Sparse LR-Zerlegung

Numerische Mathematik 1
WS 2011/12

Definition

Für $A = [a_{ij}] \in \mathbb{R}^{n,n}$ bezeichne

$$\mathrm{nnz}(A) := \left|\{(i,j) \mid a_{ij} \neq 0\}\right|$$

die Anzahl der nicht-Null Elemente von A.

Man nennt A **sparse**, wenn die Menge der nicht-Null Elemente von A klein ist, d.h.

$$\mathrm{nnz}(A) \ll n^2.$$

Speicherung Sparser Matrizen

Sparse Matrizen $A = [a_{ij}] \in \mathbb{R}^{n,n}$ kann man durch drei Arrays der Länge $\mathrm{nnz}(A)$ speichern.

Die Matrix

$$\begin{bmatrix} 0 & 0 & 3 \\ -1 & 0 & 0 \\ -2 & 3 & 0 \end{bmatrix}$$

kann man in der Form

```
INDX = [  2,  3,  3,  1 ],
JNDX = [  1,  1,  2,  3 ],
VALS = [ -1, -2,  3,  3 ],
```

speichern.

Andere Speicherformate: CSC, CSR, Bandmatrizen...

Problem

Sei $A \in \mathbb{R}^{n,n}$ sparse und sei n groß.

Bei der Berechnung einer LR-Zerlegung (ohne Pivotisierung)

$$LR = A$$

kann es passieren, dass L und R nicht mehr sparse sind.

Beispiel

Der erste Schritt in der LR Zerlegung der Matrix

$$A_1 := \begin{bmatrix} -1 & -1 & \cdots & -1 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix}$$

führt auf

Beispiel

Der erste Schritt in der LR Zerlegung der Matrix

$$A_1 := \begin{bmatrix} -1 & -1 & \cdots & -1 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix}$$

führt auf

$$\rightarrow \begin{bmatrix} -1 & -1 & -1 & \cdots & -1 \\ \hline 1 & 2 & 1 & \cdots & 1 \\ 1 & 1 & 2 & \ddots & \vdots \\ 1 & \vdots & \ddots & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 2 \end{bmatrix} =: LR.$$

Beispiel

Die LR-Zerlegung von

$$A_1 := \begin{bmatrix} -1 & -1 & \cdots & -1 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix}$$

ist voll besetzt.

Fill-in-Rate $:= \frac{\mathrm{nnz}(LR)}{\mathrm{nnz}(A)} = \frac{n^2}{n+2\cdot(n-1)} \approx \frac{n^2}{3n} = \frac{1}{3}n$

Beispiel

Vertauscht man die erste und letzte Zeile und die erste und letzte Spalte von

$$A_1 := \begin{bmatrix} -1 & -1 & \cdots & -1 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix}.$$

erhält man die Matrix

$$A_2 := \begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ -1 & \dots & -1 & -1 \end{bmatrix},$$

d.h. es gibt eine Permutationsmatrix P sodass

$$PA_1P^T = A_2.$$

Beispiel

Eine LR-Zerlegung der Matrix

$$A_2 := \begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ -1 & \dots & -1 & -1 \end{bmatrix},$$

ist gegeben durch

Beispiel

Eine LR-Zerlegung der Matrix

$$A_2 := \begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ -1 & \dots & -1 & -1 \end{bmatrix},$$

ist gegeben durch

$$L = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ -1 & \cdots & -1 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ & & & -n \end{bmatrix}$$

Fill-in-Rate $:= \frac{\mathrm{nnz}(LR)}{\mathrm{nnz}(A)} = \frac{n+2\cdot(n-1)}{n+2\cdot(n-1)} = 1$, d.h. kein Fill-in.

Idee

Bestimme Fill-in reduzierende Permutationen, d.h. zu gegebenem $A \in \mathbb{R}^{n,n}$ berechne Permutationen $P, Q \in \mathbb{R}^{n,n}$, sodass die LR-Zerlegung von

$$PAQ$$

wenig Fill-in hat.

Vereinfachung

Im folgenden sei $A = A^T$ als symmetrisch angenommen und wir suchen Permutation P der Form

$$PAP^T$$

Der Graph von A

Sei $A = A^T = [a_{ij}] \in \mathbb{R}^{n,n}$ eine (sparse) Matrix. Dann heißt der ungerichtete Graph

$$G := (V, E)$$

mit Knotenmenge

$$V := \{1, \ldots, n\}$$

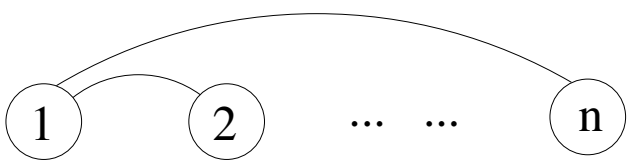
und Kantenmenge

$$E := \{\{i, j\} \mid a_{ij} \neq 0\},$$

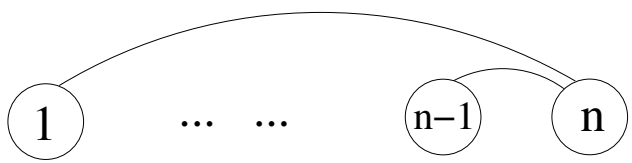
der *Graph von A*.

Beispiel

$$\begin{bmatrix} -1 & -1 & \cdots & -1 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ -1 & \cdots & -1 & -1 \end{bmatrix}$$



Eliminationsschritt

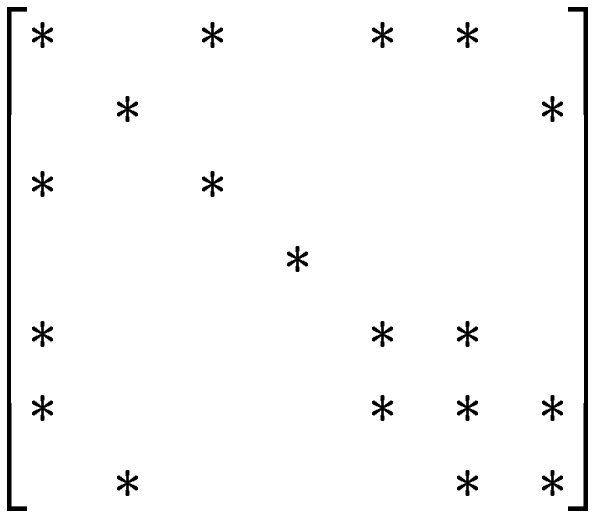
Der erste Schritt der LR-Zerlegung einer Matrix der Form

$$\begin{bmatrix} * & & * & & * & * & \\ & * & & & & & * \\ * & & * & & & & \\ & & & * & & & \\ * & & & & * & * & \\ * & & & & * & * & * \\ & * & & & & * & * \end{bmatrix}$$

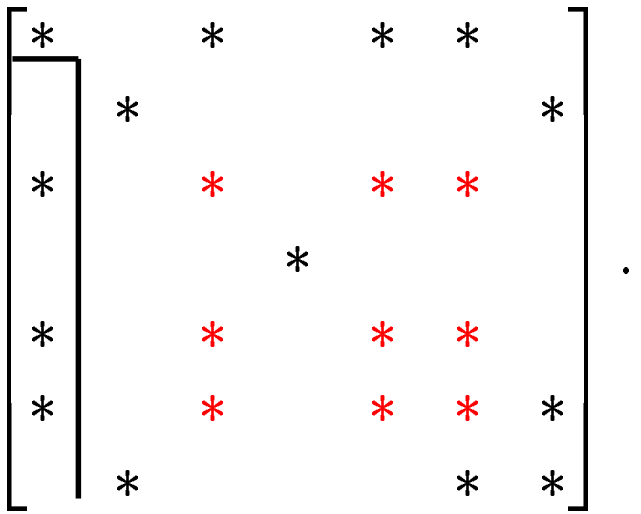
führt im schlimmsten Fall auf eine Matrix der Form

Eliminationsschritt

Der erste Schritt der LR-Zerlegung einer Matrix der Form

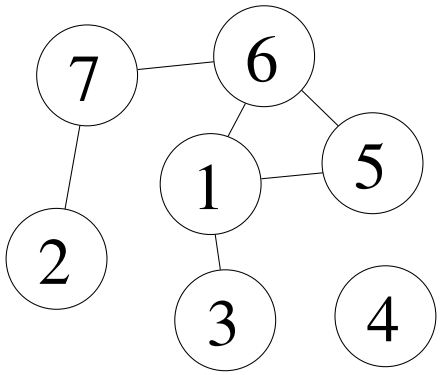


führt im schlimmsten Fall auf eine Matrix der Form



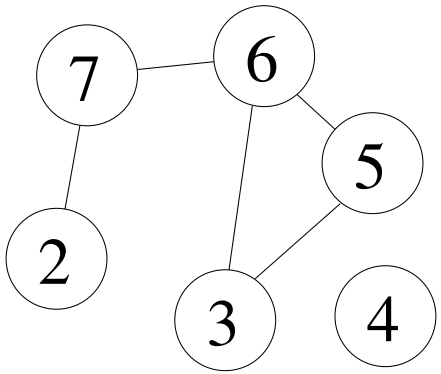
Eliminationsschritt

Der entsprechende Graph



$$\begin{bmatrix} * & & * & & * & * & \\ & * & & & & & * \\ * & & * & & & & \\ & & & * & & & \\ * & & & & * & * & \\ * & & & & * & * & * \\ & * & & & & * & * \end{bmatrix}$$

wird im schlimmsten Fall zu



$$\begin{bmatrix} * & & * & & * & * & \\ & * & & & & & * \\ * & & * & & * & * & \\ & & & * & & & \\ * & & * & & * & * & \\ * & & * & & * & * & * \\ & * & & & & * & * \end{bmatrix}.$$

Minimum-Degree-Algorithmus

Für den Graph $G = (V, E)$ und einen ausgezeichneten Knoten $v \in V$ nennt man die Anzahl seiner Nachbarn

$$\deg v := |\{j \in V \setminus \{v\} \mid \{v, j\} \in E\}|,$$

den *Grad* von v.

Im *Minimum-Degree-Algorithmus* wählt man immer den Knoten, der den kleinsten Grad hat, zum Pivotelement.

Approximate-Minimum-Degree

Schneller ist der Algorithmus

AMD,

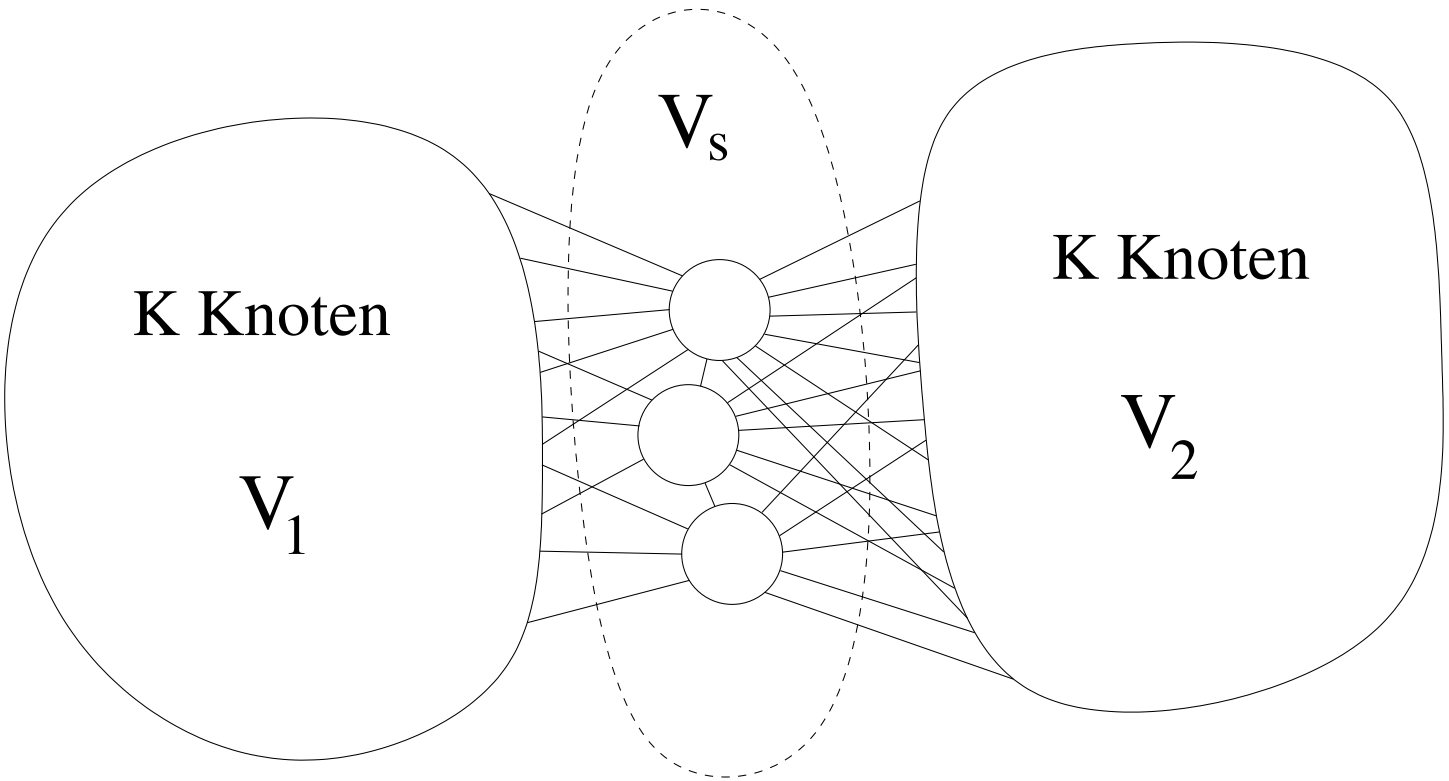
in dem eine Approximation der Minimum Degree Permutation berechnet. In Matlab:

```
>> pi = symamd(A);
```

"An approximate minimum degree ordering algorithm". P. Amestoy, T. A. Davis, and I. S. Duff, SIAM Journal on Matrix Analysis and Applications, vol 17, no. 4, pp. 886-905, Dec. 1996

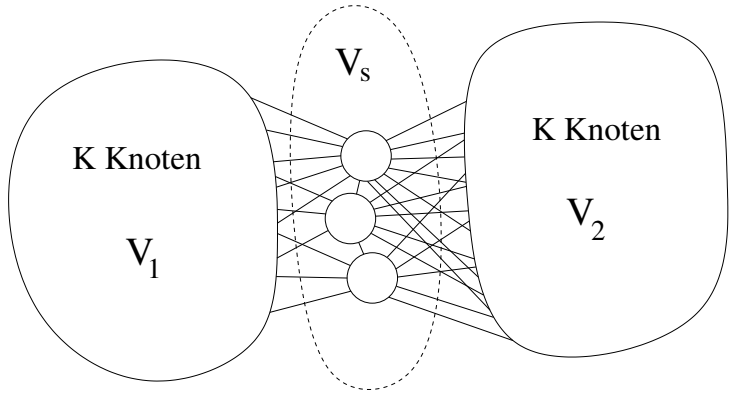
Nested-Dissection

Man sucht für den Graphen einer Matrix $A = A^T \in \mathbb{R}^{n,n}$ einen kleinen *Vertex Separator*, d.h. eine kleine Menge von Knoten V_s, welche den Graphen in zwei (etwa) gleichgroße Teilgraphen V_1, V_2 teilt:



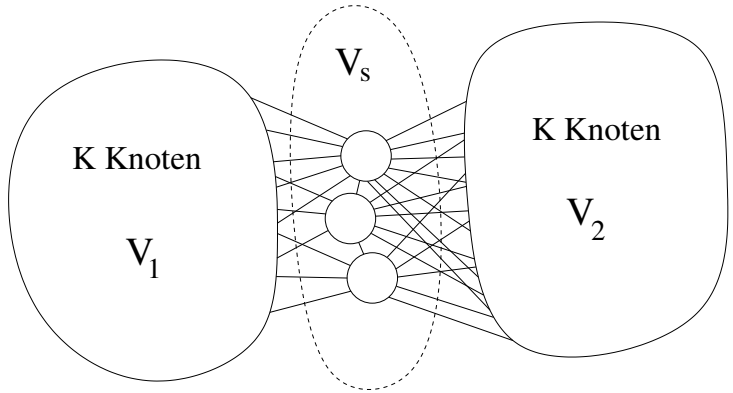
mit $|V_s| \ll |V_1| \approx |V_2|$.

Sortiert man dann die Knoten in V_1 nach vorne, die in V_s nach hinten und die in V_2 dazwischen, dann hat die Matrix zu

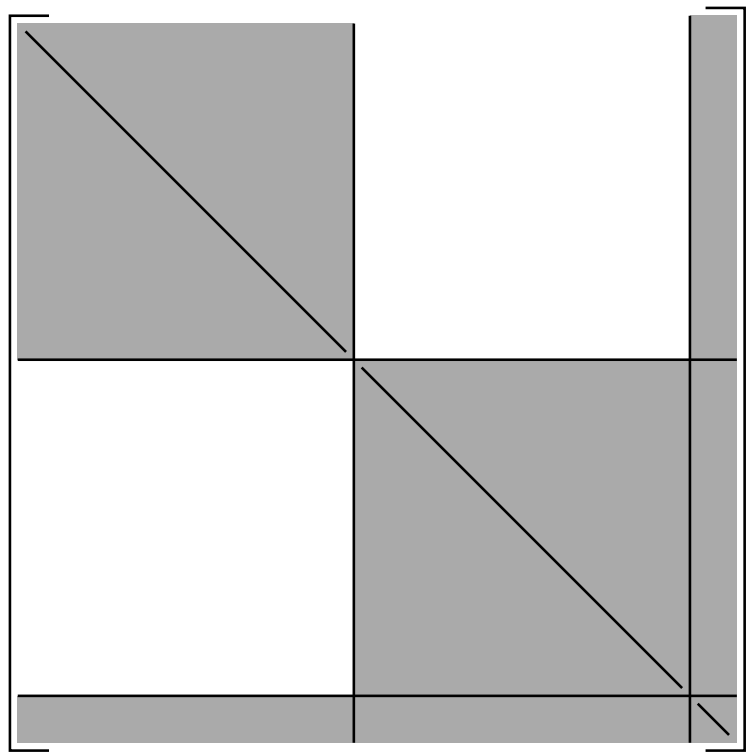


die Form

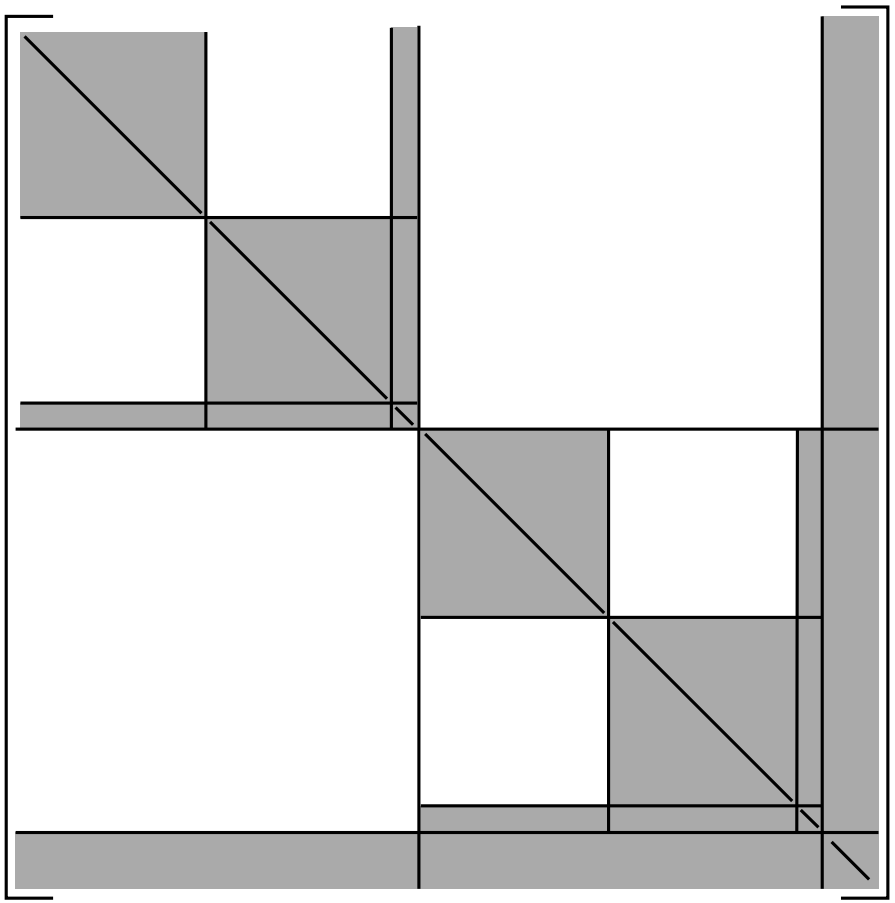
Sortiert man dann die Knoten in V_1 nach vorne, die in V_s nach hinten und die in V_2 dazwischen, dann hat die Matrix zu



die Form



Nested-Dissection



Vorteil gegen AMD: Besser bei mehreren Prozessoren

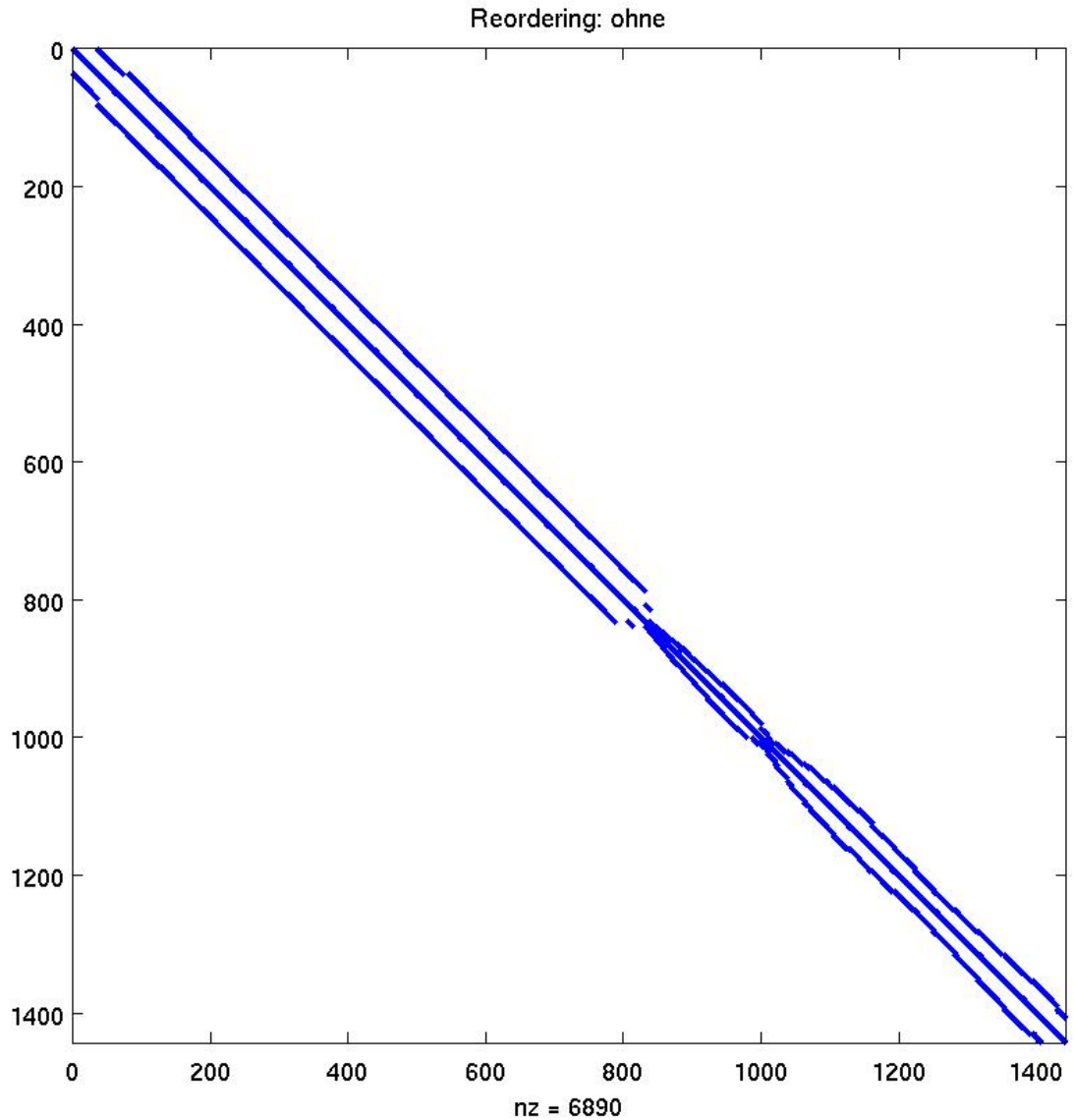
Nested-Dissection

Gebräuchlicher C-Code für Nested-Dissection:

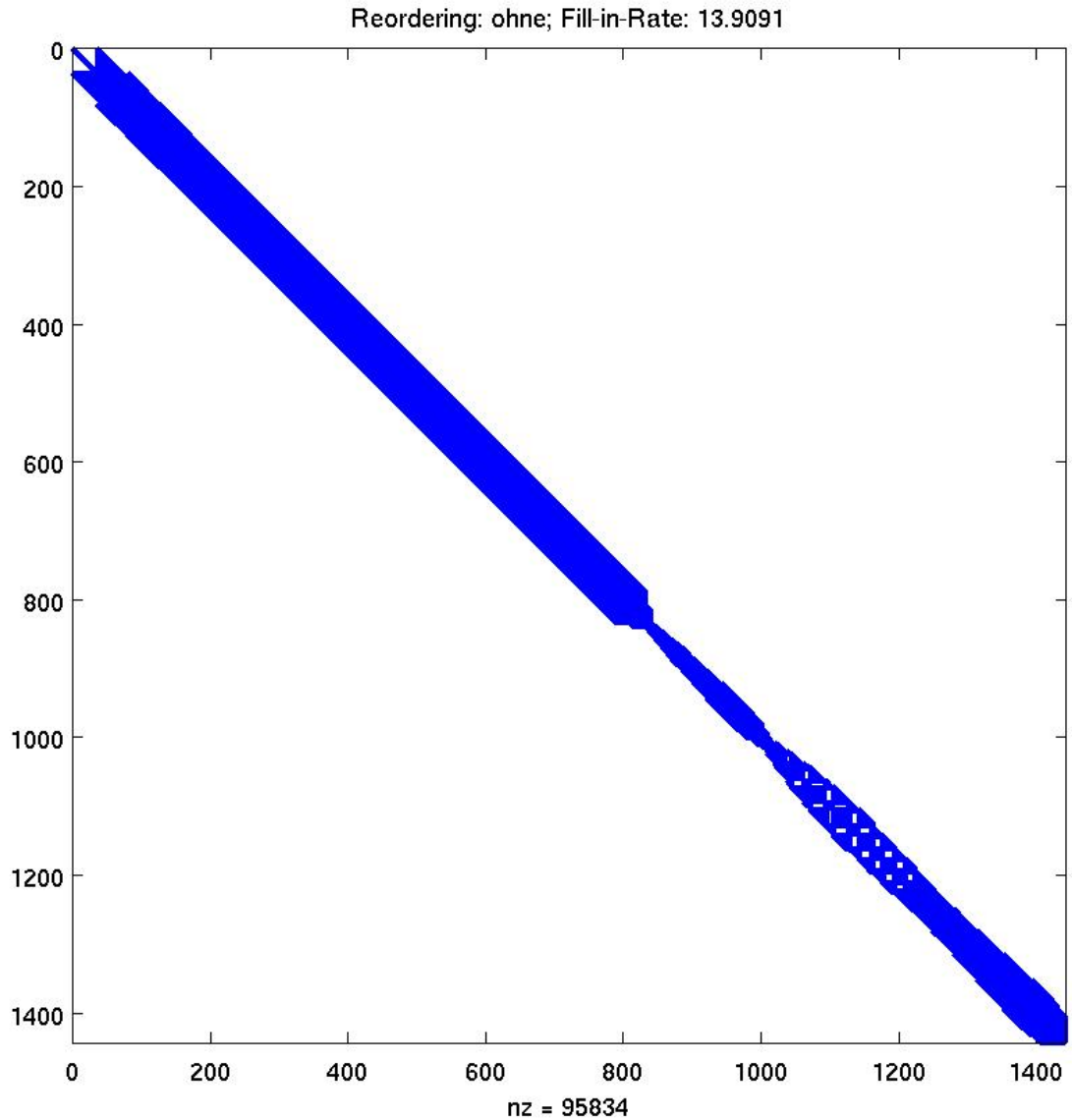
METIS

A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. George Karypis and Vipin Kumar. SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359-392, 1999

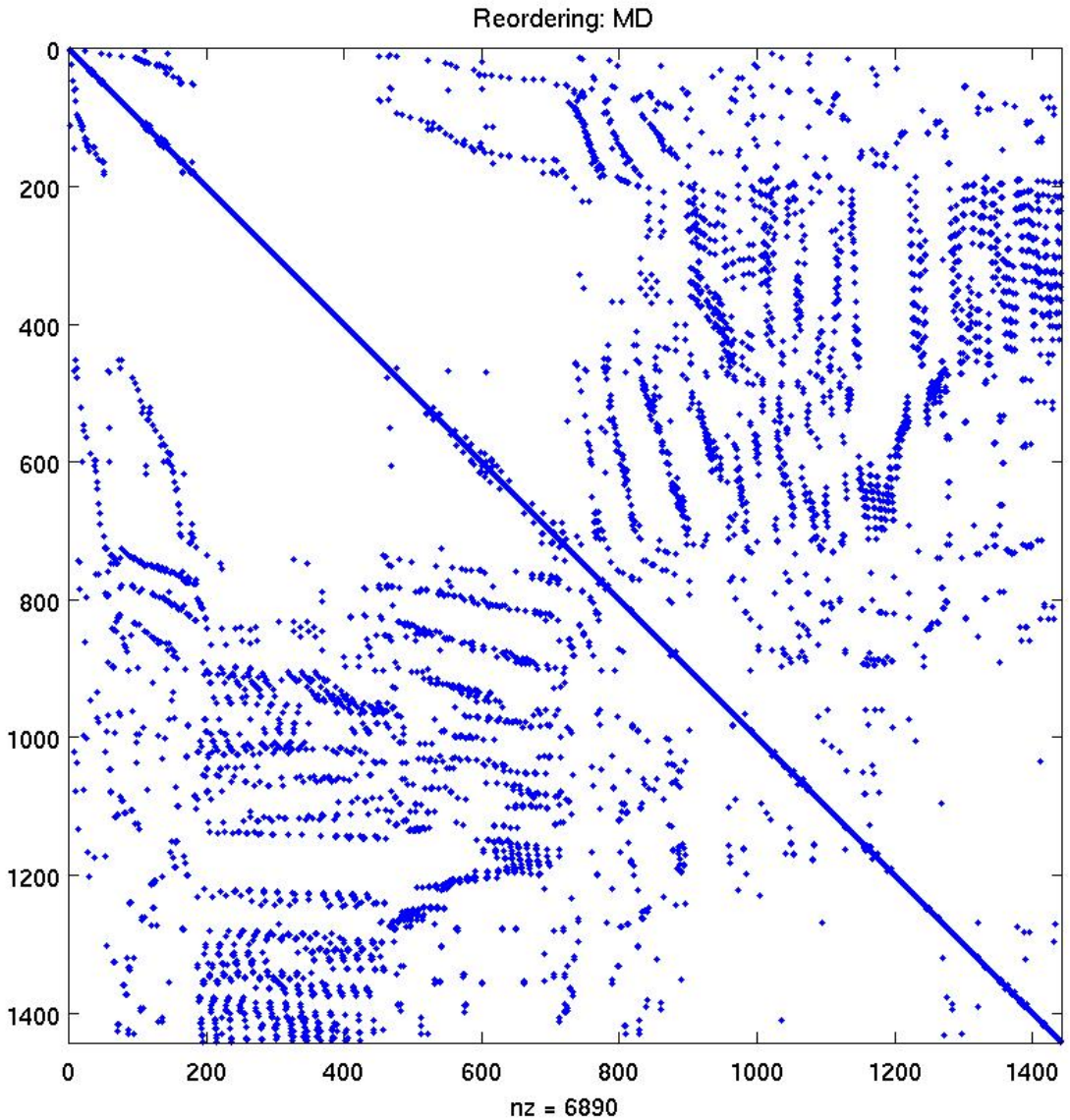
poisson_2d A ohne



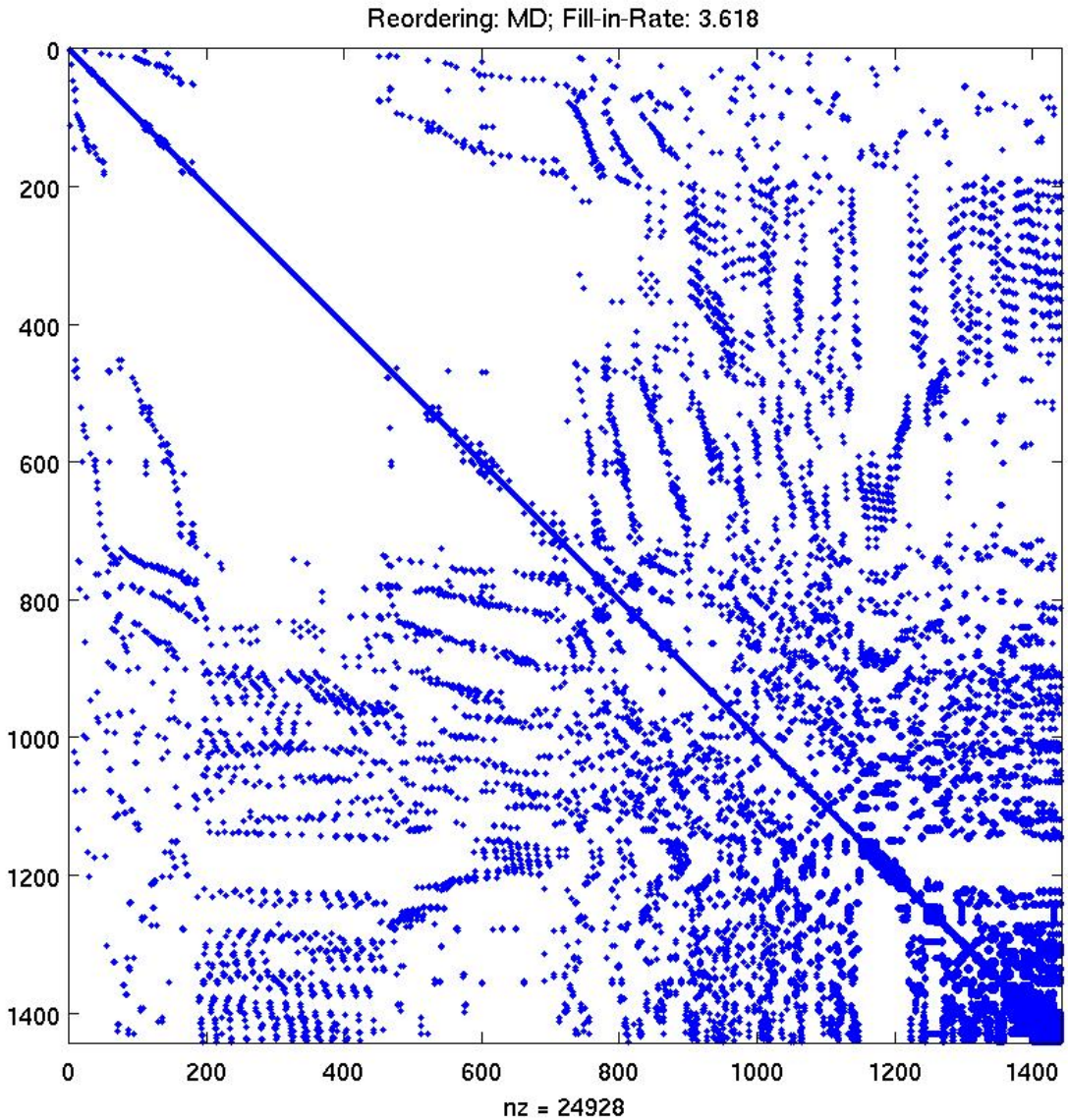
poisson_2d LR ohne



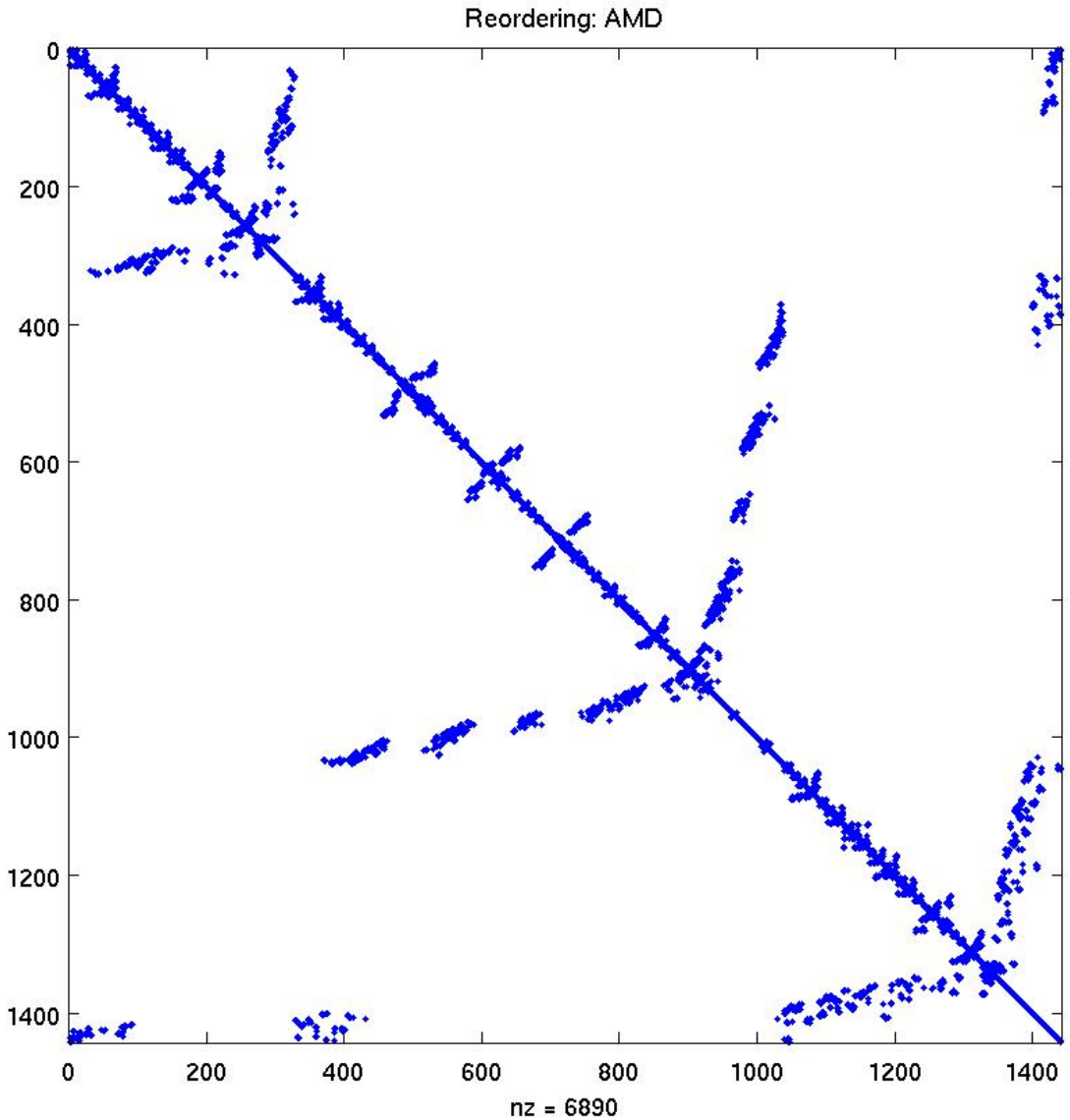
poisson_2d A MD



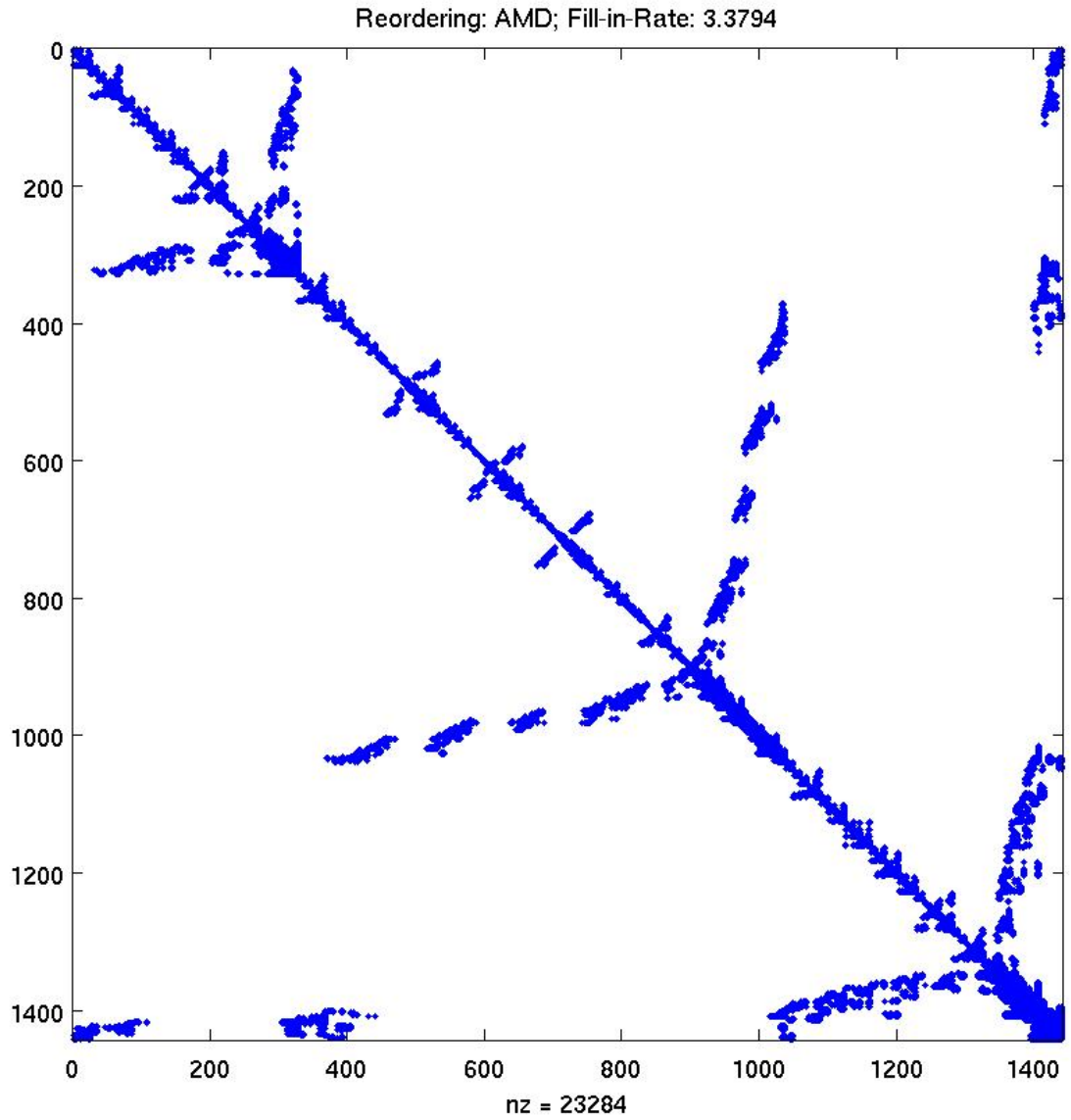
poisson_2d LR MD



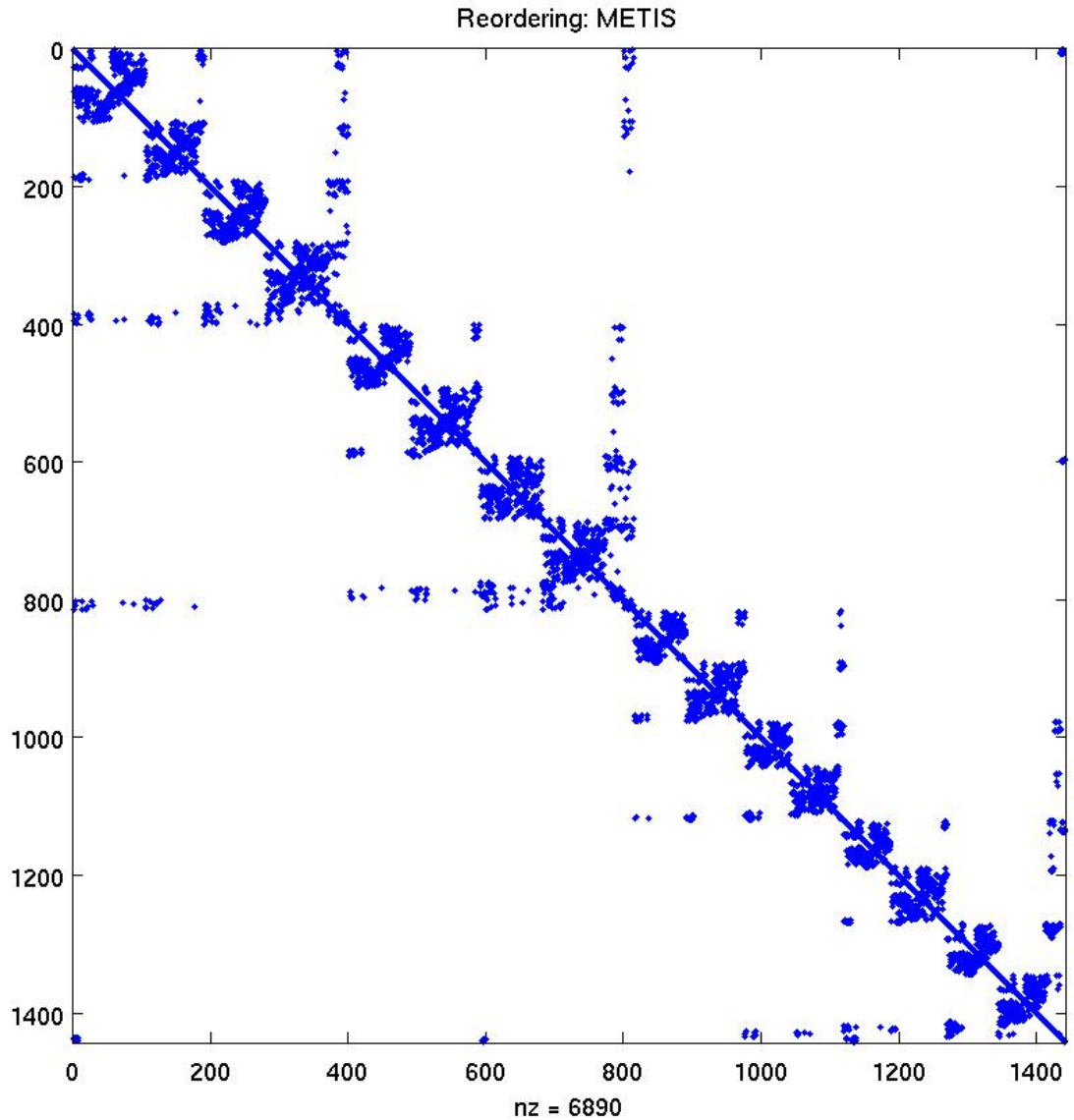
poisson_2d A AMD



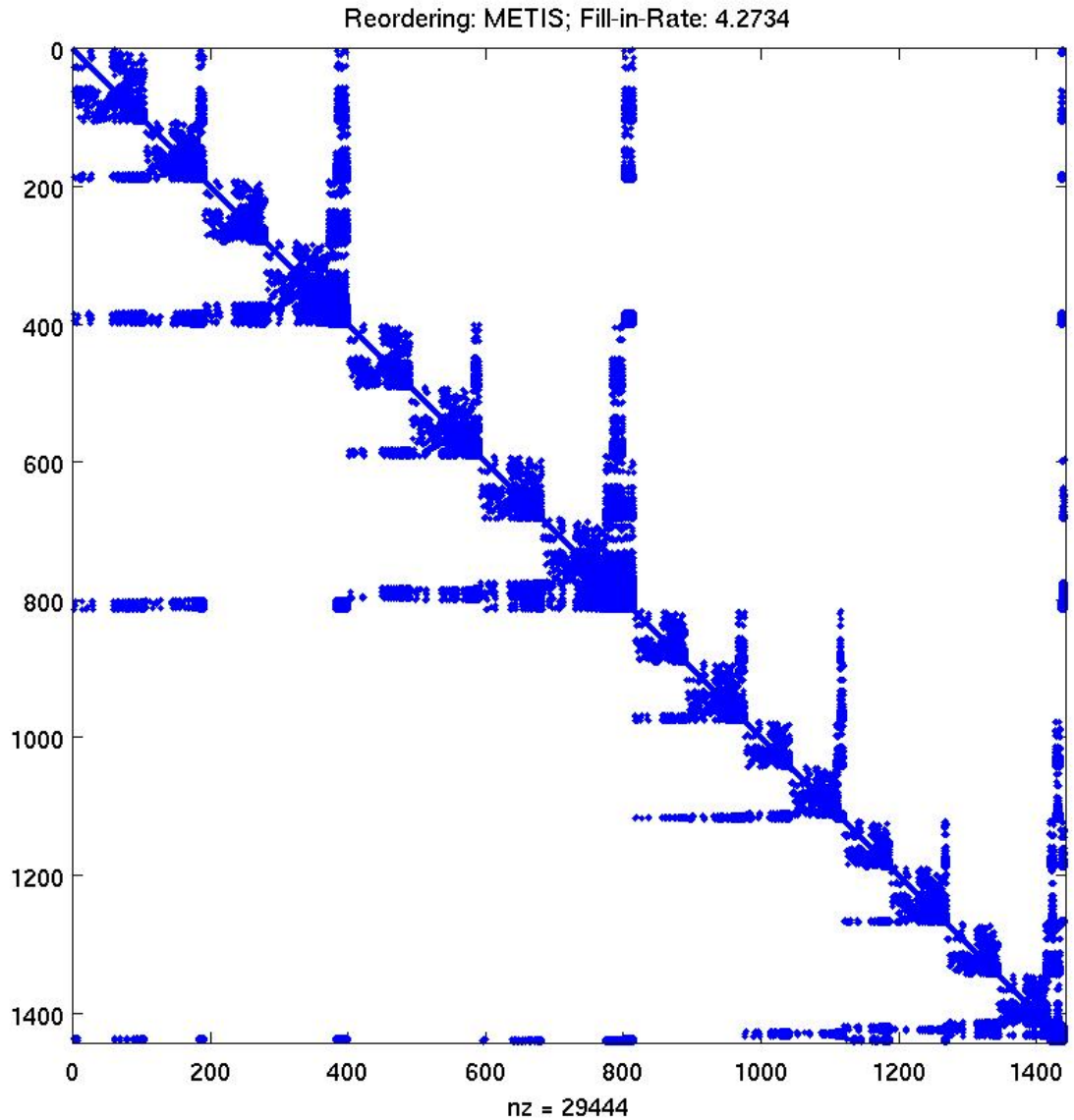
poisson_2d LR AMD



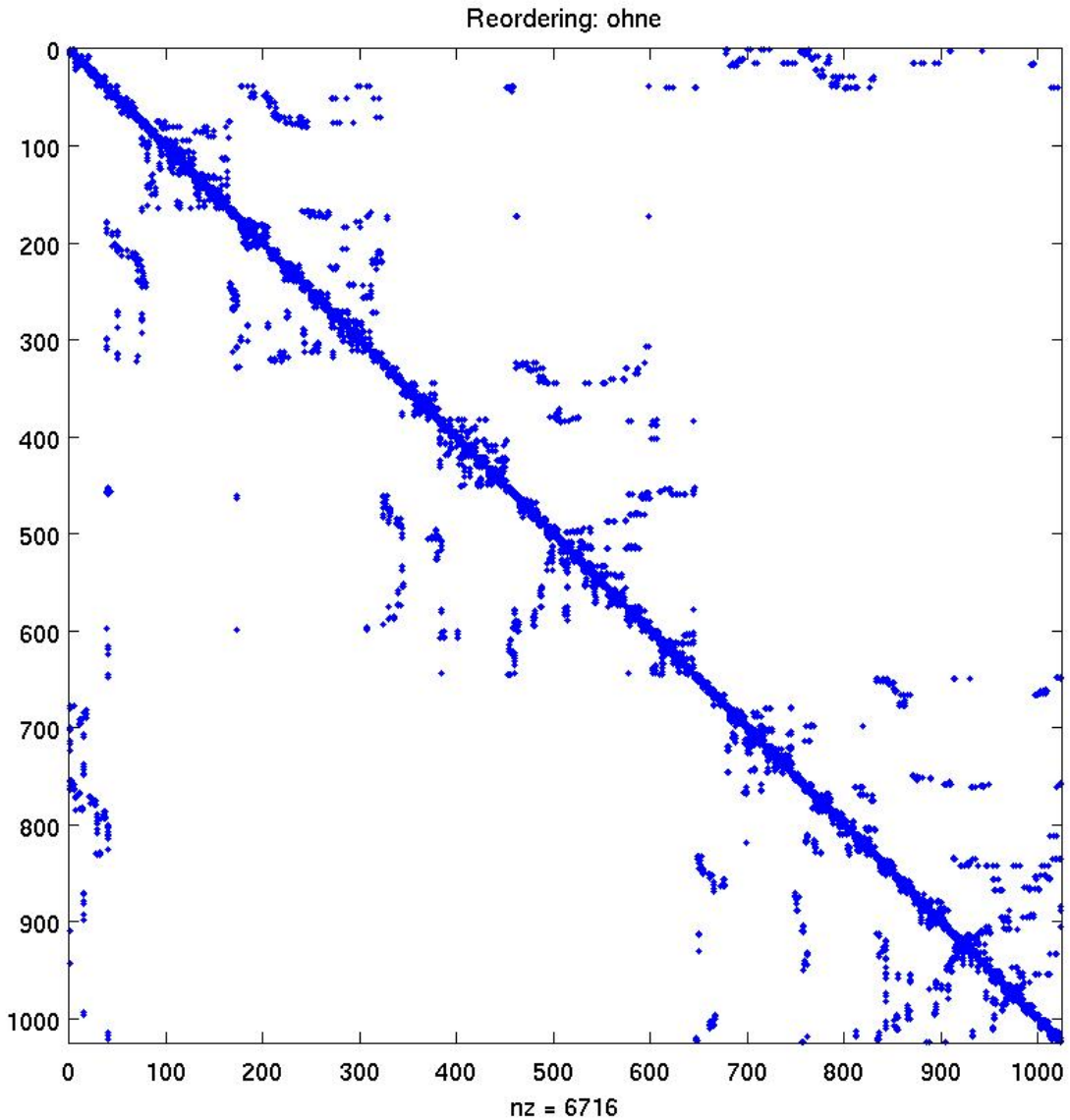
poisson_2d A METIS



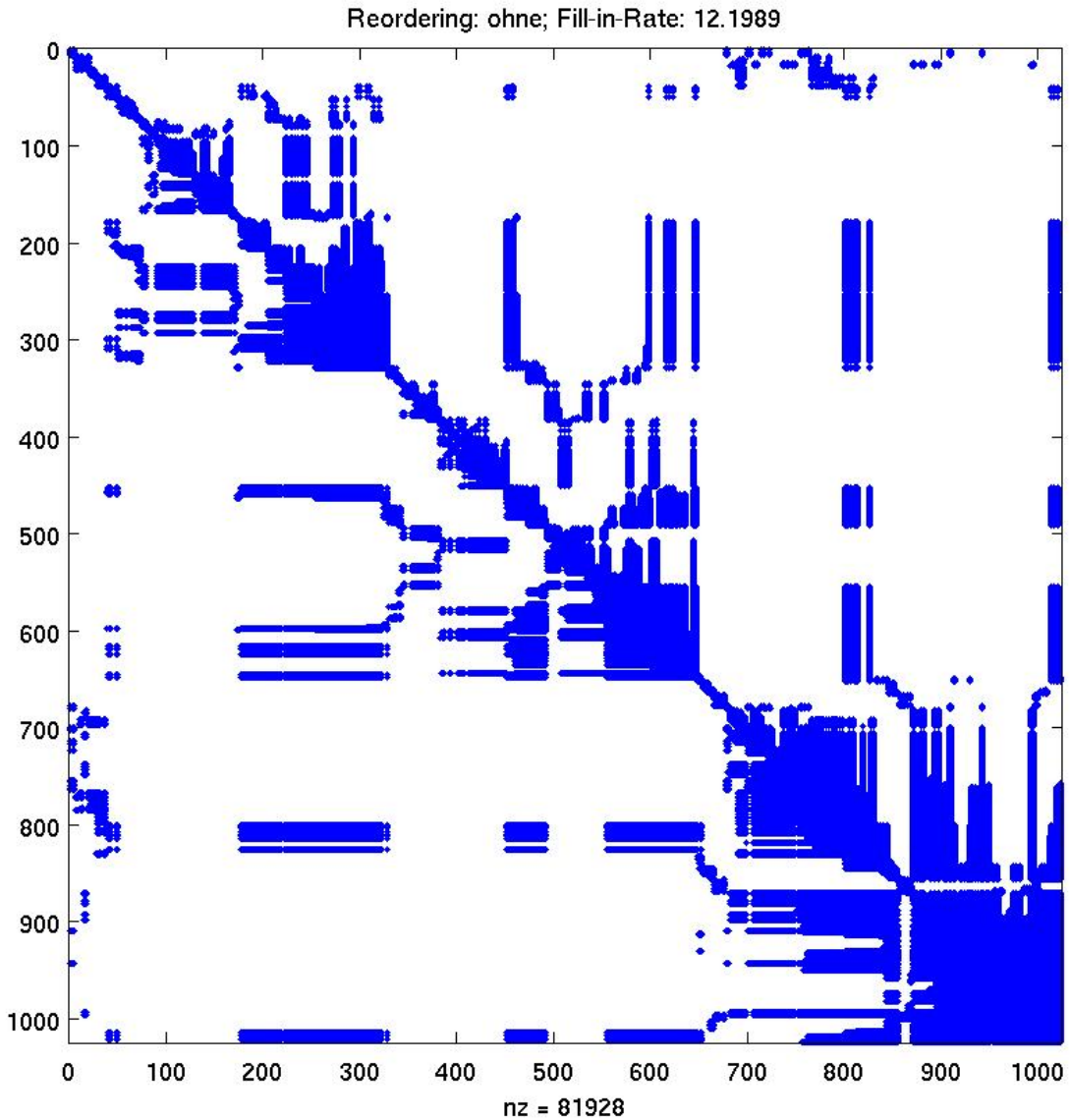
poisson_2d LR METIS

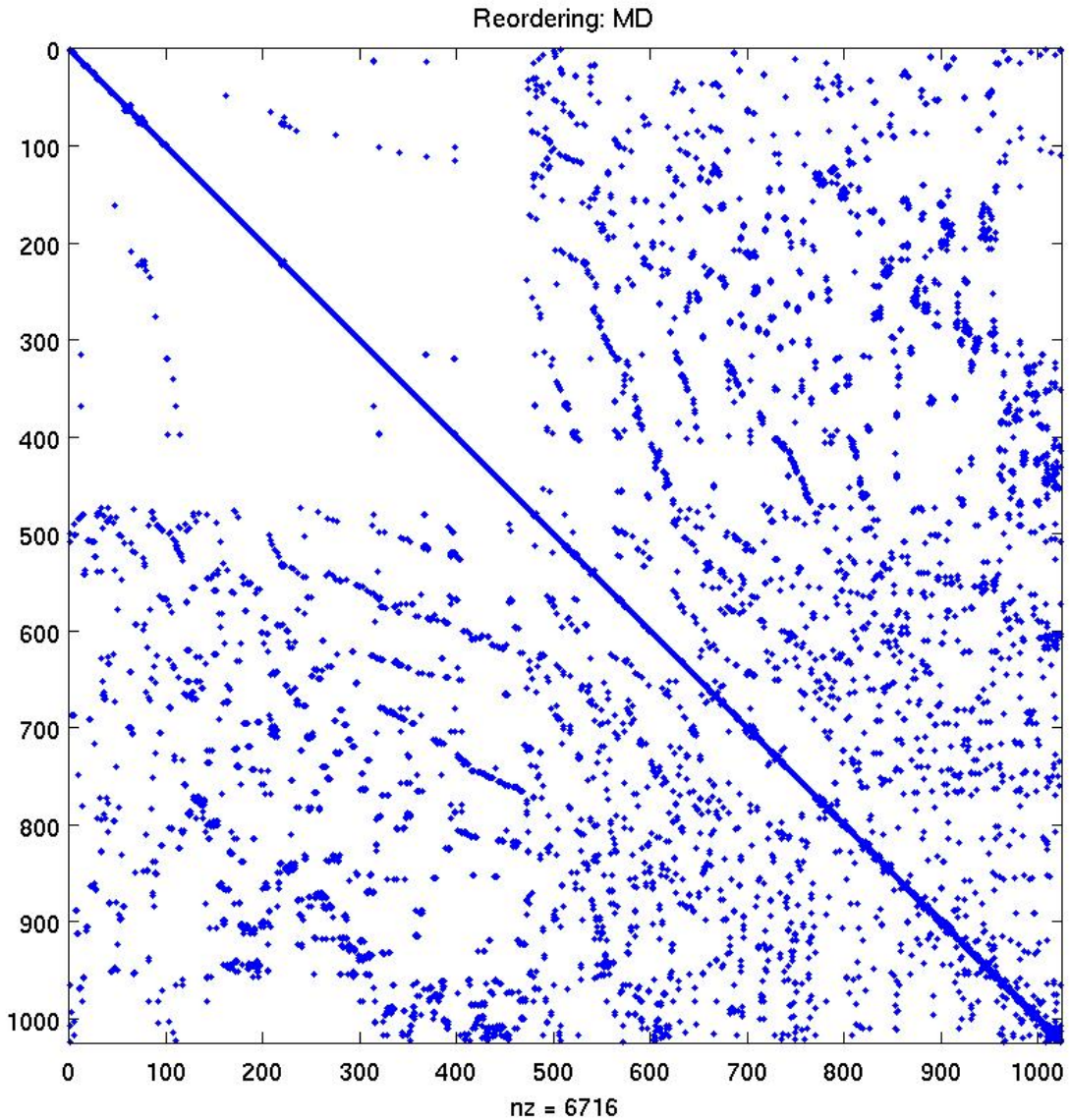


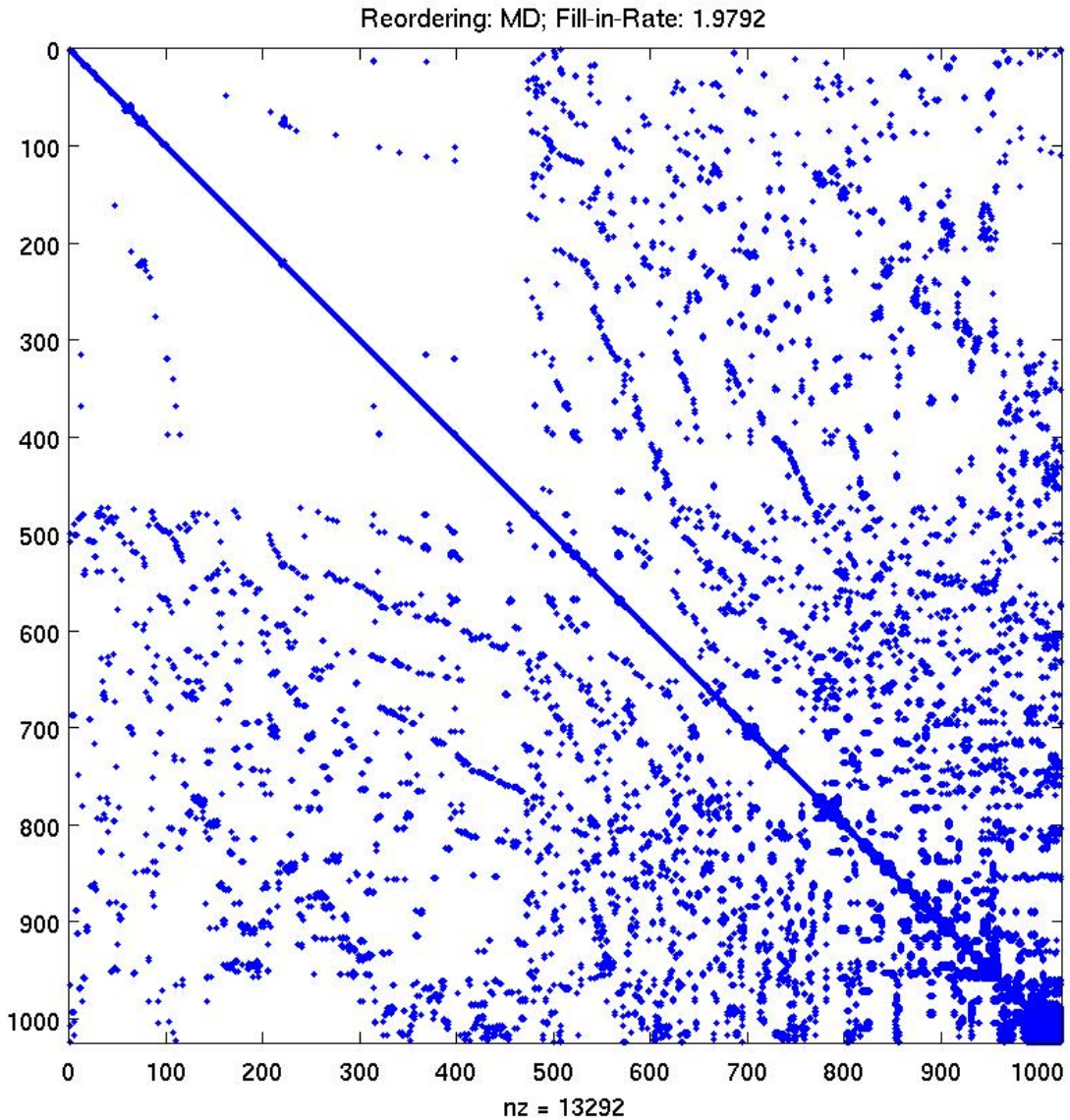
Tapir A ohne


Reordering: ohne
0
100
200
300
400
500
600
700
800
900
1000
nz = 6716

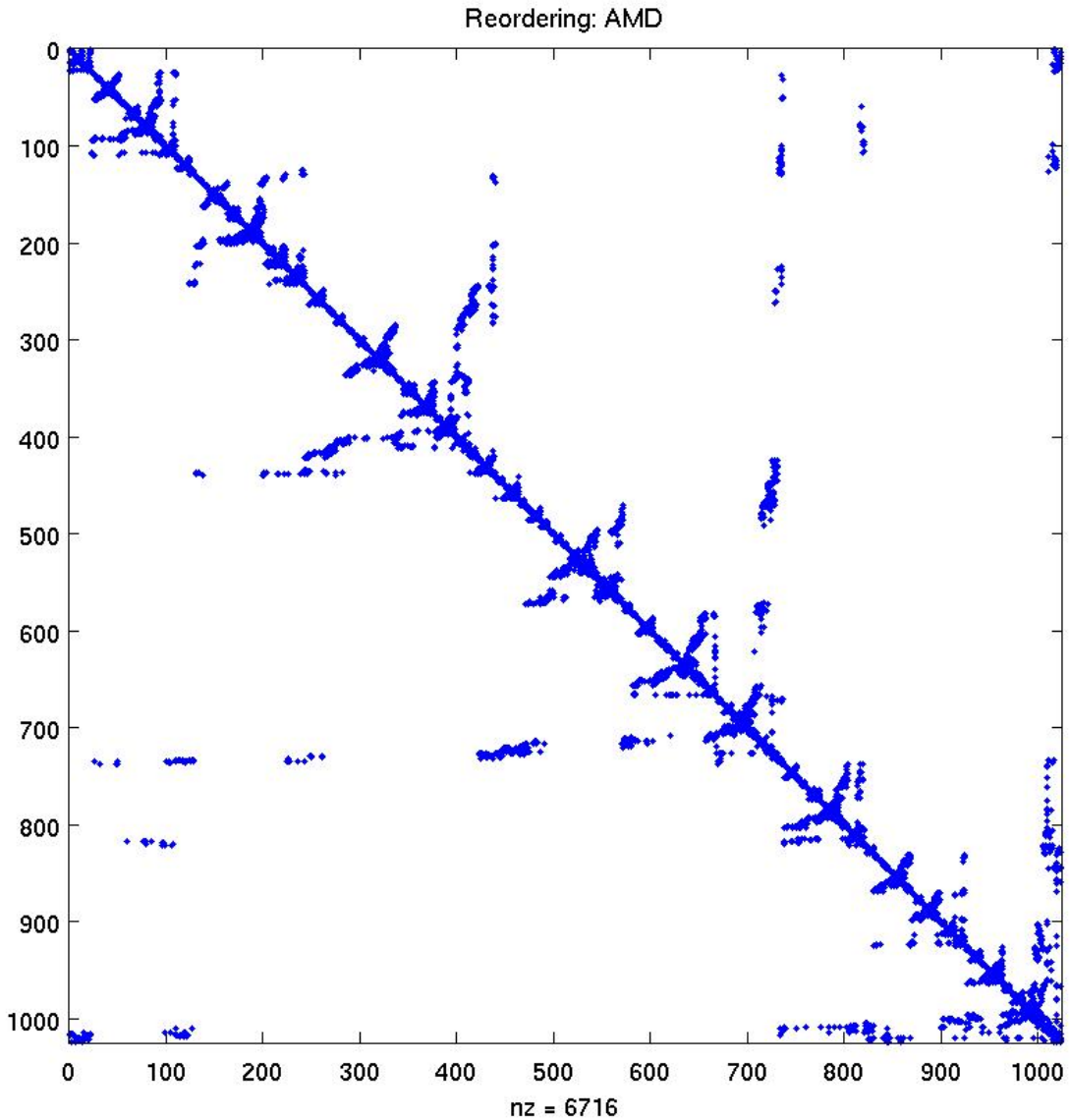
Tapir LR ohne



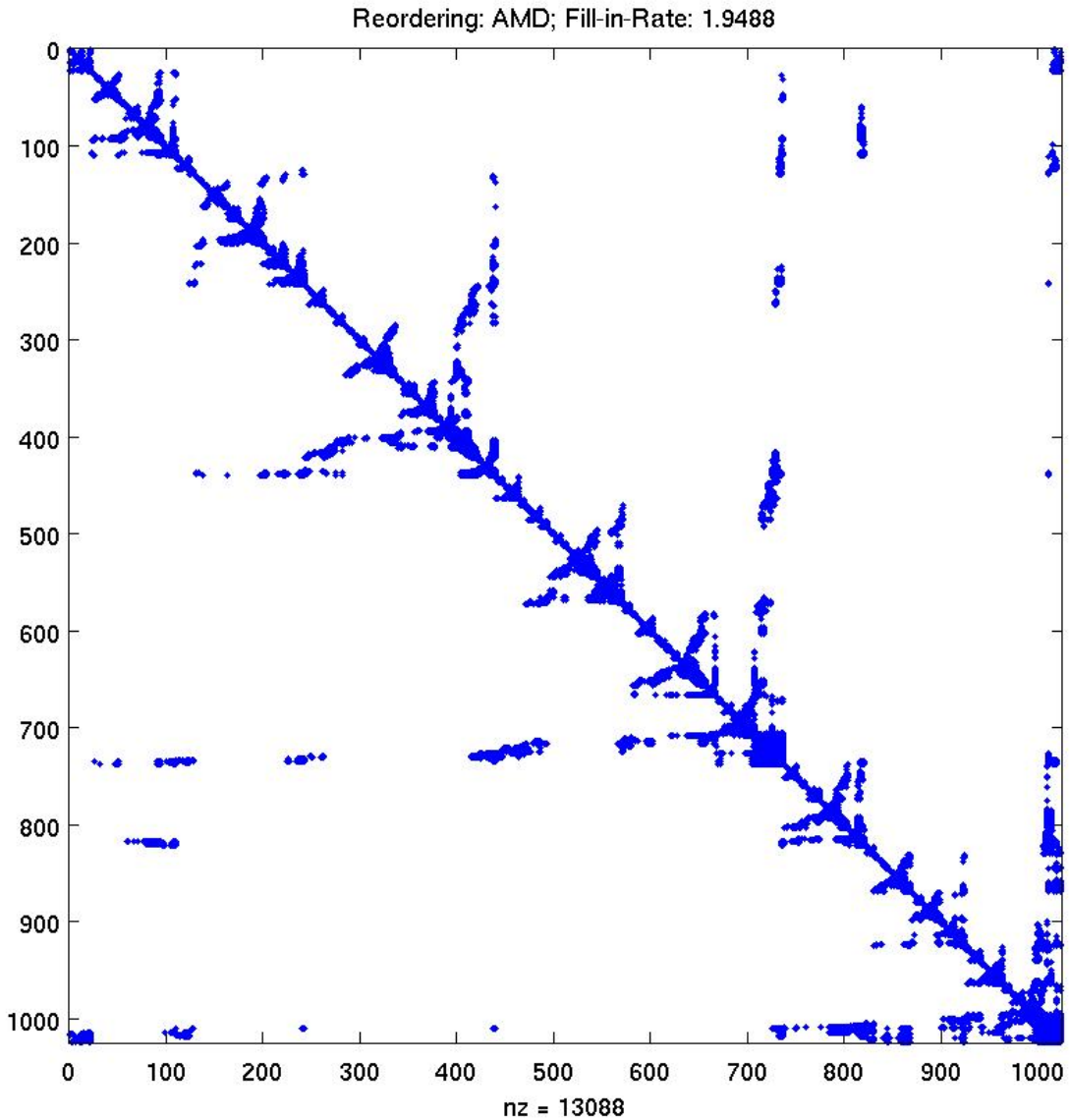

Reordering: MD
0
100
200
300
400
500
600
700
800
900
1000
0
100
200
300
400
500
600
700
800
900
1000
nz = 6716

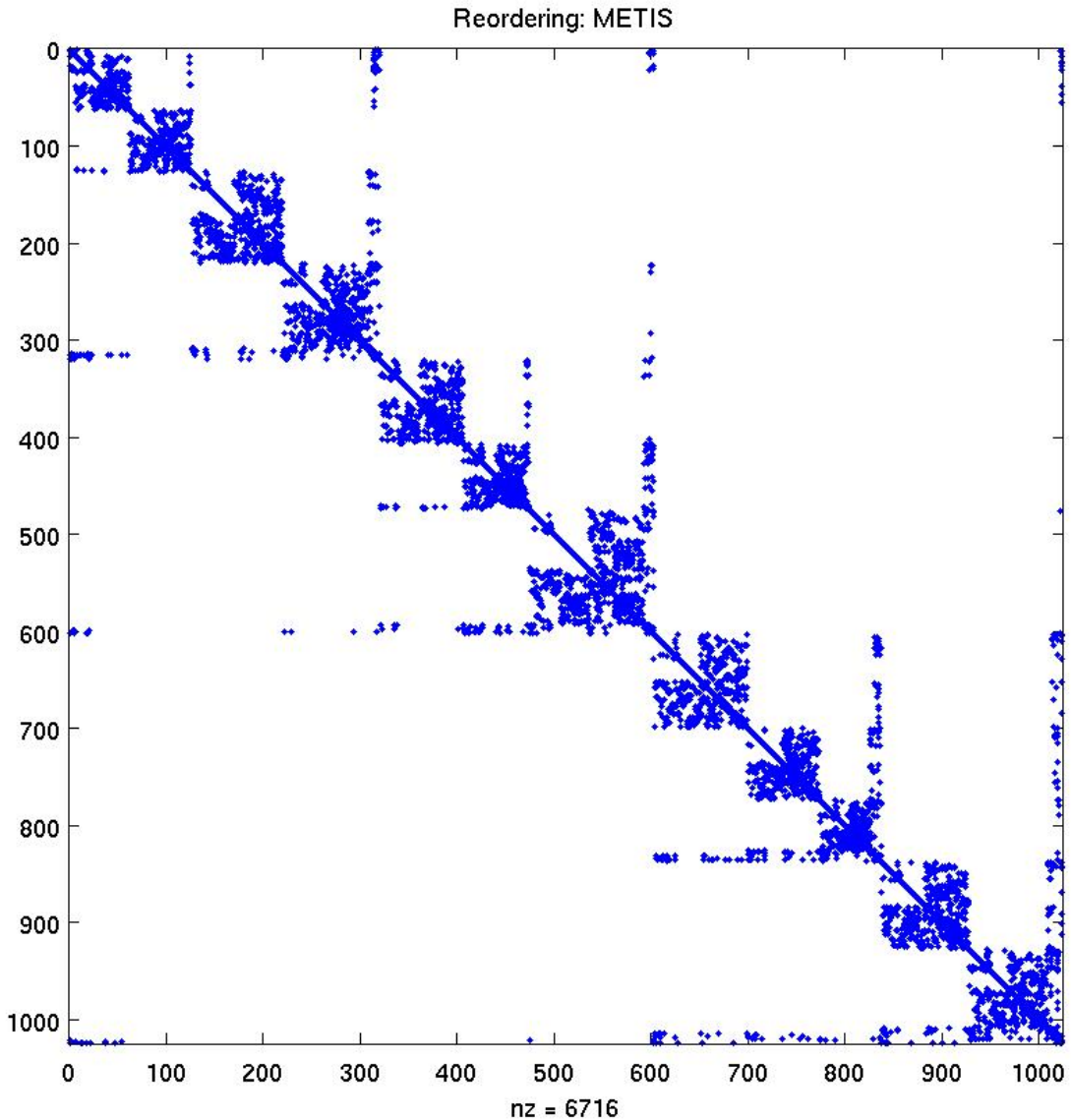

Reordering: MD; Fill-in-Rate: 1.9792
0
100
200
300
400
500
600
700
800
900
1000
0
100
200
300
400
500
600
700
800
900
1000
nz = 13292

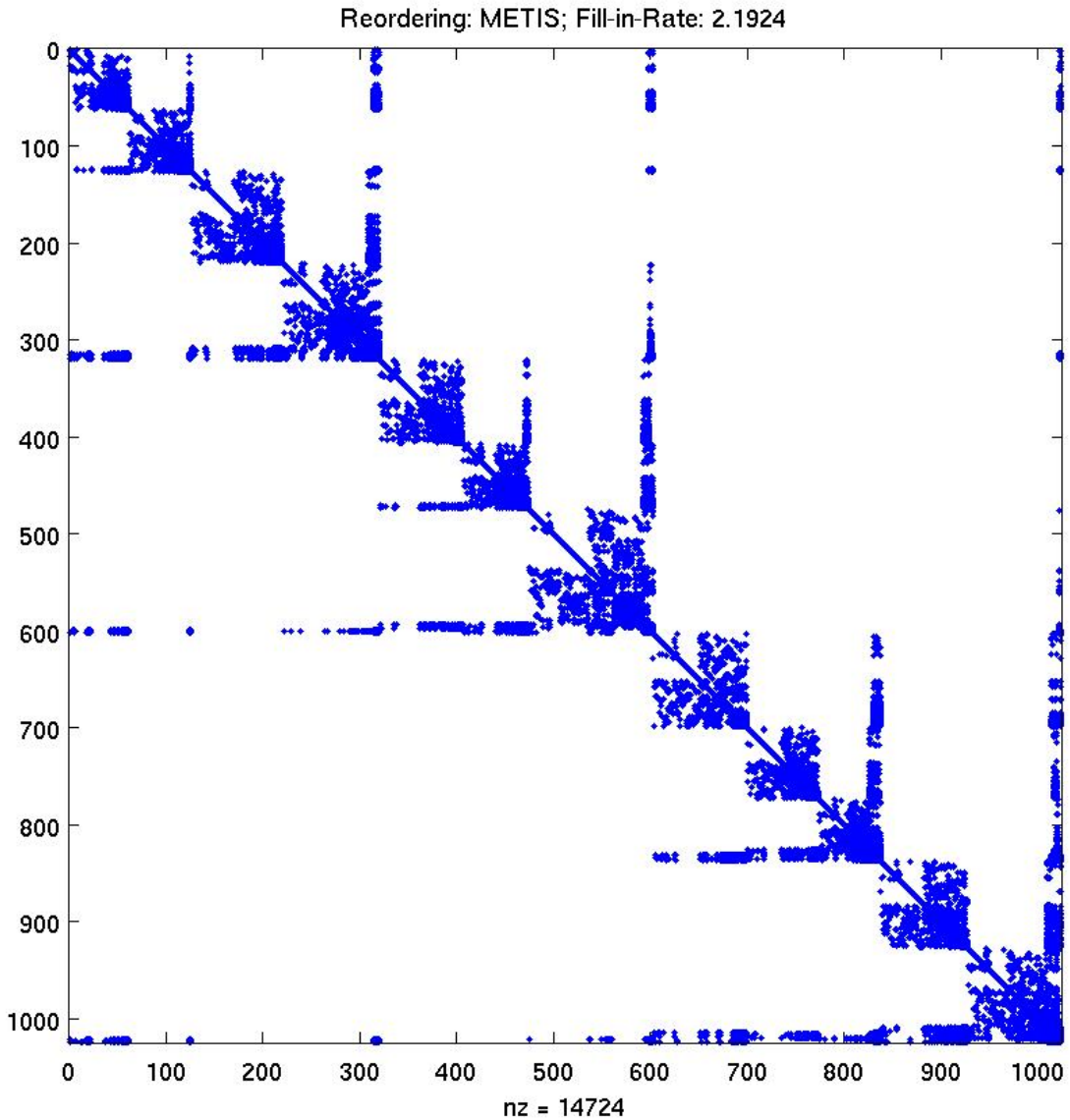
Tapir A AMD


Reordering: AMD
0
100
200
300
400
500
600
700
800
900
1000
0
100
200
300
400
500
600
700
800
900
1000
nz = 6716

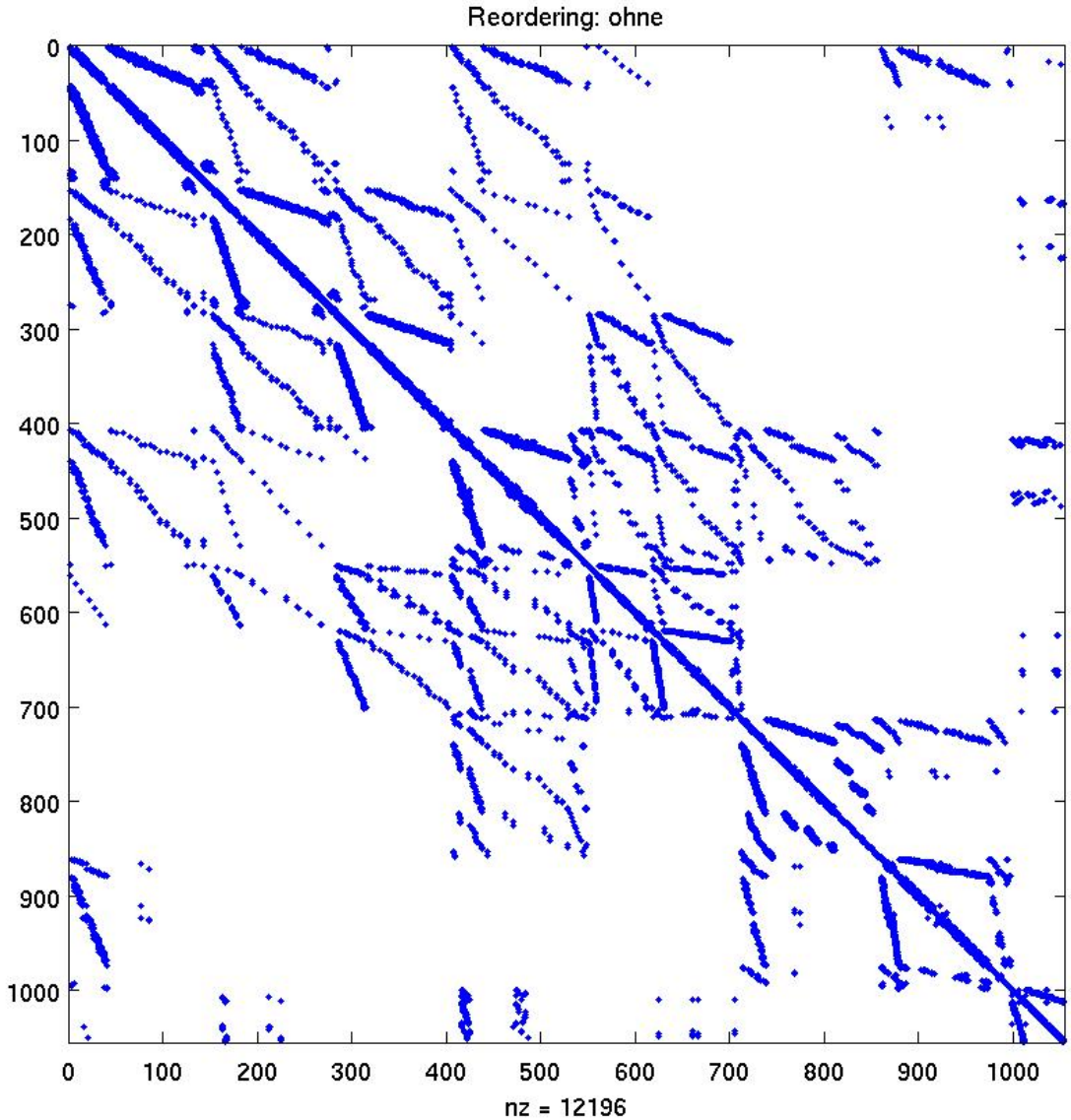
Tapir LR AMD


Reordering: AMD; Fill-in-Rate: 1.9488
0
100
200
300
400
500
600
700
800
900
1000
nz = 13088

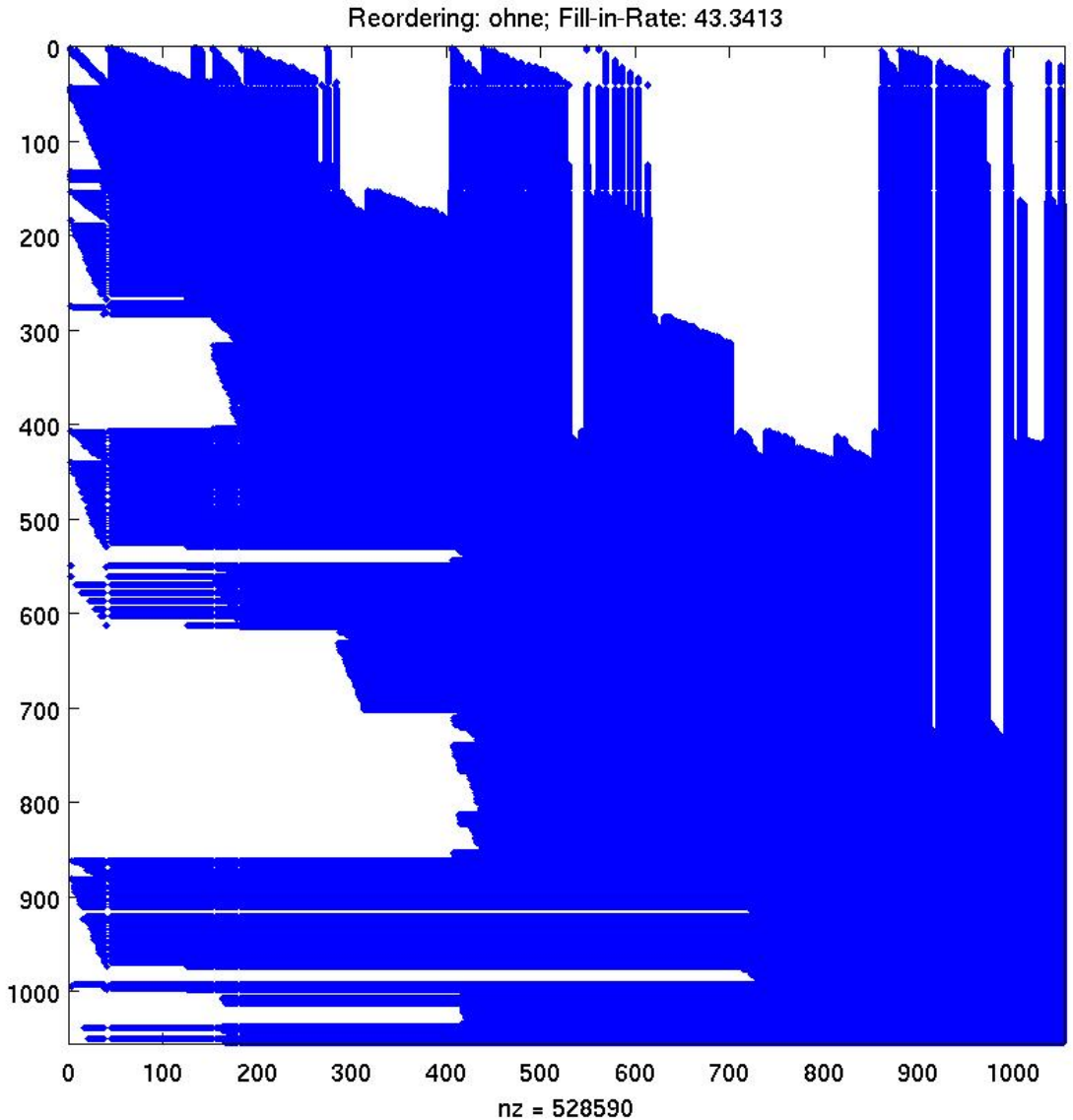

Reordering: METIS
0
100
200
300
400
500
600
700
800
900
1000
0
100
200
300
400
500
600
700
800
900
1000
nz = 6716


Reordering: METIS; Fill-in-Rate: 2.1924
0
100
200
300
400
500
600
700
800
900
1000
nz = 14724

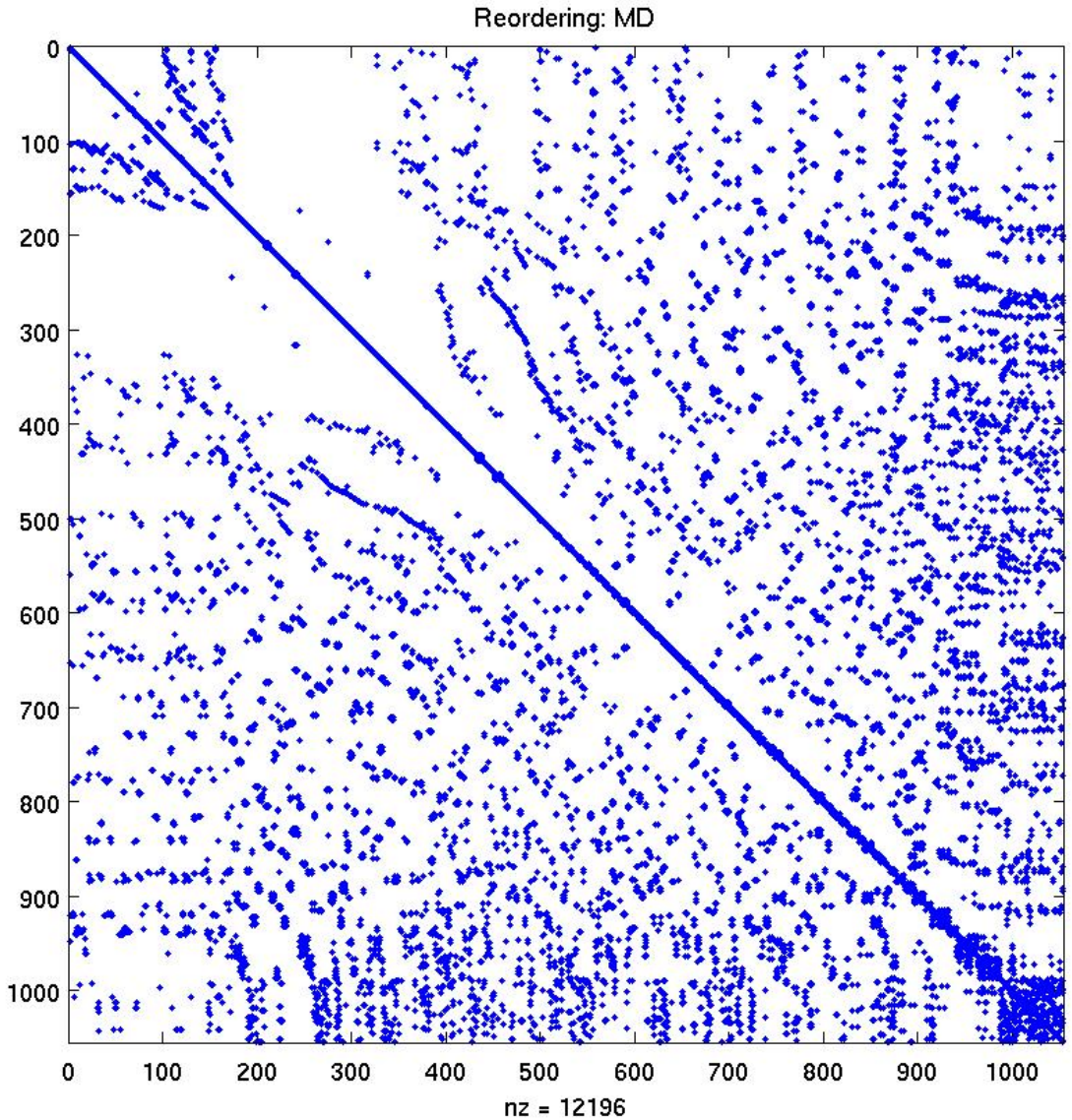
design A ohne

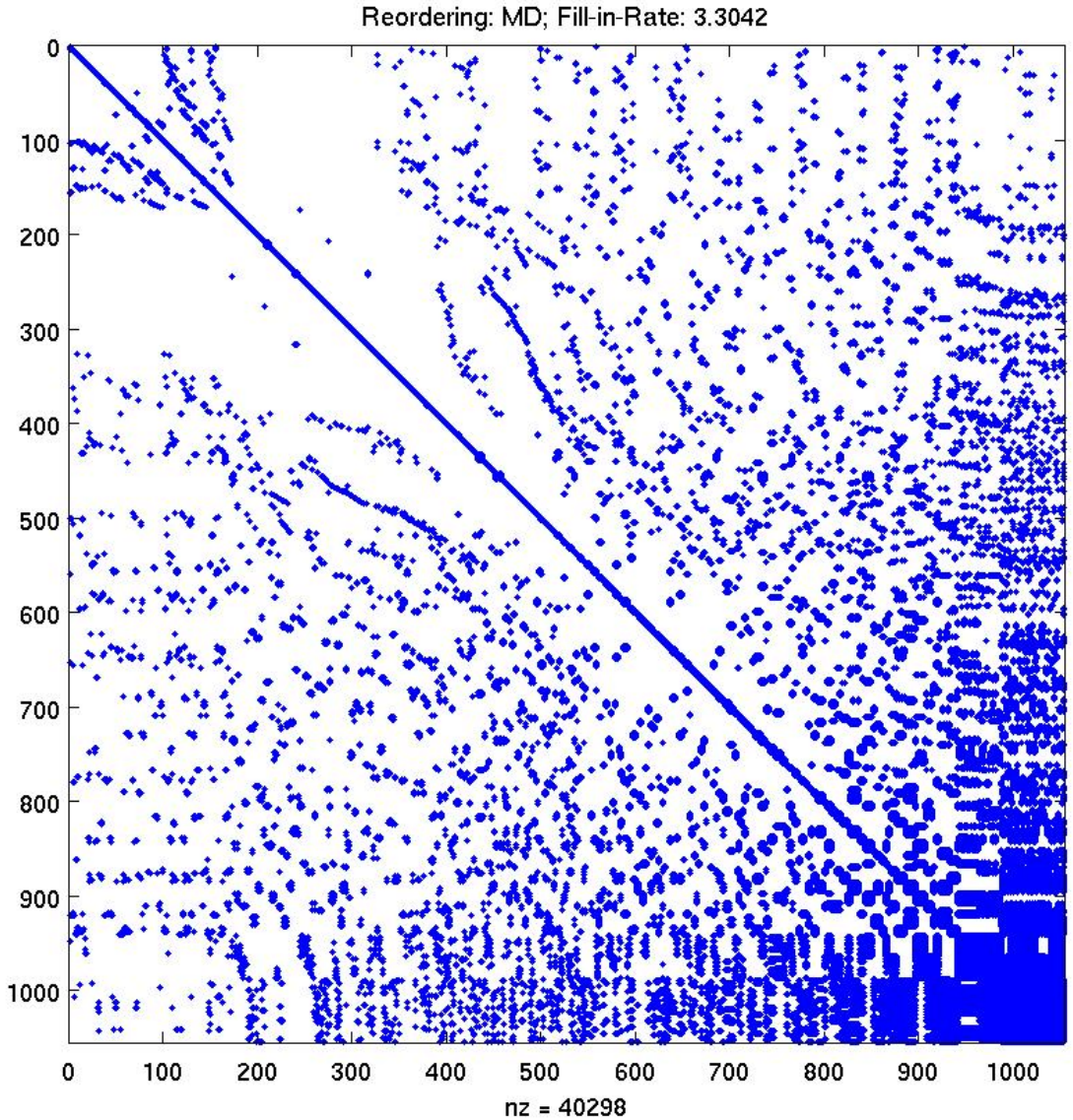


design LR ohne

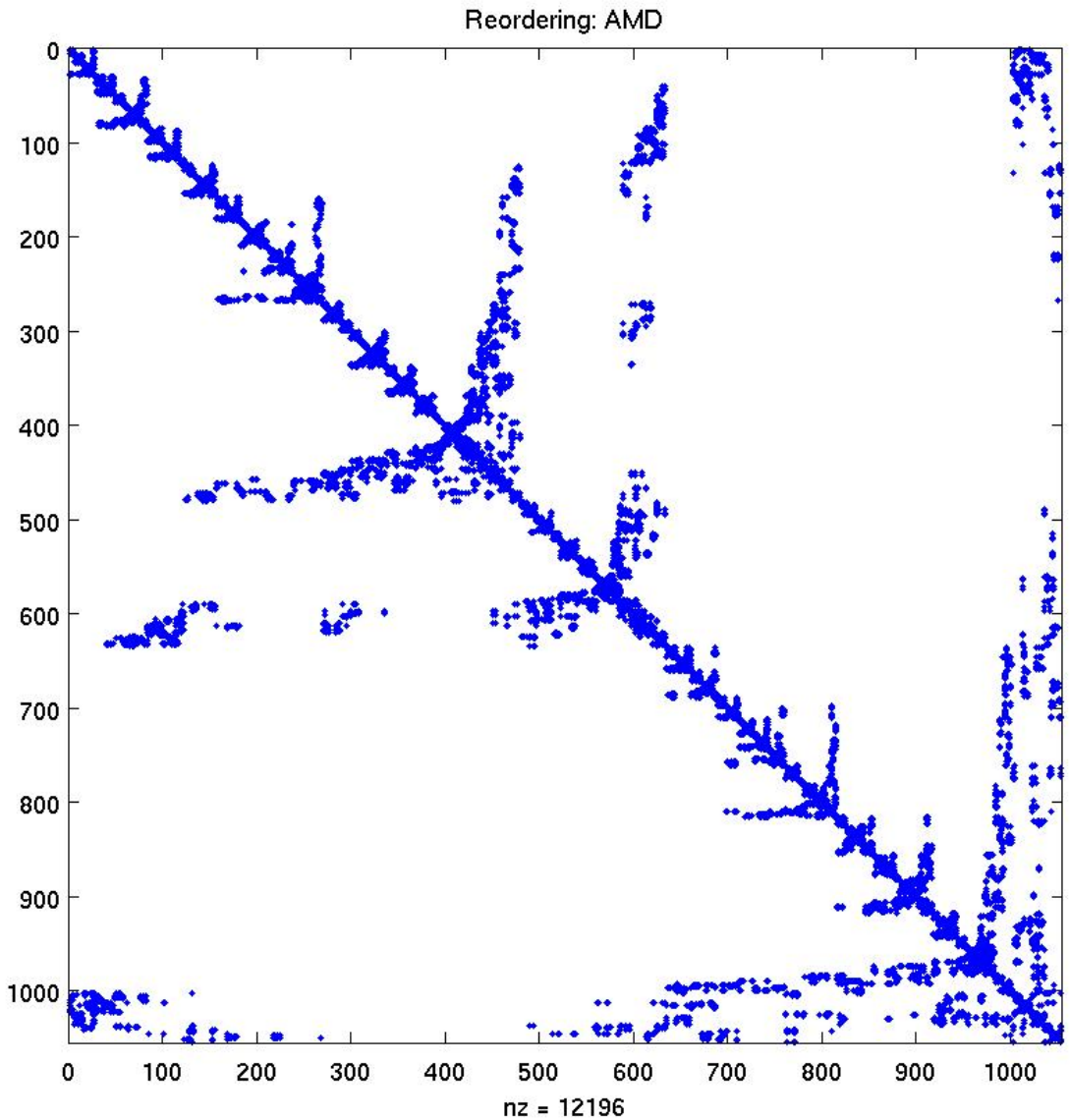


design A MD

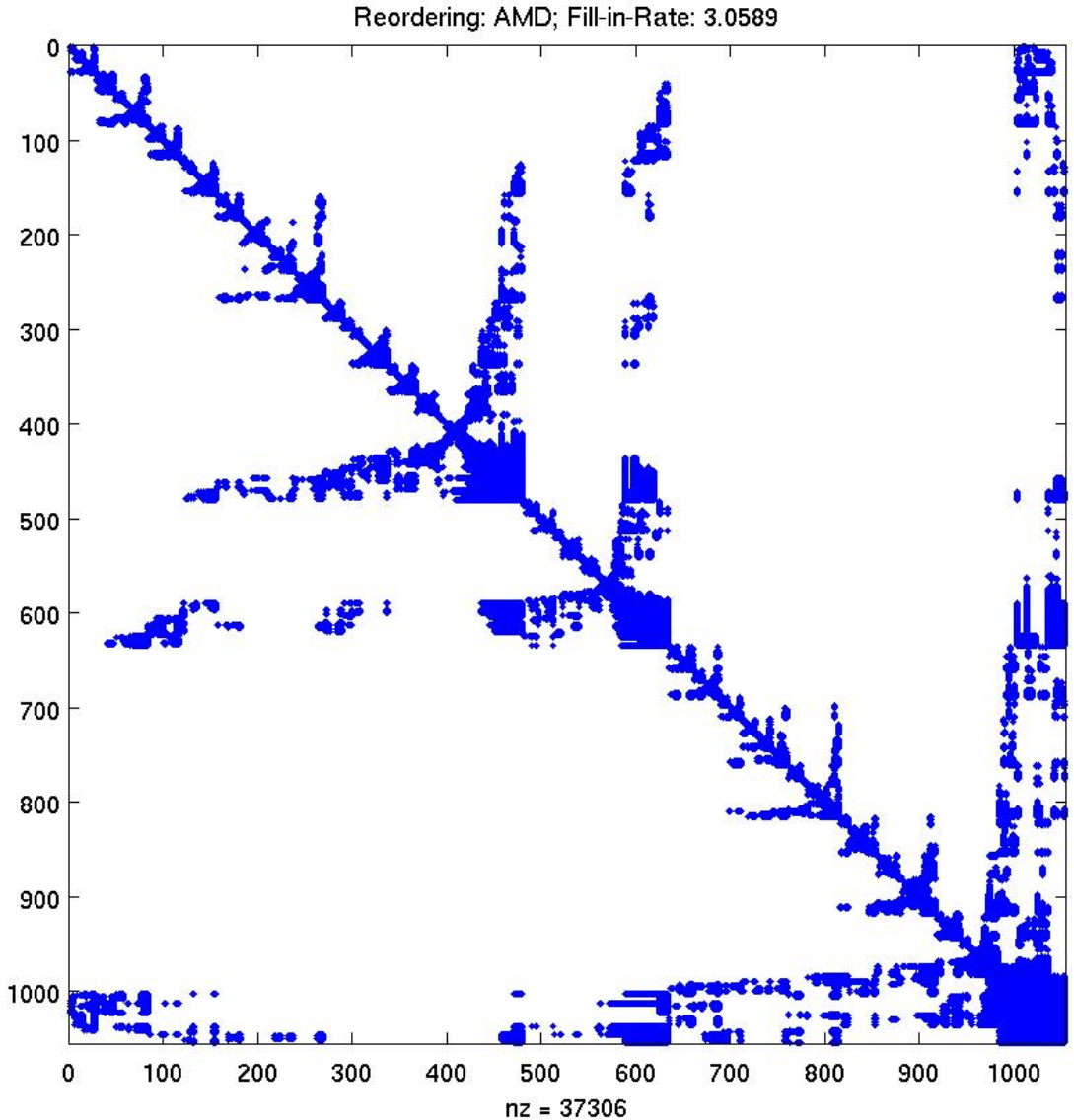

Reordering: MD
0
100
200
300
400
500
600
700
800
900
1000
nz = 12196


Reordering: MD; Fill-in-Rate: 3.3042
0
100
200
300
400
500
600
700
800
900
1000
nz = 40298

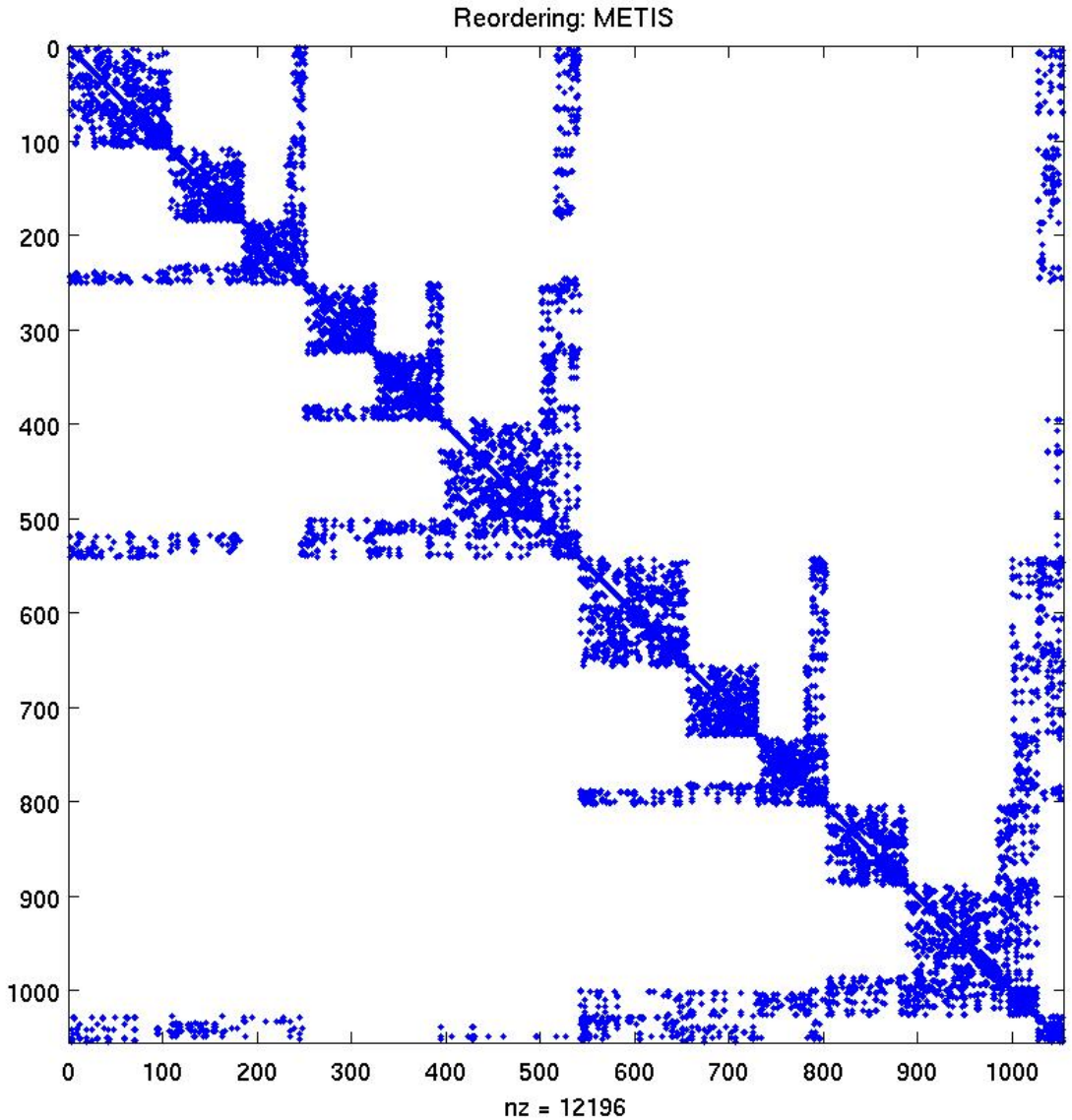
design A AMD


Reordering: AMD
0
100
200
300
400
500
600
700
800
900
1000
0
100
200
300
400
500
600
700
800
900
1000
nz = 12196

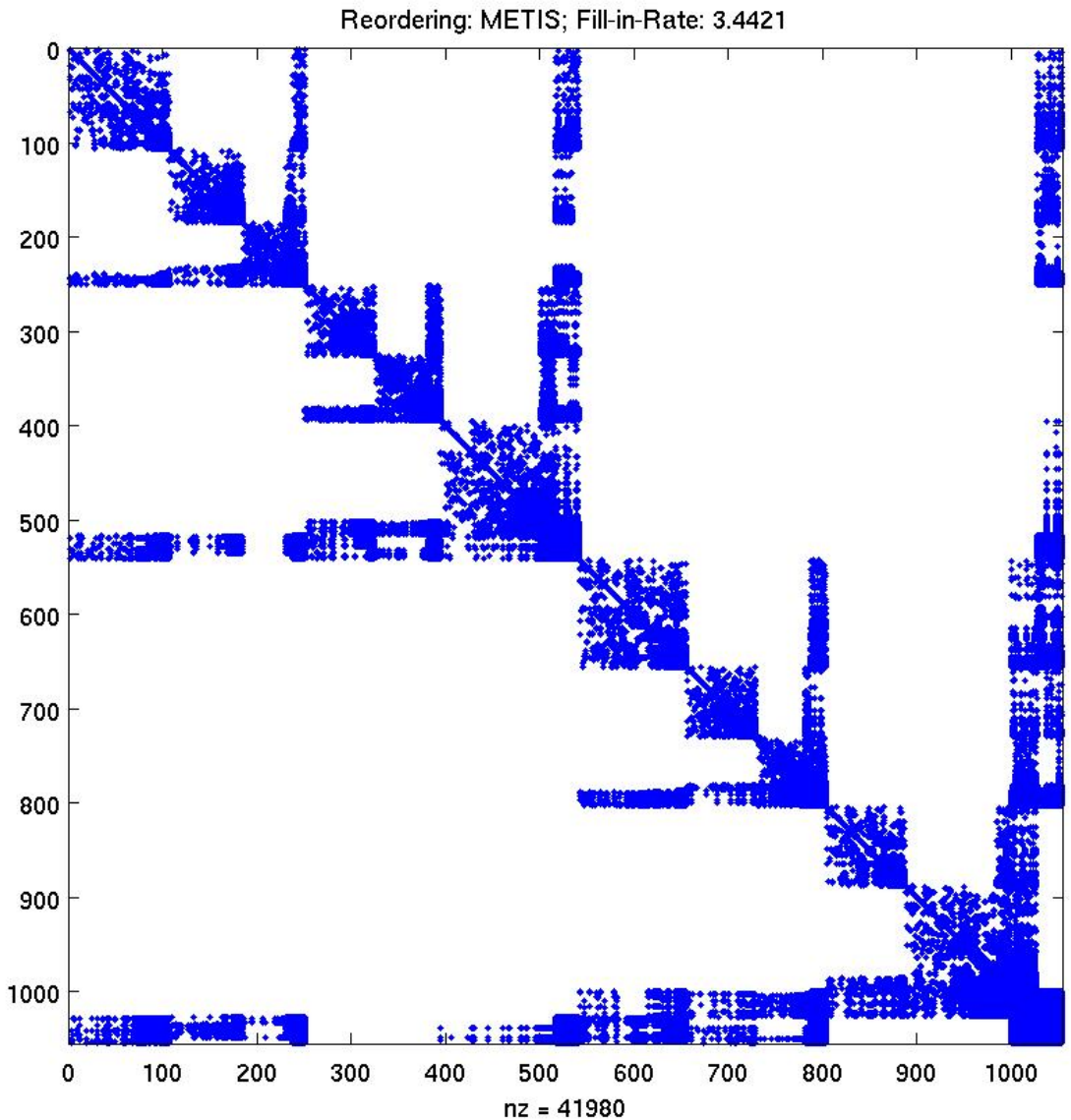
design LR AMD



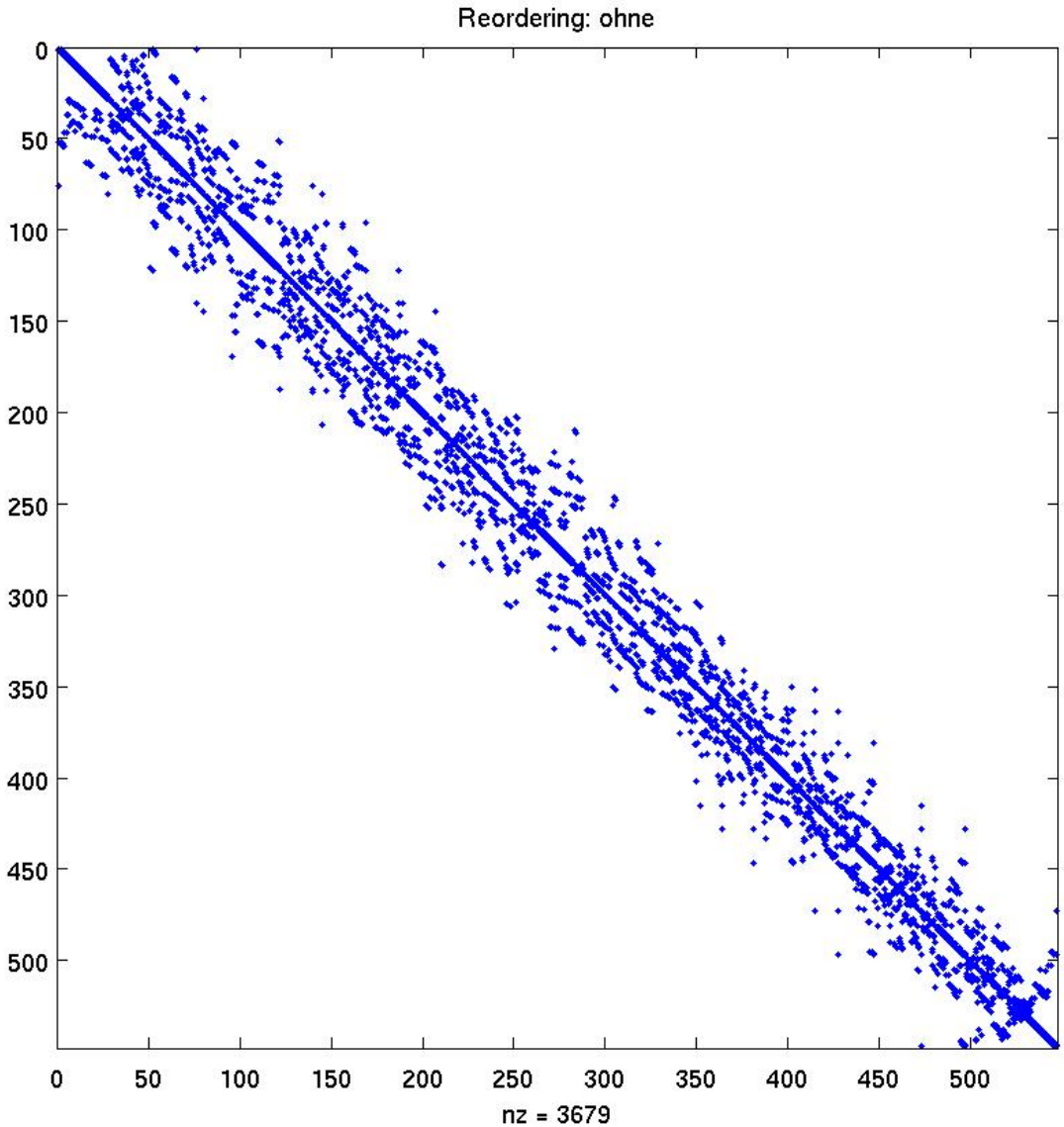
design A METIS



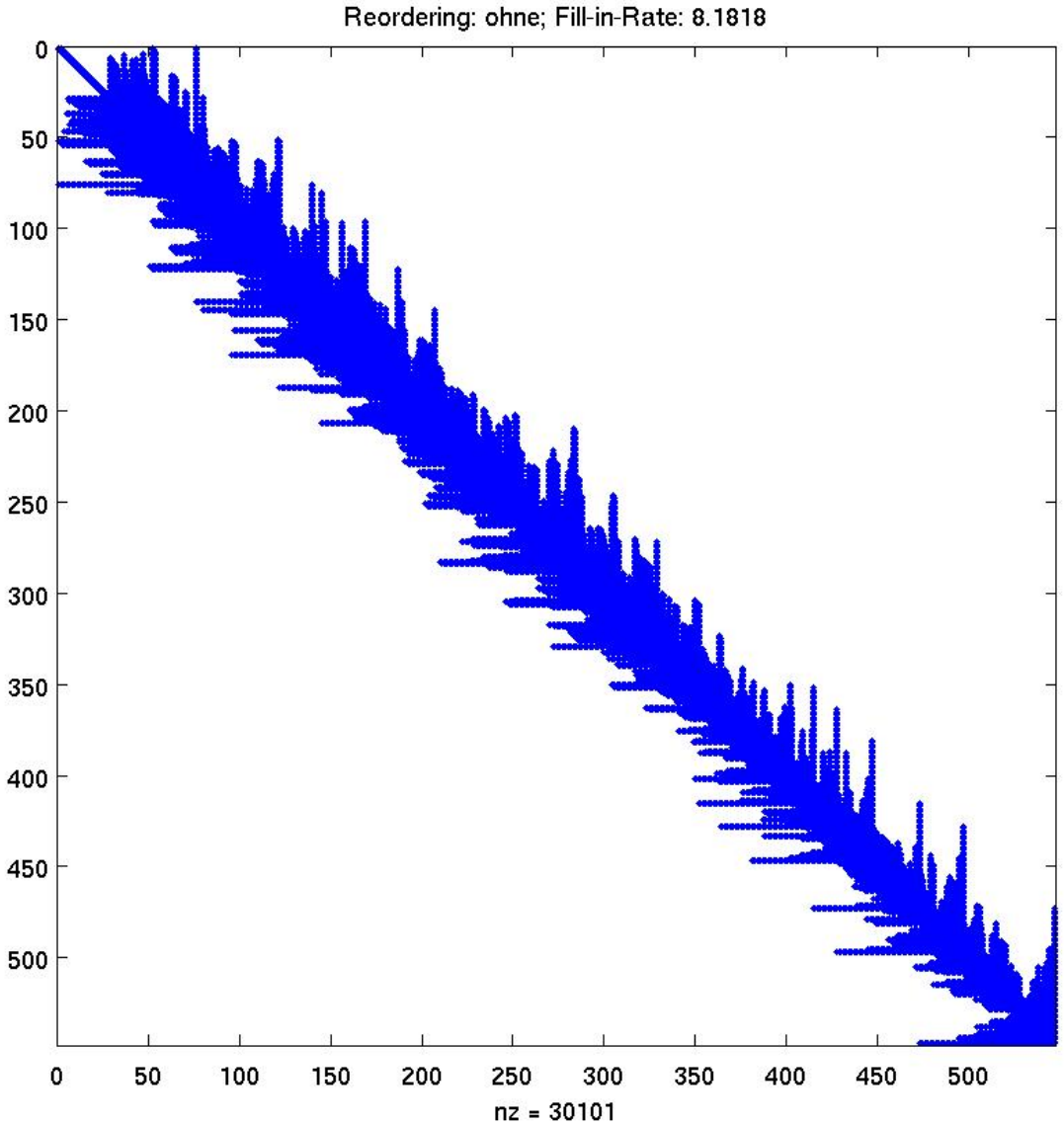
design LR METIS

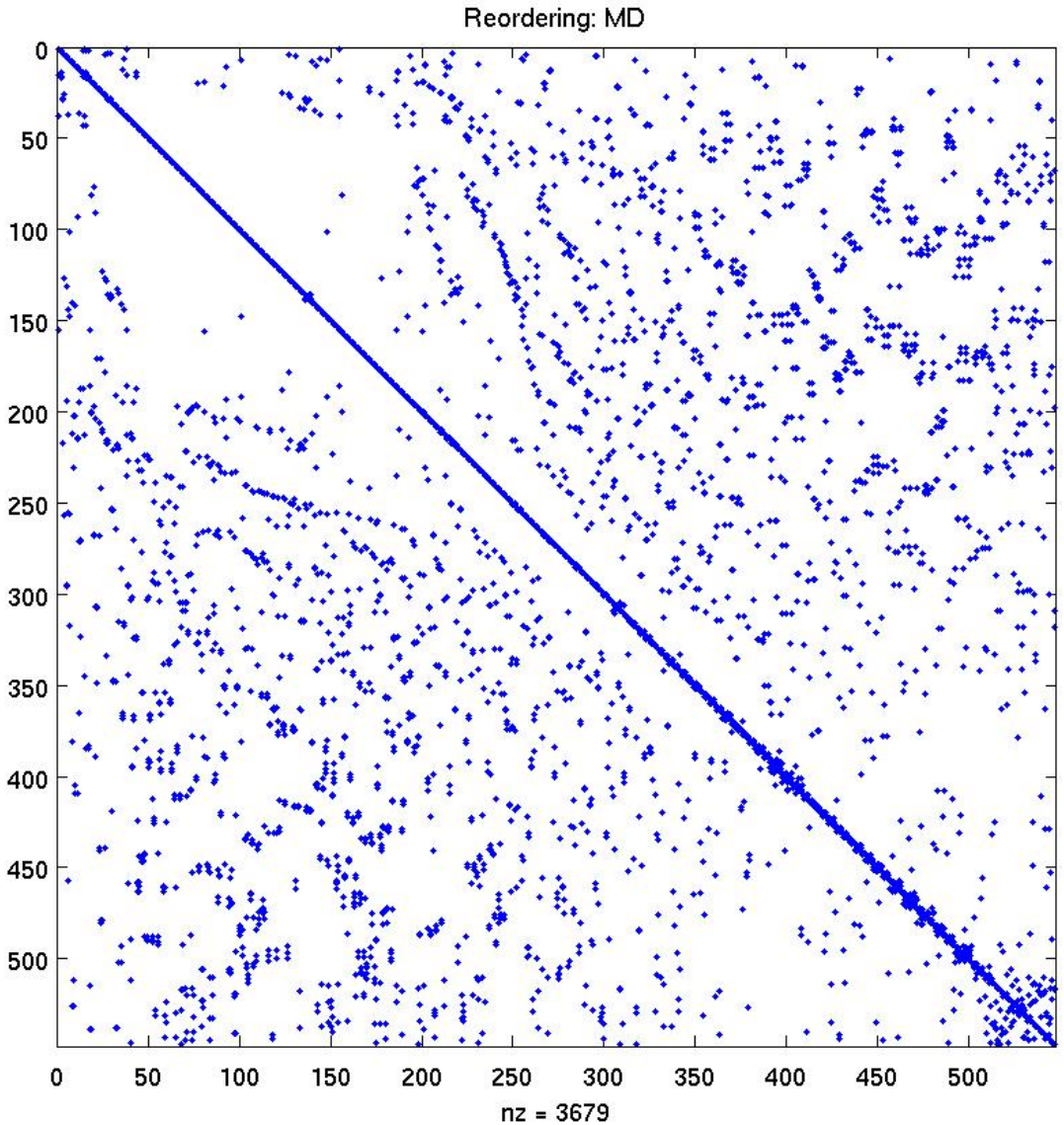


Eppstein A ohne

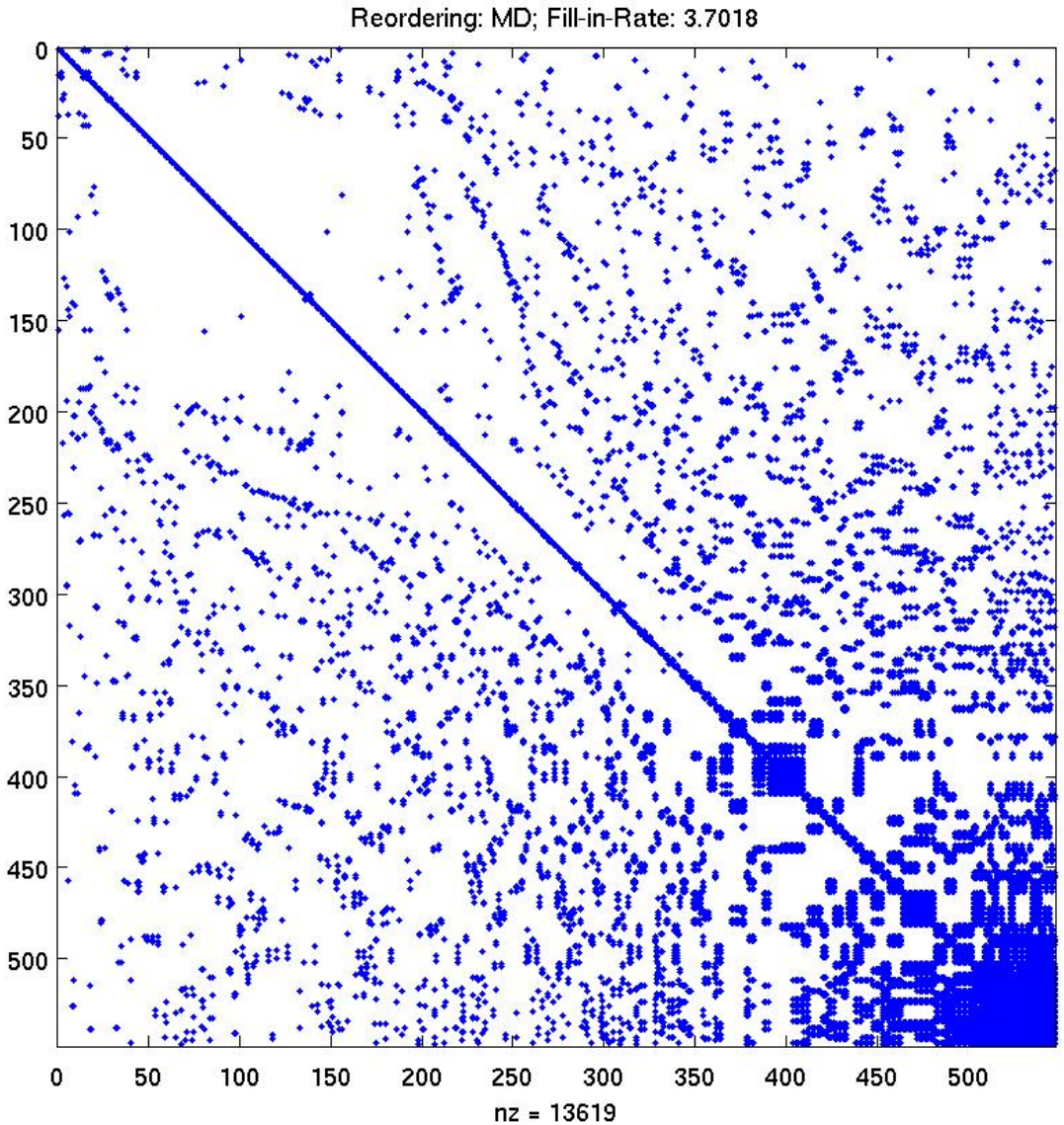


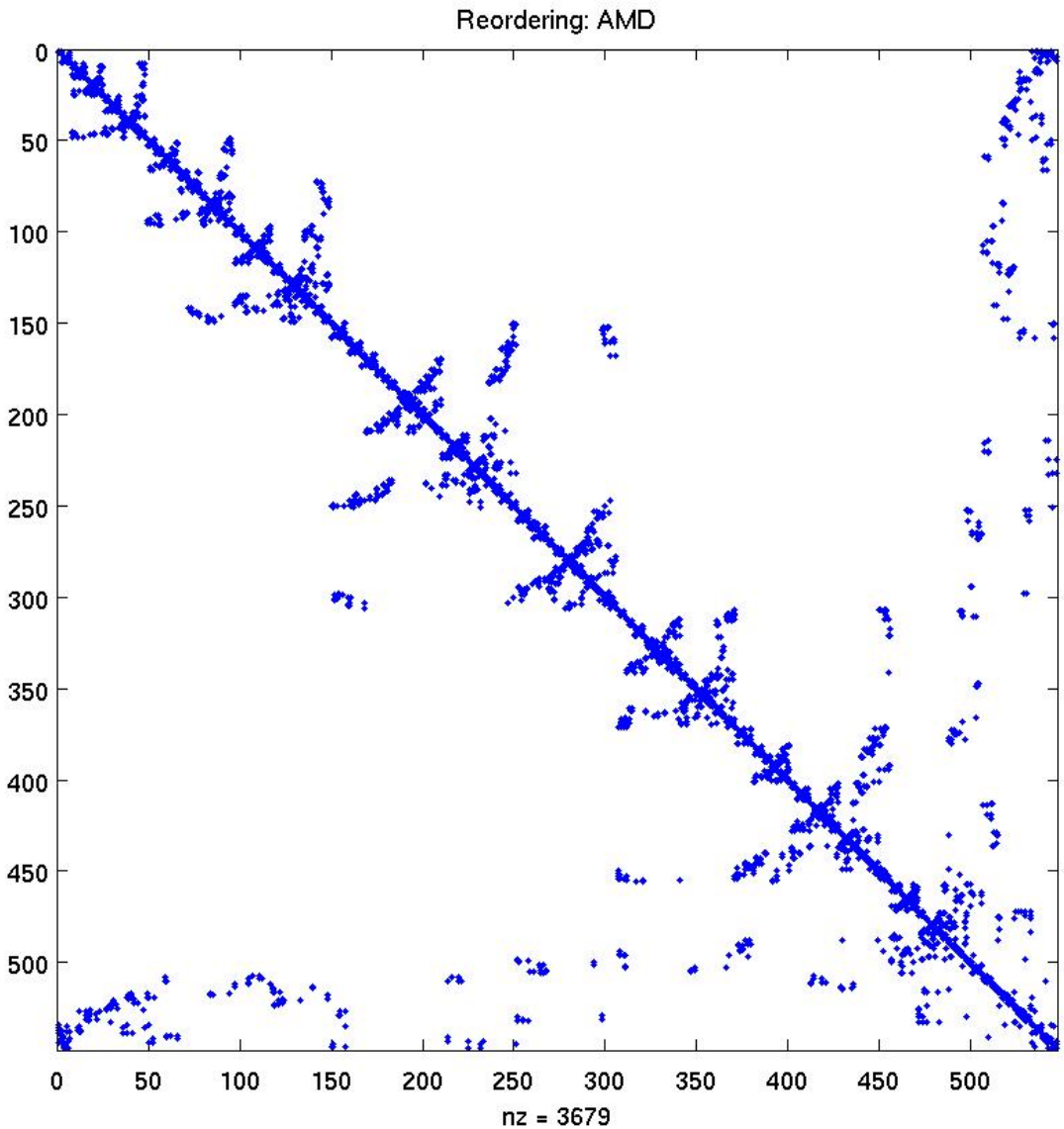
Eppstein LR ohne



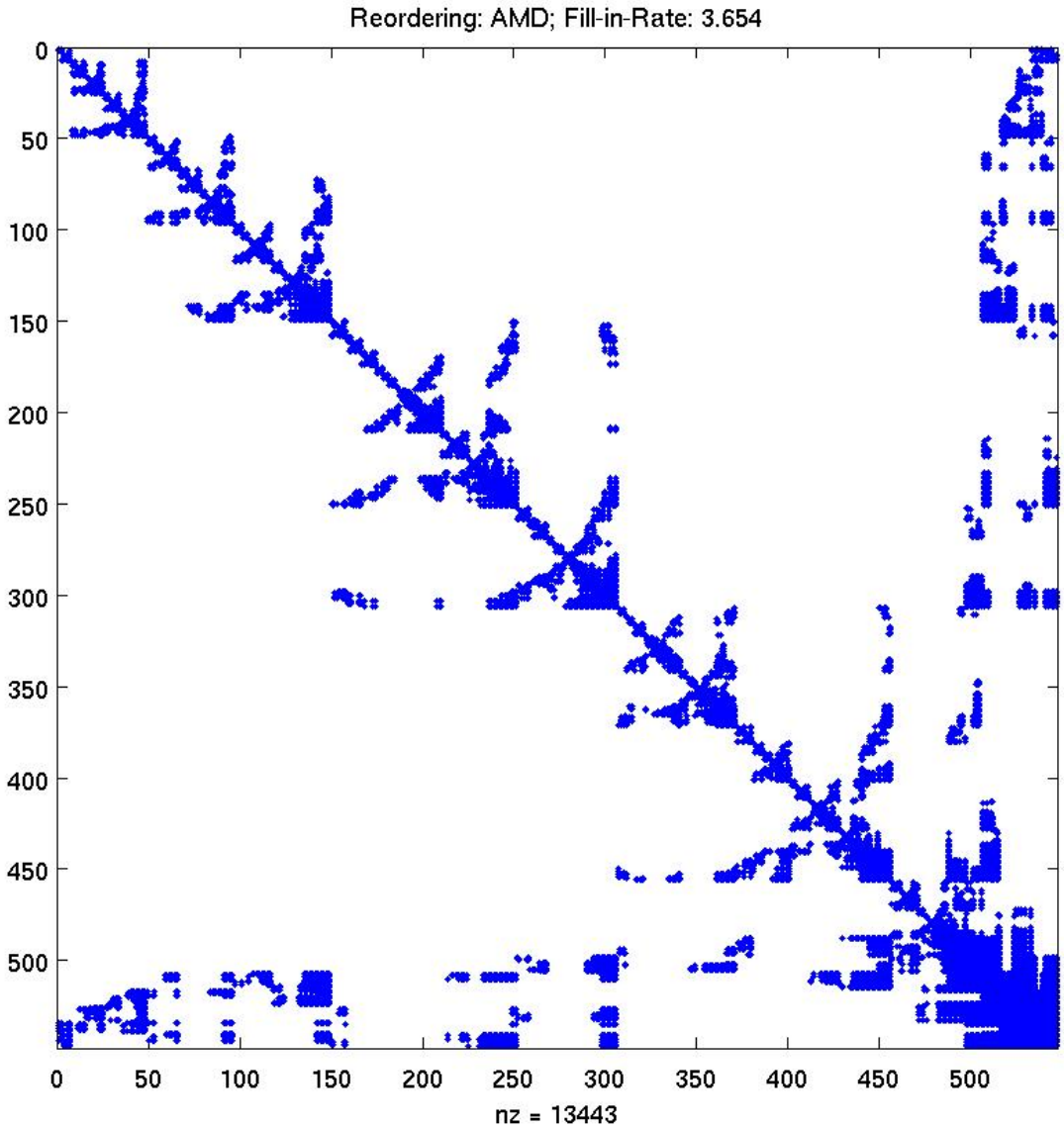

Reordering: MD
0
50
100
150
200
250
300
350
400
450
500
nz = 3679

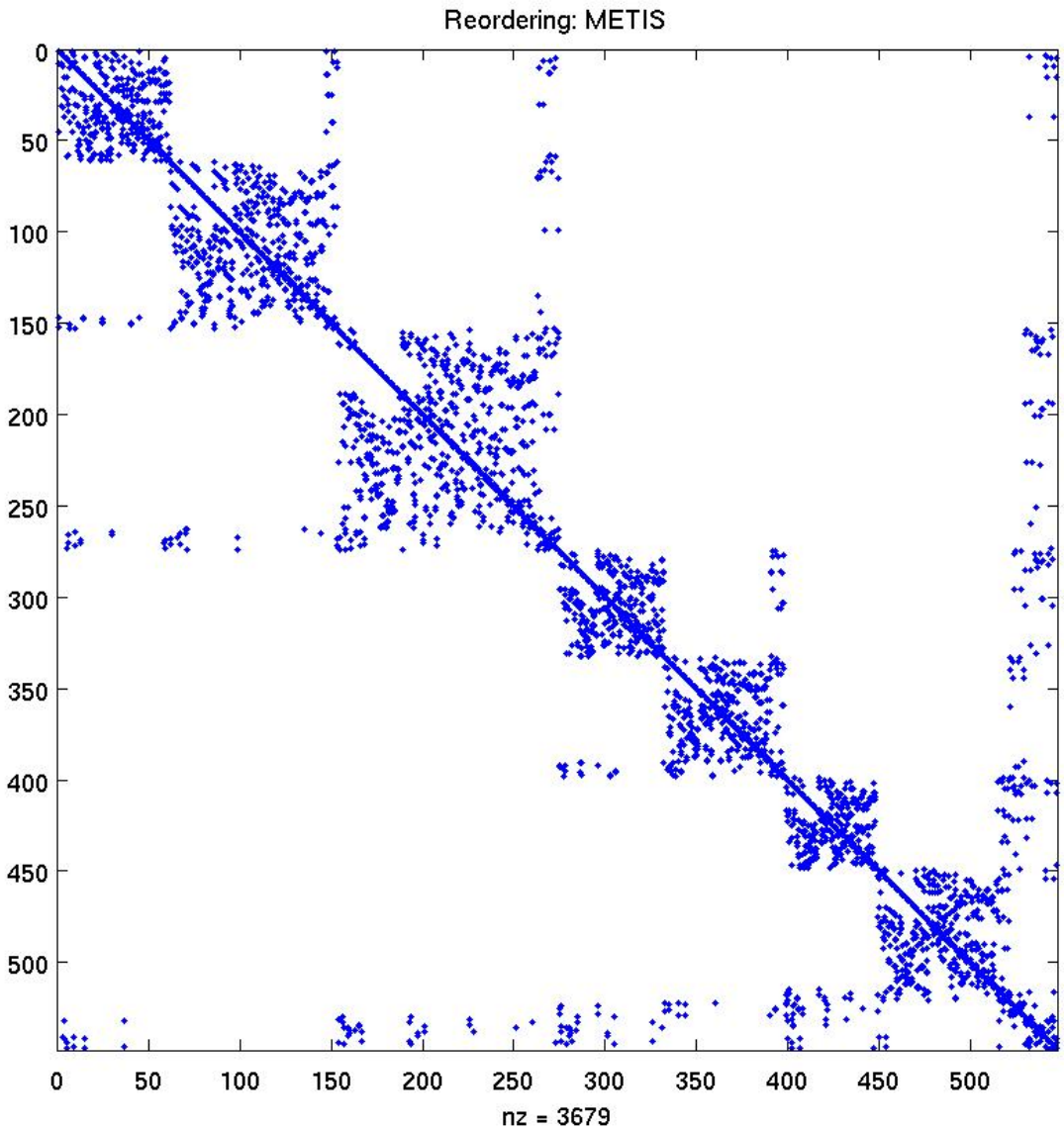
Eppstein LR MD



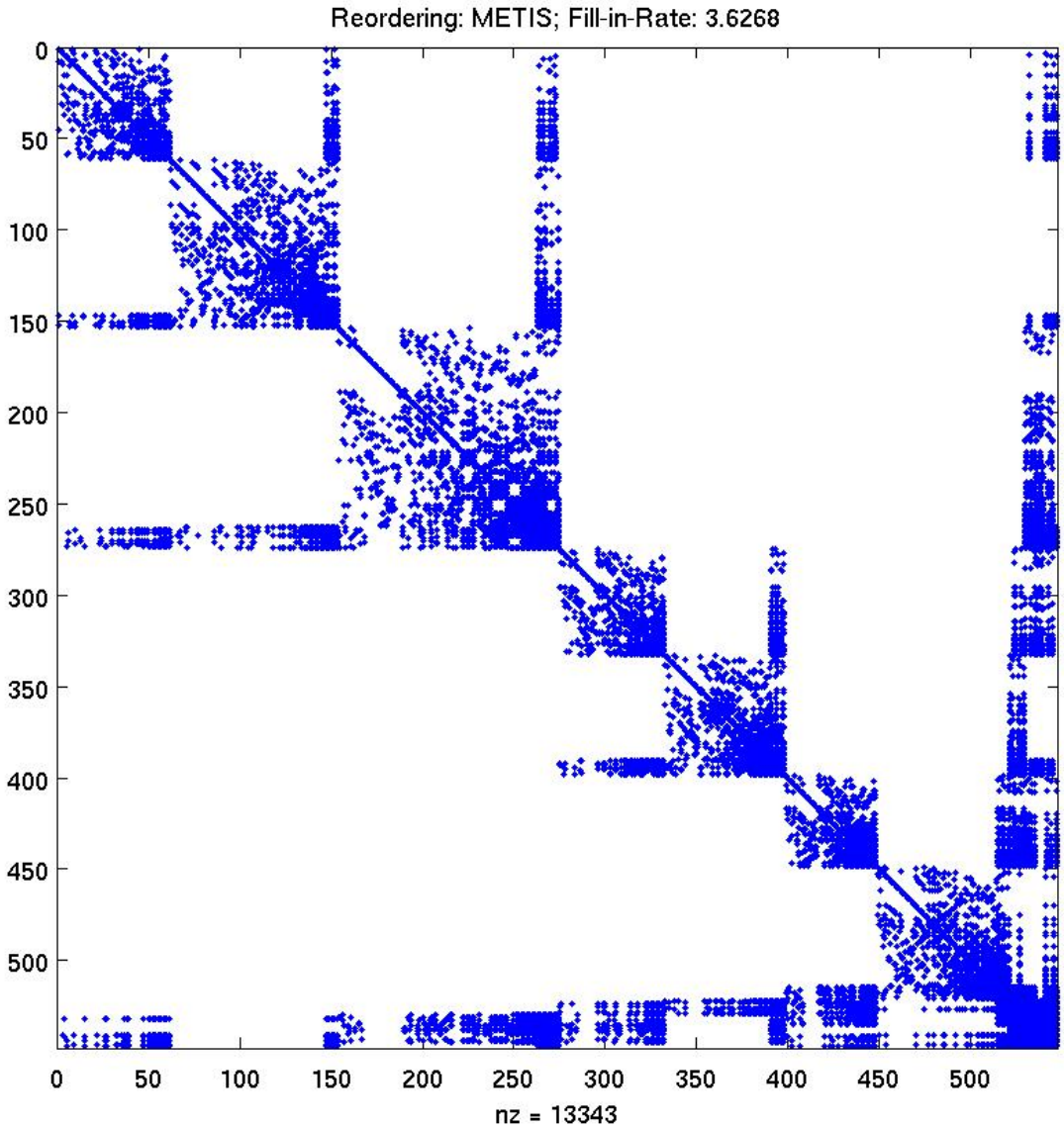

Reordering: AMD
0
50
100
150
200
250
300
350
400
450
500
0 50 100 150 200 250 300 350 400 450 500
nz = 3679

Eppstein LR AMD




Reordering: METIS
0
50
100
150
200
250
300
350
400
450
500
nz = 3679

Eppstein LR METIS



Fill-in-Raten

Problem	ohne	MD	AMD	METIS
poisson_2d	13.9091	3.618	3.3794	4.2734
Tapir	12.1989	1.9792	1.9488	2.1924
design	43.3413	3.3042	3.0589	3.4421
Eppstein	8.1818	3.7018	3.654	3.6268