

INDIAN INSTITUTE OF TECHNOLOGY INDORE DEPARTMENT OF CHEMICAL ENGINEERING

INTRODUCTION TO SOFT MATTER & POLYMER PROJECT REPORT

Submitted to:
Dr. Kailasham Ramalingam

Submitted by:

Group 2

Ayush Varun	240008007
Bothumanchi Praneeth	240008009
Dhanuka Samarth Rupesh	240008011
Garvit Mathur	240008012
Jayshil Mendhe	240008013
Kavyansh Raj Singh	240008014

Autumn Semester, 2025



Contents

1 Declaration	2
2 Theoretical framework from the Paper	3
3 Case I: Our Polymer System	4
3.1 Normalized Laplacian Matrix	4
3.2 Eigenvalue Spectrum	4
3.3 Results and Theoretical Validation	5
4 Case II: Reference Tree Polymer System	6
4.1 Normalized Laplacian Matrix	6
4.2 Eigenvalue Spectrum	7
4.3 Results and Theoretical Validation	7
5 Conclusion	7
6 Appendix	8
6.1 MATLAB Implementation for Our Polymer	8
6.2 MATLAB Implementation for Reference Tree Polymer	9



1 Declaration

We hereby declare that this project we did, along with this report on the theoretical derivation, is part of the Course Project in partial fulfillment of the requirements for the course ChE – 209: Introduction to Soft Matter and Polymer. This work represents an authentic expression of our efforts and is our own creation. We have drawn inspiration and help from various Artificial Intelligence sources and conducted thorough research, ensuring that all information presented is accurate to the best of our knowledge.

Bothumanchi Praneeth
240008009

Ayush Varun
240008007

Dhanuka Samarth Rupesh
240008011

Kavyansh Raj Singh
240008014

Garvit Mathur
240008012

Jayshil Mendhe
240008013

2 Theoretical framework from the Paper

Definition 2.1 (Normalized Laplacian) *The normalized Laplacian matrix $L(G)$ of a graph G is defined as:*

$$L_{ij} = \begin{cases} 1, & \text{if } i = j, \\ -\frac{A_{ij}}{\sqrt{\deg(v_i)\deg(v_j)}}, & \text{if } (i, j) \in E(G), \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Theorem 2.1 (Theorem 5) *For any connected multigraph G (including loops and multiple edges), let G_n be the graph obtained by dividing each edge of G into n equal parts. Then, the contraction factor satisfies:*

$$\lim_{n \rightarrow \infty} g(G_n) = \frac{3}{e(G)^2} \left(\text{Tr}(L^+(G)) + \frac{\text{Loops}(G)}{3} - \frac{1}{6} \right), \quad (2)$$

where:

- $g(G_n)$ is the g -factor of G_n ,
- $\text{Loops}(G) = e(G) - v(G) + 1$ is the cycle rank (or cyclomatic number),
- $L(G)$ is the normalized graph Laplacian of G ,
- $L^+(G)$ is the MoorePenrose pseudoinverse of $L(G)$.

Definition 2.2 (Normalized Graph Laplacian for Multigraphs) *For a multigraph G (possibly with loop edges and multiple edges), the normalized Laplacian $L(G)$ is given by:*

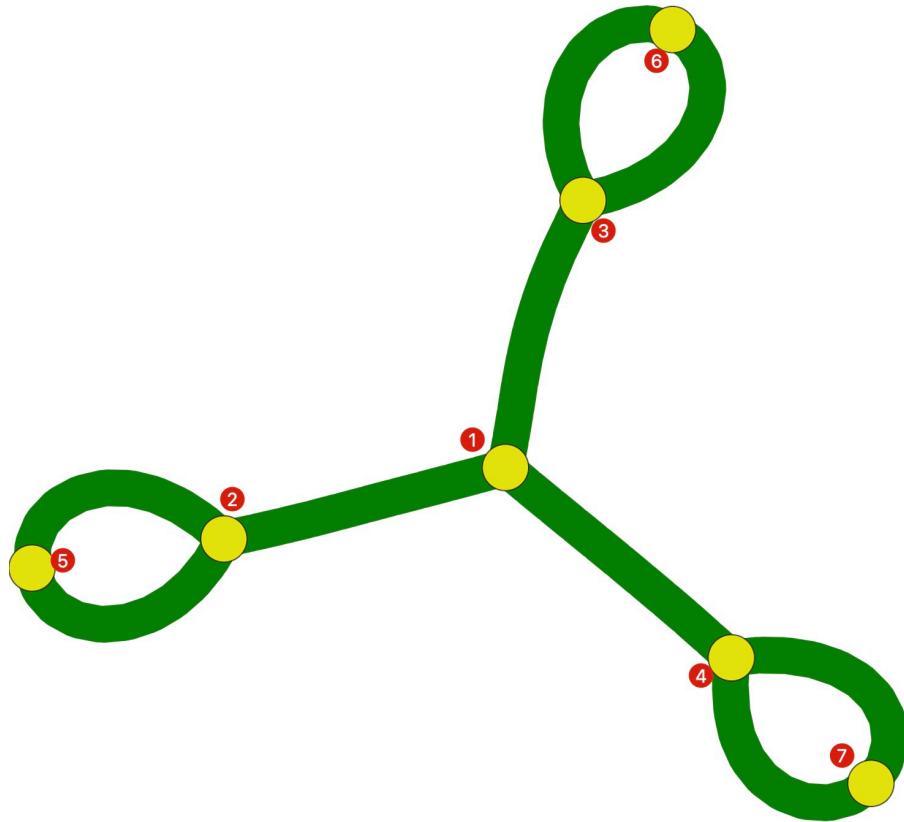
$$L_{ij} = \begin{cases} 1 - 2 \frac{\#(\text{loop edges at } v_i)}{\deg(v_i)}, & \text{if } i = j, \\ -\frac{k}{\sqrt{\deg(v_i)\deg(v_j)}}, & \text{if } v_i, v_j \text{ are joined by } k \text{ edges,} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Definition 2.3 (Relative Contraction Factor) *The relative contraction factor of two polymer graphs G_1 and G_2 is defined as:*

$$g(G_1, G_2) = \frac{\mathbb{E}(R_g^2; G_1)}{\mathbb{E}(R_g^2; G_2)} = \frac{g(G_1)}{g(G_2)}, \quad (4)$$

where $\mathbb{E}(R_g^2; G)$ denotes the expected mean-square radius of gyration for polymer graph G .

3 Case I: Our Polymer System



3.1 Normalized Laplacian Matrix

The normalized Laplacian $L(G)$ for our 7-vertex polymer (comprising 9 edges, 3 internal loops, and one large enclosing loop) is given by:

$$L(G) = \begin{bmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & -\frac{2}{\sqrt{6}} & 0 & 0 \\ -\frac{1}{3} & 0 & 1 & 0 & 0 & -\frac{2}{\sqrt{6}} & 0 \\ -\frac{1}{3} & 0 & 0 & 1 & 0 & 0 & -\frac{2}{\sqrt{6}} \\ 0 & -\frac{2}{\sqrt{6}} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{2}{\sqrt{6}} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{2}{\sqrt{6}} & 0 & 0 & 1 \end{bmatrix}.$$

3.2 Eigenvalue Spectrum

The eigenvalue spectrum of $L(G)$, obtained from the MATLAB diagonalization, is:

$$\lambda_i = [0.0000, 0.1835, 0.1835, 1.0000, 1.8165, 1.8165, 2.0000].$$

Since the polymer graph is connected, the zero eigenvalue corresponds to the uniform eigenvector. The reciprocals of the nonzero eigenvalues, which contribute to the pseudoinverse trace, are:

$$\frac{1}{\lambda_i} = [5.4495, 5.4495, 1.0000, 0.5505, 0.5505, 0.5000].$$

3.3 Results and Theoretical Validation

The trace of the pseudoinverse L^+ is computed from the nonzero eigenvalues as:

$$\text{Tr}(L^+) = \frac{1}{0.1835} + \frac{1}{0.1835} + \frac{1}{1.0000} + \frac{1}{1.8165} + \frac{1}{1.8165} + \frac{1}{2.0000} = 13.5000.$$

Thus,

$$\boxed{\text{Tr}(L^+) = 13.5.}$$

For our polymer graph:

$$v = 7, \quad e = 9, \quad \ell = e - v + 1 = 3.$$

Using Theorem 5, the asymptotic contraction factor is:

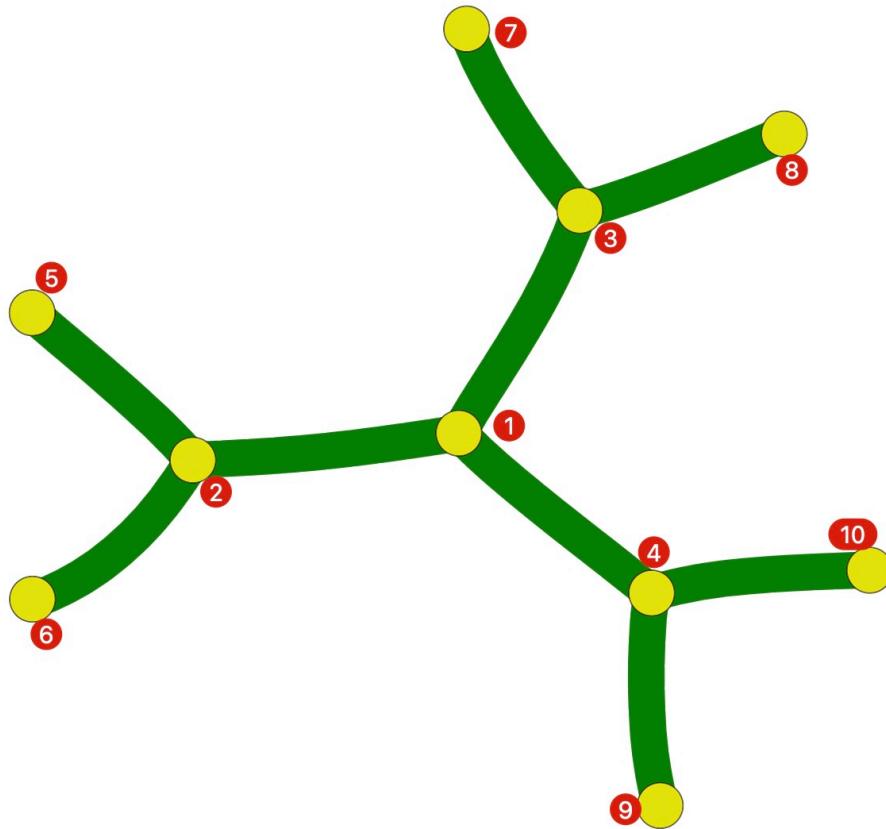
$$g(G_\infty) = \frac{3}{e^2} \left(\text{Tr}(L^+) + \frac{\ell}{3} - \frac{1}{6} \right).$$

Substituting the numerical values:

$$\begin{aligned} g(G_\infty) &= \frac{3}{9^2} \left(13.5 + \frac{3}{3} - \frac{1}{6} \right) \\ &= \frac{3}{81} (13.5 + 1 - 0.1667) \\ &= \frac{3}{81} (14.3333) \\ &= 0.530864197530865 \end{aligned}$$

$$\boxed{g(G_\infty)_{\text{our}} = 0.530864197530865}$$

This theoretical derivation exactly matches the MATLAB output, confirming internal consistency between the spectral computation and the analytical formulation of the Laplacian-based contraction factor.



4 Case II: Reference Tree Polymer System

4.1 Normalized Laplacian Matrix

The normalized Laplacian $L(G)$ for the reference 10-vertex polymer (with 9 edges and no loops) is:

$$L = \begin{bmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 0 & 1 & 0 & 0 & 0 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & 0 & 0 \\ -\frac{1}{3} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.2 Eigenvalue Spectrum

The eigenvalue spectrum of the normalized Laplacian $L(G)$ is:

$$\lambda_i = [0.0000, 0.1835, 0.1835, 1.0000, 1.0000, 1.0000, 1.0000, 1.8165, 1.8165, 2.0000].$$

The reciprocals of the nonzero eigenvalues (used for L^+) are:

$$\frac{1}{\lambda_i} = [5.4495, 5.4495, 1.0000, 1.0000, 1.0000, 1.0000, 0.5505, 0.5505, 0.5000].$$

The pseudoinverse trace is therefore

$$\text{Tr}(L^+) = \sum_{i=2}^{10} \frac{1}{\lambda_i} = 16.5.$$

Hence,

$$\boxed{\text{Tr}(L^+) = 16.5.}$$

4.3 Results and Theoretical Validation

For the reference polymer:

$$v = 10, \quad e = 9, \quad \ell = e - v + 1 = 0.$$

From Theorem 5,

$$g(G_\infty) = \frac{3}{e^2} \left(\text{Tr}(L^+) + \frac{\ell}{3} - \frac{1}{6} \right).$$

Substituting:

$$\begin{aligned} g(G_\infty) &= \frac{3}{9^2} \left(16.5 + 0 - \frac{1}{6} \right) \\ &= \frac{3}{81} (16.3333) \\ &= 0.604938271604938 \end{aligned}$$

$$\boxed{g(G_\infty)_{\text{ref}} = 0.604938271604938}$$

The theoretical evaluation perfectly matches the MATLAB output, confirming the spectral consistency and correctness of the graph-theoretic formulation.

5 Conclusion

Using Definition 13 of the Paper, the relative contraction factor between the polymer graph and its reference structure is obtained as

$$\begin{aligned} g &= \frac{g(G_\infty)_{\text{our}}}{g(G_\infty)_{\text{ref}}} = \frac{0.530864197530865}{0.604938271604938} = 0.87755102040816498583 \\ &= \frac{43.00000000000008430567}{49} \end{aligned}$$

This value quantitatively expresses the degree of configurational contraction of our polymer relative to the reference topology.

6 Appendix

6.1 MATLAB Implementation for Our Polymer

```

>> % === High-precision computation of g(G) for 7-vertex tree polymer ===
format long g % show full double-precision output (~15 digits)

% Normalized Laplacian (your 7x7 matrix)
L = [ 1,      -1/3,      -1/3,      -1/3,      0,      0,      0;
      -1/3,      1,      0,      0,      -2/sqrt(6),      0,      0;
      -1/3,      0,      1,      0,      0,      -2/sqrt(6),      0;
      -1/3,      0,      0,      1,      0,      0,      -2/sqrt(6);
      0,      -2/sqrt(6),      0,      0,      1,      0,      0;
      0,      0,      -2/sqrt(6),      0,      0,      1,      0;
      0,      0,      0,      -2/sqrt(6),      0,      0,      1 ];

disp('Normalized Laplacian L = ');
disp(L);

% Eigen-decomposition
[eigvecs, eigvals_matrix] = eig(L);
eigvals = sort(real(diag(eigvals_matrix)));

disp('Eigenvalues _i (sorted):');
disp(eigvals.']);

% Remove the zero eigenvalue (tree is connected)
eigvals_nonzero = eigvals(abs(eigvals) > 1e-12);

% Reciprocals for pseudoinverse trace
recip = 1 ./ eigvals_nonzero;
disp('Reciprocals 1/_i (for nonzero _i):');
disp(recip.'];

% Trace of pseudoinverse
tr_Lplus = sum(recip);
fprintf('\nTrace(L) = %.15f\n', tr_Lplus);

% Graph parameters
v = size(L,1);
e = 9; % total edges
loops = e - v + 1; % cycle rank (should be 3 or 4 depending on structure)

fprintf('Vertices (v) = %d\n', v);
fprintf('Edges (e) = %d\n', e);
fprintf('Loops (e - v + 1) = %d\n', loops);

% Theorem 5 formula
gG = (3 / e^2) * (tr_Lplus + loops/3 - 1/6);
fprintf('\n g(G) = (3/e^2) * (Tr(L) + Loops/3 - 1/6)\n');
fprintf(' g(G) = %.15f\n', gG);

```

Listing 1: MATLAB code for our polymer

6.2 MATLAB Implementation for Reference Tree Polymer

```
% === High-precision computation of g(G) for 10-vertex reference polymer (9
edges) ===
format long g % show full double precision (~15 digits)

% Normalized Laplacian (10x10 matrix)
L = [ 1, -1/3,      -1/3,      -1/3,      0,          0,          0,          0,
        0,          0;
      -1/3, 1,          0,          0,          -1/sqrt(3), -1/sqrt(3), 0, 0,
        0,          0;
      -1/3, 0,          1,          0,          0,          0,          -1/sqrt(3), -1/sqrt
(3), 0, 0;
      -1/3, 0,          0,          1,          0,          0,          0,          0,
        -1/sqrt(3), -1/sqrt(3);
      0,  -1/sqrt(3), 0,          0,          1,          0,          0,          0,
        0,          0;
      0,  -1/sqrt(3), 0,          0,          0,          1,          0,          0,
        0,          0;
      0,  0,          -1/sqrt(3), 0,          0,          0,          1,          0,
        0,          0;
      0,  0,          0,          -1/sqrt(3), 0,          0,          0,          1,
        0,          0;
      0,  0,          1,          0,          -1/sqrt(3), 0,          0,          0,
        0,          1 ];
      0,          1 ];

% Eigen-decomposition
[eigvecs, eigvals_matrix] = eig(L);
eigvals = sort(real(diag(eigvals_matrix)));

disp('Eigenvalues _i (sorted):');
disp(eigvals.);

% Remove the zero eigenvalue (connected graph)
eigvals_nonzero = eigvals(abs(eigvals) > 1e-12);

% Reciprocals for pseudoinverse trace
recip = 1 ./ eigvals_nonzero;
disp('Reciprocals 1/_i (for nonzero _i):');
disp(recip.);

% Trace of pseudoinverse
tr_Lplus = sum(recip);
fprintf('\nTrace(L) = %.15f\n', tr_Lplus);

% Graph parameters
v = size(L,1);
e = 9;
loops = e - v + 1; % should be 0 for tree

fprintf('Vertices (v) = %d\n', v);
fprintf('Edges (e) = %d\n', e);
fprintf('Loops (e - v + 1) = %d\n', loops);
```

```
% Theorem 5 formula  
gG = (3 / e^2) * (tr_Lplus + loops/3 - 1/6);  
fprintf('\n g(G) = (3/e^2)*(Tr(L) + Loops/3 - 1/6)\n');  
fprintf(' g(G) = %.15f\n', gG);
```

Listing 2: MATLAB code for the reference polymer

