

Localized solutions of sparse linear systems for geometry processing


Philipp Herholz, Timothy A. Davis, Marc Alexa

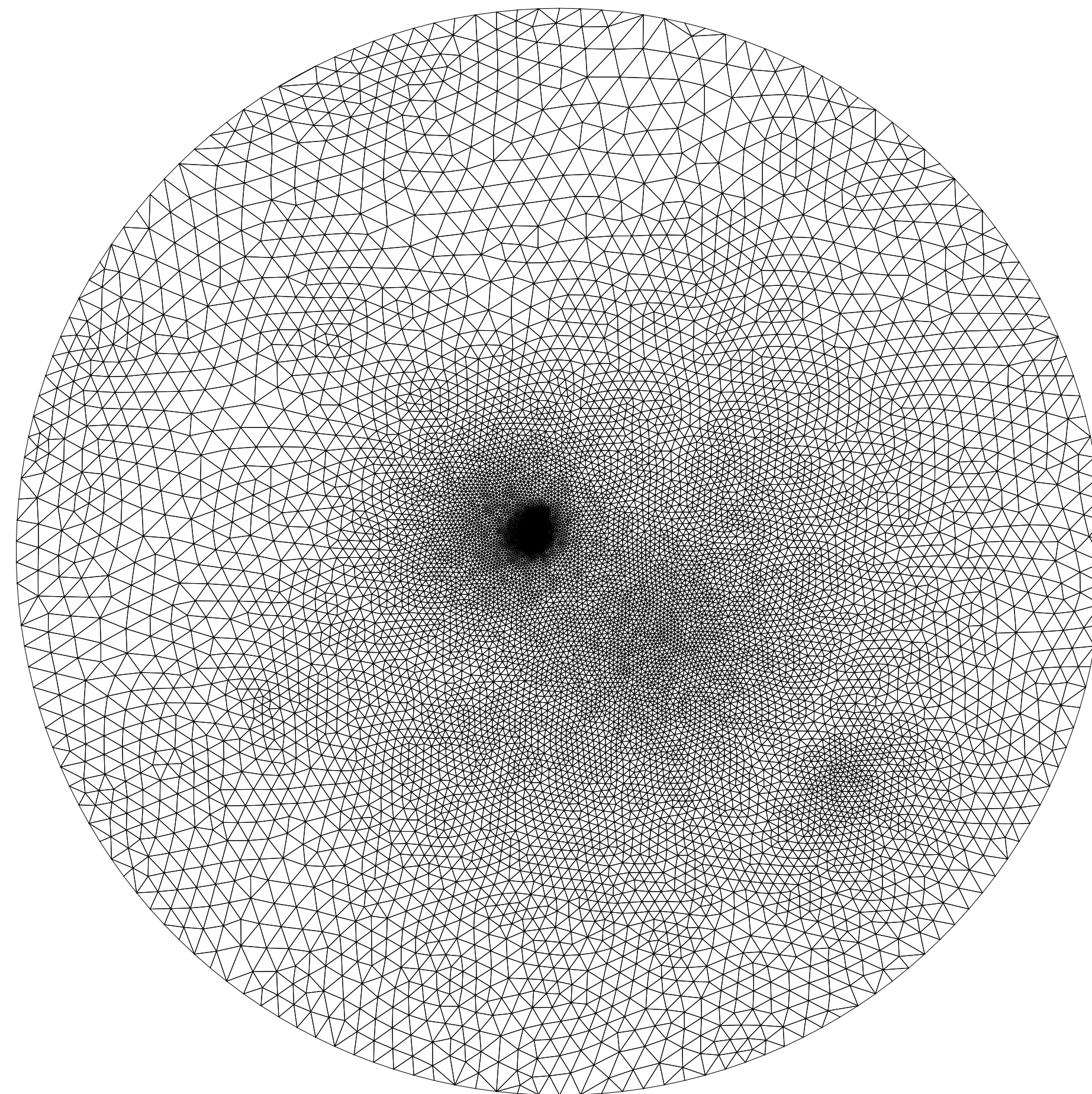
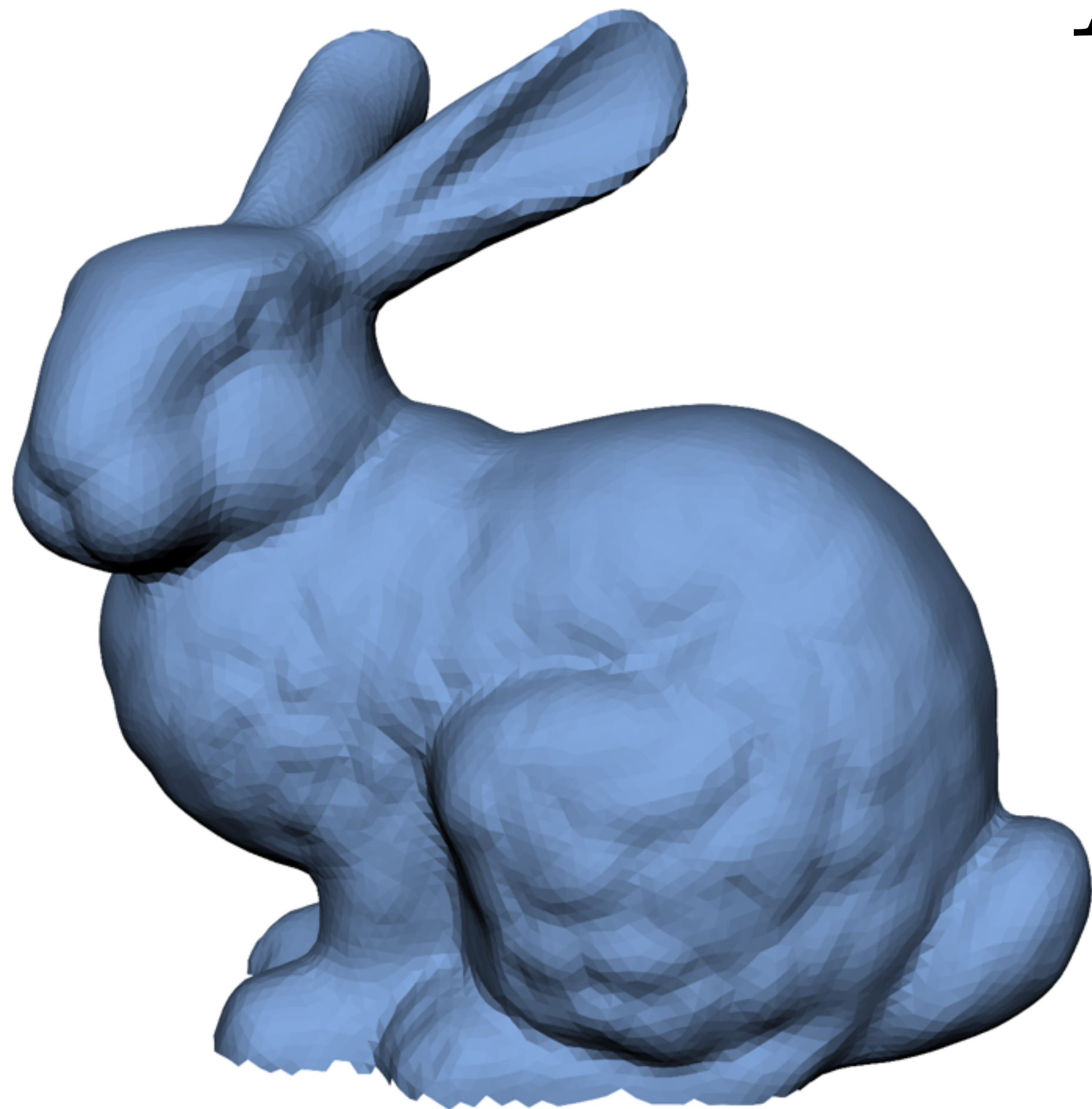
Linear solvers in geometry processing

- Many applications require repeated solutions of linear systems.



Discrete conformal parameterization


$$\mathbf{Ax} = \mathbf{b}$$

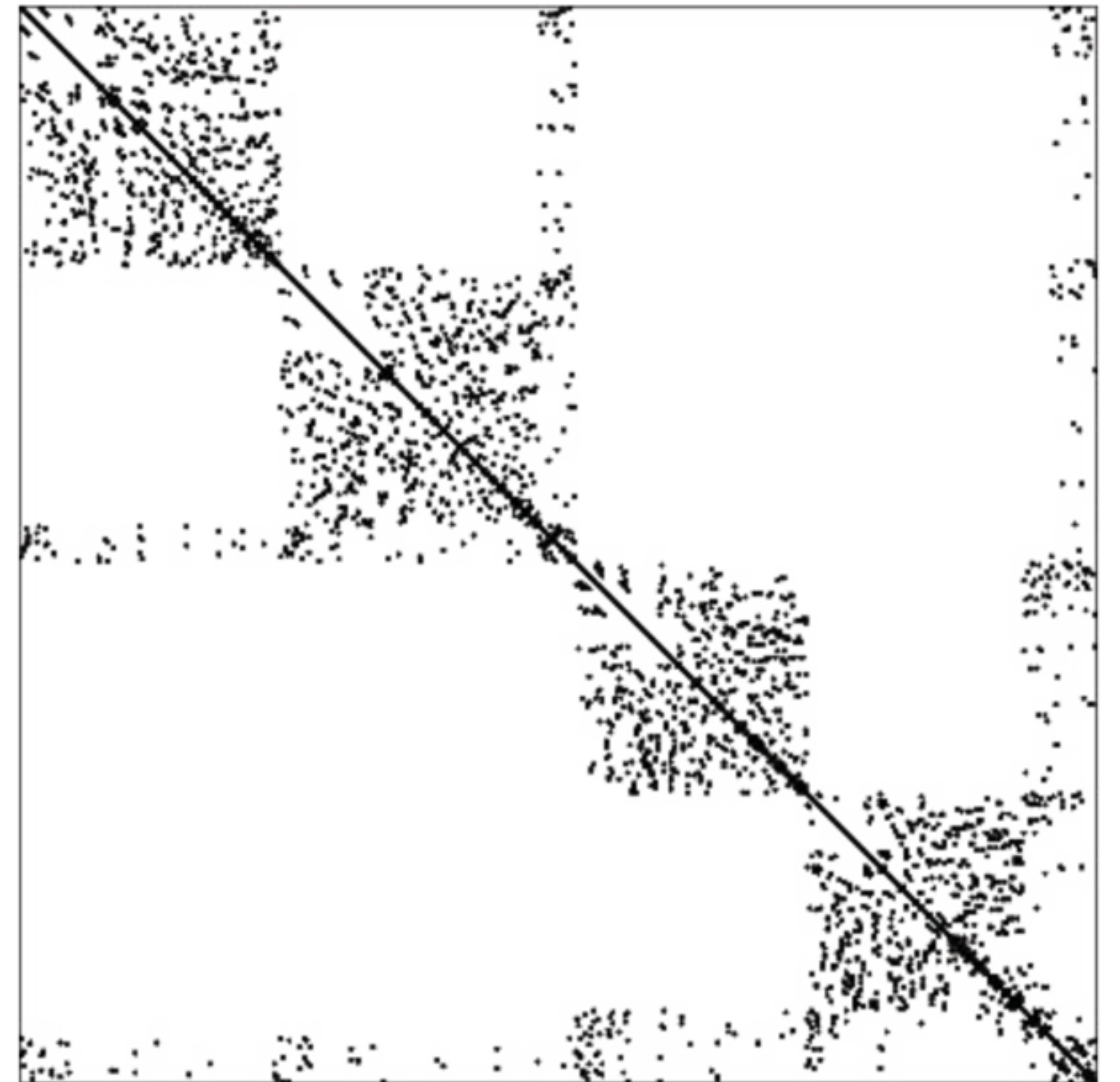


Discrete conformal parameterization

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

\mathbf{A} is typically:

- sparse
- symmetric
- positive semi-definite



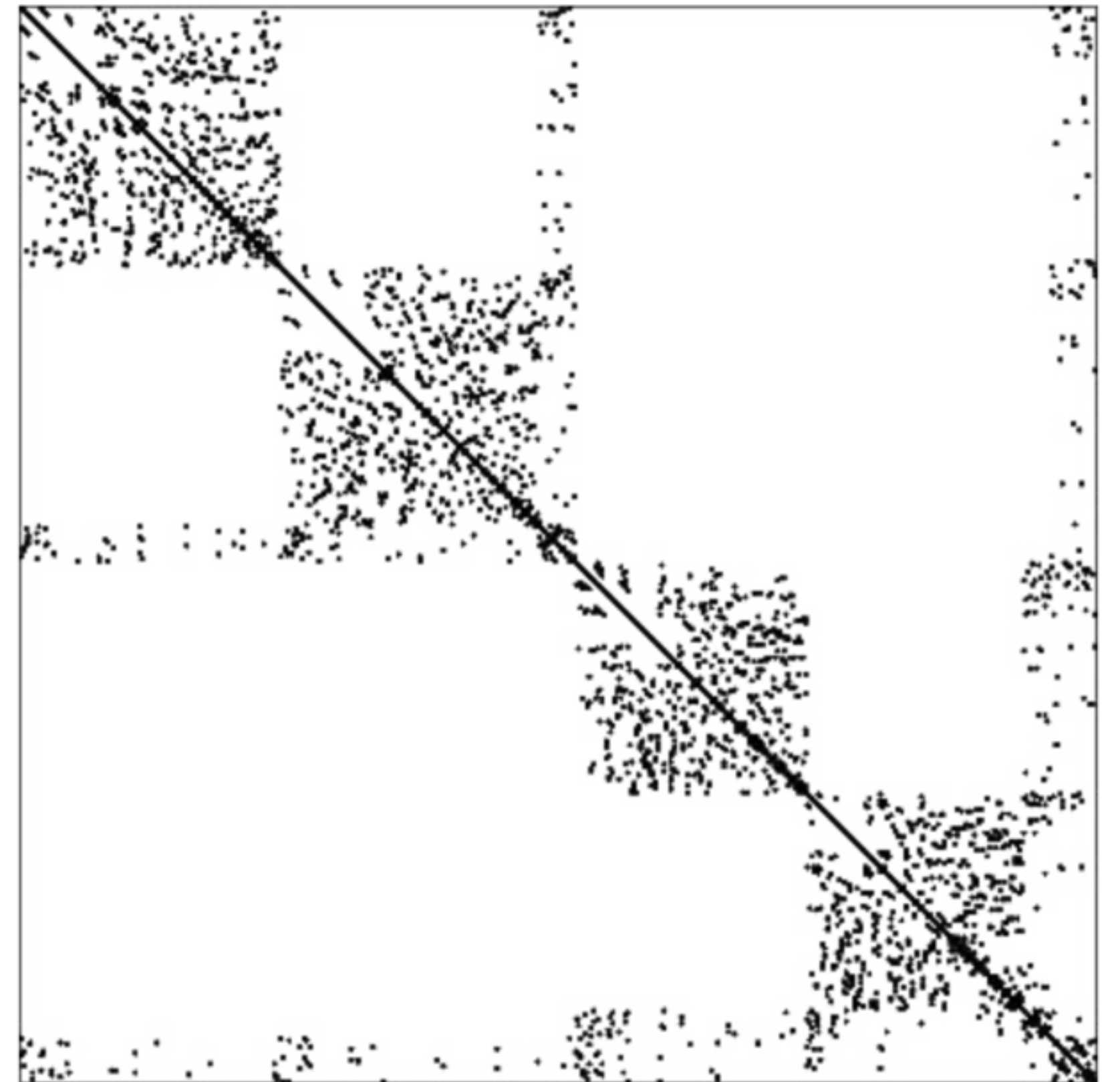
Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

\mathbf{A} is typically:

- sparse
- symmetric
- positive semi-definite

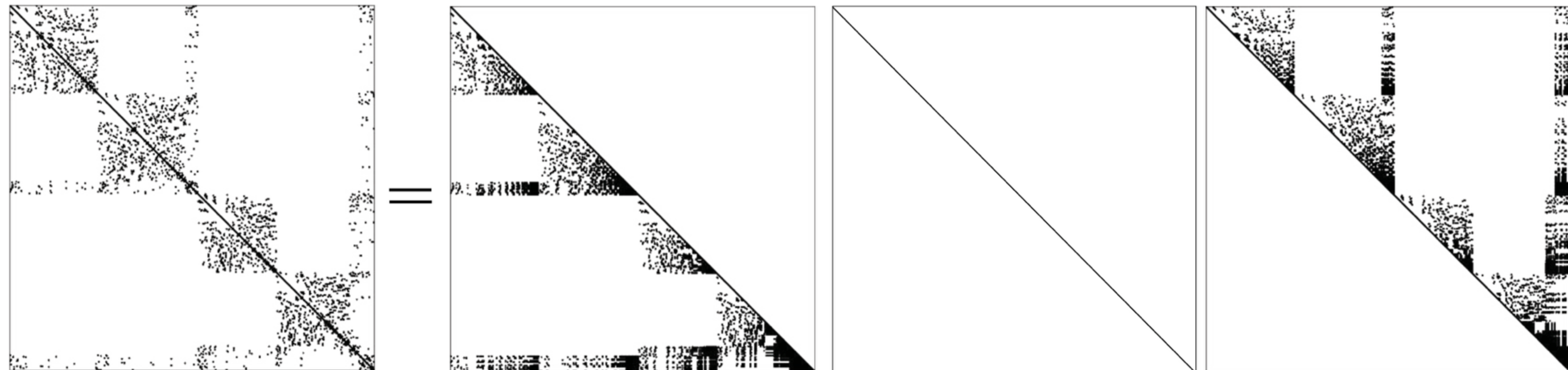
Sparse Cholesky factorization can be applied!



Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A} = \mathbf{L} \times \mathbf{D} \times \mathbf{L}^T$$



Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{LDL}^T \mathbf{x} = \mathbf{b}$$

Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{LD} \underbrace{\mathbf{L}^T \mathbf{x}}_y = \mathbf{b}$$

Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{LD} \underbrace{\mathbf{L}^T \mathbf{x}}_{\mathbf{y}} = \mathbf{b}$$

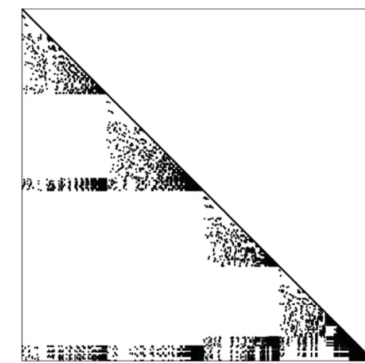
$$\mathbf{LDy} = \mathbf{b} \quad (1) \text{Forward solve}$$

$$\mathbf{L}^T \mathbf{x} = \mathbf{y} \quad (2) \text{Back solve}$$

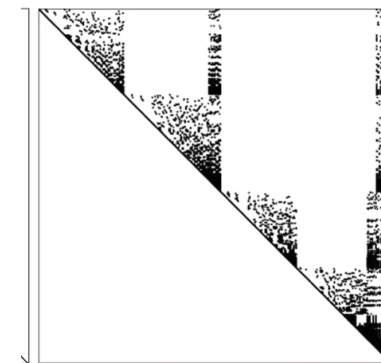
Cholesky factorization

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{LD} \underbrace{\mathbf{L}^T \mathbf{x}}_{\mathbf{y}} = \mathbf{b}$$

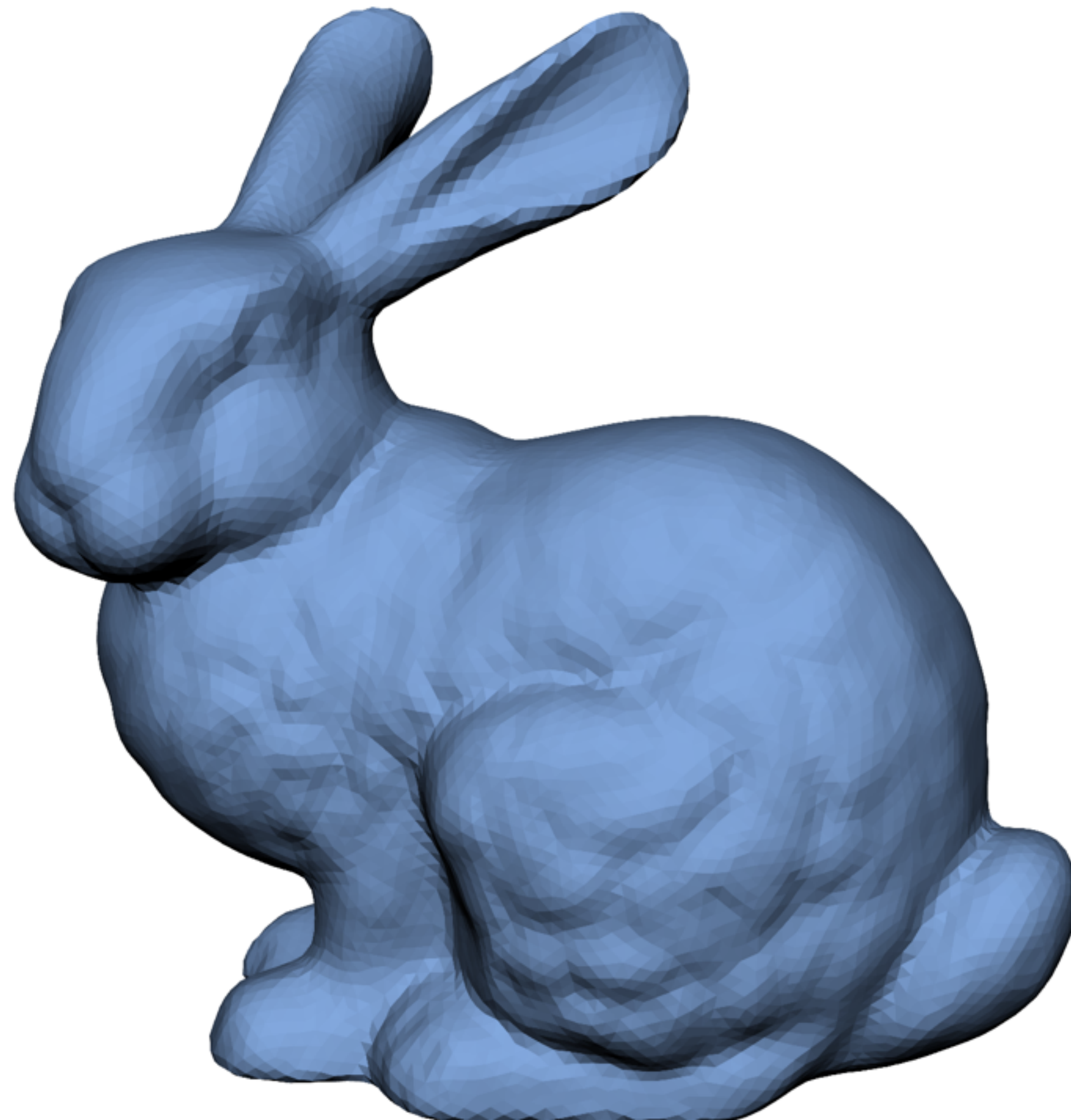


$$\mathbf{y} = \mathbf{b} \quad (1) \text{Forward solve}$$

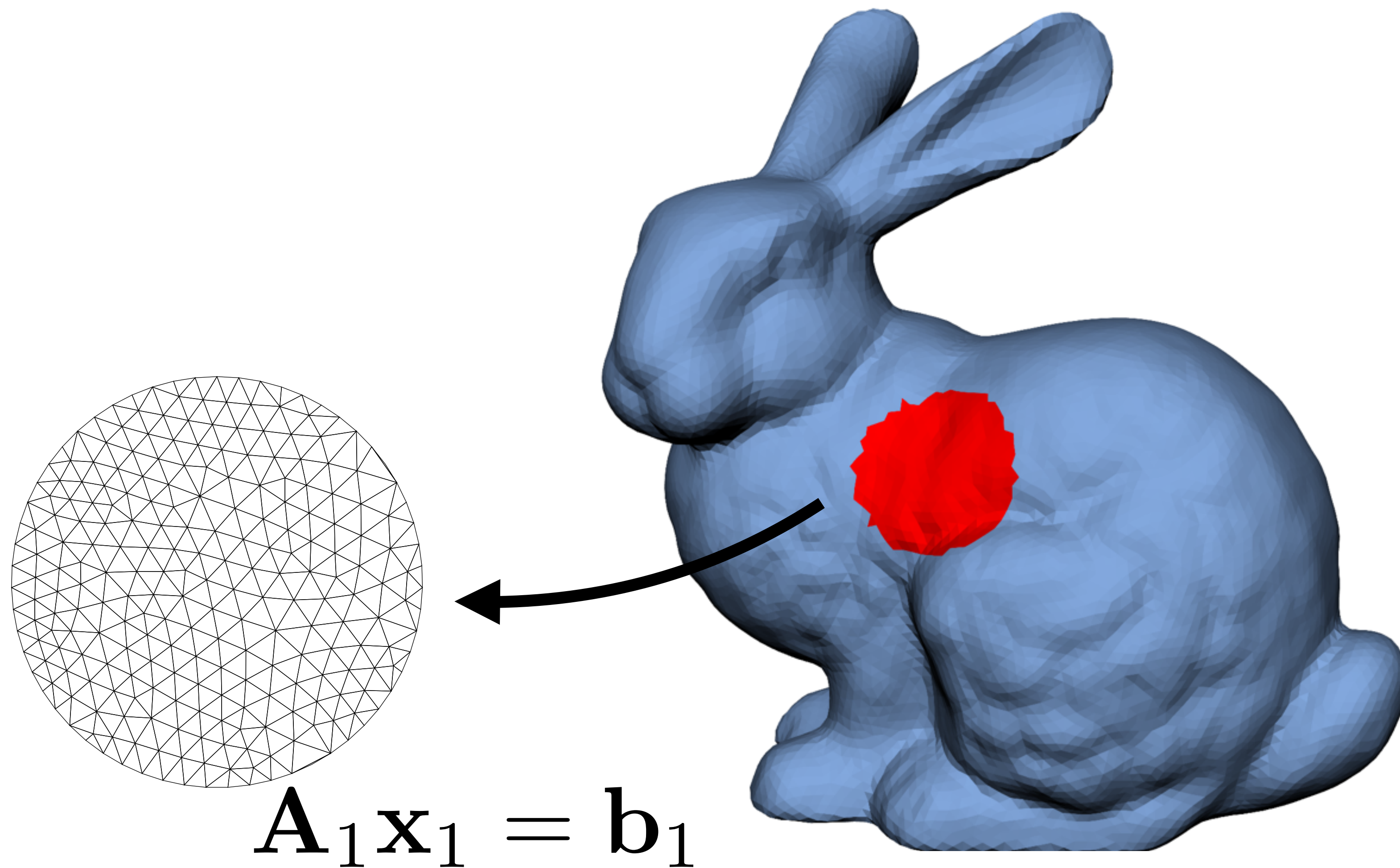


$$\mathbf{x} = \mathbf{y} \quad (2) \text{Back solve}$$

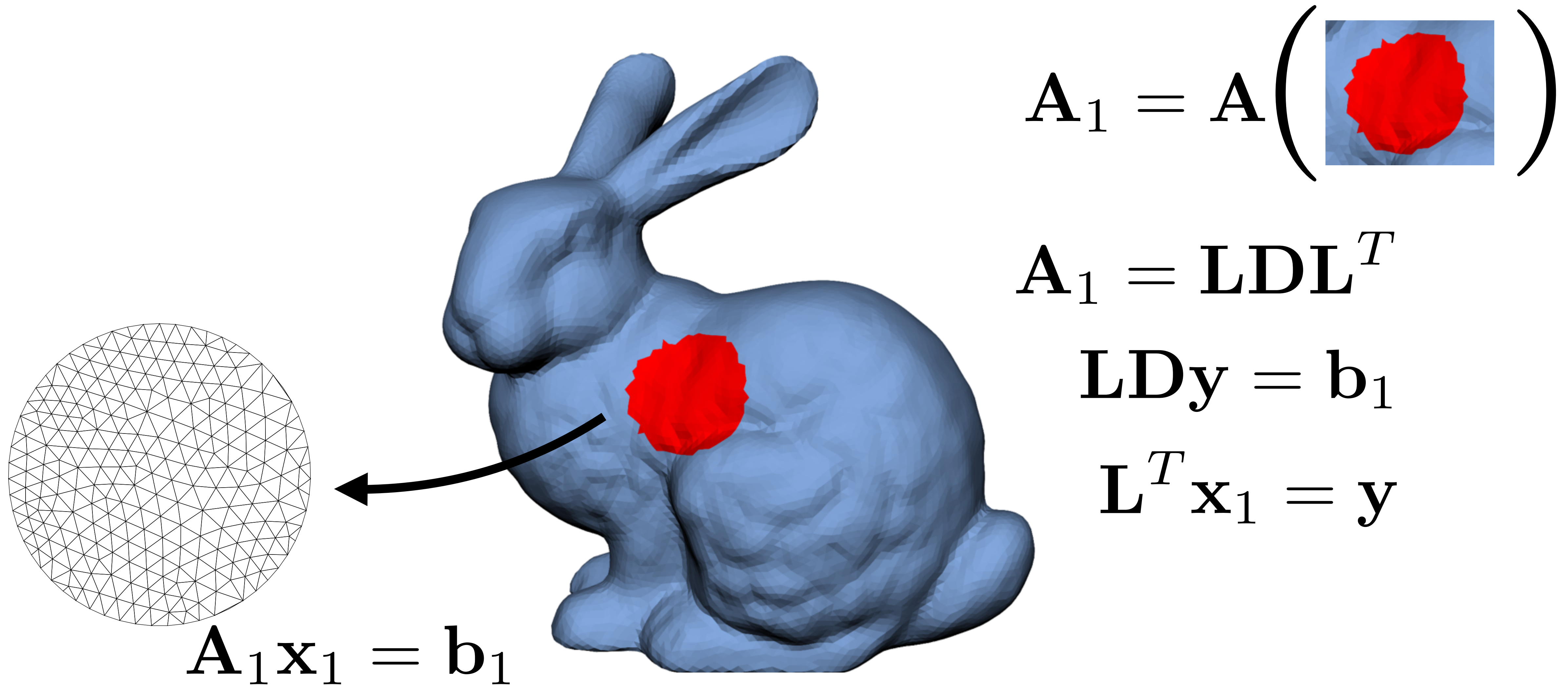
Local discrete conformal parameterization



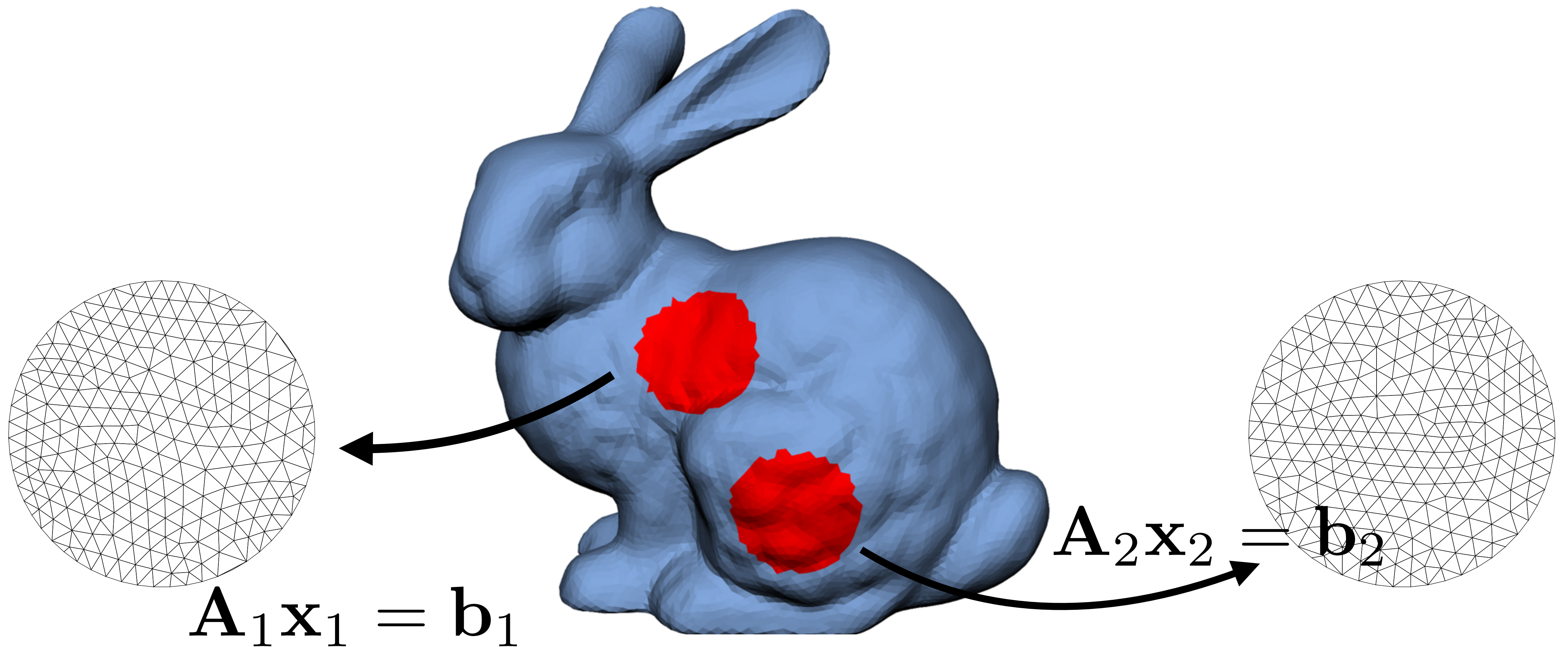
Local discrete conformal parameterization



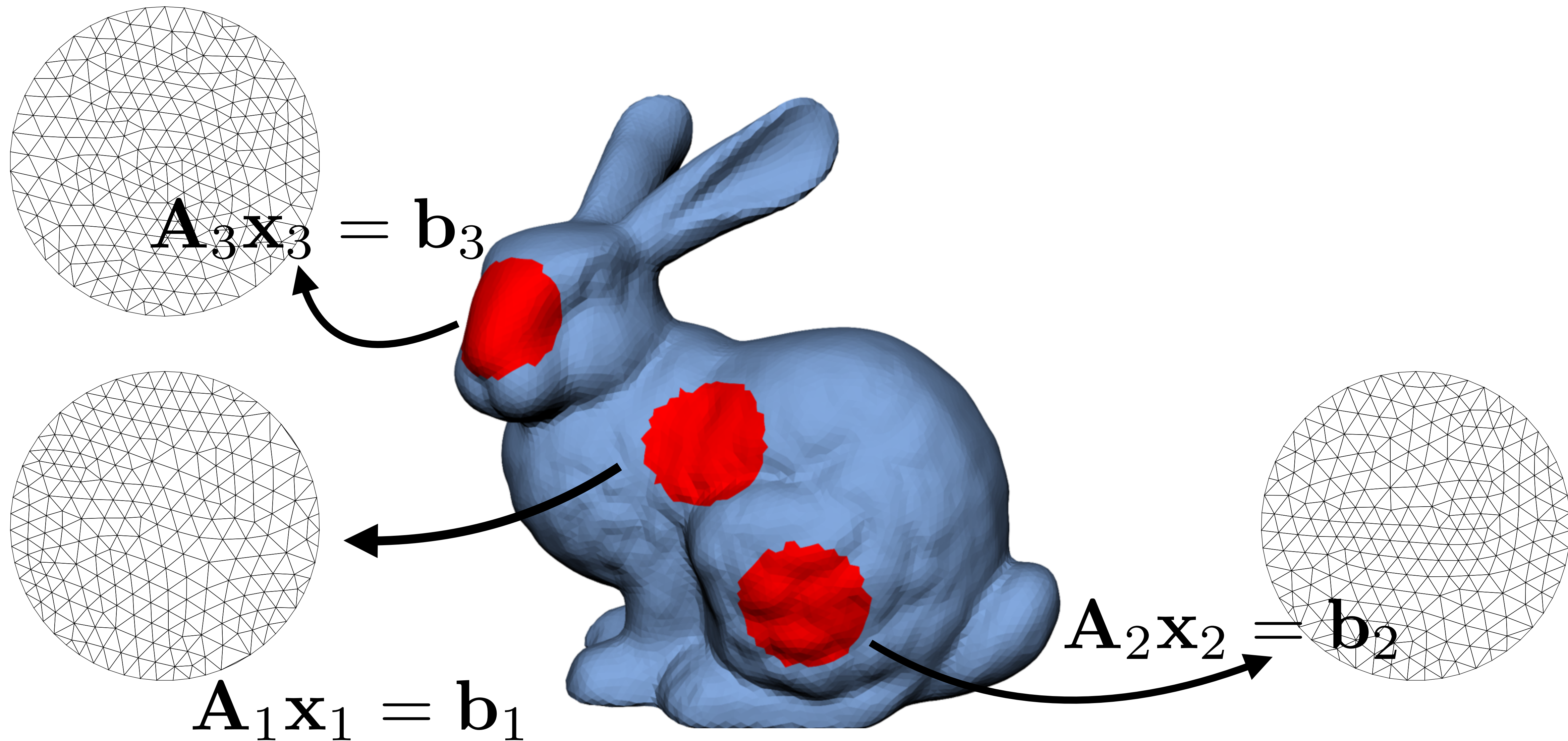
Local discrete conformal parameterization



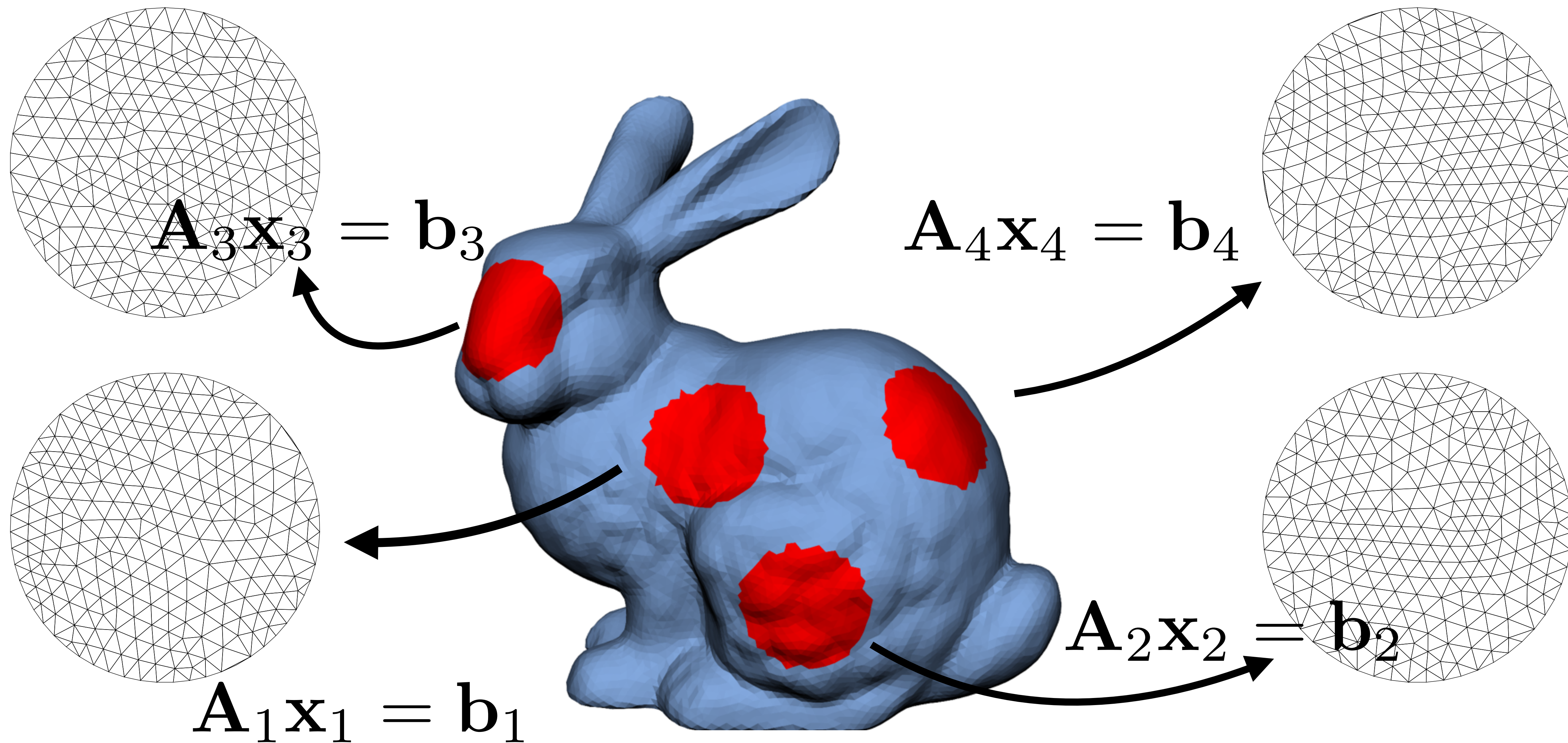
Local discrete conformal parameterization



Local discrete conformal parameterization

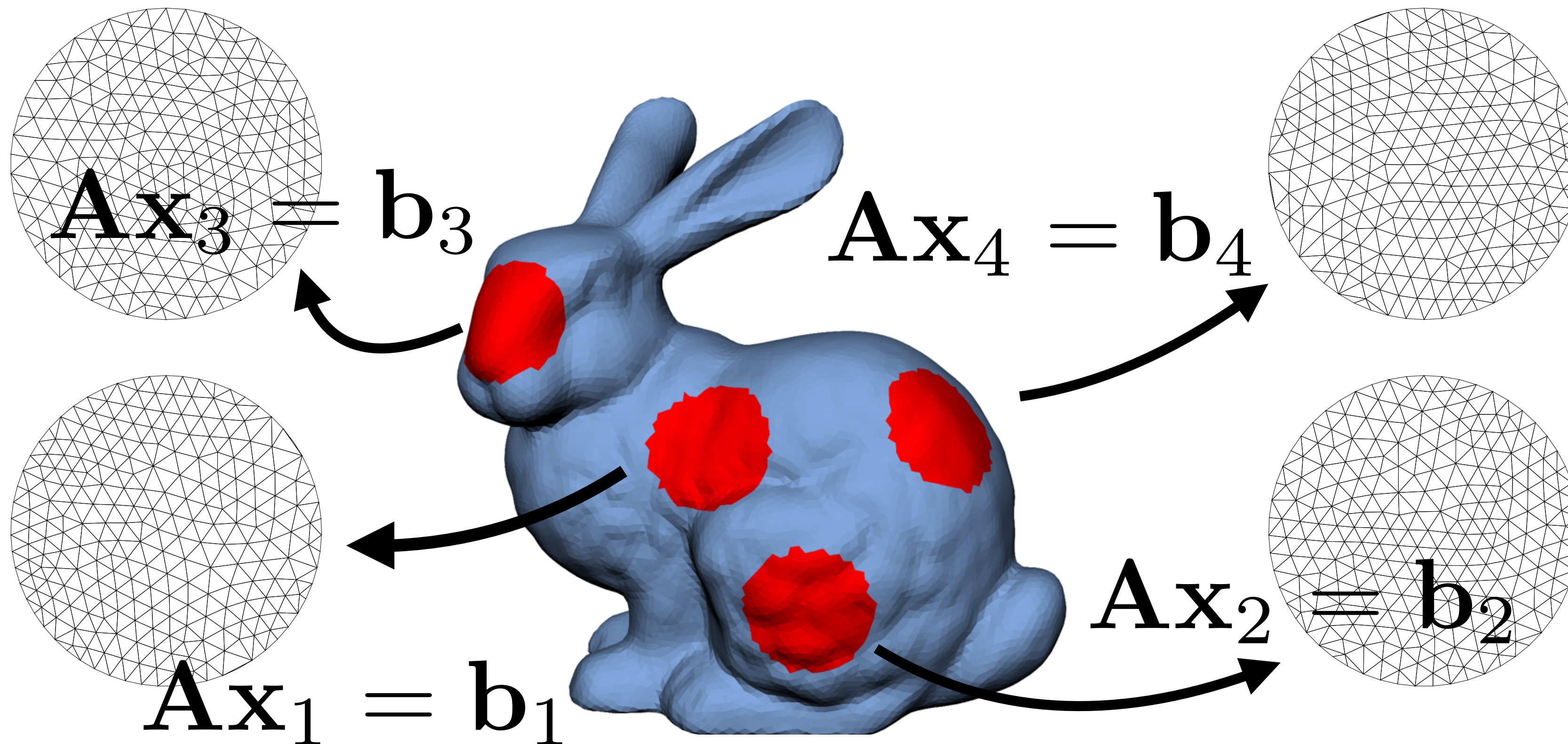


Local discrete conformal parameterization



Contribution

- Idea: Reuse a global factorization: $\mathbf{A} \left(\text{rabbit} \right) = \mathbf{L} \mathbf{D} \mathbf{L}^T$



Contribution

- Question: Can we quickly compute subset of solution?

The diagram illustrates the concept of a subset of the solution in the context of matrix factorization. It shows the equation $\mathbf{A} \begin{pmatrix} \text{Rabbit} \end{pmatrix} = \mathbf{LDL}^T \begin{pmatrix} \text{Rabbit} \end{pmatrix}$. The matrix \mathbf{A} is represented by a blue rabbit model. The vector $\begin{pmatrix} \text{Rabbit} \end{pmatrix}$ is represented by a vertical bar with three red segments. The matrix \mathbf{LDL}^T is represented by a vertical bar with three red segments and a thick black bar. The vector $\begin{pmatrix} \text{Rabbit} \end{pmatrix}$ is also represented by a vertical bar with three red segments. Arrows point from the red segments of the vector to the corresponding red segments of the matrix \mathbf{LDL}^T , indicating that the subset of the solution is being computed.

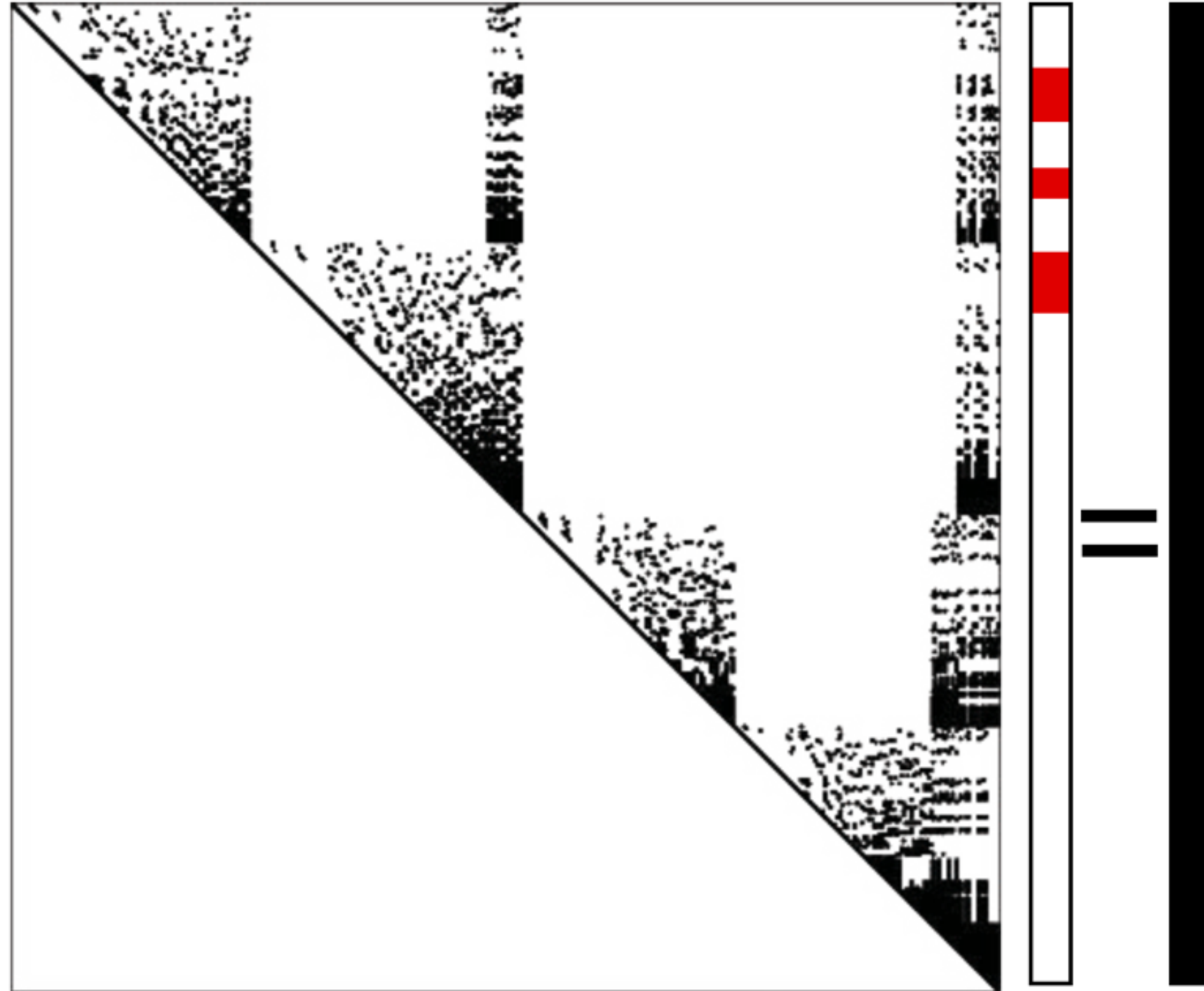
Sparse solution vector

$$\mathbf{LDy} = \mathbf{b} \quad (1) \text{Forward solve}$$

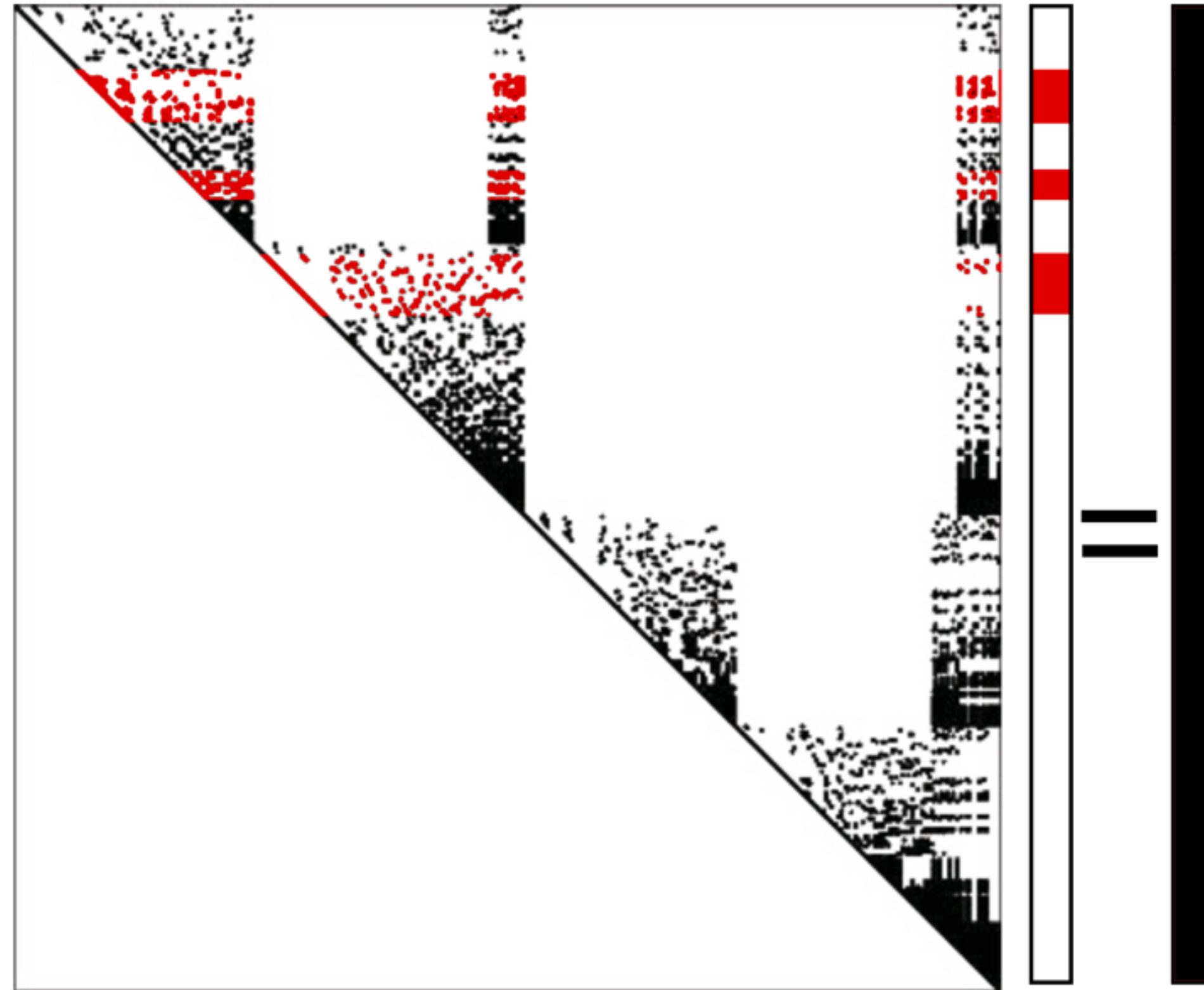
$$\mathbf{L}^T \mathbf{x} = \mathbf{y} \quad (2) \text{Back solve}$$

- \mathbf{x} will generally be dense.
- **Central insight:** If we are interested in only a subset of values in \mathbf{x} we do not have to compute all values during the back solve!

Sparse solution vector



Sparse solution vector



- How many values do we have to compute additionally?
- Can we identify them efficiently?

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates the sparse back-solve equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$. The matrix \mathbf{L}^T is a 7x7 sparse matrix with non-zero entries (black squares) at the following positions: (1,1), (1,2), (1,3), (2,2), (2,3), (3,3), (3,6), (3,7), (4,4), (4,5), (4,6), (4,7), (5,5), (6,6), (6,7), and (7,7). The vector \mathbf{x} is a 7x1 column vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The element x_2 is highlighted in red. The vector \mathbf{y} is a 7x1 column vector with all elements represented by black squares. The multiplication is indicated by a large \times symbol, and the equality is indicated by a large $=$ symbol.

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates a sparse back-solve operation. It features three main components: a 7x7 matrix \mathbf{L}^T , a 7x1 vector \mathbf{x} , and a 7x1 vector \mathbf{y} .

The matrix \mathbf{L}^T is shown as a 7x7 grid. The diagonal elements are black. The upper triangular part is sparse, with non-zero elements (black) at positions (1,2), (1,3), (2,4), (2,5), (3,6), and (3,7). The lower triangular part is also sparse, with non-zero elements (black) at positions (4,1), (4,2), (5,1), (5,2), (6,1), (6,2), and (7,1). The element at (2,2) is dark red, and the elements at (2,1), (2,3), (2,4), (2,5), (2,6), and (2,7) are light red.

The vector \mathbf{x} is shown as a 7x1 column of boxes labeled x_1 through x_7 . The box for x_2 is red, while the others are white.

The vector \mathbf{y} is shown as a 7x1 column of black boxes.

The multiplication symbol \times is placed between the matrix and the vector \mathbf{x} , and the equals sign $=$ is placed between the vector \mathbf{x} and the vector \mathbf{y} .

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates a sparse back-solve operation. It features three main components: a 7x7 matrix \mathbf{L}^T , a 7x1 vector \mathbf{x} , and a 7x1 vector \mathbf{y} .

The matrix \mathbf{L}^T is shown as a 7x7 grid. The diagonal elements are black. The upper triangular part is sparse, with non-zero elements (black) at positions (1,2), (1,3), (1,4), (1,5), (1,6), and (1,7). The lower triangular part is also sparse, with non-zero elements (black) at positions (2,3), (3,4), (3,5), (4,6), (5,6), and (6,7). The element at (2,3) is labeled $\mathbf{L}_{2,3}^T$.

The vector \mathbf{x} is shown as a 7x1 column of boxes labeled x_1 through x_7 . The box for x_2 is highlighted in red.

The vector \mathbf{y} is shown as a 7x1 column of black boxes.

The equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$ is represented by the matrix \mathbf{L}^T multiplied by the vector \mathbf{x} (indicated by a \times symbol) equals the vector \mathbf{y} (indicated by an $=$ symbol).

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates the sparse back-solve equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$. The matrix \mathbf{L}^T is a 7x7 matrix with a block structure. The vector \mathbf{x} is a 7x1 column vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The vector \mathbf{y} is a 7x1 column vector.

The matrix \mathbf{L}^T is represented by a 7x7 grid. The cells are colored as follows:

- Row 1: (1,1), (1,2), (1,3) are black; (1,4), (1,5), (1,6), (1,7) are white.
- Row 2: (2,2), (2,3) are black; (2,1), (2,4), (2,5), (2,6), (2,7) are white.
- Row 3: (3,1), (3,2) are light red; (3,3) is dark red; (3,4), (3,5) are light red; (3,6), (3,7) are dark red.
- Row 4: (4,4), (4,5), (4,6), (4,7) are black; (4,1), (4,2), (4,3) are white.
- Row 5: (5,5) is black; (5,4), (5,6), (5,7) are white.
- Row 6: (6,6), (6,7) are black; (6,1), (6,2), (6,3), (6,4), (6,5) are white.
- Row 7: (7,7) is black; (7,1), (7,2), (7,3), (7,4), (7,5), (7,6) are white.

The vector \mathbf{x} is represented by a 7x1 column vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The element x_2 is highlighted in red.

The vector \mathbf{y} is represented by a 7x1 column vector with all elements black.

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates the sparse back-solve equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$. The matrix \mathbf{L}^T is a 7x7 sparse matrix with a block structure. The vector \mathbf{x} is a 7x1 column vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The vector \mathbf{y} is a 7x1 column vector. The matrix \mathbf{L}^T has a 3x3 block $\mathbf{L}_{3,6}^T \mathbf{L}_{3,7}^T$ highlighted in dark red. The vector \mathbf{x} has elements x_2, x_6, x_7 highlighted in pink. The vector \mathbf{y} is entirely black.

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

The diagram illustrates the sparse back-solve process for the equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$.

Matrix \mathbf{L}^T : A 7x7 matrix with a sparse structure. The diagonal elements are black. The off-diagonal elements are colored red, indicating non-zero values. The structure is as follows:

Row \ Column	1	2	3	4	5	6	7
1	Black	Black	Black	White	White	White	White
2	White	Black	Black	White	White	White	White
3	White	White	Black	White	White	Black	Black
4	White	White	White	Black	Black	Black	Black
5	White	White	White	White	Black	White	Black
6	Red	Red	Red	Red	Red	Dark Red	Red
7	White	White	White	White	White	White	Black

The element $L_{6,7}^T$ is highlighted in dark red.

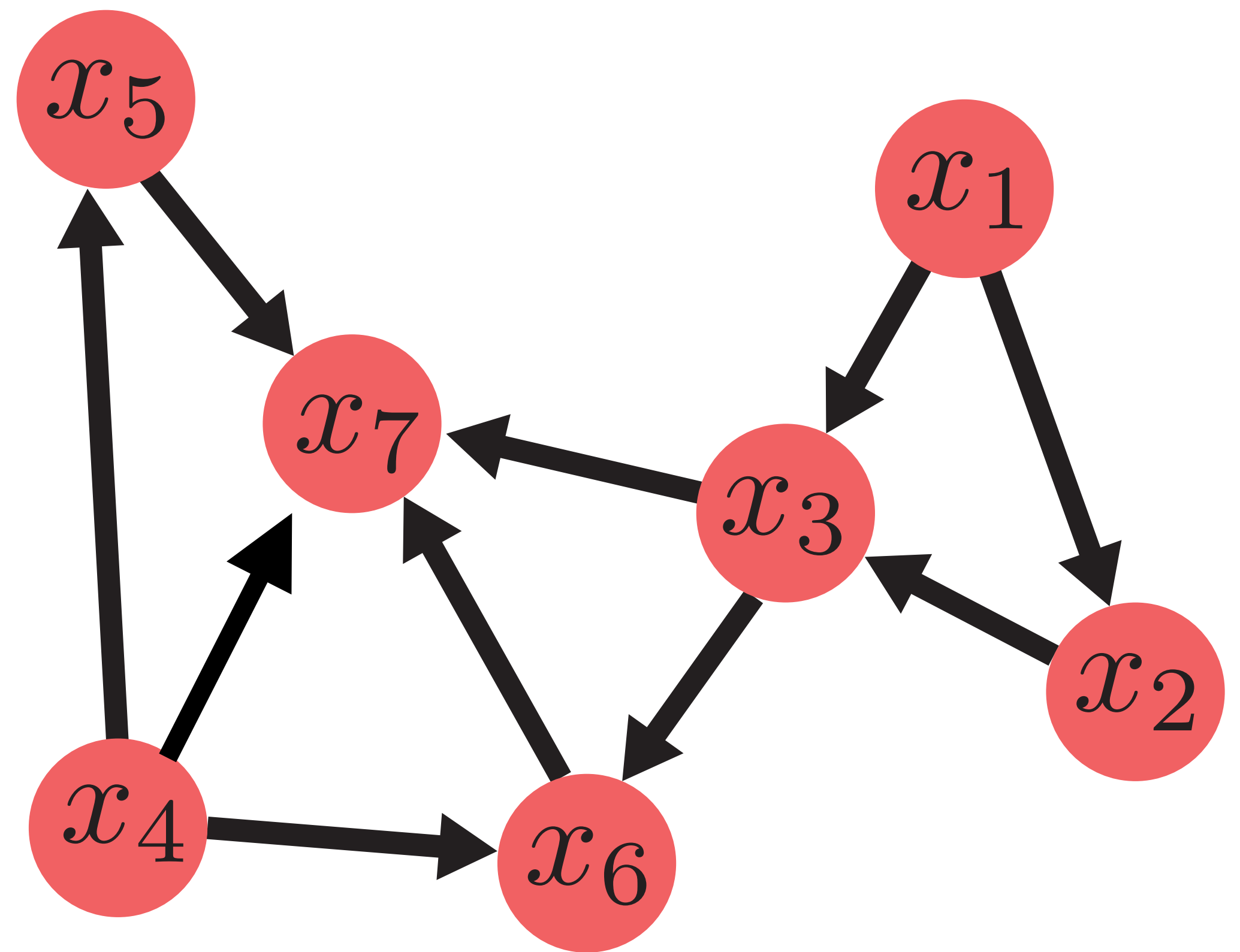
Vector \mathbf{x} : A 7x1 vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The elements x_2 and x_6 are highlighted in red.

Vector \mathbf{y} : A 7x1 vector with elements $y_1, y_2, y_3, y_4, y_5, y_6, y_7$. All elements are black.

Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

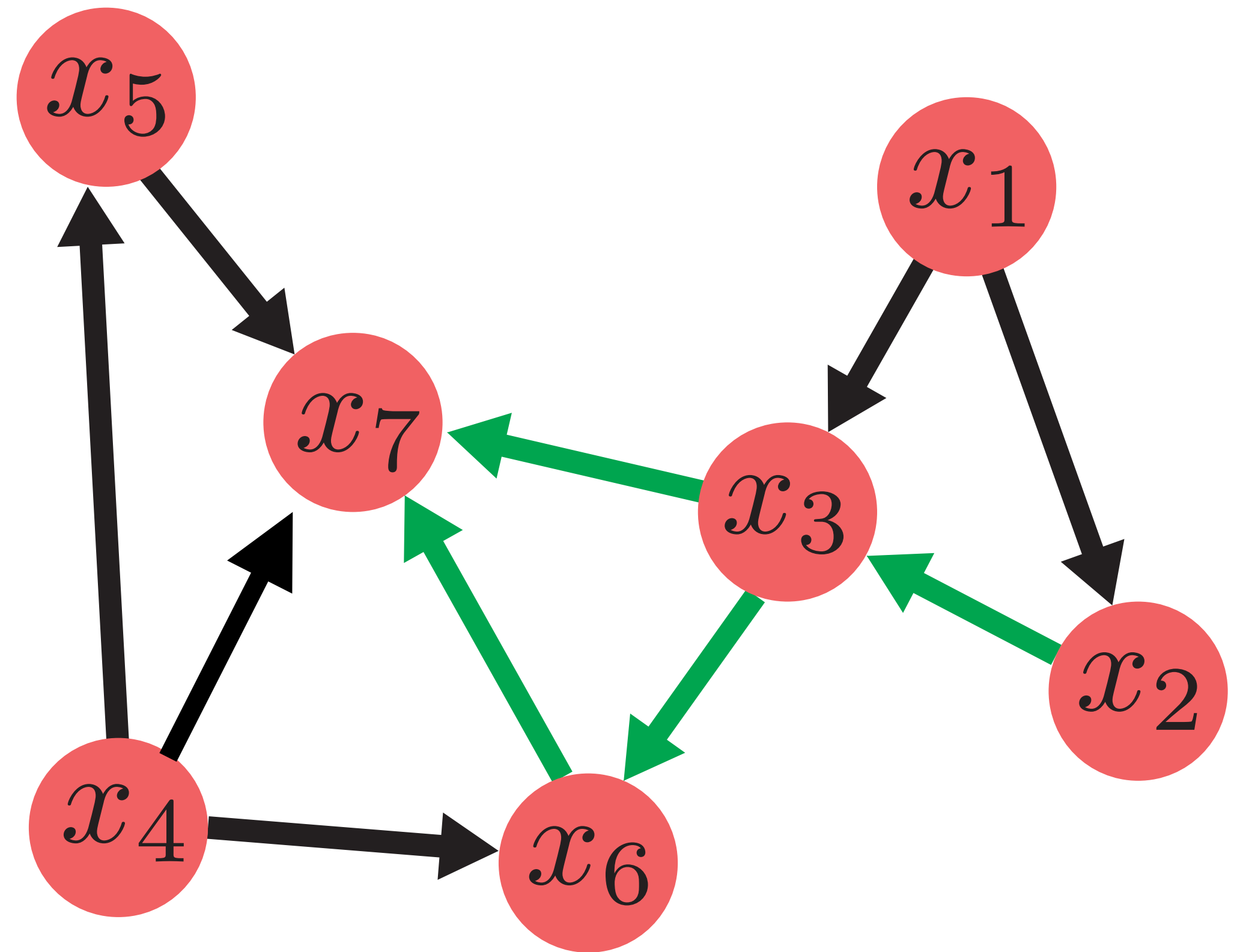
The diagram illustrates the sparse back-solve equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$. The matrix \mathbf{L}^T is a 7x7 sparse matrix with black squares indicating non-zero entries. The vector \mathbf{x} is a 7x1 vector with elements $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The vector \mathbf{y} is a 7x1 vector of all black squares. The elements x_2, x_3, x_6, x_7 are highlighted in pink, indicating they are the unknowns to be solved for.



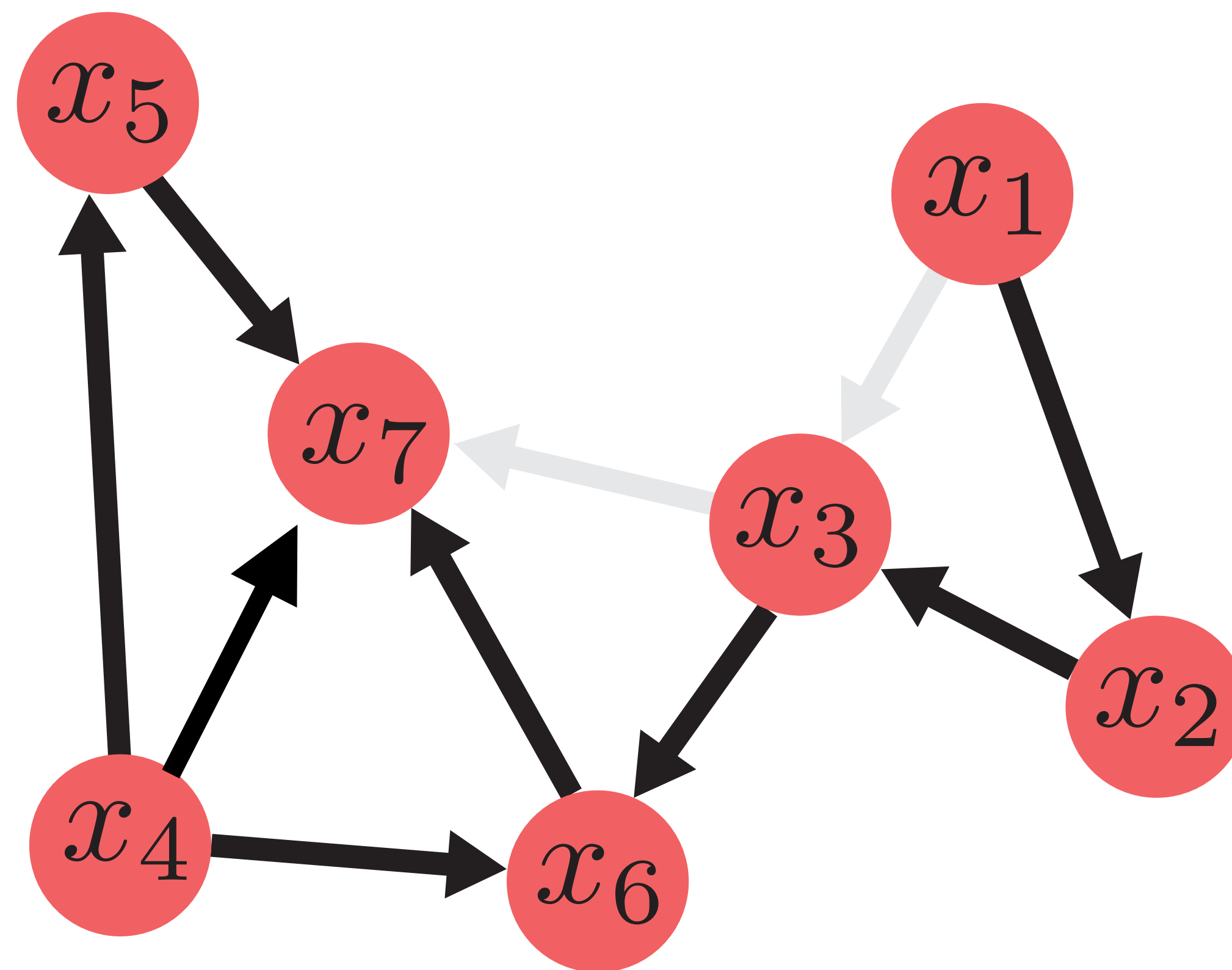
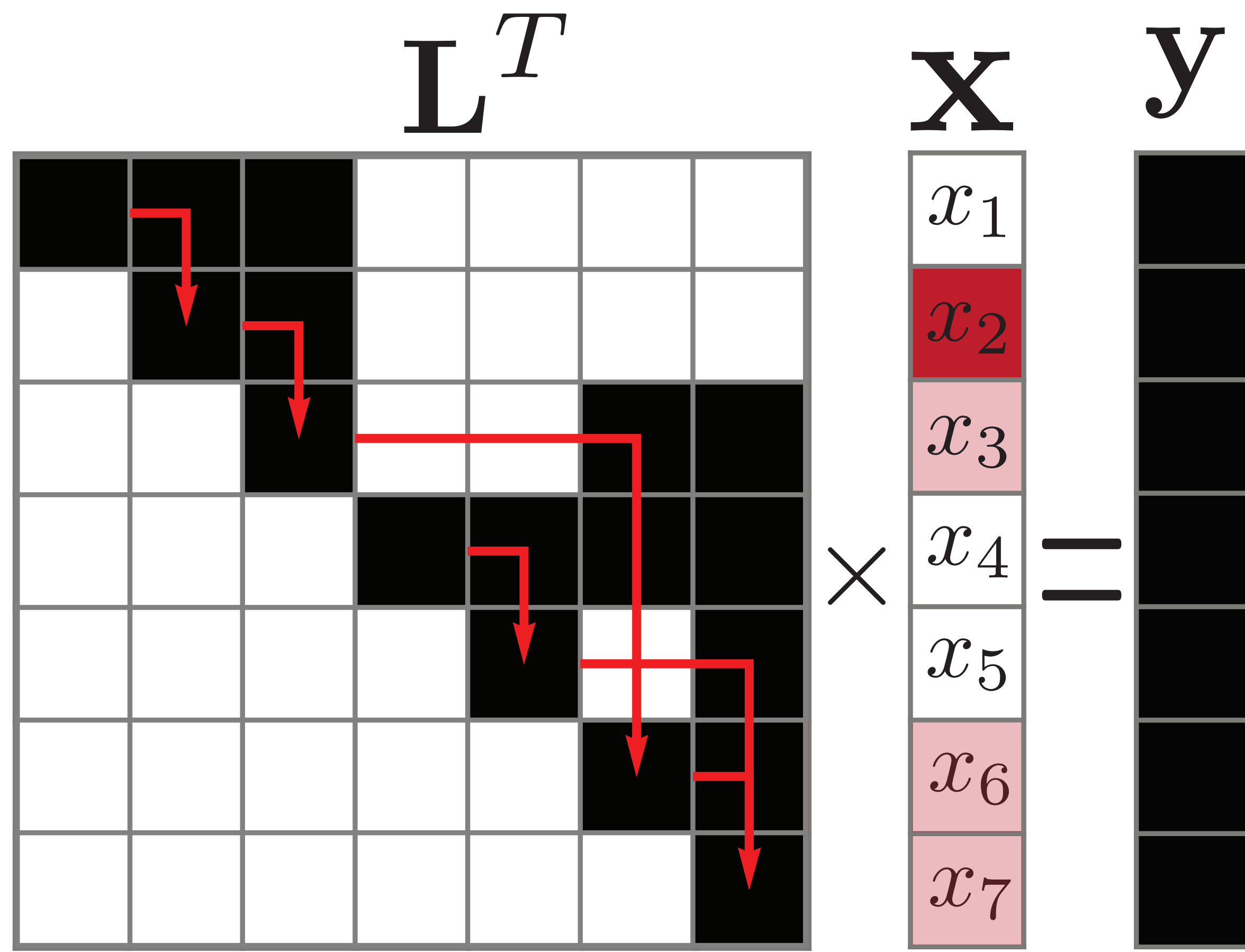
Sparse back-solve

$$\mathbf{L}^T \mathbf{x} = \mathbf{y}$$

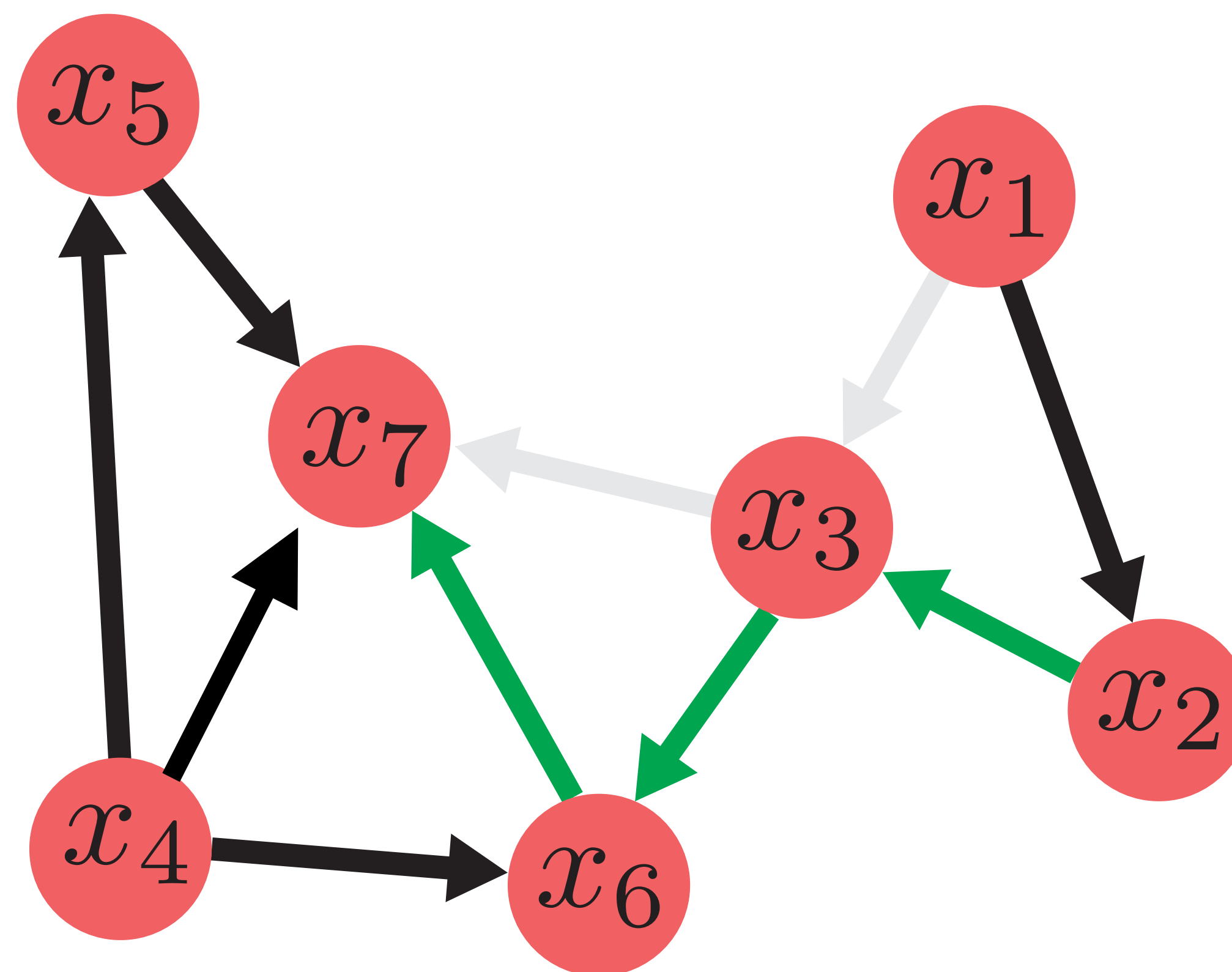
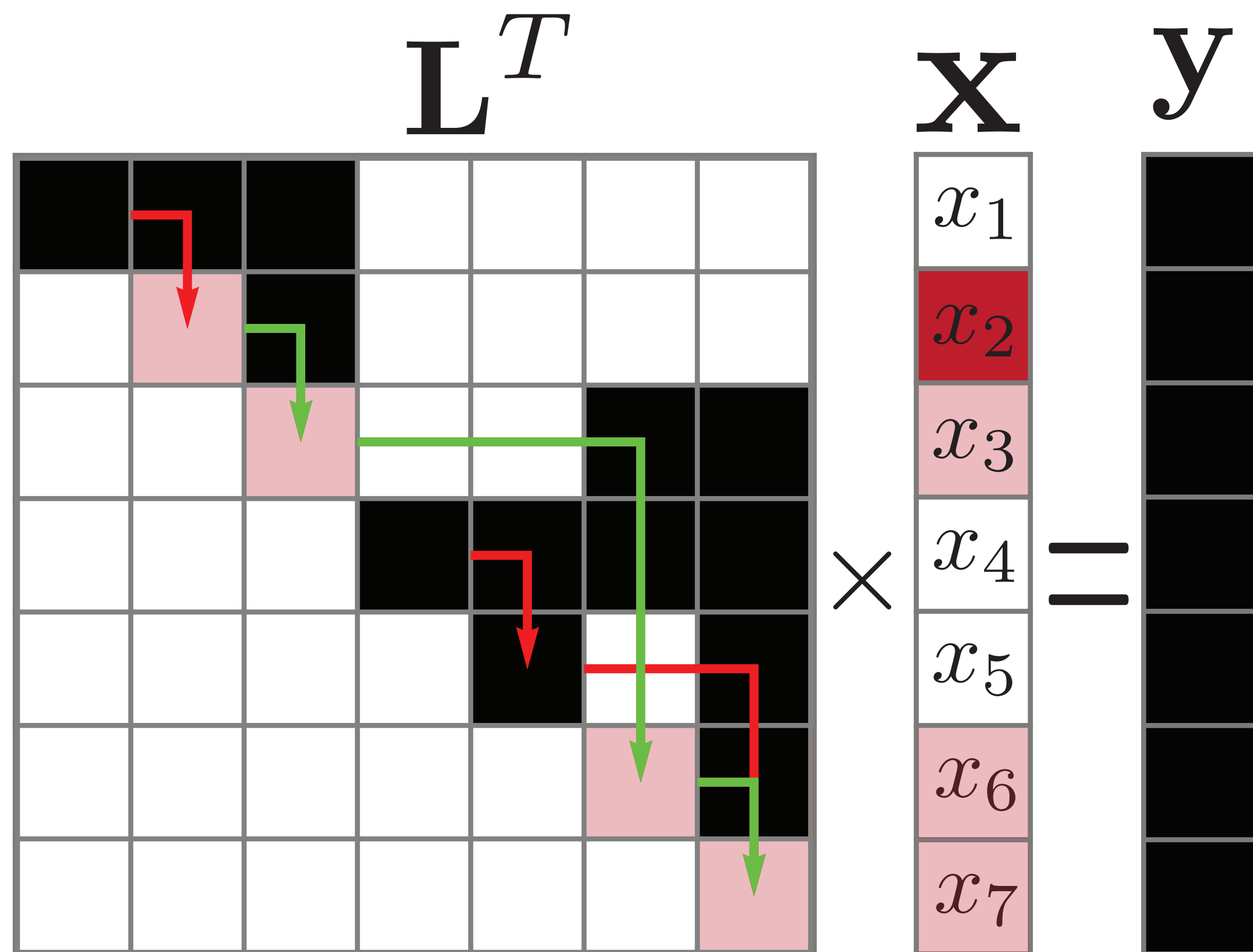
The diagram illustrates the sparse back-solve equation $\mathbf{L}^T \mathbf{x} = \mathbf{y}$. The matrix \mathbf{L}^T is a 7x7 sparse matrix with black squares indicating non-zero entries. The vector \mathbf{x} contains elements x_1 through x_7 , with $x_2, x_3, x_6,$ and x_7 highlighted in pink. The vector \mathbf{y} is a 7x1 vector of all black squares.



Sparse back-solve

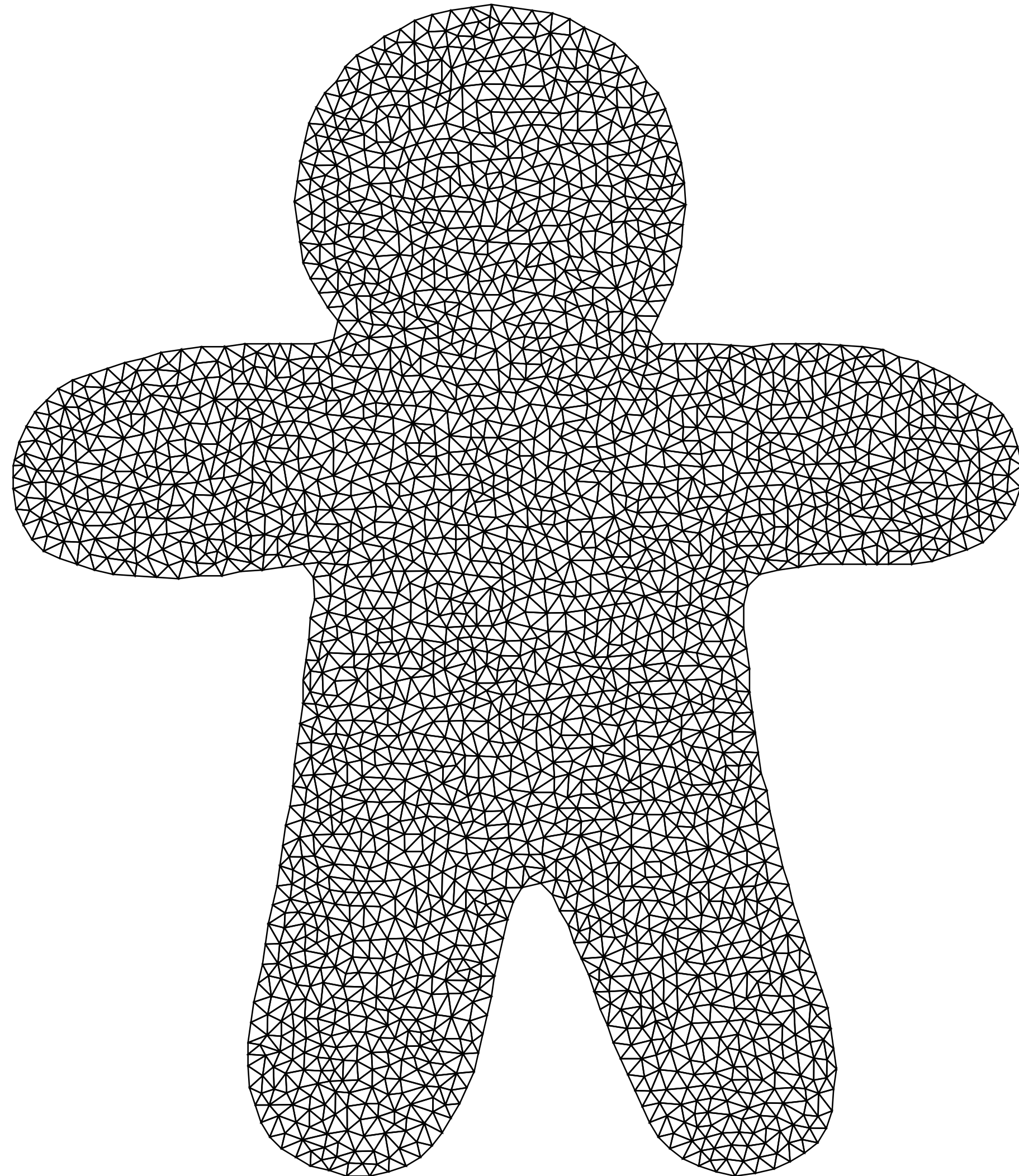


Sparse back-solve

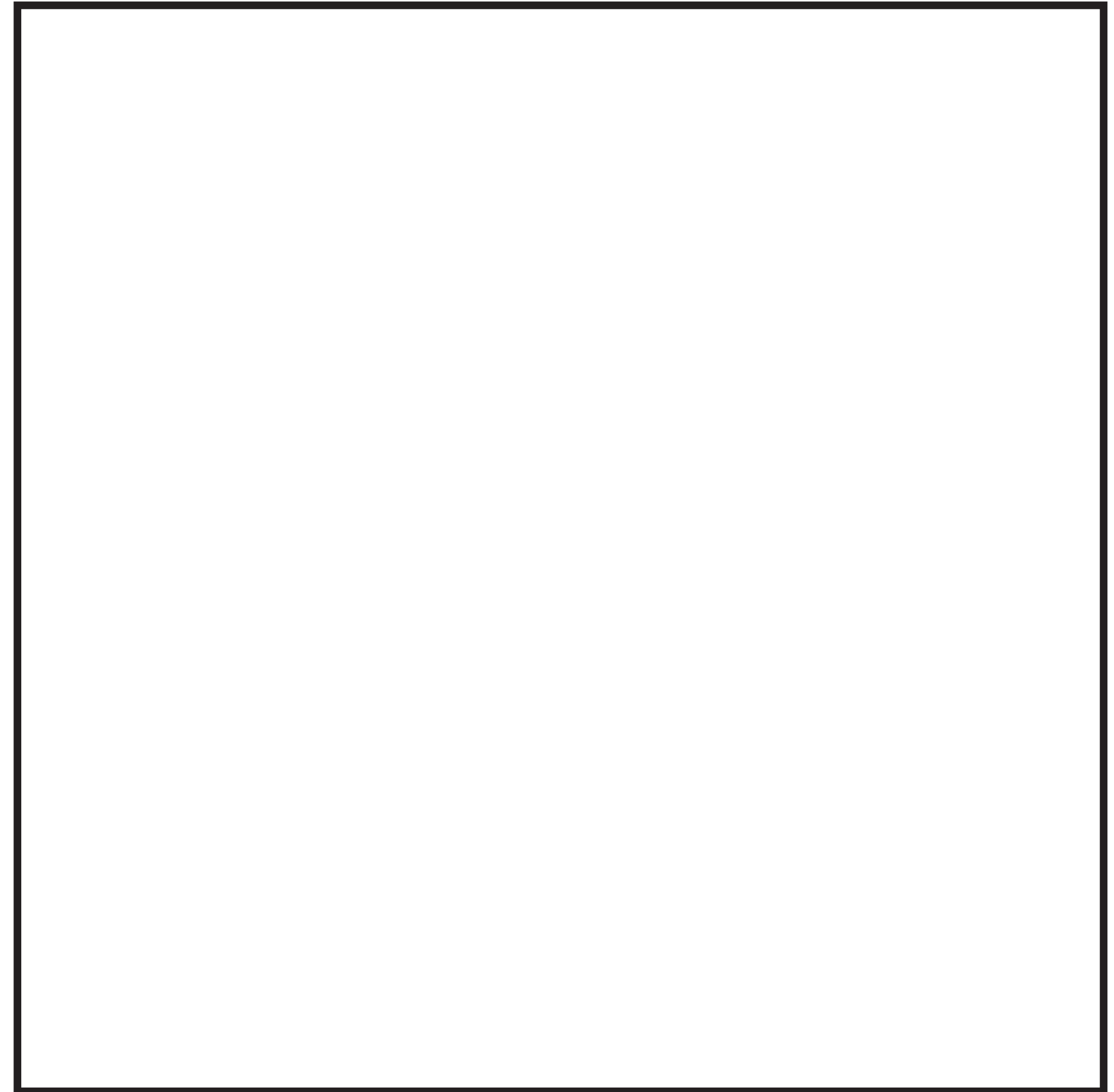


Nested dissection reordering

Input mesh

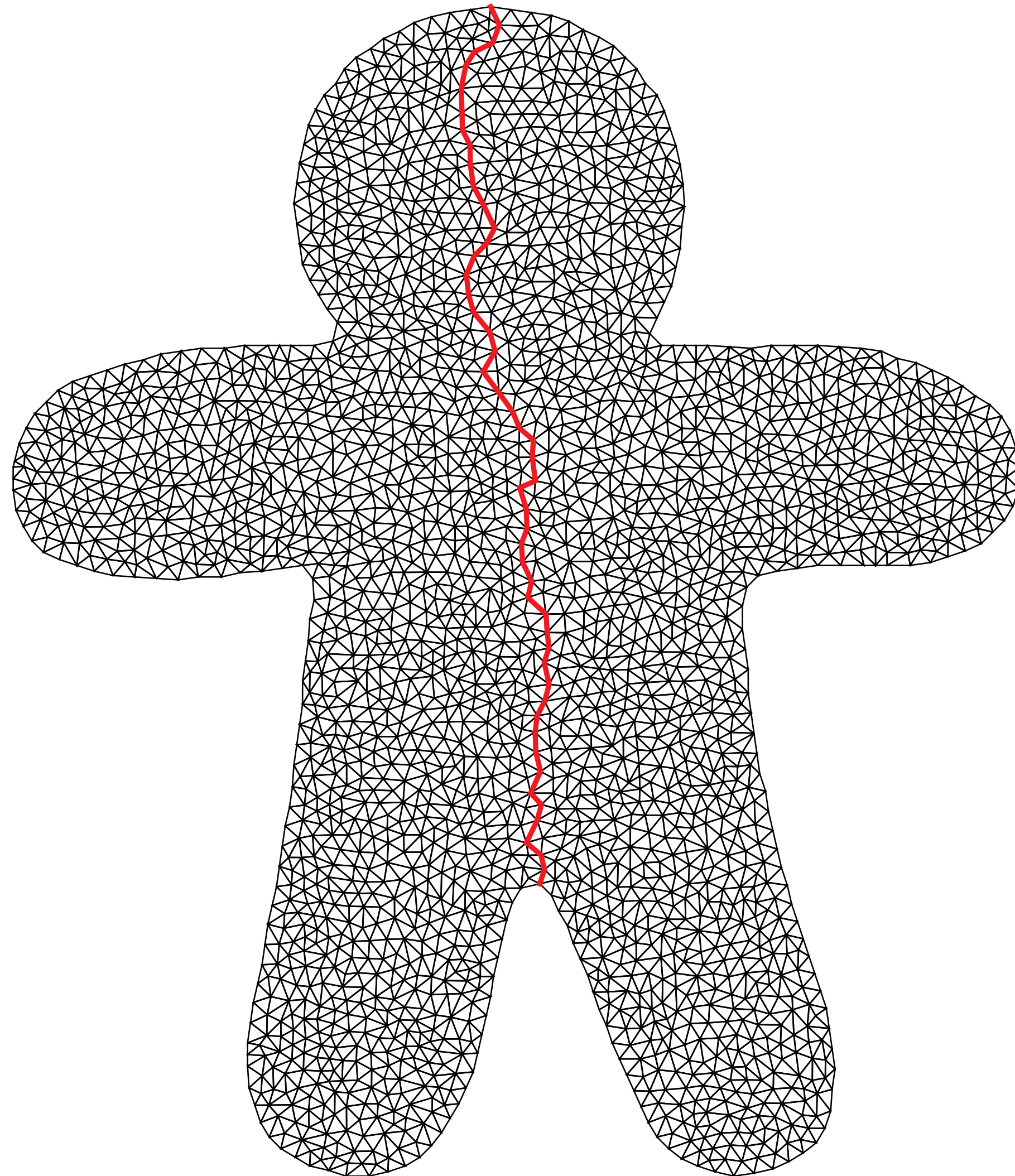


Ordered Laplacian

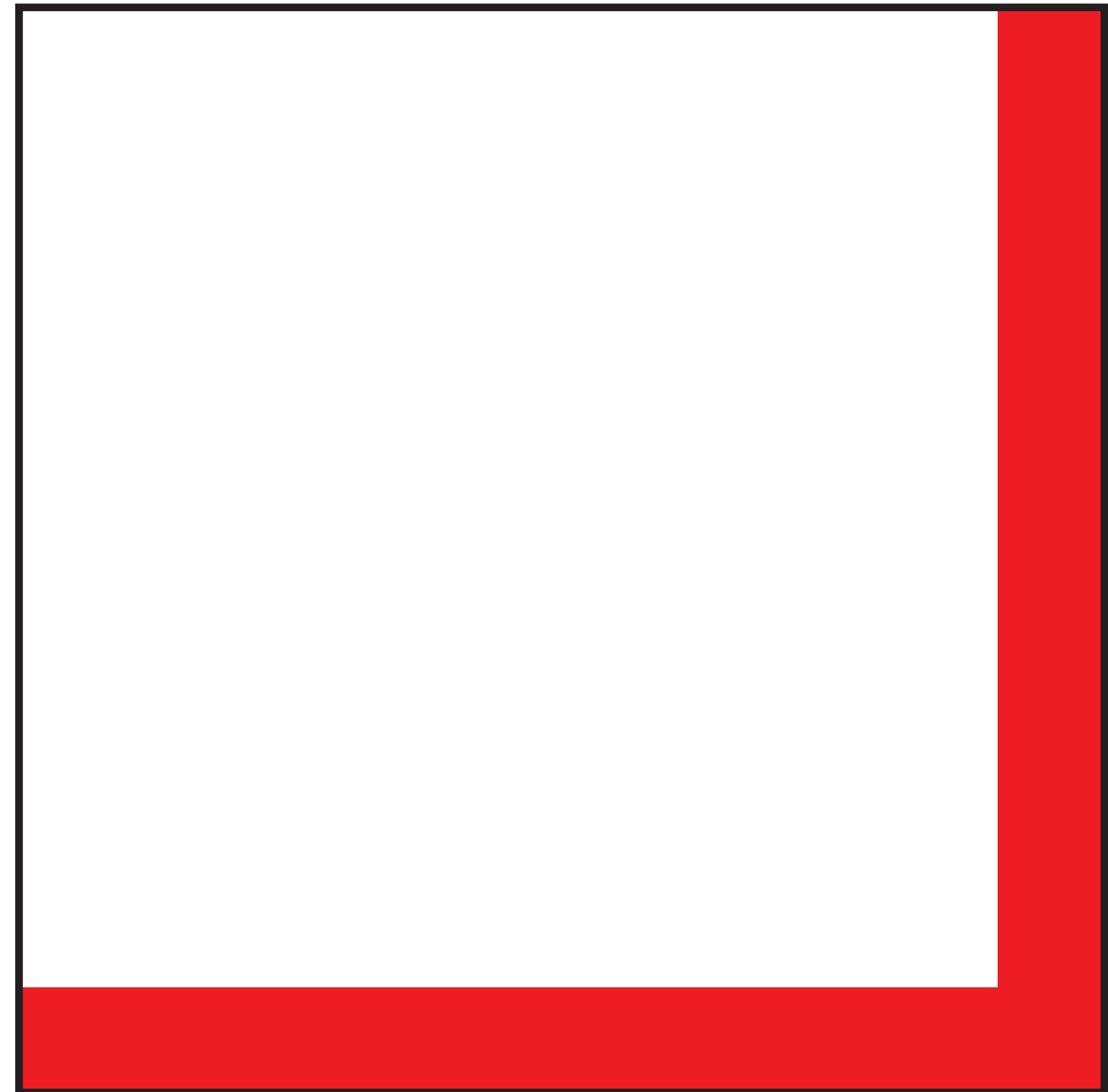


Nested dissection reordering

Input mesh

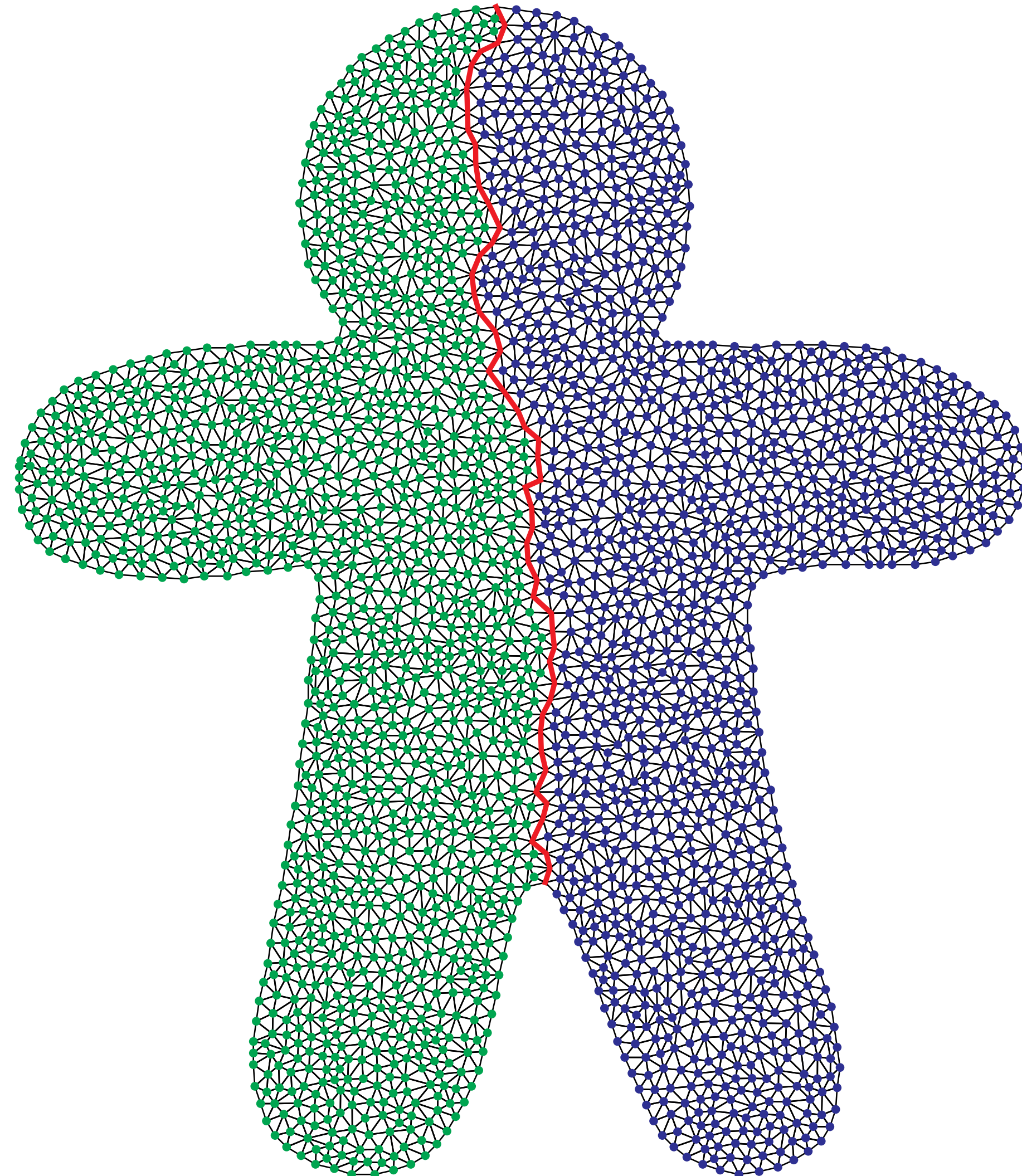


Ordered Laplacian

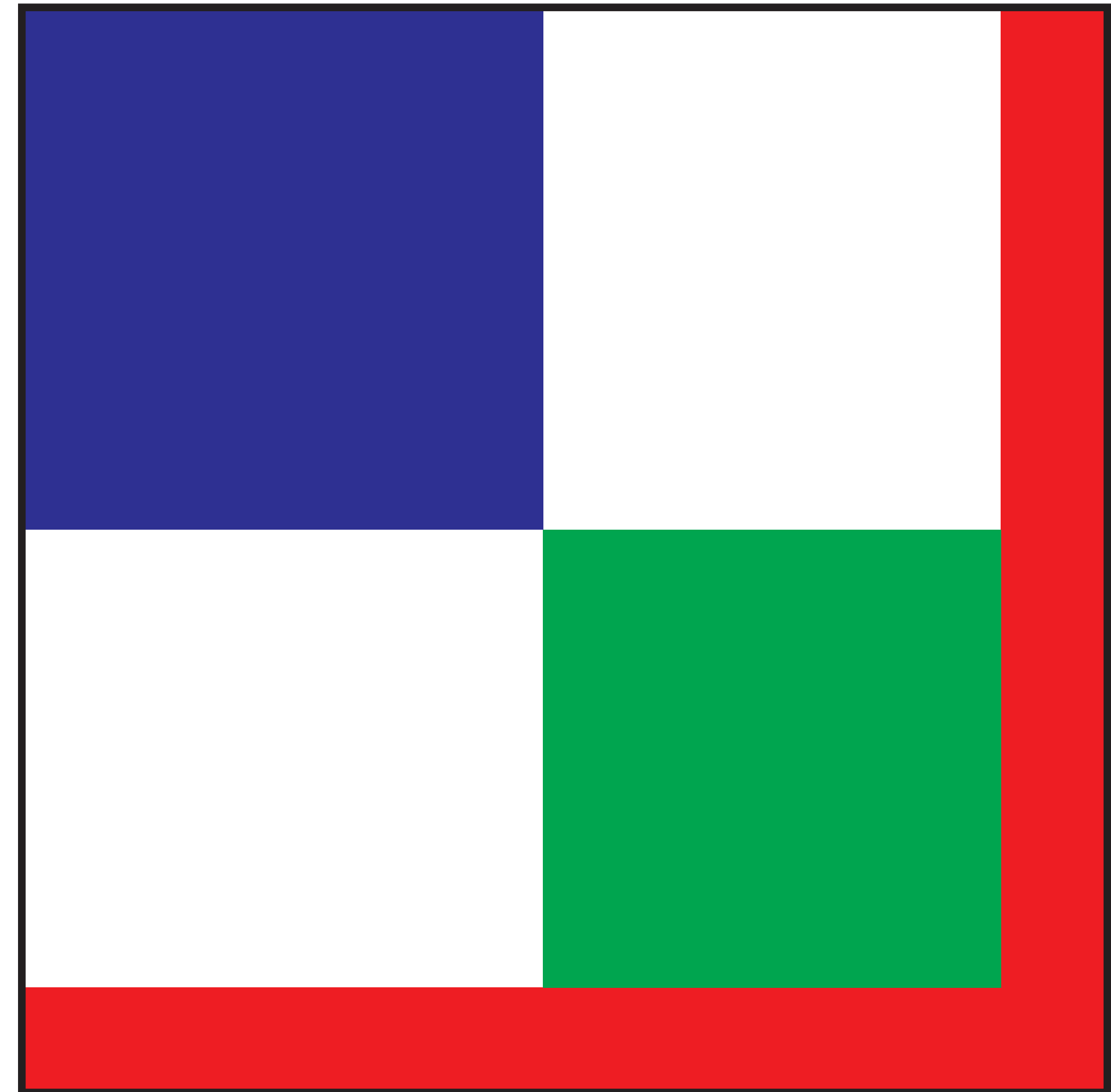


Nested dissection reordering

Input mesh

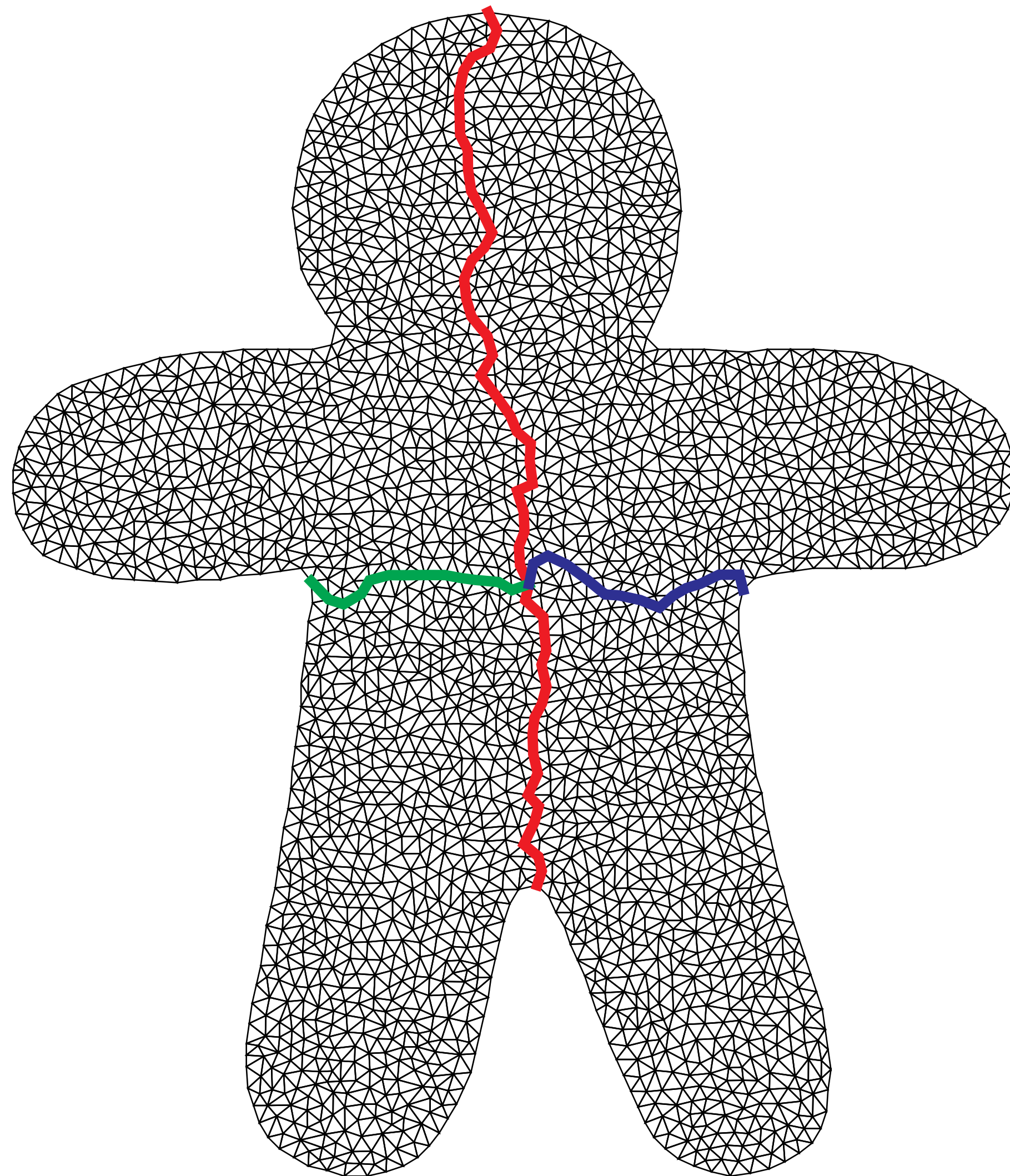


Ordered Laplacian

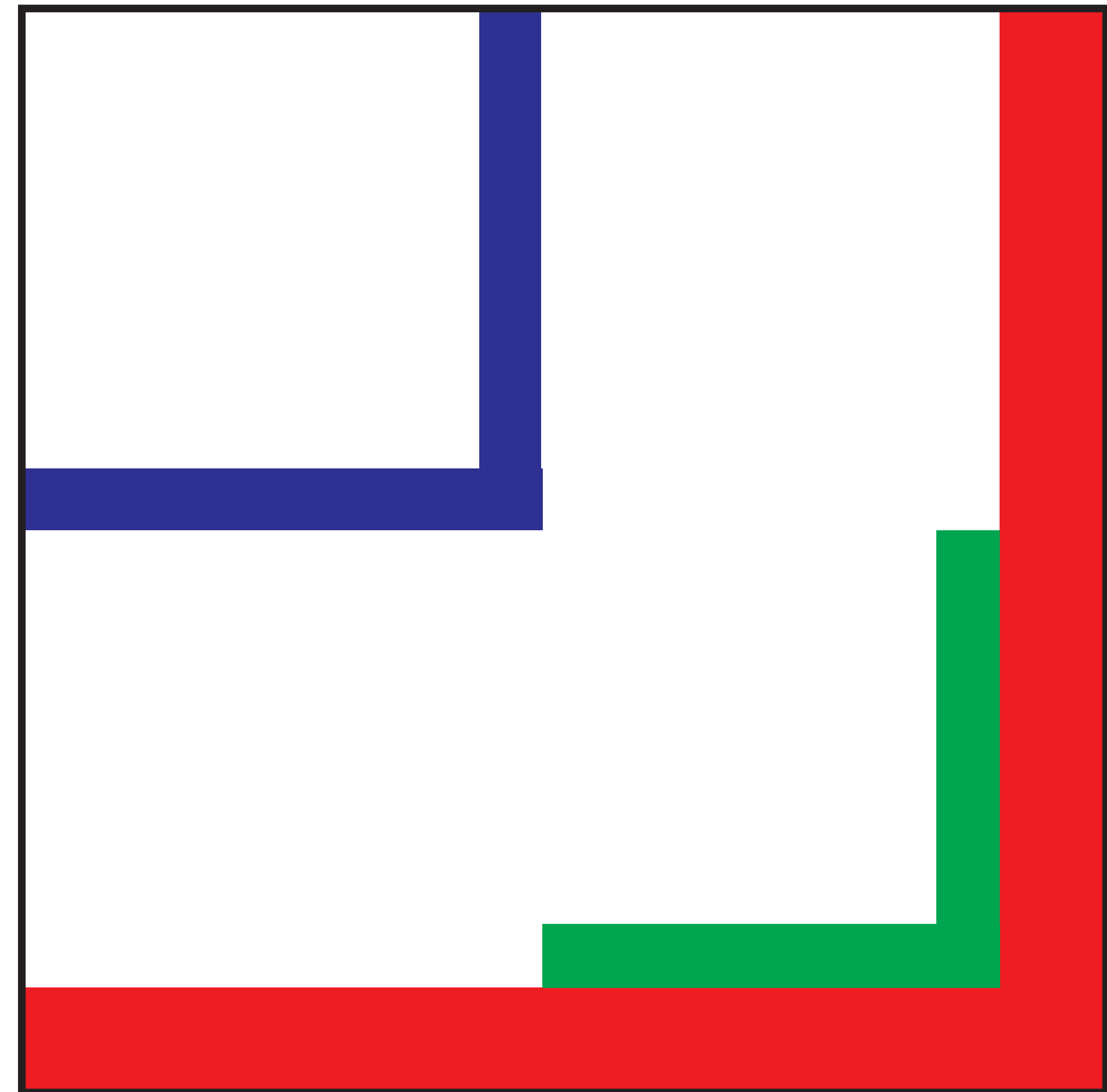


Nested dissection reordering

Input mesh

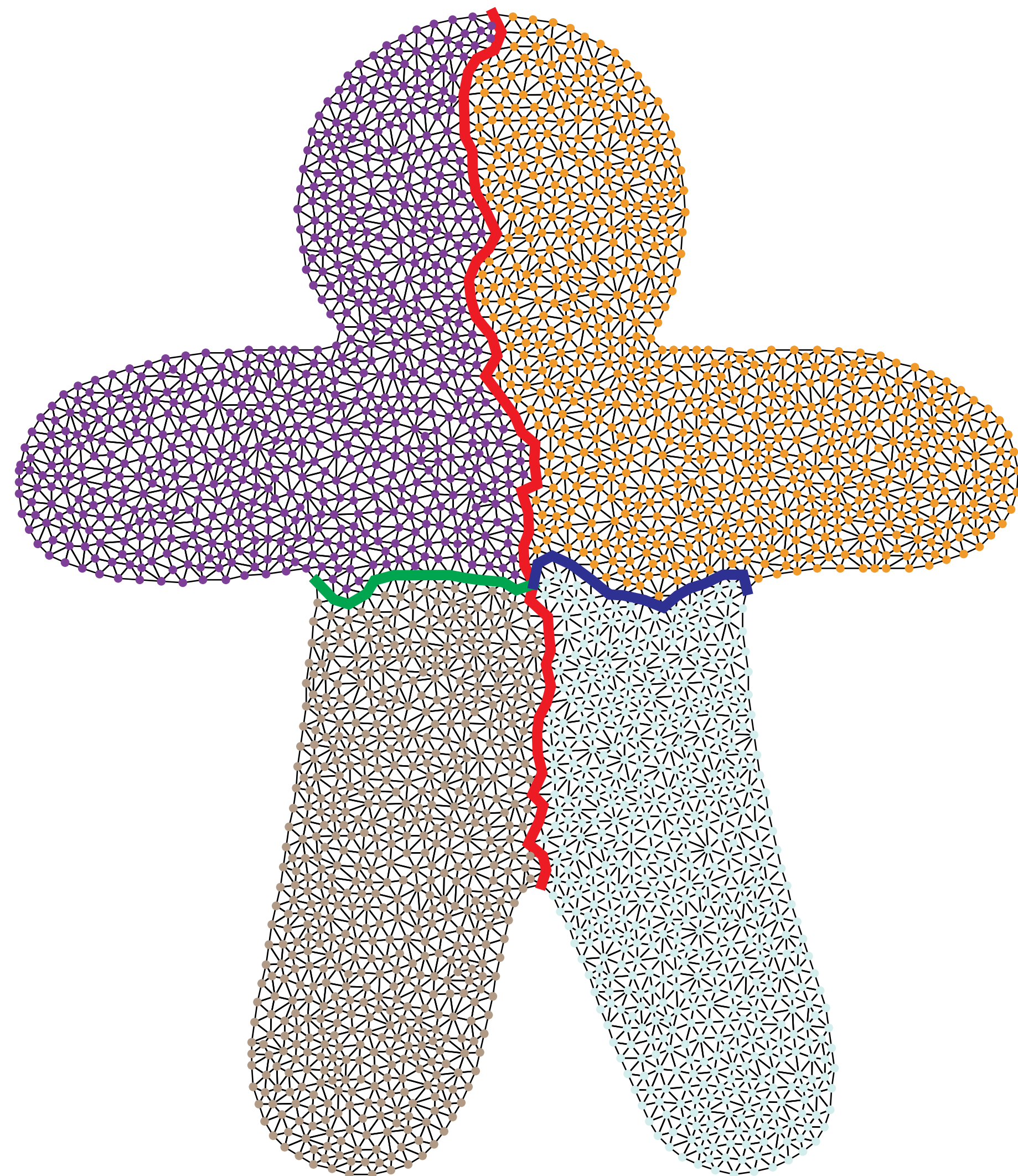


Ordered Laplacian

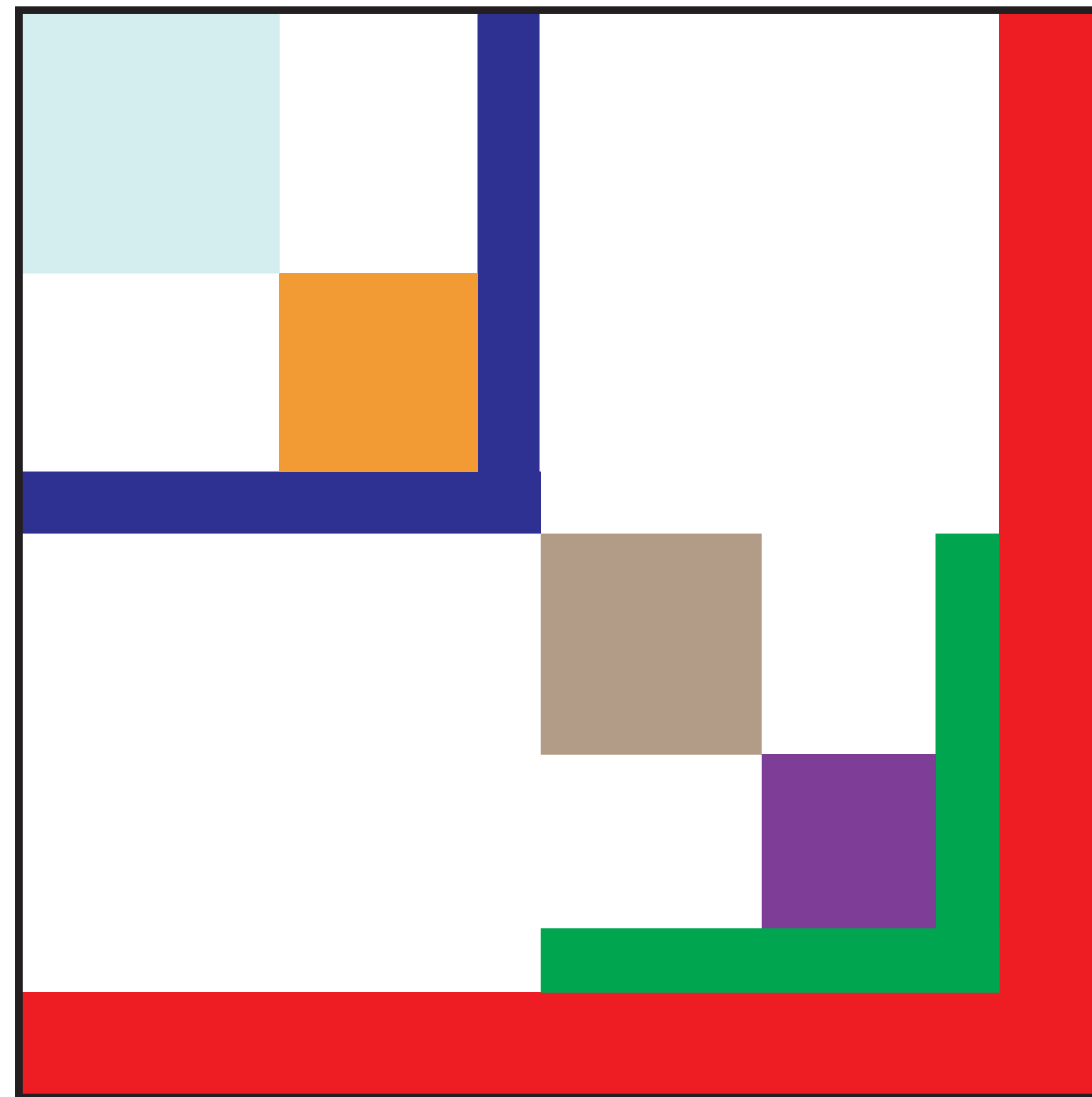


Nested dissection reordering

Input mesh

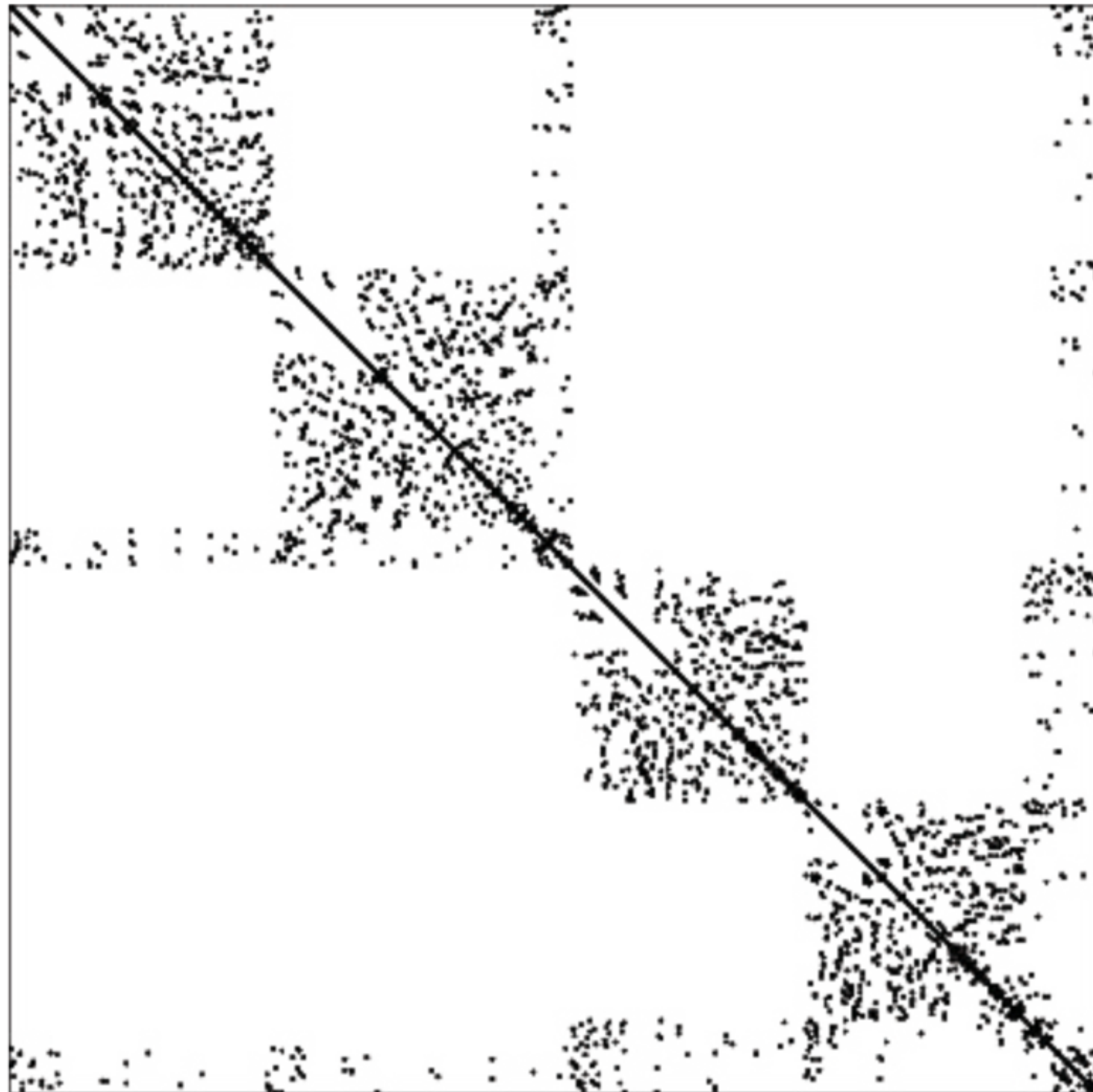


Ordered Laplacian

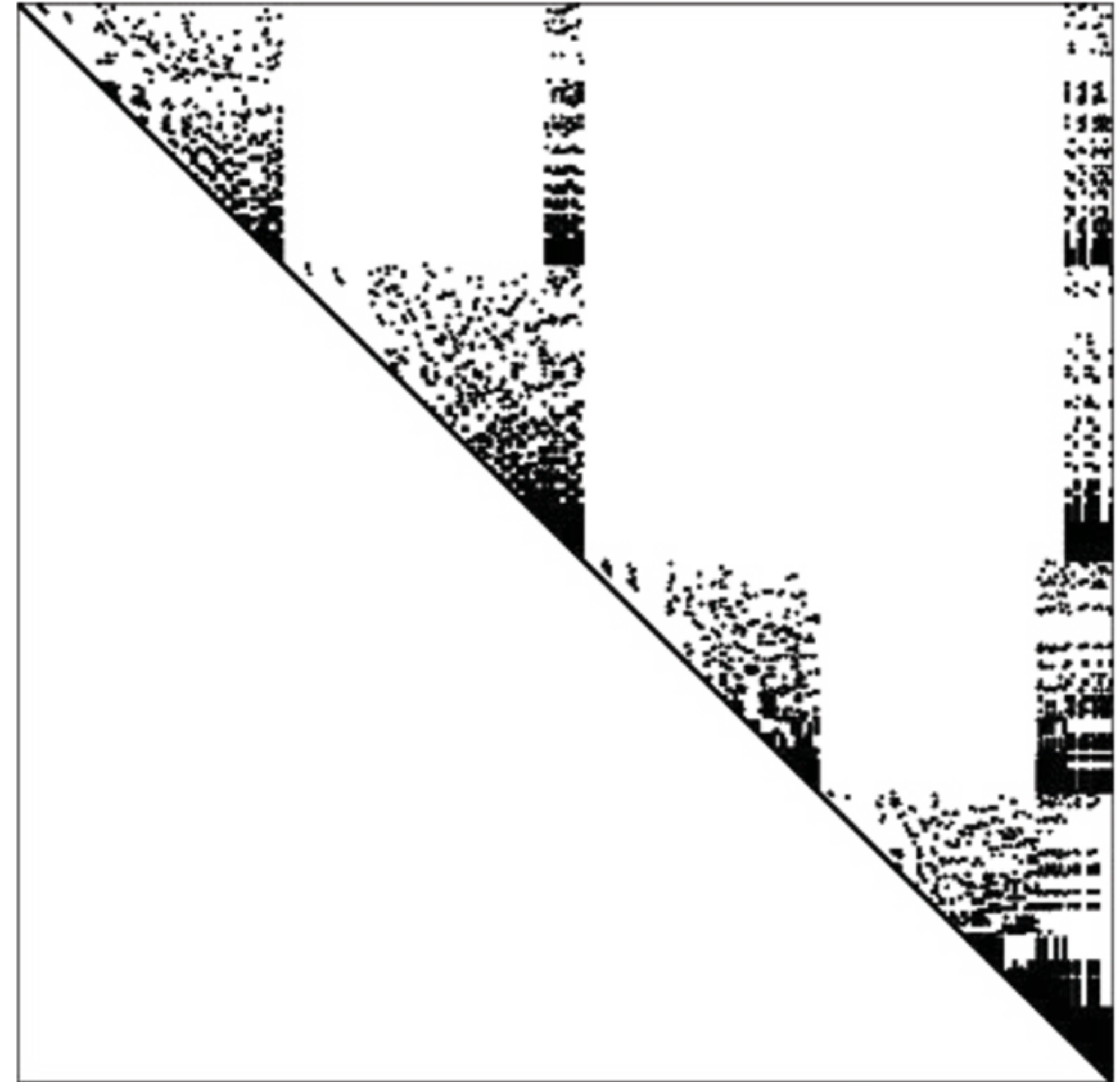


Nested dissection reordering

Input matrix

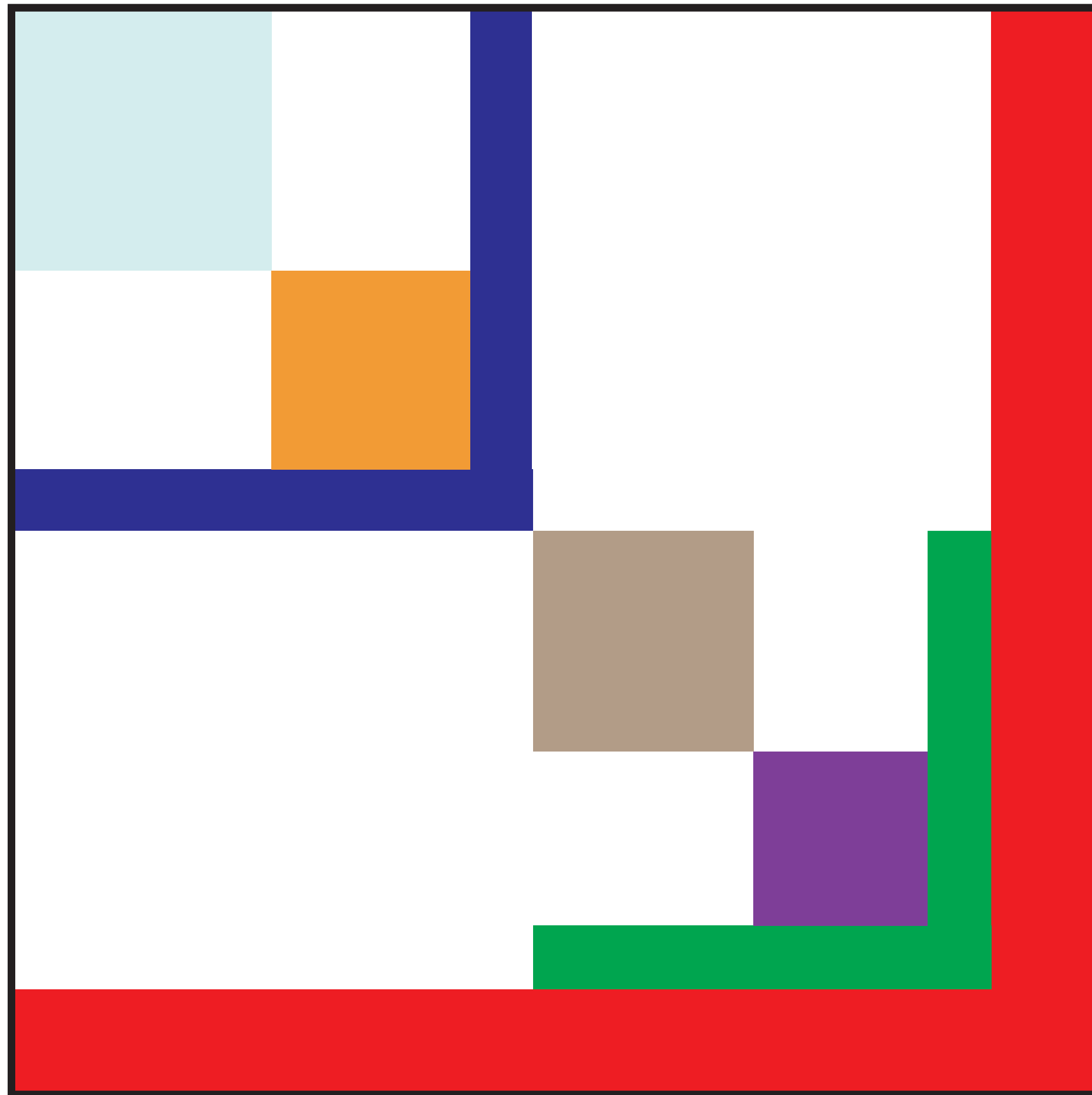


Cholesky factor \mathbf{L}^T

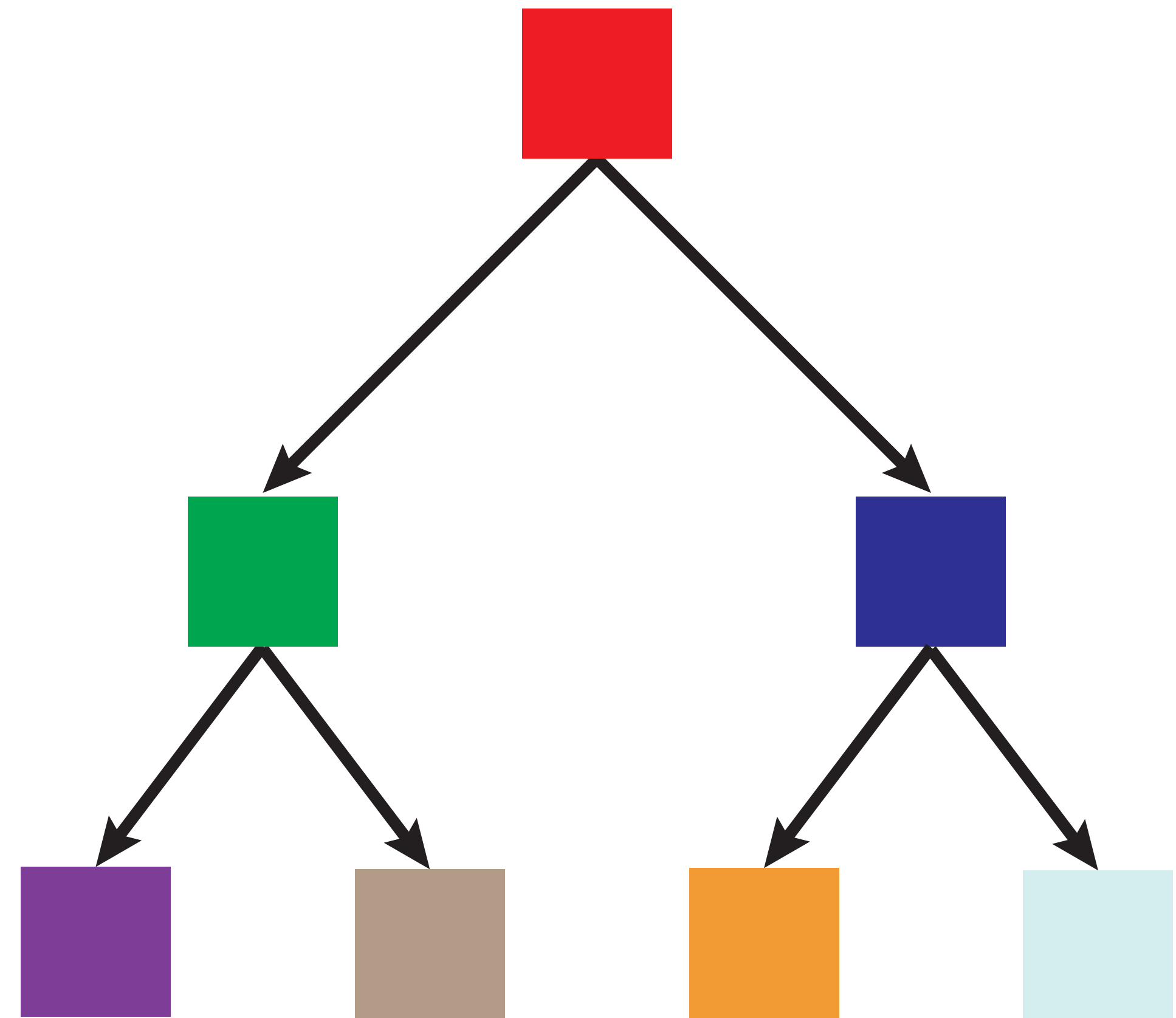


Nested dissection reordering

Cholesky factor

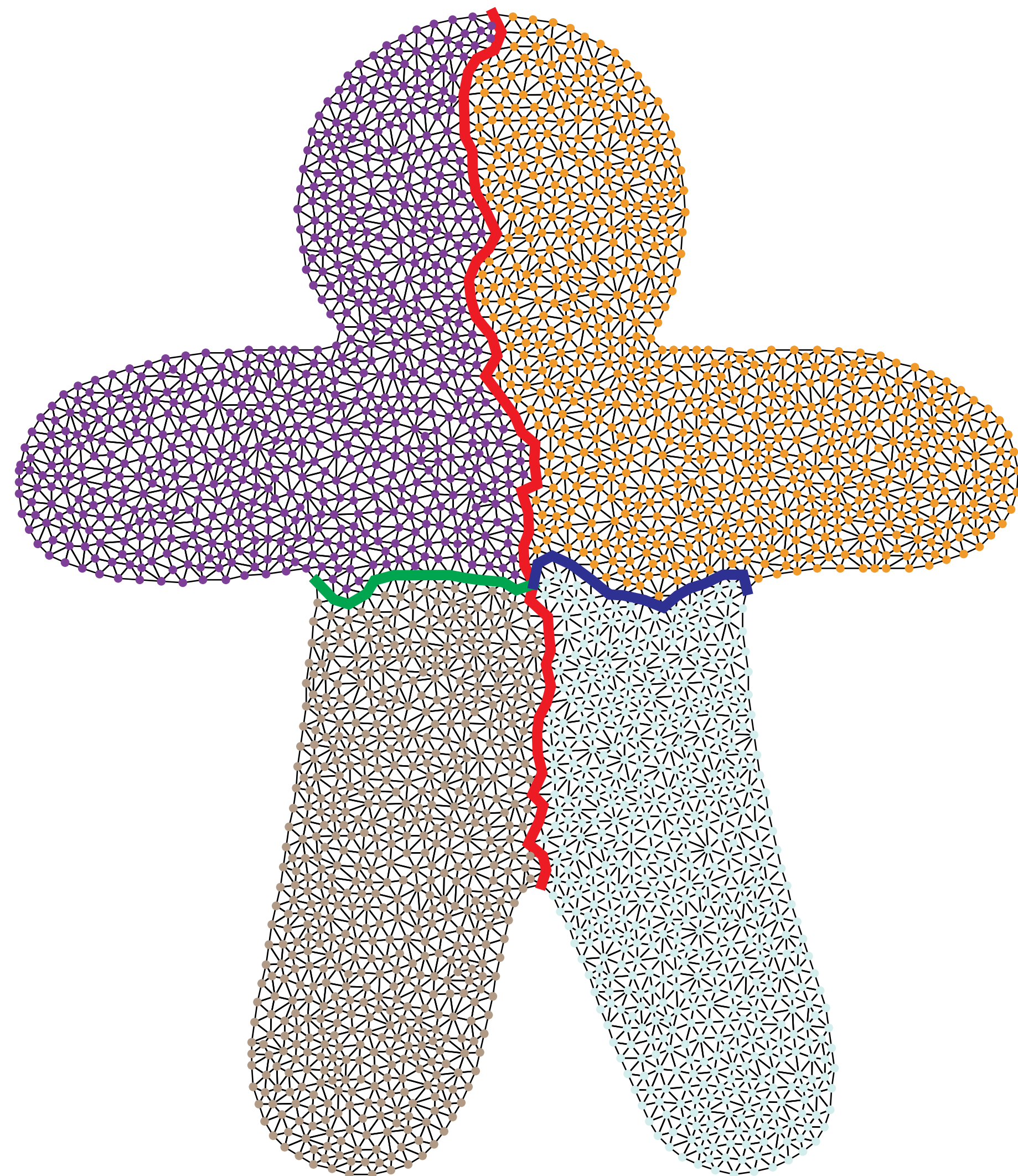


Elimination tree

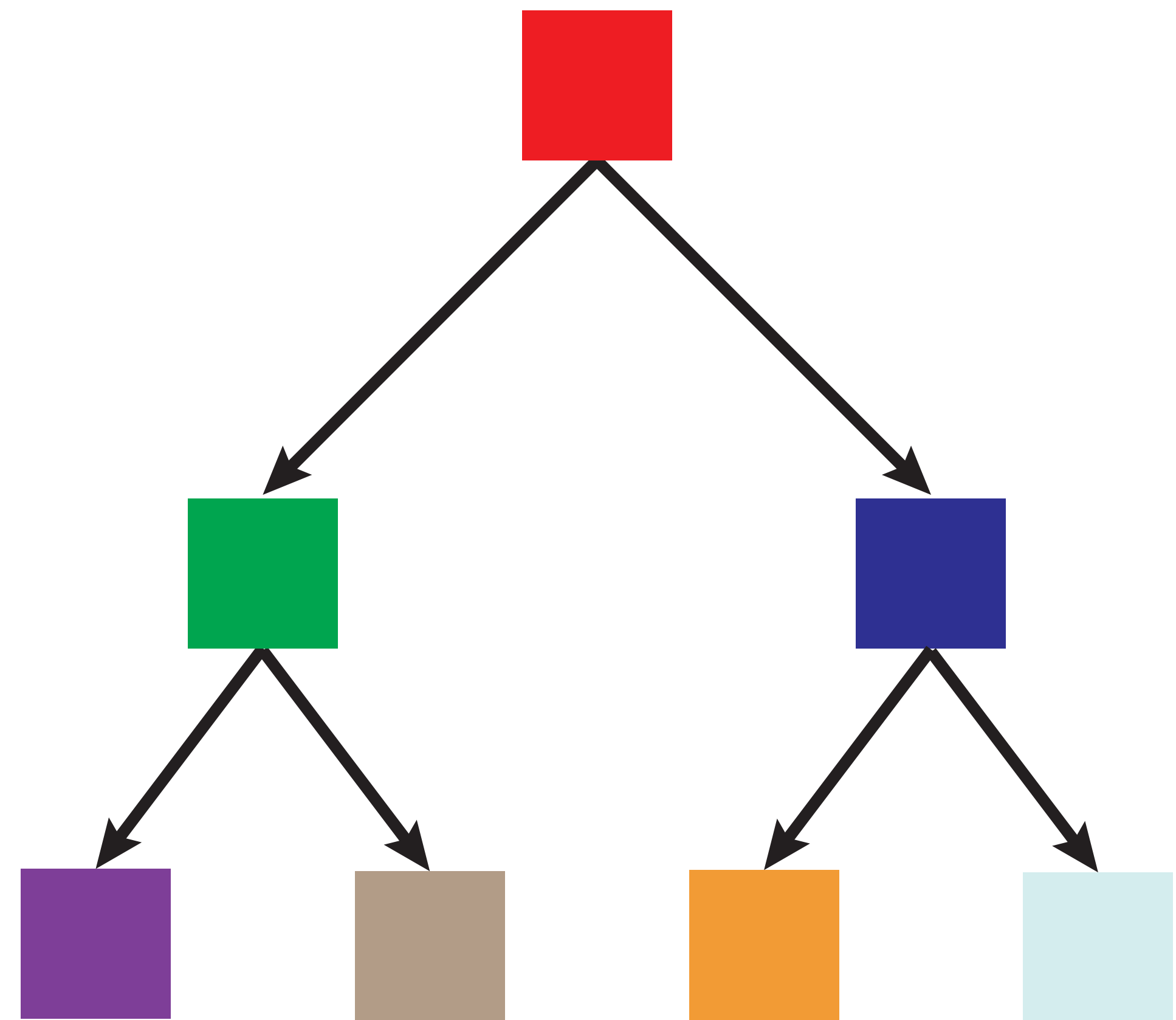


Nested dissection reordering

Input mesh

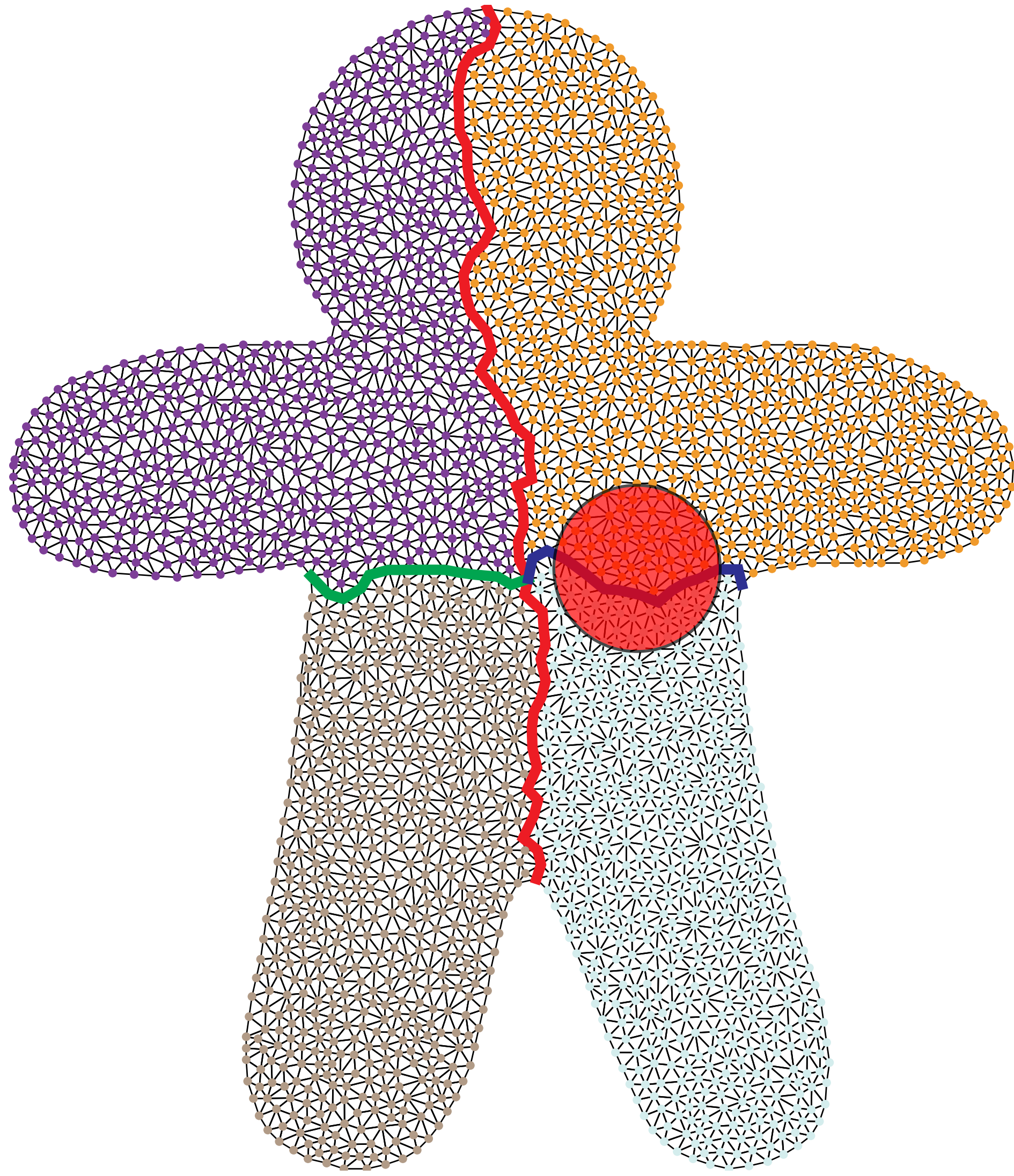


Elimination tree

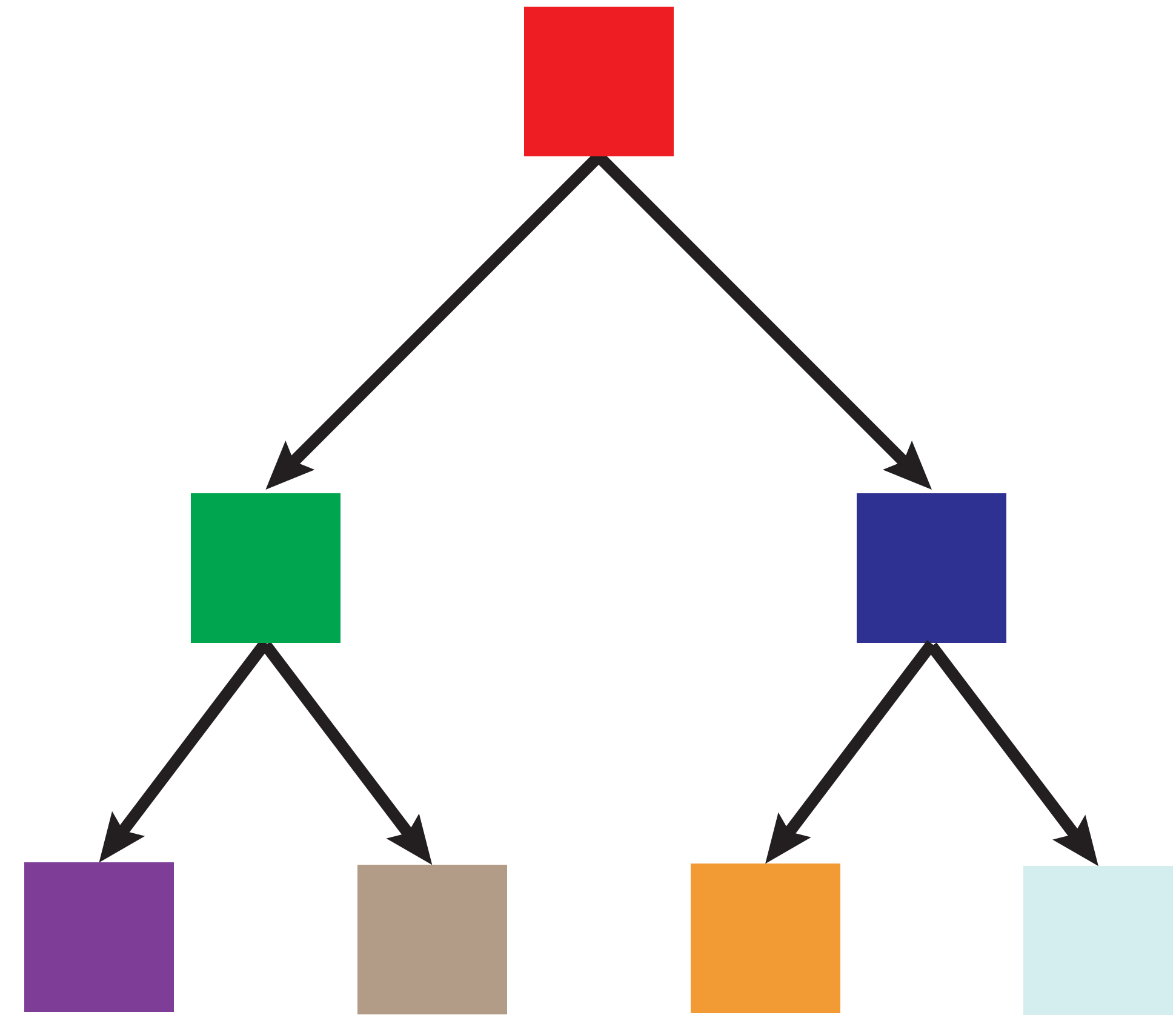


Nested dissection reordering

Input mesh

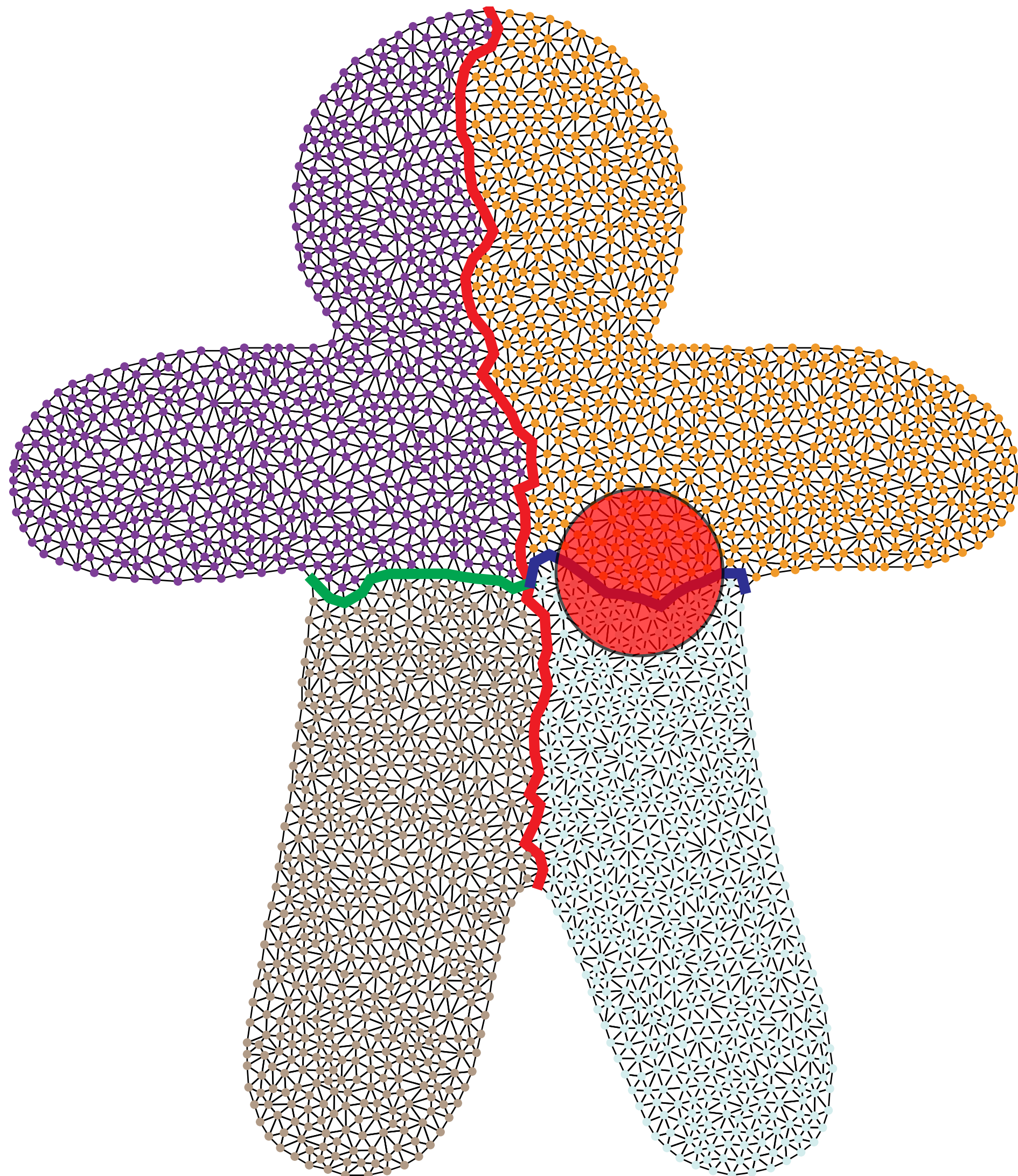


Elimination tree

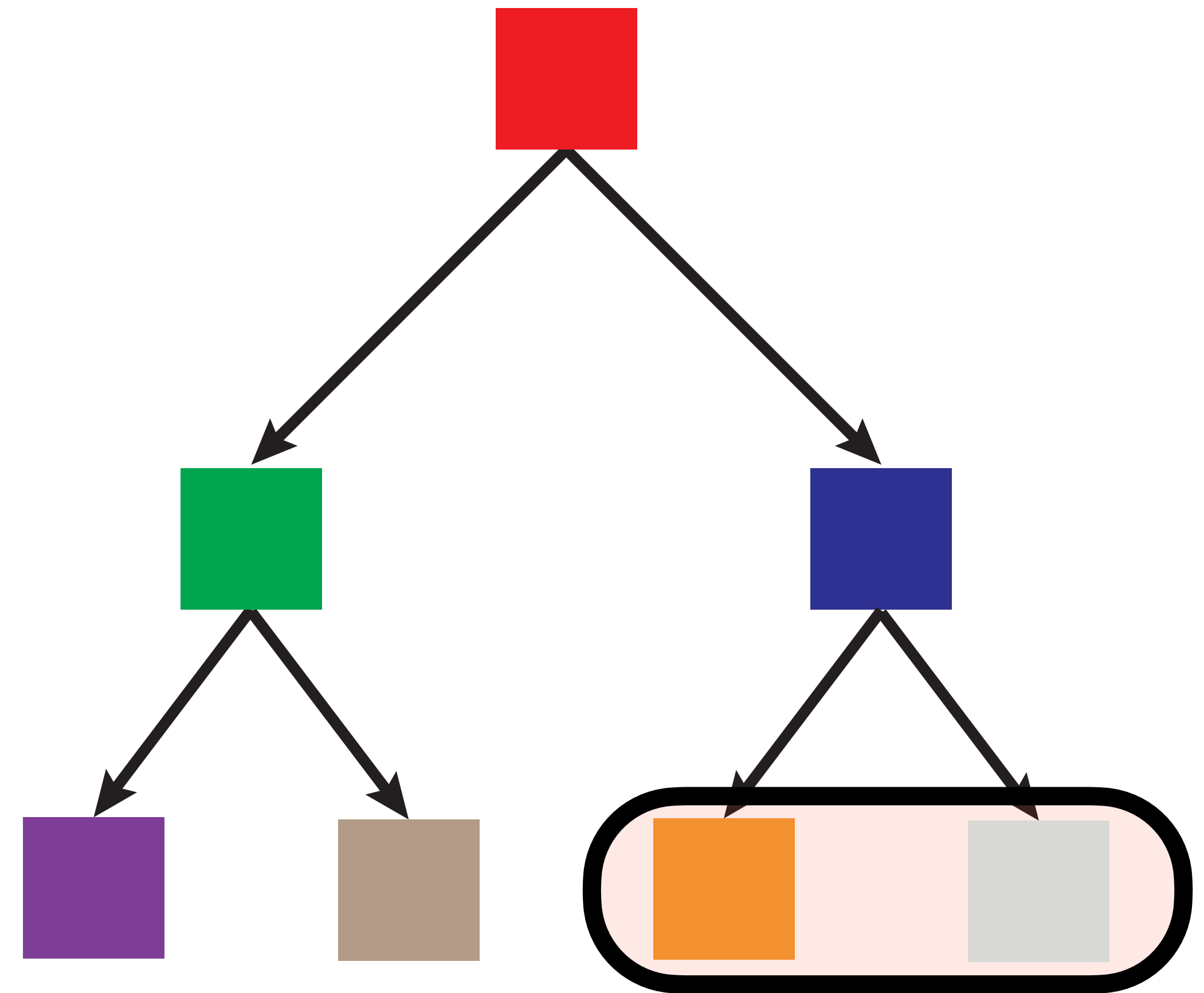


Nested dissection reordering

Input mesh

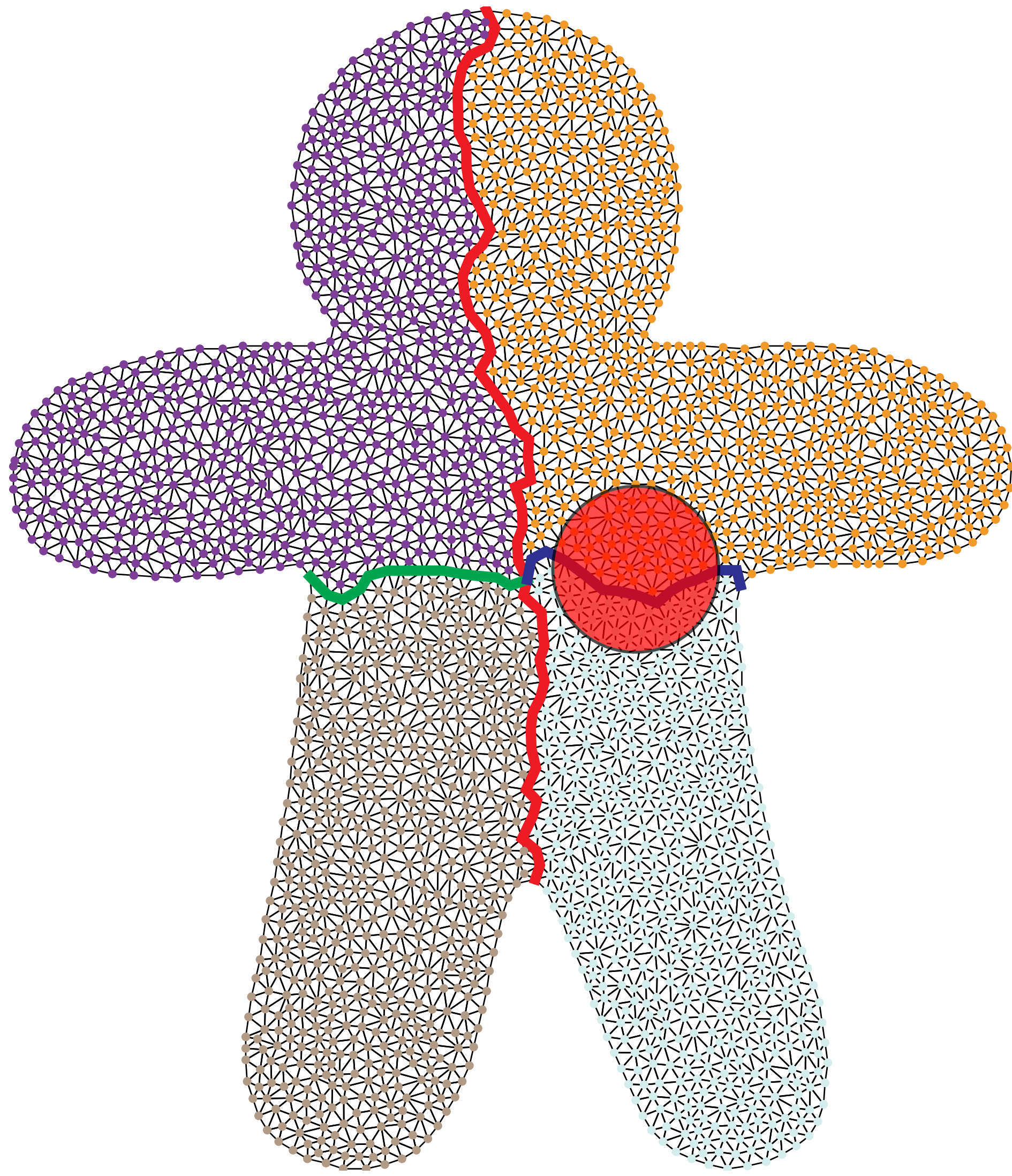


Elimination tree

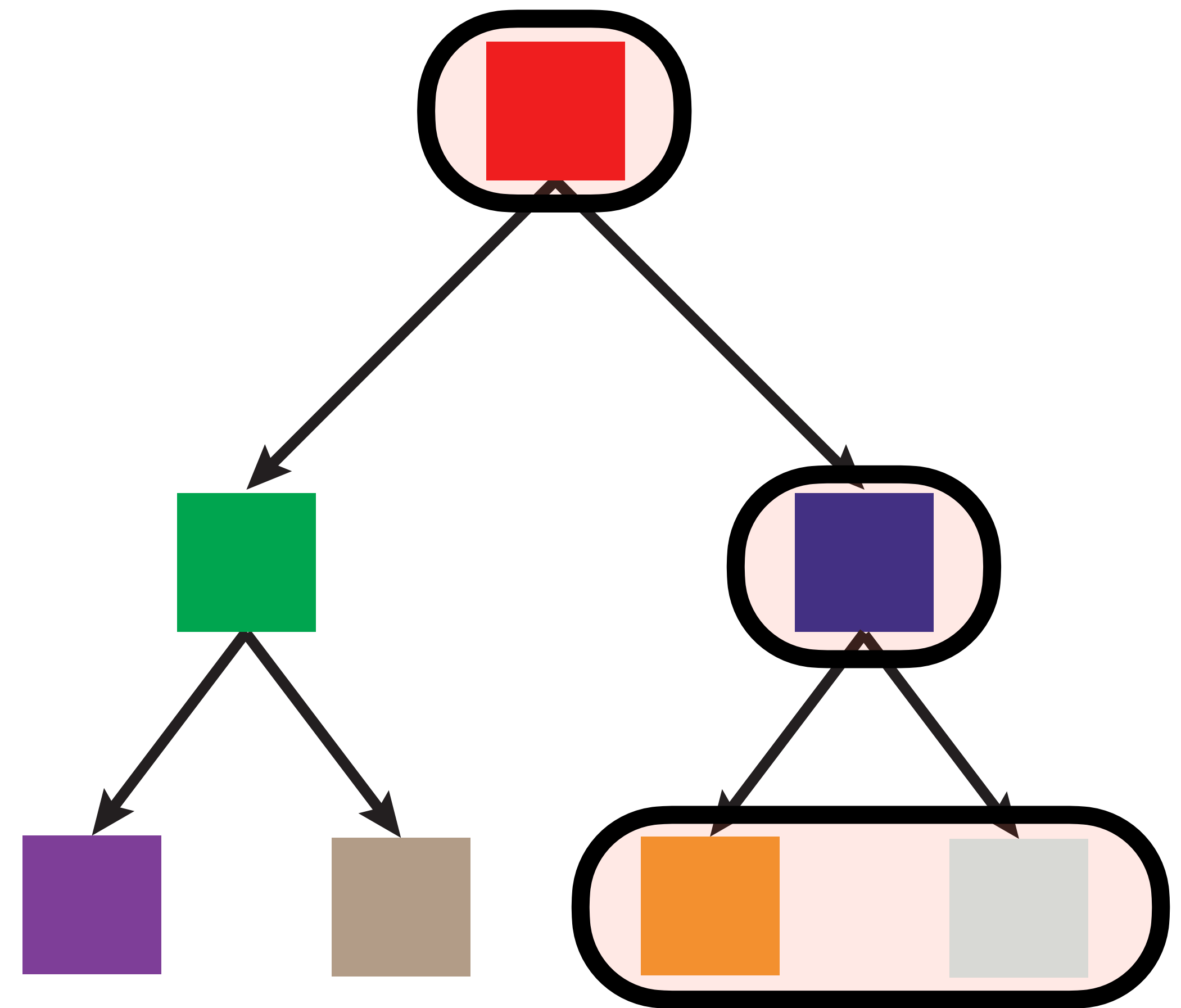


Nested dissection reordering

Input mesh

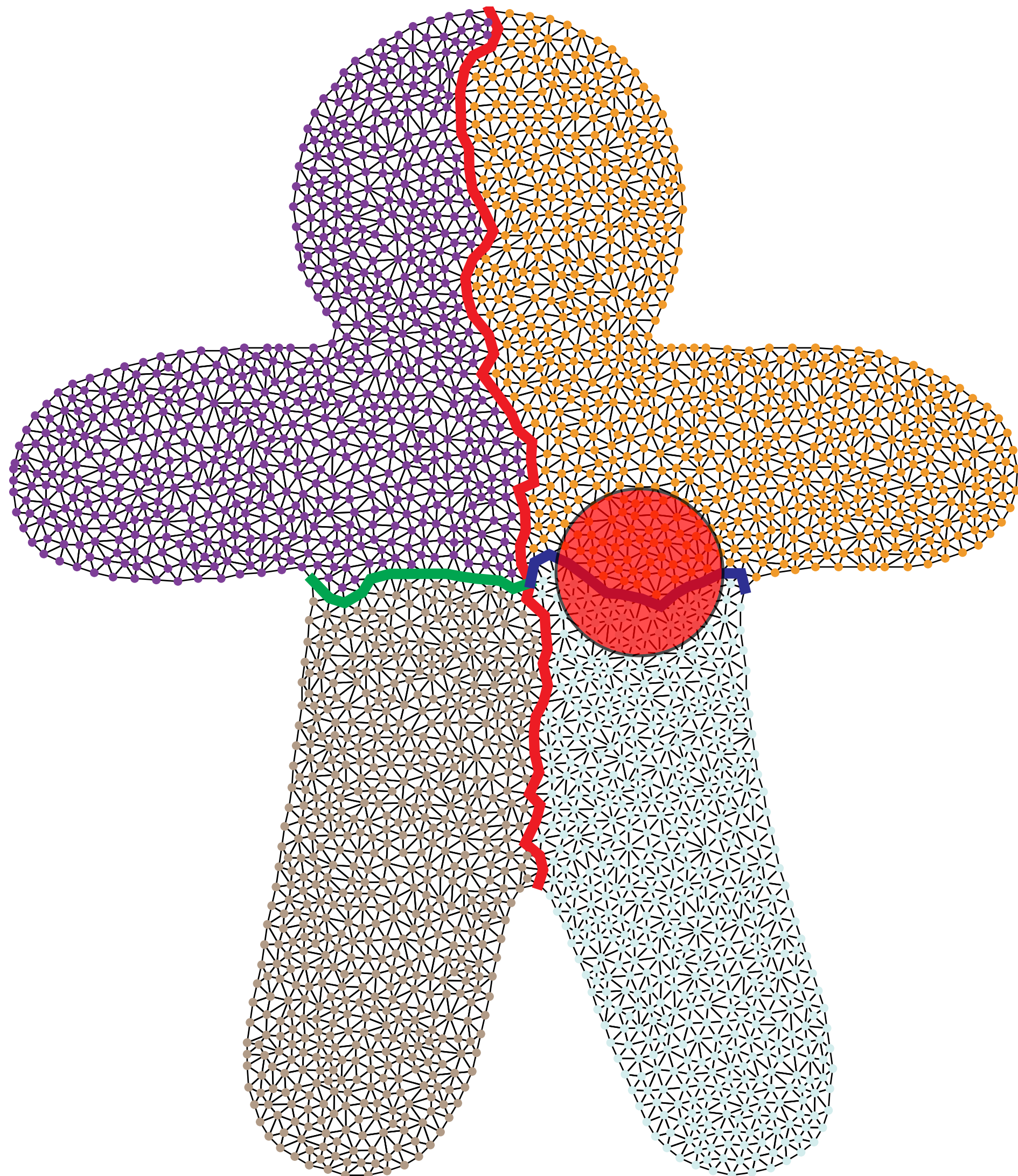


Elimination tree

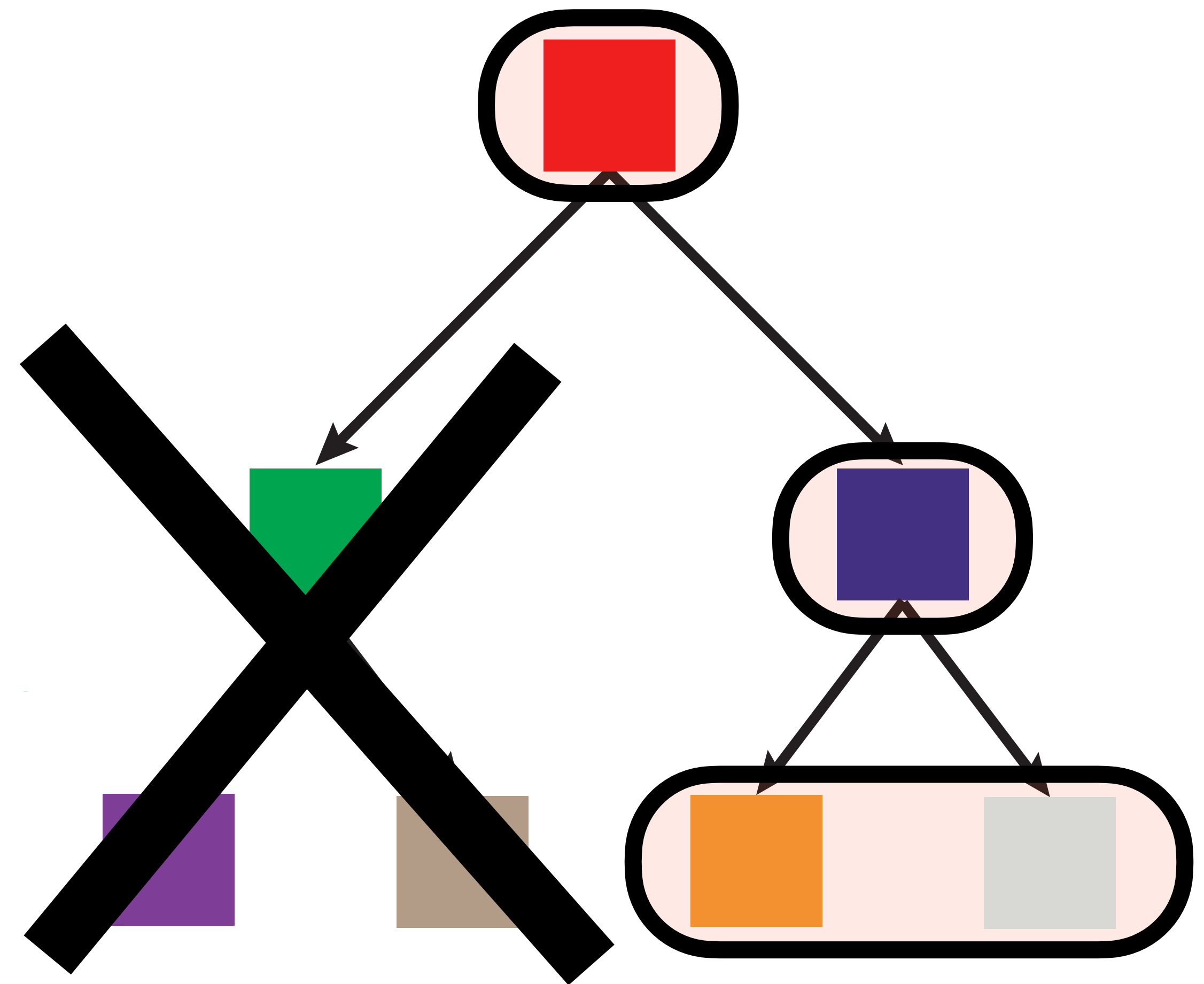


Nested dissection reordering

Input mesh

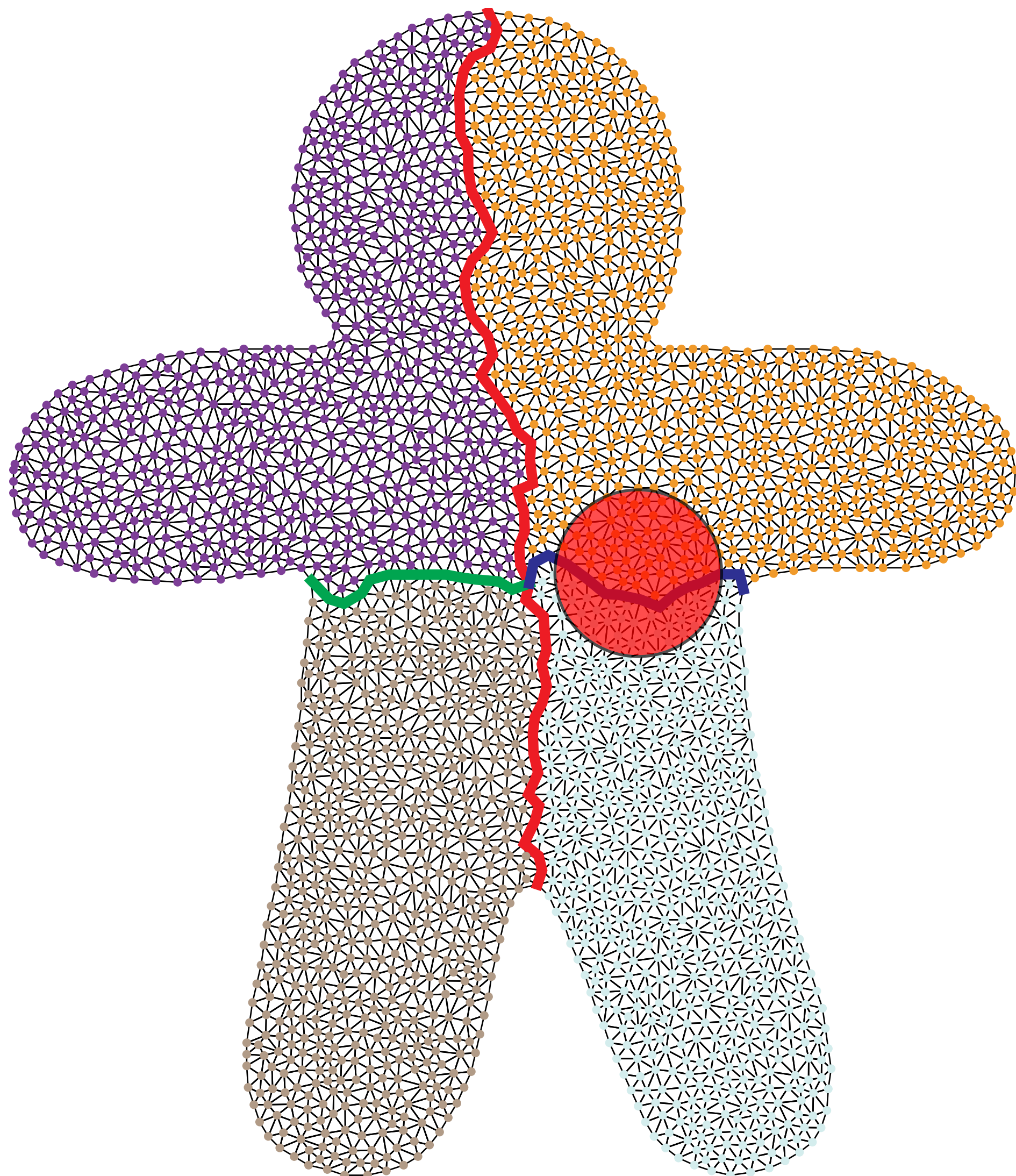


Elimination tree

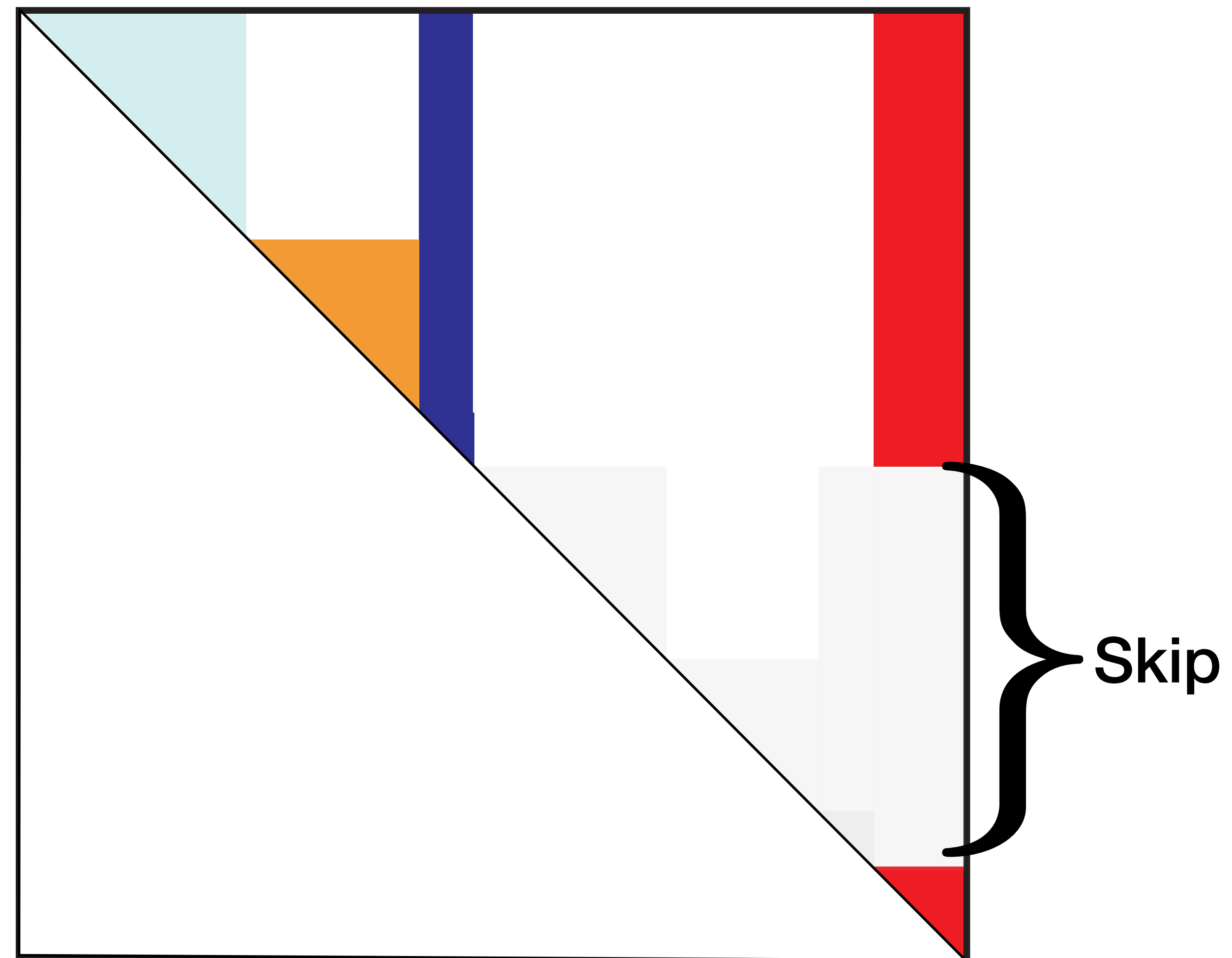


Nested dissection reordering

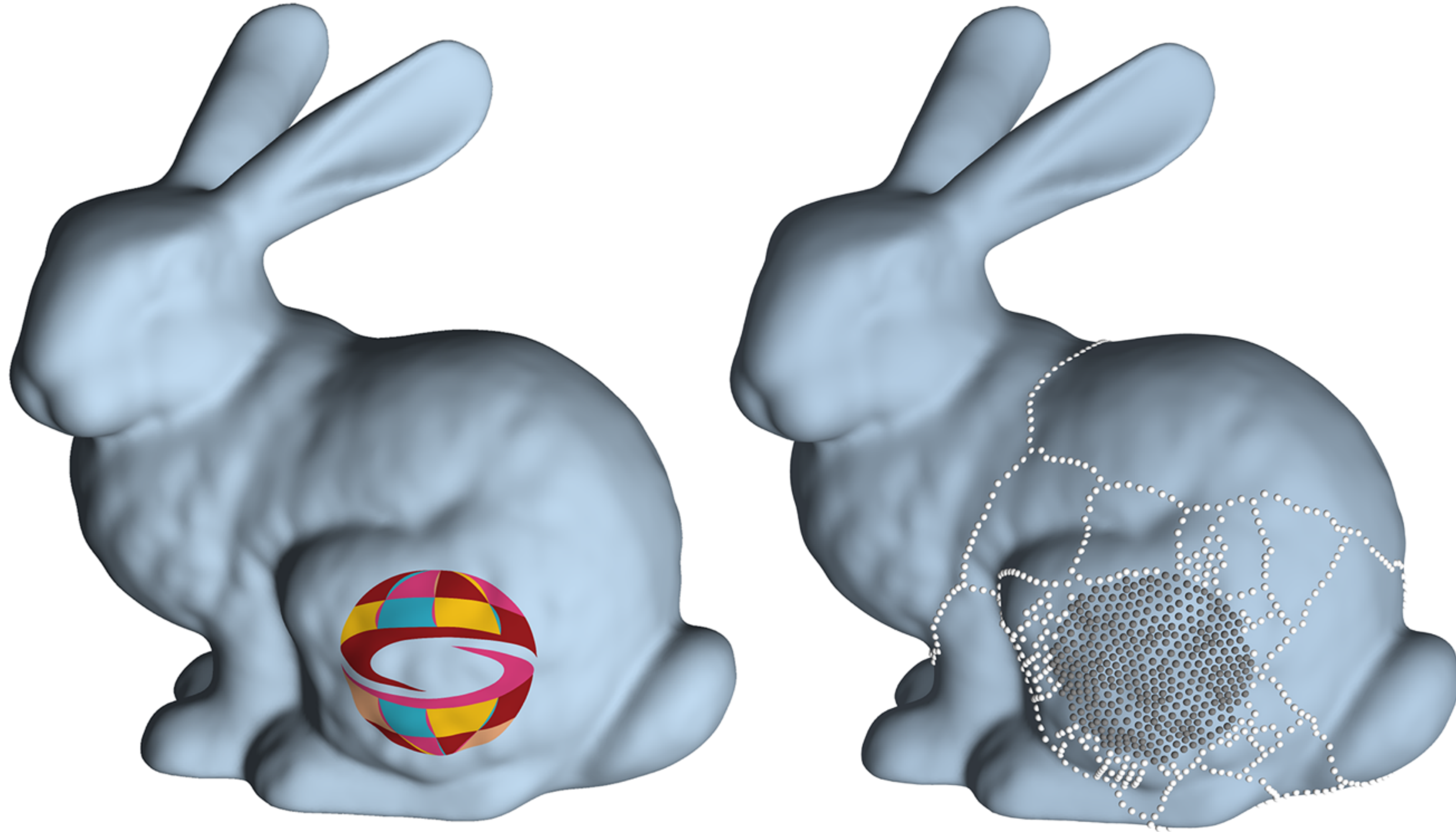
Input mesh



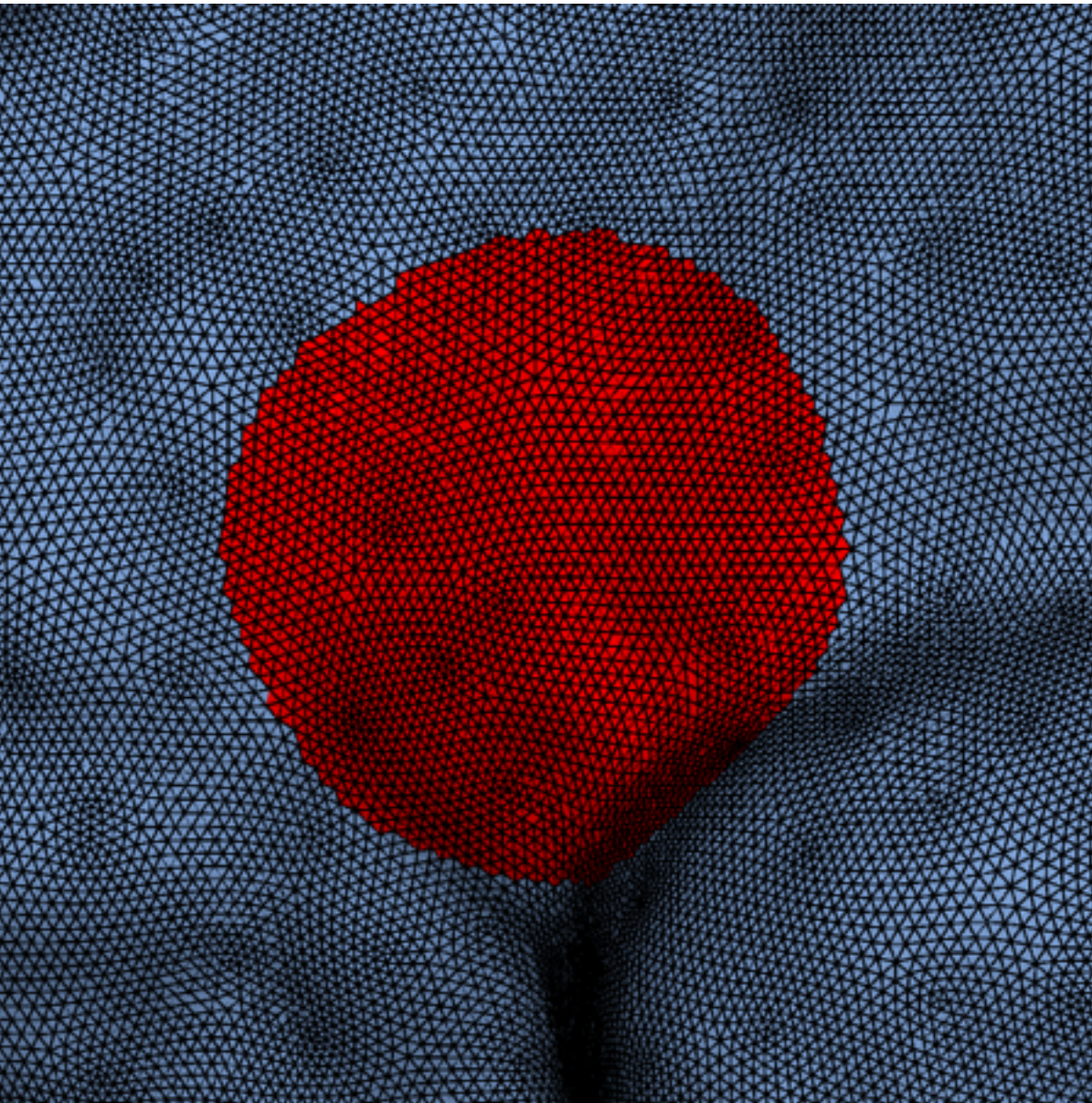
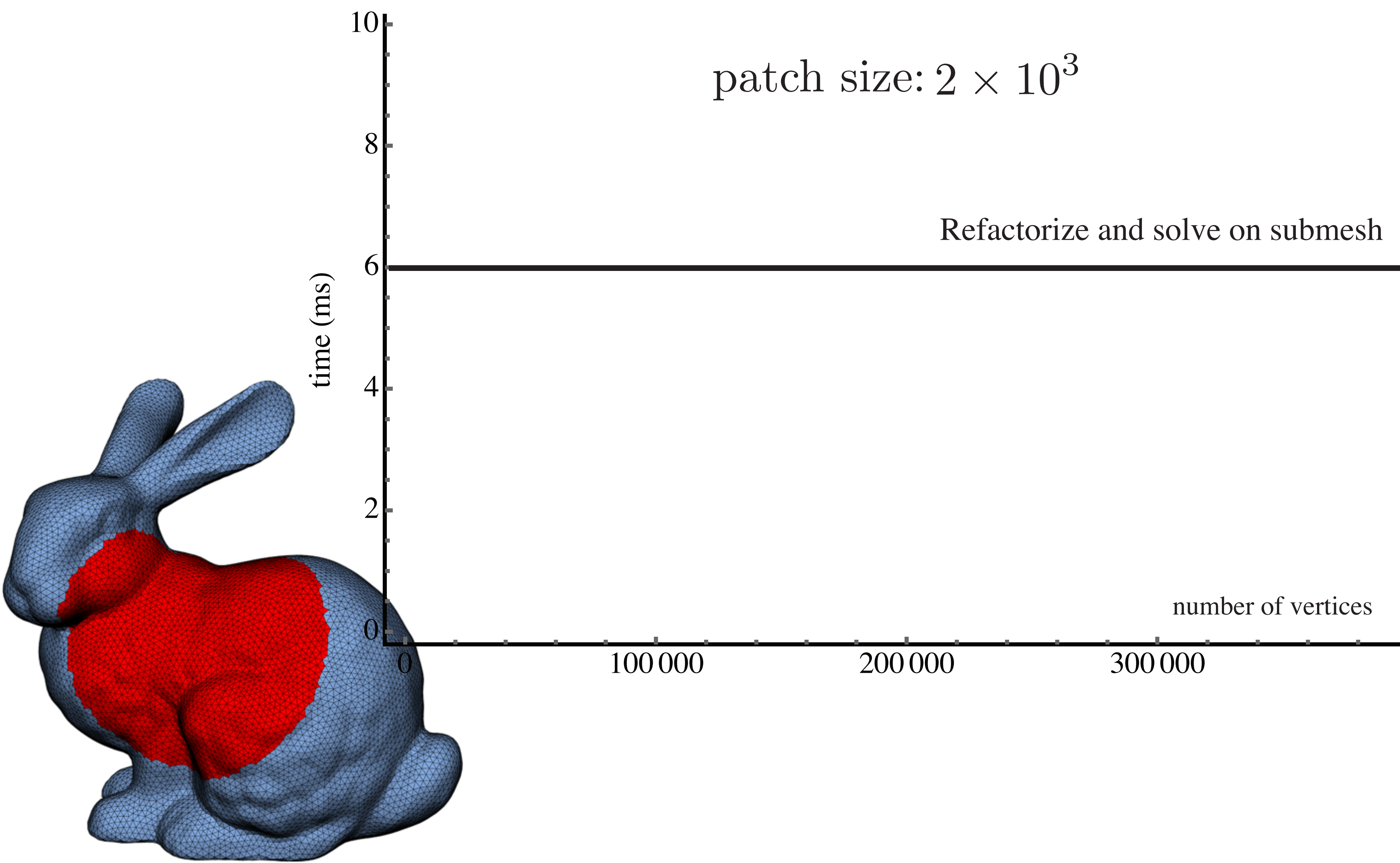
Cholesky factor



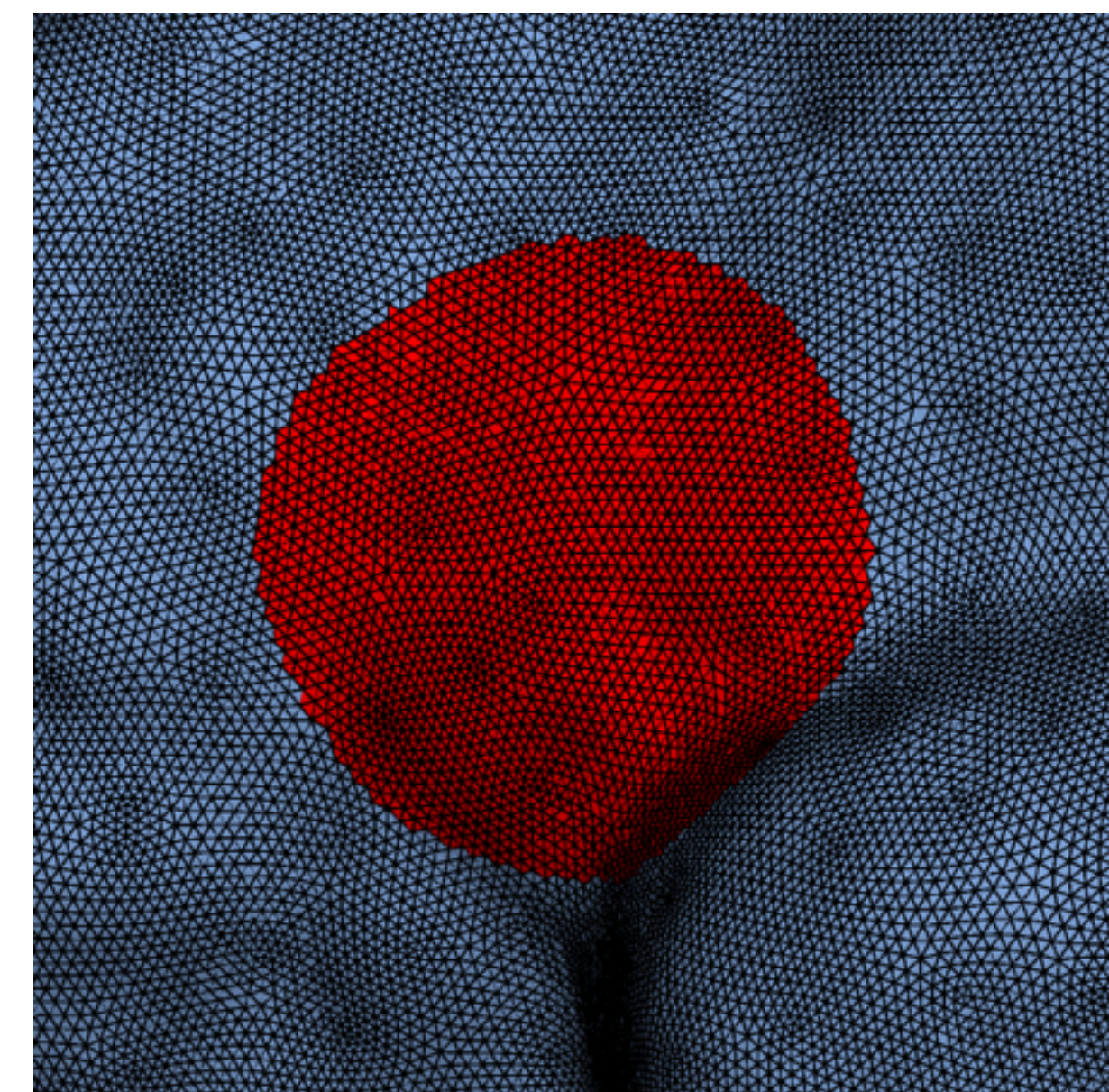
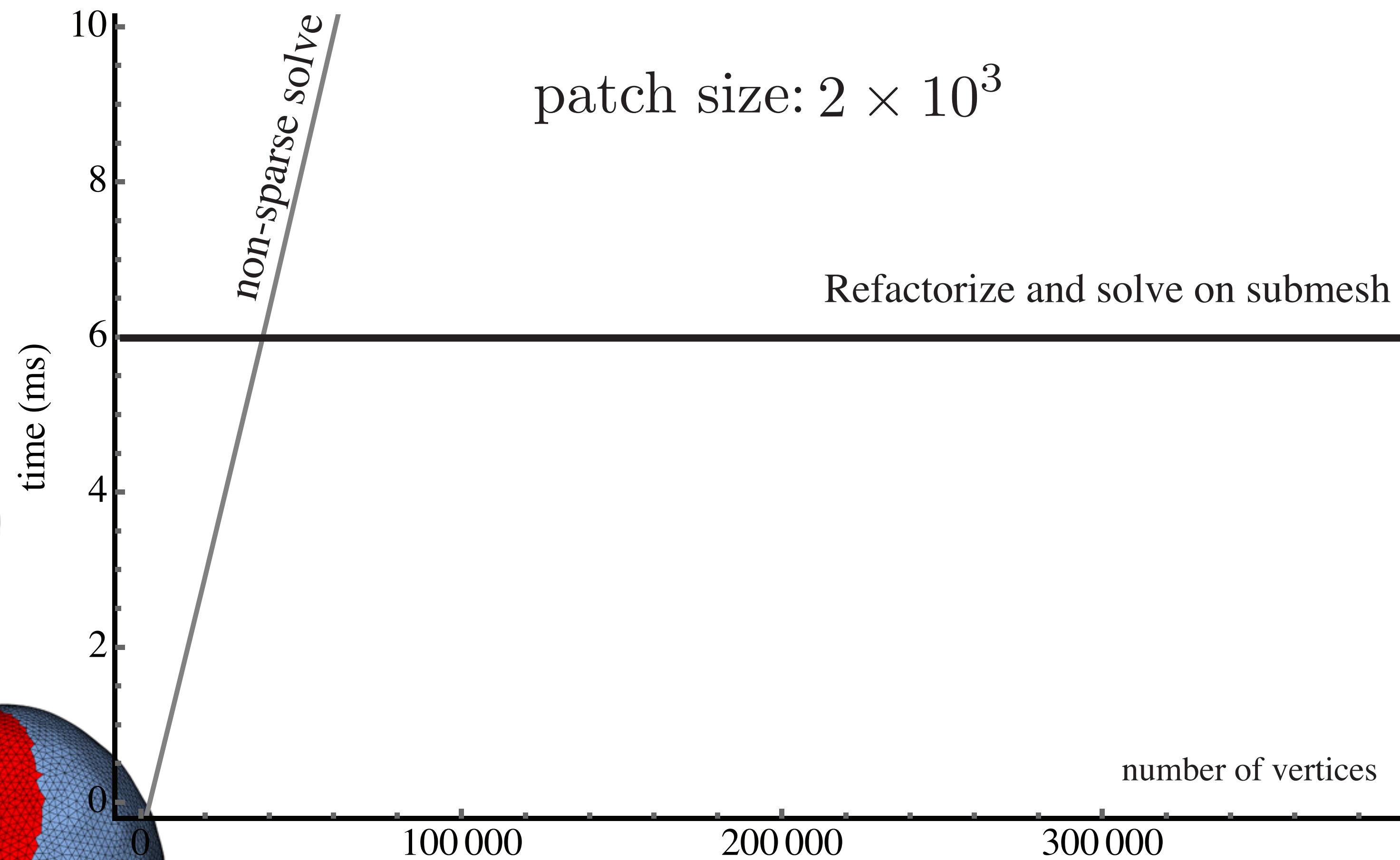
Applications: Parameterization



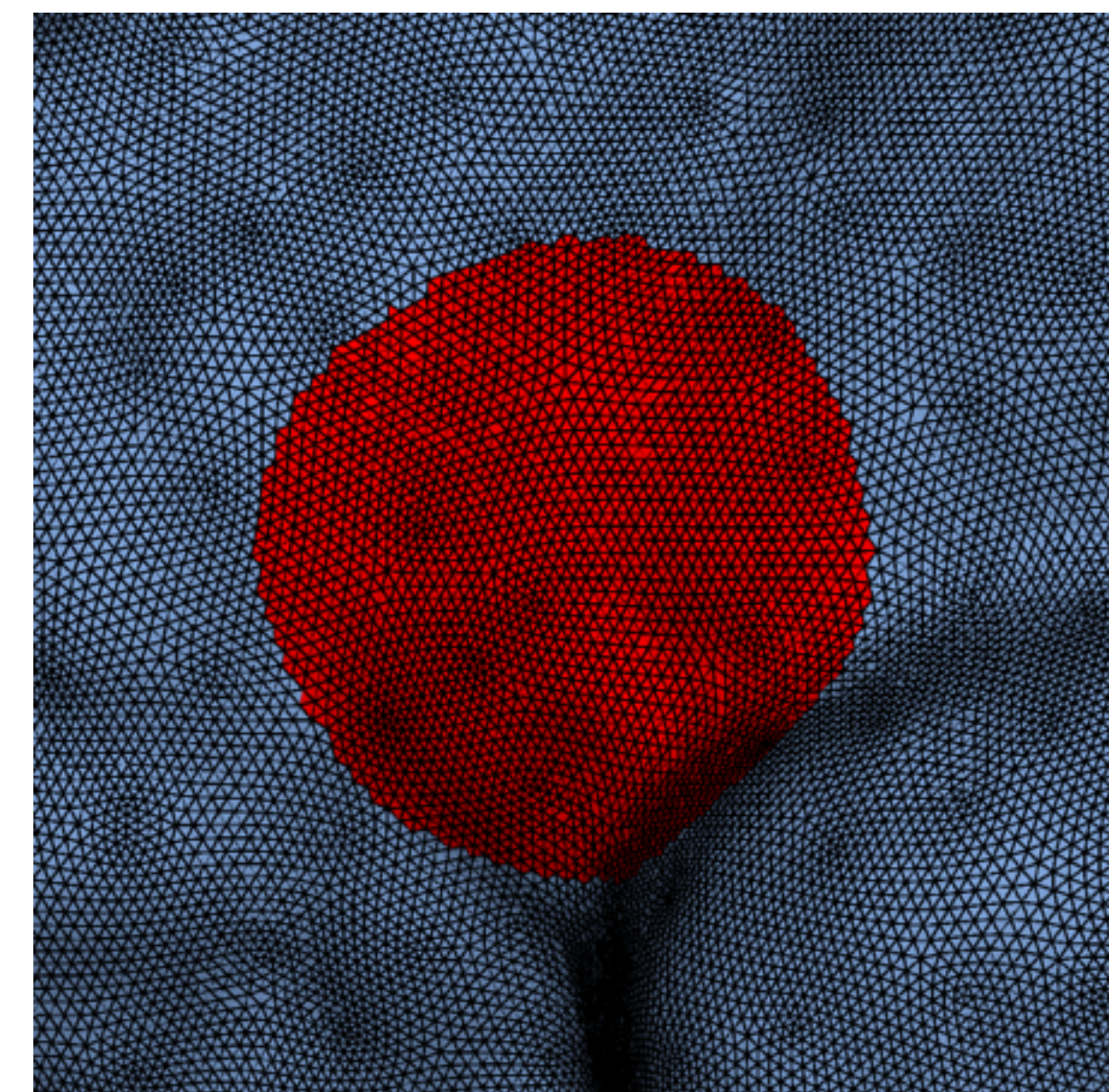
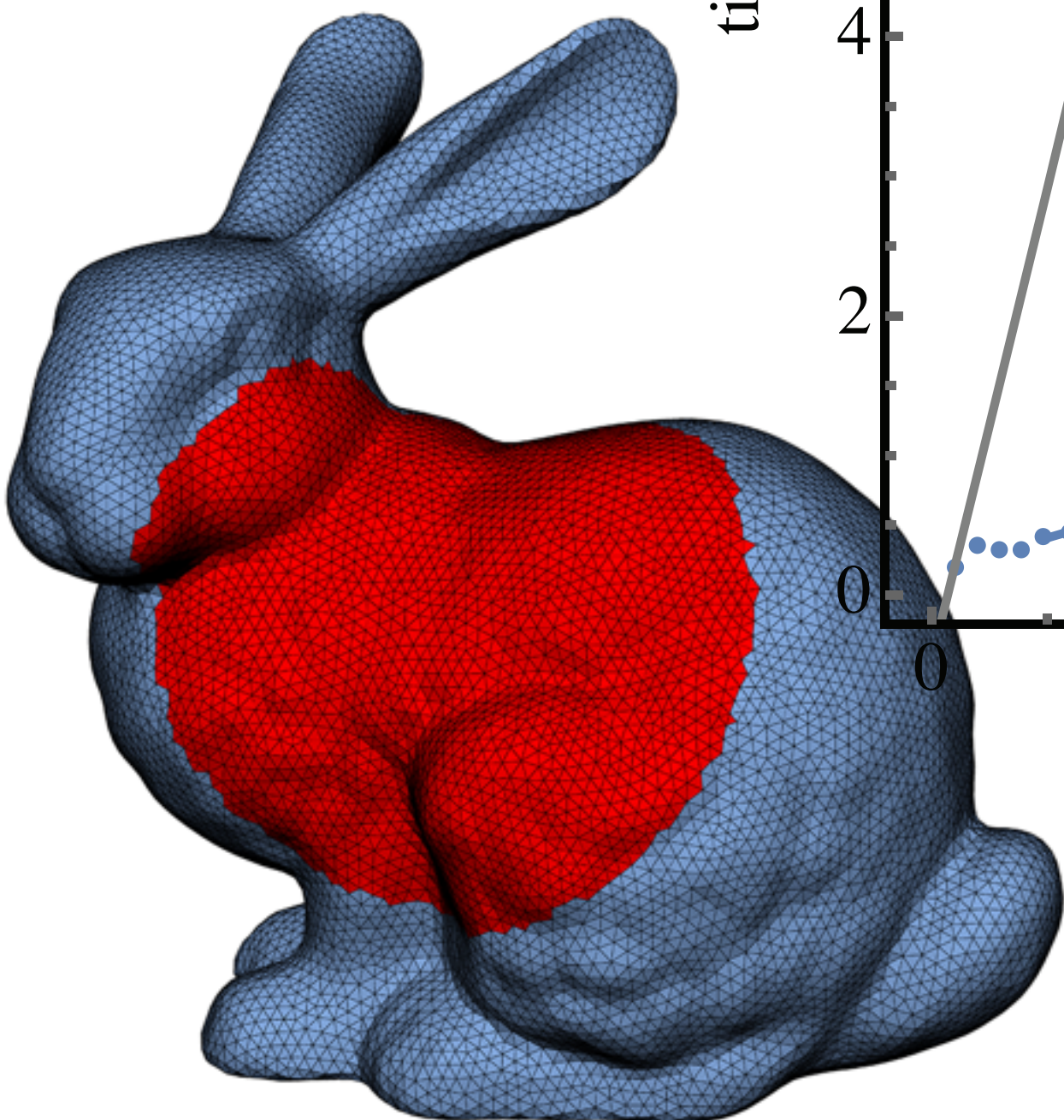
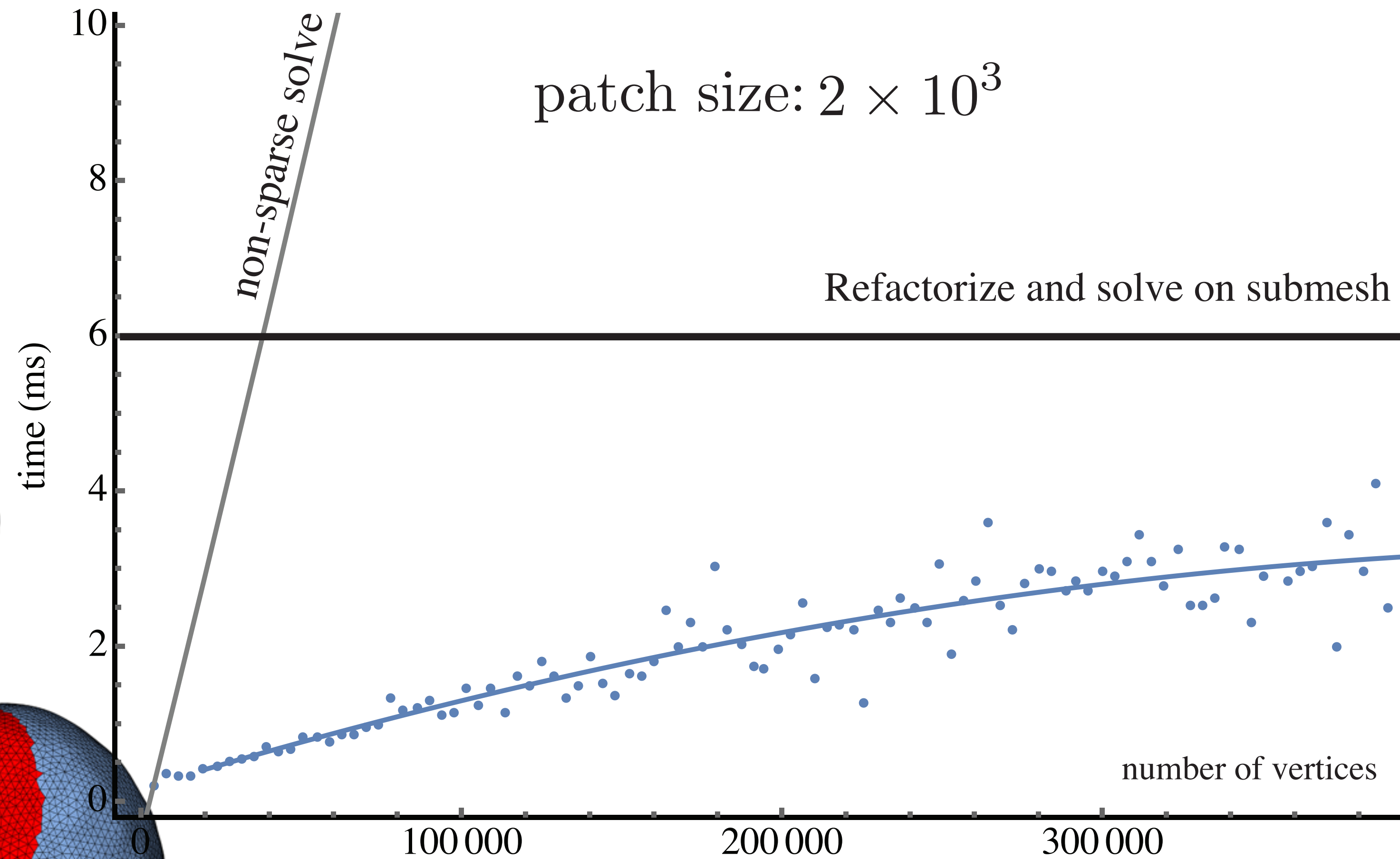
Is it worth it?



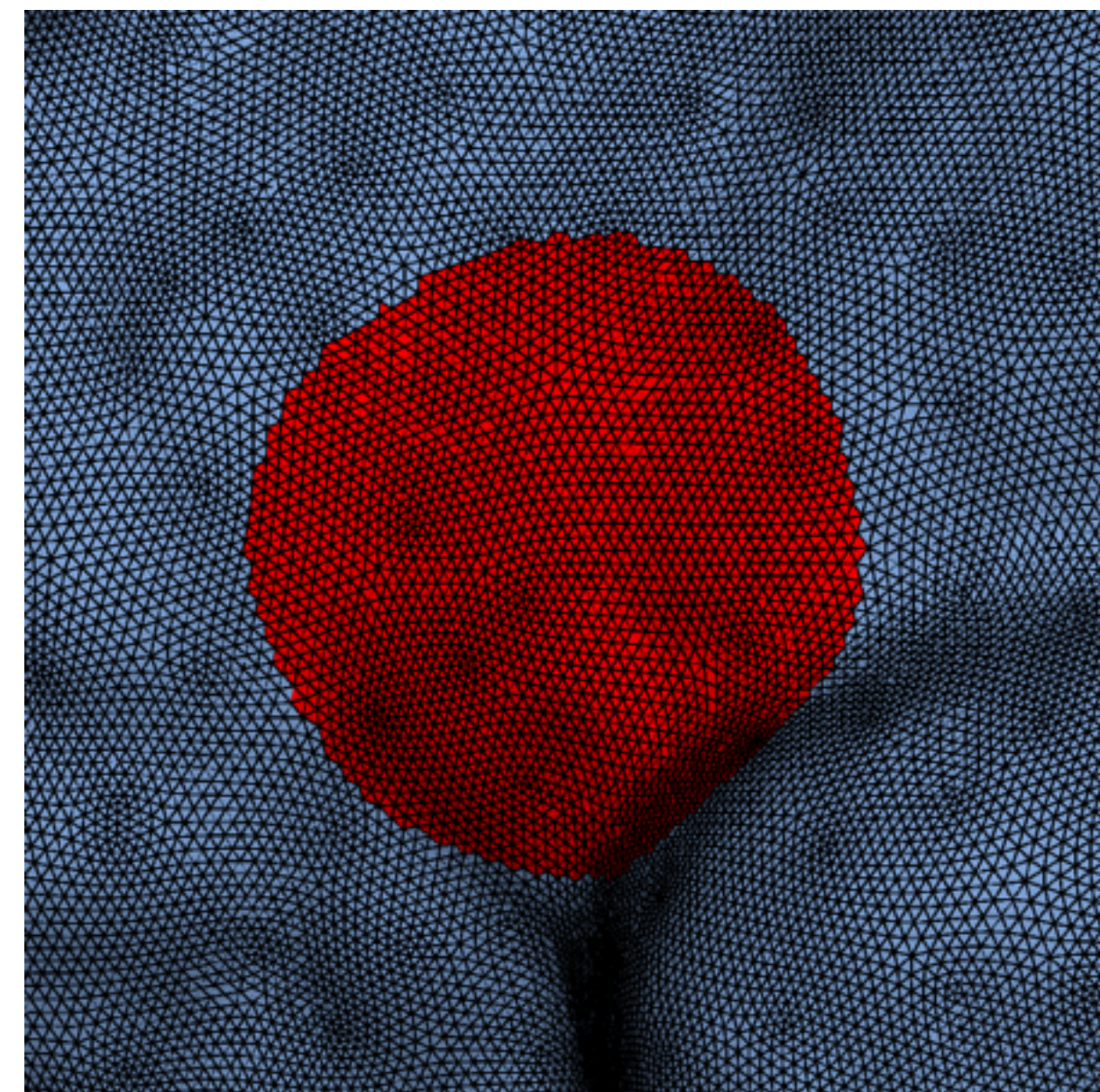
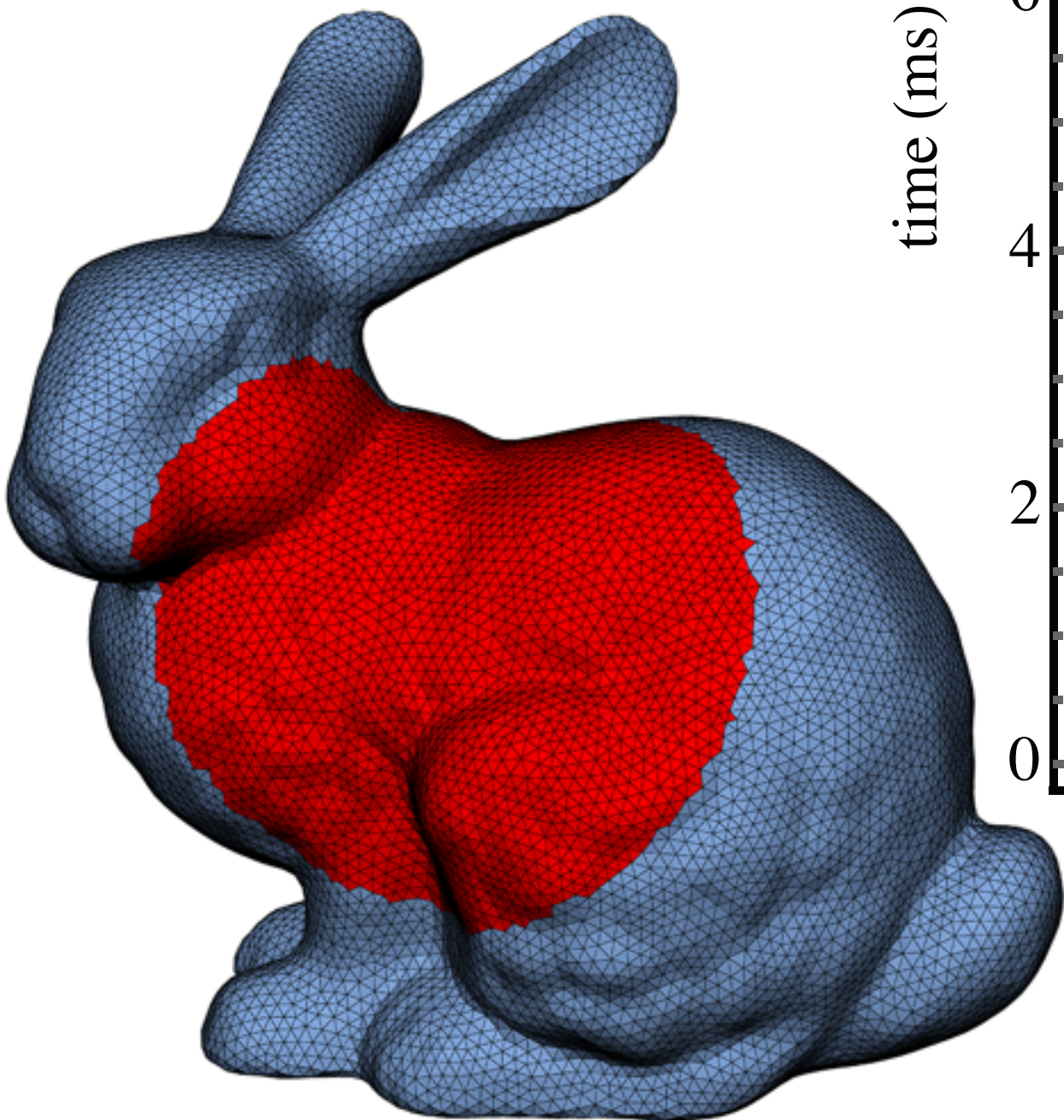
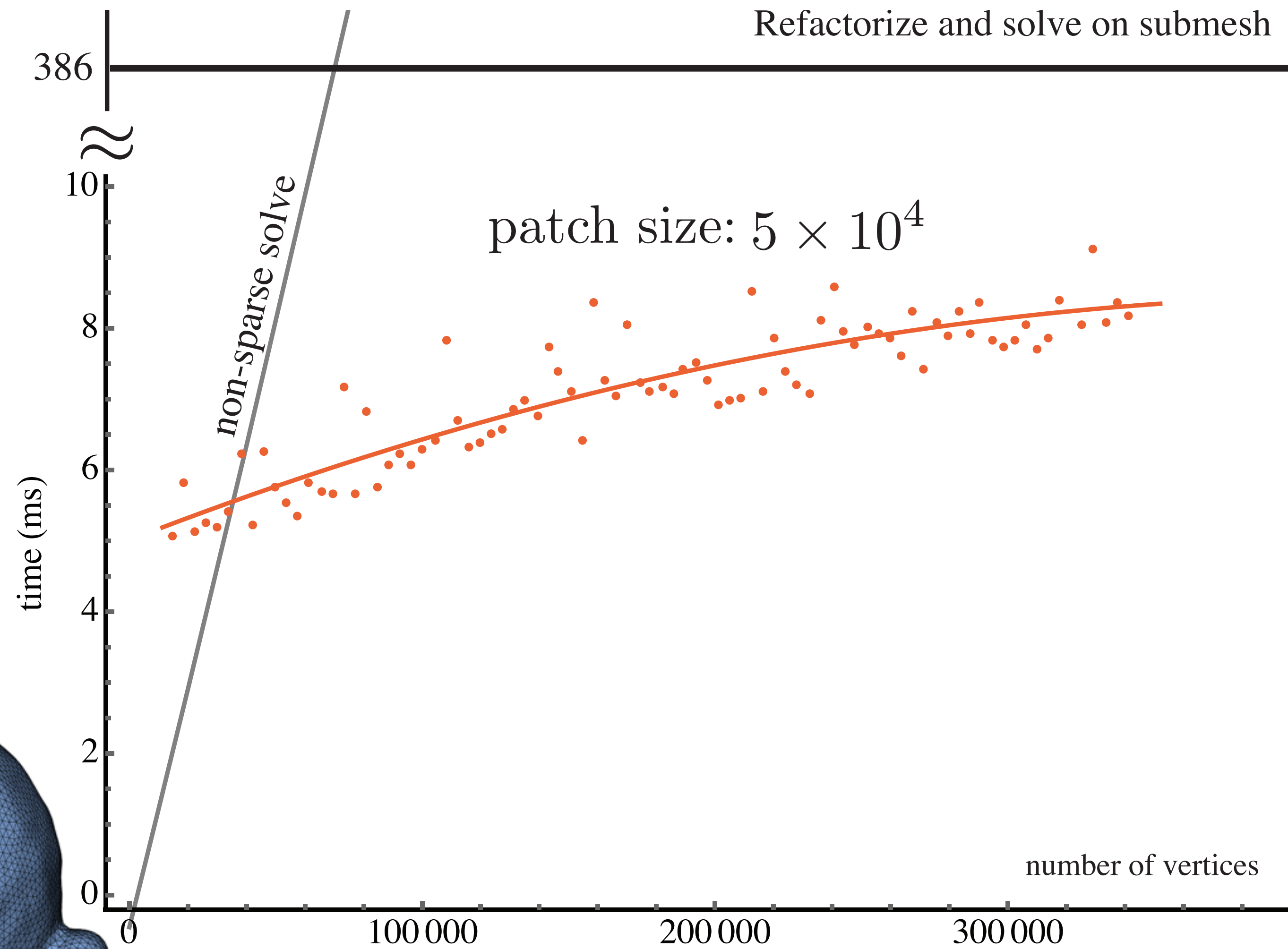
Is it worth it?



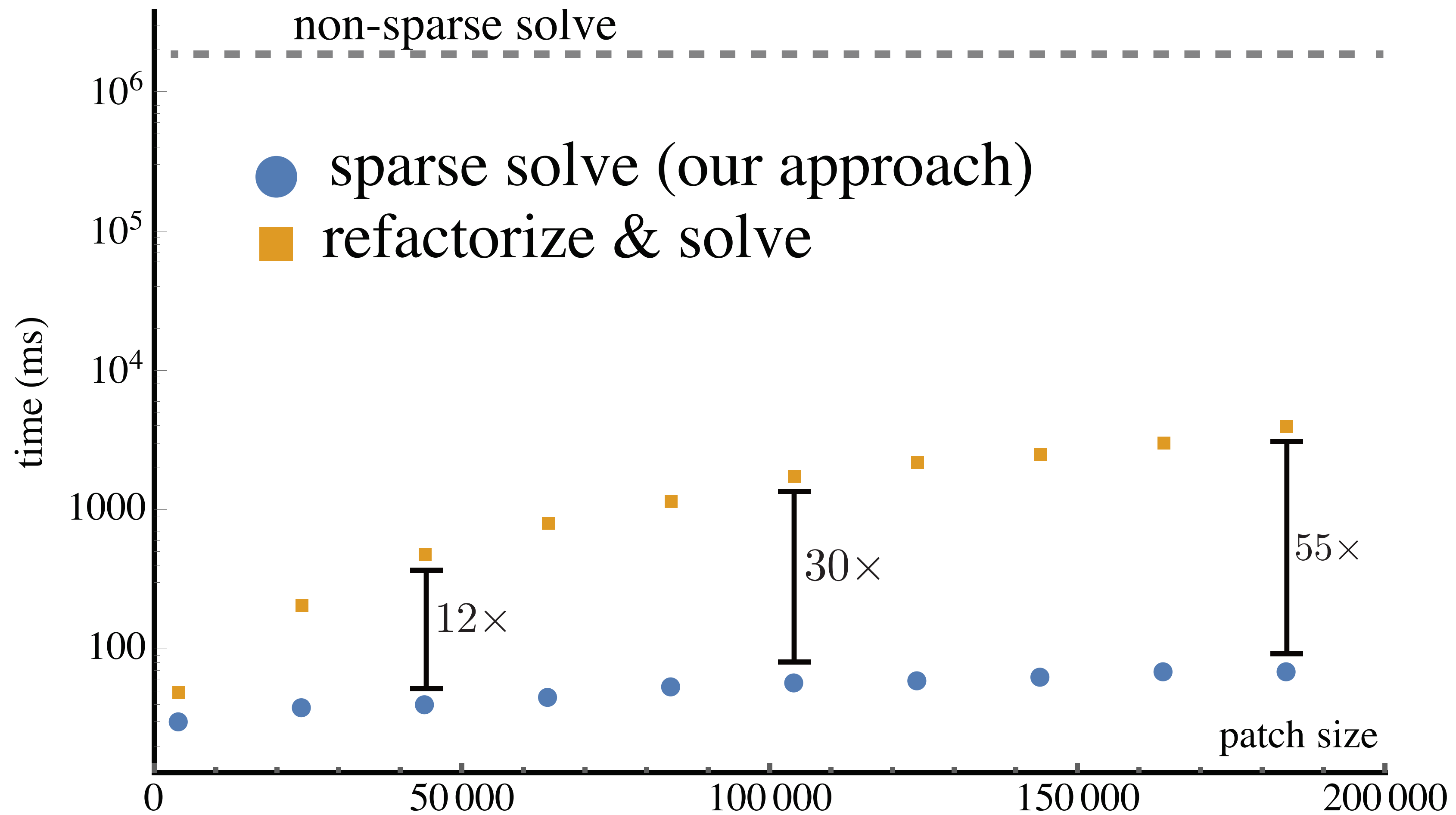
Is it worth it?



Is it worth it?



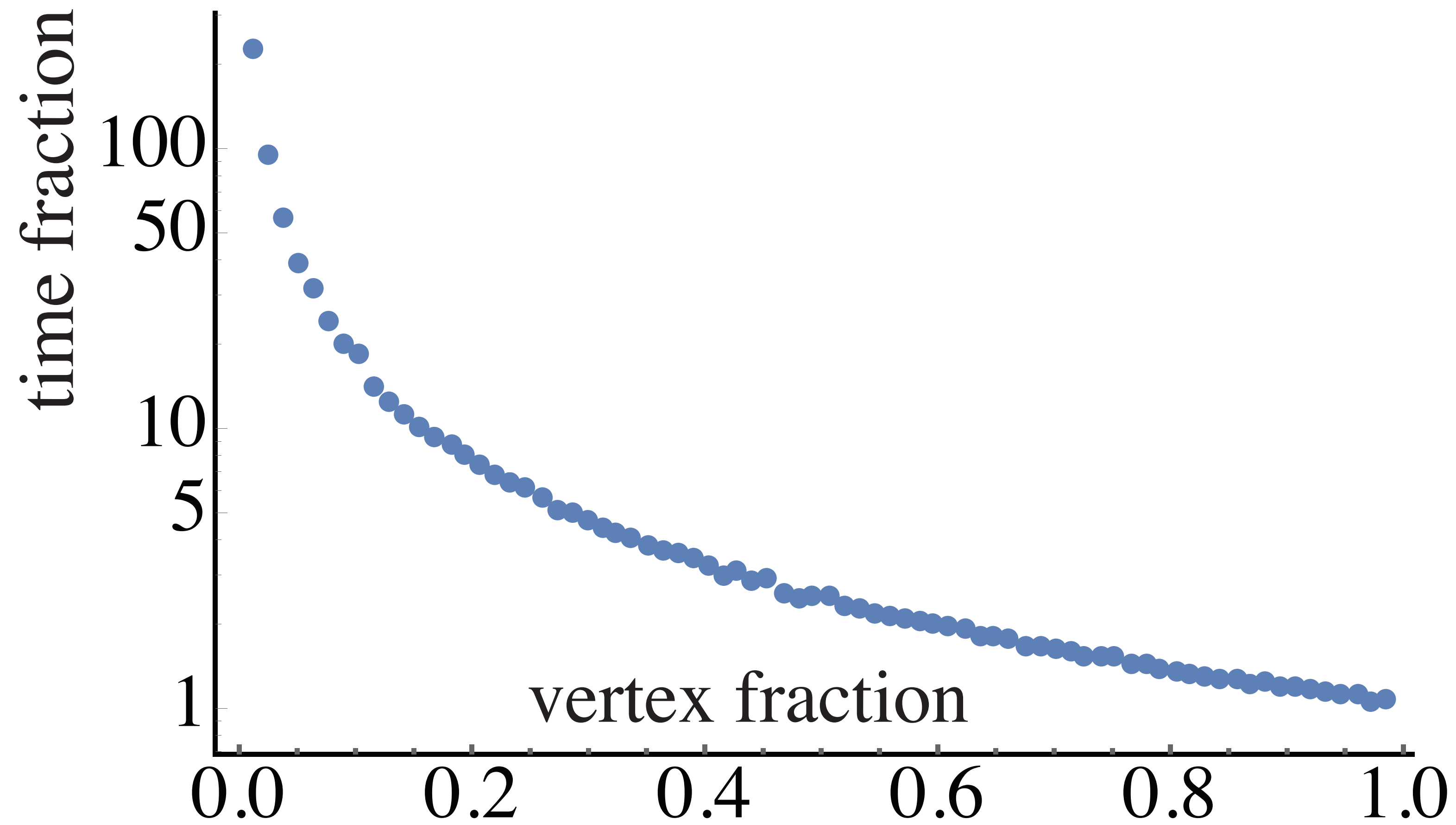
Is it worth it?



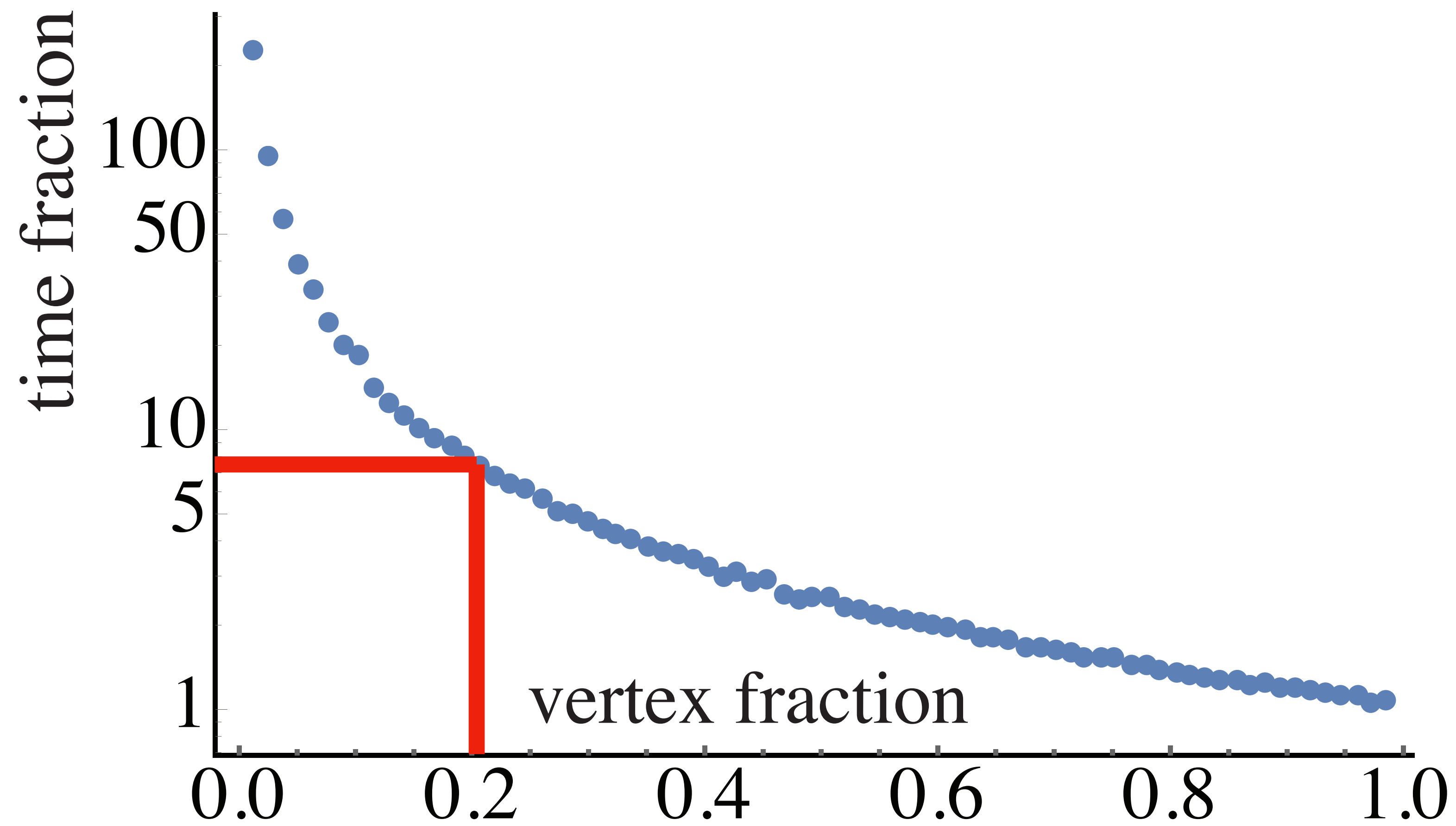
Amortization of factorization time

$$n \times \left(\begin{array}{l} \mathbf{A}_1 = \mathbf{A} \left(\text{red sphere} \right) \\ \mathbf{A}_1 = \mathbf{LDL}^T \end{array} \right) \quad \mathbf{vs} \quad \begin{array}{l} \mathbf{A} = \mathbf{A} \left(\text{blue rabbit} \right) \\ \mathbf{A} = \mathbf{LDL}^T \end{array}$$

Amortization of factorization time



Amortization of factorization time

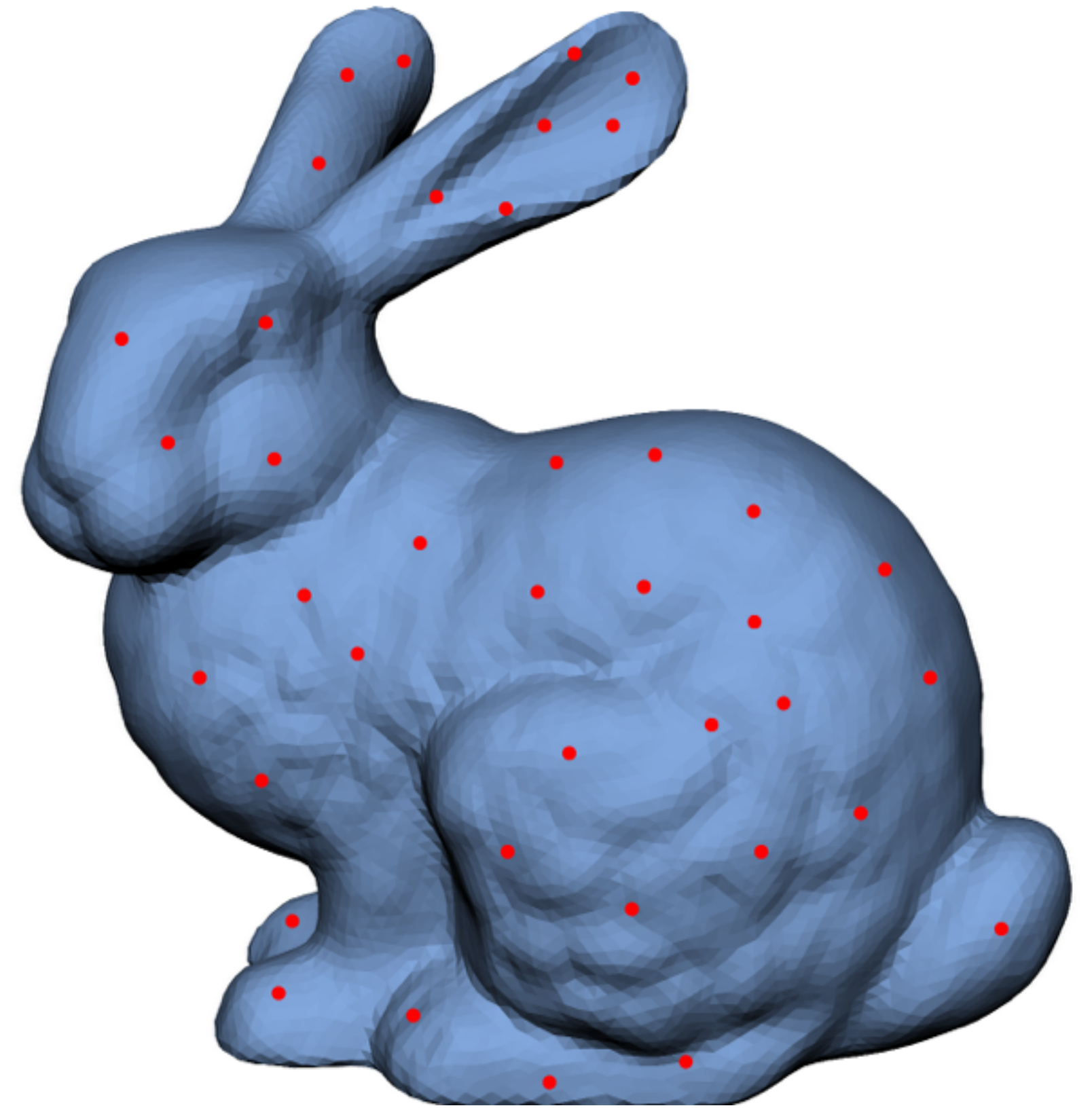
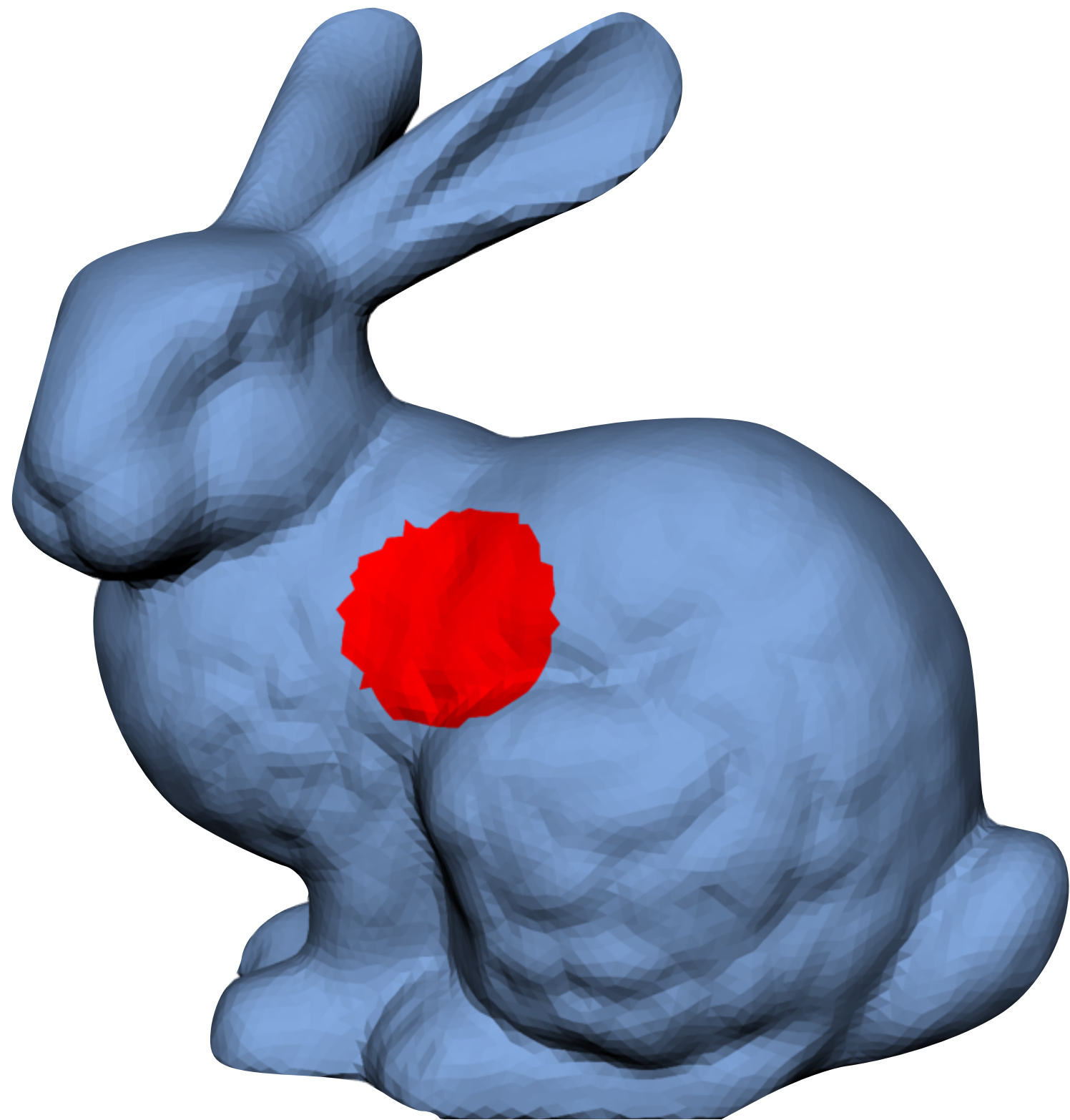


Limitations

- Only works if the requested values are local.

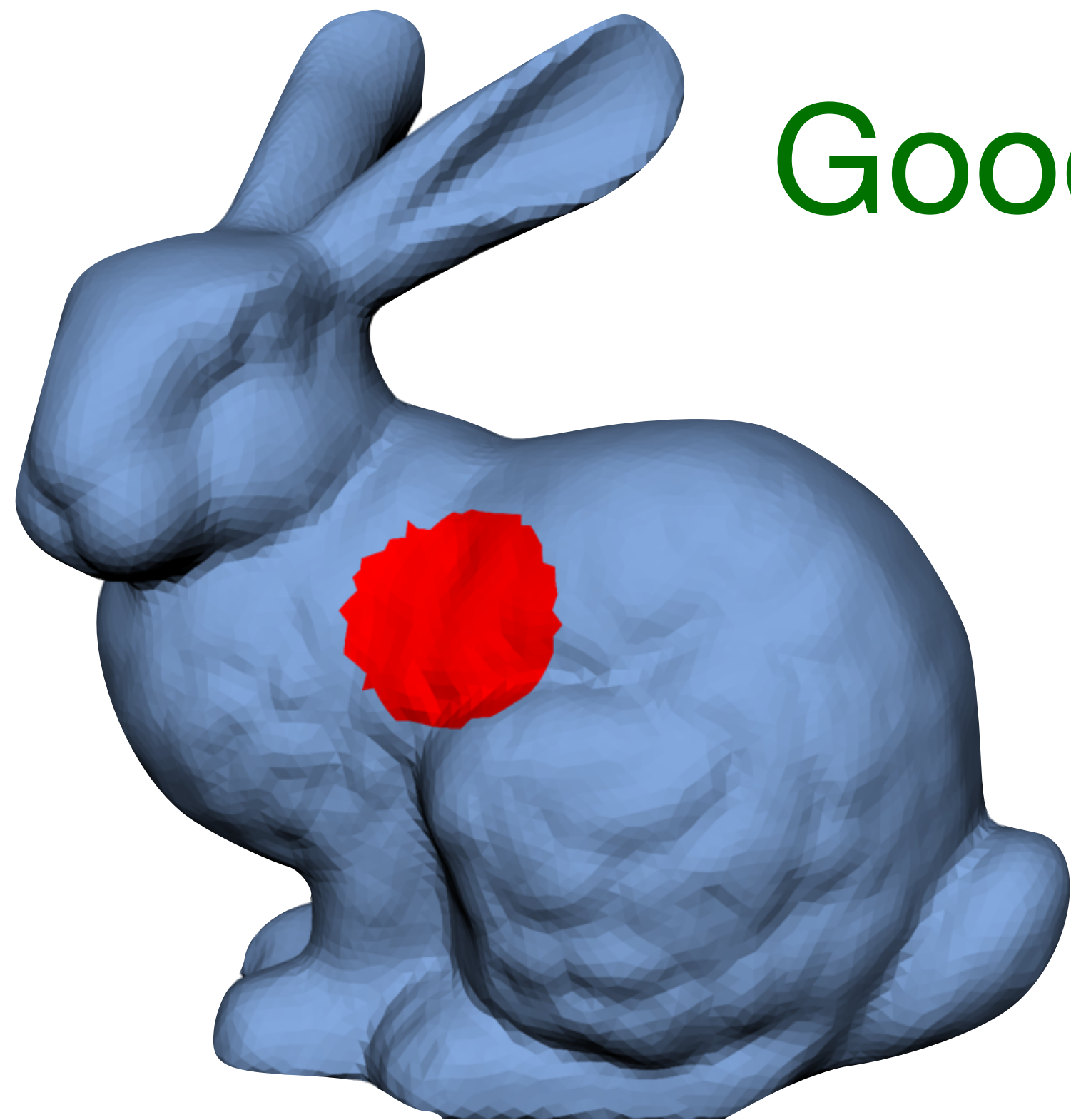
Limitations

- Only works if the requested values are local.

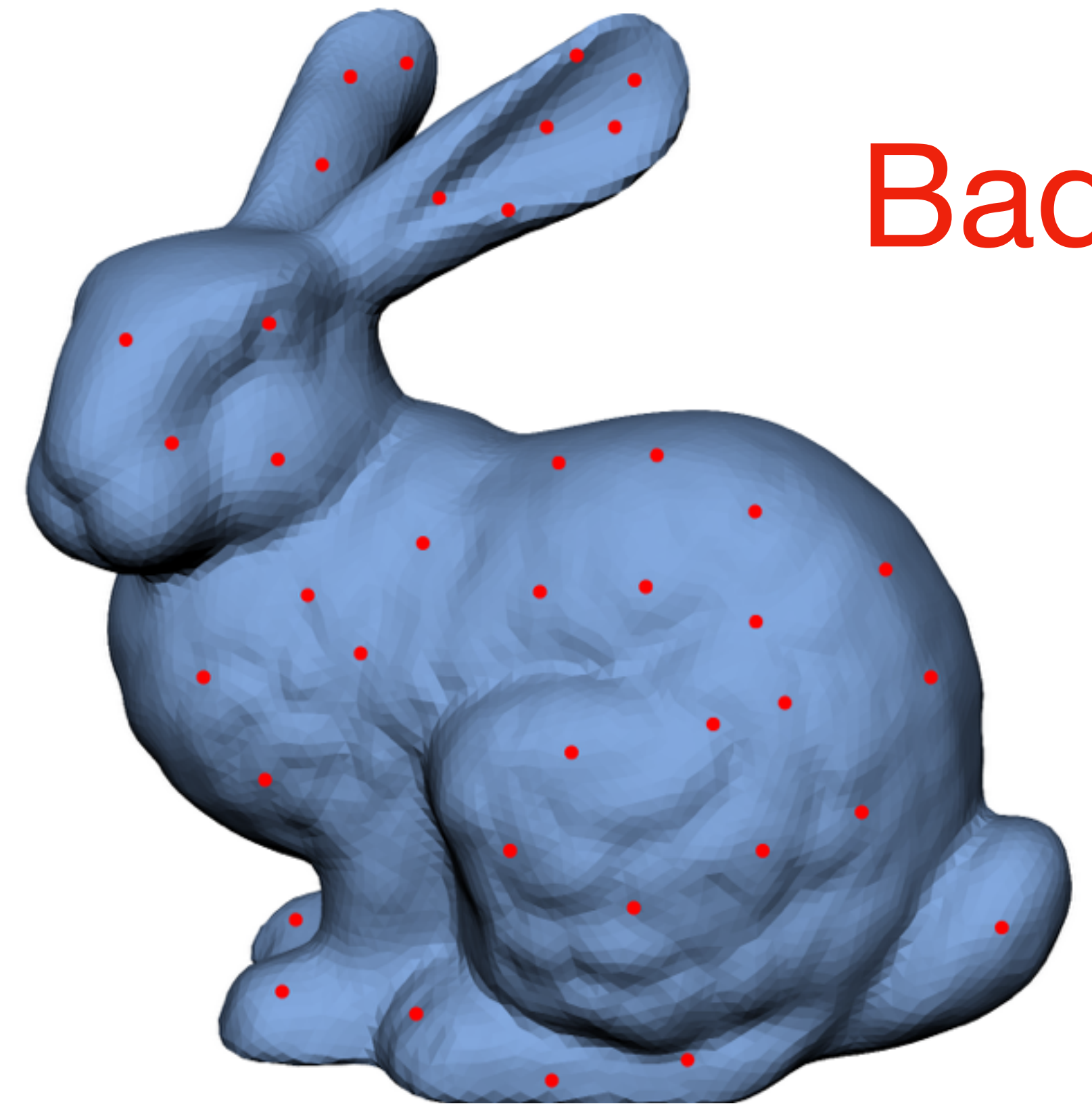


Limitations

- Only works if the requested values are local.



Good



Bad

Limitations

- Only works if the requested values are local.
- The local problem needs to be formulated in terms of a global operator.

Conclusion

- Back-substitution can be significantly accelerated if only a few values are requested.
- If requested values belong to a localized patch on a mesh, performance benefits the most (because of nested dissection).
- Considering sparsity in the solution can improve performance by orders of magnitude.
- The method is trivial to implement.

Thank you for your attention!

Contact: philipp.herholz@tu-berlin.de

Applications: Auto-diffusion

- Compute heat diffusion from a single source vertex and evaluate the amount of heat staying at the source after some time.
- Only one value in \mathbf{x} is required.

