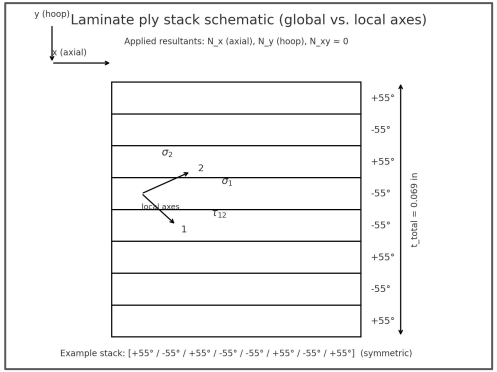
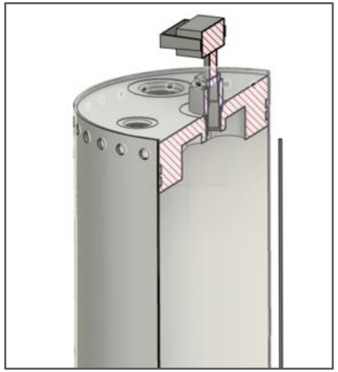
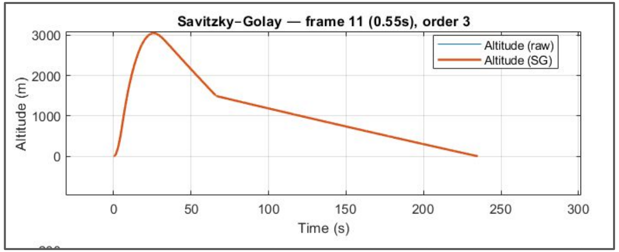
Parachute Shock Loading Analysis on Spaceshot Carbon Fiber Tanks: A Walkthrough

By Ethan Chen







General Overview + End-to-End Guide

This project looks at how the sudden shock from a parachute opening affects a rocket’s fuel tank. Flight data is analyzed to find the sharp slowdown of parachute deployment and measure the length of that force. That length is then compared to known ranges from parachute testing to check if the numbers are feasible. This parachute load is finally combined with the tank’s internal pressure to see if the carbon fiber structure can safely withstand the stress.

In plain terms, the project analyzes how hard a parachute “yanks” on a rocket during deployment, and whether a carbon fiber propellant tank can survive that sudden load. **A full pipeline from 1) raw flight data, 2) parachute shock load modeling 3) structural stress analysis 4) empirical validation check**

General Overview + End-to-End Guide

- Computes peak opening shock loads from parachute deployment and maps to carbon fiber shell stresses on MATLAB
- Consists of 3 separate files: Tau_Inflate, CF_shock, and Flight_Data_Sav_Golay (flight data graphing)
- What you get: F_peak, N_x_shock, global stresses, ply stresses, Tsai–Wu FS
- How to use: run tau_inflate.m (data) → run cf_shock.m (loads) → read the report & sanity checks.

Workflow

1) Input Data (Flight Logs + Vehicle/Parachute Properties)

- Raw CSV flight data
- Parachute, tank, & vehicle parameters

2) tau_inflate.m (Preprocessing)

- Resample/ smooth data
- Compute velocity and acceleration
- Restrict dataset

3) tau_inflate.m (Event Detection & Analysis)

- Find cumulative impulse
- Derive tau_inflate from 5–95% impulse duration
- Model peak decel

4) cf_shock.m (Peak Load Calculation)

- Convert forces to strain
- Distribute F_peak circumferentially
- Add internal pressure resultants

5) cf_shock.m (Composite Stress Analysis)

- Build stiffness matrices
- Solve for global strains & local ply stresses
- Evaluate failure index (Tsai–Wu)

6) Output & Sanity Checks

- Report stresses and safety factors
- Evaluate sanity checks
- Flag if tank is at risk of failure under parachute shock

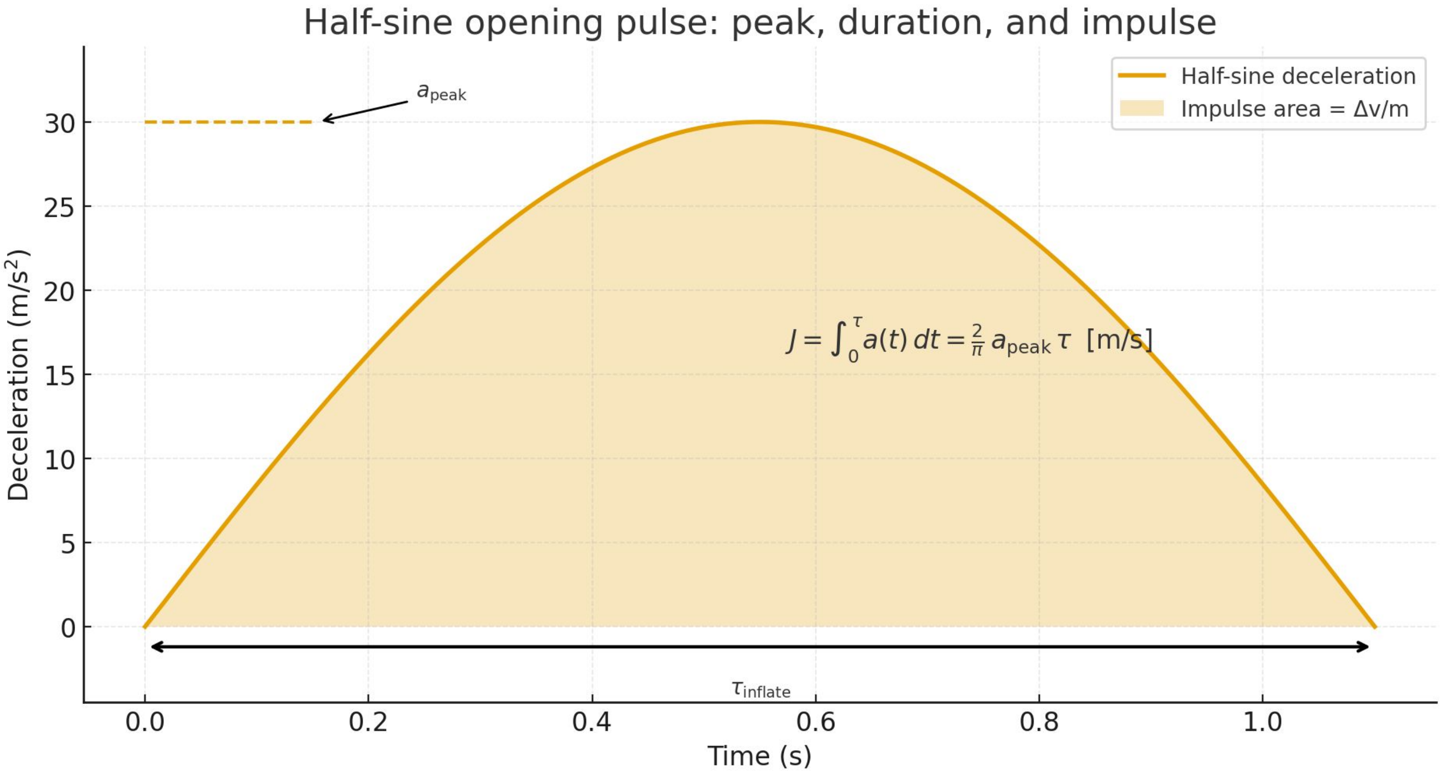
Tau Inflate

What Is Tau_Inflate?

When a parachute deploys, the stress does not just jump instantly to its maximum. Instead, it ramps up and down in a pulse. Tau_inflate is the time length of that pulse, describing how long the parachute takes to inflate effectively. This is critical because it links the parachute's opening behavior to the forces felt by the rocket.

- During deployment the force on the vehicle rises from near zero to a peak, then decays as the canopy inflates and settles into steady drag–**tau_inflate is the length of that half-sine, representing the effective inflation time**
- With velocity and parachute size, tau_inflate can be used to predict the maximum force exerted by the parachute

What Is Tau_Inflate?



Data visualization of tau-inflate as the time duration of a half-sine deceleration window

Required Inputs (Parachute & Vehicle)

- Stage: drogue or main
- Parachute coefficients: Cd, reference surface area A_full
- Vehicle mass m; air density ρ at deployment altitude.
- Tank: thickness t (in), radius R (in), internal pressure p (psi).

Robust File Reading

- Handles lines starting with '#'; recovers a commented header automatically.
- Falls back to likely column name matching if exact headers missing.
- Drops NaNs; sorts by time; asserts enough samples for processing.

Required Inputs (Data)

- Flight log CSV with columns: Time (s), Altitude (m). Header can be commented with '#'. Units must be in SI for tau_inflate
- Sampling can be uneven; script resamples to uniform grid.
- Formatting example provided below

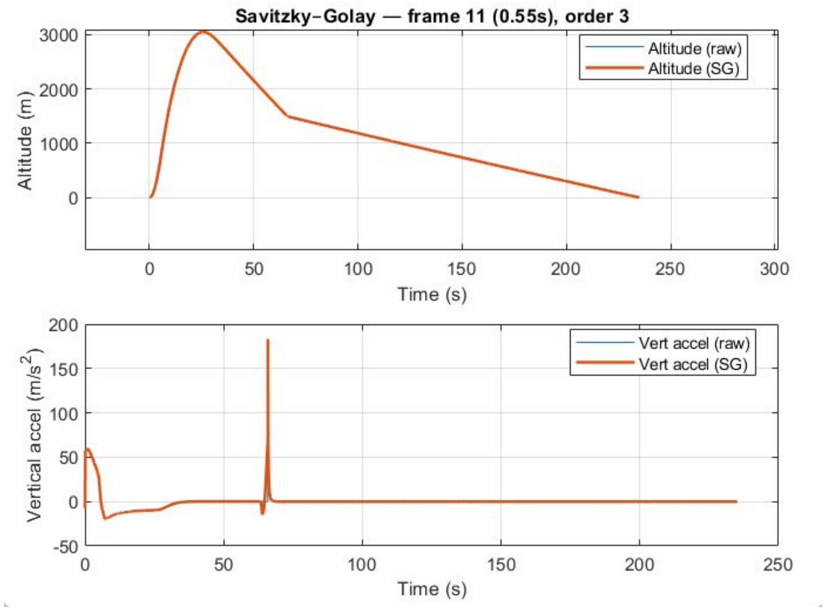
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	# Simulation 1 (Up to date)													
2	# 1183 data points written for 14 variables.													
3	# Simulation warnings:													
4	# Recovery device deployment at high speed (40.5 m/s)													
5	#													
6	# Time (s)	Altitude (m)	Vertical velocity	Vertical accelera	Total velocity (m	Total acceleratio	Position East of	Position North of	Lateral distance	Roll rate (°/s)	Pitch rate (°/s)	Yaw rate (°/s)	Mass (g)	Motor mass (g)
7	0.01	-3.77E-04	-0.075	-7.534	0.075	7.534	0	0	0	0	0	0	4.20E+04	1.23E+04
8	0.02	-1.51E-04	-0.03	-3.009	0.03	3.009	0	0	0	0	0	0	4.20E+04	1.23E+04
9	0.03	7.58E-05	0.015	1.518	0.015	1.518	0	0	0	0	0	0	4.20E+04	1.23E+04
10	0.04	5.30E-04	0.076	6.046	0.076	6.046	0	0	0	0	0	0	4.20E+04	1.23E+04
11	0.05	0.002	0.181	10.576	0.181	10.576	0	0	0	0	0	0	4.20E+04	1.22E+04
12	0.06	0.004	0.332	15.108	0.332	15.108	0	0	0	0	0	0	4.20E+04	1.22E+04
13	0.07	0.009	0.529	19.642	0.529	19.642	0	0	0	0	0	0	4.20E+04	1.22E+04
14	0.08	0.015	0.771	24.177	0.771	24.177	0	0	0	0	0	0	4.20E+04	1.22E+04
15	0.09	0.024	1.058	28.714	1.058	28.714	0	0	0	0	0	0	4.20E+04	1.22E+04
16	0.1	0.037	1.39	33.253	1.39	33.253	0	0	0	0	0	0	4.20E+04	1.22E+04
17	0.11	0.052	1.768	37.793	1.768	37.793	0	0	0	0	0	0	4.19E+04	1.22E+04
18	0.12	0.072	2.191	42.335	2.191	42.335	0	0	0	0	0	0	4.19E+04	1.22E+04

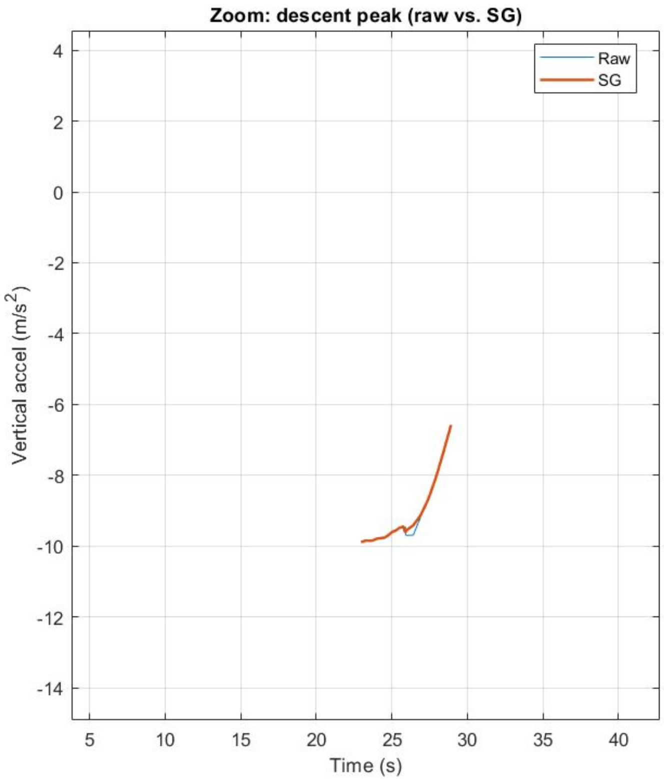
tau_inflate.m — Preprocessing & Derivatives

Before analysis, the raw flight data is preprocessed. The data is first resampled to a consistent time step so calculations stay accurate. Then altitude is smoothed to remove noise with Savitzky-Golay filtering, and velocity + acceleration are derived. Finally, only the descent portion of the flight is kept, since parachute openings happen after apogee.

- Resamples to uniform grid (fs_target = 100 Hz by default)
- Smooths altitude with Savitzky–Golay filter (~0.30 s window)
- Velocity v(t) and acceleration a(t) from numerical gradient on uniform grid
- Descent only: takes data after apogee (max altitude)

Savitzky-Golay Altitude/Acceleration Smoothing From Flight Data





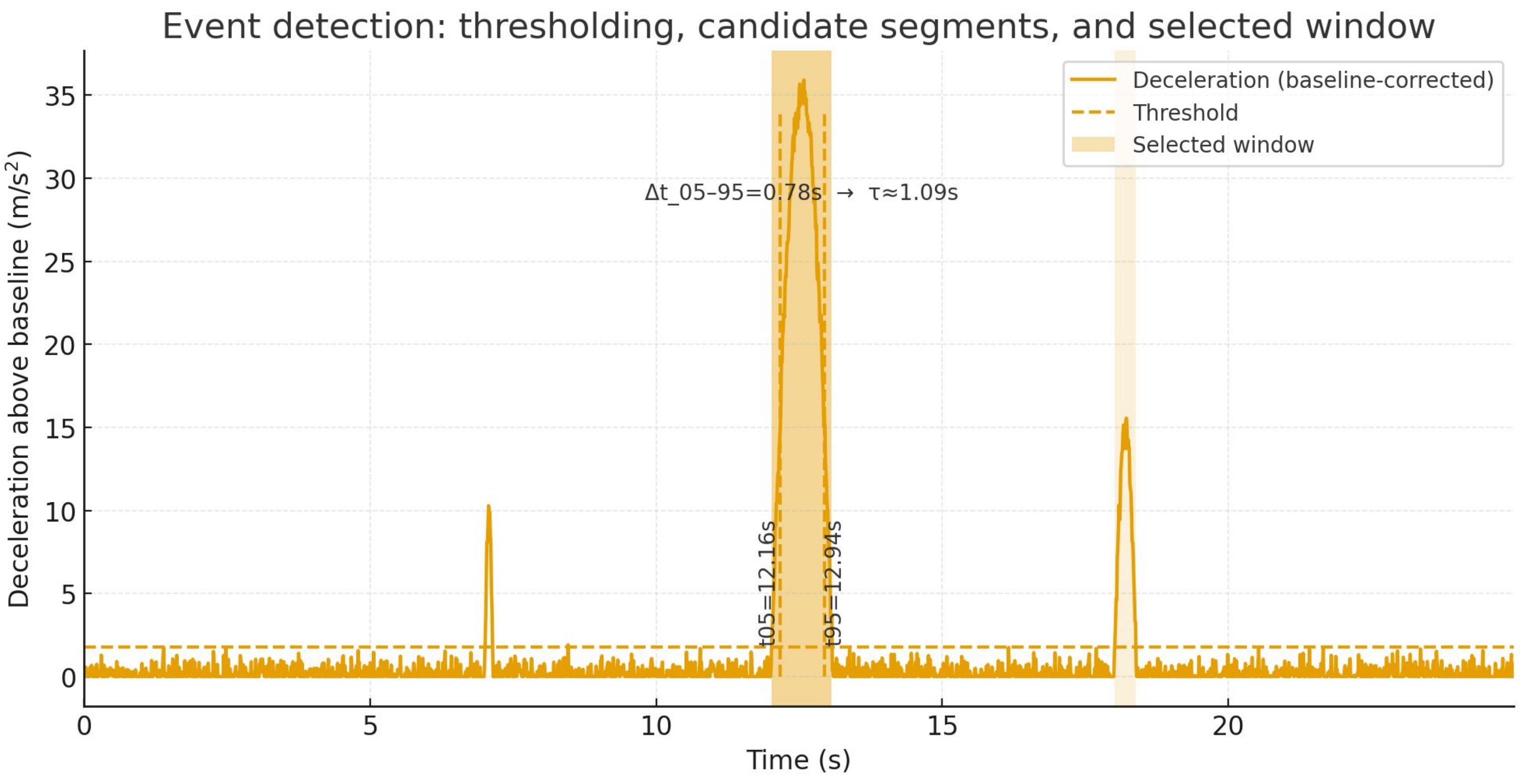
Savitzky-Golay smoothing filter (order 3) applied to altitude and acceleration data

tau_inflate.m — Event Detection

This step identifies the parachute opening event in the data. It looks for strong upward deceleration while descending and ignores small noisy spikes. Only nontrivial events are kept. The program chooses the segment with the largest impulse, since that corresponds to peak parachute shock.

- Deceleration signal: decel = max(0, a_d) (upward accel while descending)
- Find contiguous segments above threshold: max(0.05·peak, a_floor)
- Guards: minimum duration (≥0.30 s) and minimum samples to reject noise
- Selects the segment with the largest impulse (∫decel dt)

tau_inflate.m — Event Detection



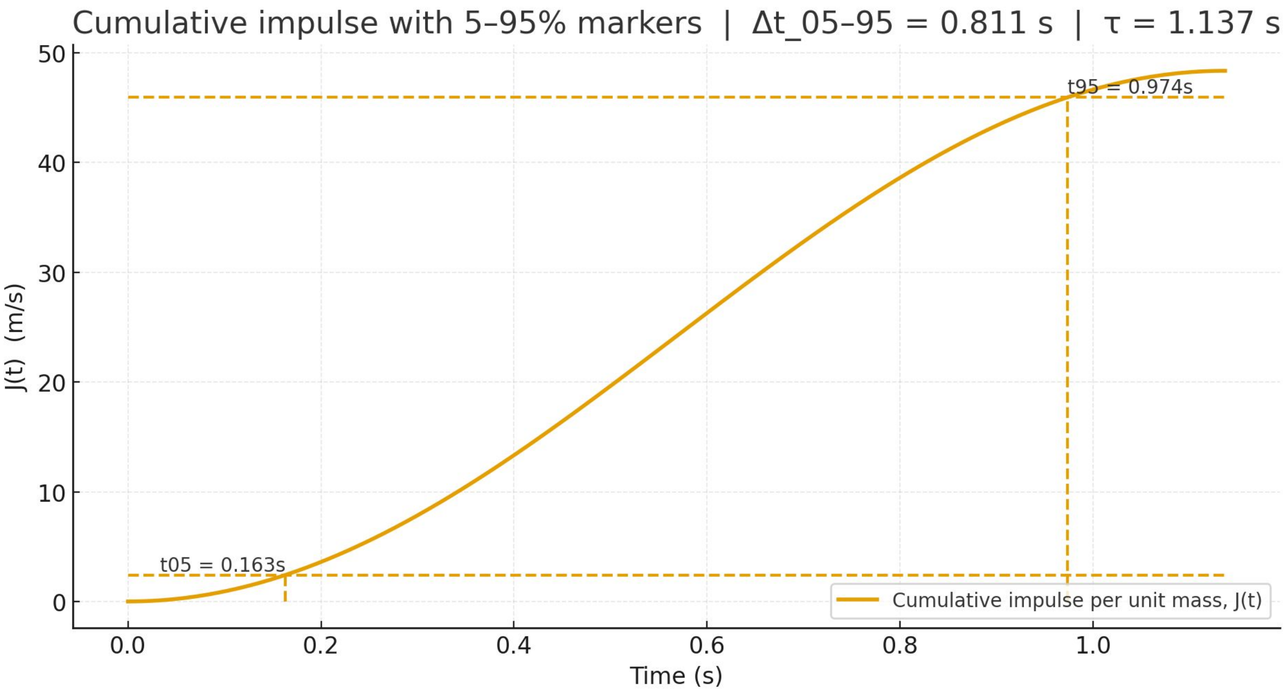
Visualization of event detection window using sample dataset

tau_inflate.m — τ and Δv

This step converts the acceleration data into impulse. By looking at the 5%–95% portion, the analysis eliminates noise. From that, tau_inflate is calculated as the effective parachute inflation time, and a corresponding peak deceleration is estimated assuming a half-sine pulse shape.

- Compute cumulative impulse per unit mass over the selected window
- Find where the cumulative impulse reaches 5% and 95% to minimize noise
- tau_inflate = (t95 − t05)/0.713 (5–95% duration mapped to half-sine tau)
- a_peak_tau = (π/2)·Δv/tau; estimates peak decel for a half-sine pulse
- Prints peak decel, tau_inflate, and peak velocity before deployment

tau_inflate.m — Cumulative Impulse Plot with 5% & 95% markers



Data visualization of 5% and 95% cutoffs over cumulative impulse window

Console Print Out

Command Window

```
>> Tau_Inflate
dt_med = 0.050000 s  (~20.0 Hz)
dt_med = 0.050000 s  (~20.0 Hz)
V0 ≈ 40.65 m/s (|v|),  or  40.65 m/s (downward +)
Δt_05-95 = 0.811 s | tau_inflate = 1.137 s | a_peak = 66.819 m/s^2
Window: [65.826, 66.637] s | samples = 122 | fs_target = 100 Hz
fx >>
```

Shock Mapping

cf_shock.m — Peak Load & Resultants

The parachute's peak deceleration is translated into forces onto the rocket tank. The force carried by the shell is spread evenly around the circumference of the tank. These parachute shock loads are then added to the normal internal pressure loads the tank already experiences. The result is a set of stresses (axial, hoop, and shear) that can be passed into the composite analysis.

- a_peak = a_peak_tau (tau-governed)
- F_peak = m·(a_peak + g). Convert to lbf for shell resultants
- N_x_shock = F_peak(lb) / (2πR(in)) (uniform circumferential smear)
- Add pressure resultants: N_x = p·R/2 + N_x_shock; N_y = p·R; N_xy = 0

cf_shock.m — Tank & Bulkhead Diagram

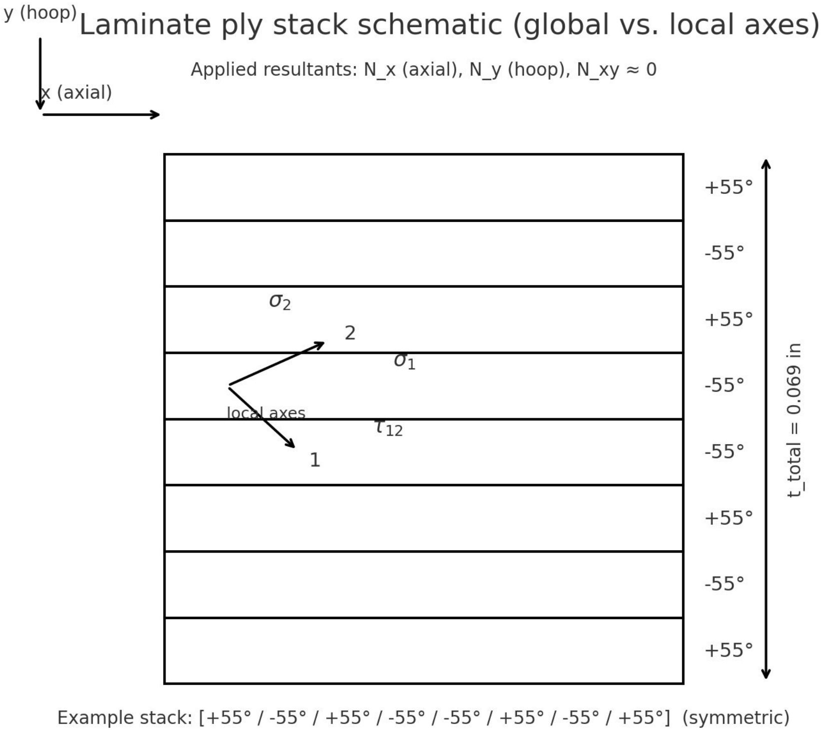
Diagram depicting a cross-section of the carbon fiber tank, the pink shaded cross-section shows the bulkhead. The parachutes are attached to this bulkhead. Shock centers on bolts connecting bulkhead to tank.

cf_shock.m — CLT & Ply Stresses

Classical Laminate Theory is applied to translate overall tank forces into stresses within each carbon fiber layer. The stiffness of the laminate's plies is defined in an A-matrix based on fiber orientation and material properties. The shock and pressure loads are used to compute strains, which are then converted into local stresses for each ply direction.

- Build reduced stiffness Q with E1, E2, v12, G12; compute Q at ±θ plies
- Assemble A-matrix for the laminate (use actual stack; example uses ±55°)
- Solve membrane strains: $\varepsilon = A^{-1} \cdot N$
- Transform to local ply axes; compute ply stresses (σ_1, σ_2, τ_{12})
- **Thank you to Christopher Liang for providing stress matrix computation**

cf_shock.m — CLT & Ply Stresses



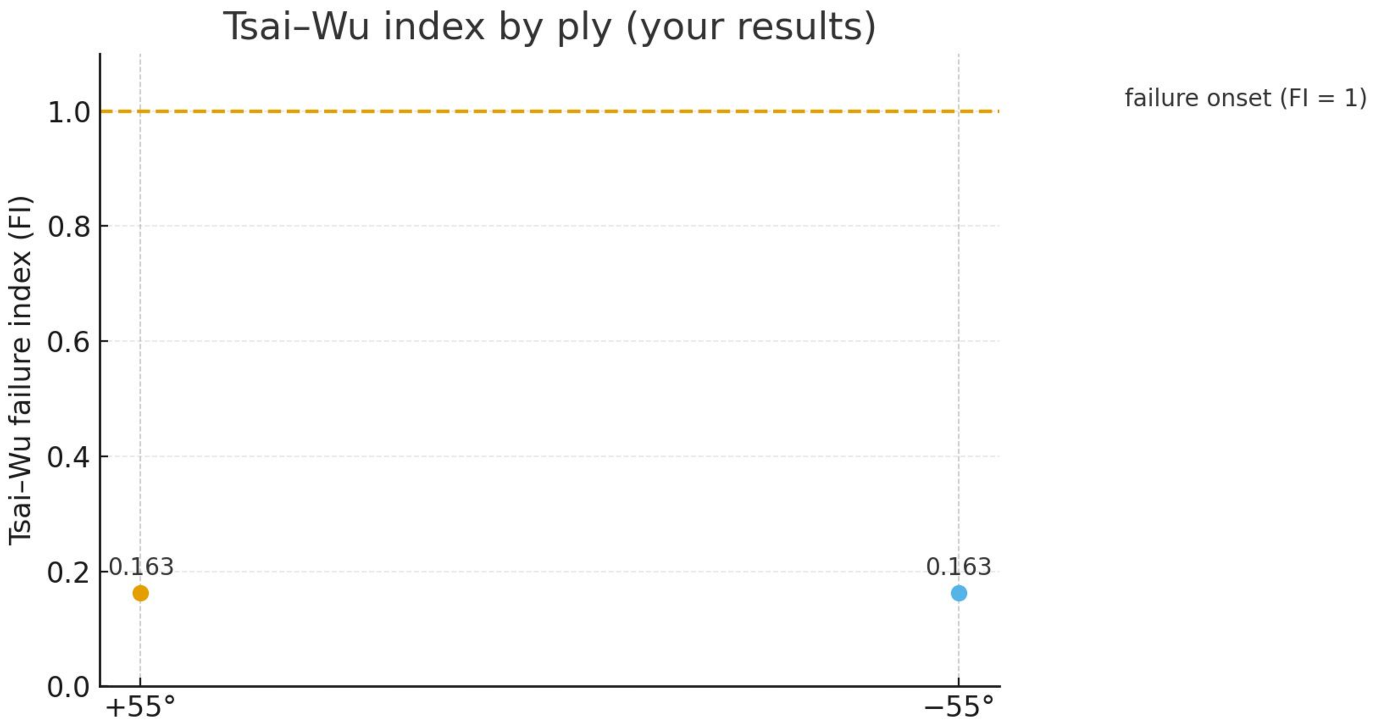
Laminate ply stack schematic [+55/−55/+55/−55/−55/+55/−55/+55 stack] with applied resultants (N_x, N_y, N_xy)

cf_shock.m — Tsai–Wu Failure Criteria

Here the carbon fiber tank is treated as a layered laminate. We check the stresses in each ply against Tsai-Wu material failure criteria. If the failure index stays below 1, the tank can handle the load with margin. If it's above, the parachute shock could cause structural failure.

- Tsai–Wu index per ply uses Xt, Xc, Yt, Yc, S
- Reserve factor FS: multiplier on loads to reach TW=1 (FS>1 passes)
- Report TW_total and FS for +55° and −55° plies

cf_shock.m — Tsai–Wu Failure Criteria



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

Opening Shock Factor: K_osf

K_osf (Opening Shock Factor) is a dimensionless multiplier that scales the steady-state drag force of a parachute to estimate the peak force during inflation. K_osf “packages” parachute effects (deployment speed, parachute design, inflation time, etc.) into a single number so that worst-case peak loads can be quickly estimated.

K_osf Sanity Check

- If tau_inflate implies a K_osf value outside feasibility, it flags that windowing or baseline f_peak values are wrong: **use K_osf to validate tau_inflate**
- Cf_shock is therefore tau_inflate-governed by empirical data
- Back-solving for K_osf and cross-checking with tau_inflate can determine tau_inflate's feasibility (normally falls between 1-2 seconds)

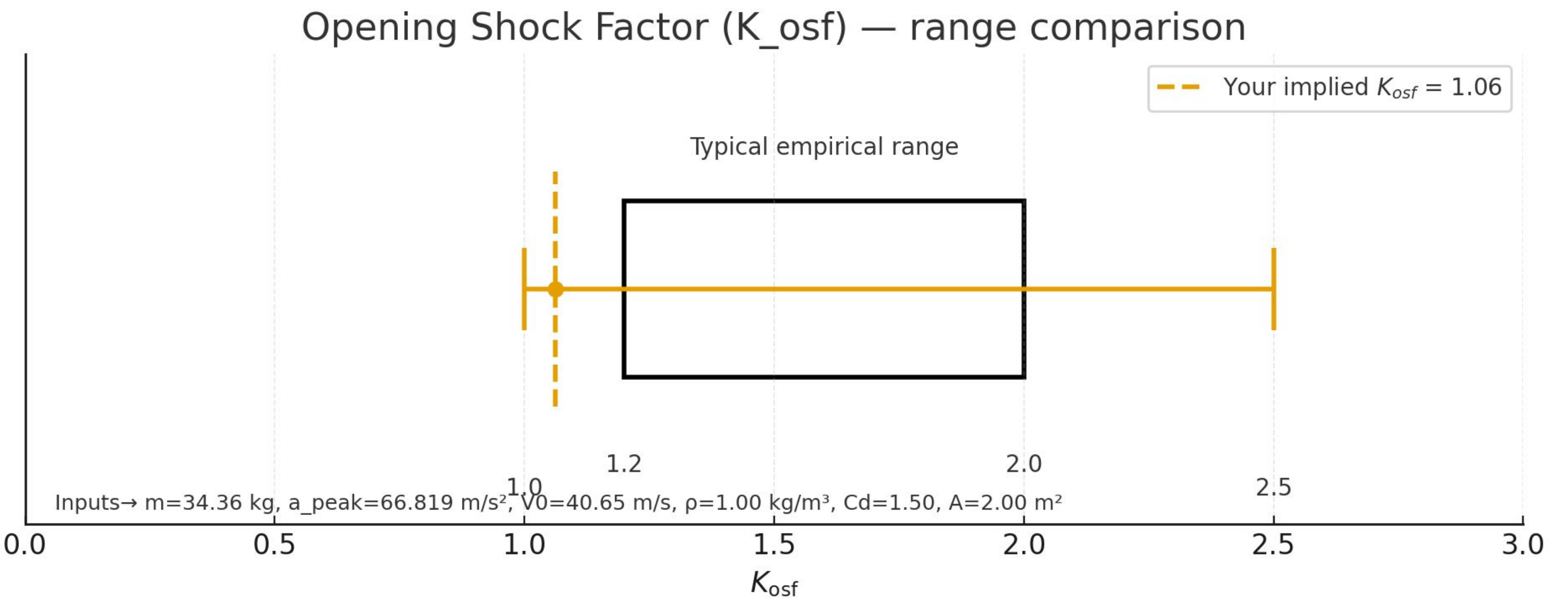
A. Opening shock factor C_k basics for clusters

Knowing the value of C_k allows for the computation of peak loads (F_{max}) from a simple equation, namely [1, 2],

$$F_{\max} = \left(\tfrac{1}{2}\rho V_i^2\right)\left(S_{steady} C_{Dsteady}\right) \cdot C_k \quad , \tag{1}$$

Potvin, J., and Ray, E., "Extrapolating the Trends of Test Drop Data with Opening Shock Factor Calculations: The Case of the Orion Main and Drogue Parachutes Inflating to 1st Reefed Stage," *25th AIAA Aerodynamic Decelerator Systems Technology Conference*, Denver, CO, June 5–9, 2018.

K_osf Sanity Check



K_osf range comparison chart displaying typical empirical range and computed K_osf value from inputs

Empirical range from Potvin and Ray, AIAA ADSC, 2018

How to Run — Step by Step

1) Open MATLAB in the folder with tau_inflate.m, cf_shock.m and your CSV

2) Run tau_inflate.m → it prints τ, Δt_05–95, a_peak (data), V_ls, and saves tau_outputs.mat

3) Set stage parameters in cf_shock.m (drogue/main, Cd, A_full, reef_ratio).

4) Run cf_shock.m → it loads tau_outputs.mat, computes F_peak, stresses, Tsai–Wu, K_osf(implied)

5) Review console summary & compare K_osf(implied) to typical ranges

6) Review Tsai-Wu failure criteria satisfaction

Console Print Out

Command Window

```
--- Parachute Opening (tau_inflate governs) ---
a_peak_tau = 42.162 m/s^2
F_peak = 1786 N (401 lb)
Durations: tau = 1.137 s | dt_rect = 0.724 s | dt_05-95 = 0.811 s | dt_fwhm = 0.758 s
Calculated opening shock factor (Implied) = 1.036
Membrane resultants [lb/in]: N_x_shock = 16.46 | N_x_total = 1472.21 | N_y_total = 2911.50
Global membrane stresses [kpsi]: Axial = 21.336 | Hoop = 42.196 | Shear = 0.000
--- Local Ply Stresses [kpsi] ---
+55° lamina
  Longitudinal: 60.090  |  Transverse: 3.442  |  Shear: 0.790
-55° lamina
  Longitudinal: 60.090  |  Transverse: 3.442  |  Shear: -0.790
--- Tsai-Wu ---
+55°: TW_total=0.163  |  FS=3.471 (reserve factor)
-55°: TW_total=0.163  |  FS=3.471 (reserve factor)
fx >>
```

tau_inflate

k_osf

Tsai-Wu evaluations

Interpreting Outputs

- τ_inflate (s): inflation duration for test pulses and timing comparisons
- a_peak_tau (m/s^2): half-sine peak; use this for F_peak (τ-governed)
- F_peak (N) and N_x_shock (lb/in): size the shell globally
- Global stresses (kpsi): axial/hoop/shear under pressure + opening load
- Tsai–Wu FS: >1 means pass; report minimum FS over critical plies
- K_osf(implied): feasibility check; out-of-band values prompt a data/parameter review

Results Based On Current Data

Laminate has healthy margin. With TW Failure Index = 0.163 and Safety Reserve Factor = 3.47, the ±55 stack shows strong structural capacity for shock. **It will withstand the shock loads.**

Opening pulse is moderate. Kosf at 1.04 sits at the low end of common ranges; a half-sine assumption with tau = 1.14 s is consistent with a soft, non-snappy inflation.

Shock is a minor contributor to shell resultants. Nxshock is ~1% of Nxtotal, which shows that the tank membrane is governed by internal pressure, not by the opening shock spike.