Metallurgia fisica



Esercitazione 6
Fitting e previsioni di dati sperimentali

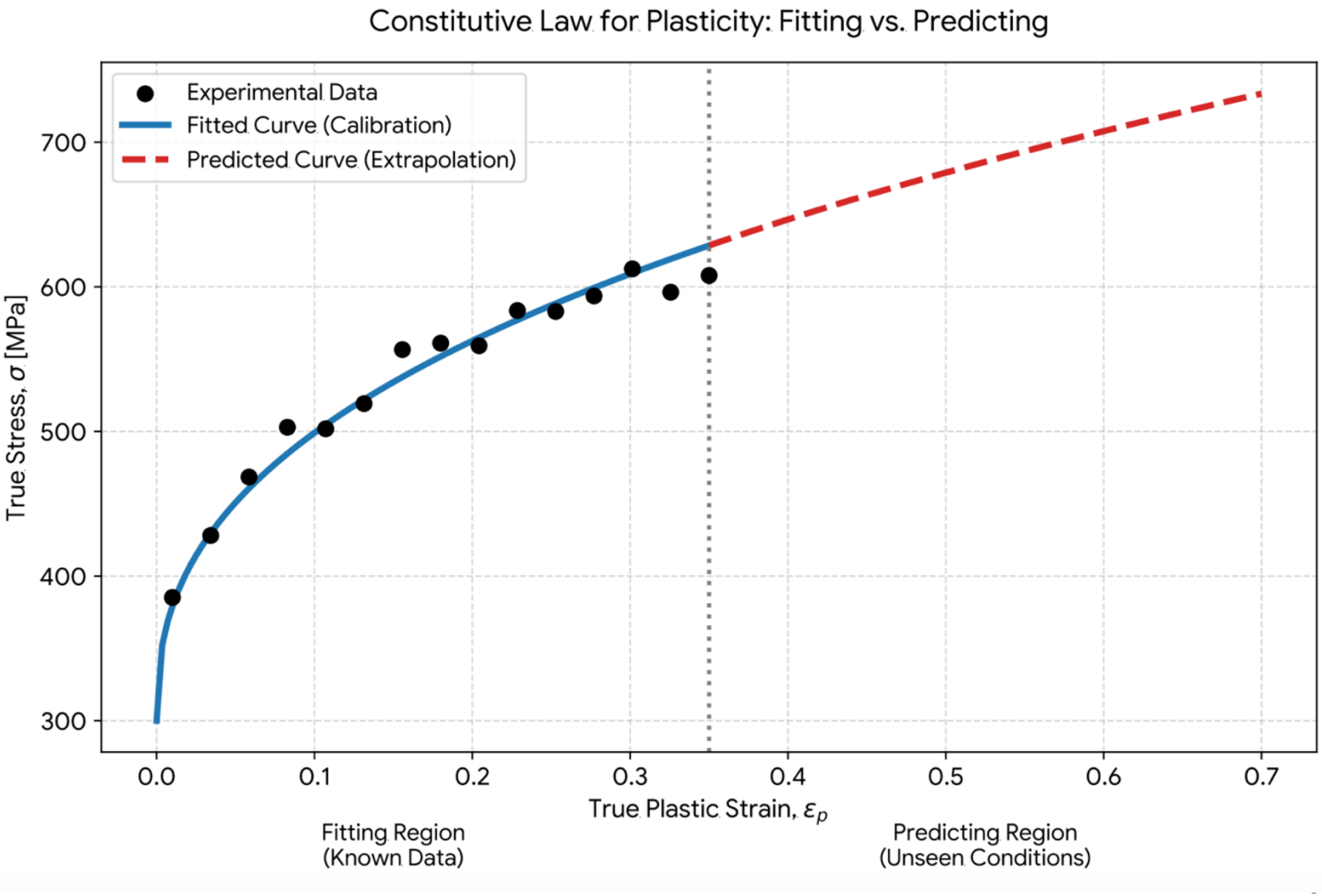
Fitting e Previsione

Fitting: Adattare i parametri di un modello matematico in modo che il suo output corrisponda a un set di dati noto.

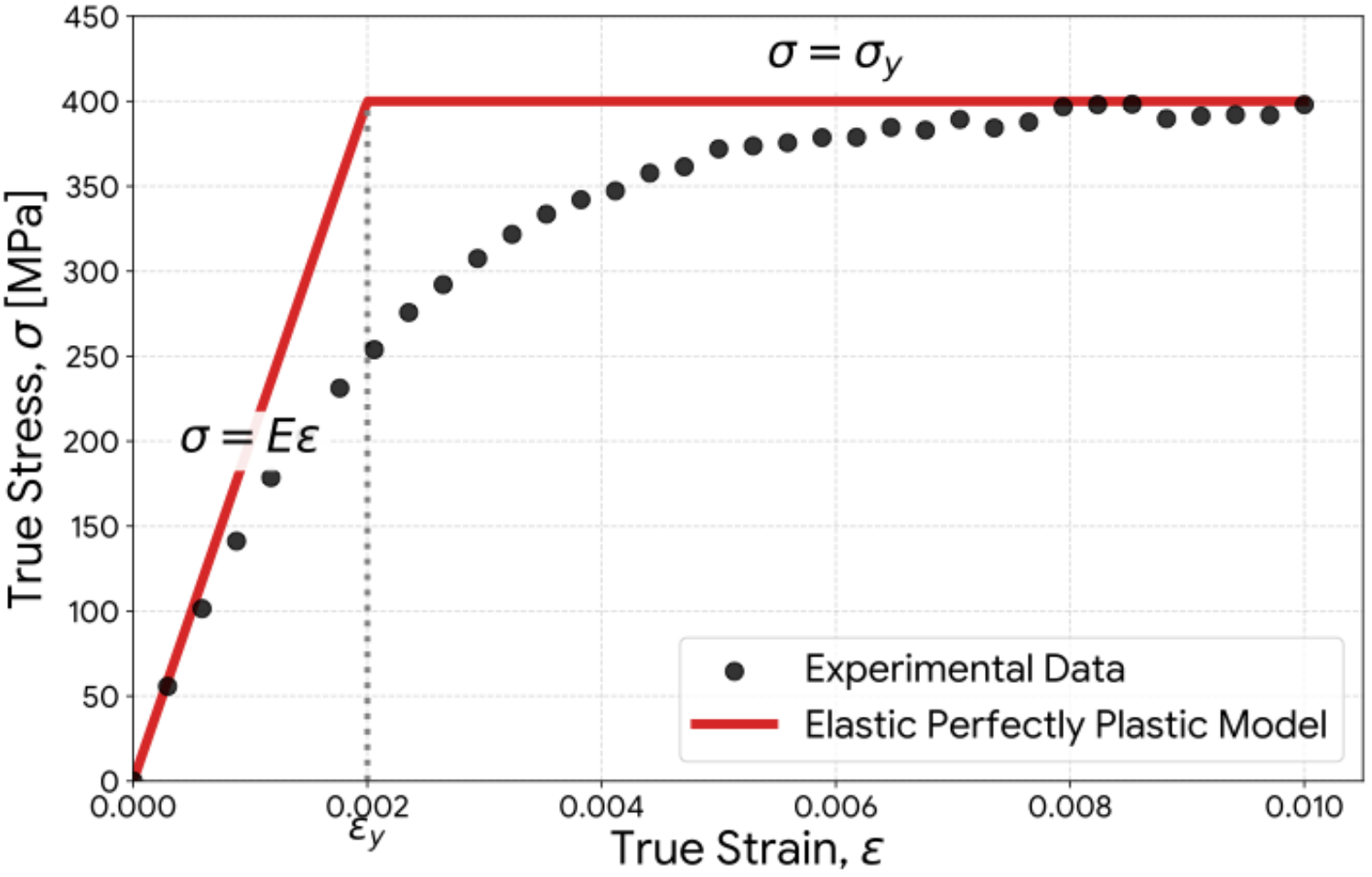
A tal fine si minimizza l'errore tra il modello e il set di dati attraverso interpolazione

Previsione: Si usa poi il modello per prevedere le proprietà o lo stato di un sistema sotto condizioni nuove (non studiate)

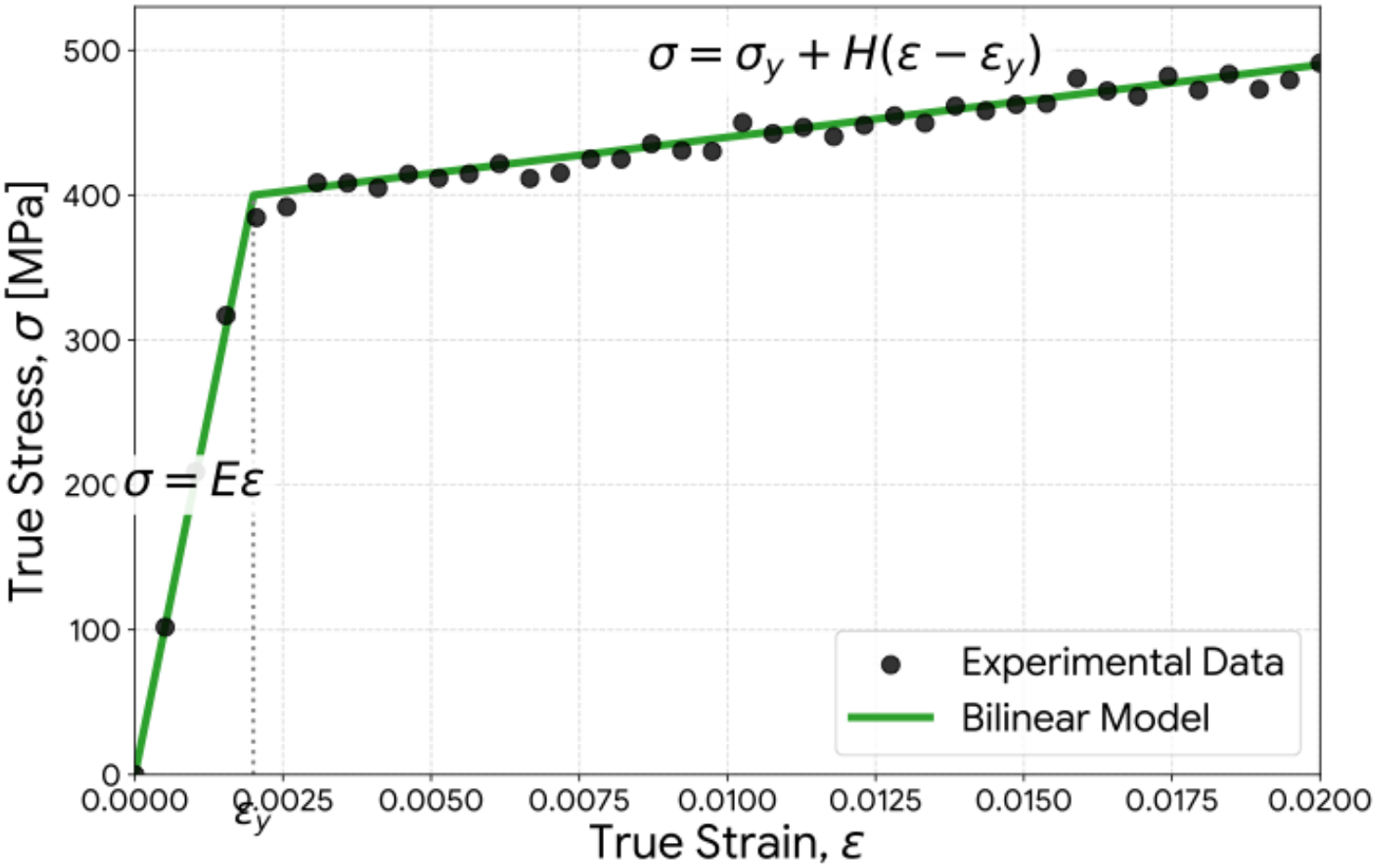
Esempio


Constitutive Law for Plasticity: Fitting vs. Predicting
Experimental Data
Fitted Curve (Calibration)
Predicted Curve (Extrapolation)
True Stress, σ [MPa]
700
600
500
400
300
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
True Plastic Strain, ε_p
Fitting Region
(Known Data)
Predicting Region
(Unseen Conditions)

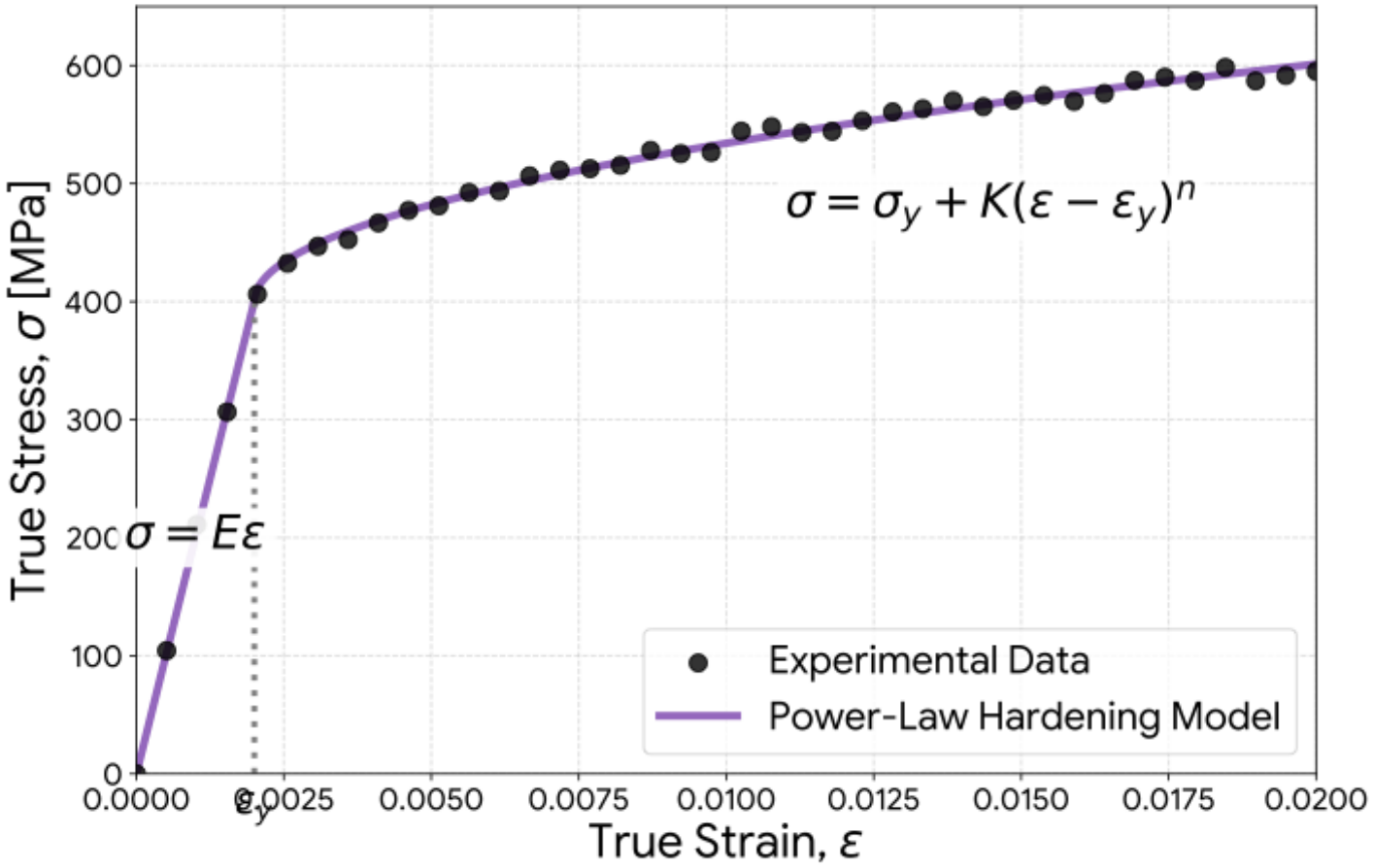
Elastico perfettamente plastico



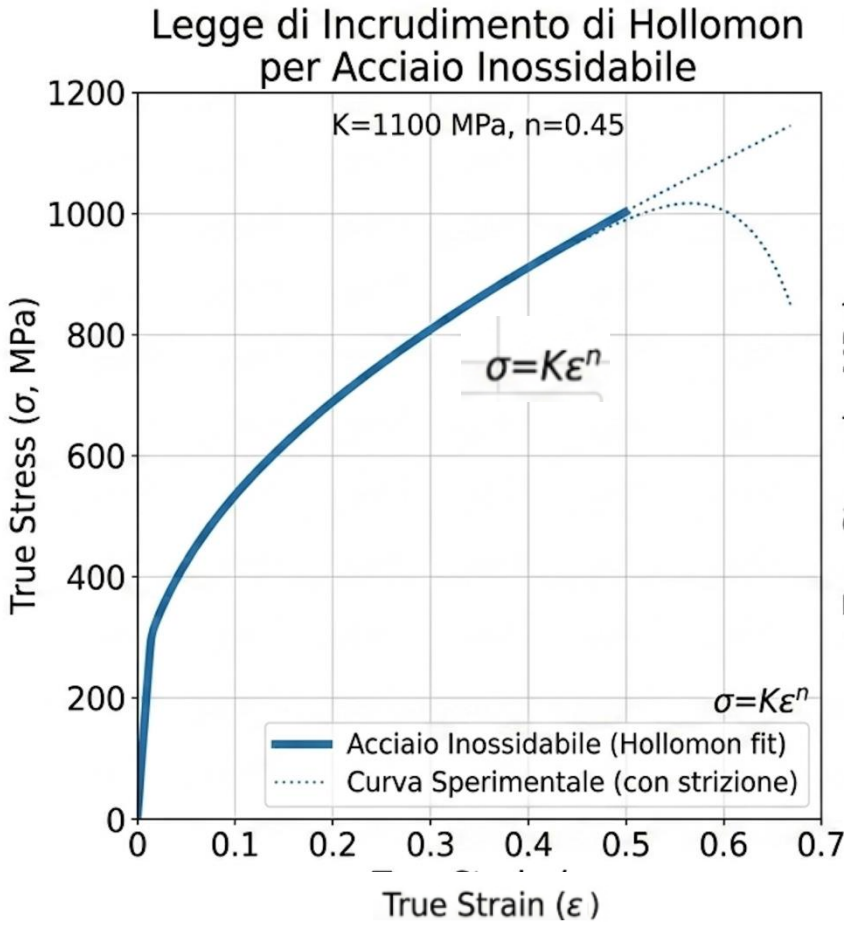
Modello di incrudimento lineare



Modello di incrudimento esponenziale



Modello di incrudimento esponenziale



Quali materiali seguono questi modelli

Material (Condition)	Common Model Fit	E (GPa)	σ_y (MPa)	H (MPa)	K (MPa)	n
Low-Carbon Structural Steel (e.g., A36)	**Elastic-Perfectly Plastic**	200	250	0	-	-
Aluminum Alloy (6061-T6)	**Linear Hardening** (Bilinear)	69	275	850	-	-
Annealed Copper (Pure)	**Power-Law Hardening**	110	35	-	315	0.54
Stainless Steel (304 Annealed)	**Power-Law Hardening**	193	205	-	1275	0.45

Questi modelli sono puramente fenomenologici

Consegna

Avete eseguito una prova a trazione su un provino in acciaio inossidabile e ottenuto un set di dati che restituiscono lo sforzo reale al variare della deformazione reale. Avete salvato i dati nel file tensile_test_data.csv.

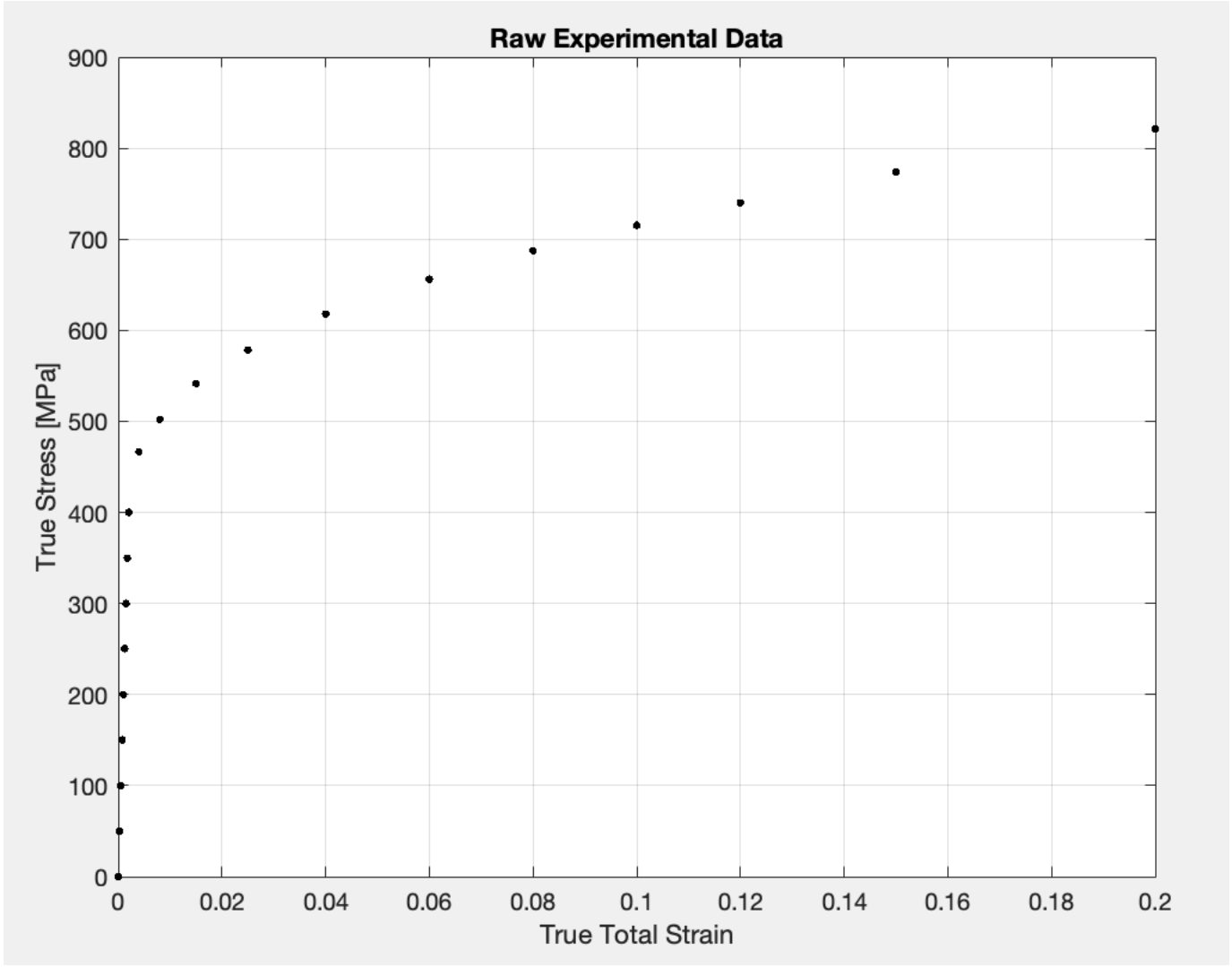
Per prima cosa visualizzate i dati per decidere con quale modello possano essere interpolati.

Leggi e plotta i dati

```
%% PART 1: Data Import, Elastic Modulus, and Yield Point
clear; clc; close all;

% 1. Load the Data
data = readmatrix('steel_tensile_test.csv');
strain_total = data(:, 1);
stress_true = data(:, 2);

% Plot the raw data
figure;
plot(strain_total, stress_true, 'k.', 'MarkerSize', 8);
xlabel('True Total Strain');
ylabel('True Stress [MPa]');
title('Raw Experimental Data');
grid on;
```


Raw Experimental Data
True Stress [MPa]
True Total Strain
0
100
200
300
400
500
600
700
800
900
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

Parte 1

Elasticità:
Sappiamo che si tratta di elasticità lineare

1. Si determini il modulo elastico
2. Si determini il punto di snervamento

Suggerimenti

1. Si selezionino solo i dati desiderati
2. Si utilizzi il commando: polyfit

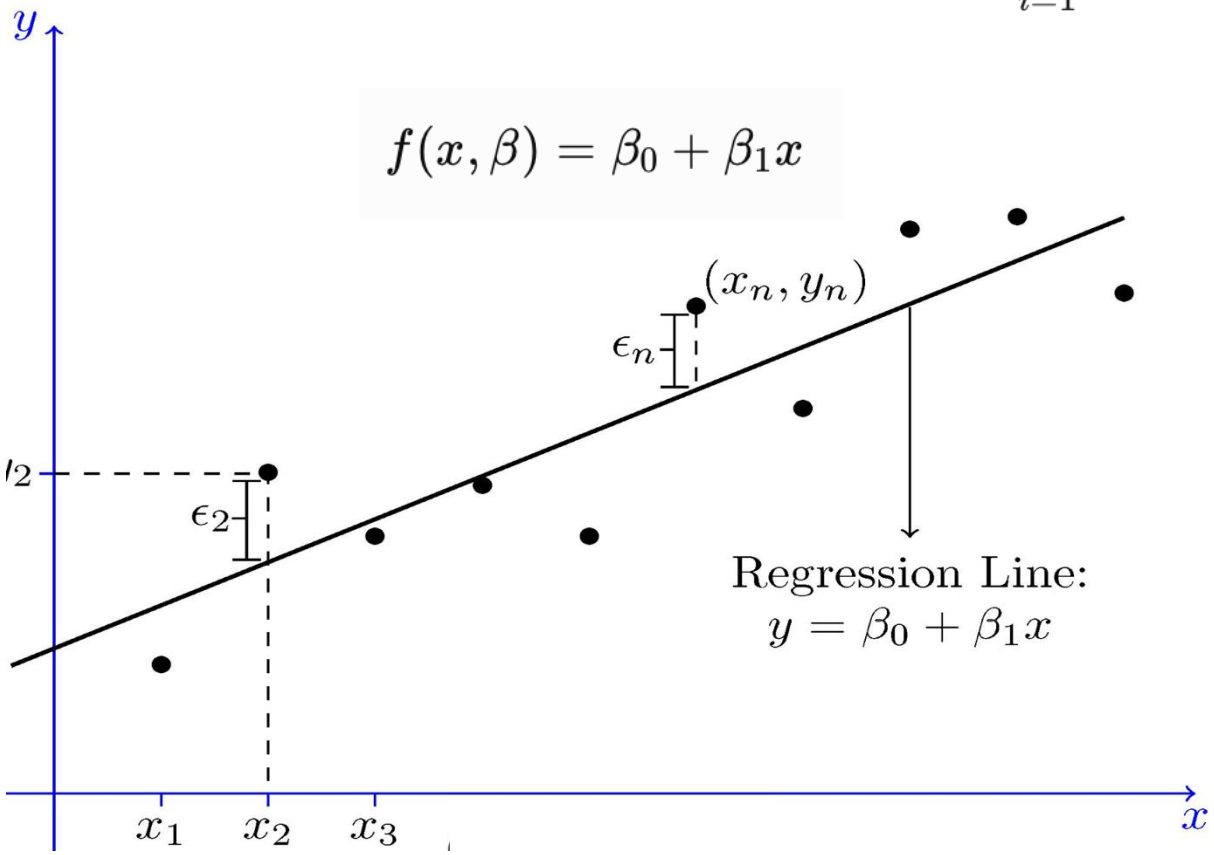
coeffs=polyfit(x,y,n) restituisce un vettore contenente gli n+1 coefficienti di un polinomio p(x) di grado n che sia la miglior approssimazione dei dati in y (ottimizza in termini di somma dei minimi quadrati)
I coefficient vengono dati in ordine di potenza decrescente.

$$p(x) = ax^2 + bx + c$$

```
coeffs = [a, b, c]
```

Esempio

Se abbiamo un modello $f(x, \beta)$ cerchiamo valori dei parametri beta che rendano minima la somma degli errori quadratic: $S = \sum_{i=1}^{n} [y_i - f(x_i, \beta)]^2$



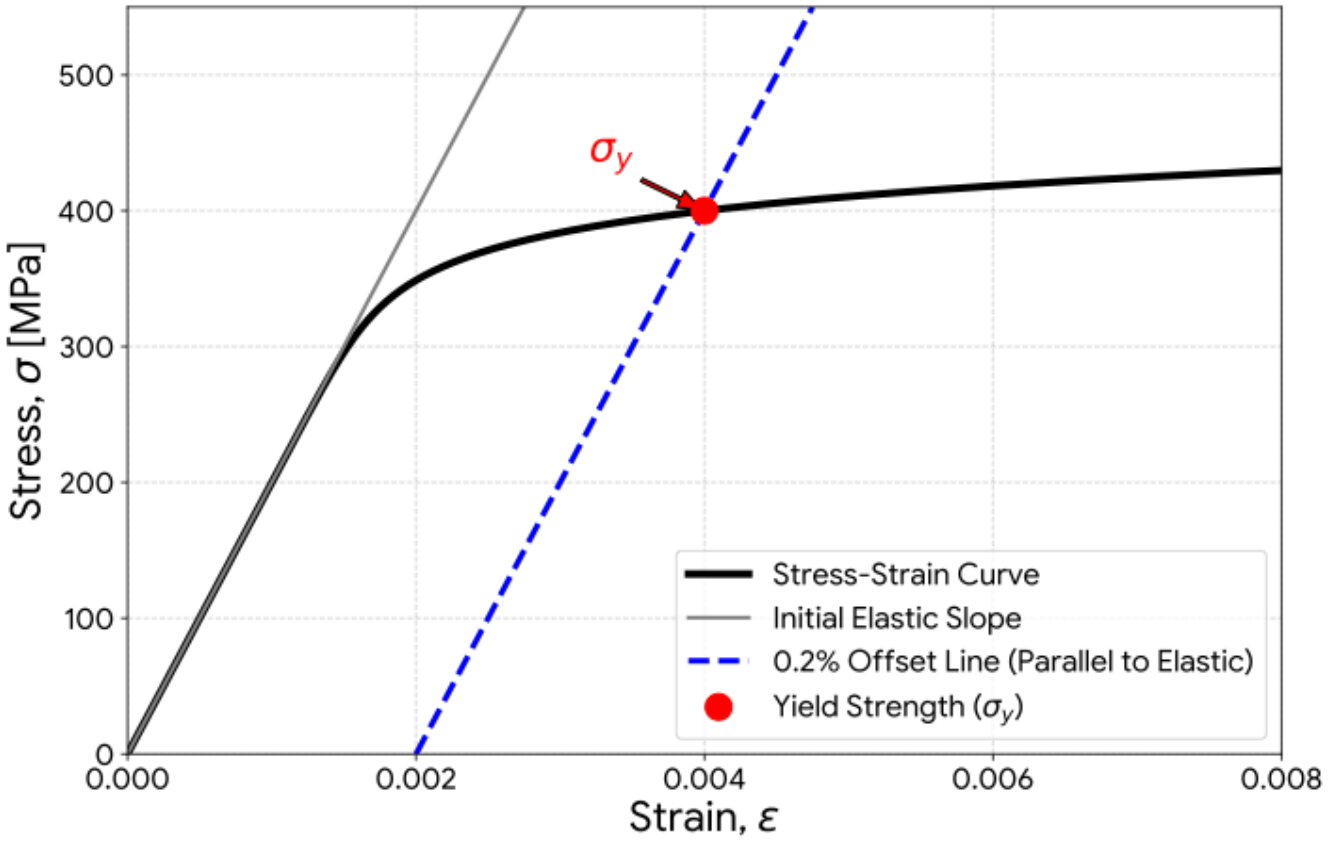
Modulo elastico

```
% 2. Calculate Young's Modulus (E)
% Use the first few linear points (e.g., points 2 through 6)
elastic_strain = strain_total(2:6);
elastic_stress = stress_true(2:6);

% Fit a 1st degree polynomial (a line) to find the slope (E)
p_elastic = polyfit(elastic_strain, elastic_stress, 1);
E = p_elastic(1);
```

Tensione di snervamento

Consiglio: si utilizzi il metodo dello 0.2% di offset per ottenere il punto di snervamento



Find Yield Strength

```
% 3. Find Yield Strength using the 0.2% Offset Method
% Create the offset line equation: sigma = E * (strain - 0.002)
offset_strain = 0.002;
stress_offset_line = E * (strain_total - offset_strain);

% Find where the actual data crosses above the offset line
yield_idx = find(stress_true < stress_offset_line, 1);

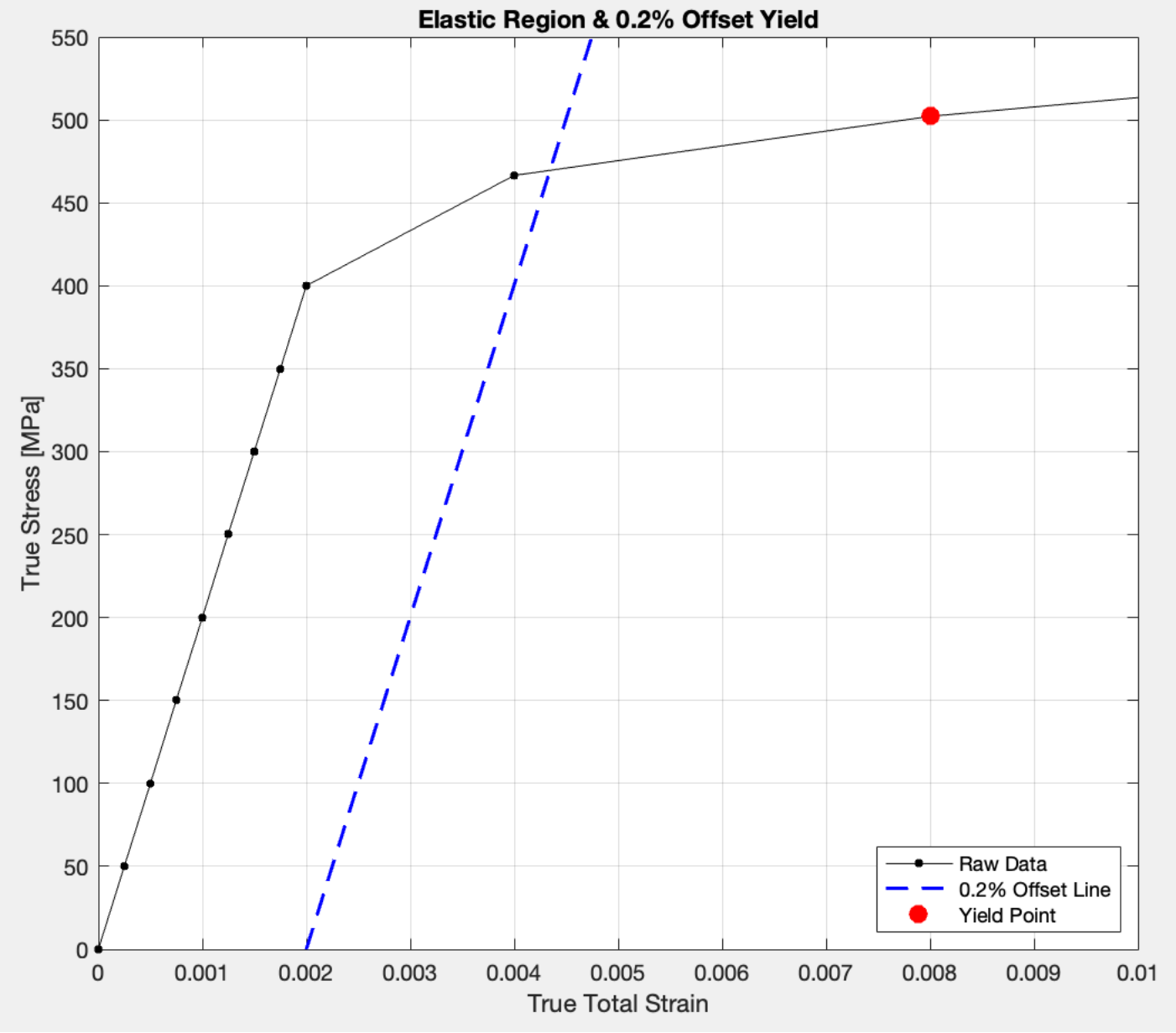
sigma_y = stress_true(yield_idx);
eps_y = strain_total(yield_idx);

% Print Part 1 Results
fprintf('--- PART 1: Elastic Properties ---\n');
fprintf('Young''s Modulus (E): %.0f MPa\n', E);
fprintf('Yield Strength (sigma_y): %.2f MPa\n\n', sigma_y);
```

Visualize part 1

```
% 4. Visualize Part 1
figure('Name', 'Part 1: Elastic Region and Yield', 'Position', [100, 100, 600, 500]);
plot(strain_total, stress_true, 'k.-', 'MarkerSize', 10, 'DisplayName', 'Raw Data');
hold on;
plot(strain_total, stress_offset_line, 'b--', 'LineWidth', 1.5, 'DisplayName', '0.2% Offset Line');
plot(eps_y, sigma_y, 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r', 'DisplayName', 'Yield Point');

% Zoom in to see the intersection clearly
xlim([0, 0.01]);
ylim([0, 550]);
xlabel('True Total Strain');
ylabel('True Stress [MPa]');
title('Elastic Region & 0.2% Offset Yield');
legend('Location', 'southeast');
grid on;
```


Elastic Region & 0.2% Offset Yield
True Stress [MPa]
True Total Strain
0
50
100
150
200
250
300
350
400
450
500
550
0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Raw Data
0.2% Offset Line
Yield Point

Part 2: plasticità

Assumendo che l'equazione di Holloman sia un buon modello per la risposta meccanica dell'acciaio testato, si trovi il coefficient K e l'esponente n.

$$\sigma = K\varepsilon_p^n$$

Part 2: plasticità

Assumendo che l'equazione di Holloman sia un buon modello per la risposta meccanica dell'acciaio testato, si trovi il coefficient K e l'esponente n.

$$\sigma = K\varepsilon_p^n$$

Suggerimento:
si noti che l'equazione contiene solo la deformazione plastica.

Isoliamo la plasticità

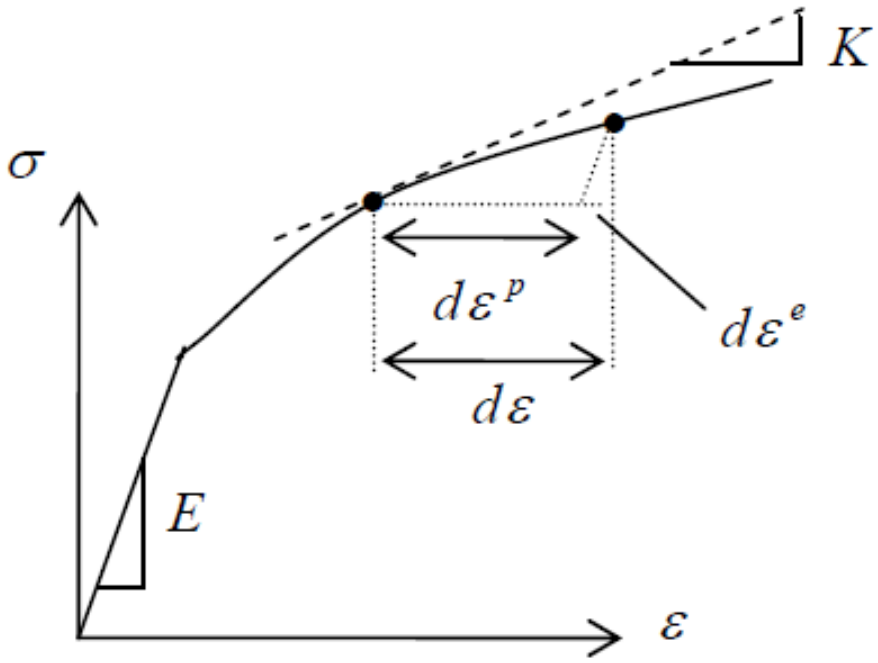
Rimuoviamo i dati con deformazione reale inferiore a quella del punto di snervamento, sono solo dati in elasticità, non ci interessano

```
%% PART 2: Plastic Region and Power-Law Hardening
% Note: Do not clear the workspace! We need E, sigma_y, and yield_idx from Part 1

% 5. Isolate the Plastic Region (from the yield point to the end)
stress_plastic_total = stress_true(yield_idx:end);
strain_plastic_total = strain_total(yield_idx:end);
```

E il resto dei dati ci va bene?

Deformazione plastica



Isoliamo la plasticità

Rimuoviamo la deformazione elastica da quella totale

```
% 6. Calculate True Plastic Strain
% Formula: epsilon_plastic = epsilon_total - (sigma / E)
eps_plastic = strain_plastic_total - (stress_plastic_total / E);
```

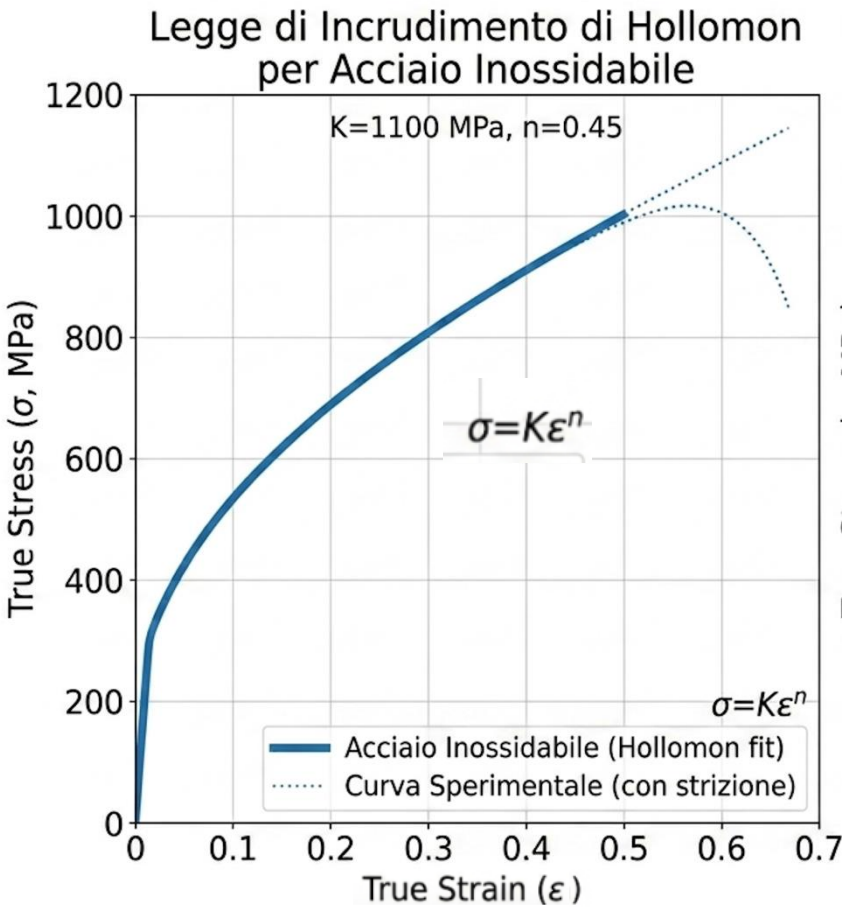
Equazione di Halloman in forma logaritmica

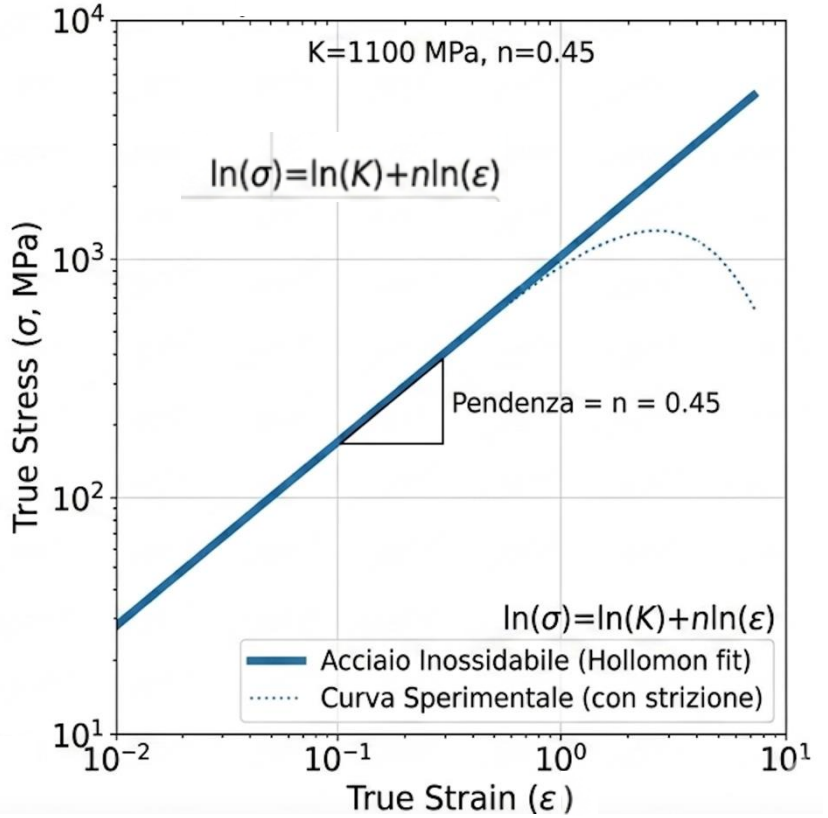
Se prendiamo il logaritmo naturale dell'equazoine di Halloman ci semplifichiamo il lavoro:

$$\ln(\sigma) = \ln(K) + n\ln(\varepsilon_p)$$

Abbiamo semplicemente una retta con pendenza *n* and intercetta *ln(K)* con l'asse *y*.

Modello di incrudimento esponenziale





Regressione lineare

```
% Remove any points where plastic strain is zero or negative to avoid log errors
valid_indices = eps_plastic > 1e-5;
eps_plastic = eps_plastic(valid_indices);
stress_plastic_total = stress_plastic_total(valid_indices);

% 7. Log-Log Linear Regression (Hollomon Equation)
% Equation: ln(sigma) = ln(K) + n * ln(epsilon_p)
log_eps_p = log(eps_plastic);
log_stress = log(stress_plastic_total);

% Fit the line to find slope (n) and intercept (ln(K))
p_plastic = polyfit(log_eps_p, log_stress, 1);
n_fit = p_plastic(1);
K_fit = exp(p_plastic(2));

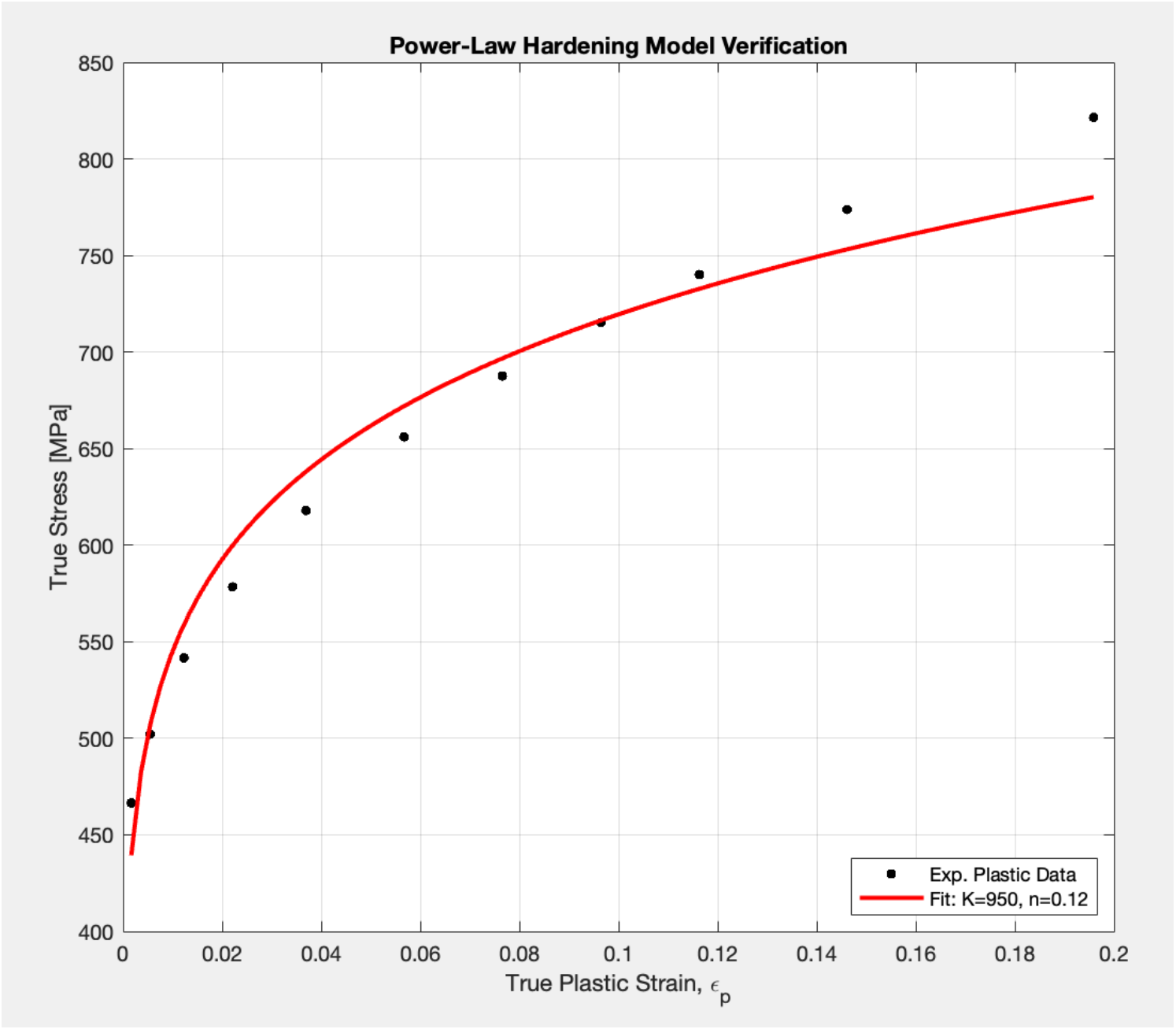
% Print Part 2 Results
fprintf('--- PART 2: Hardening Parameters ---\n');
fprintf('Strength Coefficient (K): %.2f MPa\n', K_fit);
fprintf('Strain Hardening Exponent (n): %.4f\n', n_fit);
```

Visualizzazione

```
% 8. Visualize Part 2 (Verification)
figure('Name', 'Part 2: Power-Law Fit', 'Position', [750, 100, 600, 500]);
plot(eps_plastic, stress_plastic_total, 'k.', 'MarkerSize', 12, 'DisplayName', 'Exp. Plastic Data');
hold on;

% Generate the theoretical curve from our new K and n
strain_model = linspace(min(eps_plastic), max(eps_plastic), 100);
stress_model = K_fit * (strain_model.^n_fit);

plot(strain_model, stress_model, 'r-', 'LineWidth', 2, 'DisplayName', sprintf('Fit: K=%.0f, n=%.2f', K_fit, n_fit));
xlabel('True Plastic Strain, \epsilon_p');
ylabel('True Stress [MPa]');
title('Power-Law Hardening Model Verification');
legend('Location', 'southeast');
grid on;
```


Power-Law Hardening Model Verification
True Stress [MPa]
True Plastic Strain, ϵ_p
850
800
750
700
650
600
550
500
450
400
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
Exp. Plastic Data
Fit: K=950, n=0.12

Previsione e validazione

Avete eseguito il test di tensione portando il provino ad allungamenti maggiori. Si controlli la validità del modello.
Discuti i risultati

I nuovi punti sono:

```
val_strain_total = [0.25000, 0.30000, 0.35000, 0.40000];
val_stress_true = [865.30, 915.10, 955.40, 990.20];
```

Prediction and validation

```
%% PART 3: Extrapolation and Model Validation
% We will use our computed K and n to predict stress at higher strains,
% and then compare those predictions to new, unseen experimental data.

% 1. Provide New "Unseen" Experimental Data (Larger Strains)
% These points represent true stress/strain values later in the test (up to 0.40 strain)
val_strain_total = [0.25000, 0.30000, 0.35000, 0.40000];
val_stress_true = [865.30, 915.10, 955.40, 990.20];

% 2. Calculate True Plastic Strain for the validation data
% As always: epsilon_p = epsilon_total - (sigma / E)
val_eps_plastic = val_strain_total - (val_stress_true / E);

% 3. Generate Extrapolated Predictions from our Model
% We create a new, extended array of plastic strains going up to 0.45
extended_strain_model = linspace(min(eps_plastic), 0.45, 100);

% Plug the extended strains into the Hollomon equation using our K_fit and n_fit
predicted_stress = K_fit * (extended_strain_model.^n_fit);
```

Prediction and validation

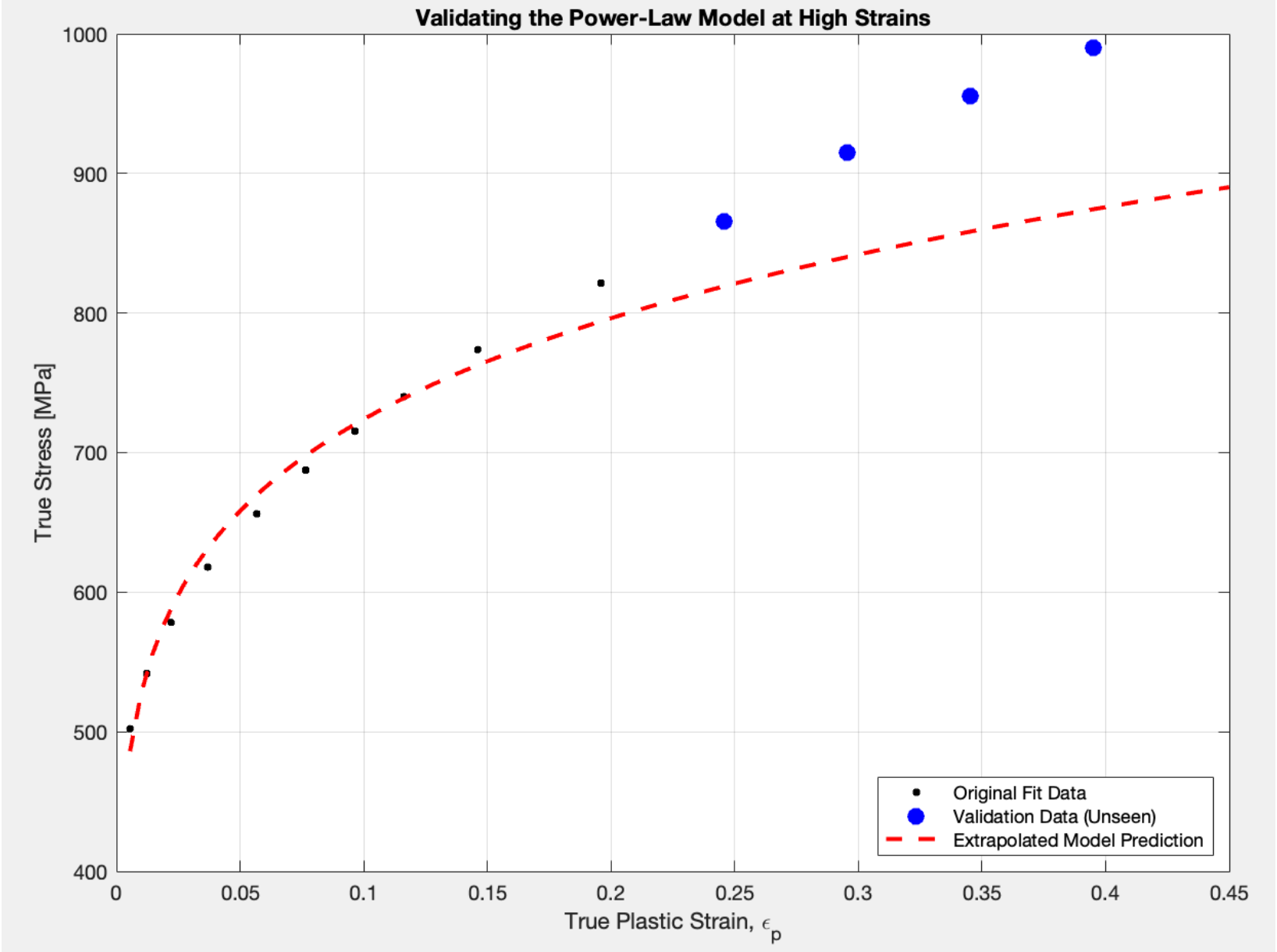
```
% 4. Visualize Extrapolation vs. Reality
figure('Name', 'Part 3: Model Validation', 'Position', [400, 300, 700, 500]);

% Plot the original data used for the fit (Black dots)
plot(eps_plastic, stress_plastic_total, 'k.', 'MarkerSize', 10, 'DisplayName', 'Original Fit Data');
hold on;

% Plot the NEW validation data (Blue circles)
plot(val_eps_plastic, val_stress_true, 'bo', 'MarkerSize', 8, 'MarkerFaceColor', 'b', 'DisplayName', 'Validation Data (Unseen)');

% Plot the extrapolated model curve (Red dashed line)
plot(extended_strain_model, predicted_stress, 'r--', 'LineWidth', 2, 'DisplayName', 'Extrapolated Model Prediction');

% Formatting
xlabel('True Plastic Strain, \epsilon_p');
ylabel('True Stress [MPa]');
title('Validating the Power-Law Model at High Strains');
legend('Location', 'southeast');
grid on;
```


Validating the Power-Law Model at High Strains
True Stress [MPa]
True Plastic Strain, ϵ_p
1000
900
800
700
600
500
400
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Original Fit Data
Validation Data (Unseen)
Extrapolated Model Prediction

Calcolare l’errore residuo

Calcola e plotta l’errore residuo, cioè la differenza tra la curva e i dati.Questo ti consente di verificare se il modello è corretto.

Se gli errori sono casuali, sopra e sotto la linea dello zero (nessuna differenza tra dati e curva) allora il modello e’ corretto. Se incvece ottieni una curva con forma ad U, cerca un nuovo modello.

Calculate the residual

```
% 1. Calculate the predicted values using our linear model
% We use the 'polyval' function to plug our plastic strain data back into our fitted line
log_stress_predicted = polyval(p_plastic, log_eps_p);

% 2. Calculate the Residuals (Error)
% Error = Actual - Predicted
residuals = log_stress - log_stress_predicted;

% 3. Visualize the Residuals
figure('Name', 'Part 4: Residual Analysis', 'Position', [150, 200, 600, 400]);

% Plot the residuals as a scatter plot
plot(log_eps_p, residuals, 'k.', 'MarkerSize', 12, 'DisplayName', 'Residual Errors');
hold on;

% Add a dashed line at zero (perfect prediction)
plot([min(log_eps_p), max(log_eps_p)], [0, 0], 'r--', 'LineWidth', 1.5, 'DisplayName', 'Zero Error Line');

% Formatting
xlabel('Natural Log of Plastic Strain, \ln(\epsilon_p)');
ylabel('Residual Error (Actual - Predicted)');
title('Residual Plot of the Hollomon Fit');
legend('Location', 'best');
grid on;
```

Calculate the residual

