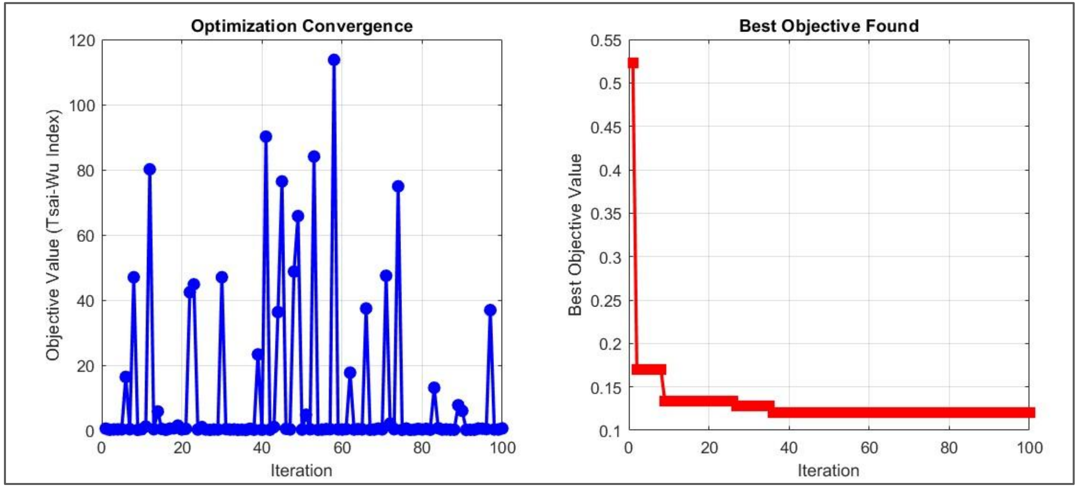
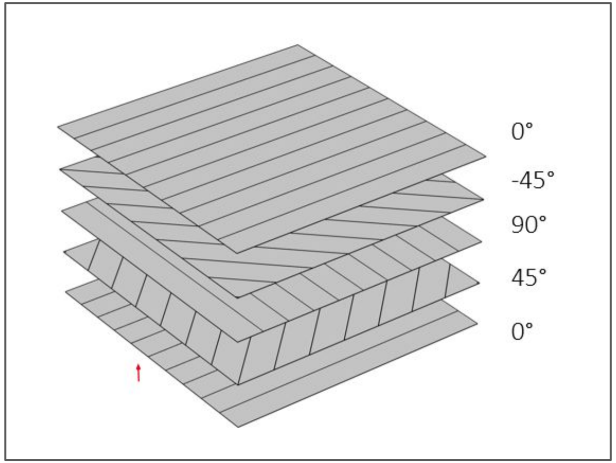
Applying Bayesian Optimization to Laminate Composite Modeling

By Ethan Chen





General Overview + End-to-End Guide

This project looks at how different ply orientations and thicknesses in a carbon-fiber rocket tank respond to combined parachute shock and internal pressure loads. A solver utilizing Classical Laminate Theory and the Tsai-Wu failure criterion evaluates a laminate configuration for failure likelihood. This solver is paired with Bayesian Optimization to run through thousands of configurations to find those with the highest safety margins. Those 'best' ply combinations are then compared to the traditional ±55 laminate to evaluate if machine learning-guided design ultimately produces more robust structures.

In short, a utilization of Bayesian optimization to design a carbon-fiber stack most resistant to failure under shock and pressure loading. The full pipeline: **1) A mechanics solver for stresses and failure indices 2) An objective function that encodes safety and design rules 3) A Bayesian optimizer to search ply angles and thicknesses, and 4) Post-optimization analysis to compare baseline and optimized laminates.**

General Overview + End-to-End Guide

- Inputs: pressure (psi), radius (in), vehicle mass (kg), peak decel a_peak (m/s^2), ply angles (°), ply thicknesses (in)
- Solver: Classical Laminate Theory (A-matrix) + ply stress transforms + Tsai–Wu FI/FS
- Objective: minimize Tsai–Wu failure index (FI) and equivalently, maximize safety factor
- Optimizer: MATLAB bayesopt with Expected Improvement; 4 angles + 4 thicknesses

Workflow

1) Define Problem & Inputs

- Ply angles & thickness bounds
- Tank, load, and material parameters

2) cf_shock_optimized (solver)

- Builds stiffness matrices
- Applies shock and pressure loads
- Computes Tsai-Wu indices

3) cf_shock_demo (objective function)

- Minimize Tsai-Wu failure index
- Add penalties (safety, thickness variation)
- Return scalar score

4) cf_shock_bayesopt (Bayesian optimizer)

- Evaluates using solver
- Proposes laminate designs
- Configuration exploration/exploitation

5) Optimization Results

- Finds best ply configuration
- Convergence history
- Safety factor improvements

6) Post-Processing & ML Prep

- Exports evaluated designs
- Builds surrogate dataset
- Enables fast future predictions

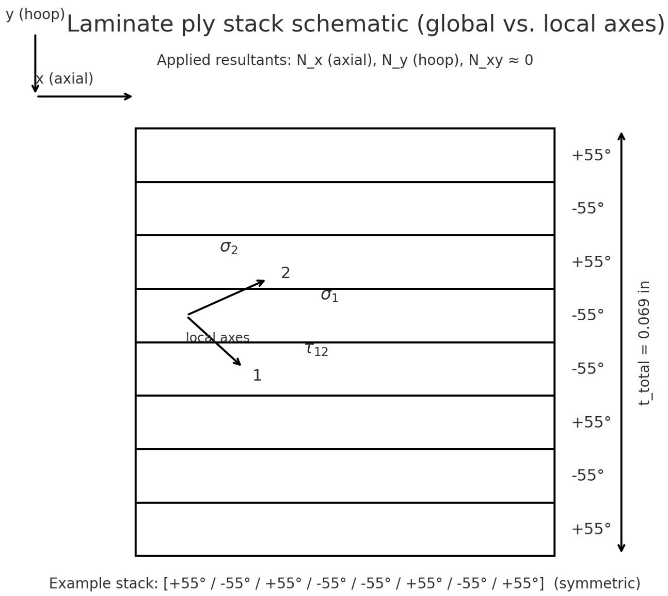
CF_Shock_Optimized

CF_Shock_Optimized — The Core Solver

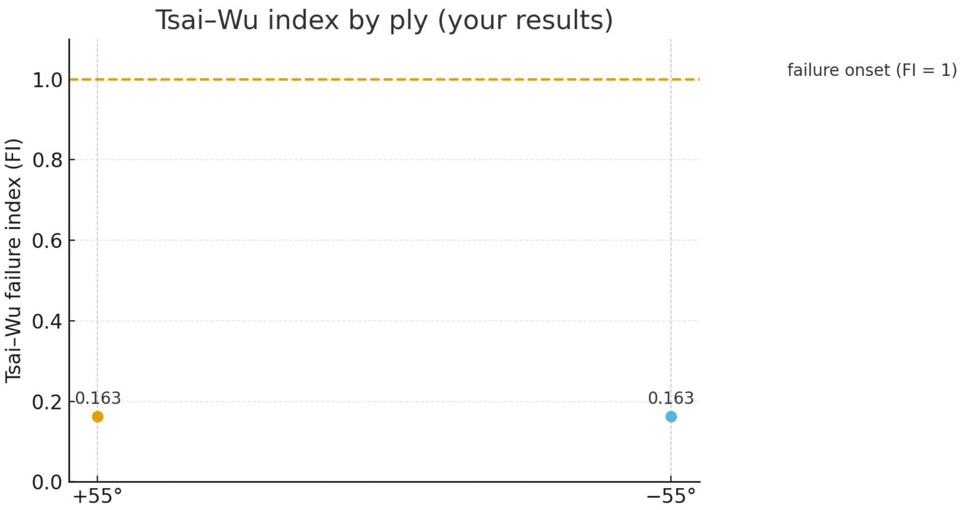
Ply orientations and thicknesses are taken as inputs, along with additional tank parameters, to run a full laminate stress analysis. It outputs Tsai-Wu failure criterion results such as failure margin, critical ply, and safety factor. Ultimately, this creates the generalized function that is the evaluation engine for the other scripts. Please see sister project *Parachute Shock Loads on Carbon Fiber Tanks: An End-to-End Guide* for a more comprehensive overview of laminate safety evaluation using Tsai-Wu failure criterion.

- Transforms loads to laminate strains and local ply stresses
- Builds Q reduced stiffness matrix

Visualization of Ply Stack Schematic & Tsai-Wu Failure Criteria



Laminate ply stack schematic
[+55/−55/+55/−55/−55/+55/−55/+55 stack] with applied resultants (N_x, N_y, N_xy)



Data visualization displaying Tsai-Wu values for +55 and -55 plies and failure onset barrier

Inputs & Assumptions

- At minimum, CF_Shock_Optimized needs:
  - ply_angles: vector of ply orientations in degrees e.g. [55, -55, 0, 90]
  - thicknesses: vector of ply thicknesses in inches (same length as ply_angles)
  - These can be inputted in CF_Shock_Demo
- Other parameters can also be optionally adjusted in CF_Shock_Optimized:
  - ‘material’
  - ‘pressure’
  - ‘tank_radius’
  - ‘vehicle_mass’
  - ‘Tau_inflate’
  - ‘a_peak_tau’
  - ‘Delta_V’
  - ‘verbose’

CF_Shock_Demo

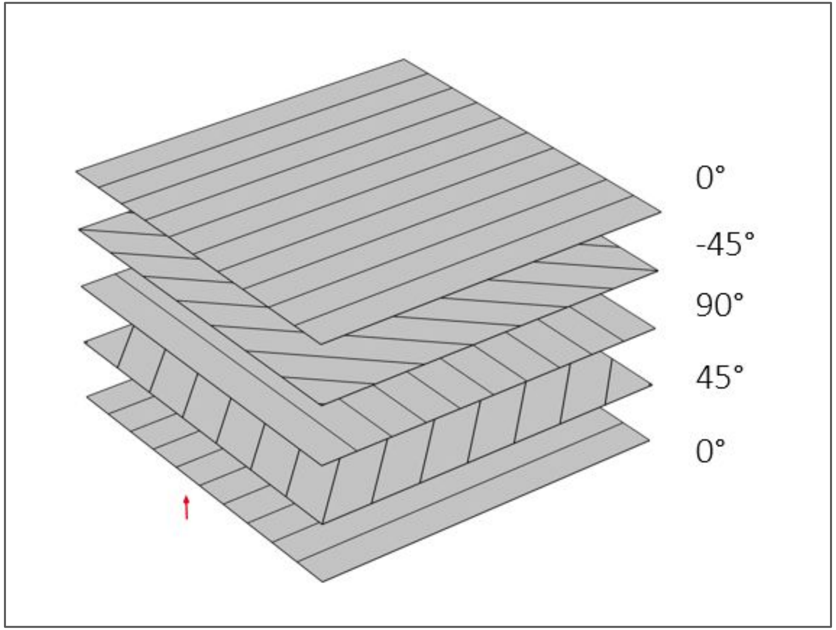
CF_Shock_Demo — Experimentation, Objective Function Creation

Tests driver and sanity check for solver. Its main purposes are to take the generalized function from CF_Shock_Optimized to create the basis for an objective function that is passed to the optimizer, and displays intuition on how various layup choices can affect performance before Bayesian optimization.

- Manually calls CF_Shock_Optimized on different inputted laminate configurations (±55°, 0/90, ±45) to “demo” Tsai-Wu performance
- Runs parameter sweeps to plot Tsai-Wu failure index/safety factor variance across angles
- Objective function penalties: angle extremities, thickness variation, safety factor < 1,5

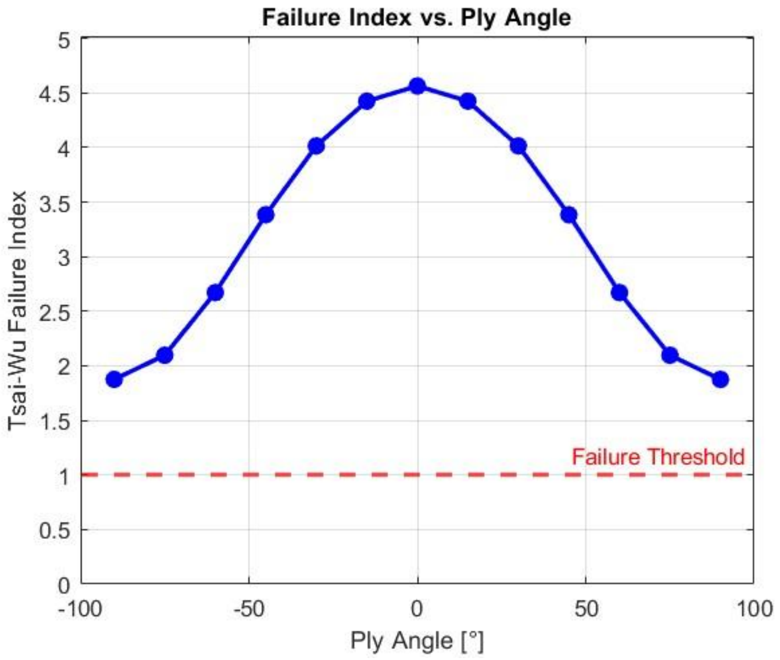
Default 4-Ply Stack Configs

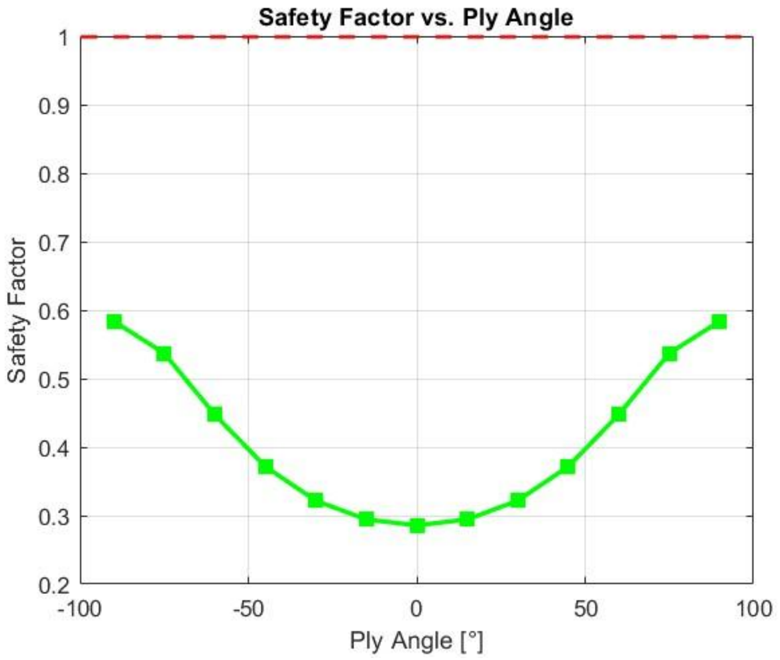
- [0,90,0,90]: fibers aligned with axial/hoop directions → good for pressure, weak in shear
- [45,−45,45,−45]: shear-oriented → good for in-plane shear, different pressure response
- [55,−55,55,−55]: typical filament-wound bias
- Mixed set (e.g., [0,45,90,−45]): quasi-ish balance to probe coupling
- [30,−30,60,−60]: off-axis variant to sample the space between ±45 and ±55



Visualization of ply stack configurations at various angles within a laminate. Courtesy of Comsol

CF_Shock_Demo — Ply Angle Sweeps





Plots of failure index and safety factor values for individual plies with angles varying from -90 to 90. Failure Index values over 1 indicate failure, and Safety Factor values under 1 indicate failure (chart for a singular ply, not whole laminate)

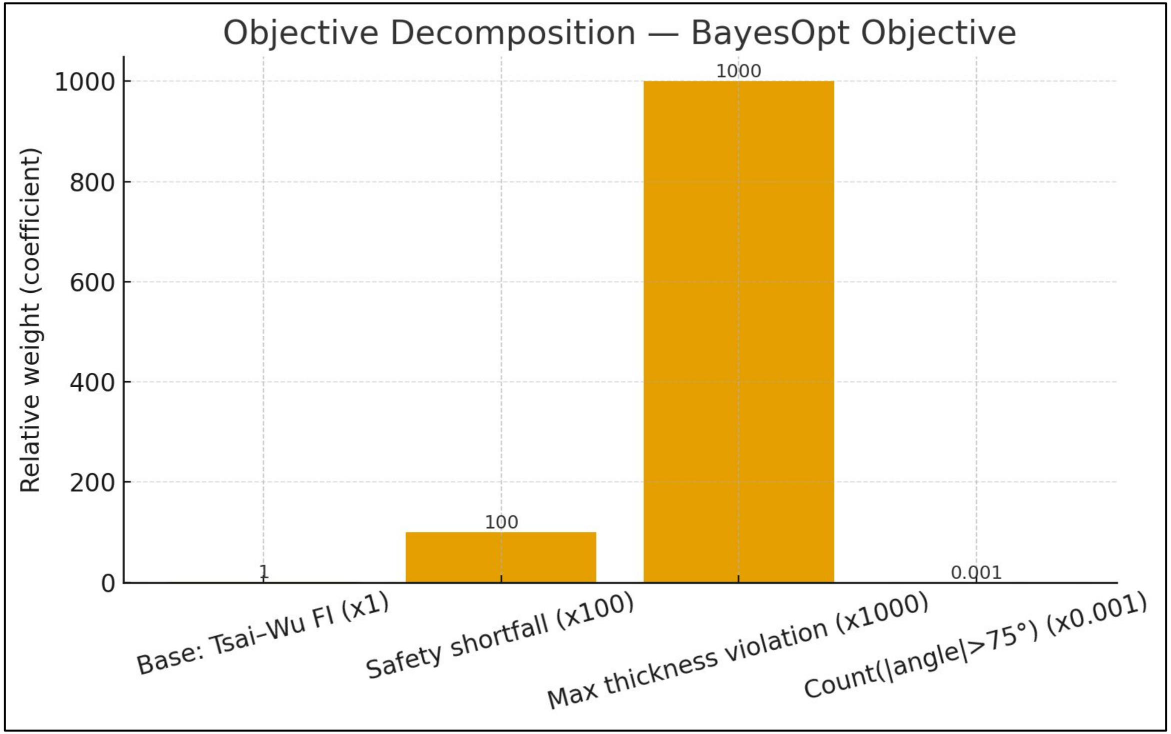
Objective Function Decomposition

- Objective Function: encoding engineering & design priorities into a single metric
- The optimizer needs a single numerical score to compare different designs. The objective function achieves this, starting with the failure index as the base term, and then adding penalty terms for designs that are unsafe and/or unbalanced
- This setup lets the optimizer search the design space intelligently: “good” laminates (low failure index, high safety factor, feasible layups) yield low objective values, while “bad” laminates accumulate penalties and rank poorly

$$J_{\text{demo}} = \underbrace{\text{FI}}_{\text{Tsai–Wu}} + \underbrace{10\left[\max(0,\ 1.5 - \text{SF})\right]^2}_{\text{safety shortfall}} + \underbrace{0.1\ \text{std}(t)}_{\text{thickness variation}} + \underbrace{0.01\ \text{mean}(|\theta|)}_{\text{angle cost}} + \underbrace{M\,\mathbf{1}(n_{\text{valid}} < 2)}_{\text{infeasible (too few plies)}}$$

“Demo” objective with arbitrary weights on penalties. Fine-tune penalties using resultant data distribution and engineering inference for Bayesian Optimization later–the demo is a “sandbox” for objective function adjustment

Objective Function Decomposition



Multipliers of penalty terms for BayesOpt objective (actual optimization penalties) showing how strongly each factor affects the optimizer

Console Print Out

```
Laminate Properties:
E_x: 1556930 psi, E_y: 4673390 psi
nu_xy: 0.447, G_xy: 5485684 psi
Total Thickness: 0.069 in

Loading Conditions:
N_x: 1474.7 lb/in, N_y: 2911.5 lb/in
Peak Shock Force: 462 lb

Original Configuration Results:
Tsai-Wu Index: 0.1639
Safety Factor: 3.46

=== Parameter Sweep Analysis ===

=== Multi-Ply Optimization Example ===
Configuration       Tsai-Wu Safety Factor   Critical Ply
--------------------------------------------------------
[0 90 0 90]             0.2840  2.87        1 (0.0°)
[45 -45 45 -45]         0.4640  1.57        2 (-45.0°)
[30 -30 60 -60]         0.3570  2.37        2 (-30.0°)
[0 45 90 -45]           0.3679  2.45        1 (0.0°)
[55 -55 55 -55]         0.1671  3.45        2 (-55.0°)

=== Preparing for Bayesian Optimization ===
Objective function created for 8 variables
Variable bounds: Angles [-90°, 90°], Thicknesses [0.005, 0.050] in

=== Testing Objective Function ===
Test input: [55 -55 0 0 0.0345 0.0345 0 0]
Objective value: 0.713930
Test input: [45 -45 45 -45 0.017 0.017 0.017 0.017]
Objective value: 0.922274
>>
```

CF_Shock_BayesOpt

Bayesian Optimization Engine

Sets up angle and thickness variables, defines their bounds, and creates a finalized objective function that wraps around CF_Shock_Optimized. Then, it calls MATLAB's bayesopt function to automatically explore the design space. It outputs the best laminate found, plots convergence history, and saves the optimization results for later analysis.

- Performance Metrics Collected: Objective value, safety factor, critical ply and angle, convergence history (objective trace), distribution of tested designs across angle/thickness space
- Optimization space: 8 continuous variables
- Gaussian Process Regression (default in MATLAB bayesopt for ≤20 variables)
- Provides both mean prediction of objective value and uncertainty estimate

Why Bayesian Optimization?

By nature, laminate design has a nonlinear, high-dimensional search space where each candidate ply configuration requires running an expensive structural simulation. The sample-efficiency of Bayesian optimization makes it most fitting, as in this case it uses a Gaussian Process surrogate model to predict the objective across the entire design space, and can test new, uncertain regions while refining known effective designs. This allows it to reduce computational cost and converge toward optimal ply orientations and thicknesses with far fewer evaluations than random or exhaustive methods.

Why Not Gradient-Based Methods?

Gradient-based methods only work when the objective function is smooth and differentiable. However, this objective function will have breaks in continuity (plies with 0 thickness will be discarded) and penalty weights (safety checks, consistency of ply thickness) will make the landscape non-smooth and noisy.

Why Bayesian Optimization?

Why Not Traditional Methods?

Traditional shock calculation workflows calls for comprehensive, brute force searches over multiple ply angles and thicknesses. Each unique ply configuration needs a solver that requires a wide range of inputs (drag coefficient, internal pressure, speed at parachute deployment, etc), which then runs through a variety of programs, from stiffness assembly to manual Tsai-Wu checks. This traditional workflow is very computationally costly and takes time.

$$J_{\mathrm{BO}} = \underbrace{\mathrm{FI}}_{\text{Tsai–Wu}} + \underbrace{100\left[\max(0,\, 1.5 - \mathrm{SF})\right]^2}_{\text{safety shortfall}} + \underbrace{1000\,\mathbf{1}\big(\max(t) > 0.050\big)}_{\text{thickness violation}} + \underbrace{0.001\sum_i \mathbf{1}\big(|\theta_i| > 75^\circ\big)}_{\text{extreme-angle count}} + \underbrace{M\,\mathbf{1}(\text{analysis fails})}_{\text{fallback penalty}}$$

Breakdown of actual BayesOpt objective function used to wrap CF_Shock_Optimized. Without Bayesian optimization, mass configuration calculations threaten to become computationally expensive without objective prediction!

Bayesian Optimizer Diagram

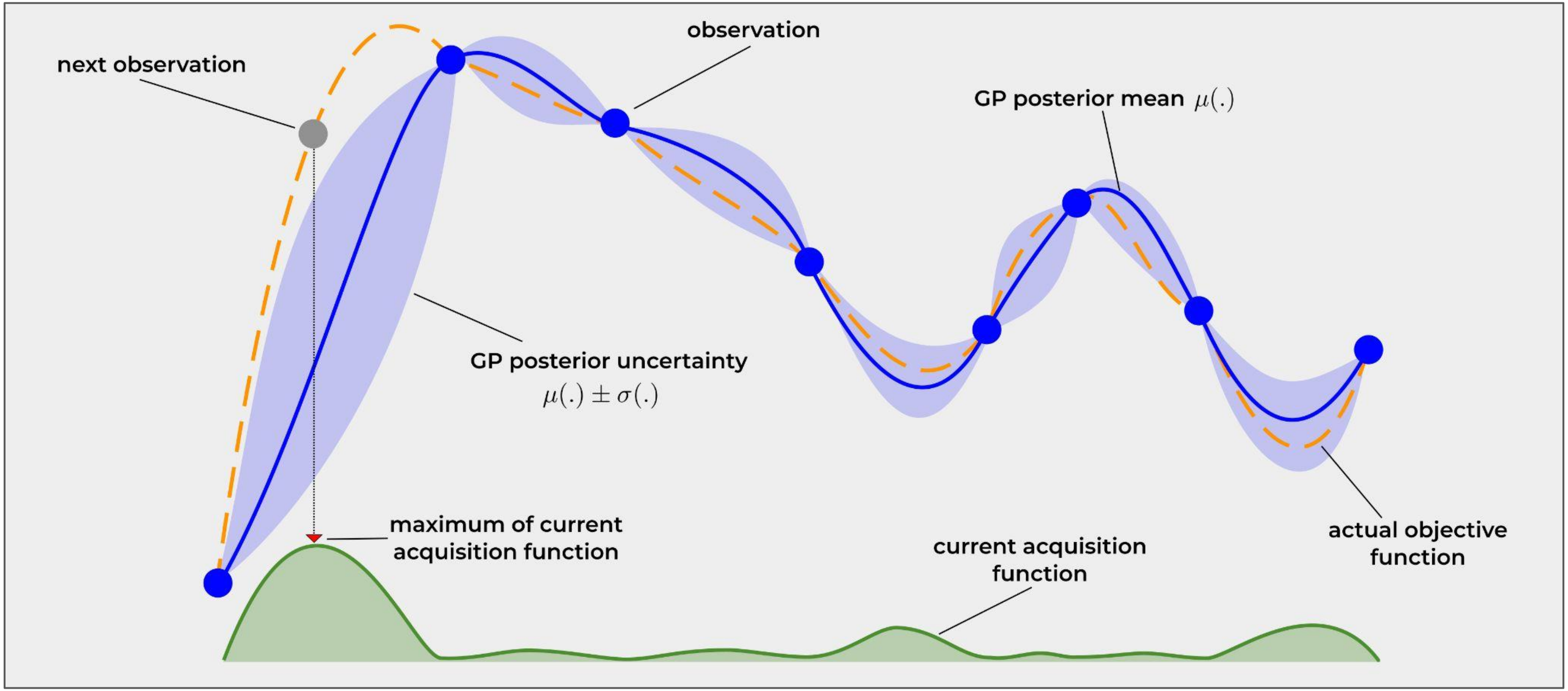


Diagram of a general Bayesian optimization model. Courtesy of Firas Al-Hafez

Orange Dashed: The objective function

Blue Dots: Laminate designs already evaluated as landmarks, with FI and safety factor established for a specific set of angles/thicknesses

Blue Solid: Surrogate’s predicted FI for any design (in picture it’s 1-D while for tested composites it’s 8-D, but conceptually identical)

Blue Band: how unsure the surrogate is. Wide where there are few designs; narrow near tested designs

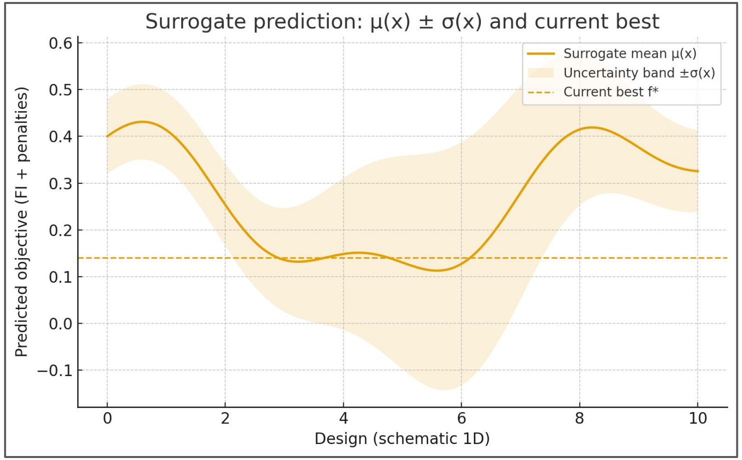
Green graph: Acquisition function, where to sample next, peaks where the model predicts low FI or where uncertainty is high

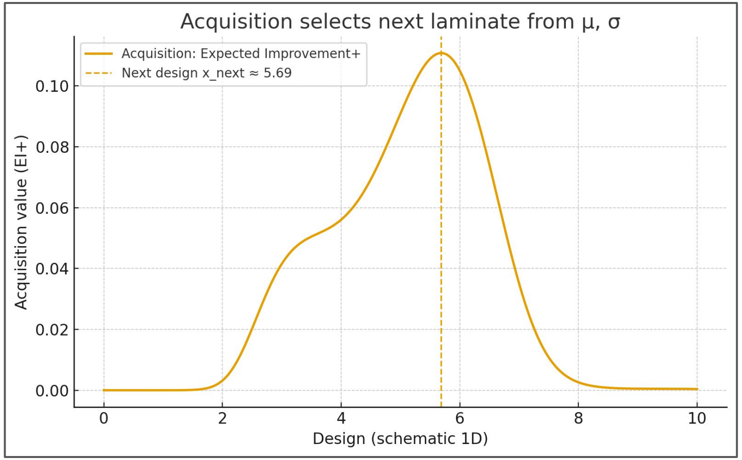
Green peak + red arrow: Acquisition peak: the proposed next laminate. In the loop this becomes the gray “next observation”

Convergence behavior: as you add observations, the blue band shrinks, green peaks flatten, and the best-so-far FI trace steps downward

Gaussian Process Integration

The surrogate model and Bayesian optimization run as a tight loop. The surrogate model, in this case a Gaussian Process (GP), fits past laminate design evaluations and returns a prediction and uncertainty value. The optimizer then uses an acquisition function built from these prediction and uncertainty values to pick the next design that trades off exploration and exploitation. That design is evaluated with the real solver and the surrogate is refitted until the objective plateaus at convergence.





Left and right: visual depictions of surrogate prediction and acquisition processes

BayesOpt + GP Integration Workflow

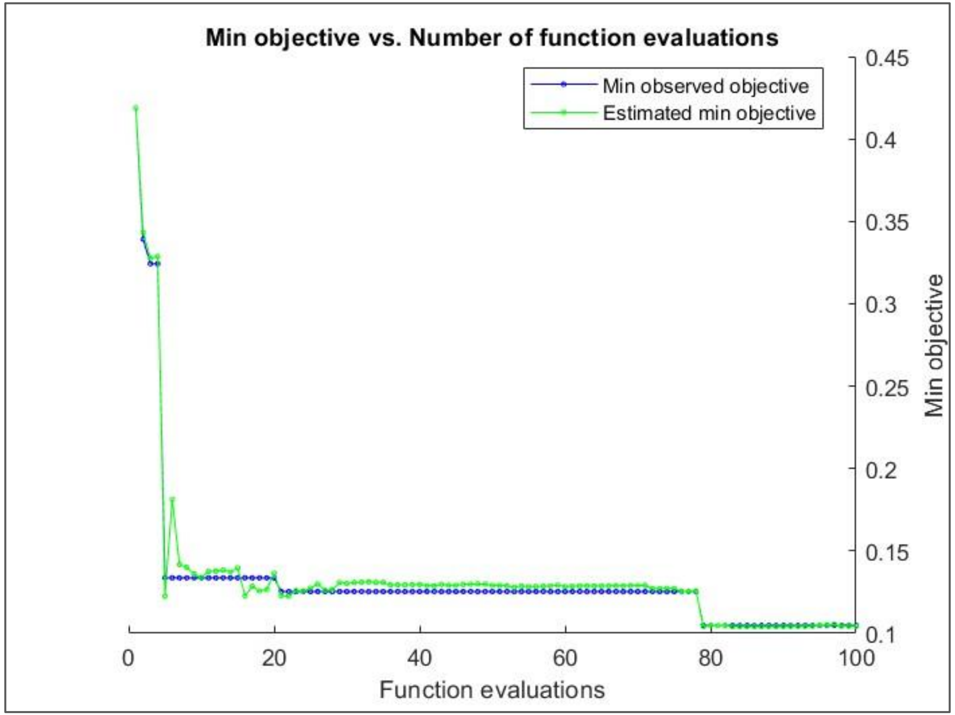
1) The GP is first trained on evaluated designs to fit the surrogate. Here angles & thicknesses and resultant Tsai–Wu FI & penalty values are used. The GP gives a smooth mean and calibrated uncertainty over untried designs.

2) Candidates are scored by computing an acquisition (e.g., Expected Improvement) from specific configuration prediction and uncertainty values. It scores designs high if they look promising or uncertain

3) The optimizer selects the next design by selecting the configuration with the highest acquisition value.

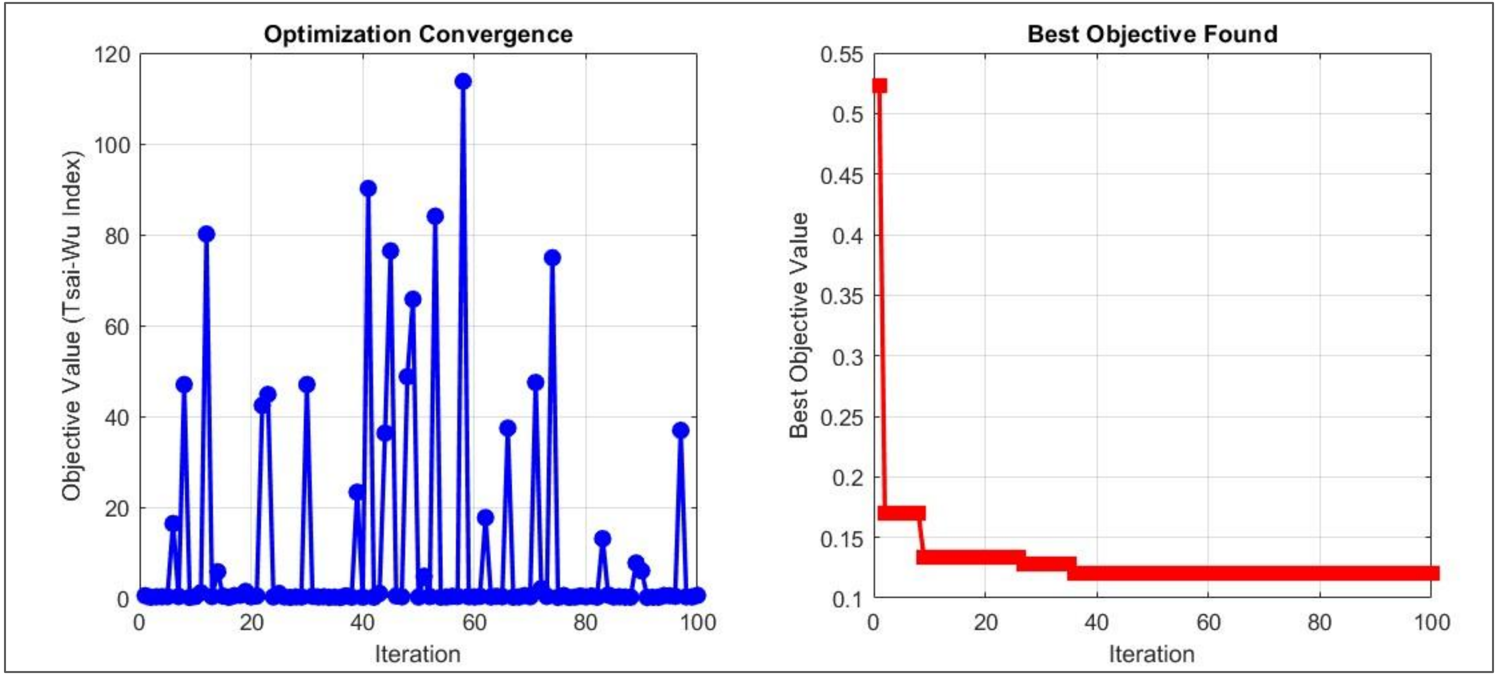
4) Evaluate & update: Run Cf_Shock_Optimized to get the true FI/SF, append Tsai-Wu and penalty values and refit the GP. These steps are iterated until convergence.

Bayesian Optimizer Estimation



Bayesian optimizer’s estimate of the minimum objective. Rapid improvement in the first ~10 evaluations (big drop from ~0.4 to ~0.15). After ~20 evaluations, progress slows—most designs stay near the same best range. Around evaluation ~80, another improvement brings the best objective down near 0.1–0.12.

Optimization Convergence & Best Objective Found



Left: Graphical representation of optimization convergence. As optimization proceeds, points cluster lower (exploitation), showing that the algorithm is finding safer, stronger laminate designs. Right: Calculated best solution per iteration. Lower objective value–more structurally robust configuration

Console Print Out

```
Best observed feasible point:
    Angle1    Angle2    Angle3     Angle4     Thickness1    Thickness2    Thickness3    Thickness4
    ______    ______    _______    _______    __________    __________    __________    __________

    54.88     64.005    -46.684    -37.003     0.031667      0.047427      0.021682      0.034681

Observed objective function value = 0.085037
Estimated objective function value = 0.08537
Function evaluation time = 0.0008301

Best estimated feasible point (according to models):
    Angle1    Angle2     Angle3     Angle4    Thickness1    Thickness2    Thickness3    Thickness4
    ______    _______    _______    ______    __________    __________    __________    __________

    44.383    -55.984    -70.659    49.967     0.037053      0.04993       0.010227      0.039006

Estimated objective function value = 0.08415
Estimated function evaluation time = 0.00084083


=== Optimization Complete ===
Best objective value: 0.085037
Best point found:
  Angle1: 54.9°
  Angle2: 64.0°
  Angle3: -46.7°
  Angle4: -37.0°
  Thickness1: 0.0317 in
  Thickness2: 0.0474 in
  Thickness3: 0.0217 in
  Thickness4: 0.0347 in
```

```
Analyzing best design...

=== CF_Shock Analysis Results ===
Critical Tsai-Wu Index: 0.0850
Safety Factor: 5.97
Failure Margin: 0.9150
Critical Ply: 1 (54.9°)

Laminate Properties:
E_x: 2297468 psi, E_y: 4349272 psi
nu_xy: 0.543, G_xy: 3312427 psi
Total Thickness: 0.135 in

Loading Conditions:
N_x: 1474.7 lb/in, N_y: 2911.5 lb/in
Peak Shock Force: 462 lb

=== Post-Optimization Analysis ===

Comparison with Original Design:
Original (+55/-55): Tsai-Wu = 0.6624
Optimized: Tsai-Wu = 0.0850, Safety Factor = 5.97
Improvement: 87.2% reduction in failure index

Results saved to optimization_results.mat
>>
```

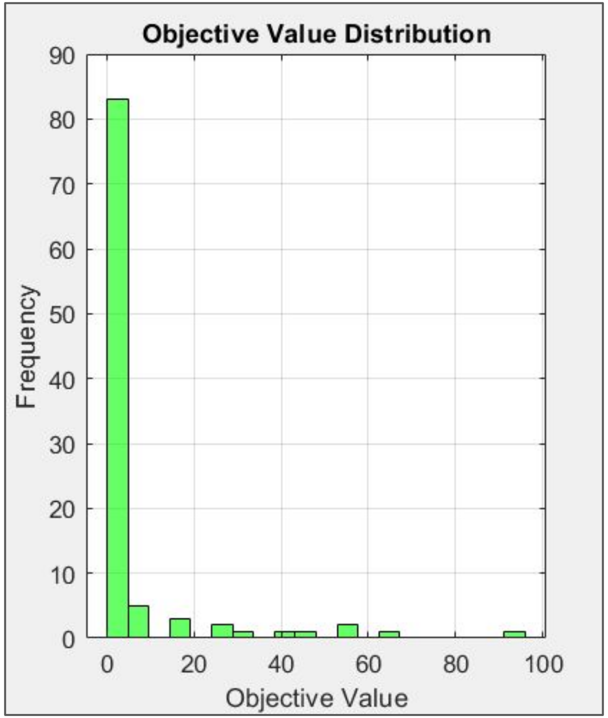
CF_Shock_Optimization_Summary

Post-Processing & Data Preparation

After Bayesian optimization runs, it loads the results, summarizes performance (best objective, convergence rate, evaluation time), and analyzes the range of angles and thicknesses explored. It plots convergence graphs and histograms, reports the best design found, and exports all evaluated designs into CSV/Mat files. It also prepares training data for future surrogate machine learning models, suggesting options like Gaussian Processes or Random Forests. Essentially, this script closes the loop: it validates optimization outcomes, packages the data, and points to the next step of using ML surrogates for even faster design exploration.

- Dataset size & dimensionality: ~8D input space (angles + thicknesses)
- Exported training data: cf_shock_training_data.csv, best_designs.csv
- Targets: $R^2 > 0.9$, MAE < 0.01, <1 ms prediction time

Bayesian Optimizer Estimation



Data visualization of the distribution of objective values shows that most laminate designs evaluated during optimization had low Tsai-Wu failure indices, clustering below 5

Console Print Out

```
=== Optimization Performance ===
Total evaluations: 100
Best objective value: 0.085037
Optimization time: 20.93 seconds
Average evaluation time: 0.2093 seconds

=== Design Space Exploration ===
Angle ranges explored:
  Angle1: [-90.0°, 89.9°] (mean: -1.3°)
  Angle2: [-90.0°, 89.9°] (mean: -25.0°)
  Angle3: [-90.0°, 90.0°] (mean: -4.2°)
  Angle4: [-89.9°, 89.9°] (mean: 1.4°)
Thickness ranges explored:
  Thickness1: [0.0065, 0.0500] in (mean: 0.0285 in)
  Thickness2: [0.0053, 0.0500] in (mean: 0.0288 in)
  Thickness3: [0.0062, 0.0498] in (mean: 0.0279 in)
  Thickness4: [0.0050, 0.0490] in (mean: 0.0267 in)

=== Convergence Analysis ===
Initial objective: 0.303150
Final objective: 0.380359
Best objective: 0.085037
Improvement: 71.9%
Convergence rate: 0.002181 per iteration
```

```
=== Best Design Analysis ===
Best configuration found:
  Angles: [54.9°, 64.0°, -46.7°, -37.0°]
  Thicknesses: [0.0317, 0.0474, 0.0217, 0.0347] in
  Total thickness: 0.1355 in

=== Preparing ML Training Data ===
Training data saved to ml_training_data.mat
Training set size: 100 samples × 8 features

=== Data Quality Assessment ===
Unique designs: 100 / 100 (100.0%)
Objective value range: [0.085037, 94.487333]
Objective value std: 15.416868
High penalty evaluations (>10): 12 (12.0%)
Good designs (TW ≤ 0.2): 23 (23.0%)
Excellent designs (TW ≤ 0.15): 12 (12.0%)
🎯 Likely found near-global optimum (TW < 0.12)

=== Exporting Data ===
Data exported to cf_shock_training_data.csv
Best designs (TW ≤ 0.2) exported to best_designs.csv
>>
```

How to Run — Step by Step

1) Open MATLAB in the folder with CF_Shock_Optimized.m, CF_Shock_Demo.m, and CF_Shock_BayesOpt.m

2) Run CF_Shock_Demo.m: tests refactored solver, runs parameter sweeps, and prepares the objective function

3) Adjust optimization settings in CF_Shock_BayesOpt.m (number of iterations, acquisition function, parallelization)

4) Run CF_Shock_BayesOpt.m: calls bayesopt, evaluates ply designs, logs Tsai-Wu failure index, safety factors, and returns best result

5) Save and review results in optimization_results.mat

6) Run CF_Shock_Optimization_Summary.m: loads results, generates convergence plots, distribution charts, and prepares ML training data

Interpreting Outputs

- Best Objective Value: minimum Tsai–Wu failure index found across all laminate designs.
- Safety Factor: lowest reserve factor among plies; must remain > 1 for structural feasibility.
- Optimization Convergence: tracks objective value reduction over iterations to show learning progress.
- Design Space Coverage: distribution of objective values reveals how many poor vs. good laminates were explored.
- Best Design Variables: ply angles and thicknesses corresponding to the best laminate found.
- Training Dataset: full set of evaluated designs exported for machine learning surrogate modeling.

Results Based On Current Data

Best design achieved Tsai–Wu Failure Index of 0.085 and Safety Factor of 5.97. This marks an **87% improvement in structural performance** (Tsai-Wu failure index: going from 0.662 to 0.085) and **172% improvement in safety factor** from 3.47 to 5.97 through optimization

Design space exploration confirms most laminates are poor performers (FI >> 1), but optimization effectively narrowed into the feasible zone

Surrogate dataset of 100 evaluated laminates is now available for training ML models to accelerate future searches