# Economic Load Dispatch with Transmission Line Losses

This script performs economic load dispatch (ELD) calculation with transmission line losses using Newton's method.

Author: Vraj Prajapati Date: March 2025

Description: The program minimizes total generation cost while satisfying power balance constraints including losses.

## Contents



## Initialize data and parameters

```
clc;
clear;
close all;

% Load economic load dispatch data
ELD_data; % Load economic load dispatch data from external file

% Set parameters
error_tolerance_ploss_diff = 0.001;
error_tolerance_newton_method = 0.001;
max_outer_iterations = 50;  % Limit the outer iterations

% Extract data
N = length(PG_data(:,1));
a = PG_data(:,1);
b = PG_data(:,2);
c = PG_data(:,3);
pg_min = PG_data(:,4);
pg_max = PG_data(:,5);
ploss_coeff = PG_data(:,7);

% Initial values
pd = 975;  % Demand
lambda = 8;  % Initial lambda guess

% Better initial guess for generator outputs
% Start with a feasible solution at average between min and max
pg = zeros(N, 1);
for i = 1:N
    pg(i) = (pg_min(i) + pg_max(i)) / 2;
end

% Adjust to meet demand
total_gen = sum(pg);
scaling_factor = pd / total_gen; % alpha value
pg = pg * scaling_factor;

% Make sure initial guess respects limits
for i = 1:N
```

```
    pg(i) = max(pg_min(i), min(pg_max(i), pg(i)));
end

% Initialize power loss
ploss_old = zeros(N, 1);
for i = 1:N
    ploss_old(i) = (pg(i)^2) * ploss_coeff(i);
end

% Initial penalty factors
pf = 1 ./ (1 - 2 * pg .* ploss_coeff);
```

## Display initial conditions

Display initial conditions

```
fprintf('Initial conditions:\n');
fprintf('Demand (Pd) = %.2f MW\n', pd);
for i = 1:N
    fprintf('Generator %d: Pg = %.2f MW (%.2f to %.2f MW)\n', ...
            i, pg(i), pg_min(i), pg_max(i));
end
fprintf('Initial power loss = %.4f MW\n', sum(ploss_old));
fprintf('Initial penalty factors: %.4f, %.4f, %.4f\n', pf(1), pf(2), pf(3));
fprintf('-------------------------------------------------\n');
```

```
Initial conditions:
Demand (Pd) = 975.00 MW
Generator 1: Pg = 487.50 MW (150.00 to 600.00 MW)
Generator 2: Pg = 325.00 MW (100.00 to 400.00 MW)
Generator 3: Pg = 162.50 MW (50.00 to 200.00 MW)
Initial power loss = 44.8906 MW
Initial penalty factors: 1.1080, 1.1080, 1.0695
-------------------------------------------------
```

## Main optimization loop

Outer iteration loop

```
for iteration = 1:max_outer_iterations
    fprintf('\nOuter Iteration %d:\n', iteration);
    fprintf('Total generation: %.4f MW\n', sum(pg));
    fprintf('Total losses: %.4f MW\n', sum(ploss_old));

    % Call Newton's method function
    [pg_new, lambda_new] = newton_method_function(N, a, b, pg, ploss_old, ...
                           ploss_coeff, lambda, pd, pg_min, pg_max, ...
                           error_tolerance_newton_method);

    % Compute new power loss
    ploss_new = zeros(N, 1);
    for i = 1:N
        ploss_new(i) = ploss_coeff(i) * (pg_new(i)^2);
    end

    % Calculate difference in power loss
    diff_ploss = sum(ploss_new) - sum(ploss_old);

    % Check output
    fprintf('After Newton method:\n');
```

```
    fprintf('Lambda = %.6f\n', lambda_new);
    for i = 1:N
        fprintf('Generator %d: Pg = %.4f MW\n', i, pg_new(i));
    end
    fprintf('Total generation: %.4f MW\n', sum(pg_new));
    fprintf('New total losses: %.4f MW\n', sum(ploss_new));
    fprintf('Loss difference: %.6f MW\n', diff_ploss);

    % Update for next iteration
    ploss_old = ploss_new;
    pg = pg_new;
    lambda = lambda_new;

    % Check for convergence
    if abs(diff_ploss) < error_tolerance_ploss_diff
        fprintf('\nConverged! Loss difference < %.6f\n', error_tolerance_ploss_diff);
        break;
    end
end
```

```
Outer Iteration 1:
Total generation: 975.0000 MW
Total losses: 44.8906 MW
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 10: lambda=9.8778, error=-29.503784
  Newton iteration 20: lambda=9.8875, error=28.325760
  Newton iteration 30: lambda=9.8352, error=11.201948
  Newton iteration 40: lambda=9.8124, error=47.845709
  Newton iteration 50: lambda=9.8601, error=33.191202
  Newton iteration 60: lambda=9.7943, error=78.768634
  Newton iteration 70: lambda=9.8437, error=5.868727
  Newton iteration 80: lambda=9.8318, error=15.940031
  Newton iteration 90: lambda=9.7995, error=69.537600
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 100: lambda=16.4691, error=180.109375
Warning: Newton method did not converge within 100 iterations
After Newton method:
Lambda = 16.469054
Generator 1: Pg = 600.0000 MW
Generator 2: Pg = 400.0000 MW
Generator 3: Pg = 200.0000 MW
Total generation: 1200.0000 MW
New total losses: 68.0000 MW
Loss difference: 23.109375 MW

Outer Iteration 2:
Total generation: 1200.0000 MW
Total losses: 68.0000 MW
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
```

```
Warning: Zero derivative in Newton method
  Newton iteration 10: lambda=26.8264, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 20: lambda=43.6973, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 30: lambda=71.1783, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 40: lambda=115.9420, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 50: lambda=188.8572, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 60: lambda=307.6285, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
```

```
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 70: lambda=501.0945, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 80: lambda=816.2301, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 90: lambda=1329.5528, error=157.000000
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
Warning: Zero derivative in Newton method
  Newton iteration 100: lambda=2165.7013, error=157.000000
Warning: Newton method did not converge within 100 iterations
After Newton method:
Lambda = 2165.701340
Generator 1: Pg = 600.0000 MW
Generator 2: Pg = 400.0000 MW
Generator 3: Pg = 200.0000 MW
Total generation: 1200.0000 MW
New total losses: 68.0000 MW
Loss difference: 0.000000 MW

Converged! Loss difference < 0.001000
```

## Display final results

Final results

```
fprintf('\n=== FINAL RESULTS ===\n');
fprintf('Optimal lambda = %.6f\n', lambda);
for i = 1:N
    incremental_cost = 2 * a(i) * pg(i) + b(i);
    fprintf('Generator %d: Pg = %.4f MW, Incremental Cost = %.4f $/MWh\n', ...
            i, pg(i), incremental_cost);
end
fprintf('Total generation: %.4f MW\n', sum(pg));
fprintf('Total losses: %.4f MW\n', sum(ploss_old));
fprintf('Generation - Losses = %.4f MW\n', sum(pg) - sum(ploss_old));
```

```
fprintf('Demand = %.4f MW\n', pd);
fprintf('Power balance check: %.6f MW\n', sum(pg) - sum(ploss_old) - pd);
```

```
=== FINAL RESULTS ===
Optimal lambda = 2165.701340
Generator 1: Pg = 600.0000 MW, Incremental Cost = 8.9040 $/MWh
Generator 2: Pg = 400.0000 MW, Incremental Cost = 9.4020 $/MWh
Generator 3: Pg = 200.0000 MW, Incremental Cost = 9.2560 $/MWh
Total generation: 1200.0000 MW
Total losses: 68.0000 MW
Generation - Losses = 1132.0000 MW
Demand = 975.0000 MW
Power balance check: 157.000000 MW
```

## Calculate total generation cost

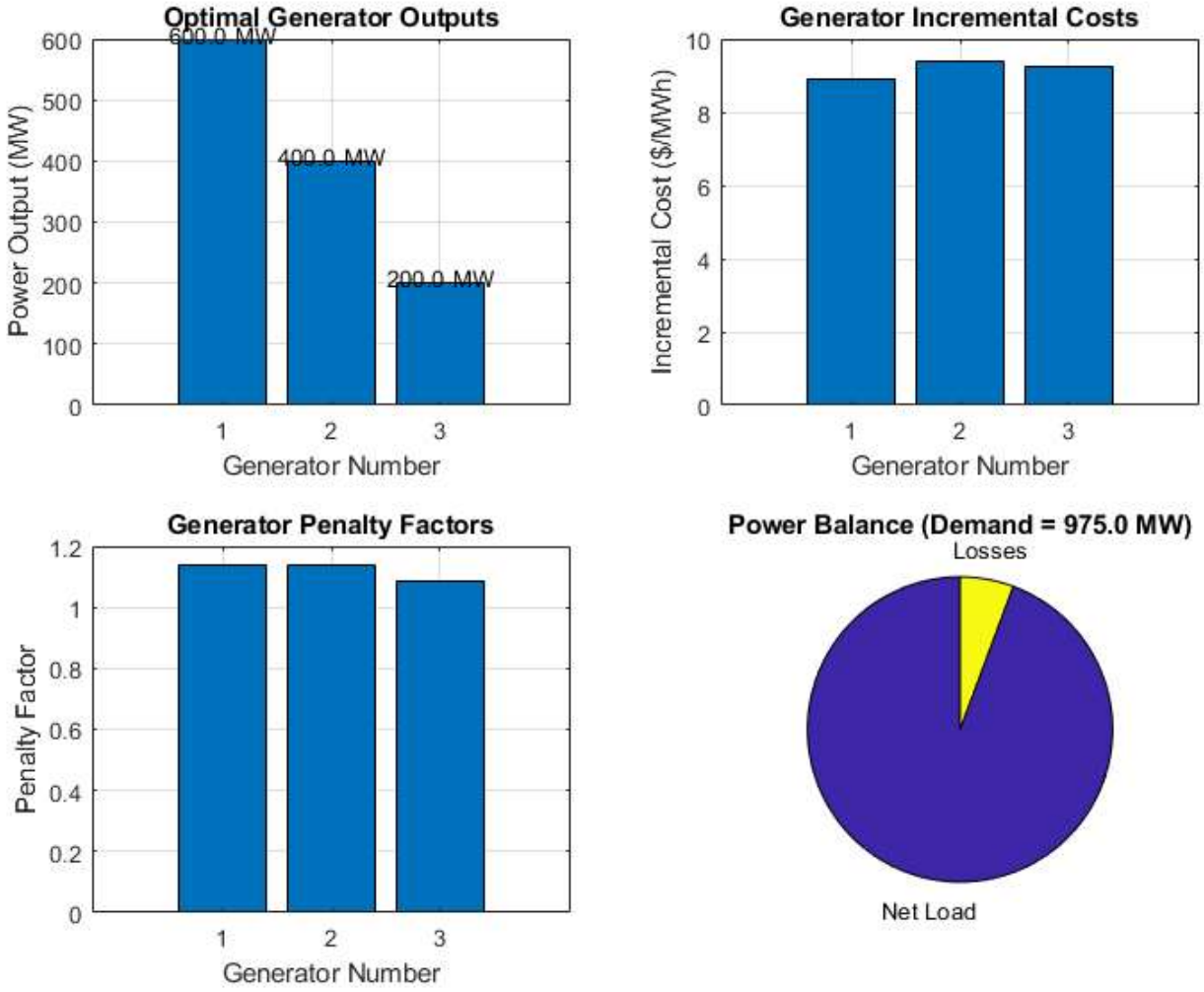
```
total_cost = 0;
for i = 1:N
    % Cost function: a*P^2 + b*P + c
    gen_cost = a(i)*(pg(i)^2) + b(i)*pg(i) + c(i);
    total_cost = total_cost + gen_cost;
end
fprintf('Total generation cost: %.2f $/h\n', total_cost);
```

```
Total generation cost: 11174.20 $/h
```

## Plot results

```
figure('Name', 'Economic Load Dispatch Results', 'Position', [100, 100, 800, 600]);

% Generator outputs
subplot(2, 2, 1);
bar(pg);
grid on;
xlabel('Generator Number');
ylabel('Power Output (MW)');
title('Optimal Generator Outputs');
xticks(1:N);
for i = 1:N
    text(i, pg(i)+10, sprintf('%.1f MW', pg(i)), 'HorizontalAlignment', 'center');
end

% Incremental costs
subplot(2, 2, 2);
inc_costs = zeros(N, 1);
for i = 1:N
    inc_costs(i) = 2 * a(i) * pg(i) + b(i);
end
bar(inc_costs);
grid on;
xlabel('Generator Number');
ylabel('Incremental Cost ($/MWh)');
title('Generator Incremental Costs');
xticks(1:N);

% Penalty factors
subplot(2, 2, 3);
pf = 1 ./ (1 - 2 * pg .* ploss_coeff); % Update penalty factors with final values
bar(pf);
```

```
grid on;
xlabel('Generator Number');
ylabel('Penalty Factor');
title('Generator Penalty Factors');
xticks(1:N);

% Power balance
subplot(2, 2, 4);
pie([sum(pg)-sum(ploss_old), sum(ploss_old)], {'Net Load', 'Losses'});
title(sprintf('Power Balance (Demand = %.1f MW)', pd));

% Save figures
saveas(gcf, 'ELD_Results.fig');
saveas(gcf, 'ELD_Results.png');
```




## Export results to Excel

```
results_table = table((1:N)', pg, inc_costs, pf, ...
                      'VariableNames', {'Generator', 'Power_MW', 'Incremental_Cost', 'Penalty_Factor'});
writetable(results_table, 'ELD_Results.xlsx', 'Sheet', 'Generator_Results');

fprintf('\nResults saved to ELD_Results.xlsx\n');

% Vraj did it
```

```
Results saved to ELD_Results.xlsx
```

---

*Published with MATLAB® R2024b*