymbol{f}$ are the body forces. Thus, for a compressible fluid in motion:

$$\frac{tr\boldsymbol{\sigma}}{3} = -P + \frac{tr\boldsymbol{\tau}^{vs}}{3}. \tag{2.5}$$

[1] An inviscid or "perfect" fluid is one where $\boldsymbol{\tau}^{vs}$ is taken to be zero, even when motion is present.

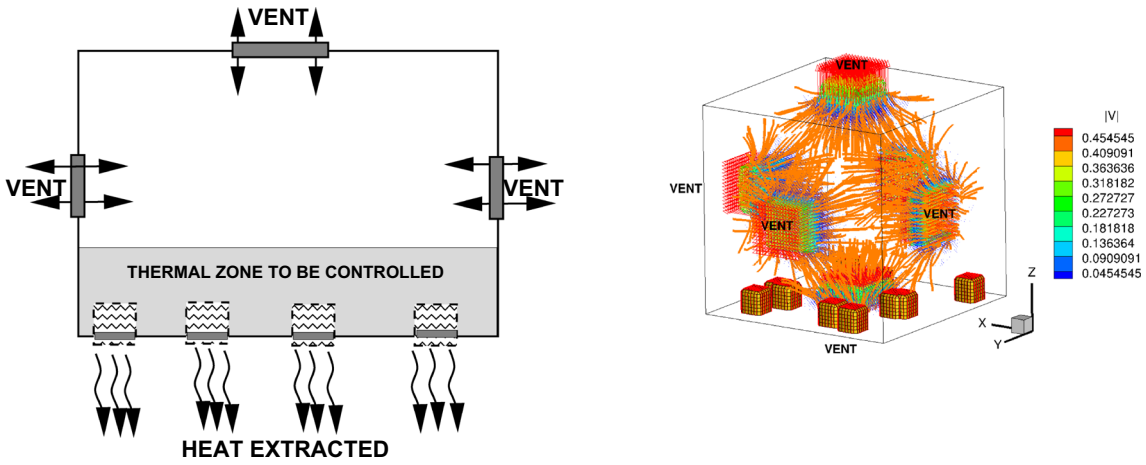

Fig. 1 LEFT: A cross-section of the schematic for an energy management system comprised of air-vents and base cooling units. RIGHT: Using the Navier–Stokes equations (streamlines shown) with 4 side vents, a bottom vent and a top vent for a 'pod' of 10 processors, also with base-cooling-flow streamlines shown

In general, for a fluid we have

$$\boldsymbol{\tau}^{vs} = \mathcal{G}(\boldsymbol{D}) \qquad \text{and} \qquad \boldsymbol{D} \stackrel{\text{def}}{=} \frac{1}{2}(\nabla_x \boldsymbol{v} + (\nabla_x \boldsymbol{v})^T), \tag{2.6}$$

where $\boldsymbol{v}$ is the velocity and $\boldsymbol{D}$ is the symmetric part of the velocity gradient. For a Newtonian fluid, where a linear relation exists between the viscous stresses $\boldsymbol{\tau}^{vs}$ and $\boldsymbol{D}$

$$\boldsymbol{\tau}^{vs} = \mathcal{G}(\boldsymbol{D}) = \boldsymbol{C} : \boldsymbol{D} \tag{2.7}$$

where $\boldsymbol{C}$ is a symmetric positive definite (fourth-order) viscosity tensor. For an isotropic (standard) Newtonian fluid we have

$$\begin{aligned}\boldsymbol{\sigma} &= -P\mathbf{1} + \lambda tr\boldsymbol{D}\mathbf{1} + 2\mu\boldsymbol{D} \\ &= -P\mathbf{1} + 3\kappa\frac{tr\boldsymbol{D}}{3}\mathbf{1} + 2\mu\boldsymbol{D}',\end{aligned} \tag{2.8}$$

where κ is called the bulk viscosity, λ is a viscosity constant, μ the shear viscosity and $\boldsymbol{D}' = \boldsymbol{D} - \frac{tr\boldsymbol{D}}{3}\mathbf{1}$. Explicitly, with an (x_1, x_2, x_3) Cartesian triad

$$\underbrace{\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{Bmatrix}}_{\stackrel{\text{def}}{=}\{\boldsymbol{\sigma}\}} = \underbrace{\begin{Bmatrix}-P\\ -P\\ -P\\ 0\\ 0\\ 0\end{Bmatrix}}_{\stackrel{\text{def}}{=}\{-\boldsymbol{P}\}} + \underbrace{\begin{bmatrix}c_1 & c_2 & c_2 & 0 & 0 & 0\\ c_2 & c_1 & c_2 & 0 & 0 & 0\\ c_2 & c_2 & c_1 & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu\end{bmatrix}}_{\stackrel{\text{def}}{=}[\boldsymbol{C}]} \underbrace{\begin{Bmatrix}D_{11}\\ D_{22}\\ D_{33}\\ 2D_{12}\\ 2D_{23}\\ 2D_{31}\end{Bmatrix}}_{\stackrel{\text{def}}{=}\{\boldsymbol{D}\}}, \tag{2.9}$$

where $c_1 = \kappa + \frac{4}{3}\mu$ and $c_2 = \kappa - \frac{2}{3}\mu$, where $D_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$. The so-called "Stokes' condition" attempts to force the thermodynamic pressure to collapse to the classical definition of mechanical pressure, i.e.

$$\frac{tr\boldsymbol{\sigma}}{3} = -P + 3\kappa\frac{tr\boldsymbol{D}}{3} = -P, \tag{2.10}$$

leading to the conclusion that $\kappa = 0$ or $\lambda = -\frac{2}{3}\mu$. Thus, a Newtonian fluid obeying the Stokes' condition has the following constitutive law:

$$\boldsymbol{\sigma} = -P\mathbf{1} - \frac{2}{3}\mu tr\boldsymbol{D}\mathbf{1} + 2\mu\boldsymbol{D} = -P\mathbf{1} + 2\mu\boldsymbol{D}'. \tag{2.11}$$

Note that

$$\begin{aligned}\dot{J} &= \frac{d}{dt}det\boldsymbol{F} = (det\boldsymbol{F})tr(\dot{\boldsymbol{F}}\cdot\boldsymbol{F}^{-1}) \\ &= Jtr\boldsymbol{L} = J\nabla_x\cdot\boldsymbol{v},\end{aligned} \tag{2.12}$$

where $\boldsymbol{L} = \nabla_x\boldsymbol{v}$ is the velocity gradient. Note that $\nabla_x\cdot\boldsymbol{v} = tr\boldsymbol{L} = tr\boldsymbol{D}$. Therefore, if the fluid is incompressible, $\dot{J} = 0$, then $\nabla_x\cdot\boldsymbol{v} = tr\boldsymbol{L} = tr\boldsymbol{D} = 0$. Therefore,

$$\boldsymbol{\sigma} = -P\mathbf{1} + 2\mu\boldsymbol{D}. \tag{2.13}$$

A conservation of mass dictates

$$\frac{d}{dt}(\rho_o) = \frac{d}{dt}(\rho J) = J\frac{d\rho}{dt} + \rho\frac{dJ}{dt} = 0, \tag{2.14}$$

which leads to

$$\frac{d\rho}{dt} + \frac{\rho}{J}\frac{dJ}{dt} = 0. \tag{2.15}$$

Using Eq. 2.12, Eq. 2.14 becomes

$$\frac{d\rho}{dt} + \rho \nabla_x \cdot \boldsymbol{v} = 0. \tag{2.16}$$

Now write the total temporal ("material") derivative in convective form:

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + (\nabla_x \rho) \cdot \frac{d\boldsymbol{x}}{dt} = \frac{\partial \rho}{\partial t} + \nabla_x \rho \cdot \boldsymbol{v}. \tag{2.17}$$

Thus, Eq. 2.16 becomes

$$\frac{\partial \rho}{\partial t} + \nabla_x \rho \cdot \boldsymbol{v} + \rho \nabla_x \cdot \boldsymbol{v} = \frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho \boldsymbol{v}) = 0. \tag{2.18}$$

Thus, writing the total time derivatives appearing previously as

$$\frac{d\boldsymbol{v}}{dt} = \frac{\partial \boldsymbol{v}}{\partial t}|_{\boldsymbol{x}} + (\nabla_x \boldsymbol{v})|_t \cdot \frac{d\boldsymbol{x}}{dt}, \tag{2.19}$$

the coupled governing equations are (momentarily ignoring thermal effects)

$$\boxed{\begin{aligned} &\frac{\partial \rho}{\partial t} = -\nabla_x \rho \cdot \boldsymbol{v} - \rho \nabla_x \cdot \boldsymbol{v}, \\ &\rho\left(\frac{\partial \boldsymbol{v}}{\partial t} + (\nabla_x \boldsymbol{v}) \cdot \boldsymbol{v}\right) = \nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f}, \\ &\boldsymbol{\sigma} = -P\mathbf{1} + \lambda tr\boldsymbol{D}\mathbf{1} + 2\mu \boldsymbol{D} = -P\mathbf{1} + 3\kappa \tfrac{tr\boldsymbol{D}}{3}\mathbf{1} + 2\mu \boldsymbol{D}', \end{aligned}} \tag{2.20}$$

where, for example, P is given by an Equation of State. Collectively, we refer to these equations as the 'Navier-Stokes' equations. There are a total of three variables: ρ, $\boldsymbol{v}$, and P. It is customary to specify $\boldsymbol{v}$ and P on the boundary, and to determine ρ on the boundary through the Equation of State.

2.2 Thermophysics model

The interconversions of mechanical, thermal and chemical energy in a system are governed by the first law of thermodynamics. It states that the time rate of change of the total energy, $\mathcal{K} + \mathcal{I}$, is equal to the work rate, $\mathcal{P}$ and the net heat supplied, $\mathcal{H} + \mathcal{Q}$,

$$\frac{d}{dt}(\mathcal{K} + \mathcal{I}) = \mathcal{P} + \mathcal{H} + \mathcal{Q}. \tag{2.21}$$

Here the kinetic energy of a subvolume of material contained in Ω, denoted ω, is $\mathcal{K} \stackrel{\text{def}}{=} \int_\omega \frac{1}{2}\rho \boldsymbol{v} \cdot \boldsymbol{v}\, d\omega$, the rate of work or power of external forces acting on ω is given by $\mathcal{P} \stackrel{\text{def}}{=} \int_\omega \rho \boldsymbol{b} \cdot \boldsymbol{v}\, d\omega + \int_{\partial\omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v}\, da$, $\boldsymbol{b}$ being the body forces, the heat flow into the volume by conduction is $\mathcal{Q} \stackrel{\text{def}}{=} -\int_{\partial\omega} \boldsymbol{q} \cdot \boldsymbol{n}\, da = -\int_\omega \nabla_x \cdot \boldsymbol{q}\, d\omega$, $\boldsymbol{q}$ being the heat flux, the heat generated due to sources, *such as chemical reactions*, is $\mathcal{H} \stackrel{\text{def}}{=} \int_\omega \rho z\, d\omega$, z are sources, and the stored energy is $\mathcal{I} \stackrel{\text{def}}{=} \int_\omega \rho w\, d\omega$, w being the stored energy. If we make the assumption that the mass in the system is constant, one has,

$$\begin{aligned} \text{current mass} &= \int_\omega \rho\, d\omega \\ &= \int_{\omega_0} \rho J\, d\omega_0 \approx \int_{\omega_0} \rho_0\, d\omega_0 = \text{original mass}, \end{aligned} \tag{2.22}$$

which implies $\rho J = \rho_0 \Rightarrow \dot{\rho} J + \rho \dot{J} = 0$. Using this and the energy balance leads to

$$\begin{aligned} \frac{d}{dt}\int_\omega \frac{1}{2}\rho \boldsymbol{v} \cdot \boldsymbol{v}\, d\omega &= \int_{\omega_0} \frac{d}{dt}\frac{1}{2}(\rho J \boldsymbol{v} \cdot \boldsymbol{v})\, d\omega_0 \\ &= \int_{\omega_0} (\frac{d}{dt}\rho_0)\frac{1}{2}\boldsymbol{v} \cdot \boldsymbol{v}\, d\omega_0 \\ &+ \int_\omega \rho \frac{d}{dt}\frac{1}{2}(\boldsymbol{v} \cdot \boldsymbol{v})\, d\omega \\ &= \int_\omega \rho \boldsymbol{v} \cdot \dot{\boldsymbol{v}}\, d\omega. \end{aligned} \tag{2.23}$$

We also have

$$\begin{aligned} \frac{d}{dt}\int_\omega \rho w\, d\omega &= \frac{d}{dt}\int_{\omega_0} \rho J w\, d\omega_0 \\ &= \int_{\omega_0} \frac{d}{dt}(\rho_0) w\, d\omega_0 + \int_\omega \rho \dot{w}\, d\omega. \end{aligned} \tag{2.24}$$

By using the divergence theorem, we obtain

$$\begin{aligned} \int_{\partial\omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v}\, da &= \int_\omega \nabla_x \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{v})\, d\omega \\ &= \int_\omega (\nabla_x \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{v}\, d\omega + \int_\omega \boldsymbol{\sigma} : \nabla_x \boldsymbol{v}\, d\omega. \end{aligned} \tag{2.25}$$

Combining the results, and enforcing balance of momentum, leads to

$$\begin{aligned} &\int_\omega (\rho \dot{w} + \boldsymbol{v} \cdot (\rho \dot{\boldsymbol{v}} - \nabla_x \cdot \boldsymbol{\sigma} - \rho \boldsymbol{b}) - \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} + \nabla_x \cdot \boldsymbol{q} - \rho z)\, d\omega \\ &= \int_\omega (\rho \dot{w} - \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} + \nabla_x \cdot \boldsymbol{q} - \rho z)\, d\omega = 0. \end{aligned} \tag{2.26}$$

Since the volume ω is arbitrary, the integrand must hold locally and we have

$$\rho \dot{w} - \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} + \nabla_x \cdot \boldsymbol{q} - \rho z = 0. \tag{2.27}$$

A typical approximation in fluid mechanics is $w \approx \rho C \theta$, where C is the heat capacity and θ is the temperature in

Kelvin. As in the Navier-Stokes equations, breaking the thermal rate term into a fixed part and a convective part yields

$$\rho \dot{w} = \rho C(\frac{\partial \theta}{\partial t} + \nabla_x \theta \cdot \boldsymbol{v}) = \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} - \nabla_x \cdot \boldsymbol{q} + \rho z. \tag{2.28}$$

Remark 1 For the remainder of the work, we will assume that the fluid is incompressible, homogeneous and that it's properties are thermally-insensitive.

3 Discretization of the fluid

3.1 Temporal discretization

For the fluid, we write

$$\frac{d\boldsymbol{v}}{dt} = \frac{\partial \boldsymbol{v}}{\partial t} + \nabla_x \boldsymbol{v} \cdot \boldsymbol{v} = \frac{1}{\rho}\left(\nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f}\right), \tag{3.1}$$

leading to

$$\frac{\partial \boldsymbol{v}}{\partial t} = \frac{1}{\rho}\left(\nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f}\right) - \nabla_x \boldsymbol{v} \cdot \boldsymbol{v} \stackrel{\text{def}}{=} \boldsymbol{L}. \tag{3.2}$$

We discretize for time=$t + \phi\Delta t$, and using a trapezoidal "$\phi-scheme$" ($0 \le \phi \le 1$)

$$\begin{aligned} \frac{\partial \boldsymbol{v}}{\partial t} &\approx \frac{\boldsymbol{v}(t+\Delta t) - \boldsymbol{v}(t)}{\Delta t} \approx \boldsymbol{L}(t + \phi\Delta t) \\ &\approx \phi \boldsymbol{L}(t+\Delta t) + (1-\phi)\boldsymbol{L}(t). \end{aligned} \tag{3.3}$$

Rearranging yields

$$\boldsymbol{v}(t+\Delta t) \approx \boldsymbol{v}(t) + \Delta t\left(\phi \boldsymbol{L}(t+\Delta t) + (1-\phi)\boldsymbol{L}(t)\right) \tag{3.4}$$

where the previously introduced spatial discretization is applied to the derivative terms (such as $\nabla_x \cdot \boldsymbol{\sigma}$) in $\boldsymbol{L}$. The discretized system is formulated next as an implicit time-stepping scheme within each time step L.

Remark 2 The same process is applied to the thermal field

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{\partial \theta}{\partial t} + \nabla_x \theta \cdot \boldsymbol{v} \\ &= \frac{1}{\rho C}\left(\boldsymbol{\sigma} : \nabla_x \boldsymbol{v} - \nabla_x \cdot \boldsymbol{q} + \rho z\right) \stackrel{\text{def}}{=} Z, \end{aligned} \tag{3.5}$$

yielding

$$\theta(t+\Delta t) \approx \theta(t) + \Delta t\left(\phi Z(t+\Delta t) + (1-\phi)Z(t)\right) \tag{3.6}$$

3.2 Spatial stencil-based discretization

Referring to Fig. 2, the following standard approximations are used:

1. For the first derivative of a primal variable v at (x_1, x_2, x_3):

$$\frac{\partial v}{\partial x_1} \approx \frac{v(x_1 + \Delta x_1, x_2, x_3) - v(x_1 - \Delta x_1, x_2, x_3)}{2\Delta x_1} \tag{3.7}$$

2. For the derivative of a flux at (x_1, x_2, x_3):

$$\begin{aligned} &\frac{\partial}{\partial x_1}\left(A\frac{\partial v}{\partial x_1}\right) \\ &\approx \frac{\left(A\frac{\partial v}{\partial x_1}\right)|_{x_1+\frac{\Delta x_1}{2},x_2,x_3} - \left(A\frac{\partial v}{\partial x_1}\right)|_{x_1-\frac{\Delta x_1}{2},x_2,x_3}}{\Delta x_1} \\ &= \frac{1}{\Delta x_1}\left[A(x_1 + \frac{\Delta x_1}{2}, x_2, x_3)\right. \\ &\times\left.\left(\frac{v(x_1+\Delta x_1, x_2, x_3) - v(x_1, x_2, x_3)}{\Delta x_1}\right)\right] \\ &-\frac{1}{\Delta x_1}\left[A(x_1 - \frac{\Delta x_1}{2}, x_2, x_3)\right. \\ &\times\left.\left(\frac{v(x_1, x_2, x_3) - v(x_1-\Delta x_1, x_2, x_3)}{\Delta x_1}\right)\right], \end{aligned} \tag{3.8}$$

where we have used

$$\begin{aligned} A(x_1 + \frac{\Delta x_1}{2}, x_2, x_3) &\approx \frac{1}{2}\left(A(x_1 + \Delta x_1, x_2, x_3)\right. \\ &\left.+A(x_1, x_2, x_3)\right) \end{aligned} \tag{3.9}$$

and

$$\begin{aligned} A(x_1 - \frac{\Delta x_1}{2}, x_2, x_3) &\approx \frac{1}{2}\left(A(x_1, x_2, x_3)\right. \\ &\left.+A(x_1 - \Delta x_1, x_2, x_3)\right) \end{aligned} \tag{3.10}$$

3. For the cross-derivative of a flux at (x_1, x_2, x_3):

$$\begin{aligned} &\frac{\partial}{\partial x_2}\left(A\frac{\partial v}{\partial x_1}\right) \\ &\approx \frac{\partial}{\partial x_2}\left(A(x_1, x_2, x_3)\right. \\ &\left.\left(\frac{v(x_1+\Delta x_1, x_2, x_3) - v(x_1-\Delta x_1, x_2, x_3)}{2\Delta x_1}\right)\right) \\ &\approx \frac{1}{4\Delta x_1 \Delta x_2}(A(x_1, x_2 + \Delta x_2, x_3) \\ &\quad times\left[v(x_1+\Delta x_1, x_2+\Delta x_2, x_3)\right. \\ &\quad \left.-v(x_1-\Delta x_1, x_2+\Delta x_2, x_3)\right] \\ &\quad -A(x_1, x_2 - \Delta x_2, x_3)\left[v(x_1+\Delta x_1, x_2-\Delta x_2, x_3)\right. \\ &\quad \left.-v(x_1-\Delta x_1, x_2-\Delta x_2, x_3)\right]). \end{aligned} \tag{3.11}$$

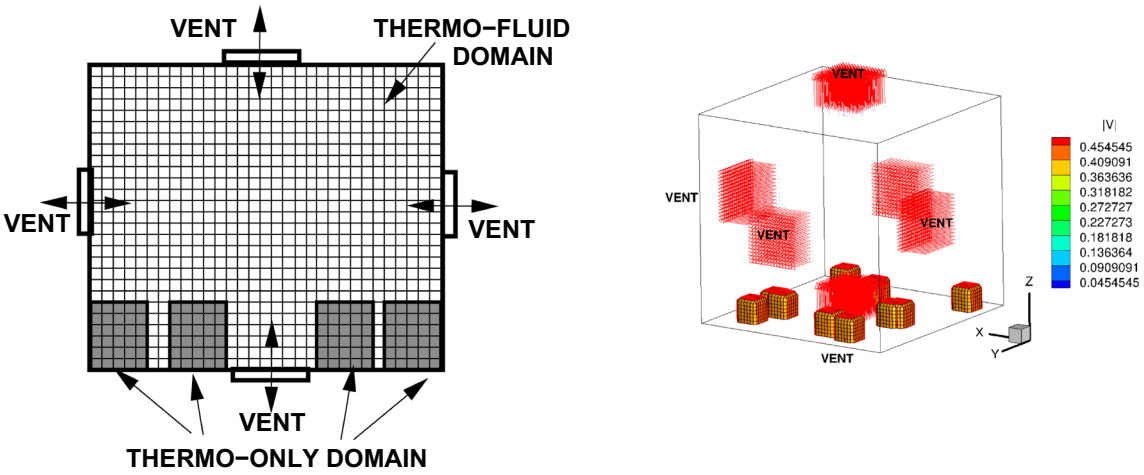

Fig. 2 A cross-section of the schematic for an energy management system with the two types of domains: **a** A thermo-fluid domain (ambient interior domain) and **b** a thermo-only domain(processors), where the velocity field is set to zero ($\boldsymbol{v} = \boldsymbol{0}$). Figure 3 illustrates the results (evolution of flow streamlines)

Remark 3 To illustrate second-order accuracy, consider a Taylor series expansion for an arbitrary function w

$$w(x+\Delta x) = w(x) + \frac{\partial w}{\partial x}|_x \Delta x + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}|_x (\Delta x)^2 + \frac{1}{6}\frac{\partial^3 w}{\partial x^3}|_x (\Delta x)^3 + \mathcal{O}((\Delta x)^4) \tag{3.12}$$

and

$$w(x-\Delta x) = w(x) - \frac{\partial w}{\partial x}|_x \Delta x + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}|_x (\Delta x)^2 - \frac{1}{6}\frac{\partial^3 w}{\partial x^3}|_x (\Delta x)^3 + \mathcal{O}((\Delta x)^4) \tag{3.13}$$

Subtracting the two expressions yields

$$\frac{\partial w}{\partial x}|_x = \frac{w(x+\Delta x) - w(x-\Delta x)}{2\Delta x} + \mathcal{O}((\Delta x)^2). \tag{3.14}$$

All other derivatives follow from this basic process, which is relatively standard in the Finite Difference community.

3.3 Overall iterative (implicit) solution method

Following the basic framework in Zohdi [77–86], let us consider the finite difference nodes (i):

$$\boldsymbol{v}_i^{L+1,K} = \boldsymbol{v}_i^{L} + \Delta t \left(\phi \boldsymbol{L}_i^{L+1,K-1} + (1-\phi)\boldsymbol{L}_i^{L} \right), \tag{3.15}$$

where i is the node counter, which is of the form

$$\boldsymbol{v}_i^{L+1,K} = \mathcal{G}(\boldsymbol{v}_i^{L+1,K-1}) + R_i, \tag{3.16}$$

where $K = 1, 2, 3, \ldots$ is the index of iteration within time step $L+1$ and

- $\mathcal{G}(\boldsymbol{v}_i^{L+1,K-1}) = \phi \Delta t \boldsymbol{L}_i^{L+1,K-1}$ and
- $R_i = \boldsymbol{v}_i^{L} + \Delta t(1-\phi)\boldsymbol{L}_i^{L}$.

The term R_i is a remainder term that does not depend on the current solution (only on the previous time step's solution). The convergence of such a scheme is dependent on the behavior of $\mathcal{G}$. Namely, a sufficient condition for convergence is that $\mathcal{G}$ is a contraction mapping for all $\boldsymbol{v}_i^{L+1,K}$, $K = 1, 2, 3, \ldots$ In order to investigate this further, we define the iteration error as

$$\epsilon_i^{L+1,K} \stackrel{\text{def}}{=} \boldsymbol{v}_i^{L+1,K} - \boldsymbol{v}_i^{L+1}. \tag{3.17}$$

A necessary restriction for convergence is iterative self-consistency, i.e. the "exact" (discretized) solution must be represented by the scheme, $\boldsymbol{v}_i^{L+1} = \mathcal{G}(\boldsymbol{v}_i^{L+1}) + R_i$. Enforcing this restriction, a sufficient condition for convergence is the existence of a contraction mapping

$$||\underbrace{\boldsymbol{v}_i^{L+1,K} - \boldsymbol{v}_i^{L+1}}_{\epsilon_i^{L+1,K}}|| = ||\mathcal{G}(\boldsymbol{v}_i^{L+1,K-1}) - \mathcal{G}(\boldsymbol{v}_i^{L+1})|| \leq \eta^{L+1,K} ||\boldsymbol{v}_i^{L+1,K-1} - \boldsymbol{v}_i^{L+1}||, \tag{3.18}$$

where, if $0 \leq \eta^{L+1,K} < 1$ for each iteration K, then $\epsilon_i^{L+1,K} \rightarrow \boldsymbol{0}$ for any arbitrary starting value $\boldsymbol{r}_i^{L+1,K=0}$, as $K \rightarrow \infty$, which is a contraction condition that is sufficient, but not necessary, for convergence. The convergence of Eq. 3.15 is scaled by $\eta \propto \frac{(\phi \Delta t)^2}{m_i}$. Therefore, we see that the contraction constant of $\mathcal{G}$ is:

- directly dependent on the magnitude of the interaction forces ($||\boldsymbol{L}||$),
- directly proportional to $(\Delta t)^2$.

Thus, decreasing the time step size improves the convergence. *In order to maximize the time-step sizes (to decrease overall computing time) and still meet an error tolerance on the numerical solution's accuracy*, we build on an approach originally developed for continuum thermo-chemical multifield problems (Zohdi [77–86]), where one assumes: (1) $\eta^{L+1,K} \approx S(\Delta t)^p$, ($S$ is a constant) and (2) the error within an iteration behaves according to $(S(\Delta t)^p)^K \epsilon^{L+1,0} = \epsilon^{L+1,K}$, $K = 1, 2, \ldots$, where $\epsilon^{L+1,0} = \boldsymbol{v}_i^{L+1,K=1} - \boldsymbol{v}_i^L$ is the initial norm of the iterative (relative) error and S is intrinsic to the system. For example, for second-order problems, due to the quadratic dependency on Δt, $p \approx 2$. The objective is to meet an error tolerance in exactly a preset (the analyst sets this) number of iterations. To this end, one writes $(S(\Delta t_{\text{tol}})^p)^{K_d} \epsilon^{L+1,0} = TOL$, where TOL is a tolerance and where K_d is the number of desired iterations. If the error tolerance is not met in the desired number of iterations, the contraction constant $\eta^{L+1,K}$ is too large. Accordingly, one can solve for a new smaller step size, under the assumption that S is constant,

$$\Delta t_{\text{tol}} = \Delta t \underbrace{\left(\frac{(\frac{TOL}{\epsilon^{L+1,0}})^{\frac{1}{pK_d}}}{(\frac{\epsilon^{L+1,K}}{\epsilon^{L+1,0}})^{\frac{1}{pK}}} \right)}_{\stackrel{\text{def}}{=} \Lambda_K}. \tag{3.19}$$

The assumption that S is constant is not critical, since the time steps are to be recursively refined and unrefined throughout the simulation. Clearly, the expression in Eq. 3.19 can also be used for time step enlargement, if convergence is met in less than K_d iterations (typically chosen to be between five to ten iterations). Specifically, the solution steps are, within a time-step:

- (1): Start a global fixed iteration (set $i = 1, \ldots, N_n$ (node counter) and $K = 0$ (iteration counter))
- (2): If $i > N_n$ then go to (4)
- (3): If $i \leq N_n$ then:
  - (a) Compute the velocity $\boldsymbol{v}_i^{L+1,K}$
  - (b) Go to (2) for the next node ($i = i + 1$)
- (4): Measure error (normalized) quantities
  - (a) $\epsilon_K \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{N_n} ||\boldsymbol{v}_i^{L+1,K} - \boldsymbol{v}_i^{L+1,K-1}||}{\sum_{i=1}^{N_n} ||\boldsymbol{v}_i^{L+1,K}||}$
  - (b) $E_K \stackrel{\text{def}}{=} \frac{\epsilon_K}{TOL_r}$
  - (c) $\Lambda_K \stackrel{\text{def}}{=} \left(\frac{(\frac{TOL}{\epsilon^0})^{\frac{1}{pK_d}}}{(\frac{\epsilon^K}{\epsilon^0})^{\frac{1}{pK}}} \right)$.
- (5): If the tolerance is met: $E_K \leq 1$ and $K < K_d$ then
  - (a) Increment time: $t = t + \Delta t$
  - (b) Construct the next time step: $\Delta t^{new} = \Lambda_K \Delta t^{old}$,
  - (c) Select the minimum size: $\Delta t = MIN(\Delta t^{lim}, \Delta t^{new})$ and go to (1)
- (6): If the tolerance is not met: $E_K > 1$ and $K < K_d$ then
  - (a) Update the iteration counter: $K = K + 1$
  - (b) Reset the node counter: $i = 1$
  - (c) Go to (2)
- (7): If the tolerance is not met ($E_K > 1$) and $K = K_d$ then
  - (a) Construct a new time step: $\Delta t^{new} = \Lambda_K \Delta t^{old}$
  - (b) Restart at time t and go to (1)

Time-step size adaptivity is critical, since the system's dynamics and configuration can dramatically change over the course of time, possibly requiring quite different time step sizes to control the iterative error. However, to maintain the accuracy of the time-stepping scheme, one must respect an upper bound dictated by the discretization error, i.e., $\Delta t \leq \Delta t^{lim}$. Note that in step (5), Λ_K may enlarge the time-step if the error is lower than the preset tolerance. At a given time, once the process is complete, then the time is incremented forward and the process is repeated. The overall goal is to deliver solutions where the iterative error is controlled and the temporal discretization accuracy dictates the upper limit on the time step size (Δt^{lim}). Clearly, there are various combinations of solution methods that one can choose from. For example, for the overall field coupling, one may choose implicit or explicit staggering and within the staggering process, either implicit ($0 < \phi \leq 1$) or explicit time-stepping ($\phi = 0$), and, in the case of implicit time-stepping, iterative or direct solvers for the Navier-Stokes equations. Furthermore, one could employ internal iterations for each field equation, then update, more sophisticated metrics for certain components of the error, etc.

3.4 Model problem and numerical example

As an example, we consider the direct numerical simulation of the fluid flow using the Navier-Stokes equations and first law of thermodynamics (streamlines shown) with 4 side vents, a bottom vent and a top vent for a pod of 10 processors with base-cooling. Figure 2 illustrates a cross-section of the schematic for an energy management system with the two types of domains: (a) A thermo-fluid domain (ambient

interior domain) and (b) a thermo-only domain (processors), where the velocity field is set to zero ($\boldsymbol{v} = \mathbf{0}$). Figure 3 illustrates the results (evolution of flow streamlines). In the model problem, we have made the vent sizes 0.25 that of the wall, the processor heat zone height $h = 0.1$ of the wall and width $w = 0.1$ of the wall. A $20 \times 20 \times 20$ stencil grid was used. A standard Macbook Pro laptop was used for all calculations using a FORTRAN code written by the author. We consider base cooling using the formula for the power extracted

$$Power\ extracted = -(D + Asin(2\pi\omega\frac{t}{T})) = \rho z, \quad (3.20)$$

where D is the DC-power extracted, A is the amplitude of the AC part of the power extracted, ω is the alternating frequency, t is the time and T is the total time period. This cooling power extraction is applied to all parts of the processor domain below a height h and within a width w, as shown in Figs. 1 and 2 as a negative source term in Eq. 2.28.

4 Genomic machine-learning cooling optimization

The rapid rate at which these simulations can be completed enables the ability to explore inverse problems seeking to determine what parameter combinations can deliver a desired result (Fig. 4). 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 [88–91], we formulate the objective as a cost function minimization problem that seeks system parameters that match a desired response

$$\Pi(\Lambda_1, \ldots \Lambda_N) \stackrel{\text{def}}{=} w_1\Pi^{(1)} + w_2\Pi^{(2)} + w_3\Pi^{(3)} + w_4\Pi^{(4)} \stackrel{\text{def}}{=} \Pi^{total}, \quad (4.1)$$

where the error in achieving the target temperature is

$$\Pi^{(1)} = \frac{||\theta^{target} - \theta^{simulated}||}{||\theta^{target}||} \quad (4.2)$$

and the normalized base cooling power is

$$\Pi^{(2)} = \frac{||\int_0^T (D + Asin(2\pi\omega\frac{t}{T}))dt - 0||}{||D^{max}||}, \quad (4.3)$$

and the normalized ventilation flow power is

$$\Pi^{(3)} = \frac{\sqrt{\Sigma_{i=1}^{6}(v_i)^2} - 0}{||v^{max}||}, \quad (4.4)$$

and normalized cooling power in the vents is

$$\Pi^{(4)} = \frac{\sqrt{\Sigma_{i=1}^{6}(\theta_i - \theta^a)^2} - 0}{||\theta^{max}||}. \quad (4.5)$$

We systematically minimize Eq. 4.1, $min_{\Lambda}\Pi$, 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.

4.1 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 methods. This motivates nonderivative search methods, for example those found in Machine-Learning Algorithms (MLA's). 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 (89, 90]), as well as Goldberg [94], Davis [95], Onwubiko [96] and Goldberg and Deb [97]. A description of the algorithm will be described next (Zohdi [88–91]).

4.2 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. 4):

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 $\boldsymbol{\Lambda}^i$, $i = 1, \ldots, S$ 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*

4.3 Specifics

Following Zohdi [88–91], the algorithm is as follows:

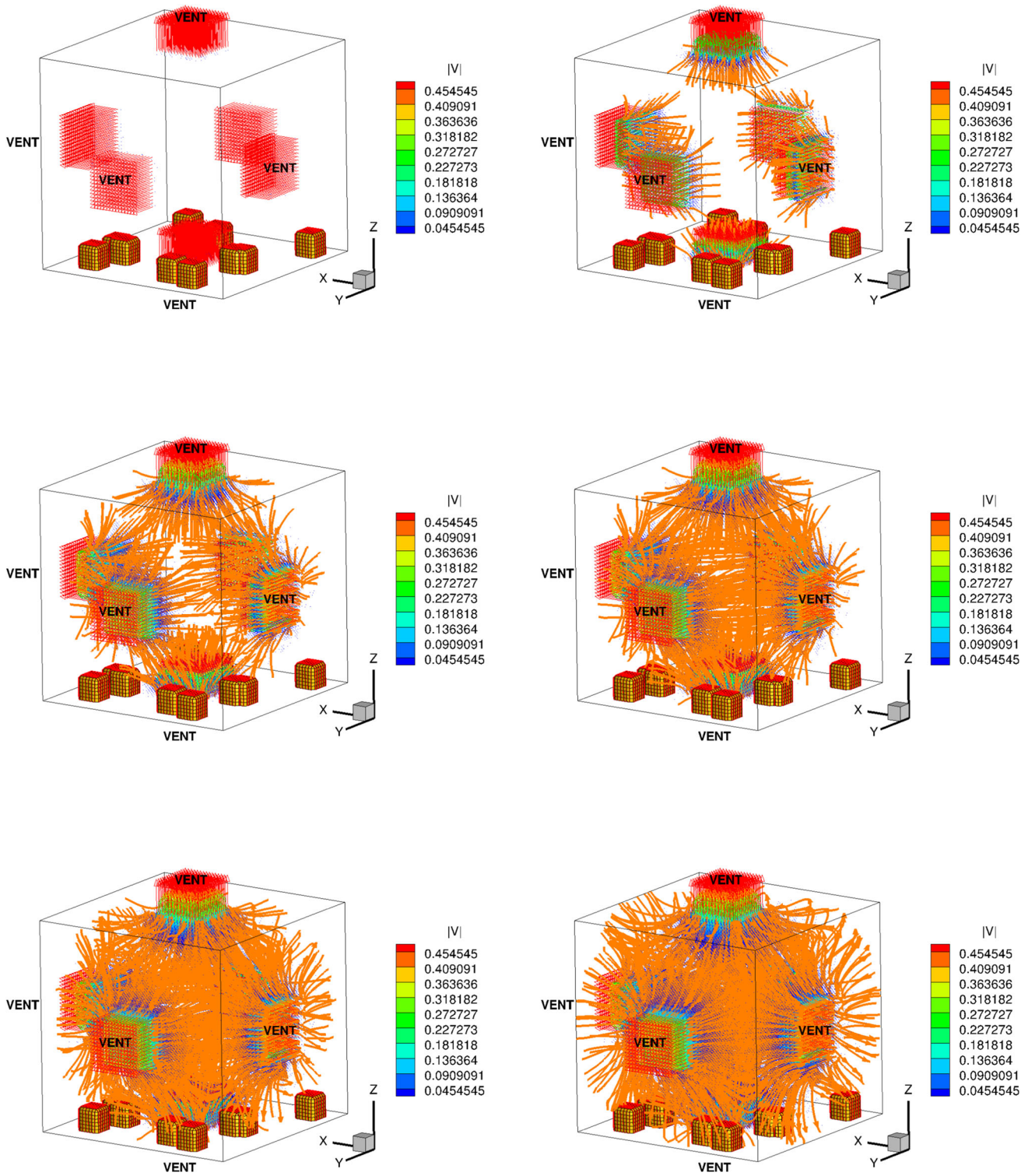

Fig. 3 Successive frames of flow using a direct numerical simulation of the fluid flow using the Navier–Stokes equations (streamlines shown) with 4 side vents, a bottom vent and a top vent for a pod of 10 processors, also with base-cooling. The evolution of flow streamlines are shown

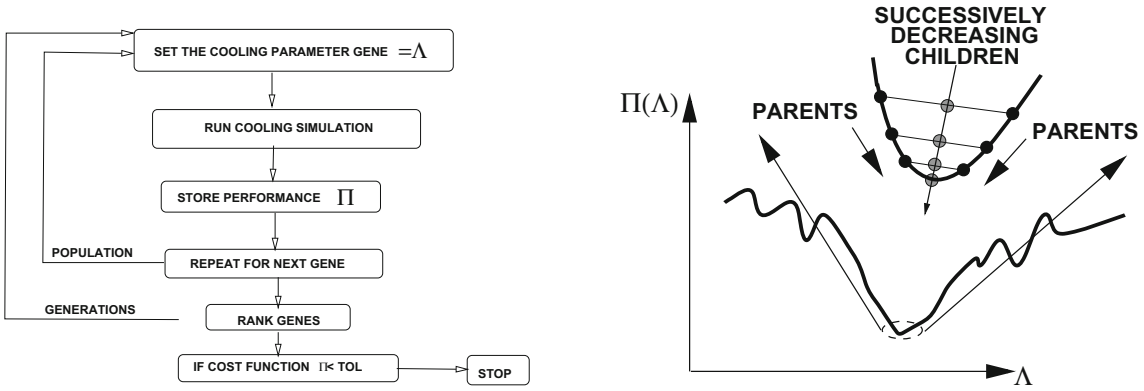

Fig. 4 The basic action of a MLA/GA-machine learning algorithm/genetic algorithm. Zohdi [88–91]

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

$$\mathbf{\Lambda}^i \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \Lambda_1^i \\ \Lambda_2^i \\ \Lambda_3^i \\ \cdots \\ \Lambda_N^i \end{array} \right\} \tag{4.6}$$

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

$$\begin{aligned} \boldsymbol{\lambda}^i &\stackrel{\text{def}}{=} \mathbf{\Phi} \circ \mathbf{\Lambda}^i + (\mathbf{1} - \mathbf{\Phi}) \circ \mathbf{\Lambda}^{\mathbf{i+1}} \\ &\stackrel{\text{def}}{=} \left\{ \begin{array}{c} \phi_1 \Lambda_1^i \\ \phi_2 \Lambda_2^i \\ \phi_3 \Lambda_3^i \\ \cdots \\ \phi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1-\phi_1)\Lambda_1^{i+1} \\ (1-\phi_2)\Lambda_2^{i+1} \\ (1-\phi_3)\Lambda_3^{i+1} \\ \cdots \\ (1-\phi_N)\Lambda_N^{i+1} \end{array} \right\} \end{aligned} \tag{4.7}$$

and

$$\begin{aligned} \boldsymbol{\lambda}^{i+1} &\stackrel{\text{def}}{=} \mathbf{\Psi} \circ \mathbf{\Lambda}^i + (\mathbf{1} - \mathbf{\Psi}) \circ \mathbf{\Lambda}^{\mathbf{i+1}} \\ &\stackrel{\text{def}}{=} \left\{ \begin{array}{c} \psi_1 \Lambda_1^i \\ \psi_2 \Lambda_2^i \\ \psi_3 \Lambda_3^i \\ \cdots \\ \psi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1-\psi_1)\Lambda_1^{i+1} \\ (1-\psi_2)\Lambda_2^{i+1} \\ (1-\psi_3)\Lambda_3^{i+1} \\ \cdots \\ (1-\psi_N)\Lambda_N^{i+1} \end{array} \right\} \end{aligned} \tag{4.8}$$

where for this operation, the ϕ_i and ψ_i are random numbers, such that $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$, 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
- Option: One can rescale and restart search around best performing parameter set every few generations, thus refocussing the computation effort around the most promising (optimal) areas of design space.

Remark 4 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. Furthermore, retained parents do not need to be reevaluated, making the algorithm less computationally expensive, since these parameter sets do not have to be reevaluated (or ranked) in the next generation. Numerous studies of the author (Zohdi [88–91]) 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. After application of such a global search 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 they are generally much more efficient for convex optimization of smooth functions. An exhaustive review of these

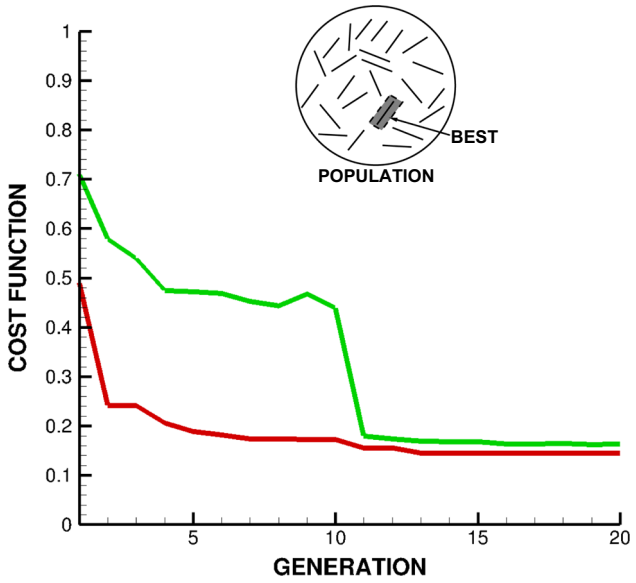

Fig. 5 Shown are the cost function for the best performing gene (*red*) as a function of successive generations, as well as the average cost function of the entire population of genes (*green*). We allowed the MLA/GA to readapt every 10 generations. Often, this action is more efficient than allowing the algorithm not to readapt, since it probes around the current optimum for better local alternatives. The weights were all set to $w_1 = w_2 = w_3 = w_4 = 1$. The final cost functions were $\Pi^{(1)} = 0.01277$, $\Pi^{(2)} = 0.01873$, $\Pi^{(3)} = 0.07859$ and $\Pi^{(4)} = 0.03537$. The total is $\Pi^{total} = 0.1453$

methods can be found in the texts of Luenberger [98] and Gill, Murray and Wright [99].

4.4 Algorithmic settings

In the upcoming example, the design parameters $\mathbf{\Lambda} = \{\Lambda_1, \Lambda_2 \ldots \Lambda_N\}$ are optimized over the search intervals (15 variables): $\mathbf{\Lambda}_i^- \leq \mathbf{\Lambda}_i \leq \mathbf{\Lambda}_i^+$, $i = 1, 2, \ldots 15$. Specifically (Fig. 5), we varied the 15 parameters associated with vents and base system cooling and used the following MLA settings:[2]

- Number of design variables: 15,
- 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: 10 and
- Number of generations: 20.

[2] As in the previous example, a $20 \times 20 \times 20$ stencil grid was used along with a standard Macbook Pro laptop for all calculations using a FORTRAN code written by the author.

4.5 Parameter search ranges and results

We considered a 15 parameter cooling system design, with a target average temperature of $\theta^{desired} = 305 K^o$ in a lower layer of the data-center $x^{layer} \leq 0.1$ of the wall height (Fig. 1). The following search parameter ranges were used (with $w_1 = w_2 = w_3 = w_4 = 1$):

- $\Lambda_{i=1-6}$ = Flow in/out of vent i:$\Lambda_i^- = -5 \leq \Lambda_i \leq \Lambda_i^+ = 5$,
- Λ_7 = DC cooling parameter:$\Lambda_7^- = 10 \times 10^6 \leq \Lambda_7 \leq \Lambda_7^+ = 50 \times 10^6$,
- Λ_8 = AC cooling parameter:$\Lambda_8^- = 0 \leq \Lambda_8 \leq \Lambda_8^+ = 10 \times 10^6$,
- Λ_9 = AC cooling frequency:$\Lambda_9^- = 0 \leq \Lambda_9 \leq \Lambda_9^+ = 10$ and
- $\Lambda_{i=10-15}$ = Temperature in vent i:$\Lambda_i^- = 280 \leq \Lambda_i \leq \Lambda_i^+ = 320$.

Figure 5 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*). We allowed the MLA/GA to readapt every 10 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, although for this model problem the effect was mild. The weights were all set to $w_1 = w_2 = w_3 = w_4 = 1$. The final cost functions were $\Pi^{(1)} = 0.01277$, $\Pi^{(2)} = 0.01873$, $\Pi^{(3)} = 0.07859$ and $\Pi^{(4)} = 0.03537$. The total is $\Pi^{total} = 0.1453$. Table 1 shows the final design parameters. The entire 20 generation simulation, with 24 genes per evaluation (480 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 approximately one second, thus hundreds 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.

5 Summary and extensions

In summary, the massive growth in data-centers has led to increased interest and regulations for management of waste heat and its utilization. This work sought to develop a combined Digital-Twin and Machine-Learning framework to optimize such systems by controlling both the ventilation and the cooling of the bases of data units/processors in the system. This framework ascertains optimal cooling strategies to deliver a target temperature in the system using a

Table 1 The system parameters ($\mathbf{\Lambda}_1 - \mathbf{\Lambda}_{15}$) for the best performing design (gene) with design weights of $w_1 = 1$, $w_2 = 1$, $w_3 = 1$ and $w_4 = 1$

$\mathbf{\Lambda}_1$	$\mathbf{\Lambda}_2$	$\mathbf{\Lambda}_3$	$\mathbf{\Lambda}_4$	$\mathbf{\Lambda}_5$	$\mathbf{\Lambda}_6$	$\mathbf{\Lambda}_7$	$\mathbf{\Lambda}_8$
-0.0231	0.3295	0.8116	0.0210	0.2258	0.3274	17276463	4718675
$\mathbf{\Lambda}_9$	$\mathbf{\Lambda}_{10}$	$\mathbf{\Lambda}_{11}$	$\mathbf{\Lambda}_{12}$	$\mathbf{\Lambda}_{13}$	$\mathbf{\Lambda}_{14}$	$\mathbf{\Lambda}_{15}$	Π
5.819	316.7	307.6	313.3	299.7	314.1	309.3	0.1453

minimum amount of energy. A model problem was studied whereby, for a data-center, the design variables are the flow rates and air-cooling of the multiple ventilation ports and ground-level conduction-based base-cooling of processors. A fast solution method, based on a CFD representation of the data-center was developed using a stencil-based discretization of the Navier-Stokes equations and the first law of thermodynamics, which is combined with a genomic-based machine-learning algorithm to develop a digital-twin of the system, i.e. a digital replica that can run in real-time or faster than the actual physical system, which is suitable as either a design tool or a controller. Numerical examples were provided to illustrate the framework. Ultimately, the use of such a model for real time control may need the inclusion of simplified reduced-order models that can be trained on the data generated by more complex models, such as the one introduced in the body of this work. For example, on the simplest level this model simulates the temperature in a data-center and attempts to cool it to a target. Accordingly, consider a simplified reduced-order model for the overall balance of heat:

$$mC\frac{d\theta}{dt} = \frac{dH^{gen}}{dt} - \frac{dH^{out}}{dt}, \tag{5.1}$$

with constraints

$$if\ \theta > \theta^{target}\ then \frac{dH^{out}}{dt} = a(\theta^{target} - \theta), \tag{5.2}$$

and

$$if\ \theta \leq \theta^{target}\ then \frac{dH^{out}}{dt} = 0 \tag{5.3}$$

where the effective parameters of the simplified system are

- m = effective mass of data-center,
- θ = effective temperature of data-center,
- θ^{target} = effective target temperature of data-center,
- C = effective heat capacity of data-center,
- H^{gen} = effective heat energy generated by data-center,
- H^{out} = effective heat energy extracted by cooling system and
- a = effective rate parameter.

A simple explicit numerical solution is

$$mC\frac{\theta(t+\Delta t) - \theta(t)}{\Delta t} = \frac{dH^{gen}}{dt}(t) - \frac{dH^{cool}}{dt}(t) \tag{5.4}$$

and

$$\theta(t+\Delta t) = \theta(t) + \frac{\Delta t}{mC}(\frac{dH^{gen}}{dt}(t) - \frac{dH^{cool}}{dt}(t)) \tag{5.5}$$

For the next time step, the cooling needed predicted by the model to achieve the target temperature is

$$\frac{dH^{cool,predicted}}{dt} = mC\frac{\theta^{target} - \theta(t+\Delta t)}{\Delta t} - \frac{dH^{gen}}{dt} \tag{5.6}$$

This is an explicit time-staggered predictor-corrector method. However, for this to have some connection to physical reality, the model and effective parameters needs to be calibrated, i.e. 'trained' using more complex models, such as the ones developed in this work. If calibrated properly, the simplified model can serve as an instantaneous real-time controller. Alternatively, or in conjunction with the proceeding approaches, paradigms, such neural nets, which are based on constructing simple input-output type models that are also adept for such tasks and are essentially, adaptive nonlinear regressions of the form $OUTPUT = \mathcal{B}(INPUT, W_1, W_2, \ldots W_N)$ where $\mathcal{B}$ is an Artificial Neural Network (ANN) constructed from:

- *SYNAPSES*, which multiply inputs by weights that represent the inputs' relevance to the desired output,
- *NEURONS*, which add outputs from all incoming synapses and applies activation functions and
- *TRAINING*, which recalibrates the weights to achieve a desired overall output.

Ultimately, one constructs a system with optimized weights to mimic an artificial 'input-output' brain for rapid control. Blending of these various paradigms (complex models, simplified reduced-order models and neural nets) is the subject of current work of the author.

Acknowledgements This work has been partially supported by the UC Berkeley College of Engineering and the USDA AI Institute for Next Generation Food Systems (AIFS), USDA award number 2020-67021-32855.

Appendix: Related websites

1. https://www.datacenterdynamics.com/en/analysis/taking-next-steps-stockholm-circular-city/
2. https://www.google.com/about/datacenters/efficiency/
3. https://en.wikipedia.org/wiki/Data_center
4. https://eta.lbl.gov/publications/united-states-data-center-energy
5. https://energyinnovation.org/2020/03/17/how-much-energy-do-data-centers-really-use/
6. https://www.nature.com/articles/d41586-018-06610-y
7. https://www.energy.gov/eere/buildings/data-centers-and-servers
8. https://www.vxchnge.com/blog/growing-energy-demands-of-data-centers
9. https://www.sciencedirect.com/science/article/pii/S0306261921003019
10. https://www.wsj.com/articles/data-center-company-aligned-energy-raises-1-25-billion-in-debt-to-fund-sustainable-facilities-11629407307mod=hp_minor_pos4
11. https://datacenterfrontier.com/waste-heat-utilization-data-center-industry/

Appendix: Further discussion on fluid mechanics models

Although we considered an incompressible thermally-insensitive fluid in the body of the work, for completeness, we briefly discuss enhancements to such models.

Pressure-density approximation

There are a variety of possible Equations of State that connect the density to the pressure, such as a Boussinesq-like relation, which is adequate to describe dense gases and fluids, derived from[3]

$$\rho \approx \rho_o(P_o) + \frac{\partial \rho}{\partial P}\Delta P, \tag{8.1}$$

where ρ_o and P_o are reference values and $\Delta P = P - P_o$. We define the bulk (compressibility) modulus by $\zeta \stackrel{\text{def}}{=} \rho \frac{\partial P}{\partial \rho}$, yielding

$$\rho \approx \rho_o \left(1 + \frac{1}{\zeta}\Delta P\right) \Rightarrow P \approx P_o + \zeta\left(\frac{\rho}{\rho_o} - 1\right). \tag{8.2}$$

For a constant density case, $\rho = \rho_o$, and utilizing the Boussinesq-like relation, $P = P_o$.

[3] We have ignored thermal effects in this representation.

Buoyancy

Although we will not consider buoyancy in the present analysis, for completeness we illustrate a typical model. Consider the following decomposition of the body forces:

$$\boldsymbol{f} = \rho \boldsymbol{g} = \rho_o \boldsymbol{g} + (\rho - \rho_o)\boldsymbol{g}. \tag{8.3}$$

Now we approximate

$$(\rho - \rho_o)\boldsymbol{g} \approx -\rho_o \beta(\theta - \theta_o)\boldsymbol{g}, \tag{8.4}$$

where β is the thermal expansion coefficient. Thus,

$$\Delta\rho = (\rho - \rho_o) \approx -\rho_o \beta(\theta - \theta_o), \tag{8.5}$$

thus

$$\rho \approx \rho_o(1 - \beta(\theta - \theta_o)). \tag{8.6}$$

A generalization

$$\rho \approx \rho_o e^{-\beta(\theta - \theta_o)} \approx \rho_o(1 - \beta(\theta - \theta_o)) + \cdots \tag{8.7}$$

References

1. Masanet E, Shehabi A, Lei N, Smith S, Koomey J (2020) Recalibrating global data-center energy-use estimates. Science 367(6481):984–986
2. Belkhir L, Elmeligi A (2018) Assessing ICT global emissions footprint: trends to 2040 & recommendations. J Clean Prod 177:448–463
3. Jones N (2018) How to stop data centres from gobbling up the worlds electricity. Nature. 561(7722):163–166. https://doi.org/10.1038/D41586-018-06610-Y
4. Shehabi A et al (2016) United States data-center Energy Usage Report. No. LBNL-1005775. Lawrence Berkeley National Lab. (LBNL), Berkeley, CA, USA
5. U.S. Department Of Energy (2020) Annual energy outlook 2020. https://www.eia.gov/Outlooks/Aeo/
6. U.S. Energy Information Administration (2020) How much carbon dioxide is produced per kilowatthour of U.S. Electricity Generation. https://Www.eia.gov/Tools/Faqs/Faq.phpId=74&T=11
7. Brown R, Masanet E, Nordman B, Tschudi W, Shehabi A, Stanley J, Koomey J, Sartor D, Chan P, Loper J, Capana S, Hedman B, Duff R, Haines E, Sass D, Fanara A (2007) Report to congress on server and data-center energy efficiency: Public Law 109-431. Lawrence Berkeley National Laboratory, Berkeley, CA. LBNL-363E
8. Masanet E, Brown RE, Shehabi A, Koomey JG, Nordman B (2011) Estimating the energy use and efficiency potential of U.S. data-centers. In: Proceedings of the IEEE, vol 99, no 8
9. Koomey JG (2011) Growth in data-center electricity use 2005 to 2010. Analytics Press, Oakland, CA. http://www.analyticspress.com/datacenters.html
10. Upton F (2015) North American Energy Security and Infrastructure Act of 2015. H.R. 8, 114th Congress. https://www.congress.gov/bill/114th-congress/house-bill/8
11. Koomey J (2008) Worldwide electricity used in data-centers. Environ Res Lett 3:034008

12. Koomey JG (2007) Estimating total power consumption by servers in the U.S. and the World. February 15. http://www.mediafire.com/file/exywo1hf6ionskw/AMDserverpowerusecomplete-final.pdf
13. Van Heddeghem W, Lambert S, Lannoo B, Colle D, Pickavet M, Demeester P (2014) Trends in worldwide ICT electricity consumption from 2007 to 2012. Comput Commun 50:64–76
14. Lanzisera S, Nordman B, Brown RE (2012) Data network equipment energy use and savings potential in buildings. Energy Effic 5(2):149–162
15. Reviriego P, Maestro JA, Larrabeiti D (2010) Burst transmission for energy-efficient ethernet. IEEE Internet Comput 14(4):50–57
16. Dudkowski D, Hasselmeyer P (2015) Energy-efficient networking in modern data-centers. In: Samdanis K, Rost P, Maeder A, Meo M, Verikoukis C (eds) Green communications: principles, concepts and practice. Wiley, Hoboken
17. Tschudi W, Xu T, Sartor D, Stein J (2003) High performance data centers: a research roadmap. Berkeley, CA: Lawrence Berkeley National Laboratory. LBNL53483. http://hightech.lbl.gov/documents/DataCenters_Roadmap_Final.pdf
18. Greenberg S, Mills E, Tschudi B, Rumsey P, Myatt B (2006) Best practices for data-centers: lessons learned from benchmarking 22 data-centers. In: Proceedings of the ACEEE summer study on energy efficiency in buildings in Asilomar, CA. ACEEE, August, vol 3, pp 76–87. http://eetd.lbl.gov/emills/PUBS/PDF/ACEEE-datacenters.pdf
19. Malone C, Belady C (2006) Metrics to characterize datacenter & IT equipment energy use. In: Proceedings of the digital power forum, Richardson, TX, September
20. Sullivan A (2010) Energy star for data-centers. Green Grid Forum. February 4
21. Cheung IH, Greenberg S, Mahdavi R, Brown R, Tschudi W (2014) Energy efficiency in small server rooms: field surveys and findings. In: Proceedings the 2014 ACEEE summer study on energy efficiency in buildings. LBNL-6952E
22. Masanet E, Shehabi A, Ramakrishnan L, Liang J, Ma X, Walker B, Mantha P (2013) The energy efficiency potential of cloud-based software: a US case study. Lawrence Berkeley National Laboratory, Berkeley, CA
23. Shehabi A, Masanet E, Price H, Traber K, Horvath A, Nazaroff WW (2011) Data center design and location: consequences for electricity use and greenhouse-gas emissions. Build Environ 46(5):990–998
24. Koomey JG, Berard S, Sanchez M, Wong H (2011) Implications of historical trends in the electrical efficiency of computing. Ann Hist Comput IEEE 33(3):46–54
25. Koomey J, Samuel N (2015) Efficiency's brief reprieve: Moore's Law slowdown hits performance more than energy efficiency. In: IEEE Spectrum. April. http://spectrum.ieee.org/computing/hardware/moores-law-might-be-slowing-down-but-not-energyefficiency
26. Torcellini P, Long N, Judkoff R (2003) Consumptive water use for U.S. power production. NREL/TP-550- 33905, http://www.nrel.gov/docs/fy04osti/33905.pdf
27. Miller R (2009) Data-centers move to cut water waste. Data-center knowledge. http://www.datacenterknowledge.com/archives/2009/04/09/data-centers-move-to-cut-waterwaste
28. Fitzgerald D (2015) Data-centers and hidden water use. The Wall Street Journal. http://www.wsj.com/articles/SB10007111583511843695404581067903126039290
29. Federal Data-Center Consolidation Initiative (FDCCI) (2014) GSA FDCCI Inventory Data Fields. https://cio.gov/wp-content/uploads/downloads/2012/10/FDCCI-InventoryData-Fields-Release-2.pdf
30. Boyd A (2015) Experts: data-center consolidation goals not aggressive enough. Federal Times.com. http://www.federaltimes.com/story/government/it/datacenter/2015/02/17/data-center-consolidation-goals-not-aggressive/23556995/
31. McMillian R (2015) Zombie servers: theyre here and doing nothing but burning energy. The Wall Street Journal. September 13, 2015. http://www.wsj.com/articles/zombie-serverstheyre-here-and-doing-nothing-but-burning-energy-1442197727
32. Delforge P, Whitney J (2014) Issue paper: data-center efficiency assessment scaling up energy efficiency across the data-center industry: evaluating key drivers and barriers. Natural Resources Defense Council (NRDC)
33. Lo D, Cheng L, Govindaraju R, Ranganathan P, Kozyrakis C (2015) Heracles: improving resource efficiency at scale. In: Proceedings of the 42nd annual international symposium on computer architecture. ACM, pp 450–462
34. Masanet E, Shehabi A, Ramakrishnan L, Liang J, Ma X, Walker B, Hendrix V, Mantha P (2013) The energy efficiency potential of cloud-based software: a US case study. Lawrence Berkeley National Laboratory, Berkeley, CA
35. Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N (2010) ElasticTree: saving energy in data-center networks. NSDI 10:249–264
36. Hilty LM, Aebischer B (eds) (2015) ICT innovations for sustainability, advances in intelligent systems and computing. Springer, Cham
37. Masanet ER, Matthews HS (eds) (2010) Environmental applications of information and communication technology [special issue]. J Ind Ecol 14:685–862
38. Rejeski D (ed) (2002) E-Commerce, the Internet, and the Environment [special issue]. J Ind Ecol 6:1–161
39. Roberts S (2009) Measuring the relationship between ICT and the environment. Organization for Economic Co-operation and Development. http://www.oecd.org/sti/43539507.pdf
40. Auweter DK, Tahamtan A, Tjoa AM, Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G (eds) (2012) ICT as key technology against global warming, vol 7453. Lecture notes in computer science. Springer, Berlin. https://doi.org/10.1007/978-3-642-32606-6
41. Erdmann L, Hilty L, Goodman J, Arnfalk P (2004) The future impact of ICTs on environmental sustainability. Technical Report No. EUR 21384 EN. Institute for Prospective Technological Studies, Seville, Spain
42. Jorgensen MS, Andersen MM, Hansen A, Wenzel H, Thoning T, Pedersen UJ, Falch M, Rasmussen B, Olsen SI, Willum O (2006) Green technology foresight about environmentally friendly products and materials. Environmental Protection Agency Den. http://www2.mst.dk/udgiv/publications/2006/87-7052-216-2/pdf/87-7052-217-0.pdf
43. Romm J, Rosenfeld A, Herrmann S (1999) The internet economy and global warming: a scenario of the impact of e-commerce on energy and the environment. Cent. Energy Clim. Solut. Dec. http-www Cool-Co, Orgenergycecs Cfm
44. Laitner JA, Ehrhardt-Martinez K (2008) Information and communication technologies: the power of productivity (E081). American Council for an Energy-Efficient Economy, Washington, DC
45. Elliott N, Molina M, Trombley D (2012) A defining framework for intelligent efficiency. American Council for an Energy-Efficient Economy, Washington, DC. http://aceee.org/sites/default/files/publications/researchreports/e125.pdf
46. Accenture (2015) SMARTer2030: ICT solutions for 21st century challenges. http://smarter2030.gesi.org/downloads/Full_report2.pdf
47. Rattle R (2010) Computing our way to paradise: the role of internet and communication technologies in sustainable consumption and globalization. Rowman & Littlefield, Lanham

48. Berkhout F, Hertin J (2004) De-materialising and re-materialising: digital technologies and the environment. Futures 36:903–920. https://doi.org/10.1016/j.futures.2004.01.003
49. Borjesson Rivera M, Hakansson C, Svenfelt A, Finnveden G (2014) Including second order effects in environmental assessments of ICT. Environ Model Softw 56:105–115. https://doi.org/10.1016/j.envsoft.2014.02.005
50. Erdmann L, Hilty LM (2010) Scenario analysis: exploring the macroeconomic impacts of information and communication technologies on greenhouse gas emissions. J Ind Ecol 14:826–843. https://doi.org/10.1111/j.1530-9290.2010.00277.x
51. Koomey JG, Matthews HS, Williams E (2013) Smart everything: will intelligent systems reduce resource use. Annu Rev Environ Resour 38:311–343. https://doi.org/10.1146/annurev-environ-021512-110549
52. Yi L, Thomas HR (2007) A review of research on the environmental impact of e-business and ICT. Environ Int 33:841–849. https://doi.org/10.1016/j.envint.2007.03.015
53. Allenby B, Unger D (2001) Information technology impacts on the US energy demand profile. RAND Corp, Santa Monica
54. Azevedo IML (2014) Consumer end-use energy efficiency and rebound effects. Annu Rev Environ Resour 39:393–418. https://doi.org/10.1146/annurev-environ-021913-153558
55. Borenstein S (2013) A microeconomic framework for evaluating energy efficiency rebound and some implications. National Bureau of Economic Research
56. Hilty LM, Arnfalk P, Erdmann L, Goodman J, Lehmann M, Wager PA (2006) The relevance of information and communication technologies for environmental sustainability—a prospective simulation study. Environ Model Softw 21:1618–1629. https://doi.org/10.1016/j.envsoft.2006.05.007
57. Hesse M (2002) Shipping news: the implications of electronic commerce for logistics and freight transport. Resour Conserv Recycl 36:211–240
58. Shorr Packaging Corp (2015) The Amazon effect: impacts on shipping and retail. http://www.shorr.com/packaging-news/2015-06/amazon-effect-impacts-shipping-and-retail. Accessed 30 Nov 2015
59. Greening LA, Greene DL, Difiglio C (2000) Energy efficiency and consumption—the rebound effect—a survey. Energy Policy 28:389–401
60. Plepys A (2002) The grey side of ICT. Environ Impact Assess Rev 22:509–523
61. Williams E (2011) Environmental effects of information and communications technologies. Nature 479:354–358. https://doi.org/10.1038/nature10682
62. Weber C et al (2008) Life cycle comparison of traditional retail and e-commerce logistics for electronic products: a case study of buy.com. Green Design Institute, Carnegie Mellon University. http://www.ce.cmu.edu/~greendesign/research/Buy_com_report_final_030209.pdf
63. Shehabi A, Walker B, Masanet E (2014) The energy and greenhouse-gas implications of internet video streaming in the United States. Environ Res Lett 9:054007
64. Kitou E, Horvath A (2008) External air pollution costs of telework. Int J Life Cycle Assess 13:155–165
65. Henderson P, Waitner M (2013) Real-time energy management: a case study of three large commercial buildings in Washington, DC. Natural Resources Defense Council. http://www.nrdc.org/business/casestudies/files/tower-companies-case-study.pdf
66. Seidel S, Ye J (2012) Leading by example: using information and communication technologies to achieve federal sustainability goals. Center for Climate and Energy Solutions. http://www.c2es.org/docUploads/federal-sustainability-ict.pdf
67. Laitner JA, Ehrhardt-Martinez K (2008) Information and communication technologies: the power of productivity. American Council for an Energy-Efficient Economy. http://aceee.org/sites/default/files/publications/researchreports/E081.pdf
68. Baer WS, Hassell S, Vollaard BA (2002) Electricity requirements for a digital society. RAND Corporation, Santa Monica
69. Hilty LM et al (2006) The relevance of information and communication technologies for environmental sustainability—a prospective simulation study. Environ Model Softw 21:1618–1629
70. Romm J, Rosenfeld A, Herrmann S (1999) The internet economy and global warming: a scenario of the impact of e-commerce on energy and the environment. Cent. Energy Clim. Solut. Dec. http-www Cool-Co. Orgenergycecs Cfm. http://www.fraw.org.uk/files/tech/romm_1999.pdf
71. Laitner JA, McDonnell MT, Keller RM (2015) ICT-enabled intelligent efficiency: shifting from device-specific approaches to system optima. http://cda.iea4e.org/document/11/ict-enabled-intelligent-efficiency-shifting-from-device-specific-approachesto-system-optima
72. Horner N, Azevedo I (2016) Power usage effectiveness in datacenters: overloaded and underachieving. Electr J 29(4):61–69
73. Azevedo D, Rawson A (2008) Measuring data-center productivity. Metrics and Measurements Work Group. The Green Grid. http://www.thegreengrid.org/~/media/TechForumPresentations2008/Measuring_Data_Center_P roductivity.pdf lang=en
74. Kaplan J, Forrest W, Kindler N (2008) Revolutionizing data-center energy efficiency. McKinsey & Company. http://www.sallan.org/pdfdocs/McKinsey_Data_Center_Efficiency.pdf
75. Cappiello C, Chen D, Ferreria AM, Henis R, Jiang T, Kat I, Kipp A, Liu J, Sotnikov D, Vitali M (2011) Usage centric green performance indicators. http://www.sigmetrics.org/sigmetrics2011/greenmetrics/green11_Kat_email.pdf
76. Shehabi A, Smith SJ, Horner N, Azevedo I, Brown R, Koomey J, Masanet E, Sartor D, Herrlin M, Lintner W (2016) United States data-center energy usage report. Lawrence Berkeley National Laboratory, Berkeley, CA, LBNL-1005775
77. Zohdi TI (2002) An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids. Int J Numer Methods Eng 53:1511–1532
78. Zohdi TI (2005) Charge-induced clustering in multifield particulate flow. Int J Numer Methods Eng 62(7):870–898
79. Zohdi TI (2007) Computation of strongly coupled multifield interaction in particle–fluid systems. Comput Methods Appl Mech Eng 196:3927–3950
80. Zohdi TI (2010) On the dynamics of charged electromagnetic particulate jets. Arch Comput Methods Eng 17(2):109–135
81. Zohdi TI (2010) Simulation of coupled microscale multiphysical-fields in particulate-doped dielectrics with staggered adaptive FDTD. Comput Methods Appl Mech Eng 199:79–101
82. Zohdi TI (2012) Electromagnetic properties of multiphase dielectrics. A primer on modeling, theory and computation. Springer, Berlin
83. Zohdi TI (2012) Dynamics of charged particulate systems. Modeling, theory and computation. Springer, Berlin
84. Zohdi TI (2013) Numerical simulation of charged particulate cluster-droplet impact on electrified surfaces. J Comput Phys 233:509–526
85. Zohdi TI (2014) A direct particle-based computational framework for electrically-enhanced thermo-mechanical sintering of powdered materials. Math Mech Solids. https://doi.org/10.1007/s11831-013-9092-6
86. Zohdi TI (2014) Embedded electromagnetically sensitive particle motion in functionalized fluids. Comput Part Mech 1:27–45
87. Zohdi TI (2018) Electrodynamic machine-learning-enhanced fault-tolerance of robotic free-form printing of complex mixtures. Comput Mech. https://doi.org/10.1007/s00466-018-1629-y
88. Zohdi TI (2009) Mechanistic modeling of swarms. Comput Methods Appl Mech Eng 198(21–26):2039–2051

89. Zohdi TI (2018) Multiple UAVs for Mapping: a review of basic modeling, simulation and applications. Annu Rev Environ Resour. https://doi.org/10.1146/annurev-environ-102017-025912
90. Zohdi TI (2019) The game of drones: rapid agent-based machine-learning models for multi-UAV path planning. Comput Mech. https://doi.org/10.1007/s00466-019-01761-9
91. Zohdi T (2020) A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments. Comput Methods Appl Mech Eng. https://doi.org/10.1016/j.cma.2020.112907
92. Holland JH (1975) Adaptation in natural & artificial systems. University of Michigan Press, Ann Arbor
93. Holland JH, Miller JH (1991) Artificial adaptive agents in economic theory (PDF). Am Econ Rev 81(2):365–371 (**Archived from the original (PDF) on October 27, 2005**)
94. Goldberg DE (1989) Genetic algorithms in search, optimization & machine learning. Addison-Wesley, Reading
95. Davis L (1991) Handbook of genetic algorithms. Thompson Computer Press, Boston
96. Onwubiko C (2000) Introduction to engineering design optimization. Prentice-Hall, Englewood Cliffs
97. Goldberg DE, Deb K (2000) Special issue on genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):121–124
98. Luenberger D (1974) Introduction to linear & nonlinear programming. Addison-Wesley, Menlo Park
99. Gill P, Murray W, Wright M (1995) Practical optimization. Academic Press, London