Convolution-Based Numerical Solutions of Transient Temperature Fields during Powder Bed Fusion Additive Manufacturing: Theory, Accuracy, and Computational Cost

Brodan Richter[1], **Joshua D. Pribe[2]**, George Weber[1], Peter Spaeth[1], Edward H. Glaessgen[1]

[1]NASA Langley Research Center, Hampton, VA
[2]Analytical Mechanics Associates, Hampton, VA

brodan.m.richter@nasa.gov
joshua.pribe@nasa.gov

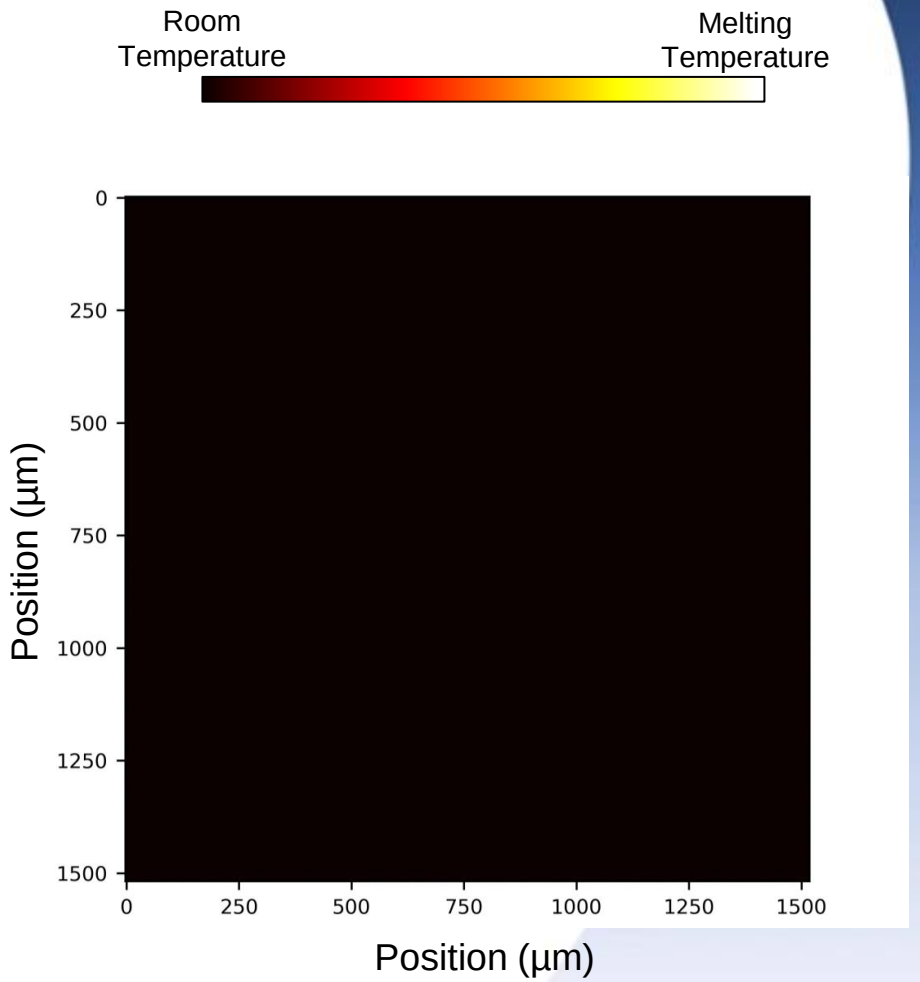
MS&T24 Technical Meeting and Exhibition

Outline and Bottom-Line Up Front

- Powder bed fusion (PBF) components have a complex build time-temperature history that significantly impacts subsequent mechanical performance
  - Understanding thermal history and deviations is important for PBF component qualification and certification (Q&C)

- Convolution-based numerical solutions of transient temperature fields support PBF simulations with arbitrary beam shapes and paths
  - Support future computational materials-informed Q&C approaches

- Method highlights:
  - Similar to the explicit Euler method but with a stencil that increases in size based on the time step
  - Computationally stable and efficient
  - Easily implemented on GPUs
  - Integrates well with microstructure evolution models



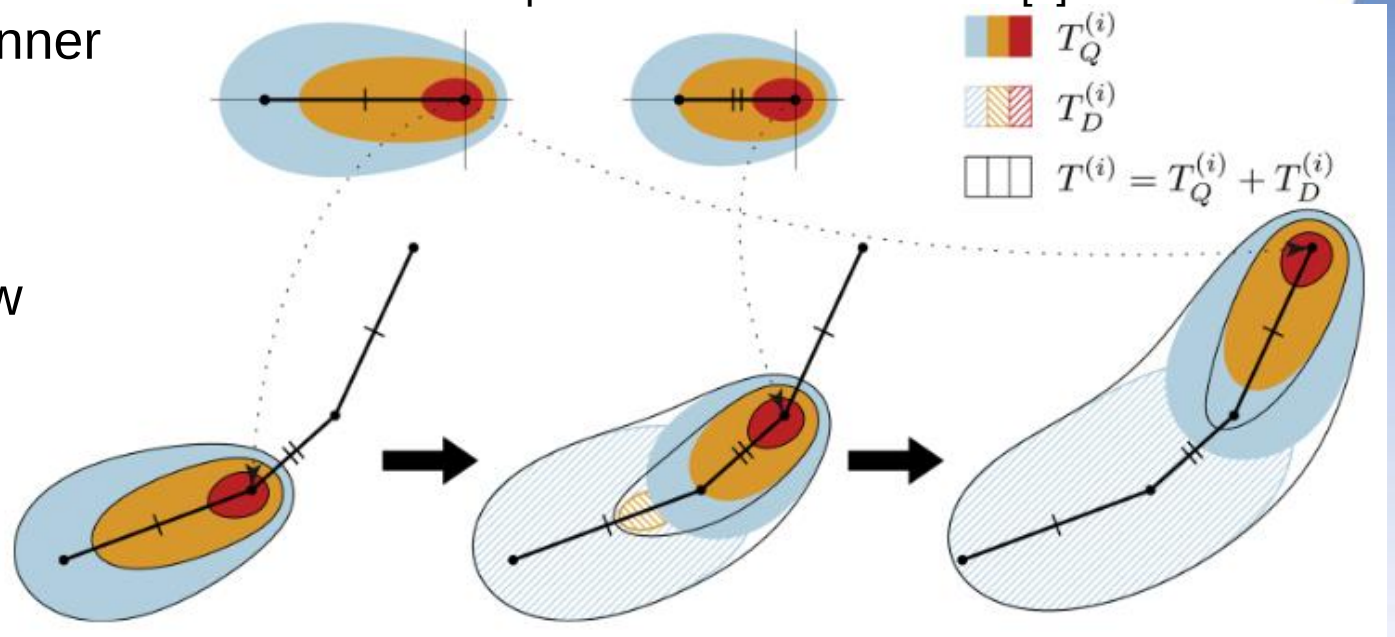
Background

- Wolfer et al. developed the Repeated Use of Stored Line Solutions (RUSLS) Method [1]
  - Segment domain, pre-compute temperature raises from the beam, and repeatedly use them alongside a diffusion step
  - The diffusion step required varied time steps based on scan length

- For modeling the diffusion step, used Gaussian convolution in a similar manner as Gaussian blurring of 2D images

- Method was interesting, but it was difficult to find additional detail on how and why the convolution step worked
  - Accuracy in paper was through comparisons to finite element analysis solutions

Example of the RUSLS Method [1]



[1] Wolfer, Alexander J., et al. "Fast solution strategy for transient heat conduction for arbitrary scan paths in additive manufacturing." Additive Manufacturing 30 (2019): 100898. Figure 2 used under CC BY-NC-ND, https://creativecommons.org/licenses/by-nc-nd/4.0/,

Goals of the Present Work

- The goal of the present work is to assess convolution-based numerical solutions of transient temperature fields in three ways:
  - Theory
  - Accuracy
  - Computational Cost

- Verify accuracy the convolution methodology by comparing to analytical solutions
  - Compare analytical solutions of the heat equation for pure diffusion
  - Compare against analytical translating laser point source solution

- Benchmark computational cost against other numerical solution methods

- Assess the use of the convolution method in supporting PBF process-structure simulations

- Mature the methodology for use in computational materials PBF applications
  - Support future computational materials-informed Q&C approaches

The Heat Equation and a Characteristic Solution

The three-dimensional heat equation

$$\frac{\partial T(\mathbf{r},t)}{\partial t} = \alpha \nabla^2 T(\mathbf{r},t) + \frac{\dot{Q}(\mathbf{r},t)}{\rho c_p}$$

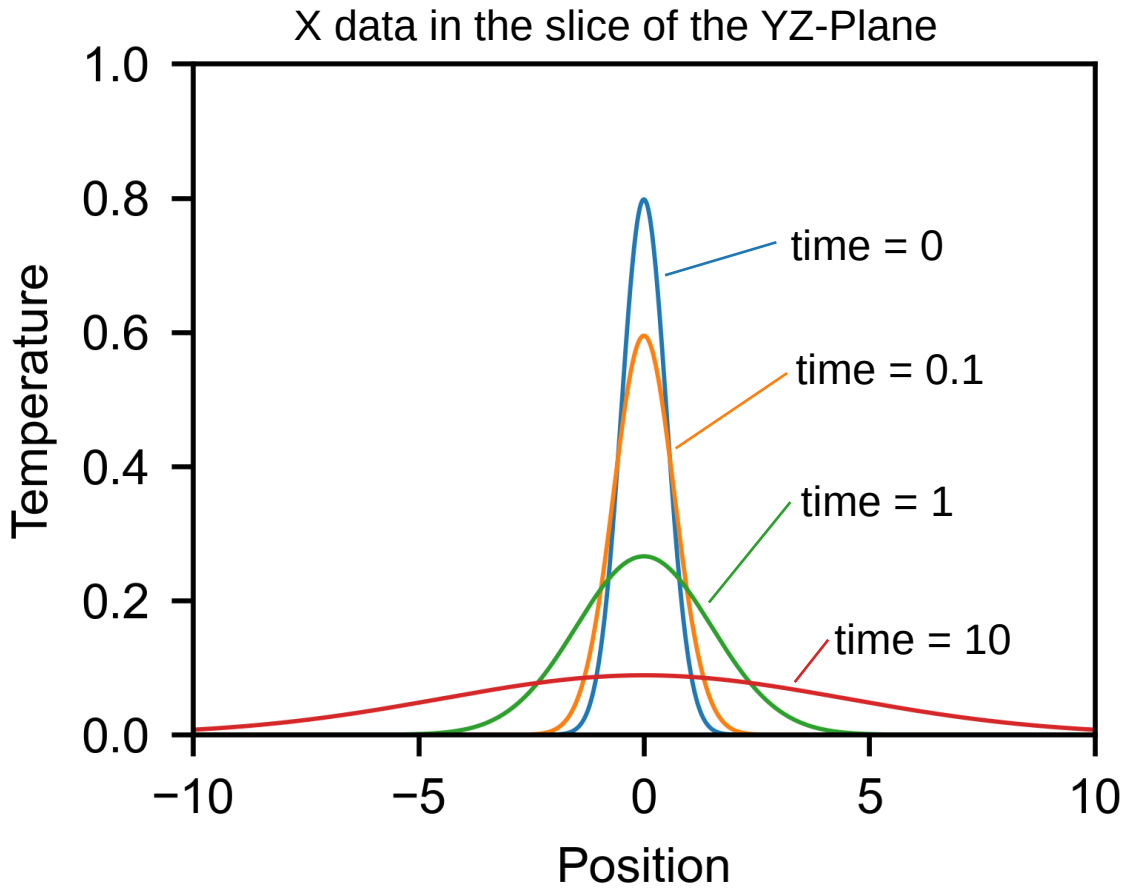
The problem definition

$$T(\mathbf{r},0) = \frac{1}{(2\pi\sigma_0^2)^{3/2}} \exp\left[-\frac{\|\mathbf{r}\|^2}{2\sigma_0^2}\right] \qquad \mathbf{r} \in \mathbb{R}^3, t > 0$$

$$\dot{Q}(\mathbf{r},t) = 0 \qquad \iiint T(\mathbf{r},t)\, dV = 1$$

Characteristic Solution

$$T(\mathbf{r},t) = \frac{1}{\left(2\pi(\sigma_0^2 + 2\alpha t)\right)^{3/2}} \exp\left[-\frac{\|\mathbf{r}\|^2}{2(\sigma_0^2 + 2\alpha t)}\right]$$



T: Temperature
t: Time
$\mathbf{r}$: Position vector
x, y, z: Spatial dimensions

α: Thermal diffusivity
$\dot{Q}$: Power generation
ρ: Density
c_p: Specific heat capacity

$\mathbb{R}$: Real space
σ_0: Standard deviation of starting temperature field

Numerical Solutions to the Heat Equation: Explicit Euler Method

Explicit Euler Method

One-dimensional heat equation $\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{\Delta x^2}$$

$$\tau = \frac{\alpha \Delta t}{\Delta x^2}$$

$$T_i^{n+1} = T_i^n(1 - 2\tau) + \tau(T_{i+1}^n + T_{i-1}^n)$$

Unstable if negative

$$(1 - 2\tau) < 0$$

Unstable in 1D if:

$$\tau > \frac{1}{2}$$

$$2\text{D: } \tau > \frac{1}{4}$$

$$3\text{D: } \tau > \frac{1}{6}$$

Matrix Form

$$T_i^{n+1} = \begin{bmatrix} \tau & 1 - 2\tau & \tau \end{bmatrix} \begin{bmatrix} T_{i-1}^n \\ T_i^n \\ T_{i+1}^n \end{bmatrix}$$

$$\mathbf{T^{n+1}} = \mathbf{A_e T^n}$$

$$\mathbf{A_e} = \begin{bmatrix} \vdots & & \vdots & & \\ \cdots & \tau & 1 - 2\tau & \tau & \cdots \\ \vdots & & \vdots & & \end{bmatrix}$$

Explicit Euler Observations:

- Method is easy to solve
- Computationally efficient
- Conditionally stable

n: Time step
i: Node spatial position
$\mathbf{A_e}$: Explicit Laplacian matrix

Δt: Time step
Δx^2: Voxel size

Numerical Solutions to the Heat Equation: Implicit Euler and Crank-Nicolson Methods

Implicit Euler Method

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1}}{\Delta x^2}$$

$$\mathbf{A_i} = \begin{bmatrix} & \vdots & \\ \cdots \quad -\tau & 1+2\tau & -\tau \quad \cdots \\ & \vdots & \end{bmatrix}$$

$$\mathbf{A_i T^{n+1}} = \mathbf{T^n} \qquad \mathbf{T^{n+1}} = \mathbf{A_i^{-1} T^n}$$

Implicit Euler Observations:

- Method is numerically stable
- Requires either inverting $\mathbf{A_i}$ (slow) or iteratively solving $\mathbf{A_i T^{n+1}} = \mathbf{T^n}$

Crank–Nicolson (CN) Method

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{2}\left[\frac{T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1}}{\Delta x^2}\right] + \frac{\alpha}{2}\left[\frac{T_{i+1}^{n} + T_{i-1}^{n} - 2T_i^{n}}{\Delta x^2}\right]$$

$$\mathbf{A_{ic} T^{n+1}} = \mathbf{A_{ec} T^n} \qquad \mathbf{T^{n+1}} = \mathbf{A_{ic}^{-1} A_{ec} T^n}$$

CN Observations:

- Method is numerically stable
- Method is more accurate
- Requires either inverting $\mathbf{A_{ic}}$ (slow) or iteratively solving $\mathbf{A_{ic} T^{n+1}} = \mathbf{A_{ec} T^n}$

$\mathbf{A_i}$: Implicit Laplacian matrix $\mathbf{A_{ic}}$: Implicit CN Laplacian matrix $\mathbf{A_{ec}}$: Explicit CN Laplacian matrix

Numerical Solutions to the Heat Equation: Spectral and Convolution Methods

Spectral Method

$$\mathcal{F}(T) = \{T\}$$

$$\frac{\partial \{T\}_k}{\partial t} = -\alpha k^2 \{T\}_k$$

$$\frac{\{T\}_k^{n+1} - \{T\}_k^n}{\Delta t} = -\alpha k^2 \{T\}_k^{n+1}$$

$$\{T\}_k^{n+1} = \frac{\{T\}_k^n}{1 + \alpha k^2 \Delta t}$$

Spectral Method Observations:

- Method is numerically stable
- Computationally fast
- Easiest with periodic boundary conditions

Convolution Method

$$T^{n+1} = G * T^n = \int_{-\infty}^{\infty} G(x') T^n(x - x')\, dx = \sum_{m=-M}^{M} G(m) T^n(x - m)$$

Continuous / Discrete

$$G(m) = \{-M, -M+1, \ldots, M-1, M\}$$

$\mathcal{F}$: Fourier Transform
k: Fourier mode coefficient
G: Convolution kernel
x': Continuous shift
m: Discrete shift
M: Set of G

Numerical Solutions to the Heat Equation: Convolution Method

Convolution Method

$$T^{n+1} = \sum_{m=-M}^{M} G(m)T(x-m)$$

$$T_i^{n+1} = \sum_{j=-1}^{1} G_j^n T_{i-j}^n$$

$$G^n = [1/4 \quad 1/2 \quad 1/4]$$

$$T^{n+1} = \sum_{j=1}^{3} G_j^n T_{i-j}^n = \frac{1}{4}T_{i+1}^n + \frac{1}{2}T_i^n + \frac{1}{2}T_{i-1}^n$$

$$T_i^{n+1} = [1/4 \quad 1/2 \quad 1/4]\begin{bmatrix} T_{i-1}^n \\ T_i^n \\ T_{i+1}^n \end{bmatrix}$$

$$\tau = 1/4$$

Explicit Euler Method:

$$T_i^{n+1} = [\tau \quad 1-2\tau \quad \tau]\begin{bmatrix} T_{i-1}^n \\ T_i^n \\ T_{i+1}^n \end{bmatrix}$$

Convolution Method Observations:

- Similar to the explicit Euler method

Numerical Solutions to the Heat Equation: Convolution Method

Convolution Method

$$T_i^{n+1} = \sum_{j=-M}^{M} G_j^n T_{i-j}^n$$

$$T_i^{n+1} = \cdots + G_{-1}^n T_{i+1}^n + G_0^n T_i^n + G_{+1}^n T_{i-1}^n + \cdots$$

$$Will\ be\ unstable\ if\ \ G_0^n < 0$$

$$G(\mathbf{r}, \sigma) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^{1/N}} \exp\left[-\frac{\|\mathbf{r}\|}{2\sigma^2}\right]$$

$$G_0^n = \frac{1}{\sqrt{2\pi}\sigma} \qquad \sigma = \frac{\sqrt{2\alpha\Delta t}}{\Delta x}$$

Always greater than 0

Stencil Size

$\alpha = \Delta x = 1$ $\quad M = \text{ceil}[\sigma]$

$\Delta t \leq 0.5, \quad M = 1 \quad G^n = [0.27 \quad 0.46 \quad 0.27]$

$\Delta t \leq 2, \quad M = 2 \quad G^n = [0.15 \quad 0.22 \quad 0.26 \quad 0.22 \quad 0.15]$

$\Delta t \leq 4.5, \quad M = 3 \quad \cdots$

- Increasing stencil size creates challenges due to interactions with the boundary

Convolution Method Observations:

- Similar to the explicit Euler method
- Numerically stable
- Stencil increases as the time step gets bigger

σ: Gaussian kernel standard deviation

Implementation in Python

- Methods are straightforward to implement in Python using SciPy* packages [2]

$$\mathbf{T}^{n+1} = \mathbf{A_e}\mathbf{T}^{n}$$

```
Explicit
        T += A_explicit @ T*dt
```

$$\mathbf{A_i}\mathbf{T}^{n+1} = \mathbf{T}^{n}$$

```
Implicit
        T = scipy.sparse.linalg.cg(A_implicit, T)[0]
```

$$\mathbf{A_{ic}}\mathbf{T}^{n+1} = \mathbf{A_{ec}}\mathbf{T}^{n}$$

```
Crank-Nicolson Method
        T = scipy.sparse.linalg.gmres(A_cn_implicit,
                                      A_cn_explicit @ T)[0]
```

$$\{T\}_k^{n+1} = \frac{\{T\}_k^n}{1 + \alpha k^2 \Delta t}$$

```
Spectral
        Tk = scipy.fft.fftn(T)
        Tk = Tk/denom
        T = np.real(scipy.fft.ifftn(Tk))
```

$$T^{n+1} = \sum_{m=-M}^{M} G(m)T(x - M)$$

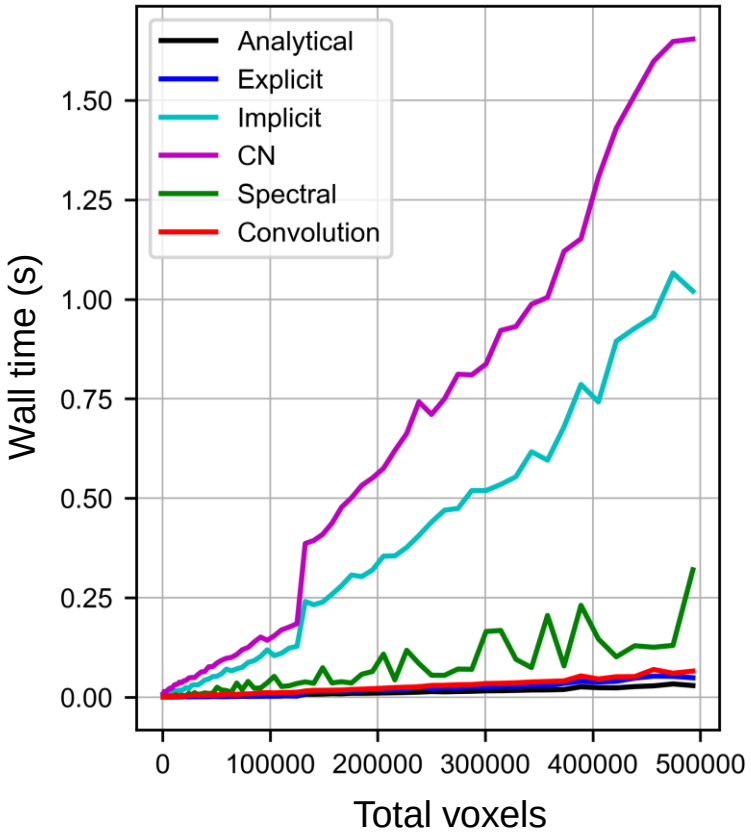
```
Convolution
        T = scipy.ndimage.gaussian_filter(T, sigma)
```

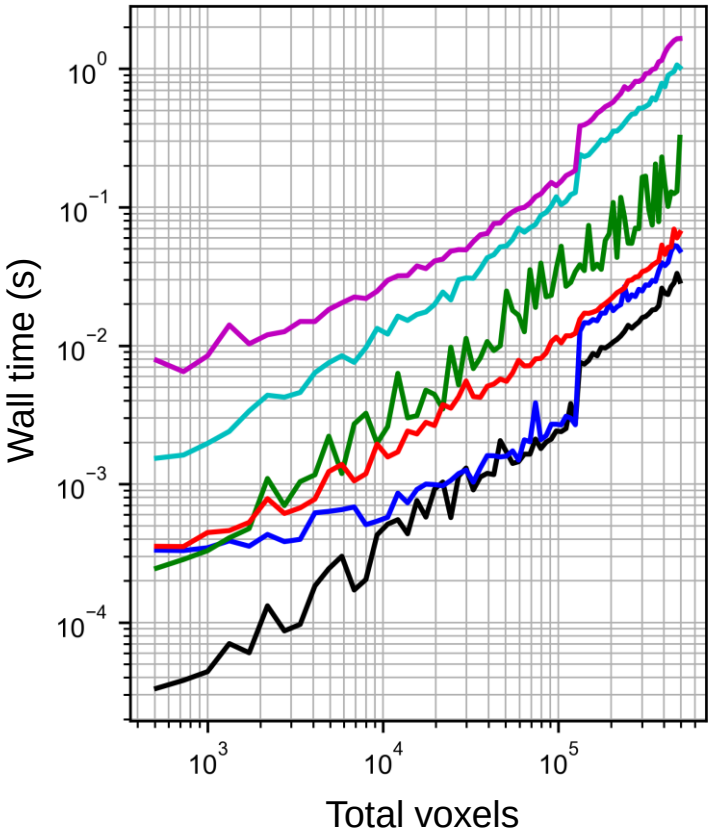
[2] Virtanen, Pauli, et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python." *Nature methods* 17.3 (2020): 261-272.

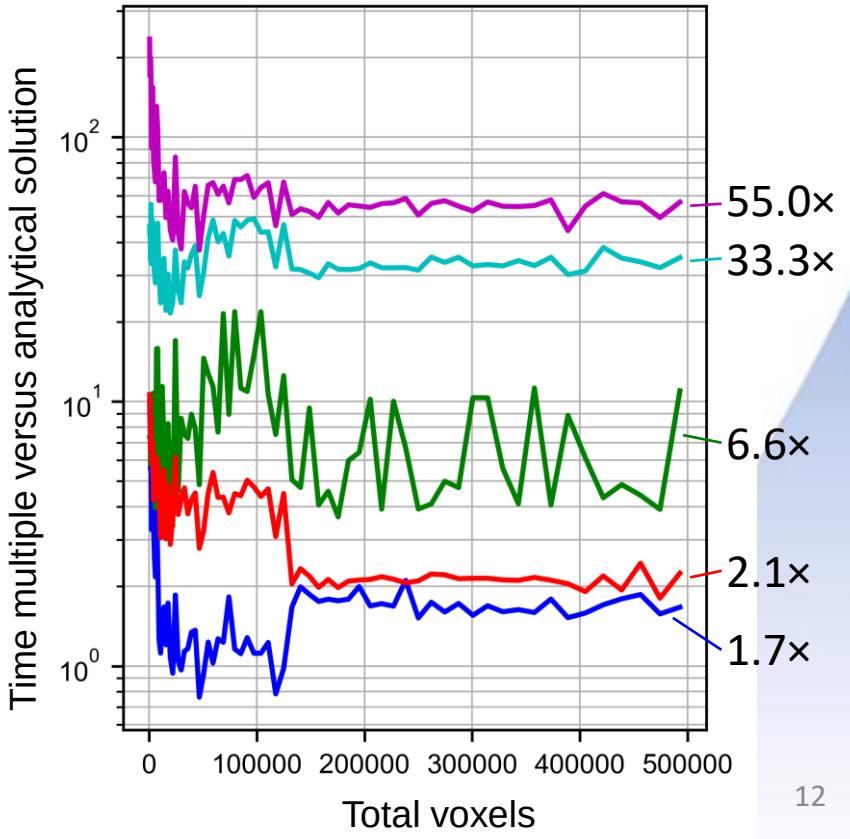
*Specific software package names are used only to accurately describe the software and does not constitute an endorsement by NASA

Computational Cost Comparisons

- Computational cost scales linearly with total number of voxels
  - All methods use the same time steps
- The implicit and CN methods both are the most expensive methods
- Explicit method is lowest cost, followed by convolution method
  - ~2× cost of analytical solution

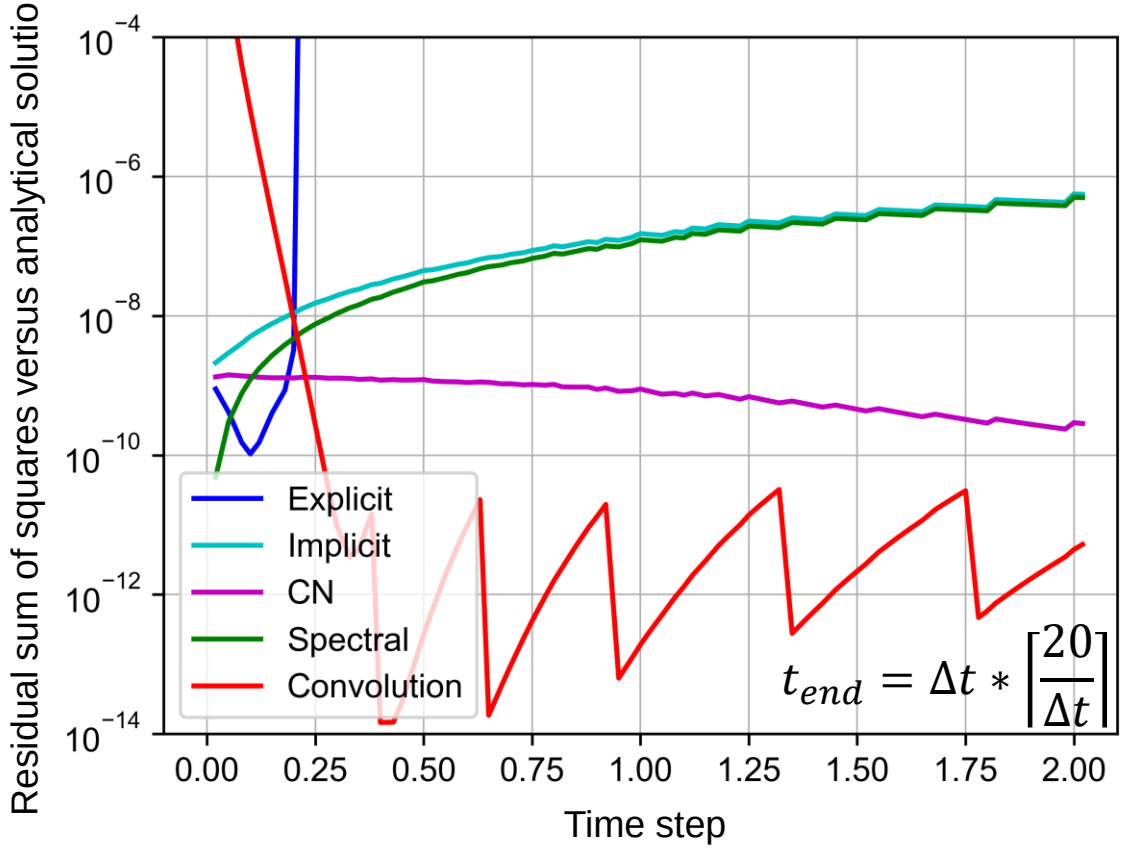


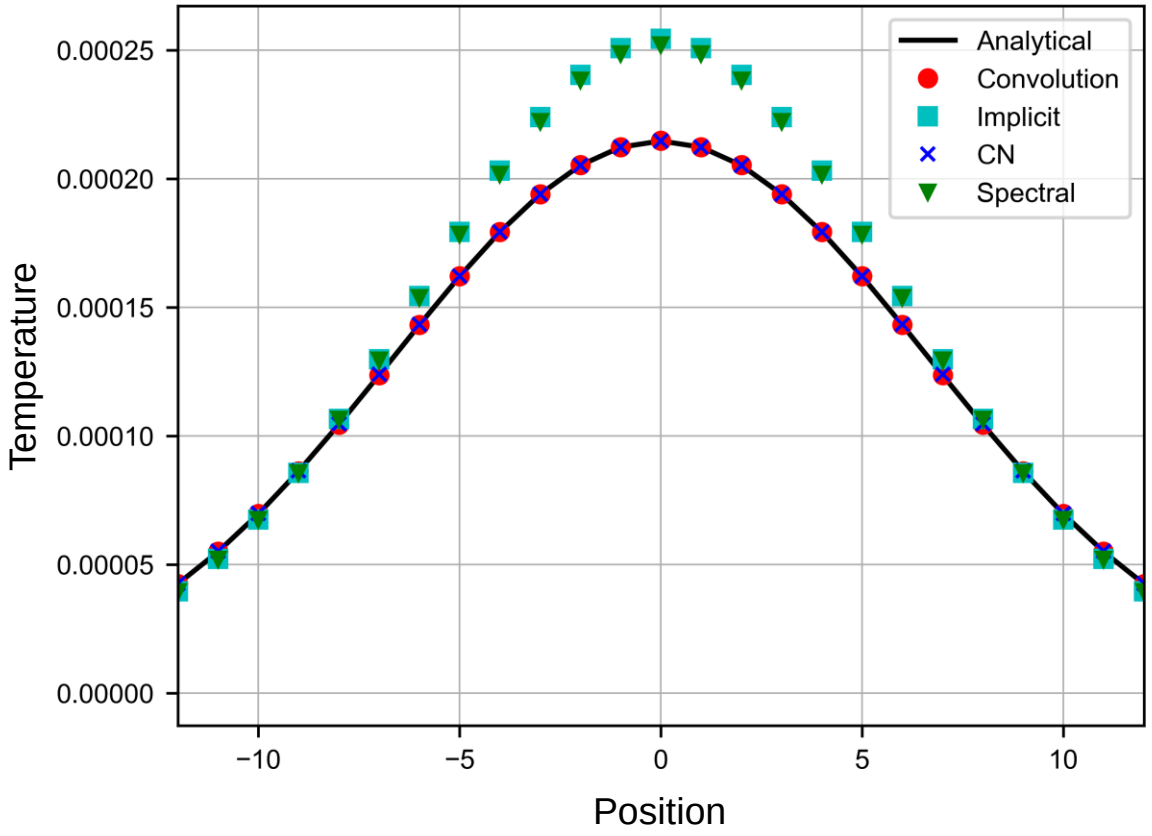




Accuracy Comparisons

- At low time steps, the convolution method has poor accuracy (due to stencil normalization)
- At high time steps, the convolution method has excellent accuracy
- The explicit method becomes unstable around a time step of 1/6 (the stability limit)
- The implicit and spectral methods remain stable, but the accuracy worsens





Summary of Findings for the Specific Problem

- Five methods for numerically solving a specific solution to the heat equation were discussed:

Method	Stable at large time steps	Computational cost	Accuracy at fixed large time step	Boundary region size
Explicit Euler	No	Low	Low	Small
Implicit Euler	Yes	High	Low-Moderate	Small
Crank-Nicolson	Yes	High	Moderate	Small
Spectral	Yes	Moderate	Low-Moderate	Periodic
Convolution	Yes	Low	High	Big

- The convolution method provides a high accuracy, stable, and low computational cost solution for simulating temperature diffusion
  - Challenges exist with the boundary region
  - Works similar to the explicit Euler method with an increasing stencil side

Extension to PBF Processing Modeling

- Method is extended to simulating PBF processing by defining an elliptical heat source, computing the heat rise, and diffusing the temperature field

- Accuracy can be compared by assessing the numerical solution against the Rosenthal solution [3]
  - Analytical solution to a translating point-source
  - Need to define elliptical radii very close to zero

Rosenthal Solution

$$T(\mathbf{r},t) = T_0 + \frac{\eta P}{2\pi k R}\exp\left[-\frac{V(R+\zeta)}{2\alpha}\right]$$

$$R = \sqrt{\zeta^2 + y^2 + z^2} \qquad \zeta = x - Vt$$

Heat Flux from Elliptical Beam

$$\dot{Q}(\mathbf{r},t) = \frac{6\sqrt{3}\eta P}{\pi\sqrt{\pi}abc}\exp\left[-\frac{3x_b^2}{a^2} - \frac{3y_b^2}{b^2} - \frac{3z_b^2}{c^2}\right]$$

$$\lim_{x_b, y_b, z_b \to 0} \dot{q}(\mathbf{r},t)$$

$$\frac{\partial T(\mathbf{r},t)}{\partial t} = \alpha\nabla^2 T(\mathbf{r},t) + \frac{\dot{Q}(\mathbf{r},t)}{\rho c_p}$$

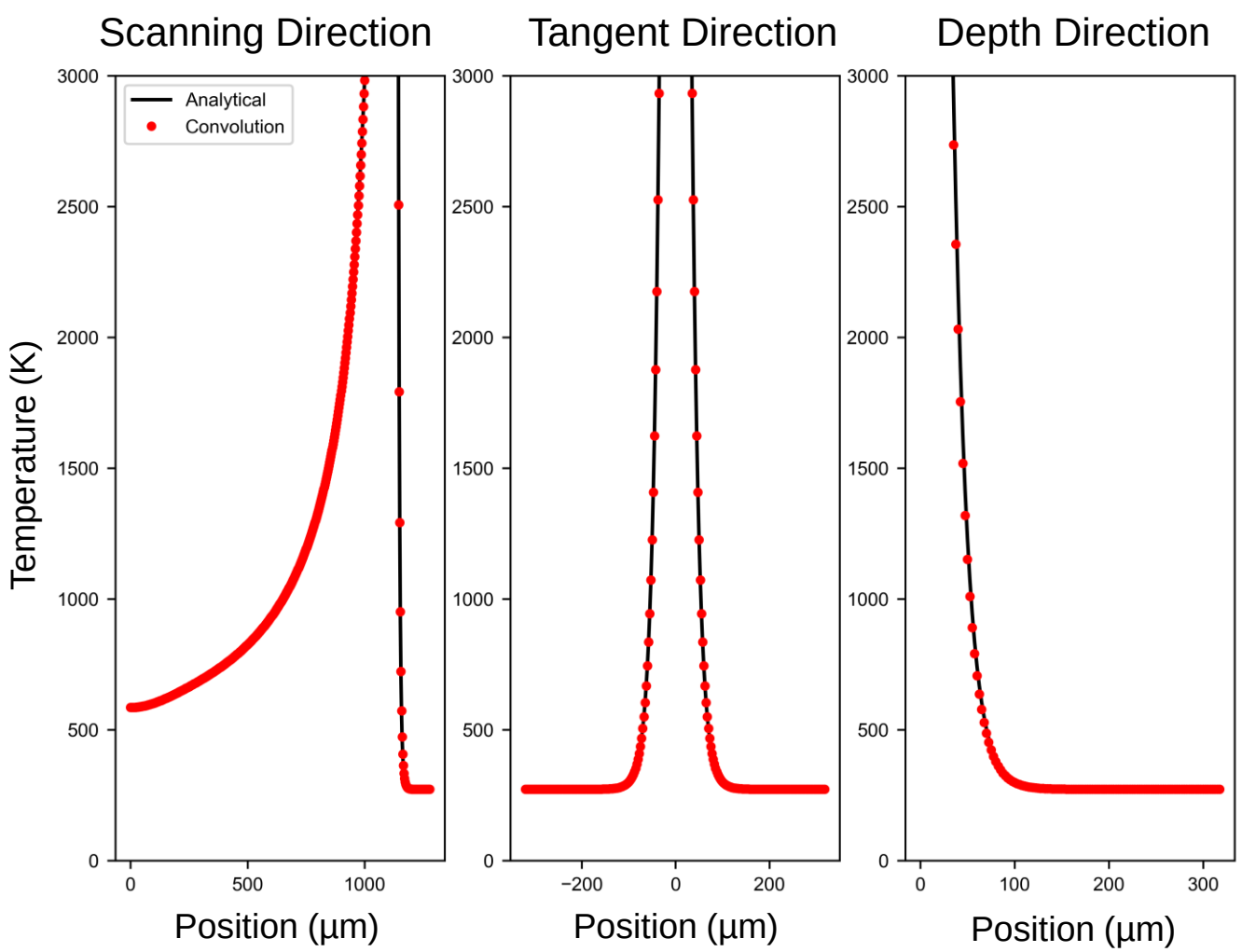
```
while time < time_end
    dtemp_beam = Q_dot / rho / c_p
    dtemp_diffusion = scipy.ndimage.gaussian_filter(temp, sigma)
    temp = dtemp_beam*dt + dtemp_diffusion
    time = time + dt
```

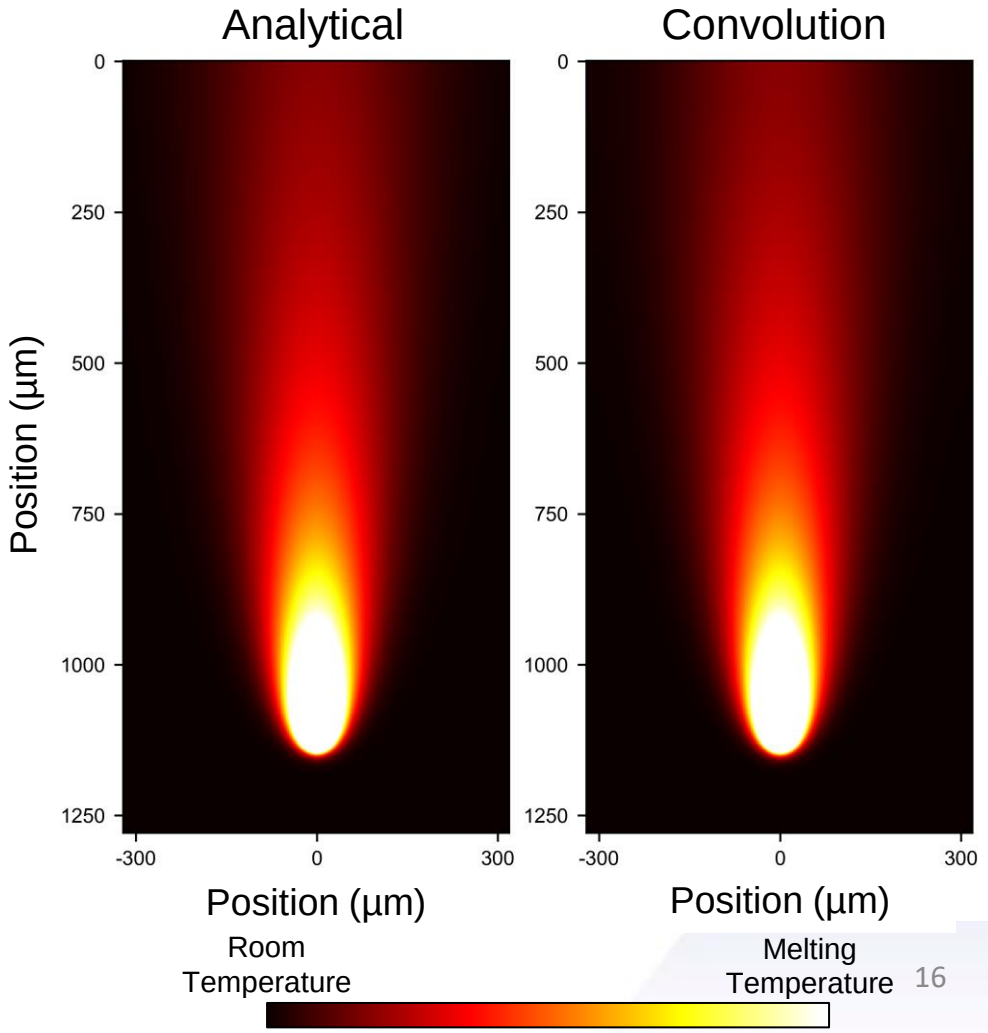
T_0: Room temperature
P: Laser power
η: Absorptivity
k: Thermal conductivity
ζ: Shifted x coordinate system
V: Laser velocity
a, b, c: Beam radii
x_b, y_b, z_b: Coordinate system in beam reference frame

[3] Rosenthal, Daniel. "The theory of moving sources of heat and its application to metal treatments." Transactions of the American Society of Mechanical Engineers 68.8 (1946): 849-865.

Comparisons to Rosenthal Solution

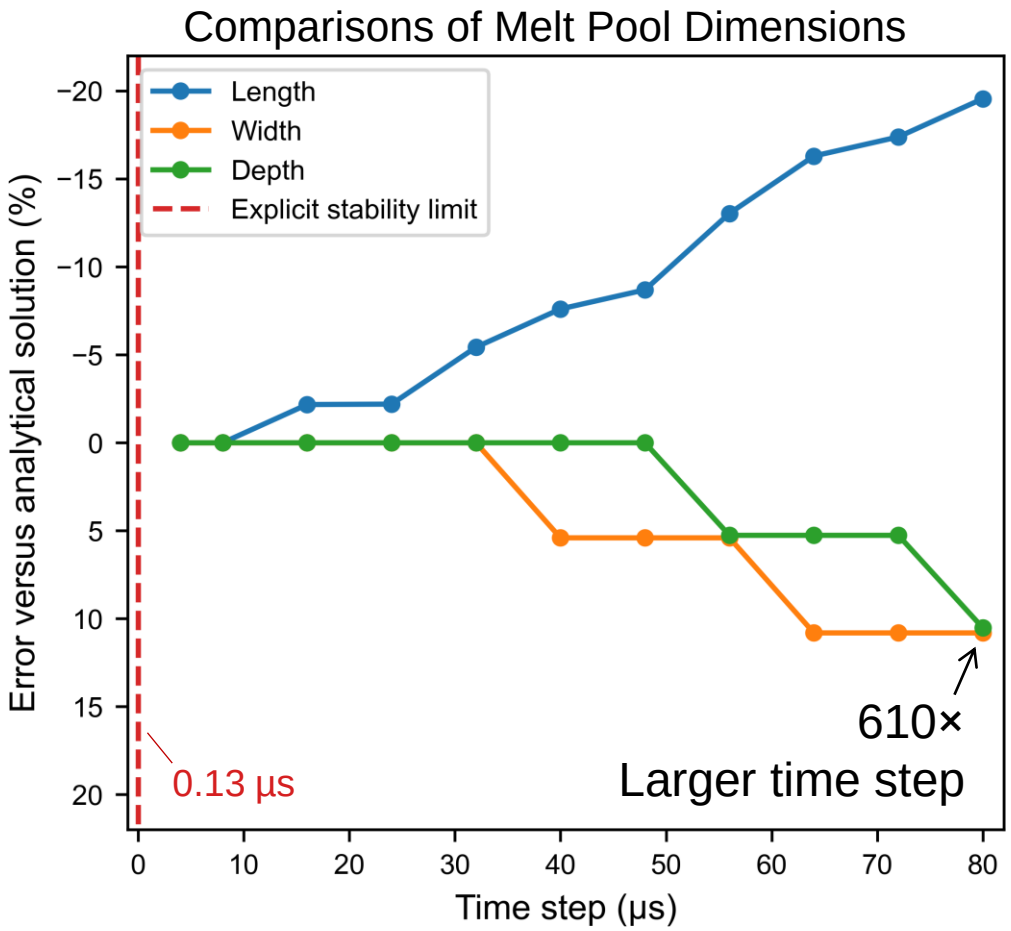
- Method shows excellent agreement with the Rosenthal solution

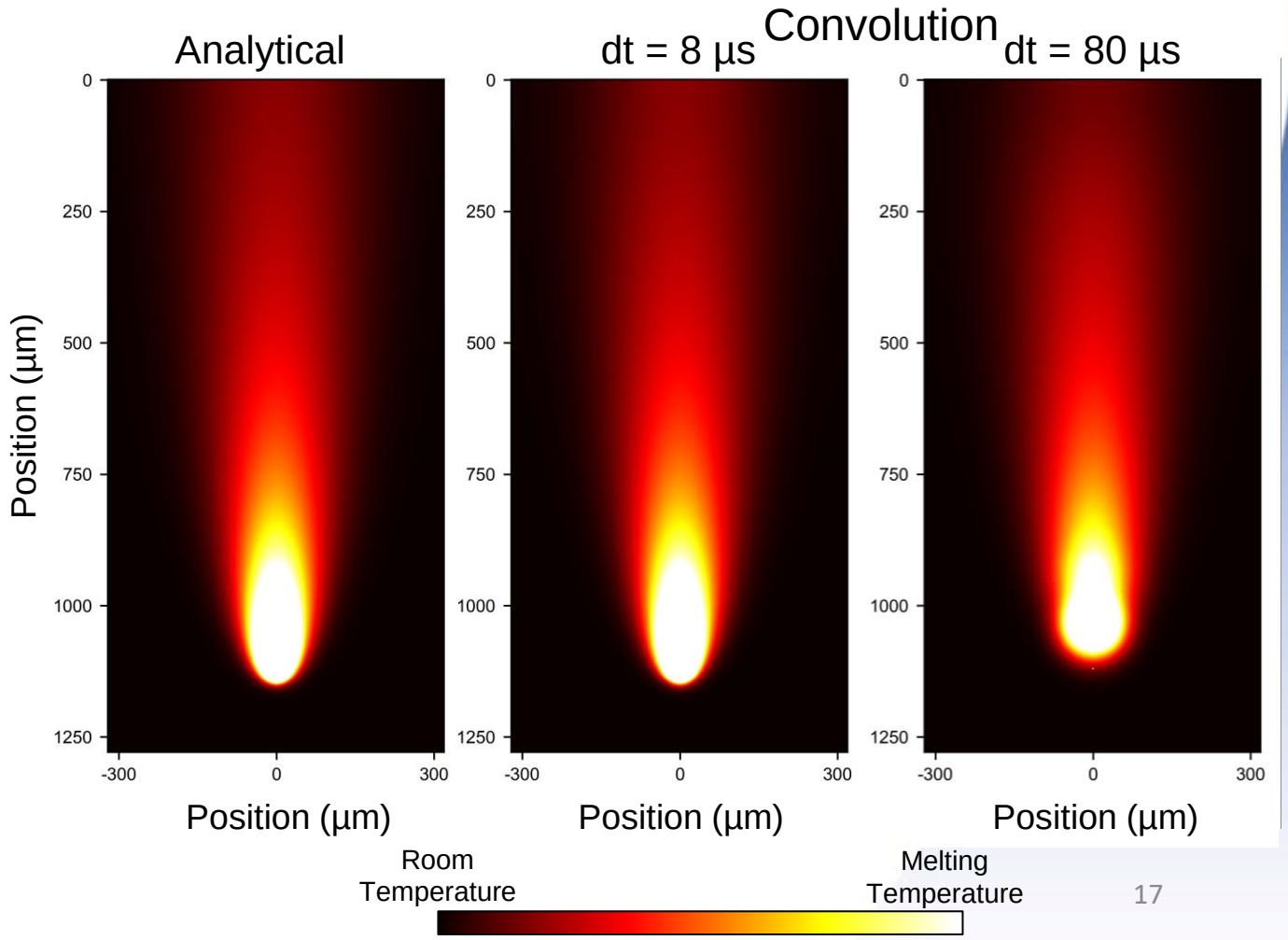




Comparisons to Rosenthal Solution

- Accuracy is also generally maintained at large time steps
- Can simulate at 610× larger time step than explicit Euler method with ~20% error

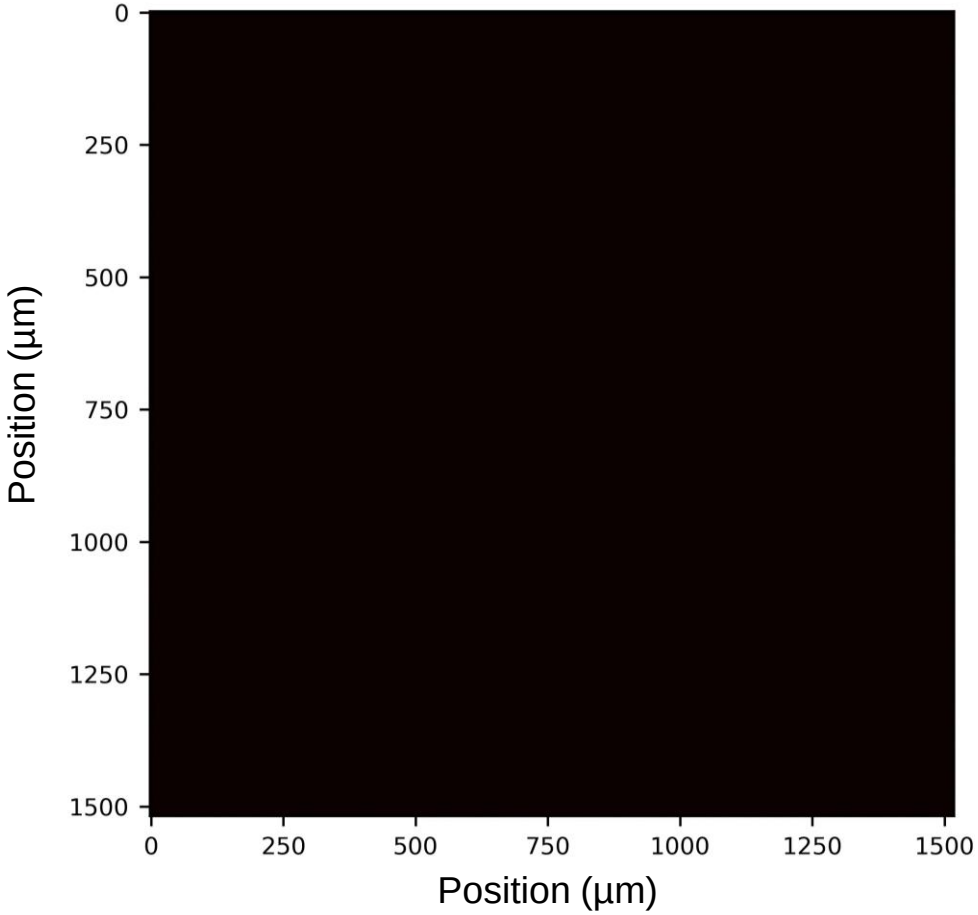




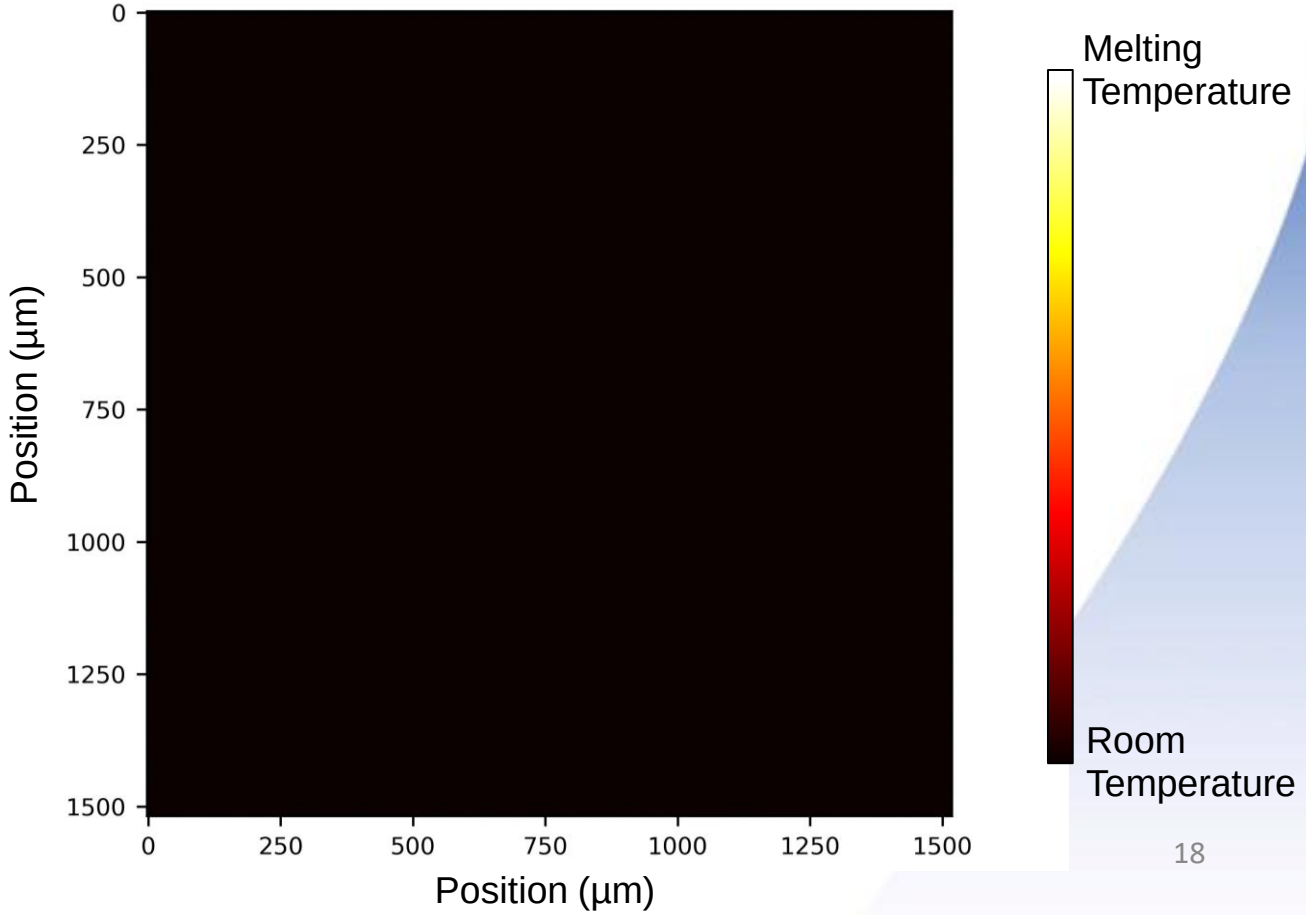
Extension to General Laser Paths

- Methodology is straightforward to extend to arbitrary laser paths with different boundary conditions
  - Laptop GPU, ~30 seconds, 304×304×40 domain, 5240 time steps

Adiabatic / Neumann boundary condition



Constant temperature / Dirichlet boundary condition



Integration with Microstructure Evolution Models

- Technique is also readily integrated with microstructure evolution models
- Implemented the Monte Carlo model of Rodgers et al. [4, 5]

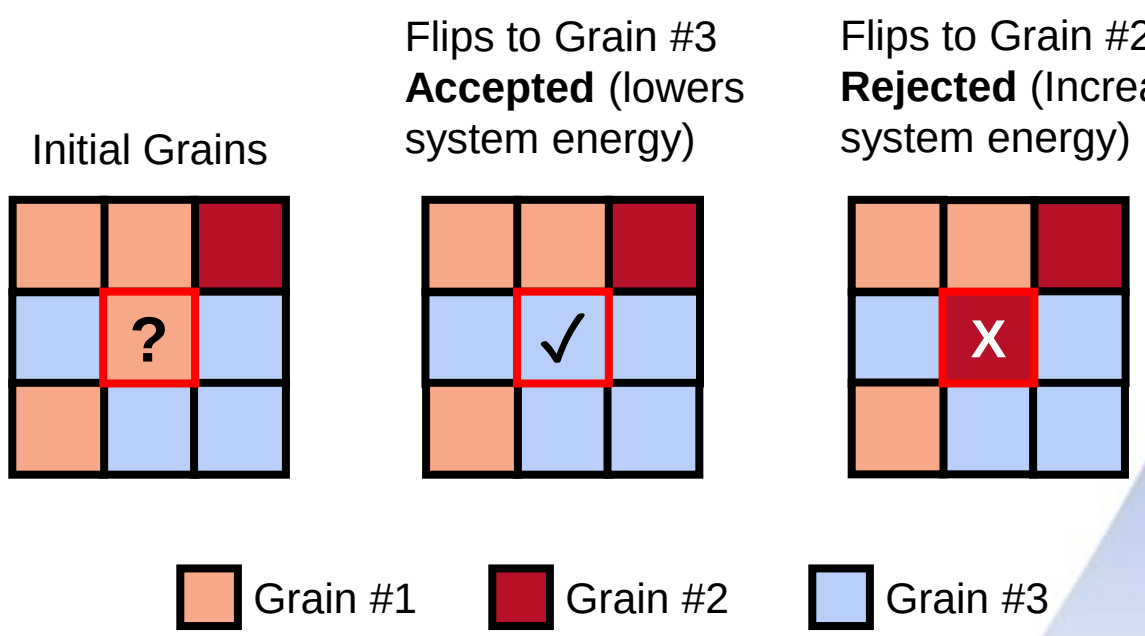
Overview of Monte Carlo Method

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{L}\left(1-\delta\left(q_i, q_j\right)\right)$$

Simulated temperature

$$M(T) = M_0 \exp\left(-\frac{Q\left(T - T_{ref}\right)}{\bar{R}\, T\, T_{ref}}\right)$$

$$P = \begin{cases} M(T)\exp\left(-\dfrac{\Delta E}{k_b T_s}\right) & if\ \Delta E > 0 \\ M(T) & if\ \Delta E \leq 0 \end{cases}$$



E: Site energy
N: Sites
L: Neighbor of sites
q: Spin

δ: Dirac delta function
Q: Boundary activation energy
T_{ref}: Reference temperature
$\bar{R}$: Ideal gas constant

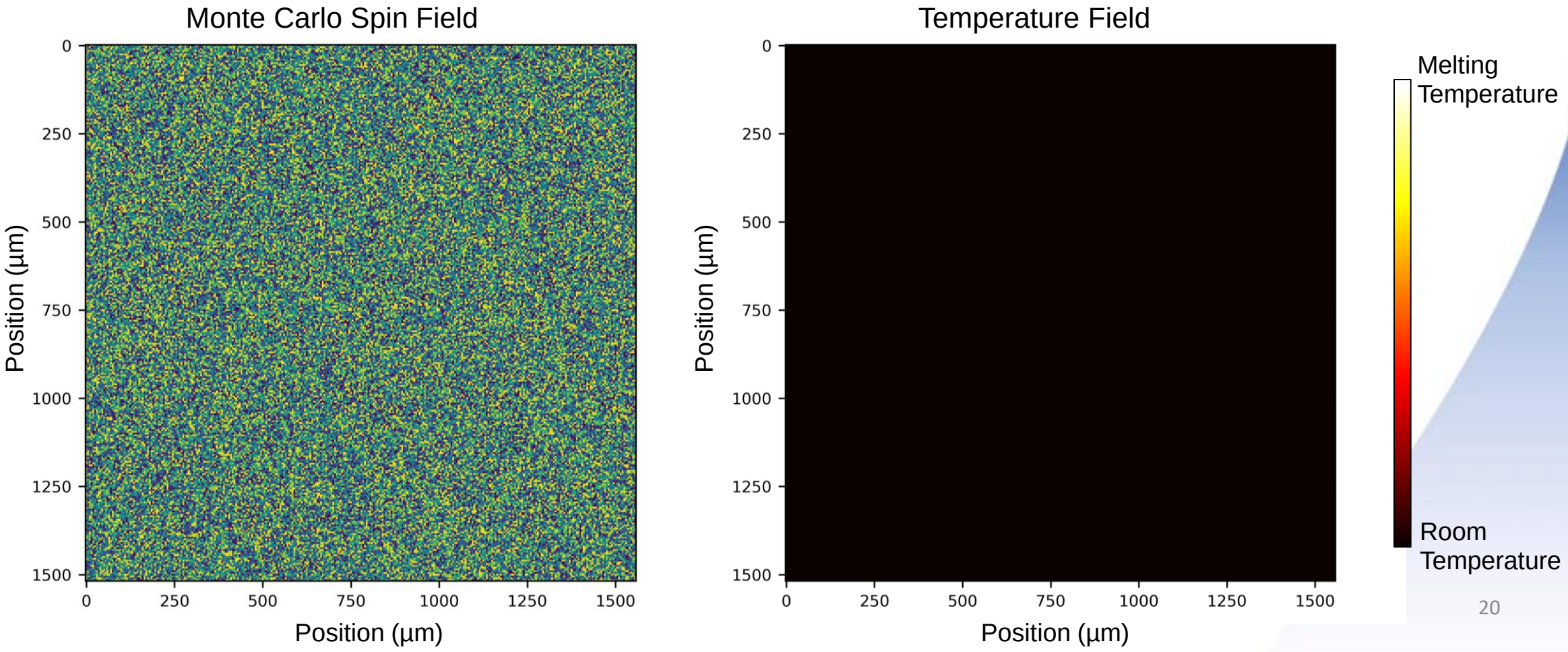
P: Flip probability
T_S: Simulation temperature
k_b: Boltzmann Constant

[4] Rodgers, Theron M., Jonathan D. Madison, and Veena Tikare. "Simulation of metal additive manufacturing microstructures using kinetic Monte Carlo." Computational Materials Science 135 (2017): 78-89.

[5] Rodgers, Theron M., et al. "Simulation of powder bed metal additive manufacturing microstructures with coupled finite difference-Monte Carlo method." Additive Manufacturing 41 (2021): 101953.

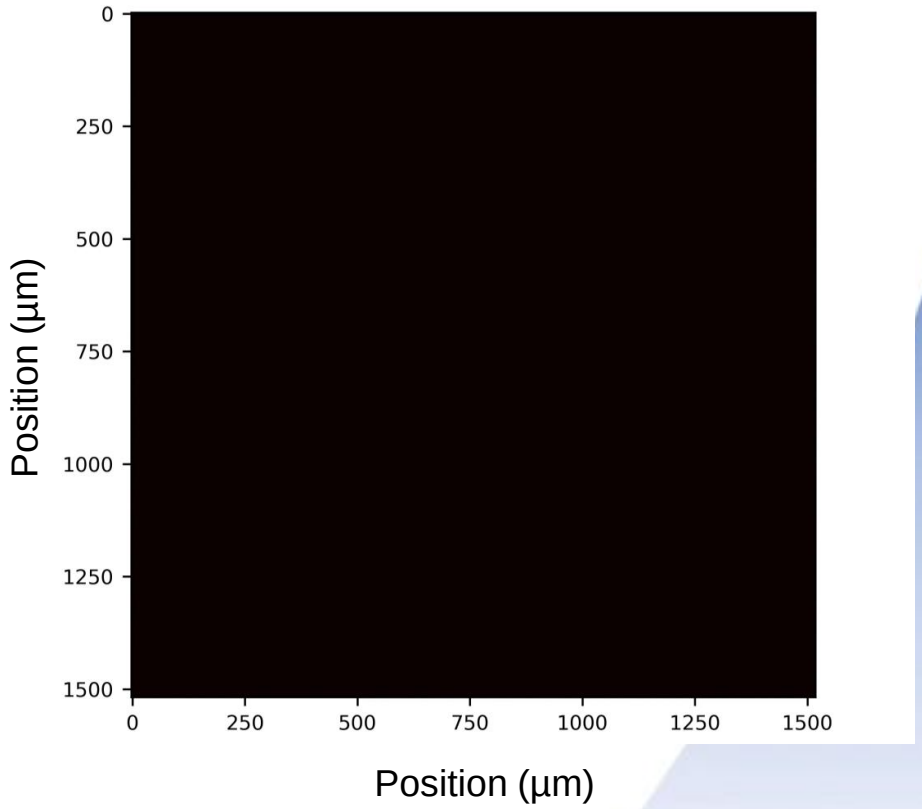
Integration with Microstructure Evolution Models

- Technique is also readily integrated with microstructure evolution models
- Implemented the Monte Carlo model of Rodgers et al. from 2017 [4, 5]
- Below shows the integration of the temperature field and Monte Carlo model for PBF



Summary, Conclusions, and Considerations

- Convolution-based numerical solutions of transient temperature fields support PBF simulations with arbitrary beam shapes and paths
- Method is similar to the explicit Euler method but with a stencil that increases in size based on the time step
- Method is computationally stable and efficient
- Method is easily implemented on GPUs
- Developing software package to closely couple temperature field simulations and microstructure evolution models
  - Three TMS 2025 talks will present on this software package



Acknowledgements and References

This work was supported by the NASA Aeronautics Research Mission Directorate (ARMD) Transformational Tools and Technologies (TTT) project

Contact Information:

Brodan Richter, brodan.m.richter@nasa.gov

Joshua Pribe, joshua.pribe@nasa.gov

Ed Glaessgen, e.h.glaessgen@nasa.gov