Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 363 (2020) 112907

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments

T.I. Zohdi

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

Received 2 November 2019; received in revised form 1 February 2020; accepted 3 February 2020
Available online xxxx

Abstract

The objective of this work is to illustrate how to algorithmically integrate Machine-Learning Algorithms (MLA's) with multistage/multicomponent fire spread models. In order to tangibly illustrate this process, this work develops a framework for a specific model problem combining: (I) a meshless discrete element "submodel" that tracks the trajectory of airborne hot particles/embers, subject to prevailing wind velocities and updrafts, (II) a topographical "submodel" of the ambient combustible material whereby airborne embers that make contact are allowed to start secondary fires (if conditions are appropriate), combined with ground-based surface spread and burn rates for generating new embers, new updrafts (due to hot air), etc., and (III) a Machine-Learning Algorithm to rapidly ascertain the multi-submodel system parameters that force the overall model to match observations. The submodels compute both ground and airborne hot-ember driven fire propagation, as well as subsequent distribution of debris/soot, which is important for air-quality assessment. The overall framework is designed for use in *digital twin* technology, which refers to an adaptive digital replica of a physical system, whereby model updates are continuously in near real-time. This necessitates a rapid simulation paradigm that can easily interface with telecommunications, cameras and sensors. The presented framework is designed to run quickly on laptops and hand held devices, with the guiding principle being to make it potentially useful for first-responders *in real-time*.

Keywords: Fire-propagation; Ember flow; Digital-twin; Machine-learning algorithms

1. Introduction

Uncontrolled large-scale fires are growing in severity worldwide. Such fires are occurring in countries around the globe and will continue to drain their resources until effective solutions are in place (Fig. 1). For example, in 2017, US fire departments responded to an estimated 1,319,500 fires which caused over 3400 civilian fatalities, 14,670 civilian fire injuries and an estimated 23 billion in direct property losses. Every minute delayed in the primary response for certain life-threatening medical emergencies leads to a measurable effect on mortality, especially in the first five minutes in the response interval, where rapid intervention makes the greatest difference. In 2017, there was a civilian fire death every 2 h, 34 min, and a civilian fire injury every 36 min (see Evarts [1], Karter [2], Zohdi [3], Grant et al. [4] and McKinzie [5] for more details).

E-mail address: zohdi@berkeley.edu.

https://doi.org/10.1016/j.cma.2020.112907
0045-7825/© 2020 Elsevier B.V. All rights reserved.

Fig. 1. Huge sprawling wildfires in the American west (public domain photo from https://pixabay.com).

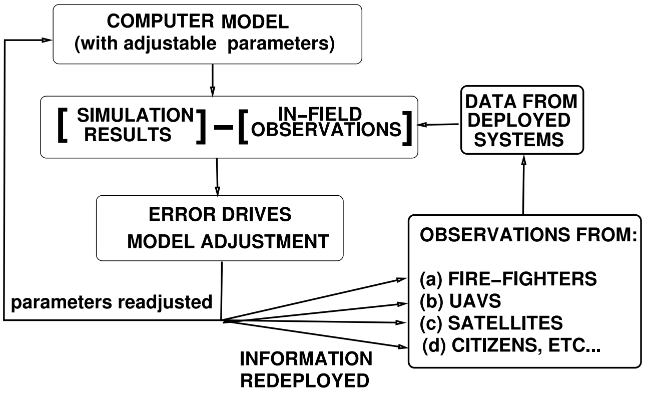

Fig. 2. A general Machine-Learning Algorithm. UAVs are starting to play a central role in modern data collection (see Zohdi [6]).

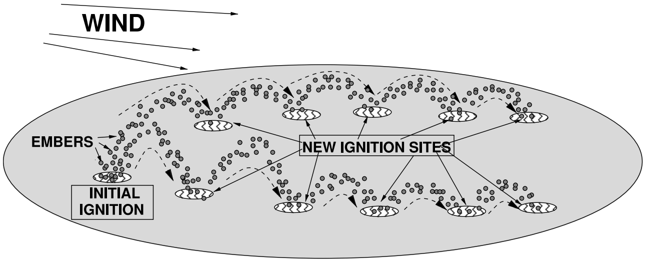

Fig. 3. Airborne propagation of embers and new ignition sites.

Generally, a signifiant portion of fire-fighting research is geared towards improving emergency response practices and the development of faster emergency response systems. In particular, while detailed large-scale computational simulations of fire growth exist, they are oftentimes too slow to be useful for field-deployed, fast-paced and mobile computing platforms in harsh environments. Accordingly, the present work focuses on rapid fire simulation by developing a framework combining three components (Figs. 2–4):

1. **COMPONENT 1:** A meshless discrete element "submodel" that tracks the trajectory of airborne hot particles/embers, subject to prevailing wind velocities and updrafts,
2. **COMPONENT 2:** A topographical "submodel" of the ambient combustible material whereby airborne embers that make contact are allowed to start secondary fires (if conditions are appropriate), combined with ground-based surface spread, burn rates for generating new embers and new updrafts due to hot air, etc., and

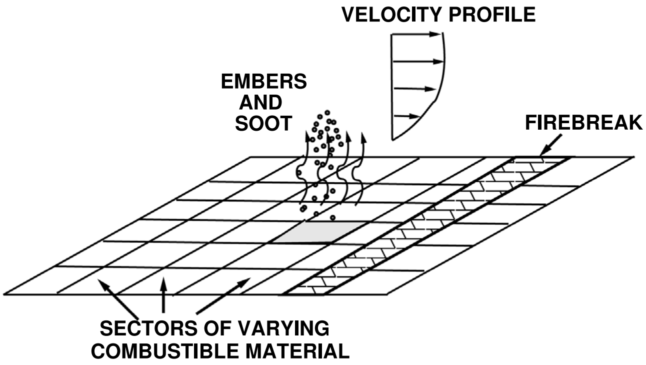

Fig. 4. Key components in the model: (1) Ember propagation (2) Combustible topology and (3) Fire Breaks.

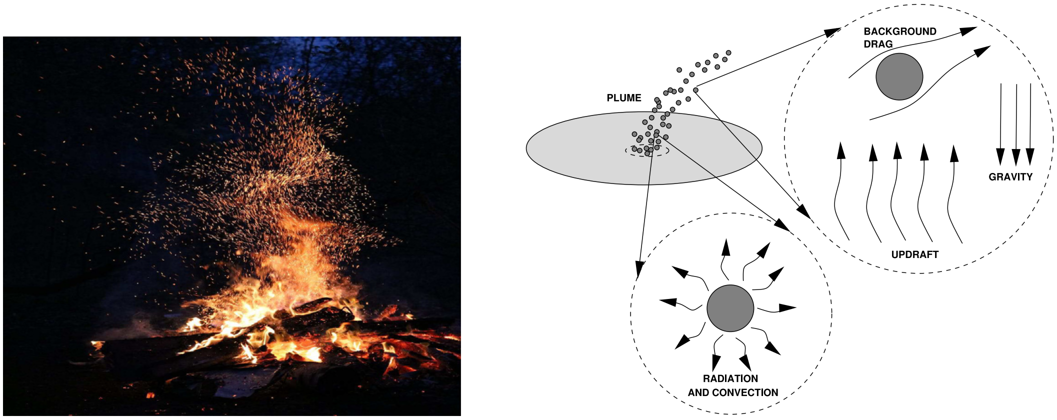

Fig. 5. Forces acting on an ember. (public domain photo from https://www.pexels.com/public-domain-images/).

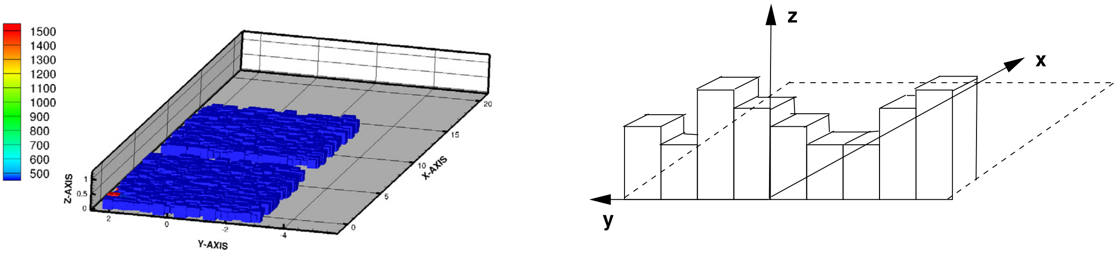

Fig. 6. Fire topography generation. Block size indicates the amount of combustible material available. The empty area between the two zones indicates no combustible materials (a firebreak).

3. **COMPONENT 3:** A Machine-Learning Algorithm to rapidly ascertain the multi-submodel system parameters that force the overall model to match observations.

The submodels chosen provide a set of components to illustrate the framework. The particular simulator developed here is based on a mesh-free discrete hot ember element flow model, which combines submodels for both ground and airborne hot-ember driven fire propagation, as well as subsequent distribution of debris/soot, which is important for air-quality assessment (Figs. 4–6). The overall framework is designed for use in *digital twin* technology, which refers to a digital replica of a physical system, where the philosophy is that updates to digital twins are made continuously in near real-time, which necessitates a rapid simulation paradigm that can easily interface with telecommunications, cameras and sensors. Ideally, such a framework should be able to integrate historical data from past model usage, that factors in its real-time digital model and optimizes its operation and maintenance of physical assets, systems and processes. The model is designed to run rapidly on laptops and hand held devices, with the guiding principle being to make it useful for first-responders *in real-time*.

Remark 1. As mentioned, wildfire modeling typically has as its objective, to determine the speed and direction of spreading of fires, utilizing fuel model-types for ambient materials, relative humidity, precipitation, etc. There is no shortage of simulation codes, but the limitations are real-time accuracy and easy practical use. The framework presented in this paper can be applied to other wildfire simulation tools, involving, for example:

- **Empirical models,** which are empirical fire spread equations, such as those published by the USDA Forest Service, for quick estimation of fundamental parameters of interest such as fire spread rate.
- **Semi-empirical models,** which are two-dimensional fire growth models, such as FARSITE, that use semi-empirical relationships for fire spread rate.
- **Physically-based models,** which combine computational fluid dynamics with wildland fire components and allow the fire to feed back into the atmosphere. These models include NCAR's Coupled Atmosphere–Wildland Fire–Environment (CAWFE) code, NCAR's, Weather Research and Forecasting (WRF) code, Los Alamos' FIRETEC simulator, NIST's Fire Dynamics Simulator (WFDS), etc.

The "high-fidelity" physically-based models discussed utilize coupled partial differential governing equations, such as:

$$\begin{aligned}
&Balance\ of\ mass: \frac{\partial \rho}{\partial t} = -\nabla_x \rho \cdot \boldsymbol{v} - \rho \nabla_x \cdot \boldsymbol{v},\\
&Balance\ of\ momentum: \rho(\frac{\partial \boldsymbol{v}}{\partial t} + (\nabla_x \boldsymbol{v}) \cdot \boldsymbol{v}) = \nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f},\\
&Constitutive\ Law: \boldsymbol{\sigma} = -P\mathbf{1} + 3\kappa \tfrac{tr\boldsymbol{D}}{3}\mathbf{1} + 2\mu \boldsymbol{D}',
\end{aligned} \tag{1.1}$$

where $\rho(\boldsymbol{x})$ is the density field of the fluid, $\boldsymbol{v}(\boldsymbol{x})$ is the fluid velocity field, $\boldsymbol{\sigma}(\boldsymbol{x})$ is the fluid stress field, $\boldsymbol{D}(\boldsymbol{x})$ is the fluid velocity gradient field, $\boldsymbol{f}(\boldsymbol{x})$ is the body force field, $P(\boldsymbol{x})$ is the fluid pressure field, $\kappa(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ are fluid material property fields. Collectively, we refer to these equations as the "Navier–Stokes" equations. It is customary to specify $\boldsymbol{v}$ and P on the boundary, and to determine ρ on the boundary through the Equation of State. P is given by an Equation of State. It is important to emphasize that physically compatible boundary data must be applied, and this is not a trivial matter for compressible flow. Additionally, the first law of thermodynamics is included (along with equations for various chemical reactions), which reads as

$$\rho \dot{w} - \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} + \nabla_x \cdot \boldsymbol{q} - \rho z = 0, \tag{1.2}$$

where $w(x)$ is the stored energy in the fluid, $\boldsymbol{q}(\boldsymbol{x})$ is the heat flux field, z is the heat source field per unit mass. When dealing with multifield problems, this equation is used extensively, when coupled to the Navier–Stokes equations. Generally such models are ineffective for real-time use, which is the subject of this work. However, they are quite useful for detailed offline background analyses, where rapid response is a nonissue. Over that last two decades, mesh-free particle-based "Lagrangian" methods, such as discrete element models have been developed which are faster to compute, but do not provide as much detailed fine-scale information — this is ideal for large-scale fire simulation. Particle-based methods form the basis of this work. Furthermore, when coupled with Machine-learning algorithms, they are able to have increased accuracy by incorporating observational data during the simulation process.

2. COMPONENT 1: Particle-based fire-spotting models

Unintentional wildfires have many sources. One frequent initiation of unwanted fires is by ignition from incandescent metallic or ceramic particles from man-made causes, such as manufacturing workplaces. This is a source of growing concern in arid and semi-arid environments, such as those found in California, where population growth has been unabated. Oftentimes, such particles are generated by manufacturing processes involving cutting, grinding, sanding, welding, etc., as well as metal-to-metal contact from construction sites, worn-out brakes, vehicles exhaust systems, ballistic impacts, explosions, pyrotechnics, etc. Also, situations involving clashing power lines in high winds, leading to hot charged material being ejected, are of concern. The generation of such incandescent particles is well-documented (Wingerden et al. [7] and Fernandez-Pello [8]). According to the National Interagency Fire Center, from 2006–2015, the average number of wildfires in the United States was 71,594, with nearly seven million acres burnt annually (NIFC report [9]). Depending on the thermal state of the particles and the material on which they land, they can be a source of ignition. There have been many fires that have been attributed to hot metal particles and sparks. Published data (Prestemon et al. [10], Ahrens [11] and Ramljak et al. [12]) indicates that powerlines, machinery and vehicles cause approximately 28,000 natural fuel fires annually in the United States (Aherns [11], NFPA [13] and USFA [14]). These processes depend on many factors, including the thermal state and trajectories of the particles. The scientific significance of understanding this phenomena is that it can potentially save lives and reduce fire damage, for example by ascertaining clearance distances along highways and railroads needed to reduce the likelihood of spot fire initiation. This type of information can help construct safety regulations.

The subject of spot ignition by metal particles involves a number of parameters. Early studies on the subject were primarily experimental (Pleasance and Hart [15], Stokes [16] and Rowntree and Stokes [17]). Fernandez-Pello and co-workers (Hadden et al. [18] and Urban et al. [19]) have investigated the problem in order to better understand the physical/chemical mechanisms controlling the spot fire ignition process (Fernandez-Pello [8]). Experimental tests are, however, quite expensive and time consuming. For example, the usual experimental approach is to heat a particle of a metal usually encountered in spot fires (typically, steel, aluminum, copper, brass), which is then dropped onto a fuel bed of interest, in order to ascertain if ignition will occur. The more complex test problem of the interaction with wind, airborne cooling, etc, is experimentally daunting — *thus motivating the development of modeling and simulation tools*. On that specific topic, we refer the reader to works dedicated to correlations or CFD simulations found in Baum and McCaffrey [20] and Tarifa [21]. The launching of embers by ground fires has been investigated by many researchers, with pioneering work conducted by Tarifa et al. [21,22], who experimentally determined drag and burning rates of spheres, cylinders and plates of various woods, and then ascertained the maximum fire spread range. This was followed by a large number of theoretical and experimental studies of ember transport (Sardoy et al. [23], Lee et al. [24] and Koo et al. [25], Pleasance and Hart [15], Tse and Fernandez-Pello [26], Mills and Hang [27], Tarifa et al. [22] and Rallis and Mangaya [28]).

In this section, the "COMPONENT 1" submodel for fire-spotting is introduced, following Zohdi [29], to characterize the progressive time-evolution of the dynamics of a set of released hot particles (Fig. 5). Thereafter, extensions are discussed, leading to the submodel associated with "COMPONENT 2". In this section, we initially restrict ourselves to computing the trajectory of hot particles. The subsequent phenomena of burning embers and ground-based surface materials is the theme of later sections (COMPONENT 2).

2.1. Hot particle propagation models

The equation of motion for the ith fragment in the system is

$$m_i \dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{tot} = \boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag} \tag{2.1}$$

where for the drag, we will employ a general phenomenological model

$$\boldsymbol{\Psi}_i^{drag} = \frac{1}{2}\rho_a c_d \|\boldsymbol{v}^f - \boldsymbol{v}_i\|(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i, \tag{2.2}$$

where c_d is the drag coefficient, A_i is the reference area, which for a sphere is $A_i = \pi R_i^2$, the surrounding fluid density is ρ_a (in the case of interest, air) and $\boldsymbol{v}^f$ is the surrounding velocity.

The inner-product of the drag force with the relative velocity of the fragment to the surrounding environment provides the drag-heating rate, thus

$$m_i C_i \dot{\theta} = \mathcal{H}_i^{drag} = \gamma\, \boldsymbol{\Psi}_i^{drag} \cdot (\boldsymbol{v}^f - \boldsymbol{v}_i) = \gamma \frac{1}{2}\rho_a c_d \|\boldsymbol{v}^f - \boldsymbol{v}_i\|(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i \cdot (\boldsymbol{v}^f - \boldsymbol{v}_i) = \gamma \frac{1}{2}\rho_a c_d \|\boldsymbol{v}^f - \boldsymbol{v}_i\|^3 A_i, \tag{2.3}$$

where $0 \leq \gamma \leq 1$ is the frictional heating efficiency. If one then includes convective and radiative cooling, this yields

$$m_i C_i \dot{\theta} = \mathcal{H}_i^{tot} = \mathcal{H}_i^{conv} + \mathcal{H}_i^{drag} + \mathcal{H}_i^{rad} = h_i(\theta_e - \theta_i)A_i^s + \gamma \frac{1}{2}\rho_a c_d \|\boldsymbol{v}^f - \boldsymbol{v}_i\|^3 A_i + \epsilon_i \mathcal{B}(\theta_e^4 - \theta_i^4)A_i^s, \tag{2.4}$$

where $0 \leq \epsilon_i \leq 1$ is the radiative efficiency, $\mathcal{B} = 5.670374 \times 10^{-8}\,\mathrm{Wm^{-2}K^{-4}}$ is the Stefan–Boltzmann constant and $A_i^s = 4\pi R_i^2$ is the radiative surface area. The effect of drag heating can be important for extremely fast moving particles in viscous environments and is less important in lower velocity scenarios.

2.2. Characterization of the drag coefficient

To accurately account for the effects of drag, we employ an empirical drag coefficient that varies with Reynolds number (Chow [30]):

- For $0 < Re \leq 1$, $c_d = \frac{24}{Re}$,
- For $1 < Re \leq 400$, $c_d = \frac{24}{Re^{0.646}}$,
- For $400 < Re \leq 3 \times 10^5$, $c_d = 0.5$,
- For $3 \times 10^5 < Re \leq 2 \times 10^6$, $c_d = 0.000366Re^{0.4275}$,
- For $2 \times 10^6 < Re < \infty$, $c_d = 0.18$,

where the local Reynolds number for a fragment is $Re \overset{\text{def}}{=} \frac{2R\rho_a\|\boldsymbol{v}^f - \boldsymbol{v}_i\|}{\mu_f}$ and μ_f is the fluid viscosity.[1]

Remark 2. This piecewise drag law is essentially a curve-fit of extensive data from Schlichting [31]. In the low Reynolds number limit, a Stokesian model is most appropriate — this is captured by the piecewise drag law since

$$\boldsymbol{\Psi}_i^{drag,Stokesian} = c(\boldsymbol{v}^f - \boldsymbol{v}_i) = \mu_f 6\pi R_i(\boldsymbol{v}^f - \boldsymbol{v}_i) = \frac{1}{2}\rho_a c_d \|\boldsymbol{v}^f - \boldsymbol{v}_i\|(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i = \boldsymbol{\Psi}_i^{drag} \tag{2.5}$$

when $0 < Re \leq 1$, $c_d = \frac{24}{Re}$.

2.3. More accurate characterization of the heat transfer coefficient

To extend the constant convection coefficient to more realistic regimes, we consider the well-known heat transfer relation relating the Nusselt number, which is the ratio between the heat transfer of convection to heat transfer of conduction

$$Nu \overset{\text{def}}{=} \frac{hL}{K} \Rightarrow h = \frac{NuK}{L}, \tag{2.6}$$

where h is the convection coefficient, $L = 2R$ is the length scale and K is the fluid conductivity, to the Reynolds number

$$Re \overset{\text{def}}{=} \frac{\rho 2R\|\boldsymbol{v}^f - \boldsymbol{v}_i\|}{\mu}, \tag{2.7}$$

and Prandtl number

$$Pr \overset{\text{def}}{=} \frac{C\mu}{K}, \tag{2.8}$$

which reads as (Whitaker [32])

$$Nu \approx 2 + (0.4Re^{1/2} + 0.06Re^{2/3})Pr^{0.4}(\frac{\mu}{\mu_s})^{0.25}, \tag{2.9}$$

where C is the heat capacity of the fluid, μ is the surrounding fluid viscosity and μ_s is the viscosity of the fluid at the surface. In the analysis to follow, we assume $\mu_s \approx \mu$.

2.4. System discretization

The governing equation, with variable drag and gravity included is

$$m_i\dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{drag} + \boldsymbol{\Psi}_i^{grav} = \frac{1}{2}\rho_a c_d\|\boldsymbol{v}^f - \boldsymbol{v}_i\|(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i + m_i\boldsymbol{g}, \tag{2.10}$$

[1] The viscosity coefficient for air is $\mu_f = 0.000018$ Pa/s.

which we must integrate the governing equations numerically, for example using an explicit Euler scheme:

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \frac{1}{m_i}\int_t^{t+\Delta t} (\boldsymbol{\Psi}_i^{drag} + \boldsymbol{\Psi}_i^{grav})\,dt \approx \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\left(\boldsymbol{\Psi}_i^{drag}(t) + \boldsymbol{\Psi}_i^{grav}(t)\right). \tag{2.11}$$

For the temperature

$$\begin{aligned}\theta_i(t+\Delta t) &= \theta_i(t) + \frac{1}{m_i C_i}\int_t^{t+\Delta t}\left(\mathcal{H}_i^{drag} + \mathcal{H}_i^{conv} + \mathcal{H}_i^{rad}\right)dt \\ &\approx \theta_i(t) + \frac{\Delta t}{m_i C_i}\left(\mathcal{H}_i^{drag}(t) + \mathcal{H}_i^{conv}(t) + \mathcal{H}_i^{rad}(t)\right).\end{aligned} \tag{2.12}$$

The solution procedure is straightforward. At a given time step:

- STEP 1: Update the velocities

  $$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\left(\boldsymbol{\Psi}_i^{drag}(t) + \boldsymbol{\Psi}_i^{grav}(t)\right) \tag{2.13}$$

  and positions

  $$\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}(t) + \boldsymbol{v}_i(t)\Delta t. \tag{2.14}$$

- STEP 2: Update the temperature from the energy balance yielding

  $$\theta_i(t+\Delta t) = \theta_i(t) + \frac{\Delta t}{m_i C_i}\left(\mathcal{H}_i^{drag} + \mathcal{H}_i^{conv} + \mathcal{H}_i^{rad}\right). \tag{2.15}$$

- STEP 3: Repeat STEPS 1–2 for each particle.
- STEP 4: Go to the next time-step.

Numerical simulations using this algorithm for scenarios involving explosions can be found in Zohdi [29] and Zohdi and Cabalo [33]. In Zohdi and Cabalo [33], a systematic study of an explosive device was considered combining the results from a set of full scale field experiments with high explosives and ballistic gelatin and reduced order fragment tracking models, similar to those presented in this section, neglecting the interaction between the shock wave and the packed fragments and any chemical aspects.[2] There were extremely close matches between experiments and numerics, indicating that it is likely that a drag-based particle noninteraction model is appropriate.[3]

We emphasize that there are different ways for hot particles to be generated. One way is by power line (generally aluminum or copper) interactions in high-winds (referred to as buffeting or galloping) which arc or clash (Pleasance and Hart [15], Russell et al. [41] and Blackburn [42]). In such situations, metal fragments may be produced and ejected from the arcing location (Pagni [43], Gilbert [44], Maraghides and Mell [45] and Ramljak et al. [12] and Pleasance and Hart [15]). We now extend the model (COMPONENT 2) to include secondary fires due to a combustible terrain.

3. COMPONENT 2: Submodel extension to propagation of "secondary" fires and ember flow

To enhance the model in COMPONENT 1, a topographical “submodel” of combustible (fuel) material (and possible firebreaks) is generated (explained shortly). Across the topography, embers that make contact are allowed to start secondary fires, if conditions are appropriate, combined with a burn rate “submodel” for generating new embers, new updrafts due to hot air, etc. Accordingly, we consider the additional components for “COMPONENT 1” model now.

3.1. Topography

We now consider a combustible terrain of varying combustability. This is achieved by assigning the topography a varying combustibility index, $0 \leq b(\boldsymbol{x}) \leq 1$, which is the initial amount of burnable mass at that location, shown here by blocks of varying sizes, placed on a partitioning of the surface topography (Fig. 6). Connecting this feature

[2] For shock analyses, see, for example, Hoover and Hoover [34], Gregoire et al. [35], Kudryashova et al. [36], Cabalo et al. [37,38].

[3] One conclusion from these experiments was that aerosols generated from a blast containing toxic materials cannot be assumed to be inactivated by the blast itself, which is consistent with findings of Eshkol and Katz [39], and Kanemitsu [40], where Hepatitis B from a suicide bomber was transmitted to survivors of the blast.

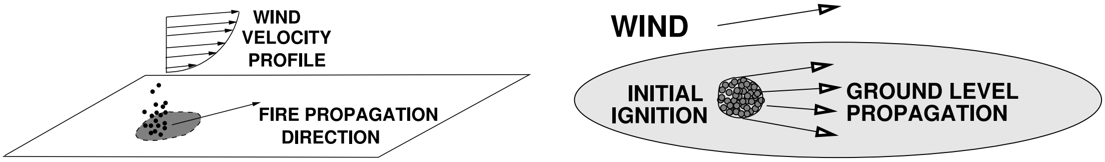

Fig. 7. The propagation is skewed to be downwind. In addition to airborne propagation, there is ground-level propagation that is influenced by ambient wind velocities.

to COMPONENT 1 is straightforward. At a given time step:

- STEP 1: Update the velocities and positions of embers.
- STEP 2: Update the temperature from the energy balance.
- STEP 3: Repeat STEPS 1–2 for each particle.
- STEP 4: Check all combustion zones for possible ignition (details shortly).
- STEP 6: For combustion zones, compute mass burned and new particle generation (details shortly).
- STEP 7: Go to the next time-step.

Remark 3. In this submodel, the propagation is skewed to be downwind. In addition to airborne propagation, there is ground-level propagation that is influenced by ambient wind velocities (Fig. 7). We now discuss this in more depth.

3.2. Submodels for ember flow

As before for the fragments, similarly for the particulate embers, the equation of motion for the ith ember in the system is, however, with variable mass:

$$\frac{d}{dt}(m_i\boldsymbol{v}_i) = \dot{m}_i\boldsymbol{v}_i + m_i\dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{tot} = \boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag} + \boldsymbol{\Psi}_i^{up}, \tag{3.1}$$

where $\boldsymbol{\Psi}_i^{up}$ is updraft force. For the thermodynamics we have

$$\frac{d}{dt}(m_iC_i(\theta_i-\theta_o)) = \dot{m}_iC_i(\theta_i-\theta_o) + m_i\dot{C}_i(\theta_i-\theta_o) + m_iC_i\dot{\theta}_i = \mathcal{H}_i^{tot} = \mathcal{H}_i^{rad} + \mathcal{H}_i^{conv} + \mathcal{H}_i^{drag}. \tag{3.2}$$

3.3. Further augmentation to the ember-flow model

The mass of burning particles and ground fuels involve rate equations. For illustration purposes, we employ simple ones which involve burning/no burning cut-offs.

3.3.1. Mass of a burning particle

The mass of a burning particle involves a rate equation:

$$\dot{m}_i = \xi_i \underbrace{f(\theta_i,\theta_b)}_{cut-off}, \tag{3.3}$$

where ξ_i is a burning rate parameter for particle i and the cut-off function f is given by:

$$If\ \theta_i > \theta_b \Rightarrow f(\theta_i,\theta_b) = 1 \tag{3.4}$$

and

$$If\ \theta_i \leq \theta_b \Rightarrow f(\theta_i,\theta_b) = 0. \tag{3.5}$$

3.3.2. The updraft parameter

As the particle burns, there is also a localized updraft, which we can model by an updraft parameter that is activated if the particle is burning:

$$\boldsymbol{\Psi}_i^{up} = m_i\eta_i f(\theta_i,\theta_b), \tag{3.6}$$

where η_i is an updraft parameter representing the amount of lift due to hot gases from the burning particle.

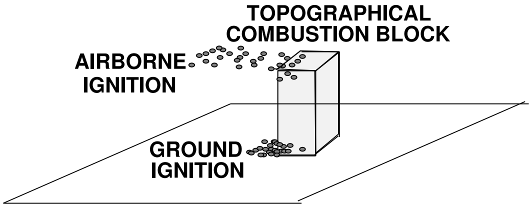

Fig. 8. Ground-based and airborne ignition.

3.4. The topographical “block” burning

The ground topography can burn in three ways: (1) ground spread, (2) airborne embers or (3) a combination of ground spread and airborne embers (Fig. 8). For ground-based ignition the following condition must hold

$$\boldsymbol{v}^e \cdot \boldsymbol{n}_{I\to J} \geq 0, \tag{3.7}$$

for ground-level “lower” burning to spread to a neighboring block/cell. For airborne ignition the following condition must hold for the i_{th} burning particle and J_{th} cell/block:

$$|x_i(t) - x_J| \leq tol \ and \ |y_i(t) - y_J| \leq tol \tag{3.8}$$

and it must be below a certain height,

$$z_i(t) \leq z_J. \tag{3.9}$$

The ground block/cell mass burning is given by

$$\dot{M}_I = \beta_I \zeta_I \tag{3.10}$$

where β_I is the rate of burning, and ζ_I is an “evolutionary” cut-off function that evolves from 0 to 1, governed by (until the mass burned is exhausted, determined by the starting mass, which is the “combustibility index” b, introduced earlier):

- If ground ignition has been initiated

  $$\dot{\zeta}_I = \lambda_g, \tag{3.11}$$

  where λ_g is a rate parameter for ground-initiated burning.
- If air ignition has been initiated

  $$\dot{\zeta}_I = \lambda_a \tag{3.12}$$

  where λ_a is a rate parameter for airborne-initiated burning.
- If ground and air ignition have been initiated

  $$\dot{\zeta}_I = \lambda_g + \lambda_a. \tag{3.13}$$

- If neither ground and air ignition have been initiated

  $$\dot{\zeta}_I = 0. \tag{3.14}$$

- The cut-off has a maximum of $\zeta_I = 1$

  $$If \ \zeta_I \geq 1 \Rightarrow \zeta_I = 1. \tag{3.15}$$

3.5. System discretization

The governing equation, with variable drag and gravity included is

$$\frac{d}{dt}(m_i \boldsymbol{v}_i) = \dot{m}_i \boldsymbol{v}_i + m_i \dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{tot} = \boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag} + \boldsymbol{\Psi}_i^{up}, \tag{3.16}$$

which we integrate numerically, using an explicit Euler scheme:

$$\boldsymbol{v}_i(t+\Delta t)=\boldsymbol{v}_i(t)+\int_t^{t+\Delta t}\frac{1}{m_i}\left(\boldsymbol{\Psi}_i^{drag}(t)+\boldsymbol{\Psi}_i^{grav}(t)+\boldsymbol{\Psi}_i^{up}(t)-\dot{m}_i\boldsymbol{v}_i\right)dt. \tag{3.17}$$

Similarly, for the temperature

$$\theta_i(t+\Delta t)=\theta_i(t)+\int_t^{t+\Delta t}\frac{1}{m_iC_i}\left(\mathcal{H}_i^{drag}+\mathcal{H}_i^{conv}+\mathcal{H}_i^{rad}-\dot{m}_iC_i(\theta_i-\theta_o)-m_i\dot{C}_i(\theta_i-\theta_o)\right)dt. \tag{3.18}$$

3.6. Algorithmic combination of COMPONENT 1 and COMPONENT 2 submodels

After assigning the topography a combustability index, $0\leq b_I(\boldsymbol{x})\leq 1$ (shown in Fig. 6), the solution procedure is straightforward. At a given time step:

- STEP 1: Update the velocities

$$\boldsymbol{v}_i(t+\Delta t)=\boldsymbol{v}_i(t)+\int_t^{t+\Delta t}\frac{1}{m_i}\left(\boldsymbol{\Psi}_i^{drag}(t)+\boldsymbol{\Psi}_i^{grav}(t)+\boldsymbol{\Psi}_i^{up}(t)-\dot{m}_i\boldsymbol{v}_i\right)dt. \tag{3.19}$$

  and positions

$$\boldsymbol{r}_i(t+\Delta t)=\boldsymbol{r}(t)+\boldsymbol{v}_i(t)\Delta t. \tag{3.20}$$

- STEP 2: Update the temperature from the energy balance yielding

$$\theta_i(t+\Delta t)=\theta_i(t)+\int_t^{t+\Delta t}\frac{1}{m_iC_i}\left(\mathcal{H}_i^{drag}+\mathcal{H}_i^{conv}+\mathcal{H}_i^{rad}-\dot{m}_iC_i(\theta_i-\theta_o)-m_i\dot{C}_i(\theta_i-\theta_o)\right)dt. \tag{3.21}$$

- STEP 3: Repeat STEPS 1–2 for each particle.
- STEP 4: Check all combustion zones for possible ignition.
- STEP 6: For combustion zones, compute mass burned and new particle generation.
- STEP 7: Go to the next time-step.

At every time step, each burning block is checked. If there is mass still available, then a representative particle is released. It is important to note that while the mass of the block is dictated by the burning block rate equation, the entire mass is not converted into a large number of particles — in order to make the computations manageable. In this model, every 0.05 s new particles were generated. As an example, the domain size was $-0.5\leq x\leq 15$ and $-3\leq y\leq 2.5$. 60 blocks were generated in the $x\text{-}direction$ and 20 in the $y\text{-}direction$. Each block was given a random combustability index ($0\leq b(\boldsymbol{x})\leq 1$), which was the initial amount of burnable mass, shown in Fig. 9 by blocks of varying sizes. A complete simulation example is shown in Fig. 6. The fire was started at $(x, y)=(0, 2.5)$. With the chosen parameters, the results are shown in Figs. 9 and 10. The complete set of parameters are shown in the Appendix.

There are clearly many parameters, some of which are phenomenological. Modern real engineering solutions utilize a multitude of engineering tools

- Remote sensing, mapping, etc.
- Wireless communication, sensor technologies, data analytics, etc.
- Computer vision, infrared technologies, etc.
- Experiments: combustion, toxicology, etc. and
- Ground to air technologies, UAV's, UGV's, Robotics, etc.

The decisions needed to deploy and operate these tools optimally requires guidance from real-time modeling and simulation that the preceding model can provide, *if the parameters are known.* This requires that the model be adapted quickly, with input from surrounding observations. Models which attempt to include such data are classified as Data Assimilation Models, and the process of including data is a type of Machine-Learning-Algorithm. This typically requires an overlay of a "wrapper" around the simulator (such as the preceding model) to determine with parameter sets match the externally-observed data as closely as possible. This is essentially an inverse problem. Following Zohdi [46,47], next we formulate the objective as a cost-function minimization problem, using Machine-Learning Algorithms and is referred to as "COMPONENT 3".

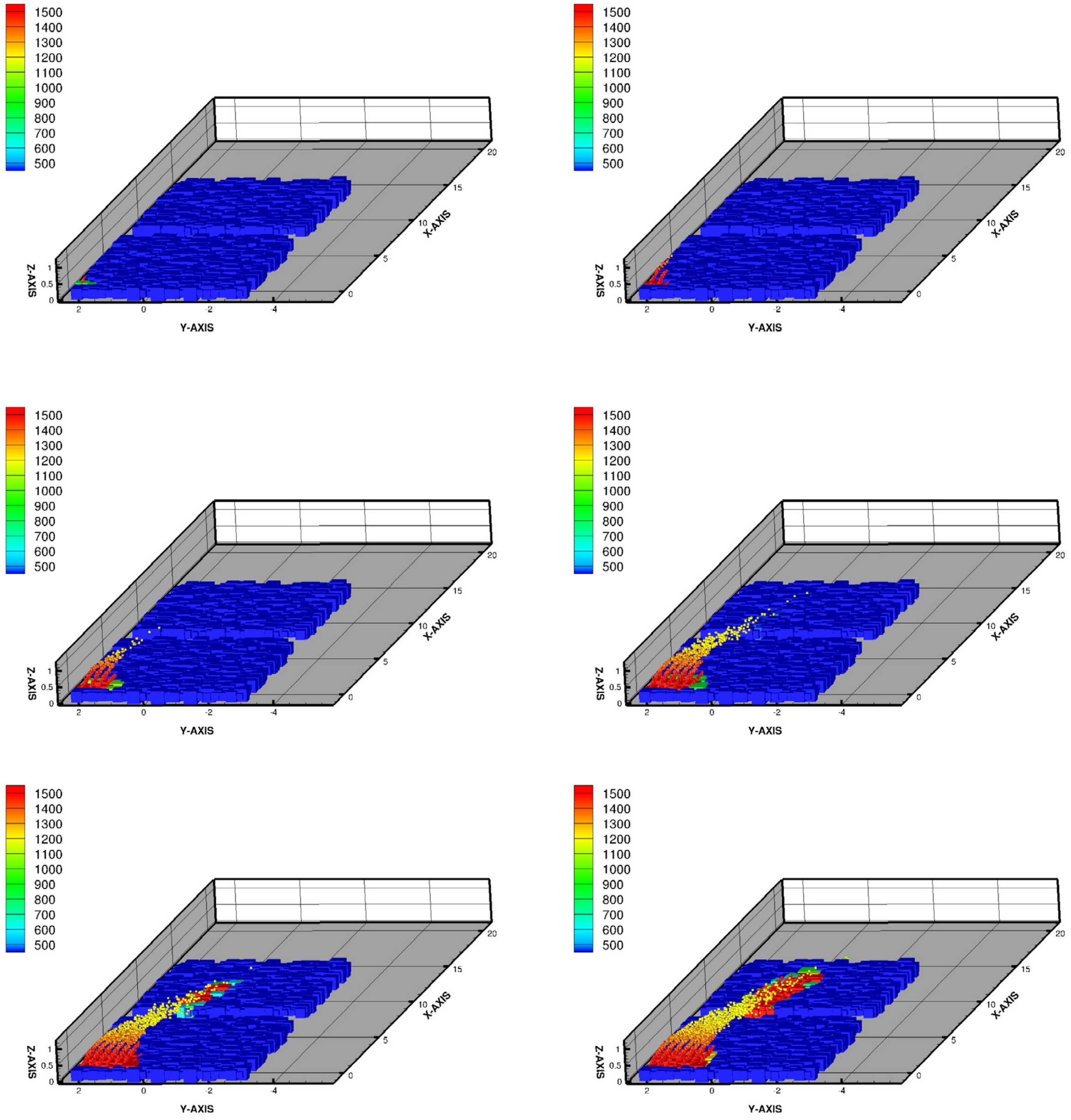

Fig. 9. From top to bottom and left to right over simulation time $T = 2.5$ s. Sequences of fire frames at intervals of $T/12$.

Remark 4. In Zohdi [48,49], more detailed, computationally intensive models were developed to characterize the motion of small-scale particles embedded in a flowing fluid where the dynamics of the particles affects the dynamics of the fluid and vice-versa. That model is significantly more complex than the suite of models that will be used in the body of the upcoming work, primarily because of the incorporation of the Navier–Stokes equations. In such a framework, a fully implicit Finite-Difference discretization of the Navier–Stokes equations is used for the fluid and a direct particle-dynamics discretization is performed for the particles. Because of the large computational difficulty and expense of a conforming spatial discretization needed for large numbers of embedded particles, simplifying assumptions are made for the coupling, based on semi-analytical computation of drag-coefficients, which allows for the use of coarser meshes. Even after these simplifications, the particle–fluid system is strongly-coupled. The approach taken in that work was to construct a sub-model for each primary physical process. In order to resolve

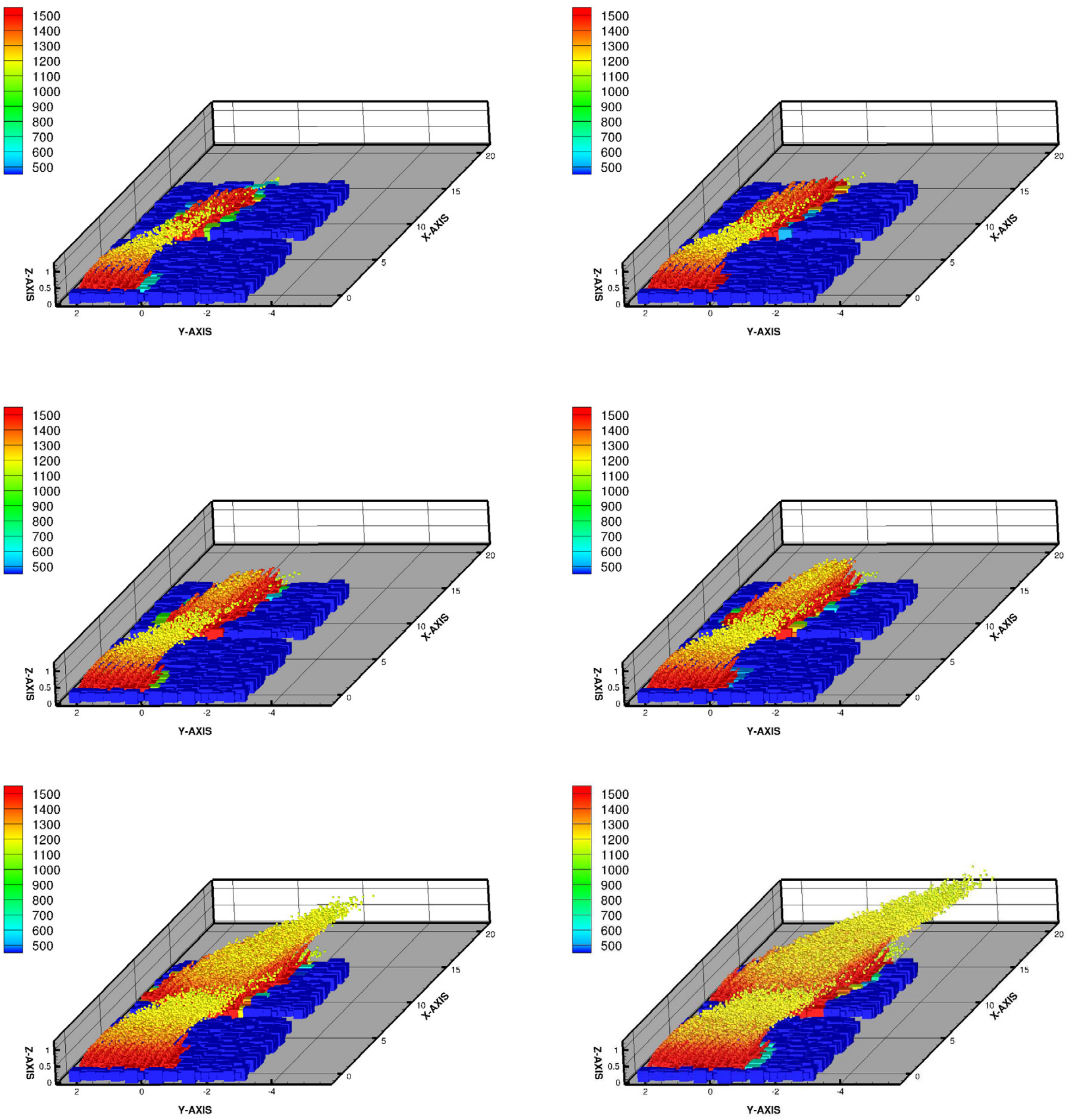

Fig. 10. From top to bottom and left to right over simulation time $T = 2.5$ s. Sequences of fire frames of simulation time $T/12$.

the coupling, a recursive staggering scheme was constructed, which was built on works found in Zohdi [46,47,50]. The procedure was as follows (at a given time increment): (1) each submodel equation (fluid or particle-system) is solved individually, "freezing" the other (coupled) fields in the system, allowing only the primary field to be active, (2) after the solution of each submodel, the associated field variable was updated, and the next submodel was solved and (3) the process is then repeated, until convergence. The time-steps are adjusted to control the rates of convergence, which is dictated by changes in the overall physics. Specifically, the approach was a staggered implicit time-stepping scheme, with an internal recursion that automatically adapted the time-step sizes to control the rates of convergence within a time-step. If the process did not converge (below an error tolerance) within a preset number of iterations, the time-step was adapted (reduced) by utilizing an estimate of the spectral radius of the coupled system. The developed approach can be incorporated within any standard computational fluid mechanics code based on finite difference, finite element, finite volume or discrete/particle element discretization (see Labra

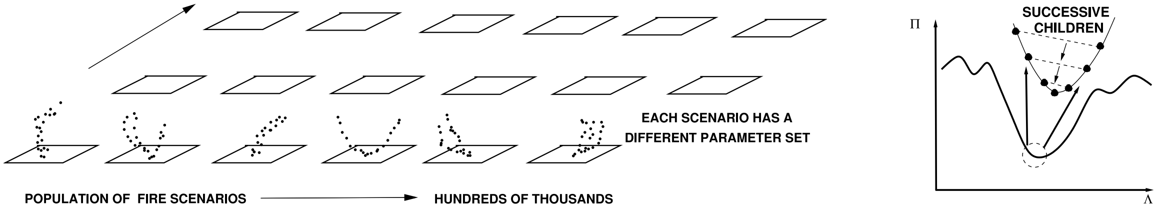

Fig. 11. LEFT: Fire simulation populations: the key is an algorithm that can test thousands of scenarios and then to select the one that best fits all available data. RIGHT: The basic action of a MLA/GA-Machine Learning Algorithm/Genetic Algorithm.

and Onate [51], Onate et al. [52,53], Rojek et al. [54] and Avci and Wriggers [55]). However, while useful in many industrial applications where high precision is required, the use of such a model for the coarser applications of interest in this work is unwarranted.

4. COMPONENT 3: Adaptive submodels and Machine-Learning Algorithms for parameter estimation

4.1. Basic Machine-Learning for parameter estimation

The speed at which this type of simulation can be completed allows one to answer the inverse question of what the combination of parameters should be for an experimentally-observed result (Figs. 11 and 12). In order to cast the objective mathematically, we set the problem up as a Machine Learning Algorithm (MLA); specifically a Genetic Algorithm (GA) variant. Following Zohdi [46,47], we formulate the objective as a cost function minimization problem whereby simulation parameters are adjusted to match observations as closely as possible:

$$\Pi(\lambda_1,\ldots,\lambda_{14}) = w_1|(1-\frac{A_b}{A*})| + w_2|(1-\frac{T_b}{T^*})|, \tag{4.1}$$

where A_b represents the percent area burned, target percent area A^* burned, $0 \leq T_b \leq 1$ represents the time for this event to occur and T^* represents target simulation time. Also, we add a penalty that if all of the target area is not burned, that $T^* = P \gg T_b$. This forces the function only to register a differentiating value of T^* if the event burns everything in an extremely small time — i.e. a catastrophe:

- Specifying the observed topographical areas that are burning for a given initial condition — for example for a controlled burn.
- Determining topographical areas that are burning for a given initial condition — for a trial set of parameters.
- Determining the difference between the observed and generated results.
- Systematically minimize Eq. (4.1), $min_\Lambda \Pi$, by varying the design parameters: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda^i_1, \Lambda^i_2, \Lambda^i_3, \Lambda^i_4, \ldots, \Lambda^i_N\} \stackrel{\text{def}}{=} \{thermal, flow, ignition\ parameters, \ldots\}$.
- The system parameter search is conducted within the constrained ranges of $\Lambda_1^{(-)} \leq \Lambda_1 \leq \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \leq \Lambda_2 \leq \Lambda_2^{(+)}$ and $\Lambda_3^{(-)} \leq \Lambda_3 \leq \Lambda_3^{(+)}$, etc. These upper and lower limits would, in general, be dictated by what is physically feasible.

4.2. System parameter search: Machine Learning Algorithm (MLA)

Here we follow Zohdi [46,47,50] in order to minimize Eq. (4.1), which we will refer to as a "cost function". Cost functions such as Eq. (4.1) 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 subset of MLA's are so-called Genetic Algorithms (GA's). Typically, one will use a GA first in order to isolate multiple local minima, and then use a gradient based algorithm in these locally convex regions or reset the GA to concentrate its search over these more constrained regions. GA's are typically the simplest scheme to start the analysis, and one can, of course, use more sophisticated methods if warranted. For a review of GA's, see the pioneering work of John Holland (Holland [56]), as well as Goldberg [57], Davis [58], Onwubiko [59], Lagaros et al. [60], Papadrakakis et al. [61–64] and Goldberg and Deb [65].

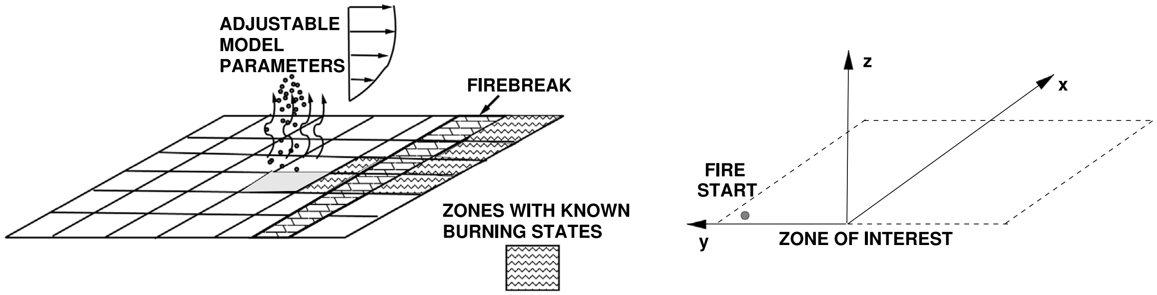

Fig. 12. LEFT: Machine Learning Algorithm model problem of a fire with known states supplied by observations. RIGHT: Zone of interest for the numerical example.

4.2.1. *Generalities*

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

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

4.2.2. *Specifics*

Following Zohdi [46,47,50] the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i, (i = 1, 2, 3, \ldots, S) : \boldsymbol{\Lambda}^i \overset{\text{def}}{=} \{\Lambda^i_1, \Lambda^i_2, \Lambda^i_3, \Lambda^i_4, \ldots, \Lambda^i_N\}$
- **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)
- **STEP 4:** Mate nearest pairs and produce two offspring, (i = 1, …, S)
$\boldsymbol{\lambda}^i \overset{\text{def}}{=} \Phi^{(I)}\boldsymbol{\Lambda}^i + (1 - \Phi^{(I)})\boldsymbol{\Lambda}^{i+1}$, $\boldsymbol{\lambda}^{i+1} \overset{\text{def}}{=} \Phi^{(II)}\boldsymbol{\Lambda}^i + (1 - \Phi^{(II)})\boldsymbol{\Lambda}^{i+1}$
- **STEP 5:** Eliminate the bottom $M < S$ strings and keep top $K < N$ parents and top K offspring (K offspring+K parents+$M = S$)
- **STEP 6:** Repeat STEPS 1–6 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **NOTE:** $\Phi^{(I)}$ and $\Phi^{(II)}$ are random numbers, such that $0 \leq \Phi^{(I)} \leq 1$, $0 \leq \Phi^{(II)} \leq 1$, which are different for each component of each genetic string
- **OPTION:** Rescale and restart search around best performing parameter set every few generations

Remark 5. If one selects the mating parameter Φ 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.

Remark 6. 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 re-evaluated — making the algorithm less computationally less expensive, since these parameter sets do not have to be reevaluated (or ranked) in the next generation. Numerous studies of the author have shown that advantages of parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we remark that this algorithm is easily parallelizable.

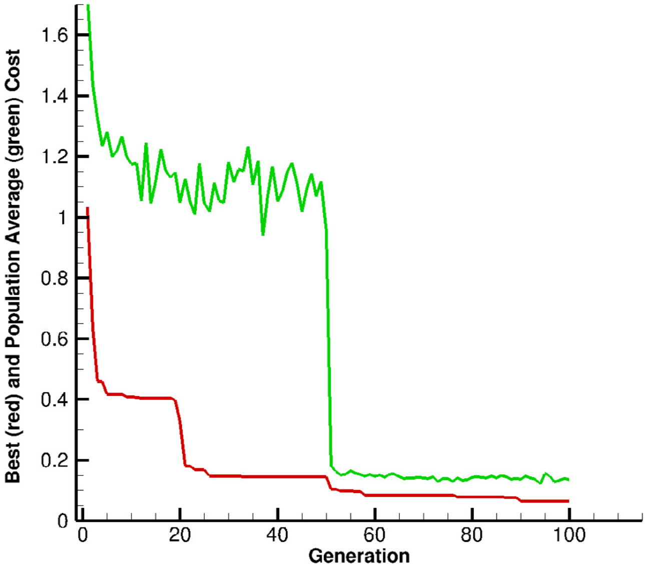

Fig. 13. The reduction of the cost function for the 14 parameter set. This cost function represents the percentage of unburned material left in the zone of interest. In other words, the system is being driven to the parameters generating the worst case scenario. THE OBJECTIVE FUNCTION (Π). Machine learning output, generation after generation. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of genes (designs, in *green*). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark 7. 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 methods can be found in the texts of Luenberger [66] and Gill, Murray and Wright [67].

4.3. Model problem

We applied the MLA algorithm, with the following design objectives (Fig. 12)

- Target area burned: $A^* = 0.5$,
- Time to burn: $T^* = 2.5$ s and
- Design weights: w_1 and $w_2 = 1$.

Fig. 13 shows the reduction of the cost function for the 14 parameter set. This cost function Π represents the percentage of unburned material left in the zone of interest. In other words, the system is being driven to the parameters generating the worst case scenario. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of the genes (designs, in *green*). The design parameters $\boldsymbol{\Lambda} = \{\Lambda_1, \Lambda_2 ... \Lambda_N\}$ and their respective search intervals are shown in the Appendix and optimized over the search intervals (14 variables): $\boldsymbol{\Lambda}_i^- \leq \boldsymbol{\Lambda}_i <= \boldsymbol{\Lambda}_i^+$, $i = 1, 2, \ldots, 14$. We used the following MLA settings:

- Number of design variables: 14,
- Population size per generation: 20,
- Number of parents to keep in each generation: 4,
- Number of children created in each generation: 4,
- Number of completely new genes created in each generation: 12

Table 1
The top system parameter performers (Λ_1–Λ_{10}).

Λ_1	Λ_2	Λ_3	Λ_4	Λ_5	Λ_6	Λ_7	Λ_8	Λ_9	Λ_{10}	Π
2.130	4.503	0.6555	0.6186	4.9646	0.00004771	4.7647	−2.8567	0.5273	0.05074	0.06525

Table 2
The top system parameter performers (Λ_{11}–Λ_{14}).

Λ_{11}	Λ_{12}	Λ_{13}	Λ_{14}	Π
0.61940293	0.23350003	6.54833116	7.37862970	0.06525

- Number of generations for readaptation around a new search interval: 10 and
- Number of generations: 100.

The algorithm was automatically reset every *10 generations*. The entire 100 generation simulation, with 20 genes per evaluation (2000 total designs) took on the order of 3–4 min on a laptop, *making it ideal as a design tool.* Fig. 13 (average top four genes performance and top gene performance) and Tables 1–2 (values of the gene components) illustrate the results. The MLA/GA is able to home in on a variety of possible parameter sets, including the one corresponding to the original set of parameters that generated the observations and alternatives that achieve virtually the same results. This allows system designers more flexibility in parameter selection. We note that, for a given set of parameters, a complete simulation takes on the order of 0.1 s, thus over 36,000 parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA/GA.*

5. Summary

In this work, we focused on the development of a fast and adaptive computational framework for fire modeling that can be potentially integrated with firefighting tools. As an example of the framework, we developed a multicomponent system comprised of relatively simple submodels. The aggregate system response was complex, with system parameters identified by a Machine-Learning Algorithm. This paradigm can be extended to other types of fire models that are in current use. For example, a notable system in use for fire propagation is $FARSITE$ (https://www.firelab.org/project/farsite), which was developed over several years using empirical data to characterize fire behavior over 2D terrain and is based on Rothermel's spread model and subsequent extensions (see Finney [68] and Andrews [69] for comprehensive reviews). The empirical basis for a such model is somewhat limited and its utility for fire scenarios depends heavily on the expertise of the fire behavior analyst operating the model. Oftentimes, fuel and wind speed factors are adjusted in real time to improve fire behavior predictions — *this task is ideal for MLA's* and the framework developed in this paper. More recently, physics-based fire models like FIRETEC (https://www.frames.gov/firetec/home) have been developed at Los Alamos National Laboratory (LANL) by coupling a physics-based fire/atmosphere models and high fidelity computational fluid dynamics (CFD) to simulate 3D fire/wind/fuel/terrain interaction. However, running FIRETEC requires the use of significant computing power such as those available at national laboratories. While FIRETEC has proven to be a valuable tool for studying wildfire behavior and gaining understanding of the ways that physical processes connect wildfires to their environment, it is too computationally expensive for real time use. Thus, QUIC-FIRE (Goodrick et al. [70]) was developed at LANL to allow faster turnaround of fire modeling using a cellular automata-based approach to approximate the CFD calculations. In addition to real time decision support, QUIC-FIRE was developed with the intention of enabling ensemble-simulation-based forecasts to capture the probabilistic nature of fire events, *ultimately employing MLAs*. Other computationally-oriented codes include, the Fire Dynamics Simulation (FDS, from NIST) https://pages.nist.gov/fds-smv/ and WRF-FIRE (from NCAR) https://www.openwfm.org/wiki/WRF-Fire, etc. The framework presented in this paper has potential utility for all of the above models and systems. A key objective is to develop paradigms that provide accurate and real-time feedback to deployed firefighters, harmoniously incorporating rapid data collection from satellites, UAVs, social media, etc. A core issue for any simulation method is the ability to adapt to rapid changes in the environment, autonomously modifying tasks and relationships with other systems. Autonomous capabilities are critical for operations involving large distances, resulting communication delays and the lack of 24/7 connectivity (i.e. limited ground stations, telemetry, command networks, etc.). These scenarios make demands that are beyond human reaction/decision times. To bridge this gap, further development of robust UAVs for firefighting, including integrated multi-UAV data gathering technologies is needed. The author is actively working in this domain. For more details, we refer the reader to Zohdi [71].

Appendix. Parameters needed in the simulations

A.1. Generic simulation parameters

- The total simulation time: $T = 5$ s,
- The time step size: $\Delta t = 0.025$ s,
- The ground-ignition rate: $\lambda_g = a\Delta t = 5$ s, $a = 100$,
- The air-ignition rate: $\lambda_a = c\Delta t = 10$ s, $c = 200$,
- Ground temperature when burning, $\theta_b = 1500$ K,
- Temperature of the immediate burning surrounding, $\theta_{be} = 1500$ K,
- Regular ambient environment, $\theta_{ae} = 300$ K,
- Temperature at which an ember will ignite ground topography, $\theta_I = 1100$ K,
- Mass burning rate of a particle, m_i, $\dot{m}_i = \xi_i f(\theta_i, \theta_b)$,
- The nominal (background) wind velocity $\boldsymbol{v}_w = (0.5, -0.25, 0.5)$,
- The random initial surrounding velocity around the generated particles at source, $\delta \boldsymbol{v}_{wo} = (0.2, 0.2, 0.1)$,

$$v_x = v_{ox}(1+\delta v_x)(\frac{z}{h})^p, \quad v_y = v_{oy}(1+\delta v_y)(\frac{z}{h})^p, \quad p = 1.5, \quad v_z = v_{oz}(1+\delta v_z)e^{-d_v z}, \quad d_v = 1. \tag{A.1}$$

- Vertical thermal decay from the surface to a height above: $\theta^e(x, y, z) = \theta_{fire} e^{-d_\theta z}$, $d_\theta = 1$,
- The random initial position of the initially-generated particles at source, $\delta \boldsymbol{r}_o = (0.05, 0.05, 0.05)$, in $\boldsymbol{r} = \boldsymbol{r}_o + \delta \boldsymbol{r}$,
- Height of released particles, $z_i(t = 0) = 1$ m,
- Gravity, $\boldsymbol{g} = (0, 0, -9.81)$,
- Density of particles, $\rho = 1000$, kg/m^3
- Nominal air density, $\rho_a = 1.225$, kg/m^3
- Generation of topography: $b_I = b_o(1 \pm \delta)$,
- Generation of burning rate: $\xi_i = \eta_{io}(1 \pm \delta_i)$,
- The updraft parameter, $\eta_i = 9.0$,
- The particle mass burning rate, $\xi_i = 0.0001$,
- The block ground burn rate $\dot{m}_{bI} = \beta_I \eta_I$, $0.015 \le \beta_I \le 0.15$,
- The nominal wind x-velocity $\boldsymbol{v}_{wx} = 1.50$,
- The nominal wind y-velocity $\boldsymbol{v}_{wy} = -0.25$,
- The (z = 0) updraft velocity $\boldsymbol{v}_{uz}(z = 0) = 0.1$,
- The starting block material available to burn $0 \le b_I \le 1$,
- The radiative efficiency, $\epsilon = 0.5$.

A.2. MLA/GA design variables and their search intervals

- The ground-ignition rate: $\lambda_g = a\Delta t = 0.01$ s, $0 \le a \le 100$,
- The air-ignition rate: $\lambda_a = c\Delta t = 0.01$ s, $0 \le c \le 200$,
- Vertical thermal decay: $\theta^e(x, y, z) = \theta_{fire} e^{-d_\theta z}$, $0.1 \le d_\theta \le 1$,
- The updraft parameter, $0 \le \eta_i \le 20$,
- The mass burning rate, $0.00001 \le \xi_i \le 0.0001$,
- The nominal wind x-velocity $0 \le \boldsymbol{v}_{wx} \le 5$,
- The nominal wind y-velocity $-5 \le \boldsymbol{v}_{wy} \le 0$,
- The (z=0) updraft velocity $0.01 \le \boldsymbol{v}_{uz}(z = 0) \le 0.1$,
- The block ground burn rate $\dot{m}_{bI} = \beta_I \eta_I$, $0.015 \le \beta_I \le 0.15$,
- The starting block material available to burn $0.1 \le b_I \le 1$,
- The radiative efficiency, $0.05 \le \epsilon \le 0.5$,
- The starting x position of the firebreak, $6 \le x \le 7$,
- The ending x position of the firebreak, $7 \le x \le 8$.

References

[1] Evarts B., Fire loss in the United States during 2017. National Fire Protection Association, Fire Analysis and Research Division, 2018, https://www.nfpa.org/media/FD0144A044C84FC5BAF90C05C04890B7.ashx.

[2] M.J. Karter Jr., Fire loss in the United States during 2013. National Fire Protection Association, Fire Analysis and Research Division, 2014, Retrieved from https://www.nfpa.org/-/media/Files/News-and-Research/Fire-statistics-and-reports/US-Fire-Problem/Old-FL-LL-and-Cat/fireloss2014.ashx?la=en.

[3] T.I. Zohdi, UC Berkeley Fire Research Group, https://frg.berkeley.edu/.

[4] C. Grant, A. Hamins, N. Bryner, A. Jones, G. Koepke, Research roadmap for smart fire fighting. nist special publication 1191, 2015, Retrieved from https://www.nfpa.org/-/media/Files/News-and-Research/Resources/Research-Foundation/Current-projects/Smart-FF/SmartFirefightingReport.pdf.

[5] K. McKinzie, The future of artificial intelligence in firefighting. fire engineering, 2018, Retrieved from https://www.fireengineering.com/articles/2018/10/artificial-intelligence-firefighting.html.

[6] T.I. Zohdi, Multiple UAVs for mapping: a review of basic modeling, simulation and applications, Annu. Rev. Environ. Resour. (2018) http://dx.doi.org/10.1146/annurev-environ-102017-025912.

[7] V.K. Wingerden, I. Hesby, R. Eckhoff, Ignition of dust layers by mechanical sparks, in: Proceedings of 7th Global Congress on Process Safety, Chicago, Ill, 2011.

[8] A.C. Fernandez-Pello, Wildland fire spot ignition by sparks and firebrands, Fire Saf. J. 91 (2017) 2–10.

[9] National Interagency Fire Center. https://www.nifc.gov/fireInfo/nfn.htm.

[10] J.P. Prestemon, T.J. Hawbaker, M. Bowden, J. Carpenter, M.T. Brooks, K.L. Abt, R. Sutphen, S. Scranton, Wildfire Ignitions: A Review of the Science and Recommendations for Empirical Modeling, Technical Report, US Department of Agriculture: Forest Service, Asheville, NC, 2013.

[11] M. Ahrens, Brush, Grass and Forest Fires, National Fire Protection Association, Quincy, Massachusetts, U.S., 2013.

[12] I. Ramljak, M. Majstrovic, S. Sutloviec, Statistical Analysis of Particles of Conductor Clashing, Dubrovnic May 13–16, 2014.

[13] National Fire Protection Association, NFPA 51B: Standard for fire prevention during welding, cutting and other hot work, 2014.

[14] US Fire Administration. https://www.usfa.fema.gov/statistics/estimates/wildfire.shtm.

[15] G.E. Pleasance, J.A. Hart, An Examination of Particles from Conductors Clashing As Possible Source of Bushfire Ignition, Report FM-1, State Electricity Commission of Victoria (SEC), Research and Development Department, Victoria, Australia, 1977.

[16] A.D. Stokes, Fire ignition by copper particles of controlled size, Aust. J. Electr. Electron. Eng. 10 (1990) 188–194.

[17] G. Rowntree, A. Stokes, Fire ignition of aluminum particles of controlled size, J. Electr. Electron. Eng. (1994) 117–123.

[18] R. Hadden, S. Scott, C. Lautenberger, C.A. Fernandez-Pello, Ignition of combustible fuel beds by hot particles: an experimental and theoretical study, Fire Technol. 47 (2011) 341–355.

[19] J.L. Urban, C.D. Zak, J. Song, A.C. Fernandez-Pello, Smolder spot ignition of natural fuels by a hot metal particle, Proc. Combust. Inst. 36 (2) (2017) 3211–3218, http://dx.doi.org/10.1016/j.proci.2016.09.014, ISSN 1540-7489.

[20] H.R. Baum, B.J. McCaffrey, Fire induced flow field-theory and experiments fire safety science, in: Wakamatsu et al. (Eds.), Proceedings of the Second International Symposium, Washington DC, 1989, pp. 129–148.

[21] C.S. Tarifa, P.P. del Notario, F.G. Moreno, On the flight paths and lifetimes of burning particles of wood, Proc. Combust. Inst. 10 (1965) 1021–1037.

[22] C.S. Tarifa, P.P. Del Notario, F.G. Moreno, A.R. Villa, Transport and Combustion of FirebrandS, Final Report of Grants GF-SP-114 and GF-SP-146. Madrid, May, U.S. Department of Agriculture Forest Service, 1967.

[23] N. Sardoy, J.L. Consalvi, B. Poterie, J.C. Loraud, C.A. Fernandez-Pello, Modeling transport and combustion of firebrands from burning trees, Combust. Flame 150 (2007) 151–169.

[24] S.L. Lee, J.M. Hellman, Firebrand trajectory study using an empirical velocity dependent burning law, Combust. Flame 15 (1970) 265–274.

[25] E. Koo, P. Pagni, R. Linn, Using FIRETEC to describe firebrand behavior in wildfires, in: Fire Mater. 2007 San Francisco, CA, 2007.

[26] S.D. Tse, A.C. Fernandez-Pello, On the flight paths of metal particles and embers generated by powerlines in high winds - a potential source of wildland fires, Fire Saf. J. (2006) 333–356.

[27] F.A. Mills, X. Hang, Trajectories of sparks from arcing aluminum power cables, Fire Technol. 20 (1984) 5–14.

[28] C.J. Rallis, B.M. Mangaya, Ignition of veld grass by hot aluminum particles ejected from clashing overhead transmission lines, Fire Technol. 38 (2002) 81–92.

[29] T.I. Zohdi, Modeling the spatio-thermal fire hazard distribution of incandescent material ejecta in manufacturing, Comput. Mech. (2018) http://dx.doi.org/10.1007/s00466-018-1617-2.

[30] C.Y. Chow, An Introduction to Computational Fluid Dynamics, Wiley, New York, 1980.

[31] H. Schlichtling, Boundary-Layer Theory, seventh ed., McGraw-Hill, New York, 1979.

[32] S. Whitaker, Forced convection heat transfer correlations for flow in pipes, past flat plates, single cylinders, single spheres, and flow in packed beds and tube bundles, AIChE J. 18 (1972) 361–371.

[33] T.I. Zohdi, J. Cabalo, On the thermomechanics and footprint of fragmenting blasts, Internat. J. Engrg. Sci. 118 (2017) 28–39.

[34] W.G. Hoover, C.G. Hoover, Tensor temperature and shock-wave stability in a strong two-dimensional shock wave, Phys. Rev. E (3) 80 (2009) 011128/1-011128/6.

[35] Y. Gregoire, M.-O. Sturtzer, B. A. Khasainov, B. Veyssiere, Investigation of the behavior of solid particles dispersed by high explosive, in: Int. Annu. Conf. ICT 40th, 35/1-35/12, 2009.

[36] O.B. Kudryashova, B.I. Vorozhtsov, E.V. Muravlev, I.R. Akhmadeev, A.A. Pavlenko, S.S. Titov, Physicomathematical modeling of explosive dispersion of liquid and powders, Propellants Explos. Pyrotech. 36 (2011) 524–530.

[37] J. Cabalo, J. Schmidt, J.O.L. Wendt, A. Scheeline, Spectrometric system for characterizing drop and powder trajectories and chemistry in reactive flows, Appl. Spect. 56 (2002) 1346–1353.

[38] Cabalo J. B., J. Kesavan, D.W. Sickenberger, G. Diviacchi, C. Maldonado-Figueroa, D. McGrady, K. Stafford, Assessing the Biological Threat Posed By Suicide Bombers, Report ECBC-TR-1363, 2016.

[39] Z. Eshkol, K. Katz, Injuries from biologic material of suicide bombers, Injury 36 (2005) 271–274.

[40] K. Kanemitsu, et al., Does incineration turn infectious waste aseptic? J. Hosp. Infect. 60 (4) (2005) 304–306.

[41] B.D. Russell, C.I. Benner, J.A. Wischkaemper, Distribution feeder caused wildfires: mechanisms and prevention, Prot. Relay Eng. 43 (2012).

[42] T. Blackburn, Conductor clashing characteristics of overhead lines, in: Proceedings of Electrical Energy Conferences, 1985, p. 202.

[43] P.J. Pagni, Causes of the 20 October 1991 Oakland-hills conflagration, Fire Saf. J. 21 (1993) 331–339.

[44] M. Gilbert, California Department of Forestry and Fire Protection Investigation Report: Incident number 07- CA-MVU-10432, http://www.fire.ca.gov/fire.

[45] A. Maranghides, W. Mell, A Case Study of a Community Affected By the Witch and Guejito Fires, NIST Technical Note 1635, 2009, http://fire.nist.gov/bfrlpubs/fire09/art028.html.

[46] 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. (2017) http://dx.doi.org/10.1016/j.cma.2017.11.030.

[47] T.I. Zohdi, Electrodynamic machine-learning-enhanced fault-tolerance of robotic free-form printing of complex mixtures, Comput. Mech. (2018) http://dx.doi.org/10.1007/s00466-018-1629-y.

[48] T.I. Zohdi, Computation of strongly coupled multifield interaction in particle-fluid systems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 3927–3950.

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

[50] T.I. Zohdi, Genetic design of solids possessing a random-particulate microstructure, Philos. Trans. R. Soc. Math. Phys. Eng. Sci. 361 (1806) (2003) 1021–1043.

[51] C. Labra, E. Onate, High-density sphere packing for discrete element method simulations, Commun. Numer. Methods. Eng. 25 (7) (2009) 837–849.

[52] E. Onate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Engrg. 197 (19–20) (2008) 1777–1800.

[53] E. Onate, M.A. Celigueta, S.R. Idelsohn, F. Salazar, B. Surez, Possibilities of the particle finite element method for fluid-soil–structure interaction problems, Comput. Mech. 48 (2011) 307–318.

[54] J. Rojek, C. Labra, O. Su, E. Onate, Comparative study of different discrete element models and evaluation of equivalent micromechanical parameters, Int. J. Solids Struct. 49 (2012) 1497–1517, http://dx.doi.org/10.1016/j.ijsolstr.2012.02.032.

[55] B. Avci, Wriggers, A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows, J. Appl. Mech. 79 (2012) 010901.

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

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

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

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

[60] N. Lagaros, M. Papadrakakis, G. Kokossalakis, Structural optimization using evolutionary algorithms, Comput. Struct. 80 (2002) 571–589.

[61] M. Papadrakakis, N. Lagaros, G. Thierauf, J. Cai, Advanced solution methods in structural optimisation using evolution strategies, Eng. Comput. J. 15 (1) (1998) 12–34.

[62] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Structural optimization using evolution strategies and neutral networks, Comput. Methods Appl. Mech. Engrg. 156 (1) (1998) 309–335.

[63] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Optimization of large-scale 3D trusses using evolution strategies and neural networks, Int. J. Space Struct. 14 (3) (1999) 211–223.

[64] M. Papadrakakis, J. Tsompanakis, N. Lagaros, Structural shape optimisation using evolution strategies, Eng. Optim. 31 (1999) 515–540.

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

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

[67] W. Gill, M. Wright, Practical Optimization, Academic Press, 1995.

[68] M.A. Finney, FARSITE: Fire area simulator-model development and evaluation. Research paper, RMRS-RP-4-Revised, United States Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2004.

[69] P.L. Andrews, The Rothermel Surface Fire Spread Model and Associated Developments: A Comprehensive Explanation, General Technical Report, RMRS-GTR-371, United States Department of Agriculture, Forest Service, Rocky Mountain Research Station, Fort Collins, CO, 2018.

[70] S. Goodrick, R. Linn, S. Brambilla, Comparing HIGRAD/FIRETEC to QUIC-fire, a coupled fire-atmosphere model for operational applications, 2018, https://ui.adsabs.harvard.edu/abs/2018AGUFMEP31B.07G.

[71] T.I. Zohdi, The game of drones: rapid agent-based machine-learning models for multi-UAV path planning, Comput. Mech. (2019) http://dx.doi.org/10.1007/s00466-019-01761-9.