Module 5G - Modeling and Simulation of Solar Farm Designs Module

Omar Betancourt, Payton Goodrich, Emre Mengi

June 22, 2023

Contents

Objectives: Understanding irradiation, simulating light-energy absorption in agrophotovoltaics.
Prerequisite Knowledge: N/A
Prerequisite Modules: 1A - Calculus, 1B - Linear Algebra, 1D - Differential Equations, 2A - Heat Transfer, 2D - Optics, 3A - Geometric Ray Tracing, 3C - Generic Time Stepping, 4A - Genetic Algorithms, 4B - Gradient-based Optimization
Difficulty: Hard
Summary: This paper focuses on rapid simulation methods based on photon- mapping to quickly evaluate solar farm configurations. The tool allows for a farm to be pulsed and for every photon packet to be tracked and the resulting reflections and absorption to be computed. The is quite useful for proposed next generation farms that blend photovoltaics, thermal-solar, ground reflections, double-sided paneling etc. This work develops an efficient and rapid computational method to simulate a optical pulse in order to ascertain the absorption efficacy of OPTICAL irradiation for a surface. It is based on decomposition of a pulse into a groups of rays, which are then tracked as they progress towards the target contact surface. The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to quickly quantify the absorption efficacy across the topology of a structure. Key variables are the reflectivity of the floor and the degree of specularity.

1 Theory

1.1 Introduction

Solar energy today is by far the most plentiful energy source available. A rudimentary calculation utilizing

- Earth's radius: $R = 6,400,000m$,
- Earth's surface area: $4\pi R$,
- Earth's surface area: $4\pi R^2$ and
- Peak solar energy per unit area: $p = 1300Watts/m^2$,

yields an upper bound on the power total of $P = 4\pi Rp \approx 245,000$ terawatts. The actual number is approximately 173,000 terawatts (due to partial illumination), which is approximately 10,000 times worldwide power usage. Solar energy in general is harnessed in a variety of ways:

- photovoltaics (largest use),
- solar heating (primarily water heating),
- solar-thermal technologies (using mirrors, concentrators and steam turbines),
- molten salt (to store heat and redeploy to run steam turbines),
- artificial photosynthesis (biomimicry of plants).

Photovoltaics systems are the most widely used solar conversion systems. They employ the semi- conducting materials that exhibit photovoltaic effect. With dramatic price drops and readily avail- able silicon, such systems have come to dominate solar energy conversion systems. The first practical was built in 1954 at Bell Labs, with the space industry being an early adopter due power limitations in space for satellites. In the US alone, installed solar energy use has increased by a factor of 20 in the last decade. Issues around the technology now focus of large-scale deployment and integration with other societal systems in urban and rural settings. For example, there are a variety of emerging solar application areas such as agrophotovoltaics, whereby solar farms cohabitate in a symbiotic way with agricultural facilities, with irrigation and other power-dependent systems driven by photo- voltaic energy and plans also cooling down the solar farm for superior performance. This approach was pioneered in the 1980s and has steadily grown as photovoltaic systems have become more ro- bust and inexpensive. Such systems can involve a variety of aspects, ranging from using pollinating insects, such as bees, to "solar grazing" systems. Regardless of the exact type of blended system, Data Science and Machine learning are a necessity to optimize these complex systems so that they operate properly. If configurations are properly optimized, and integrated with an autonomous ir- rigation system, the approach can yield the best of both worlds, yielding energy and abundant agriculture. The approach allows for agricultural use of land in areas that would otherwise have been impossible to utilize. A common theme in these new systems is optimization of solar system deployment. Accordingly, this paper develops such a tool.. This paper focuses on rapid simulation methods based on photon-mapping to quickly evaluate solar farm configurations. The tool allows for a solar farm to be pulsed and for every photon packet to be tracked and the resulting reflections and absorption to be computed. The is quite useful for proposed next generation farms that blend photovoltaics, thermal-solar, ground reflections, double-sided paneling etc. This work develops an efficient and rapid computational method to simulate an optical pulse in order to ascertain the absorption efficacy of OPTICAL irradiation for a surface. It is based on decomposition of a pulse into a groups of rays, which are then tracked as they progress towards the target contact surface. The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to quickly quantify the absorption efficacy across the topology of a structure. The system is also optimized with a machine learning algorithm to maximize the absorption by varying the following (14) parameters:

- Panel inclination: 3 angles,
- Tracking: 1 parameter,
- Refractive index of panels: 1 parameter,
- Specularity of panels: 1 parameter,
- Size of panels: 3 parameters,
- Shape of panels: 3 parameters,
- Ground refractive index: 1 parameter,
- Specularity of ground: 1 parameter.

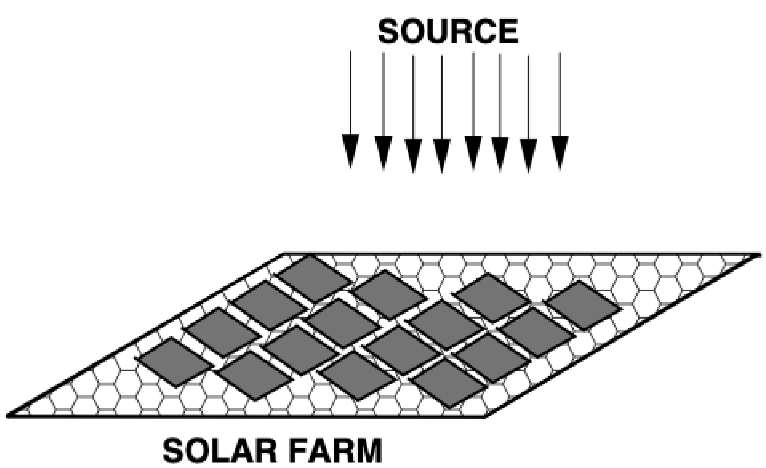


Figure 1.1: Illustration of a solar farm.

1.1.1 Objectives

This module focuses on rapid simulation methods based on photon-mapping to quickly evaluate solar farm configurations. The tool allows for a farm to be pulsed and for every photon packet to be tracked and the resulting reflections and absorption to be computed. The is quite useful for proposed next generation farms that blend photovoltaics, thermal-solar, ground reflections, double- sided paneling etc. This work develops an efficient and rapid computational method to simulate a optical pulse in order to ascertain the absorption efficacy of OPTICAL irradiation for a surface. It is based on decomposition of a pulse into a groups of rays, which are then tracked as they progress towards the target contact surface. The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to quickly quantify the absorption efficacy across the topology of a structure. Key variables are the reflectivity of the floor and the degree of specularity. The tool serves as a guide for practitioners to ascertain where problems may occur a priori to experiments. Additionally, the reflections are calculated, and can be used in ascertaining safety to a bystander. The interest here is on the absorption of an initially coherent pulse (Figure 1.2), represented by multiple collimated (parallel) rays (initially forming a planar wave front), where each ray is a vector in the direction of the flow of energy (the rays are parallel to the initial wave's propagation vector). We make the following observations:

- It is assumed that the features of the surface to be irradiated are at least an order of magnitude larger than the wavelength of the incident radiation (essentially specular surfaces), therefore "geometrical" ray tracing theory is applicable, and is well-suited for the systems of interest. It is important to emphasize the regimes of validity of such a model are where the surface features are larger than the OPTICAL wavelengths. For example, if we were to use OPTICAL-rays ($10^{-8}m \leq \lambda \leq 4 \times 10^{-7}m$), the features in this analysis would be assumed to possess scales larger than approximately $4 \times 10^{-6}m$. For systems containing features smaller than this, one can simply use the model as a qualitative guide.
- Ray-tracing is a method that is employed to produce rapid approximate solutions to wave- equations for high-frequency/small-wavelength applications where the primary interest is in the overall propagation of energy.[1]
- Ray-tracing methods proceed by initially representing wave fronts by an array of discrete rays. *Thereafter, the problem becomes one of a primarily geometric character*, where one tracks the changing trajectories and magnitudes of individual rays which are dictated by the reflectivity and the Fresnel conditions (if a ray encounters a material interface).
- Ray-tracing methods are well-suited for computation of scattering in complex systems that are difficult to mesh/discretize, relative to procedures such as the Finite Difference Time Domain Method or the Finite Element Method.
- Other high frequency irradiation regimes can also be considered in the same manner, such as UV, X-rays and gamma rays, provided that the scattering target has the appropriate (larger) length-scale. Even in the case where this clear separation of length scales is not present, this model still provides valuable information on the propagation of the beam and the reflected response of the dispersed system.

[1]Resolving diffraction (which ray theory is incapable of describing) is unimportant for the applications of interest.

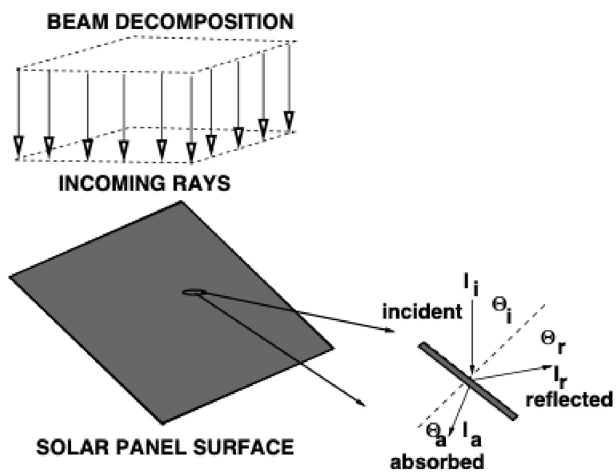


Figure 1.2: An electromagnetic pulse applied to a surface.

1.2 Electromagnetic energy propagation

The 'Optics' module supplies the theory underpinning electromagnetic wave propagation and rays, therefore, refer to that module for details. The summary of the derived relations are given in the following section.

1.2.1 Reflectivity

To observe the dependency of $\boldsymbol{I\!R}$ on $\hat{n}$ and θ_i we can explicitly write

$$\boldsymbol{I\!R} = \frac{1}{2}\left(\left(\frac{\frac{\hat{n}^2}{\hat{\mu}}\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\frac{\hat{n}^2}{\hat{\mu}}\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2 + \left(\frac{\cos\theta_i - \frac{1}{\hat{\mu}}\left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\cos\theta_i + \frac{1}{\hat{\mu}}\left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2\right). \tag{1.1}$$

We observe:

- As $\hat{n} \to \infty$, $\boldsymbol{I\!R} \to 1$, no matter what the angle of incidence's value. We note that as $\hat{n} \to 1$, provided that $\hat{\mu} = 1$, $\boldsymbol{I\!R} \to 0$, i.e. all incident energy is absorbed (it is transparent).
- With increasing $\hat{n}$, the angle for minimum reflectance grows larger. As mentioned previously, for the remainder of the work, we shall take $\hat{\mu} = 1\,(\mu_o = \mu_i = \mu_a)$, thus

$$\hat{n} = \frac{n_a}{n_i} = \sqrt{\frac{\epsilon_a \mu_a}{\epsilon_i \mu_i}} \Rightarrow \epsilon_a \mu_a = (\hat{n})^2 \epsilon_i \mu_i \Rightarrow \epsilon_a = (\hat{n})^2 \epsilon_i. \tag{1.2}$$

- The previous assumption yields

$$\boldsymbol{I\!R} = \frac{I_r}{I_i} = \frac{1}{2}\left(\left(\frac{\hat{\hat{n}}^2\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\hat{n}^2\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2 + \left(\frac{\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2\right) \tag{1.3}$$

- Since $\bar{I}$ is the energy per unit area per unit time, we obtain the energy associated with an entire pulse/beam by multiplying the irradiance by the cross-sectional area of an initially coherent beam, $\bar{I}A^b$, where A^b is the cross-sectional area of the beam (comprising all of the rays).
- The energy per unit time (power) for a ray in the pulse/beam is then given by $I = \bar{I}A^r = \bar{I}A^b/N_r$, where N_r is the number of rays in the beam (Figure 1.2) and A^r can be considered the area associated with a ray.

- The reflection relation, Equation 1.1 can then be used to compute changes in the magnitude of the reflected rays (and the amount absorbed), with directional changes given by the laws of reflection.

We refer the reader to Gross [1] and Zohdi [3-9] for details.

2 Example

From this point forth, we assume that the ambient medium behaves as a vacuum. Accordingly, there are no energetic losses as the rays move through the surrounding medium.

2.1 Modeling Light Ray Trajectories

We model the solar rays as a rectangle of collimated (parallel) rays with the same downward initial velocity. All rays will have the same initial height $\mathcal{H}$. Randomly distributed positions are to be created in the x_1 and x_2 coordinates with user defined bounds given by s_{Reg} and centered at $[x_1, x_2] = [0, 0]$. The random parameters are denoted as $\phi_{1,j}$ and $\phi_{2,j}$ in each respective dimension. The incoming angle of the collimated rays is defined by θ_R with respect to the x-axis. With this information, we can represent the initial position and velocity of the solar rays as follows:

$$\begin{aligned} r_j(t=0) = (x_{1,j}\mathbf{e}_1 + x_{2,j}\mathbf{e}_1 + x_{3,j}\mathbf{e}_3)(t=0) = ((max(x_1) - min(x_1))\phi_{1,j} + min(x_1))\mathbf{e_1} + \\ ((max(x_2) - min(x_2))\phi_{2,j} + min(x_2))\mathbf{e_2} + \\ \mathcal{H} \times max(x_3)\mathbf{e_3} \end{aligned} \tag{2.1}$$

$$v_j(t=0) = (v_{1,j}\mathbf{e}_1 + v_{2,j}\mathbf{e}_1 + v_{3,j}\mathbf{e}_3)(t=0) = 0\mathbf{e_1} - c\sin(\theta_R)\mathbf{e_2} - c\cos(\theta_R)\mathbf{e_3} \tag{2.2}$$

where $c \approx 3 \times 10^8$ m/s is the speed of light in a vacuum.

2.2 Tracking of beam-decomposed rays

Starting at $t = 0$ and ending at $t = T$, the simple overall algorithm to track rays is as follows, at each time increment:

1. Check for intersections of rays with surfaces (hence a reflection), and compute the ray magnitudes and orientation if there are reflections (for all rays that are experiencing a reflection, $I_j^{ref}, j = 1, 2, \ldots Rays$)
2. Increment all ray positions ($\boldsymbol{r}_j(t + \Delta t) = \boldsymbol{r}_j(t) + \Delta t \boldsymbol{v}_j(t), j = 1, 2, \ldots,\ Rays$)
3. Increment time forward ($t = t + \Delta t$) and repeat the process for the next time interval.

In order to capture all of the ray reflections that occur:

- The time step size Δt is dictated by the offset height of the source. A somewhat ad-hoc approach is to scale the time step size by the speed of ray propagation according to $\Delta t = \xi \frac{\mathcal{H}}{\|\boldsymbol{v}\|}$, where $\mathcal{H}$ is the height of the source and $0.0001 \leq \xi \leq 0.01$. Typically, the results are insensitive to ξ that are smaller than this range.
- Although outside the scope of this work, one can also use this algorithm to compute the thermal response by combining it with heat transfer equations via staggering schemes (Zohdi[3,6]).

2.3 Test surface

We consider a ray of light incident upon a material interface which produces a reflected ray and a transmitted/absorbed (refracted) ray. The amount of incident electromagnetic energy per unit time, power (I_i), that is reflected (I_r) is given by a total reflectance $\equiv \frac{I_r}{I_i}$ where $0 \leq I\!\!R \leq 1$ for unpolarized electromagnetic radiation. The reflectance is a function of the angle of incidence of the incoming rays, the medium which the rays travel through, and the material which the rays intersect with. In this simplified problem, we assume the rays travel through a vacuum and thus we can use the nominal speed of light given above. The material parameters defining the solar panel will be the user's choice and is to be optimized in the machine learning algorithm.

The total power per surface area is given by P_{tot} which is to be evenly distributed among the rays based on the total area of light cover being considered A_b. The ray positions are distributed randomly over a square region centered around the origin with a side length of $2s_{Reg}$. With these parameters, we can define the power per ray in a light pulse as follows:

$$P_r = \frac{P_{tot} A_b}{N_r} \tag{2.3}$$

The discrete-ray approach is flexible enough to simulate a wide variety of systems. As a *test surface*, we consider a topology to be irradiated described by $F(x_1, x_2, x_3) = 1$. The outward surface normals, $\boldsymbol{n}$, needed during the scattering calculations, are easy to characterize by writing

$$\boldsymbol{n} = \frac{\nabla F}{\|\nabla F\|} \tag{2.4}$$

The components of the gradient are

$$\nabla F = \frac{\partial F}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial F}{\partial x_2}\boldsymbol{e}_2 + \frac{\partial F}{\partial x_3}\boldsymbol{e}_3 \tag{2.5}$$

Intersections of a ray with a surface are achieved by checking the intersection of the hull envelope equation. A generalized ellipsoidal equation (Equation 2.6) is used where for exponent values of $(p1, p2, p3)$ equal to two, we generate a familiar ellipsoid, for values less than one we generate involute (nonconvex shapes), and for exponent values of $(p1, p2, p3)$ greater than two, we generate a box-like shapes (right).

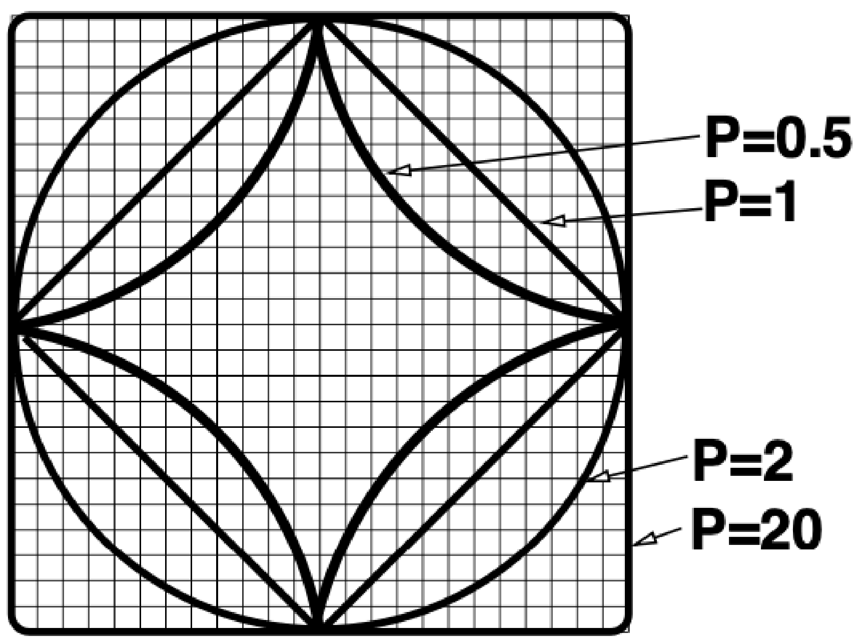


Figure 2.1: Generating an ellipsoid.

To generate a panel, a generalized ellipsoidal equation is used

$$\frac{|x_1 - x_{1o}|^{p_1}}{R_1} + \frac{|x_2 - x_{2o}|^{p_1}}{R_2} + \frac{|x_3 - x_{3o}|^{p_3}}{R_3} \leq 1 \tag{2.6}$$

where (x_1, x_2, x_3) are the coordinates panel center and (R_1, R_2, R_3) are the generalized radii and (p_1, p_2, p_3) are exponents of the generalized ellipsoid.

Note for the flat ground, the inward normal vector is constant and defined by $n_g = [0, 0, -1]$. Next, we can then compute the angle of incidence (θ_i) via the cosine formula between the ray velocity vector (v) and the inward unit normal vector of the solar panel surface (n):

$$\cos\theta_i = \frac{v_j \cdot n_j}{\|v_j\|\|n_j\|} \Rightarrow \theta_i = \cos^{-1}\left(\frac{v_j \cdot n_j}{\|v_j\|\|n_j\|}\right) \tag{2.7}$$

The component of ray velocity normal to the surface of the solar panel is given by

$$v_{j,\perp} = \|v_j\| \cos\theta_i n_j \tag{2.8}$$

We can calculate the outgoing reflected velocity (v_j^{ref}) by turning the inbound normal velocity outward by subtracting it twice:

$$v_j^{\text{ref}} = v_j - 2v_{j,\perp} \tag{2.9}$$

Next, we consider the material properties of the solar panel. We define $\hat{n}$ as the ratio of the refractive indices of the ambient (incident) medium (n_i) and absorbing medium (n_a) such that

$$\hat{n} = n_a/n_i \tag{2.10}$$

The absorbing medium refractive indices, n_a (ground and solar panel), are to be user designed based on the machine learning model. We assume the incident refractive index to be that of a vacuum as $n_i = 1$.
For this model, we will consider applications with non-magnetic media and frequencies where the magnetic permeability is virtually the same for both the incident and absorbing medium. Thus, we can assume $\mu_i \approx \mu_a$. With this assumption, we define the reflectivity as follows:

$$\boldsymbol{I\!R}(\hat{n}, \theta_i) = \frac{I_r}{I_i} = \frac{1}{2}\left(\left(\frac{\hat{n}^2\cos\theta_i - (\hat{n}^2 - \sin^2\theta_i)^{\frac{1}{2}}}{\hat{n}^2\cos\theta_i + (\hat{n}^2 - \sin^2\theta_i)^{\frac{1}{2}}}\right)^2 + \left(\frac{\cos\theta_i - (\hat{n}^2 - \sin^2\theta_i)^{\frac{1}{2}}}{\cos\theta_i + (\hat{n}^2 - \sin^2\theta_i)^{\frac{1}{2}}}\right)^2\right) \tag{2.11}$$

where all parameters are defined above. Recall this is a dimensionless parameter and exists in the range of $\in [0, 1]$. We can use the reflectance to obtain the total amount of absorbed power by a material as follows:

$$P_{abs} = (1 - \boldsymbol{I\!R})P_r \tag{2.12}$$

which will be utilized in the cost function calculation for the machine learning algorithm. We track the total power and position in a ray and stop tracking it if it has either:

1. Moved outside of the cubic domain defined by `domLim`
2. its power reduced below a threshold defined by $P_{min}P_r$

2.4 Genetic Algorithm

A good set of characteristic parameters of our solar panel will allow for optimized absorptivity of the solar panel while controlling the range of absorbed light by the ground. With these in mind, we construct the following cost function.

$$\Pi = w_1\alpha + w_2\gamma \tag{2.13}$$

where the weights are chosen to be $w_1 = 1, \quad w_2 = 0.5$, the panel losses ratio is given by $\alpha = \frac{PanelLosses}{TotalInput}$, and γ represents a range of ground absorption ratio given by

$$\begin{cases} \text{If } G \geq G^+ & \text{then } \gamma = |G - G^+| \\ \text{If } G \leq G^- & \text{then } \gamma = |G - G^-| \\ \text{else} & \gamma = 0 \end{cases} \tag{2.14}$$

where $G = \frac{GroundAbsorption}{TotalInput}$ denotes the ground absorption ratio. This term is designed as a penalty parameter such that an ideal ground absorption ratio lies in between G^+ and G^-. Otherwise, the term is penalized given by the absolute difference between the ground absorption ratio and the closes ground absorption bound.

Note that all terms in the cost function are non-dimensional. The weights reflect the relative importance of each term. The "design string" for this problem contains all 12 undetermined constants:

$$\mathbf{\Lambda}^i = \{\Lambda_1^i, ..., \Lambda_N^i\} = \{\theta_X, \theta_Y, \theta_Z, n_s, n_g, p_1, p_2, p_3, R_1, R_2, R_3, h_0\} \tag{2.15}$$

where $\hat{n}_s$ and $\hat{n}_g$ are the refractive indices of the solar panel and ground respectively.

2.5 Numerical/Quantitative examples

We have the following set up for a series of tests:

- The initial velocity vector for all initially collimated (parallel) rays comprising the beam was $\boldsymbol{v} = (c, 0, 0)$, where $c = 3 \times 10^8 m/s$ is the speed of light in a vacuum.
- We used a parametrized test surface given by Equation 2.6.
- The number of rays in the beam were steadily increased from $N_r = 100, 200$, etc, until the results were insensitive to further refinements. This approach indicated that between approximately $9500 \leq N_r \leq 10000$ parallel rays in rectangular cross-sectional plane of the beam. The rays were randomly placed within the beam (Figure 1.2), to correspond to unpolarized incoming energy, and yielded stable results across the parameter study range.
- This approach also allows an analyst to explore nonuniform beam profiles, for example exponential central irradiance decay: $I(d) = I(d = 0)e^{-ad}$, where d is the distance from the center of the initial beam, where in the case of $a = 0$, one recaptures a flat beam, $I(d) = I(d = 0)$.[1]

2.6 Summary and discussion

Because solar technologies are becoming widely used in industry, with many variants being proposed, fast computational analysis and design tools are needed to ascertain their effectiveness. Accordingly, this work developed a discrete-ray model to allow for propagation of energy encountering a surface, based on the decomposition of irradiation into a groups of rays, which are then tracked as they progress towards the target. This facilitates:

- Quick quantification of the decontamination efficacy across the topology of the structure (color coding the efficacy relative to the incoming irradiation).
- Parametric studies to the changes in absorption as a function of changes in surface geometry.

The simulations take on the order of one minute on a laptop. This type of approach makes it quite suitable for use in conjunction with mobile decontamination systems and allows provides a simpler alternative to a direct, computationally intensive, discretization of a continuum description using Maxwell's equations with a Finite Element or Finite Difference method.

2.7 Machine-Learning for system 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 (Figure **??**). 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. Refer to the 'Genetic Algorithms' Module for the details.

The system was optimized varying the following (14) parameters:

[1] Note that algorithmically, we can the set total initial irradiance via $\sum_{i=1}^{N_r} I_i^{inc}(t = 0)\mathcal{A}_r = P$ Watts. To achieve this distribution, one would first place rays randomly in the plane, and then scale the individual I^{inc} by e^{-ad} and the normalized the average so that the total was P watts.

- Panel inclination: 3 angles,
- Tracking: 1 parameter,
- Refractive index of panels: 1 parameter,
- Specularity of panels: 1 parameter,
- Size of panels: 3 parameters,
- Shape of panels: 3 parameters,
- Ground refractive index: 1 parameter,
- Specularity of ground: 1 parameter.

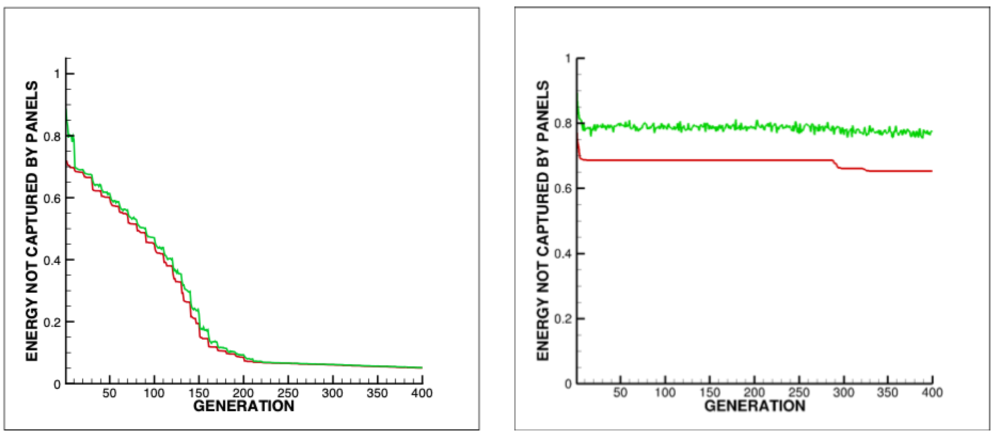


Figure 2.2: The reduction of the cost function for the 13 parameter set. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of genes (designs, in *green*). Left: Allowing the MLA/GA to readapt every 20 generations, leading to the nonmonotone reduction of the cost function. Often, this action is more efficient than allowing the algorithm not readapt, since it probes around the current optimum for better local alternatives-*as is the case in this example*. Right: With no readaptation.

Figure 2.2 shows the reduction of the cost function for the 14 parameter set. This cost function Π represents the percentage of uncaptured energy. 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 $\mathbf{\Lambda} = \{\Lambda_1, \Lambda_2 \ldots \Lambda_N\}$ are optimized over the search intervals (14 variables): $\mathbf{\Lambda}_i^- \leq \mathbf{\Lambda}_i \leq \hat{\Lambda}_i^+, i = 1, 2, \ldots 14$. We used the following MLA settings:

- Number of design variables: 14
- Population size per generation: 50,
- Number of parents to keep in each generation: 10,
- Number of children created in each generation: 10,
- Number of completely new genes created in each generation: 30,
- Number of generations for re-adaptation around a new search interval: 20 and
- Number of generations: 400.

The algorithm was automatically reset every 20 generations. The entire 400 generation simulation, with 50 genes per evaluation (20000 total designs) took a few minutes on a laptop, *making it ideal as a design tool.* Figure 2.2 (average population of 50 genes performance and top gene performance).

Note that the MLA/GA readapts every 20 generations, leading to the nonmonotone reduction of the cost function. Often, this action is more efficient than allowing the algorithm to not readapt, since it probes around the current optimum for better alternatives. 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 0.025 seconds, thus over 100,000 parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA/GA.*

3 Assignment

In this assignment, you will be introduced to the basics of light-based simulation methods. If you use Python or another language for simplifying answers or evaluating specific numbers, clearly set up the equations before showing your numerical results.

The template code to complete the project is given here on `GitHub`.

Problem 1: Theory-Based Exercises

Answer the following questions *prior* to coding the assignment to better understand the background physics and mathematics that govern the given models. You **may** solve these problems by hand **and/or** using computational tools such as `Python` etc. Please include all handwritten work and code used to solve each problem.

Problem 1.1

For a parametrized test surface given by

$$x_3 = 2 + A\left(\sin\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right) \tag{3.1}$$

Analytically solve for the gradient of the surface.

Problem 1.2

Write down the Forward Euler equation for time discretization. Explain all the terms.

Problem 2: Coding Exercises

Use the given python notebook template to complete the following coding exercises.

Problem 2.1

Define the constants used in the simulation. Use the variable glossary at the end of the assignment.

Problem 2.2

Run the genetic algorithm.

Problem 3: Analyzing Your Results

Answer the following questions about the code you created.

Problem 3.1

Report your best-performing 4 designs in a table similar to the following, but replacing $_i$ with the design variables specific to this project. Use `pandas DataFrame` to generate the table in cell `Problem 3.1`.

DESIGN	1	2	3	*etc*	Π
1					
2					
3					
4					

Table 3.1: The top 4 system parameter performers.

Problem 3.2

Provide a convergence plot showing the total cost of the best design, the mean of all parent designs, and the mean of the overall population for each generation. Discuss any important observations.

Problem 3.3

Provide a plot showing the individual performance components (i.e., plot α, γ, and Π_{total}), for the overall best performer. Discuss any important observations such as how the parameters progress over the generations and what trade-offs your algorithm has to consider.

Problem 3.4

Include a series of plots showing how your best design captures the solar rays. Include 5 approximately evenly-spaced frames from $t = 0$ until the final time for your system. The solar panel and all light rays should be clearly shown with distinct marker styles, axes should be labeled.

Table 3.2: System and Integration Parameters

Symbol	Type	Units	Value	Description
N_r	scalar	none	1000	Number of Light Rays
P_{tot}	scalar	W/m^2	1000	Total power per unit area
P_r	scalar	W	Equation 2.3	power per light ray
P_{min}	scalar	none	0.1	fraction of total power at which we stop tracking a ray
s_{Reg}	scalar	m	0.5	Bounds of square pulse of light
`domLim`	scalar	m	2.5	Bounds of domain
$\mathcal{H} = \mathtt{h_{Reg}}$	scalar	m	4	Initial ray height
c	scalar	m/s	3×10^8	Speed of light
$r_j(t = 0)$	3×1 vector	m	Equation 2.1	Initial ray positions
$v_j(t = 0)$	3×1 vector	m/s	Equation 2.2	Initial ray velocities
θ_i	scalar	radians	Equation 2.7	Surface incident angle
θ_R	scalar	radians	$[-\pi/72, 0, \pi/36]$	Angle of incoming light
n_j	3×1 vector	unitless	Equation 2.4	solar panel surface unit normal
n_g	3×1 vector	unitless	$[0, 0, -1]$	ground inward surface normal
Δt	scalar	sec	$0.05(\mathcal{H}/c)$	time step size
A_{ray}	scalar	m^2	$(2s_{Reg})^2$	Area covered by solar rays

Table 3.3: Genetic Algorithm Parameters

Symbol	Type	Units	Value	Description
`K`	scalar	none	6	Strings generated by breeding
`P`	scalar	none	6	Surviving strings for breeding
`S`	scalar	none	20	Designs per generation
`G`	scalar	none	100	Total generations
$[G^-, G^+]$	scalar	m	[0.4,0.8]	cutoff ratios for ground absorption
w_1	scalar	none	1	Weight of reflectivity in cost
w_2	scalar	none	0.5	Weight of surface area ratio in cost
$[\theta^-_{X\ or\ Y\ or\ Z}, \theta^+_{X\ or\ Y\ or\ Z}]$	scalar	radians	$[-\pi/2, \pi/2]$	Search bounds for solar panel angles
$[n^-_{g\ or\ s}, b^+_{g\ or\ s}]$	scalar	none	$[1, 100]$	Search bounds for ground/solar refractive indices
$[R^-_1, R^+_1]$	scalar	none	$[0.05, 0.1]$	Search bounds for first generalized radii
$[R^-_{2\ or\ 3}, R^+_{2\ or\ 3}]$	scalar	none	$[0.125, 1.125]$	Search bounds for second/third generalized radii
$[p^-_{1\ or\ 2\ or\ 3}, p^+_{1\ or\ 2\ or\ 3}]$	scalar	none	$[1, 20]$	Search bounds for geometric exponents
$[h^-_0, h^+_0]$	scalar	m	[2,2.5]	Search bounds for solar panel base height

4 Solution

Problem 1: Theory-Based Exercises

Problem 1.1

Components of the gradient can be computed as:

$$\nabla F = \frac{\partial F}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial F}{\partial x_2}\boldsymbol{e}_2 + \frac{\partial F}{\partial x_3}\boldsymbol{e}_3 \tag{4.1}$$

$G\left(x_1, x_2\right)$ in parametric can be computed by:

$$F\left(x_1, x_2, x_3\right) = G\left(x_1, x_2\right) - x_3 = 0 \tag{4.2}$$

The gradient can be rewritten as:

$$\nabla F = \frac{\partial G}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial G}{\partial x_2}\boldsymbol{e}_2 - \boldsymbol{e}_3 \tag{4.3}$$

For the parametrized test surface given:

$$\begin{aligned} x_3 &= 2 + A\sin\left(\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right) \\ \nabla F &= \frac{\partial\left(2 + A\sin\left(\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right)\right)}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial\left(2 + A\sin\left(\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right)\right)}{\partial x_2}\boldsymbol{e}_2 - \boldsymbol{e}_3 \\ &= \left(\frac{2A\omega_1\pi}{L_1}\right)\cos\left(\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right)\boldsymbol{e}_1 + \left(\frac{2A\omega_2\pi}{L_2}\right)\sin\left(\frac{2\omega_1\pi x_1}{L_1}\right)\cos\left(\frac{2\omega_2\pi x_2}{L_2}\right)\boldsymbol{e}_2 - \boldsymbol{e}_3 \end{aligned} \tag{4.4}$$

Problem 1.2

The forward Euler discretization for position and velocity terms are as follows:

$$r_i(t + \Delta t) \doteq r_i(t) + v_i(t)\Delta t \tag{4.5}$$

Since the speed of light is constant, the only PDE we are solving is to update our ray position.

Problem 2: Coding Exercises

Problem 2.1

Define the constants used in the simulation. Use the variable glossary at the end of the assignment.
Solution Cell Block:

```

# ################################## Problem 2.1 ####################################################

## System Parameters
Ptot = 1000 #FILL IN HERE # Total power per unit area (W/m^2)
Nr = 1000 #FILL IN HERE # Number of light rays
sReg = 0.5 #FILL IN HERE # domain size of solar rays (m)
hReg = 4 #FILL IN HERE # initial height of solar rays (m)
Pmin = 0.1 #FILL IN HERE #fraction of total power at which we stop tracking a ray

# cutoff ratios for ground absorption
Gm = 0.4 #FILL IN HERE
Gp = 0.8 #FILL IN HERE

```

```
domLim = 2.5 #FILL IN HERE # Limit of domain in e1 and e2 domain (m)
c = 3E8 #FILL IN HERE # Speed of light in a vacuum (m/s)
thetaR = 0 #FILL IN HERE # Incoming angle of light (radians)

## Genetic Algorithm Parameters
K = 6 #FILL IN HERE # Strings generated by breeding
P = 6 #FILL IN HERE # Surviving strings for breeding
S = 20 #FILL IN HERE # Design strings per generation
G = 100 #FILL IN HERE # Total Generations
numLam = 12 #FILL IN HERE # Number of parameters per design string
w1 = 1 #FILL IN HERE # Cost function weight
w2 = 0.5 #FILL IN HERE # Cost function weight

# Search Bounds
theta1SB = [-np.pi/2, np.pi/2] #FILL IN HERE # Solar Panel rotation about the e1-axis (radians)
theta2SB = [-np.pi/2, np.pi/2] #FILL IN HERE # Solar Panel rotation about the e2-axis (radians)
theta3SB = [-np.pi/2, np.pi/2] #FILL IN HERE # Solar Panel rotation about the e3-axis (radians)
ngSB = [1,100] #FILL IN HERE # Refractive index of ground
nsSB = [1,100] #FILL IN HERE # Refractive index of solar panel
R1SB = [0.05, 0.1] #FILL IN HERE # Solar Panel size parameter w.r.t. the e1-axis
R2SB = [0.125, 1.125] #FILL IN HERE # Solar Panel size parameter w.r.t. the e2-axis
R3SB = [0.125, 1.125] #FILL IN HERE # Solar Panel size parameter w.r.t. the e3-axis
p1SB = [1, 20] #FILL IN HERE # Solar Panel exponent w.r.t. the e1-axis
p2SB = [1, 20] #FILL IN HERE # Solar Panel exponent w.r.t. the e2-axis
p3SB = [1, 20] #FILL IN HERE # Solar Panel exponent w.r.t. the e3-axis
h0SB = [2, 2.5] #FILL IN HERE # Solar Panel height (m)

# max panel size parameters
R1max = R1SB[1]
R2max = R2SB[1]
R3max = R3SB[1]

# min panel exponent parameters
p1min = p1SB[0]
p2min = p2SB[0]
p3min = p3SB[0]

# #######################################################################################################
```

Problem 2.2

Run the genetic algorithm.

```

################################ Call GA to optimize solar farm system ######################################################

Lam, Pi, Min, PAve, Ave, alphaMin, gammaMin, alphaPAve, gammaPAve, alphaAve, gammaAve = \
    zu.myGA(S,G,P,K,theta1SB, theta2SB, theta3SB, ngSB, nsSB, p1SB, p2SB, p3SB, R1SB, R2SB, R3SB, h0SB,
         Nr,    c, sReg, hReg, numLam, w1, w2, domLim, Ptot, Gm, Gp, Pmin, thetaR,R1max, R2max,R3max,p1min,p2min,p3min)
```

Problem 3: Analyzing Your Results

Answer the following questions about the code you created.

Problem 3.1

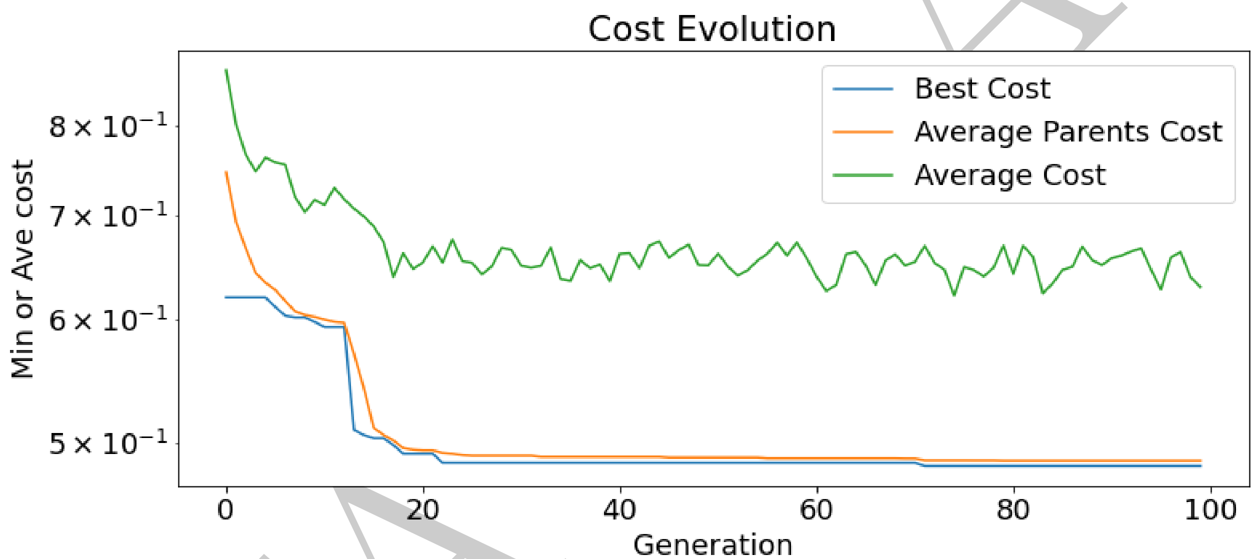
Example output table

Design	theta1	theta2	theta3	ng	ns	R1	R2	R3	p1	p2	p3	h0	Pi
1	0.676996	0.676997	0.676996	0.676997	0.676996	0.676996	0.676996	0.676997	0.676996	0.676997	0.676996	0.676996	0.482625
2	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	-0.773513	0.484849
3	-0.702613	-0.702892	-0.702565	-0.702628	-0.702604	-0.702604	-0.702694	-0.702601	-0.702604	-0.702708	-0.702619	-0.702604	0.487363
4	24.176471	24.175245	24.177630	24.176433	24.176801	24.176854	24.175261	24.176635	24.176810	24.176024	24.176663	24.176845	0.487809

Problem 3.2

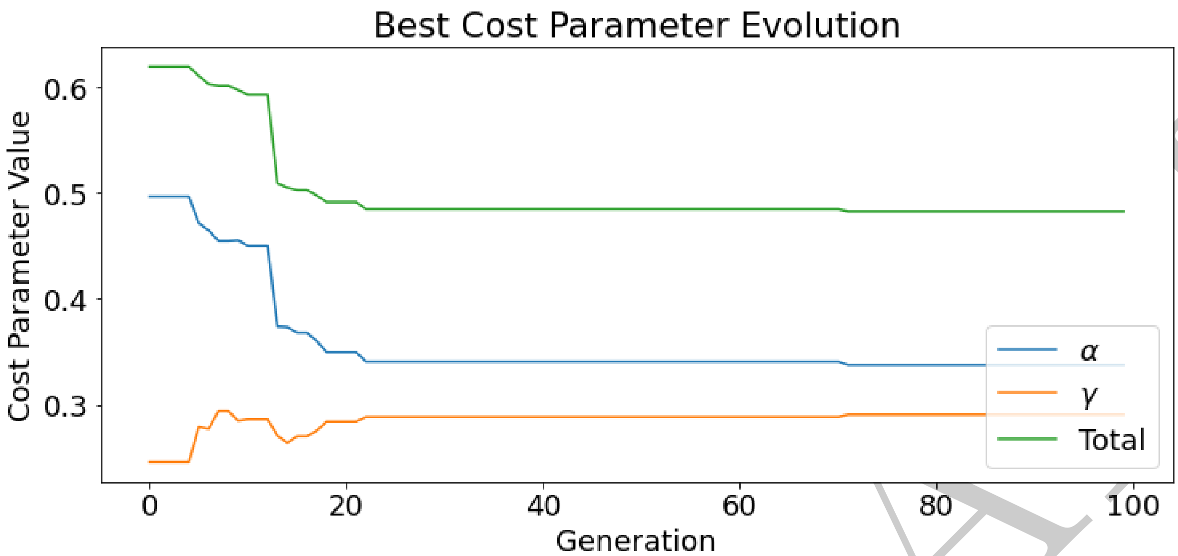
Example solution below

The convergence plot showing the total cost of the best design, the mean of all parent designs, and the mean of the overall population for each generation is as follows:



Problem 3.3

Plots showing the individual performance components (i.e., plot α, γ, and Π_{total}), for the overall best performer.



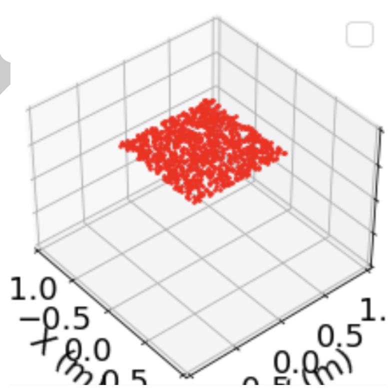
Findings:

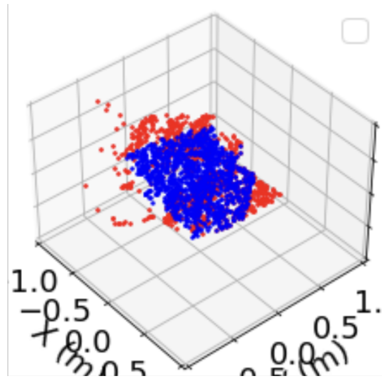
- Best cost is monotonously decreasing
- Individual cost components might increase or decrease by the end, but the total cost is still decreasing
- GA effectively finds the best solution around G=20
- (*Accept answers that are in line with the specific plots generated per student basis*)

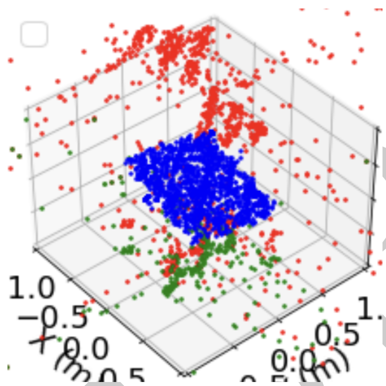
Problem 3.4

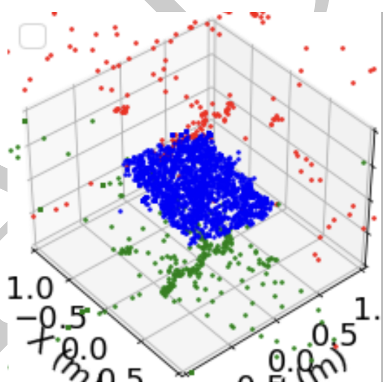
Include a series of plots showing how your best design captures the solar rays. Include 5 approximately evenly-spaced frames from $t = 0$ until the final time for your system. The solar panel and all light rays should be clearly shown with distinct marker styles, axes should be labeled.

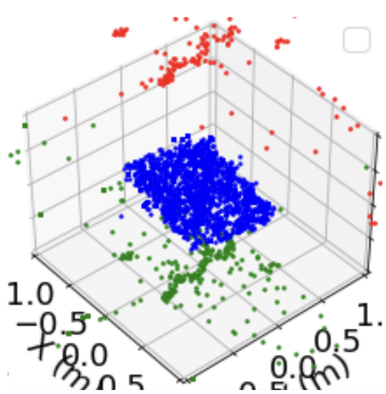
Example animation snippet:




1.0
−0.5
0.5
1.


1.0
−0.5
0.5
1.


1.0
−0.5
0.5
1.


1.0
−0.5
0.5
1.

5 Ethical Considerations for this Project

A goal of this project is to enable advancements in science and engineering through to address critical national challenges associated with next generation food systems. There are deep ethical considerations associated with any technology, in particular for food systems. While technology has tremendous potential to identify greater efficiencies, when it is created without appropriate consideration for who will have access to and control over new resources, or how the new technologies will impact those who work in the system, the efficiencies identified may come at the cost of greater societal inequity. It is important to pursue harnessing technology to disrupt existing inequities, rather than further entrench existing power structures. The following areas should be considered:

- Labor: 1) occupational health, 2) food manufacturing, and 3) outdoor agriculture labor;
- Producers: 1) Small- to mid-size farms, 2) urban agriculture, and 3) research in farm transitions; Technology: 1) research in technology and democracy;
- Health Human Rights: 1) land rights, 2) social justice, and 3) decolonization in agriculture;

Please consider the following questions:

- What are the societal implications of the technology that you are developing?
- Can this technology be distributed fairly and equitably to a wide variety of entities in agricultural industry?
- Are there any potential unintended consequences of this technology becoming available?
- Are there any harmful "spinoffs" of this technology?
- Are there any useful "spinoffs" of this technology?

6 References

1. Gross, H. (2005). Handbook of optical systems. Fundamental of technical optics. H. Gross, Editor. Wiley-VCH.

2. Jackson, J. D. (1998). Classical Electrodynamics.

3. Zohdi, T. I. (2006a). Computation of the coupled thermo-optical scattering properties of random particulate systems. Computer Methods in Applied Mechanics and Engineering. Volume 195, 5813-5830.

4. Zohdi, T. I. (2006b). On the optical thickness of disordered particulate media. Mechanics of Materials. Volume 38, 969-981.

5. Zohdi, T. I and Kuypers, F. A. (2006c). Modeling and rapid simulation of multiple red blood cell light scattering. Proceedings of the Royal Society Interface. Volume 3, Number 11 Pages 823-831.

6. Zohdi, T. I. (2012). Electromagnetic properties of multiphase dielectrics. A primer on modeling, theory and computation. Springer-Verlag.

7. Zohdi, T. I. (2015). A computational modeling framework for high-frequency particulate obscu- rant cloud performance. The International Journal of Engineering Science. 89, 75-85.

8. Zohdi, T. I. (2016). On high-frequency radiation scattering sensitivity to surface roughness in particulate media. Computational Particle Mechanics. http://dx.doi.org/10.1007/s40571-016- 0118-3

9. Zohdi, T. I. (2019). Rapid simulation-based uncertainty quantification of flash-type time-of- flight and Lidar-based body-scanning processes. Computer Methods in Applied Mechanics and Engineering. https://doi.org/10.1016/j.cma.2019.03.056

10. Zohdi, T. I. (2018). Electrodynamic machine-learning-enhanced fault-tolerance of robotic free- form printing of complex mixtures. Computational Mechanics. https://doi.org/10.1007/s00466- 018-1629-y

11. Holland, J. H. 1975. Adaptation in natural artificial systems. Ann Arbor, Mich. University of Michigan Press.

12. Holland, J.H.; Miller, J.H. (1991). Artificial Adaptive Agents in Economic Theory (PDF). Amer- ican Economic Review. 81 (2): 365-71. Archived from the original (PDF) on October 27, 2005.

13. Goldberg, D. E. 1989. Genetic algorithms in search, optimization machine learning. Addison- Wesley.

14. Davis, L. 1991. Handbook of Genetic Algorithms. Thompson Computer Press.

15. Onwubiko, C. 2000 Introduction to engineering design optimization. Prentice Hall.

16. Goldberg, D. E. Deb, K. 2000. Special issue on Genetic Algorithms. Computer Methods in Applied Mechanics Engineering. 186 (2-4) 121-124.

17. Zohdi, T. I. (2009) Mechanistic modeling of swarms. Computer Methods in Applied Mechanics and Engineering. Volume 198, Issues 21-26, Pages 2039-2051.

18. Zohdi, T. I. (2018). Multiple UAVs for Mapping: a review of basic modeling, simulation and applications. Annual Review of Environment and Resources. https://doi.org/10.1146/annurev- environ-102017-025912

19. Zohdi, T. I. (2019). The Game of Drones: rapid agent-based machine-learning models for multi- UAV path planning. Computational Mechanics. https://doi.org/10.1007/s00466-019-01761-9

20. Zohdi, T., (2020) A machine-learning framework for rapid adaptive digital-twin based fire- propagation simulation in complex environments. Computer Methods in Applied Mechanics and Engineering. https://doi.org/10.1016/j.cma.2020.112907

21. Luenberger, D. 1974. Introduction to Linear Nonlinear Programming. Addison-Wesley, Menlo Park.

22. Gill, P. Murray, W. and Wright, M. 1995. Practical optimization. Academic Press.