Mitigating Manufacturing Variability in Re-Entry Modules

Numerical Analysis Final Project

Saaras Pakanati

University of Cincinnati

December 3, 2025

Contents

1 Background

1.1 Introduction

Many engineering systems are plagued by uncertainty in material properties arising from manufacturing processes or measurement errors. These uncertainties can affect transient heat-transfer predictions and must be quantified.

In this project, I plan to develop a basic framework to address this problem. To this end, I will first analyze the stability of the explicit (Forward Time Central Space, FTCS) and implicit (Backward Time Central Space, BTCS) finite difference schemes for solving the one-dimensional unsteady heat conduction equation with uncertain diffusion coefficients.

Next, I will use the Monte Carlo simulation (MCS) framework to propagate uncertainties in the diffusion coefficient through the numerical solver, generating statistically varying temperature distributions. I will extract key parameters to represent the variation better. I will also benchmark the FTCS and BTCS methods for stability, accuracy, and time efficiency.

A physical application can be used. I considered a spacecraft's reentry heat shield; imagine the SpaceX Dragon capsule reentering Earth's atmosphere. I will model a 1-D domain to represent the heat transfer through the heat shield. For this, I will examine the maximum temperature, the temperature inside the shield, and the time to reach it, parameters you would track in this application.

Objectives

1. Derive and implement FTCS and BTCS finite difference schemes for the transient 1D heat equation.

2. Analyze their stability.

3. Incorporate stochastic variability in material properties using Monte Carlo simulation.

4. Quantify how uncertainty in inputs affects temperature distributions.

2 Problem Statement

2.1 Governing Equation

The physical system is modeled using the one-dimensional transient heat conduction equation:

$$\frac{\partial T(x,t)}{\partial t} = \alpha \frac{\partial^2 T(x,t)}{\partial x^2} \tag{2.1}$$

where $T(x,t)$ is the temperature as a function of position (x) and time (t), and α is the thermal diffusivity.

2.2 Numerical Constraints

The spatial domain is defined as $x \in [0, L]$, where $L = 5$ mm represents the thickness of the heat shield material. The temporal domain is $t \in [0, T]$, where $T = 300$ s or 5 min. The time period and thickness of the heat shield were chosen to simulate real-life conditions.

The boundary conditions are:

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = Q \quad \text{(Heat flux at Outer Surface)} \tag{2.2}$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=L} = 0 \quad \text{(Adiabatic Inner Surface)} \tag{2.3}$$

The initial condition is:

$$T(x,0) = T_0 = 300 \text{ K } \sim 27 \text{ C} \tag{2.4}$$

2.3 Baseline Material Properties

The nominal material properties used in the baseline simulation are:

- Thermal conductivity: $k = 0.5$ W/(m·K)
- Density: $\rho = 300$ kg/m^3

- Specific heat capacity: $c_p = 1000$ J/(kg·K)
- Thermal diffusivity: $\alpha = k/(\rho c_p) = 1.67 \times 10^{-6}$ m^2/s

2.4 Physical Questions

The primary questions addressed in this application are:

1. How do uncertainties in thermal diffusivity affect temperature predictions?
2. What is the probability of thermal failure given realistic manufacturing tolerances?
3. What tolerances on k are required?

3 Methodology

3.1 Discretization of the Heat Equation

3.1.1 Numerical Discretization

The domain is discretized into a uniform grid with spacings Δx and Δt. Let T_i^n denote the approximate temperature at spatial node i and time level n, where:

As such, the condition number is defined as:

$$r = \frac{\alpha \Delta t}{(\Delta x)^2} \tag{3.1}$$

3.1.2 Forward Time, Central Space (FTCS) Method

The FTCS method uses a forward difference for the temporal derivative and a central difference for the spatial derivative:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} \tag{3.2}$$

$$T_i^{n+1} = T_i^n + r(T_{i+1}^n - 2T_i^n + T_{i-1}^n) \tag{3.3}$$

$$T_i^{n+1} = rT_{i+1}^n + (1-2r)2T_i^n + rT_{i-1}^n \tag{3.4}$$

3.1.3 Backward Time, Central Space (BTCS) Method

The BTCS method uses a backward difference for the temporal derivative:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{(\Delta x)^2} \tag{3.5}$$

$$T_i^{n+1} - T_i^n = r(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) \tag{3.6}$$

$$T_i^n = -rT_{i-1}^{n+1} + (1+2r)T_i^{n+1} - rT_{i+1}^{n+1} \tag{3.7}$$

This method is solved by matrix inversion.

3.2 Error Analysis

The local truncation error for the FTCS scheme can be derived via Taylor series expansion. Expanding T_i^{n+1}, T_{i+1}^n, and T_{i-1}^n about the point (x_i, t_n):

$$\tau_i^n = \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = O(\Delta t) + O((\Delta x)^2) \tag{3.8}$$

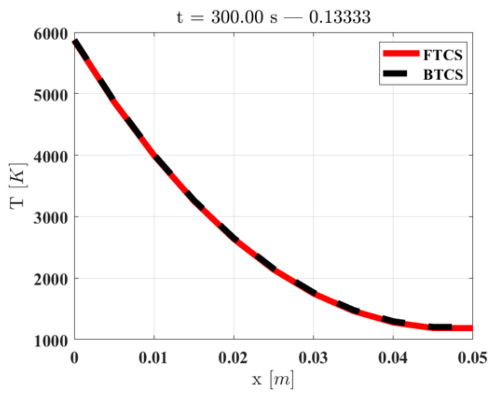
The method is therefore first-order accurate in time and second-order accurate in space. The BTCS scheme has the same truncation error.

However, error propagation in the FTCS method is much greater than in the BTCS method. Due to the internal method MATLAB employs to invert the matrices, the propagation errors are lower.

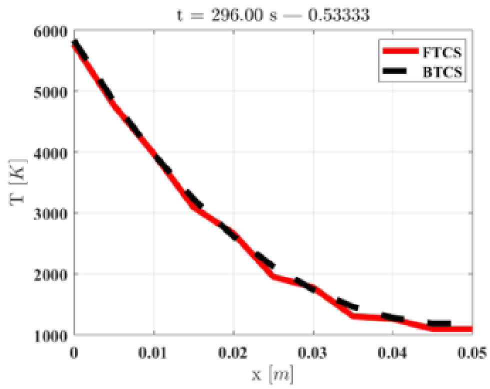
3.3 Stability Analysis

The stability of the numerical models was tested by setting the following condition numbers:

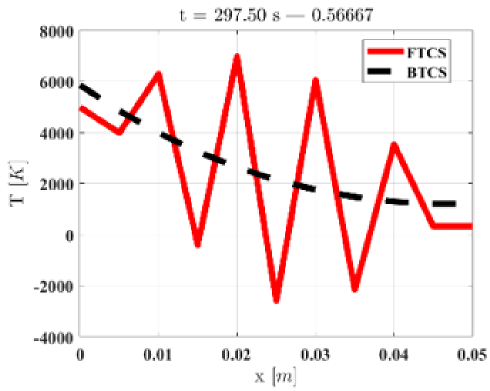
- $r = 0.13$: Stable. FTCS and BTCS match very well.



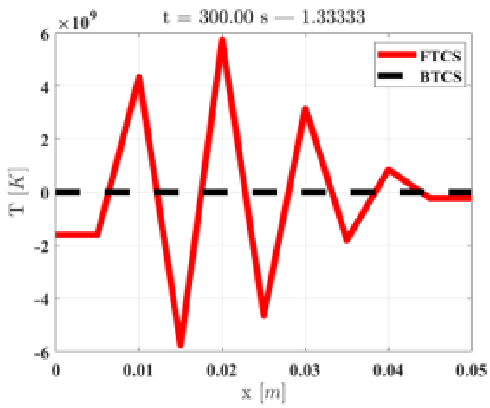
- $r = 0.53$: Slight perturbations seen in FTCS



- $r = 0.56$: Significant perturbations seen in FTCS



- $r = 1.33$: FTCS solution has diverged. BTCS is stable.



Due to its implicit stability, I employed the BTCS method for further analysis.

3.4 Monte Carlo Simulation for Uncertainty Quantification

3.4.1 Methodology

To assess how changing the thermal conductivity would affect the heat shield's performance, the following parameters were used

- Number of simulations: $N = 100,000$
- Thermal conductivity modeled as a normal random variable: $k \sim \mathcal{N}(\mu_k, \sigma_k^2)$
- Other properties (ρ, c_p) held constant
- Numerical method: BTCS

The choice of $N = 100,000$ allows detection of events at the 5-sigma level, corresponding to approximately 2.33 defects per 100,000 samples.

4 Results and Analysis

4.1 Results

4.1.1 Statistical Distribution of Maximum Temperature

Figure 4.1 shows the box plot of maximum temperatures observed across all simulation runs. Figure 4.2 shows the distribution of the maximum temperatures in the system as the thermal diffusivity is varied.

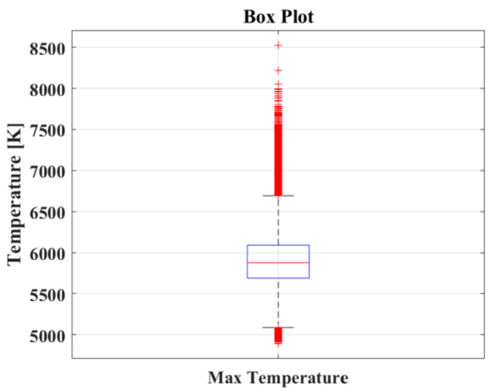


Figure 4.1: Box plot of maximum temperatures from 100,000 Monte Carlo simulations. The mean maximum temperature in the heat shield is approximately 5900 K.

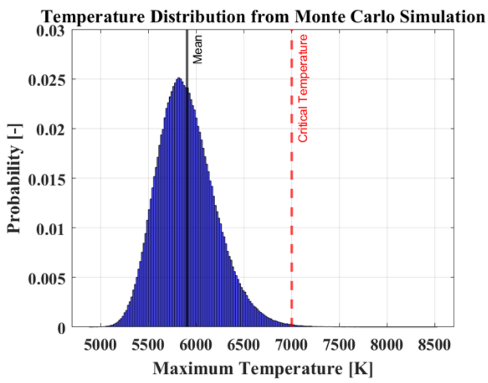


Figure 4.2: Box plot of maximum temperatures from 100,000 Monte Carlo simulations. The mean maximum temperature in the heat shield is approximately 5900 K.

Key observations:

- **Mean maximum temperature**: $\overline{T}_{\max} \approx 5900$ K
- **Distribution shape**: A fairly normal distribution with positive skewness is observed.
- **Spread**: A substantial range of maximum temperatures is observed, ranging over 2000 K!
- **Failures**: A critical temperature of 7000 K was defined, and it can be observed that there are quite a few runs that exceed this temperature.

4.1.2 Probability of Failure

While Figure 4.2 provides a clear visualization of the temperature distribution, as engineers designing such a heat shield, we are more concerned with failure. For this, a Cumulative Density Function (refer Figure 4.3) is used to find the *probability* of failure.

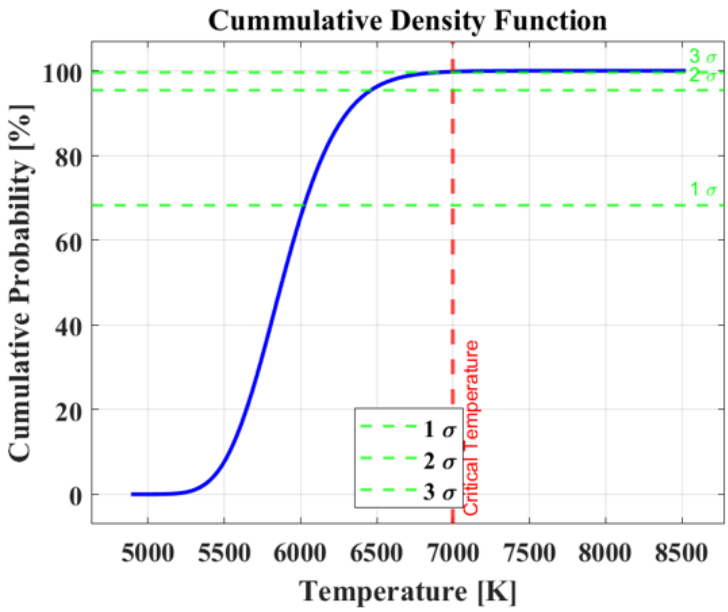


Figure 4.3: Cumulative Probability of Runs

Key observations:

- **Failure**: The process passes 3 σ, however fails 4 σ and higher.
- **Distribution**: The spread of the outcomes is more pronounced, indicating a very spread-out distribution.

4.1.3 Sensitivity to Thermal Conductivity

The Monte Carlo simulations reveal a strong correlation between thermal conductivity and maximum temperature (refer Figure 4.4):

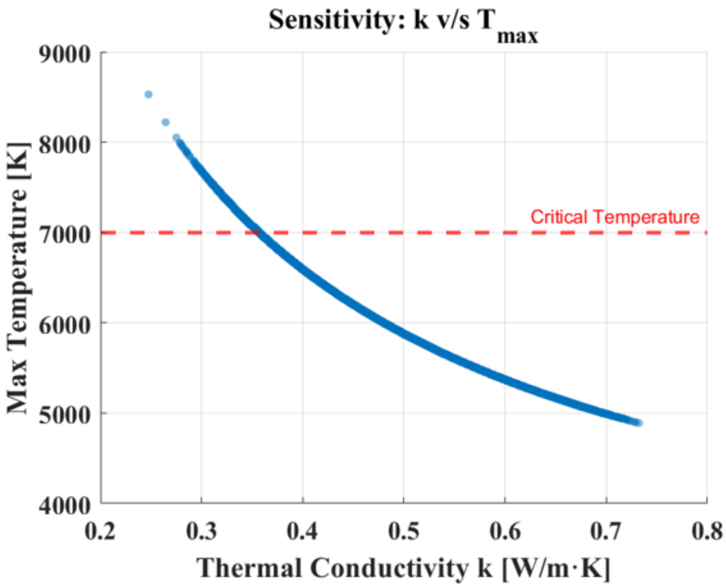


Figure 4.4: A non-linear dependence of Thermal Conductivity on the maximum temperature is theoretically observed.

- k **values**: Lower thermal conductivity inhibit efficient heat transfer. This causes higher temperatures at the outer surfaces.
- **Value of Monte Carlo**: This nonlinear relationship explains why material property uncertainty translates directly to safety risk

4.2 Analysis

4.2.1 Deriving Manufacturing Tolerances

Given the results, an engineering team designing the heat shield can make informed design choices.

For aerospace applications, targeting $P_{\text{target}} = 10^{-6}$ (one in a million failure rate) require:

- Tighter control of manufacturing processes
- More expensive materials or fabrication techniques

In the current analysis, the manufacturing process would need to be **significantly** controlled. The aerospace industry operates 6 σ and higher tolerances on virtually every component. On components like the heat shield, the margin for error would be even lower.

5 Future Work

5.1 Higher-Order Methods

The current implementation uses first-order temporal accuracy. Future work could investigate higher-order time integration schemes, such as:

- Crank-Nicolson Method (Second-order in time and space)
- Runge-Kutta Method (Commonly used ODE45 solver on MATLAB)
- Finite Element Method

5.2 Multi-Dimensional Analysis

The current analysis is one-dimensional. Higher-dimensional analysis will yield more accurate results.

5.3 Advanced Uncertainty Quantification

More complex, physically accurate models than those used in Monte Carlo methods can be employed.

6 Conclusions

This work demonstrated the application of finite difference methods and Monte Carlo simulation to a safety-critical engineering problem. Key findings include:

1. The FTCS method is conditionally stable (roughly, $r \leq 0.5$), while BTCS is unconditionally stable, for the given problem case.

2. Manufacturing variability in thermal properties can lead to failures.

3. Numerical methods provide a quantitative framework for developing manufacturing quality specifications and potentially saving lives!

While the specific use case is limited, it can be expanded to a wide variety of problems. The code is publicly available at,

GitHub repository at: `https://github.com/SaarasPakanati/One-Dimensional-Heat-Transfer-FDM-Code`