eliminacja-gaussa

December 6, 2021

1 Dopełnienie Schur'a z użyciem eliminacji Gaussa

Maciej Skoczeń, Kacper Kafara

grupa wtorek (A) 17:50

1.1 Środowisko obliczeniowe

OSOBA WYKONUJĄCA OSTATECZNE OBLICZENIA POWINNA WPISAĆ TUTAJ SPECYFIKACJĘ SWOJEGO SPRZĘTU

1.2 Importy & typy

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import os
import re
import subprocess
import matplotlib.pyplot as plt

from timeit import default_timer
from pprint import pprint
from math import sqrt

Array = np.ndarray
```

1.3 Funkcje pomocnicze

[2]:
```
class Timer(object):
    def __init__(self):
        self._start_time = None
        self._stop_time = None

    def start(self):
        self._start_time = default_timer()

    def stop(self):
        self._stop_time = default_timer()
    @property
    def elapsed(self, val = None):
        if self._stop_time is None or self._start_time is None:
            return None
        elapsed = self._stop_time - self._start_time
        return elapsed

# mock impl
def is_int(value) -> bool:
    as_int = int(value)
    return value == as_int
```

1.3.1 Wczytywanie macierzy

wygenerowanej za pomocą dostarczonego skryptu `mass_matrix`

[3]:
```
def input_matrix(octave_matrix, n, m, q=1):
    result = np.zeros((n*q, m*q), dtype=np.double)

    for elem in octave_matrix:
        m = re.match(r"\s*\((\d+),\s*(\d+)\)\s*->\s*(\d+\.\d+)\s*", elem)
        if m is not None:
            x, y, value = m.groups()
        elif len(elem) > 0:
            coord, value = elem.strip().split(' -> ')
            value = float(value)
            x, y = coord.split(',')
            x, y = x[1:], y.strip()[:-1]
        else:
            continue

        for i in range(q):
            for j in range(q):
                result[i*n + int(x) - 1, j*n + int(y) - 1] = float(value)

    return result
```

[4]:
```
def load_octave_matrix(filename):
    with open(filename, "r") as file:
        return file.readlines()
```

[5]:
```
data_dir = "../../output"

def resolve_path(matrix_type, width, height = None, generate = False):
    if height is None: height = width
    path = f"{data_dir}/{matrix_type}-{width}x{height}.txt"
    if os.path.isfile(path): return path
    else:
        if not generate:
            raise FileNotFoundError(f"Matrix file {path} not found")

        # do generowania macierzy potrzebny jest direnv, ustawiona zmienna
        # środowiskowa:
        # SCRIPT_DIR=<path-to-scripts-dir>
        # albo na sztywno ustawiona ścieżka do skryptu (ale wtedy trzeba␣
↪zmodyfikować)
        # funkcję generate_matrix

        if width != height:
            raise ValueError("Can only generate square matrix")

        generate_matrix(matrix_type, width)

        if os.path.isfile(path): return path
        else:
            print(path)
            raise RuntimeError("Failed to generate matrix")


resolve_matrix = lambda matrix_type, n, m, q = 1: input_matrix(
    load_octave_matrix(resolve_path(matrix_type, n, m)), n, m, q
)

def resolve_matrix(matrix_type, n, m, q = 1, generate = False):
    return input_matrix(
        load_octave_matrix(resolve_path(matrix_type, n, m, generate =␣
↪generate)), n, m, q
    )

def generate_matrix(matrix_type, rank):
    if matrix_type not in {'iga', 'fem'}:
        raise ValueError(f"Invalid matrix type: {matrix_type}")

    if rank < 16 or not is_int(sqrt(rank)):
        raise ValueError(f"Invalid matrix rank: {rank}. Must be >= 16 and␣
↪sqrt(rank) must be of type integer.")

    rank_root = int(sqrt(rank))

    if matrix_type == 'fem':
        for p in range(2, 5):
            double_nxx = rank_root - p + 1
            if double_nxx % 2 == 0 and double_nxx // 2 >= 2:
                nxx = double_nxx // 2
                pxx = p
                break
        else:
            raise RuntimeError(f"Failed to determine nxx, pxx for rank: {rank}")
    else:
        for p in range(2, 5):
            nxx = rank_root - p
            if nxx >= 2:
                pxx = p
                break
        else:
            raise RuntimeError(f"Failed to determine nxx, pxx for rank: {rank}")

    cwd = os.getcwd()
    scripts_dir = os.getenv('SCRIPTS_DIR')
    os.chdir(scripts_dir)
    !./generate-matrix.sh cpp {matrix_type} {nxx} {pxx} 0
    os.chdir(cwd)
```

1.4 Eliminacja Gaussa

[6]:
```
def transform_matrix_gaussian_elim(
    A: Array,
    rows_to_transform: int,
    in_place: bool = False,
    timer: Timer = None
) -> Array:

    if not in_place: A = A.copy()

    if timer is not None:
        timer.start()

    n, _ = A.shape
    for i in range(0, min(n - 2, rows_to_transform)):
        for j in range(i + 1, n):
            factor = A[j, i] / A[i, i]
            A[j, i] = 0
            for k in range(i + 1, n):
                A[j, k] -= factor * A[i, k]

    if timer is not None: timer.stop()
    if not in_place: return A
```

1.5 Dopełnienie Schur'a

[7]:
```
def schur_complement(A: Array, complement_degree: int, timer: Timer = None) ->␣
↪Array:
    transformed = transform_matrix_gaussian_elim(A,
                                                 A.shape[0] - complement_degree,
                                                 in_place = False,
                                                 timer = timer)
    return transformed[A.shape[0] - complement_degree :, A.shape[1] -␣
↪complement_degree :]
```

[8]:
```
!pwd
```

```
/home/kkafara/studies/cs/5_term/algmac/lab/gaussian-elimination/notebook
```

[20]:
```
nxxs = {}
nxxs['iga'] = [i for i in range(2, 31)]
nxxs['fem'] = [i for i in range(2, 17)]

pxx = 2
rxx = 0
ranks = {}
ranks['iga'] = [(nxx + pxx) ** 2 for nxx in nxxs['iga']]
ranks['fem'] = [(2 * nxx + pxx - 1) ** 2 for nxx in nxxs['fem']]

matrixtypes = 'iga', 'fem'

main_timer = Timer()
exec_times = {
    'iga': {},
    'fem': {}
}
exec_ranks = {
    'iga': {},
    'fem': {},
}

padding = lambda n: n * ' '

for matrix_t in matrixtypes:
    matrices = ((resolve_matrix(matrix_t, rank, rank, generate=True), rank) for␣
↪rank in ranks[matrix_t])
    print('Computations for matrix type: ', matrix_t)
    for M, rank in matrices:
        print(padding(2) + 'Computations for rank', rank)

        exec_times[matrix_t][rank] = []
        exec_ranks[matrix_t][rank] = []
        rank_cp = rank

        while rank_cp >= 2:
            rank_cp //= 2
            print(padding(4) + 'Current rank:', rank_cp, end = '  ')

            schur_complement(M, rank_cp, timer = main_timer)

            exec_times[matrix_t][rank].append(main_timer.elapsed)
            exec_ranks[matrix_t][rank].append(rank_cp)

            print(f'{main_timer.elapsed:.5f}s')


# narysować wykresy
```

```
Computations for matrix type:  iga
  Computations for rank 16
    Current rank: 8  0.00075s
    Current rank: 4  0.00123s
    Current rank: 2  0.00101s
    Current rank: 1  0.00076s
  Computations for rank 25
    Current rank: 12  0.00238s
    Current rank: 6  0.00228s
    Current rank: 3  0.00350s
    Current rank: 1  0.00379s
  Computations for rank 36
    Current rank: 18  0.00440s
    Current rank: 9  0.00462s
    Current rank: 4  0.00477s
    Current rank: 2  0.00496s
    Current rank: 1  0.00905s
  Computations for rank 49
    Current rank: 24  0.01063s
    Current rank: 12  0.01229s
    Current rank: 6  0.01202s
    Current rank: 3  0.01383s
    Current rank: 1  0.01751s
  Computations for rank 64
    Current rank: 32  0.03655s
    Current rank: 16  0.04000s
    Current rank: 8  0.04095s
    Current rank: 4  0.04186s
    Current rank: 2  0.04037s
    Current rank: 1  0.04178s
```

```
Computations for rank 81
  Current rank: 40  0.07391s
  Current rank: 20  0.07870s
  Current rank: 10  0.07801s
  Current rank: 5  0.08165s
  Current rank: 2  0.08227s
  Current rank: 1  0.08440s
Computations for rank 100
  Current rank: 50  0.13834s
  Current rank: 25  0.15109s
  Current rank: 12  0.15552s
  Current rank: 6  0.15496s
  Current rank: 3  0.15679s
  Current rank: 1  0.15757s
Computations for rank 121
  Current rank: 60  0.22200s
  Current rank: 30  0.26664s
  Current rank: 15  0.27062s
  Current rank: 7  0.27304s
  Current rank: 3  0.27369s
  Current rank: 1  0.27310s
Computations for rank 144
  Current rank: 72  0.41342s
  Current rank: 36  0.45191s
  Current rank: 18  0.45974s
  Current rank: 9  0.45623s
  Current rank: 4  0.46186s
  Current rank: 2  0.46887s
  Current rank: 1  0.46057s
Computations for rank 169
  Current rank: 84  0.66071s
  Current rank: 42  0.73841s
  Current rank: 21  0.74220s
  Current rank: 10  0.75428s
  Current rank: 5  0.75880s
  Current rank: 2  0.76519s
  Current rank: 1  0.78945s
Computations for rank 196
  Current rank: 98  0.98230s
  Current rank: 49  1.13027s
  Current rank: 24  1.15195s
  Current rank: 12  1.15543s
  Current rank: 6  1.15346s
  Current rank: 3  1.15778s
  Current rank: 1  1.18706s
Computations for rank 225
  Current rank: 112  1.62045s
  Current rank: 56  1.68813s
  Current rank: 28  1.75042s
  Current rank: 14  1.75386s
  Current rank: 7  1.75242s
  Current rank: 3  1.80162s
  Current rank: 1  1.79601s
Computations for rank 256
  Current rank: 128  2.21997s
  Current rank: 64  2.48662s
  Current rank: 32  2.52082s
  Current rank: 16  1.78867s
  Current rank: 8  1.72938s
  Current rank: 4  1.69499s
  Current rank: 2  1.69583s
  Current rank: 1  1.70110s
Computations for rank 289
  Current rank: 144  2.22495s
  Current rank: 72  2.40366s
  Current rank: 36  2.46225s
  Current rank: 18  2.47903s
  Current rank: 9  2.46434s
  Current rank: 4  2.47989s
  Current rank: 2  2.47103s
  Current rank: 1  2.47540s
Computations for rank 324
  Current rank: 162  3.12979s
  Current rank: 81  3.45529s
  Current rank: 40  3.45127s
  Current rank: 20  3.44973s
  Current rank: 10  3.53206s
  Current rank: 5  3.57458s
  Current rank: 2  3.55266s
  Current rank: 1  3.49209s
Computations for rank 361
  Current rank: 180  4.29937s
  Current rank: 90  4.92142s
  Current rank: 45  4.90069s
  Current rank: 22  4.77491s
  Current rank: 11  4.76546s
  Current rank: 5  4.76500s
  Current rank: 2  4.66289s
  Current rank: 1  4.73909s
Computations for rank 400
  Current rank: 200  5.70056s
  Current rank: 100  6.41666s
  Current rank: 50  6.47354s
  Current rank: 25  6.44148s
  Current rank: 12  6.41988s
  Current rank: 6  6.40460s
  Current rank: 3  6.47782s
  Current rank: 1  6.47268s
Computations for rank 441
  Current rank: 220  7.51674s
  Current rank: 110  8.36805s
  Current rank: 55  8.63710s
  Current rank: 27  8.45592s
  Current rank: 13  8.48101s
  Current rank: 6  8.44459s
  Current rank: 3  8.45866s
  Current rank: 1  8.48642s
Computations for rank 484
  Current rank: 242  9.79070s
  Current rank: 121  11.34586s
  Current rank: 60  11.60459s
  Current rank: 30  11.33224s
  Current rank: 15  11.27676s
  Current rank: 7  11.32292s
  Current rank: 3  11.24706s
  Current rank: 1  11.47624s
Computations for rank 529
  Current rank: 264  13.01130s
  Current rank: 132  14.79236s
  Current rank: 66  14.99826s
  Current rank: 33  14.75422s
  Current rank: 16  14.90480s
  Current rank: 8  14.85562s
  Current rank: 4  14.83468s
  Current rank: 2  15.06062s
  Current rank: 1  14.91126s
Computations for rank 576
  Current rank: 288  17.03305s
  Current rank: 144  19.04820s
  Current rank: 72  18.90196s
  Current rank: 36  19.21373s
  Current rank: 18  19.06076s
  Current rank: 9  18.98547s
  Current rank: 4  19.07566s
  Current rank: 2  19.47488s
  Current rank: 1  19.22384s
Computations for rank 625
  Current rank: 312  21.74368s
  Current rank: 156  24.25006s
  Current rank: 78  24.44085s
  Current rank: 39  24.03898s
  Current rank: 19  24.01059s
  Current rank: 9  24.11553s
  Current rank: 4  24.14060s
  Current rank: 2  24.72337s
  Current rank: 1  24.20373s
Computations for rank 676
  Current rank: 338  26.82188s
  Current rank: 169  30.30014s
  Current rank: 84  30.83771s
  Current rank: 42  41.53998s
  Current rank: 21  48.51427s
  Current rank: 10  47.82864s
  Current rank: 5  48.04088s
  Current rank: 2  48.49843s
  Current rank: 1  49.23474s
Computations for rank 729
  Current rank: 364  53.23434s
  Current rank: 182  59.87689s
  Current rank: 91  60.75340s
  Current rank: 45  59.46213s
  Current rank: 22  59.85515s
  Current rank: 11  59.61798s
  Current rank: 5  59.42579s
  Current rank: 2  60.62136s
  Current rank: 1  59.70957s
Computations for rank 784
  Current rank: 392  65.51775s
  Current rank: 196  72.98892s
  Current rank: 98  72.82517s
  Current rank: 49  75.04273s
  Current rank: 24  74.07694s
  Current rank: 12  74.71456s
  Current rank: 6  73.92650s
  Current rank: 3  76.11925s
  Current rank: 1  73.85607s
Computations for rank 841
  Current rank: 420  79.83563s
  Current rank: 210  89.99321s
  Current rank: 105  91.10169s
  Current rank: 52  91.53710s
  Current rank: 26  91.66941s
  Current rank: 13  91.35388s
  Current rank: 6  91.57323s
  Current rank: 3  91.46118s
  Current rank: 1  91.33152s
Computations for rank 900
  Current rank: 450  98.07199s
  Current rank: 225  110.30664s
  Current rank: 112  112.65123s
  Current rank: 56  112.48517s
  Current rank: 28  111.79289s
```

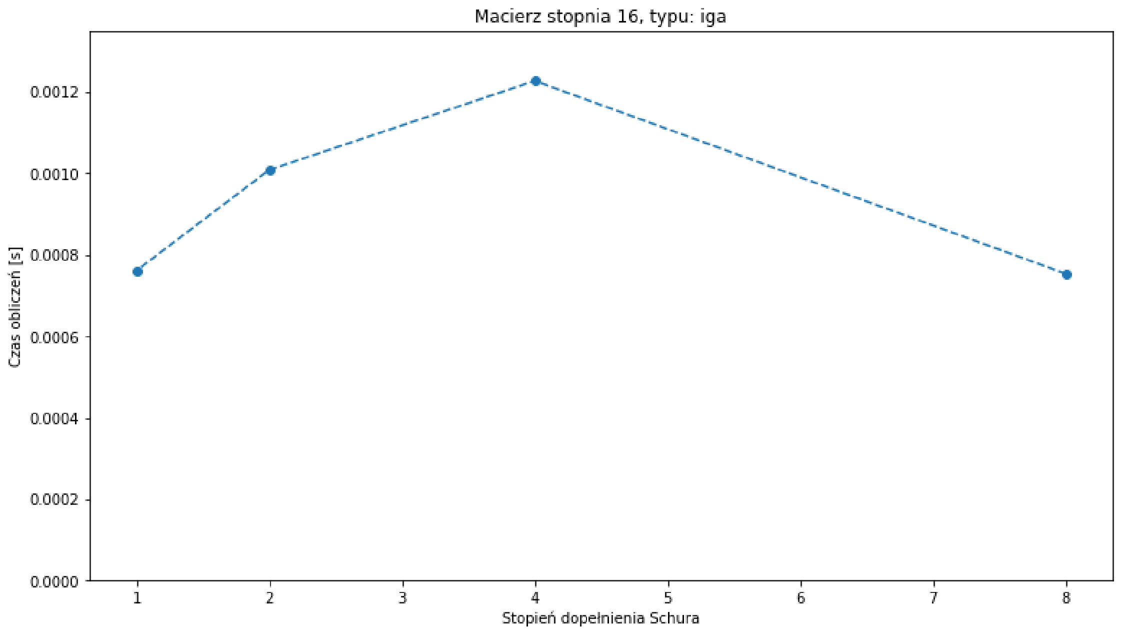
```
    Current rank: 14  111.92657s
    Current rank: 7  111.98247s
    Current rank: 3  113.57932s
    Current rank: 1  112.77645s
  Computations for rank 961
    Current rank: 480  119.19338s
    Current rank: 240  134.86661s
    Current rank: 120  135.18978s
    Current rank: 60  136.31368s
    Current rank: 30  136.67959s
    Current rank: 15  136.31056s
    Current rank: 7  136.33844s
    Current rank: 3  136.22324s
    Current rank: 1  136.18598s
  Computations for rank 1024
    Current rank: 512  144.57645s
    Current rank: 256  162.25881s
    Current rank: 128  165.22331s
    Current rank: 64  164.94650s
    Current rank: 32  165.36069s
    Current rank: 16  165.75301s
    Current rank: 8  167.27332s
    Current rank: 4  171.30454s
    Current rank: 2  166.94476s
    Current rank: 1  165.00125s
Computations for matrix type:  fem
  Computations for rank 25
    Current rank: 12  0.00216s
    Current rank: 6  0.00241s
    Current rank: 3  0.00243s
    Current rank: 1  0.00243s
  Computations for rank 49
    Current rank: 24  0.01721s
    Current rank: 12  0.01805s
    Current rank: 6  0.01826s
    Current rank: 3  0.01832s
    Current rank: 1  0.01829s
  Computations for rank 81
    Current rank: 40  0.07063s
    Current rank: 20  0.08016s
    Current rank: 10  0.08086s
    Current rank: 5  0.08032s
    Current rank: 2  0.08097s
    Current rank: 1  0.08159s
  Computations for rank 121
    Current rank: 60  0.22443s
    Current rank: 30  0.26429s
    Current rank: 15  0.27094s
    Current rank: 7  0.26522s
    Current rank: 3  0.26269s
    Current rank: 1  0.26750s
  Computations for rank 169
    Current rank: 84  0.62810s
    Current rank: 42  0.71580s
    Current rank: 21  0.72528s
    Current rank: 10  0.72653s
    Current rank: 5  0.72665s
    Current rank: 2  0.72303s
    Current rank: 1  0.72598s
  Computations for rank 225
    Current rank: 112  1.49737s
    Current rank: 56  1.68422s
    Current rank: 28  1.70750s
    Current rank: 14  1.70824s
    Current rank: 7  1.71026s
    Current rank: 3  1.70135s
    Current rank: 1  1.70961s
  Computations for rank 289
    Current rank: 144  3.21928s
    Current rank: 72  3.60245s
    Current rank: 36  3.63444s
    Current rank: 18  3.62478s
    Current rank: 9  3.58227s
    Current rank: 4  3.64339s
    Current rank: 2  3.64402s
    Current rank: 1  3.64012s
  Computations for rank 361
    Current rank: 180  6.18690s
    Current rank: 90  7.06585s
    Current rank: 45  7.10753s
    Current rank: 22  7.16536s
    Current rank: 11  7.11833s
    Current rank: 5  7.13548s
    Current rank: 2  7.09971s
    Current rank: 1  7.12072s
riga=1 (fem)
nxx=10
pxx=2
rxx=0
  Computations for rank 441
    Current rank: 220  11.54784s
    Current rank: 110  13.04338s
    Current rank: 55  13.20003s
    Current rank: 27  13.27893s
    Current rank: 13  13.29883s
    Current rank: 6  13.30256s
    Current rank: 3  13.29342s
    Current rank: 1  13.43963s
riga=1 (fem)
nxx=11
pxx=2
rxx=0
  Computations for rank 529
    Current rank: 264  20.37553s
    Current rank: 132  22.88775s
    Current rank: 66  22.97887s
    Current rank: 33  23.40335s
    Current rank: 16  23.36707s
    Current rank: 8  23.28913s
    Current rank: 4  22.73321s
    Current rank: 2  22.64538s
    Current rank: 1  22.65424s
riga=1 (fem)
nxx=12
pxx=2
rxx=0
  Computations for rank 625
    Current rank: 312  32.71539s
    Current rank: 156  37.93587s
    Current rank: 78  37.32883s
    Current rank: 39  37.47362s
    Current rank: 19  37.30623s
    Current rank: 9  37.25037s
    Current rank: 4  37.77985s
    Current rank: 2  37.38508s
    Current rank: 1  37.35437s
riga=1 (fem)
nxx=13
pxx=2
rxx=0
  Computations for rank 729
    Current rank: 364  53.46581s
    Current rank: 182  58.63274s
    Current rank: 91  59.56702s
    Current rank: 45  59.63339s
    Current rank: 22  59.66795s
    Current rank: 11  59.52414s
    Current rank: 5  59.35871s
    Current rank: 2  59.29677s
    Current rank: 1  59.32461s
riga=1 (fem)
nxx=14
pxx=2
rxx=0
```

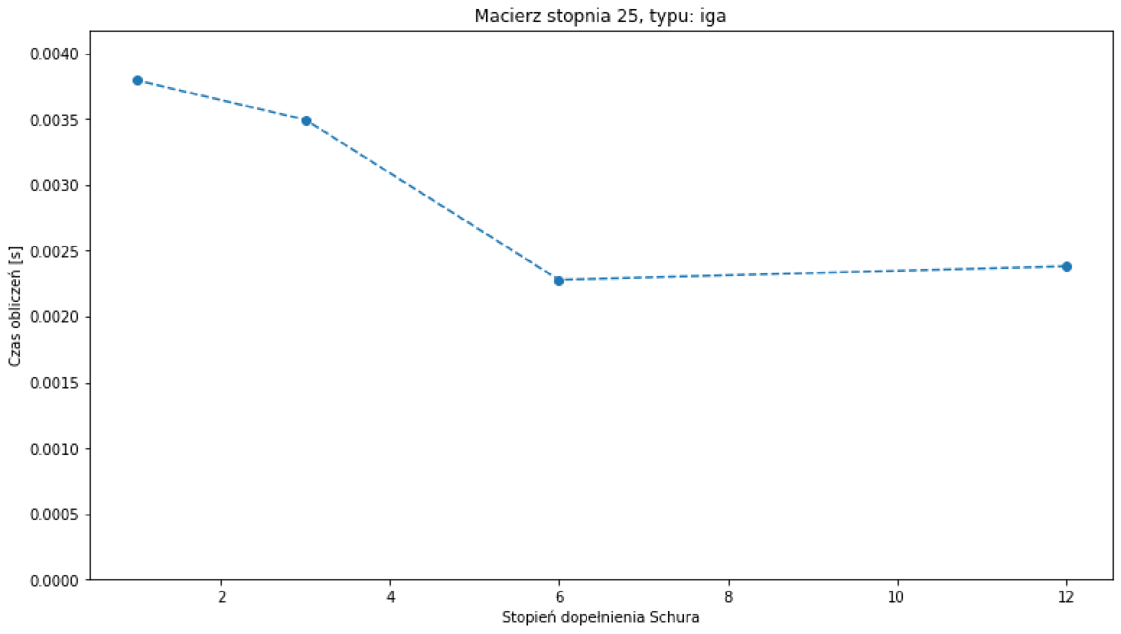
```
  Computations for rank 841
    Current rank: 420  79.68035s
    Current rank: 210  90.05779s
    Current rank: 105  91.06791s
    Current rank: 52  91.44742s
    Current rank: 26  91.70011s
    Current rank: 13  91.44929s
    Current rank: 6  91.18409s
    Current rank: 3  91.59026s
    Current rank: 1  91.97174s
riga=1 (fem)
nxx=15
pxx=2
rxx=0
  Computations for rank 961
    Current rank: 480  119.58912s
    Current rank: 240  102.46927s
    Current rank: 120  88.42724s
    Current rank: 60  88.37991s
    Current rank: 30  87.62772s
    Current rank: 15  89.47669s
    Current rank: 7  88.42573s
    Current rank: 3  88.50560s
    Current rank: 1  88.66723s
riga=1 (fem)
nxx=16
pxx=2
rxx=0
  Computations for rank 1089
    Current rank: 544  112.89838s
    Current rank: 272  194.15977s
    Current rank: 136  198.74234s
    Current rank: 68  198.69099s
    Current rank: 34  199.46881s
    Current rank: 17  198.57189s
    Current rank: 8  162.81246s
    Current rank: 4  182.19425s
    Current rank: 2  198.33317s
    Current rank: 1  198.39083s
riga=1 (fem)
nxx=17
pxx=2
rxx=0
  Computations for rank 1225
    Current rank: 612
```

```
KeyboardInterrupt                         Traceback (most recent call last)
/tmp/ipykernel_186981/3375954883.py in <module>
     35             print(padding(4) + 'Current rank:', rank_cp, end = '   ')
     36
---> 37             schur_complement(M, rank_cp, timer = main_timer)
     38
     39             exec_times[matrix_t][rank].append(main_timer.elapsed)

/tmp/ipykernel_186981/3888681352.py in schur_complement(A, complement_degree,␣
 ↪timer)
      1 def schur_complement(A: Array, complement_degree: int, timer: Timer =␣
 ↪None) -> Array:
----> 2     transformed = transform_matrix_gaussian_elim(A,

      3                                               A.shape[0] -␣
 ↪complement_degree,
      4                                               in_place = False,
      5                                               timer = timer)

/tmp/ipykernel_186981/3300222045.py in transform_matrix_gaussian_elim(A,␣
 ↪rows_to_transform, in_place, timer)
     17             A[j, i] = 0
     18             for k in range(i + 1, n):
---> 19                 A[j, k] -= factor * A[i, k]
     20
     21     if timer is not None: timer.stop()

KeyboardInterrupt:
```

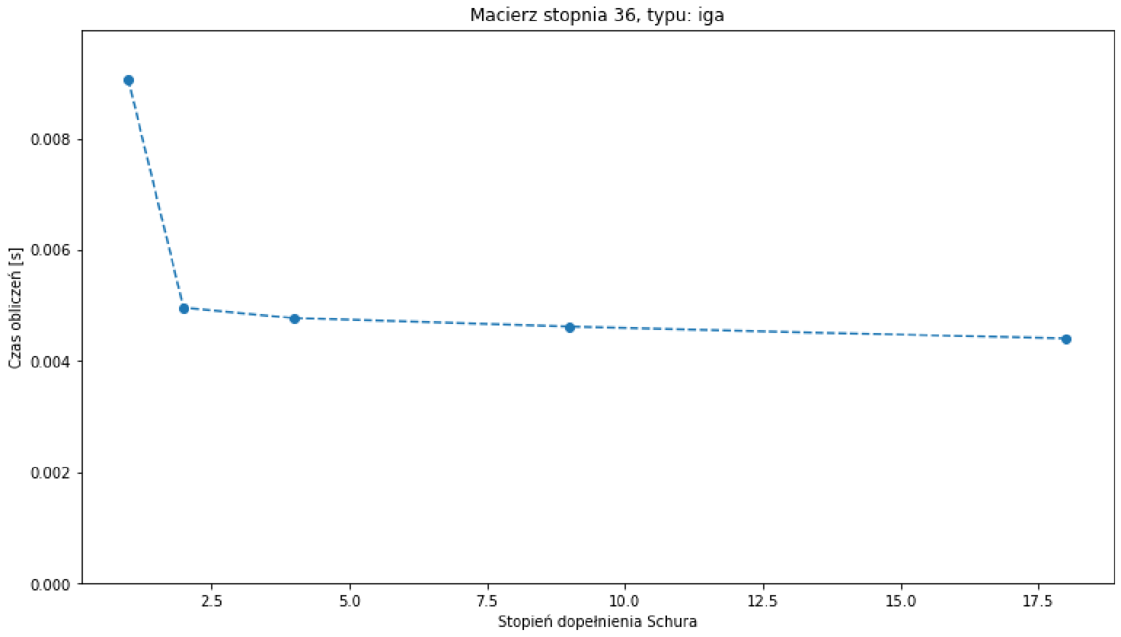
[23]:
```
%matplotlib inline

for matrix_t in matrixtypes:
    for rank in ranks[matrix_t]:
        _, ax = plt.subplots(figsize=(12.7, 7))

        max_y = max(exec_times[matrix_t][rank])
        max_y += 0.1 * max_y
        plt.ylim(0, max_y)

        ax.scatter(
            exec_ranks[matrix_t][rank],
            exec_times[matrix_t][rank],
            label=f'{matrix_t}'
        )

        ax.plot(
            exec_ranks[matrix_t][rank],
            exec_times[matrix_t][rank],
            linestyle='--'
        )

        ax.set(
            xlabel='Stopień dopełnienia Schura',
            ylabel='Czas obliczeń [s]',
            title=f'Macierz stopnia {rank}, typu: {matrix_t}'
        )
```

```
/tmp/ipykernel_186981/2887758238.py:5: RuntimeWarning: More than 20 figures have
been opened. Figures created through the pyplot interface
(`matplotlib.pyplot.figure`) are retained until explicitly closed and may
consume too much memory. (To control this warning, see the rcParam
`figure.max_open_warning`).
  _, ax = plt.subplots(figsize=(12.7, 7))
```

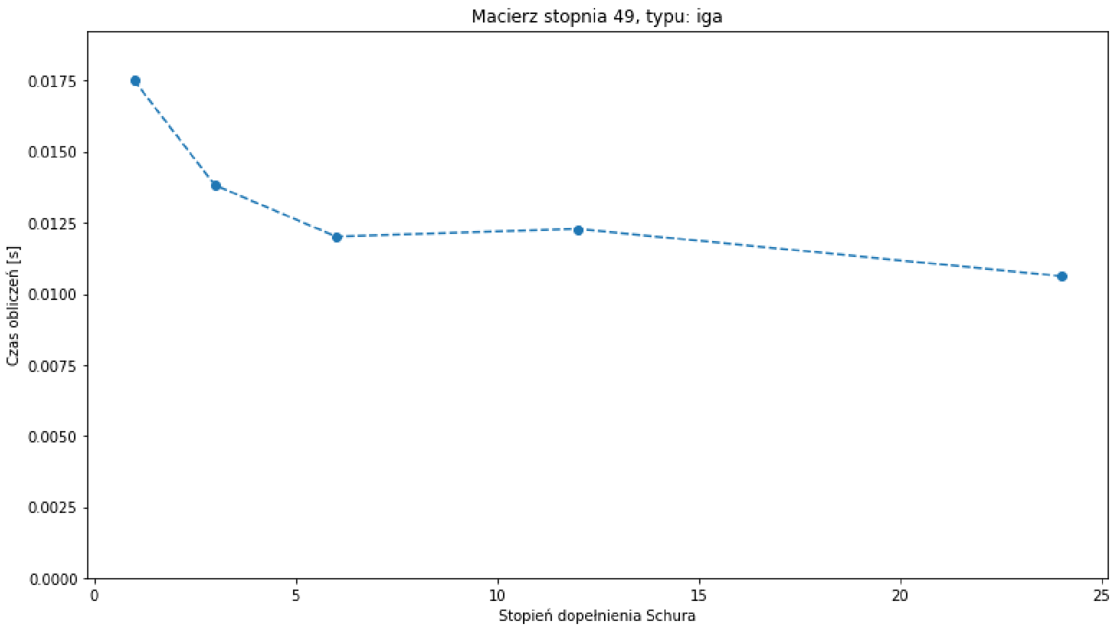


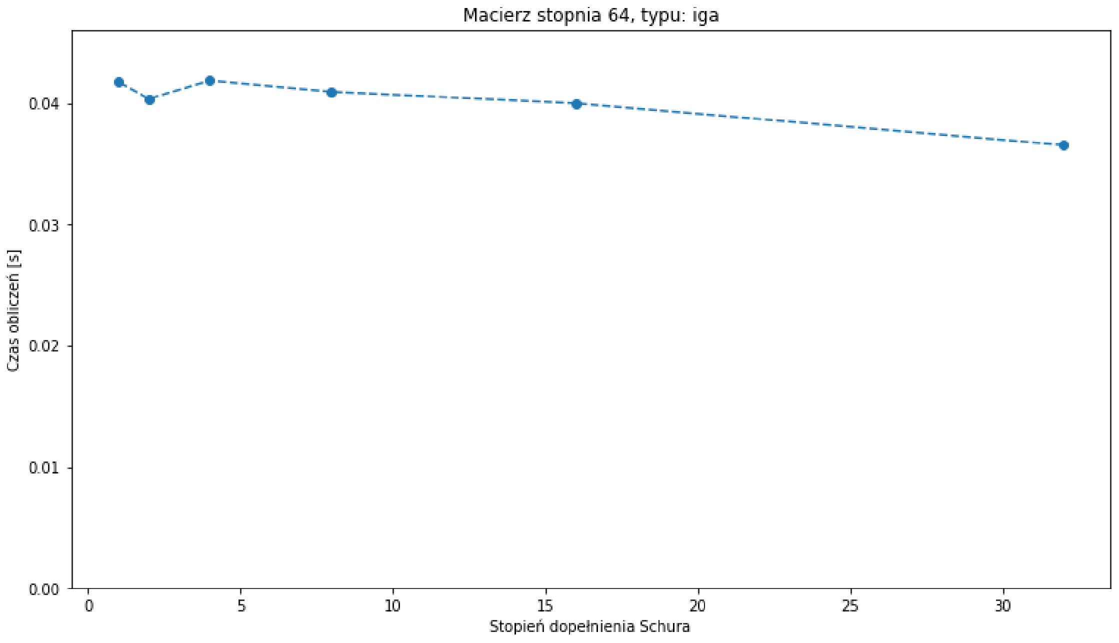


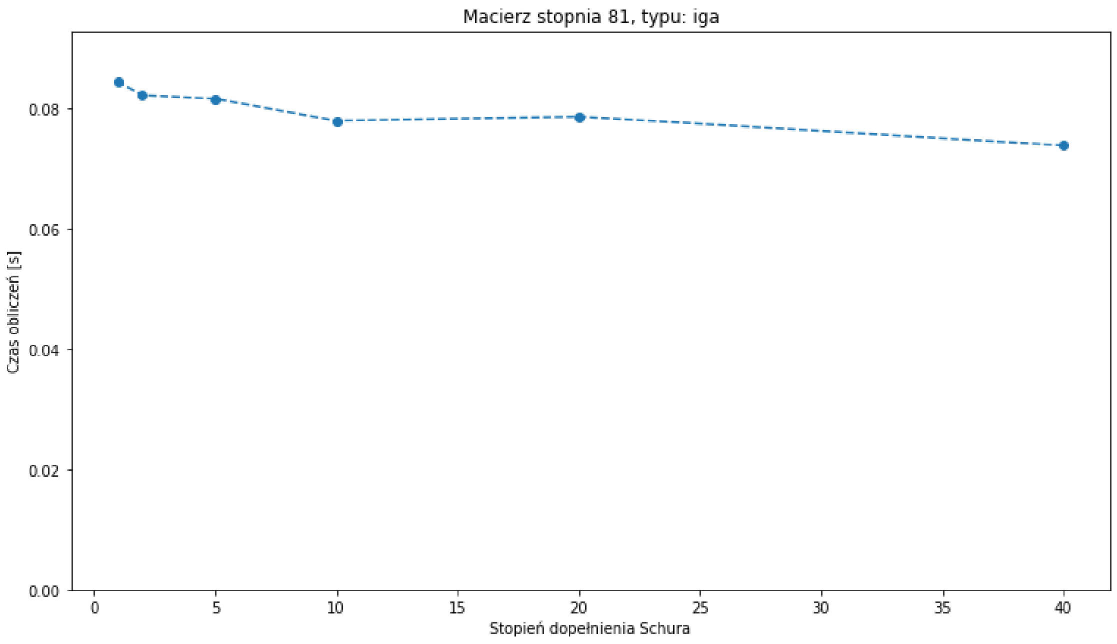
Macierz stopnia 25, typu: iga
Czas obliczeń [s]
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
0.0040
2
4
6
8
10
12
Stopień dopełnienia Schura

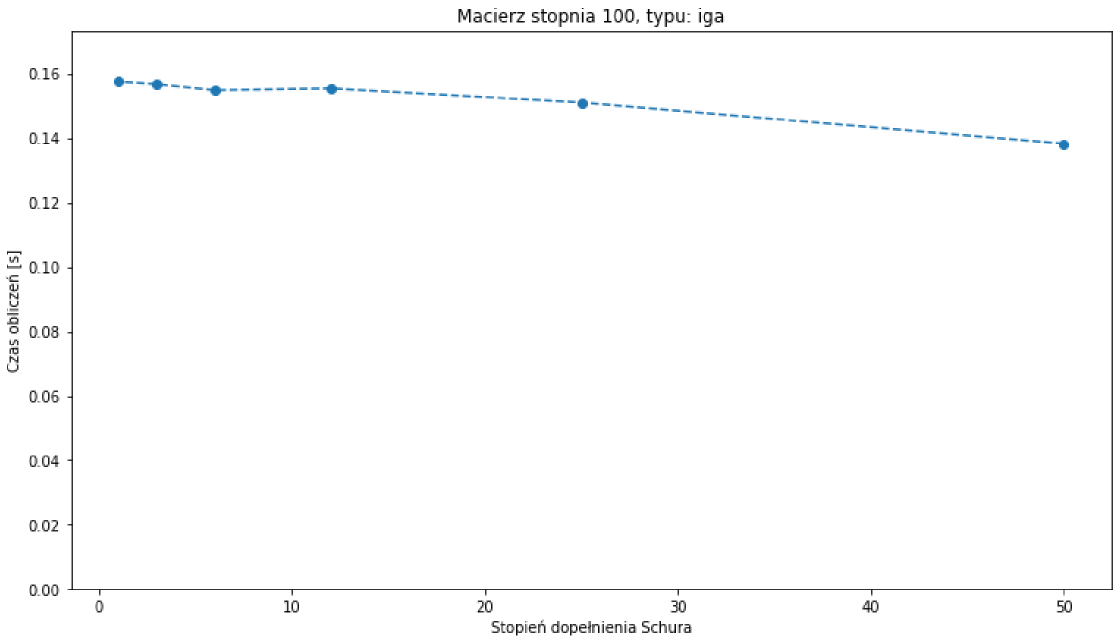


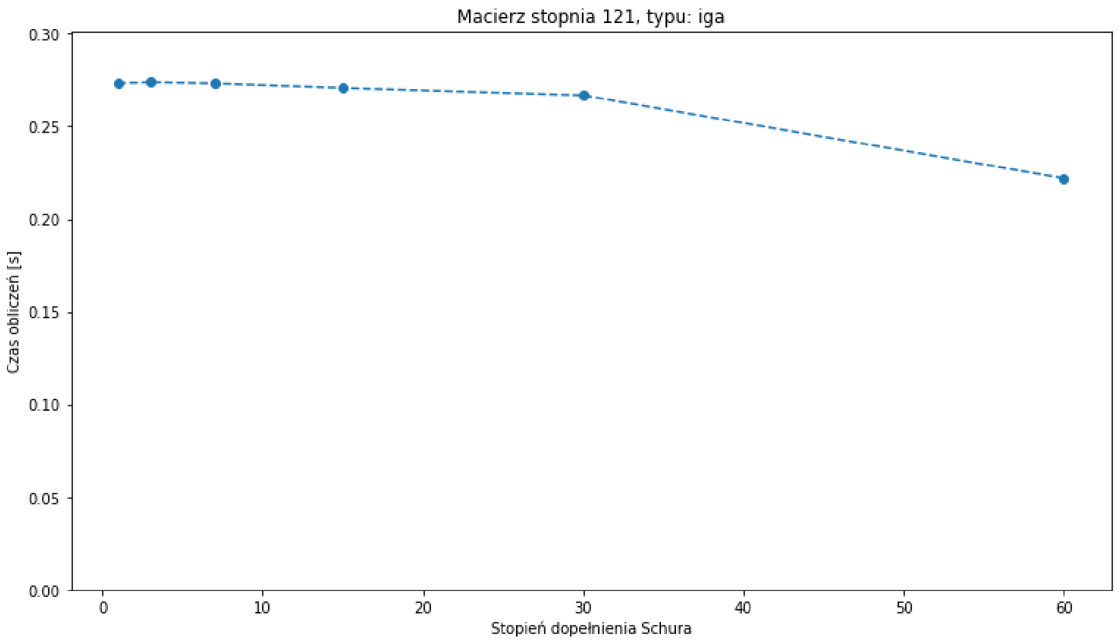
Macierz stopnia 36, typu: iga
Czas obliczeń [s]
0.000
0.002
0.004
0.006
0.008
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Stopień dopełnienia Schura

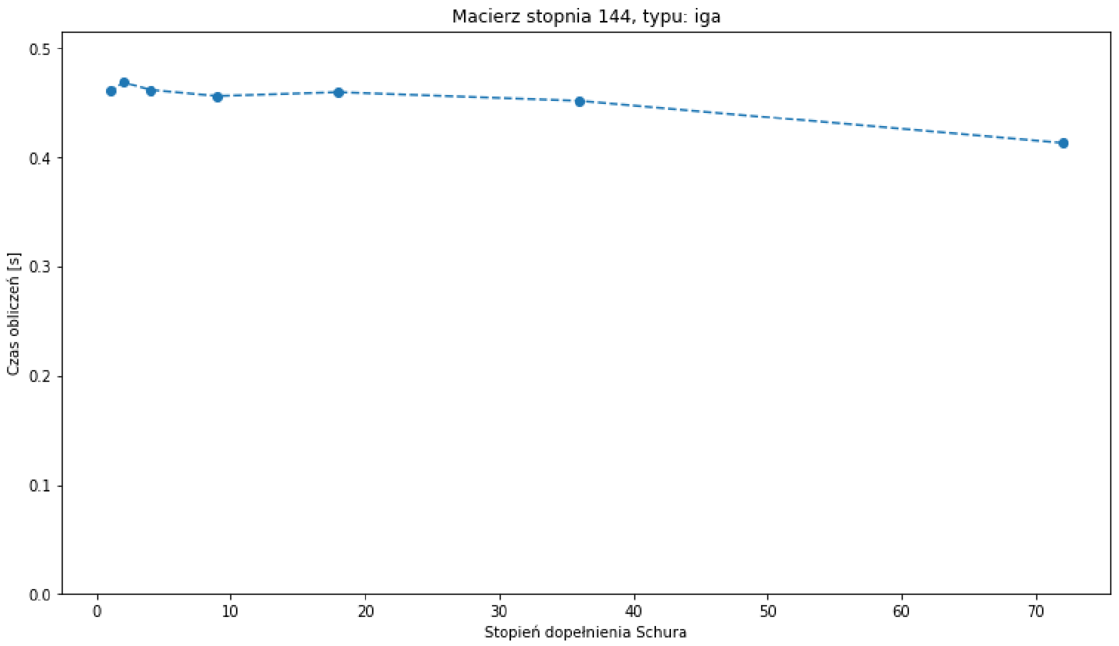

Macierz stopnia 49, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0
5
10
15
20
25

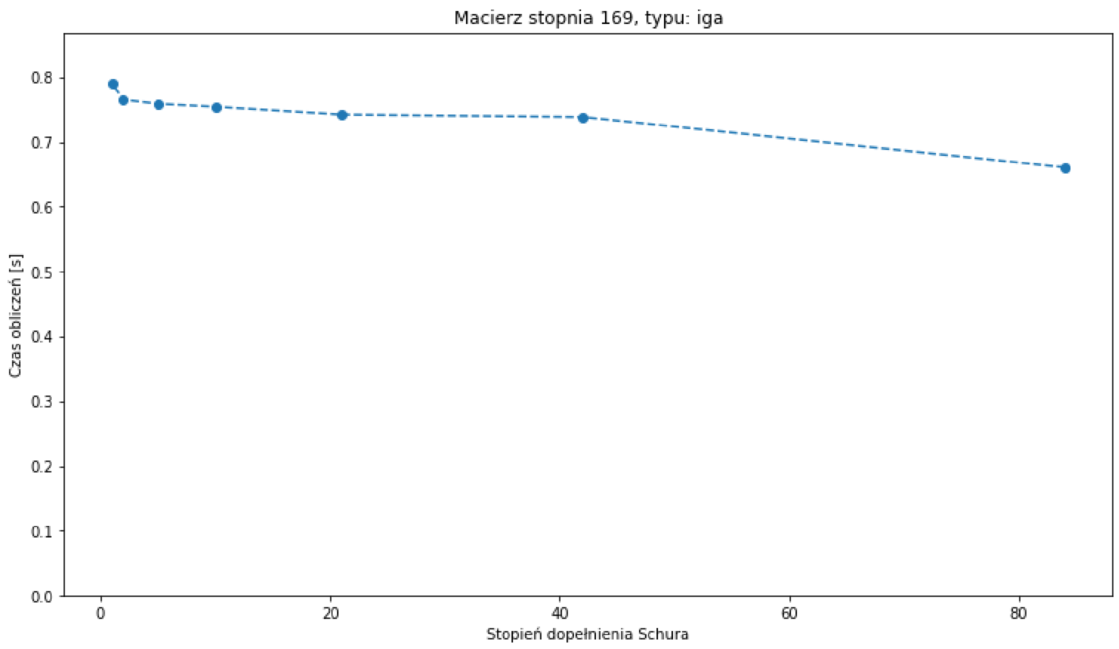

Macierz stopnia 64, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.01
0.02
0.03
0.04
0
5
10
15
20
25
30

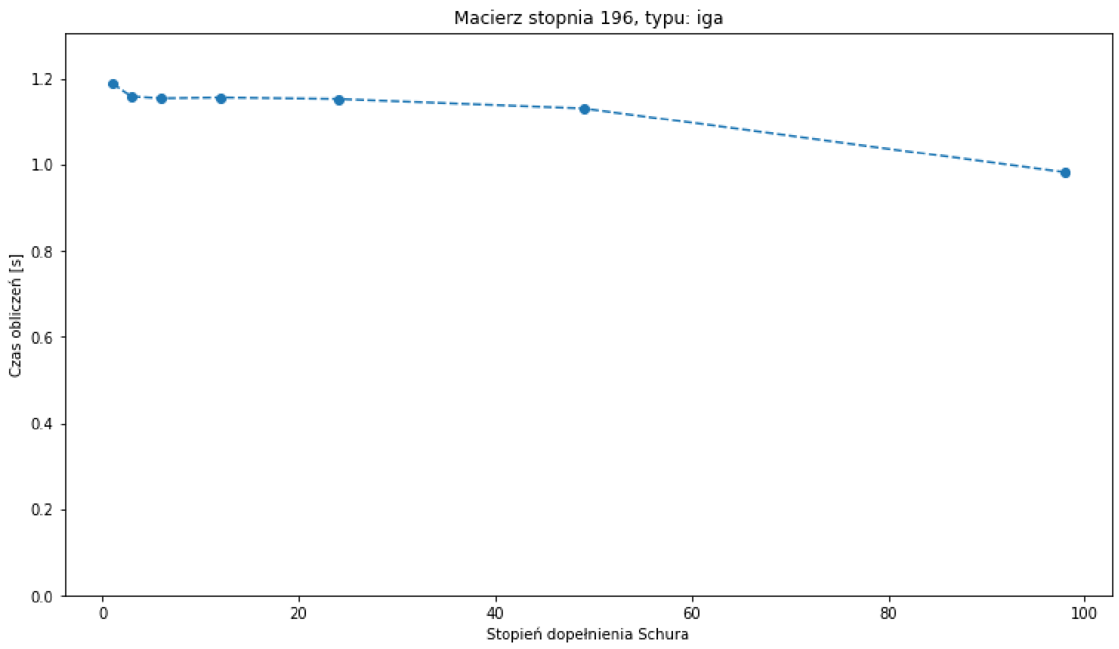

Macierz stopnia 81, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.02
0.04
0.06
0.08
0
5
10
15
20
25
30
35
40

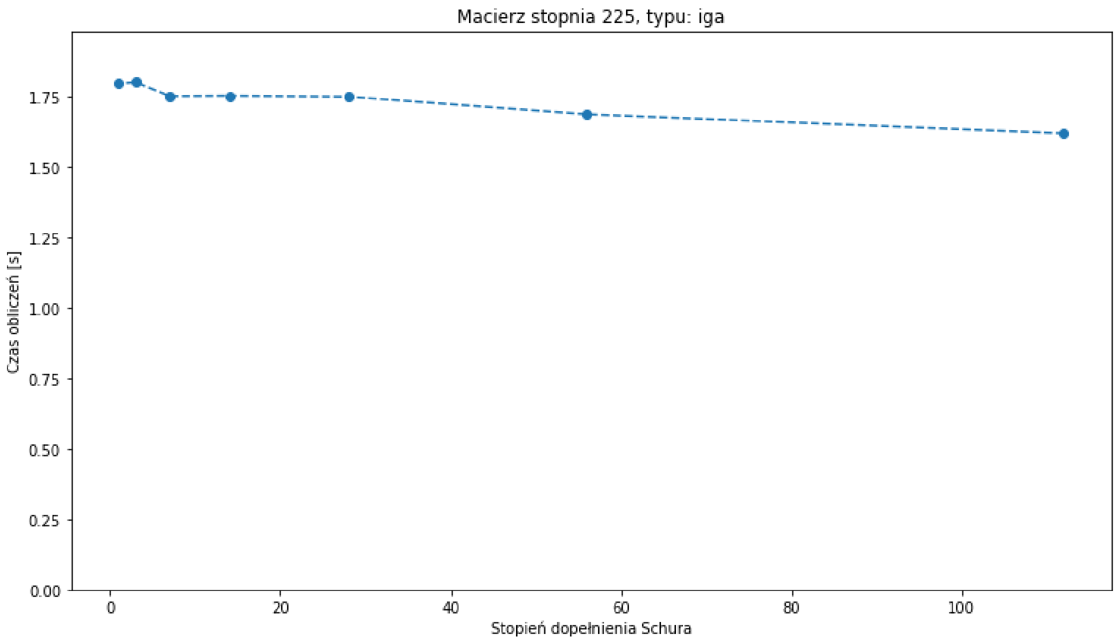

Macierz stopnia 100, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0
10
20
30
40
50

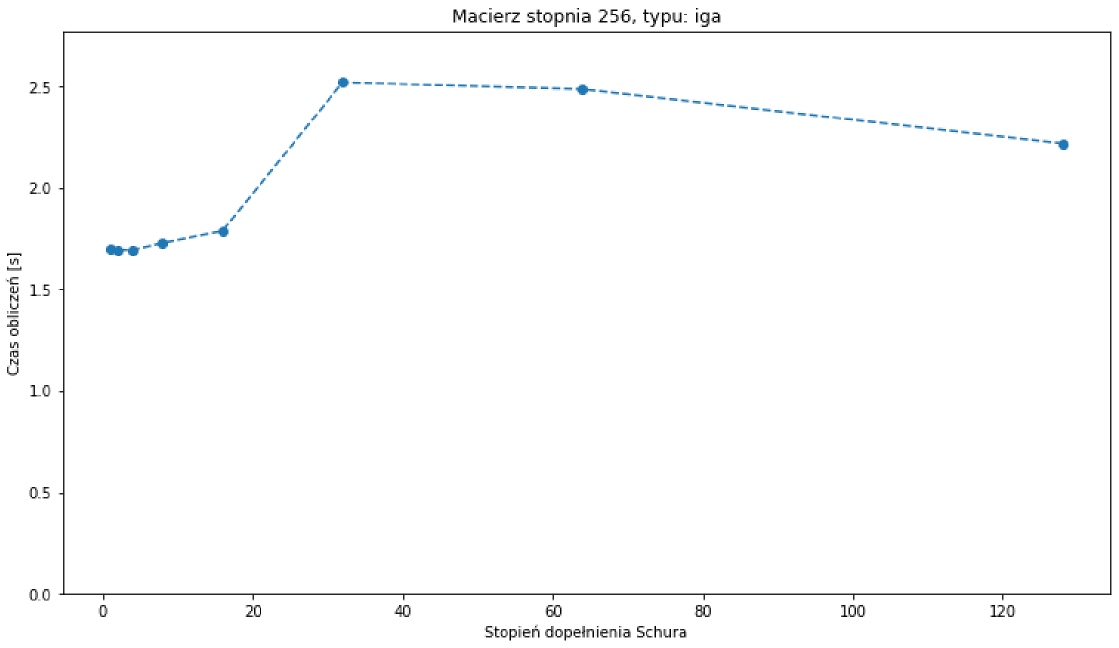

Macierz stopnia 121, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60

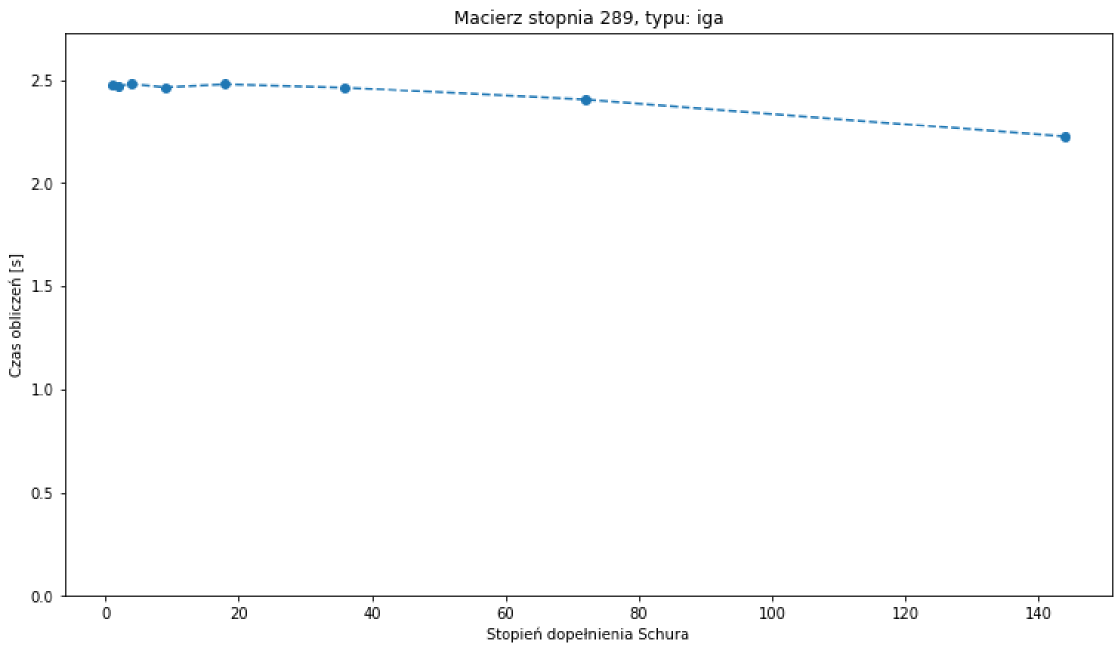

Macierz stopnia 144, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
70


Macierz stopnia 169, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80

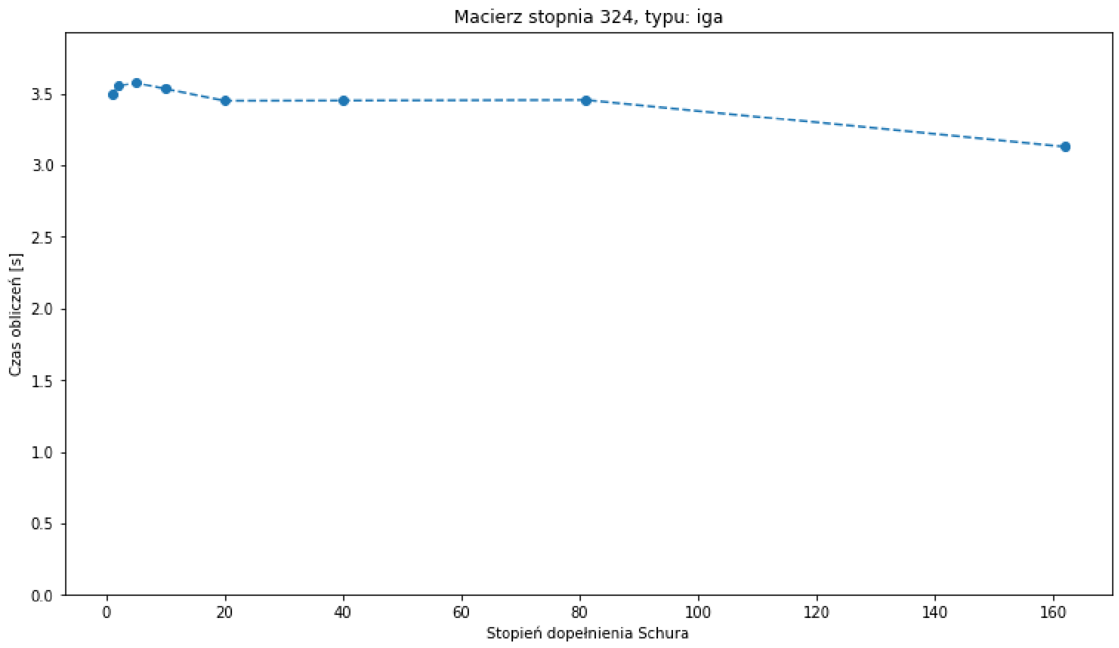

Macierz stopnia 196, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20
40
60
80
100


Macierz stopnia 225, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0
20
40
60
80
100

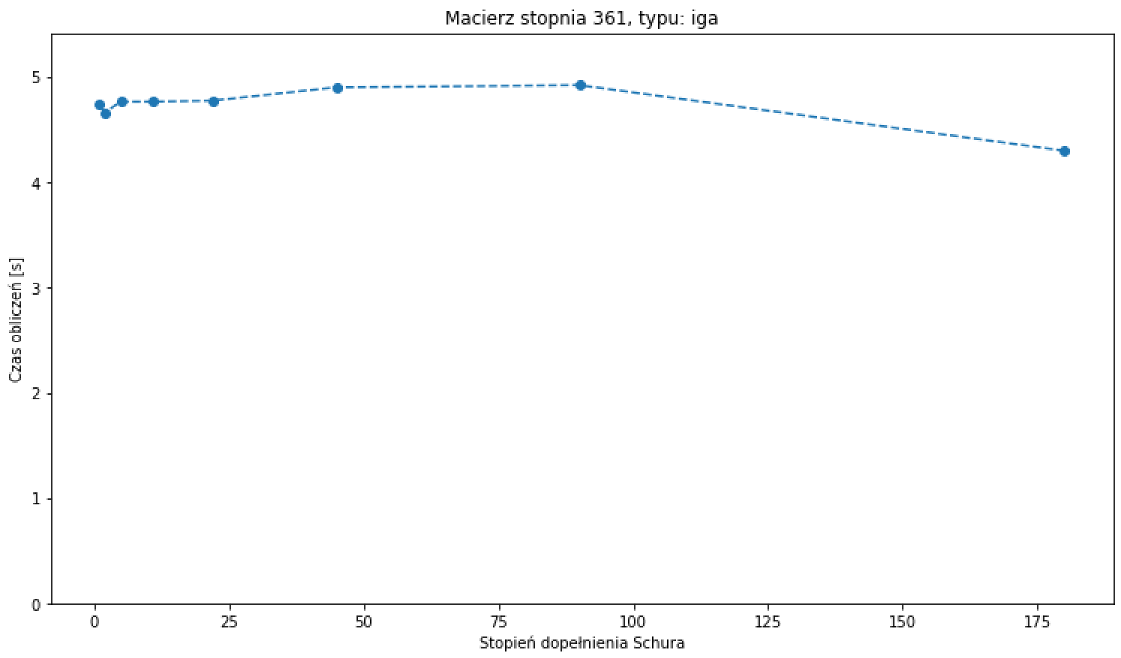

Macierz stopnia 256, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
100
120

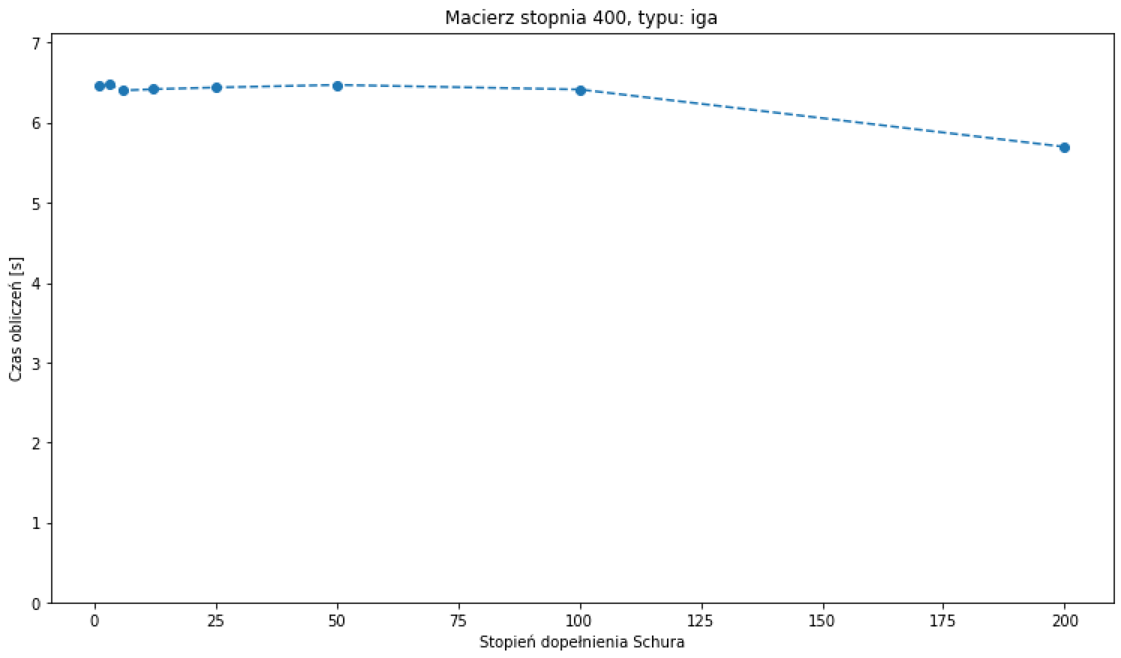


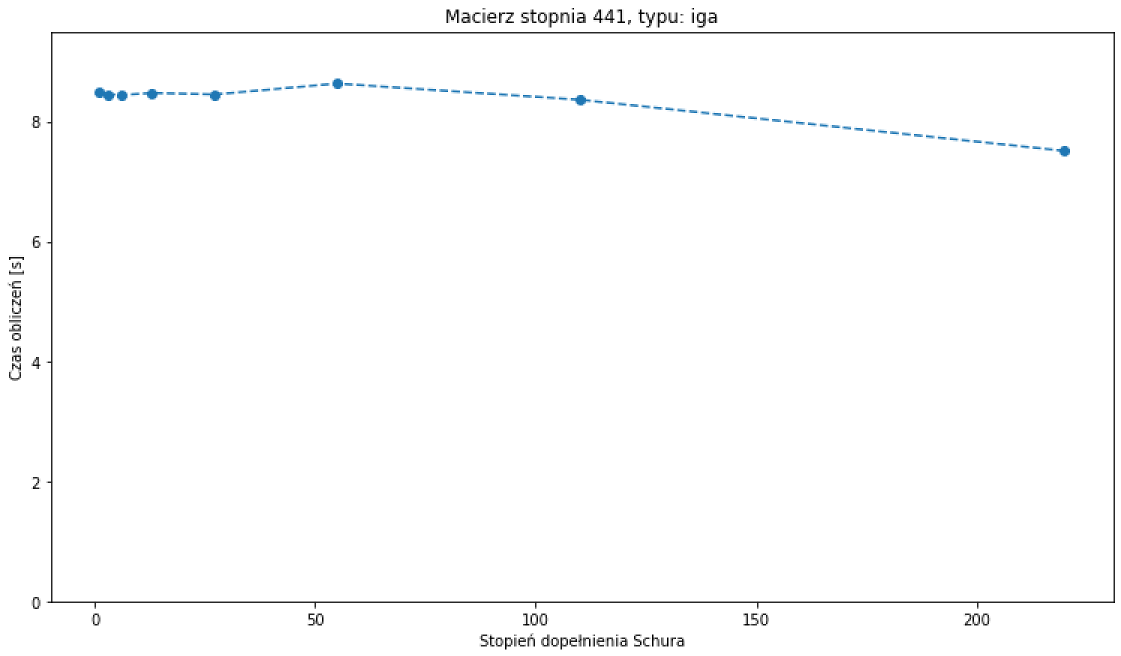
Macierz stopnia 289, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura

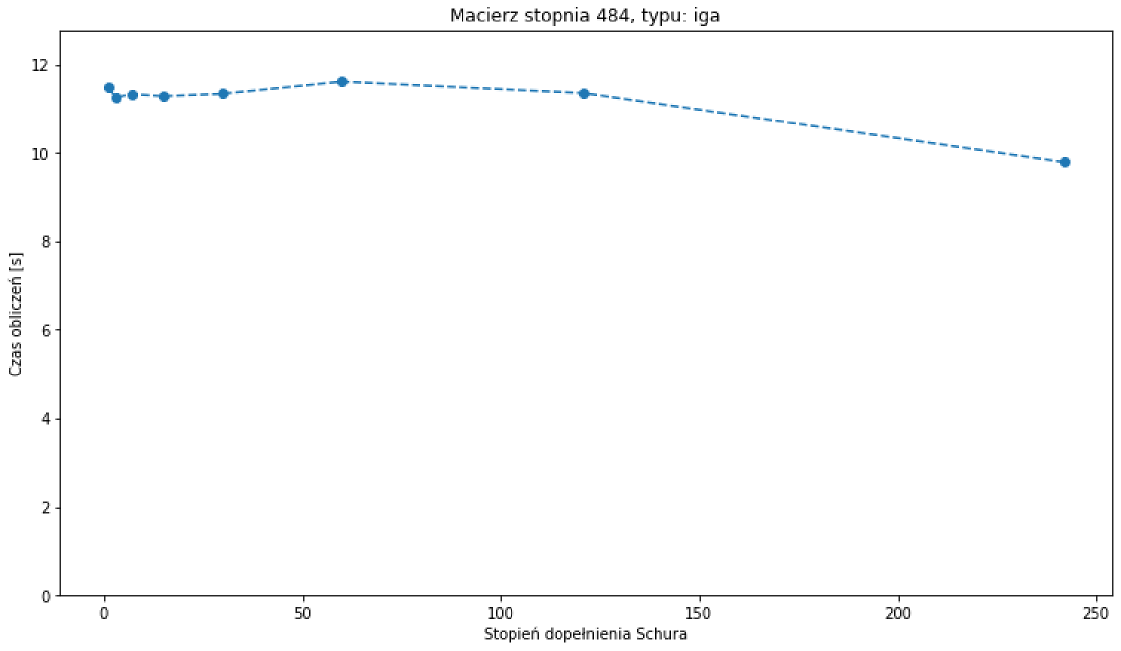


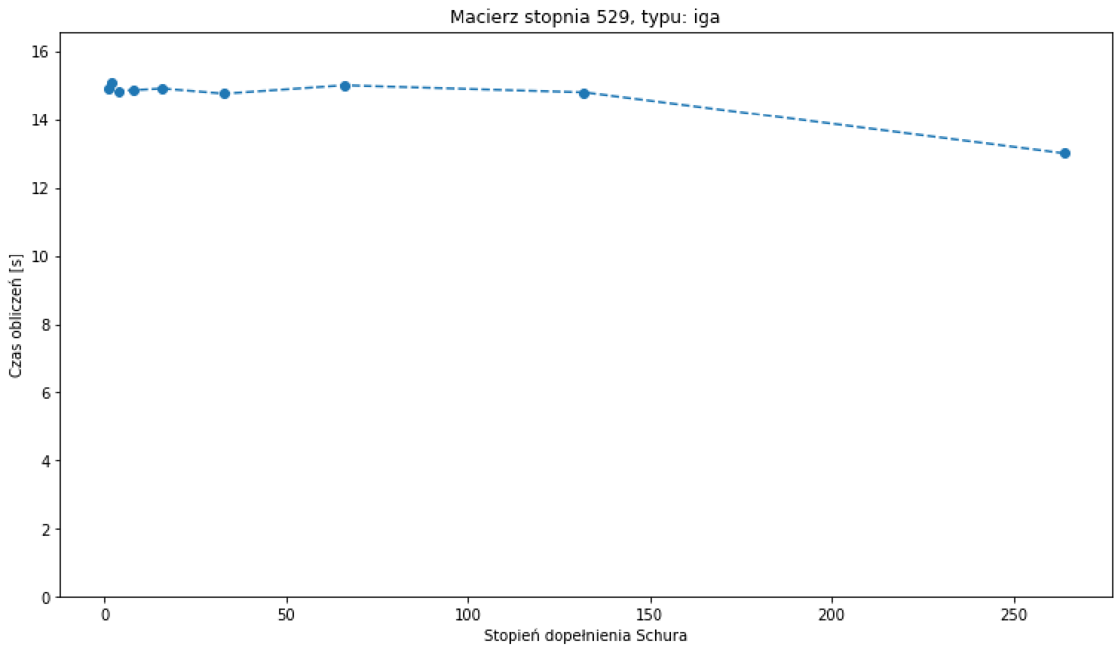
Macierz stopnia 324, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura

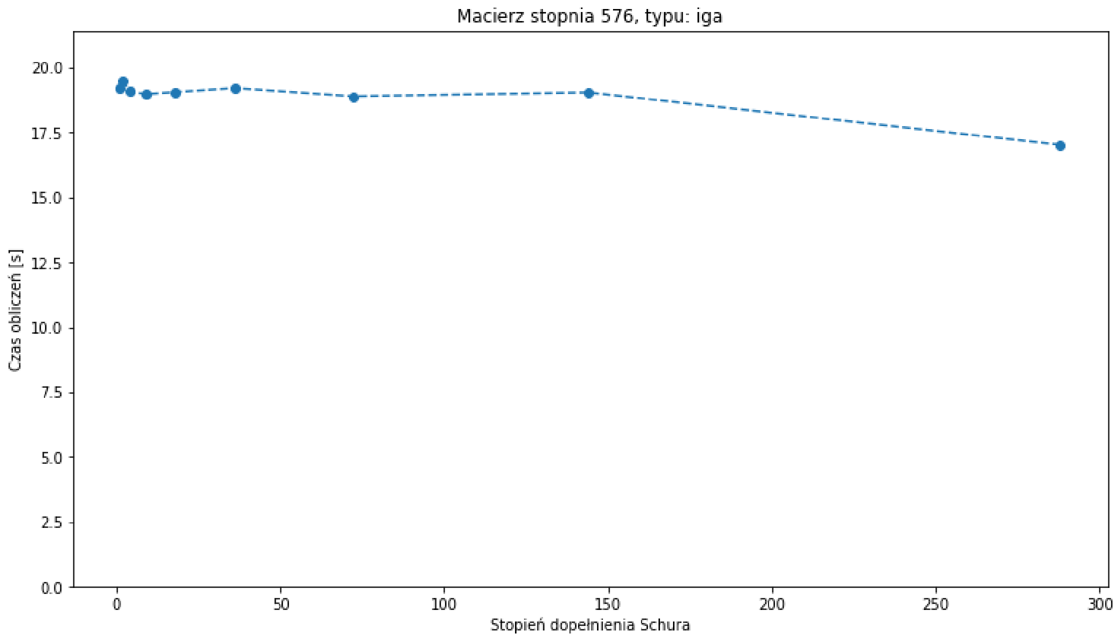

Macierz stopnia 361, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
1
2
3
4
5
0
25
50
75
100
125
150
175

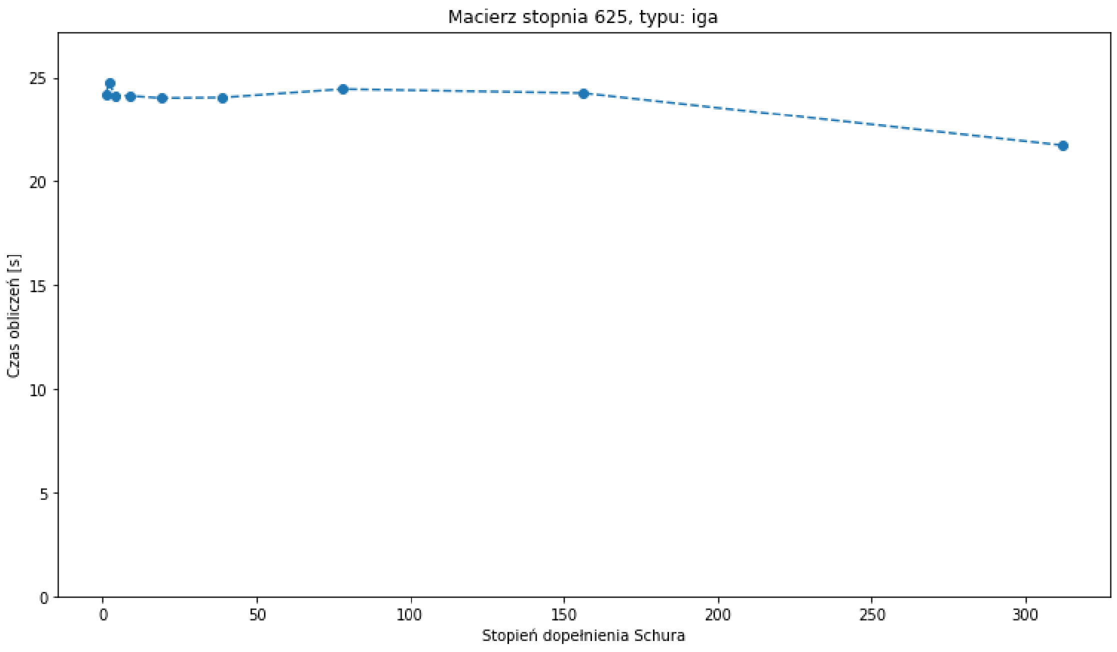

Macierz stopnia 400, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
1
2
3
4
5
6
7
0
25
50
75
100
125
150
175
200


Macierz stopnia 441, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
2
4
6
8
0
50
100
150
200

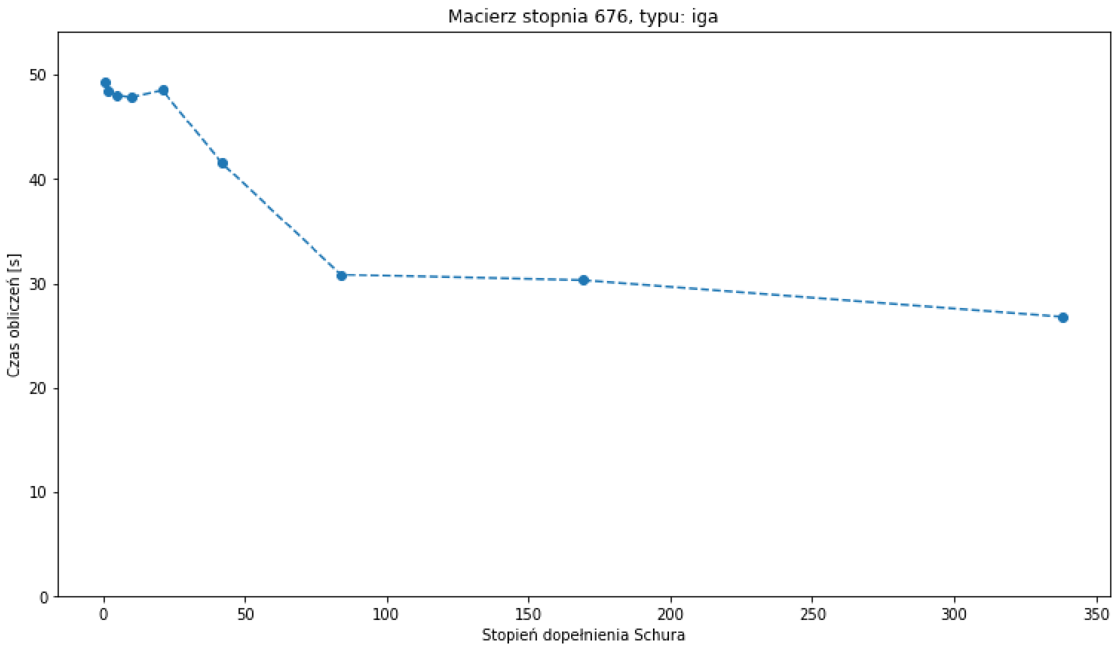

Macierz stopnia 484, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
2
4
6
8
10
12
0
50
100
150
200
250


Macierz stopnia 529, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
2
4
6
8
10
12
14
16
0
50
100
150
200
250

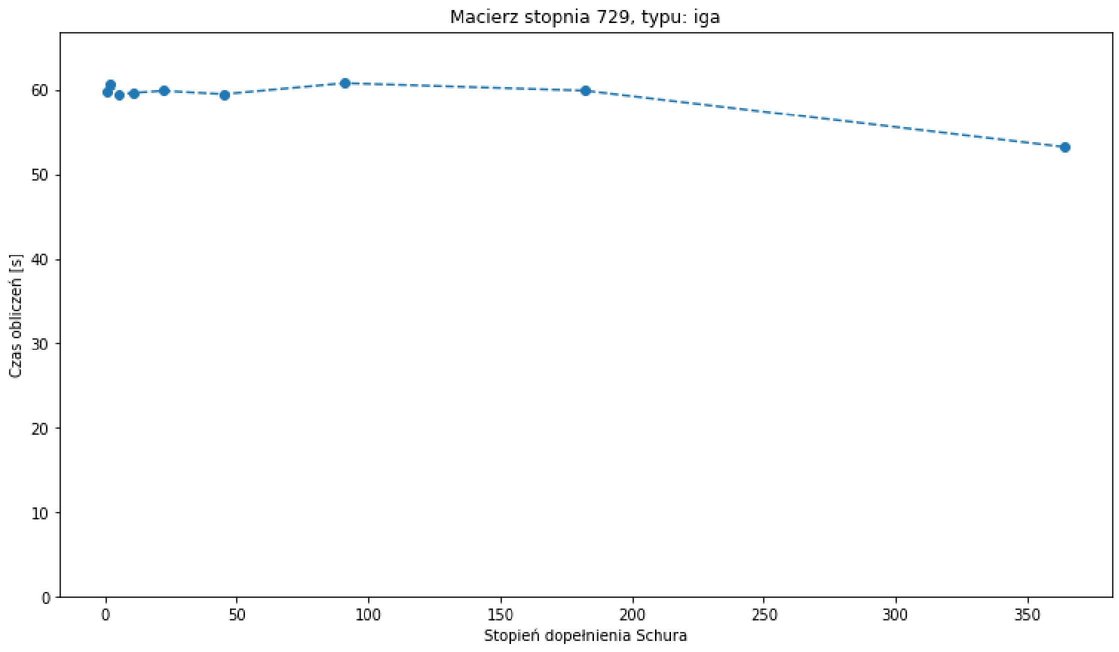

Macierz stopnia 576, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
50
100
150
200
250
300

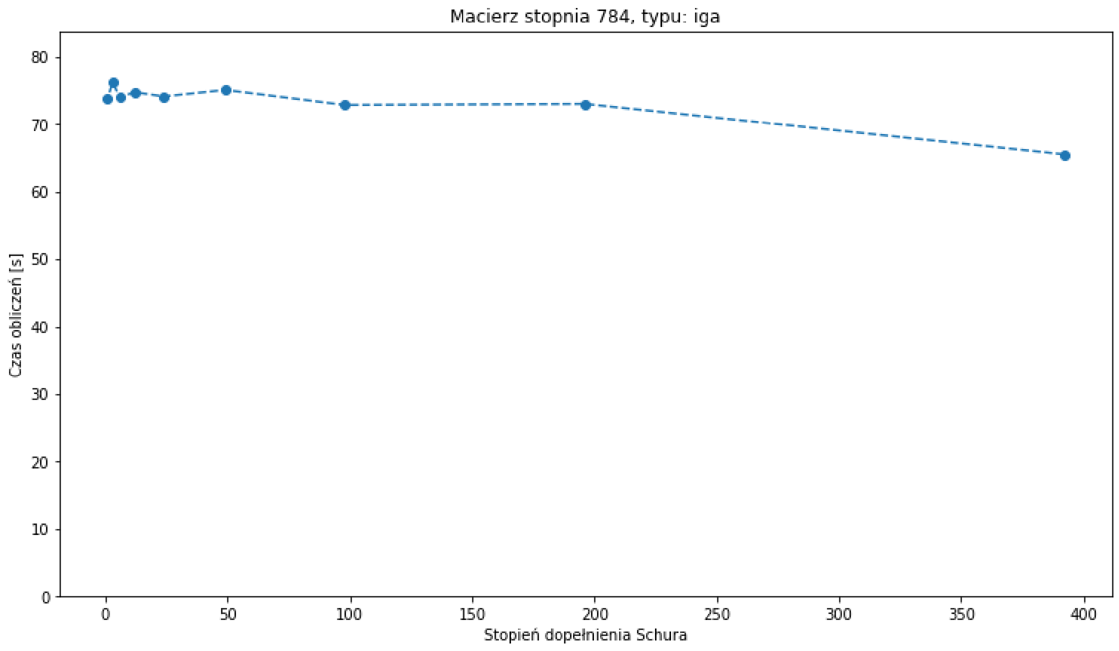


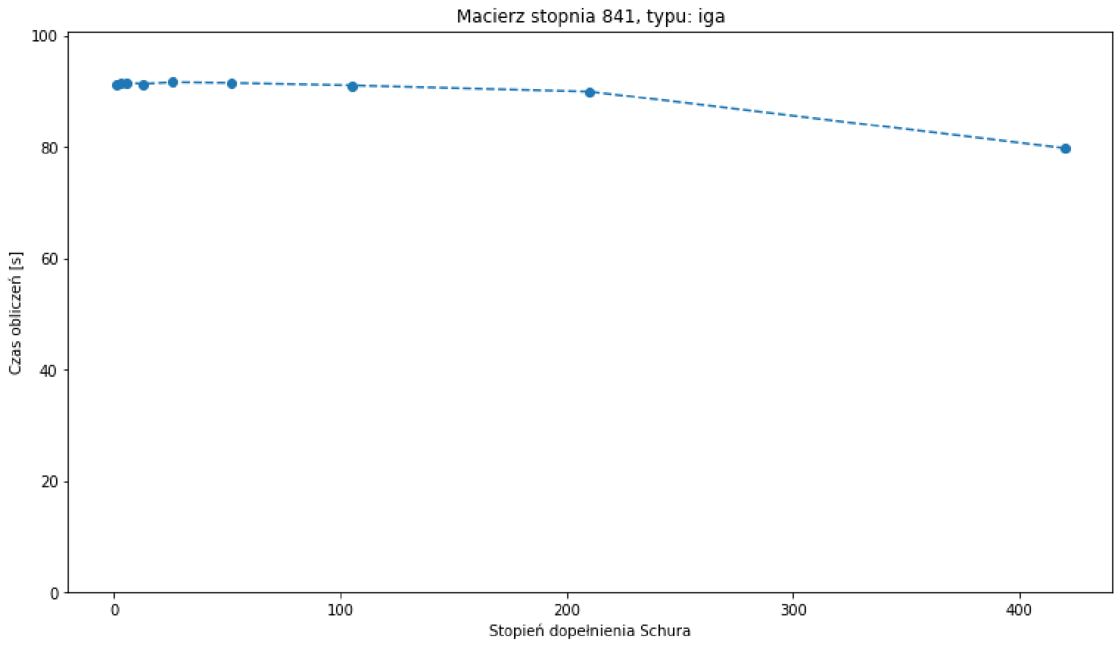
Macierz stopnia 625, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura

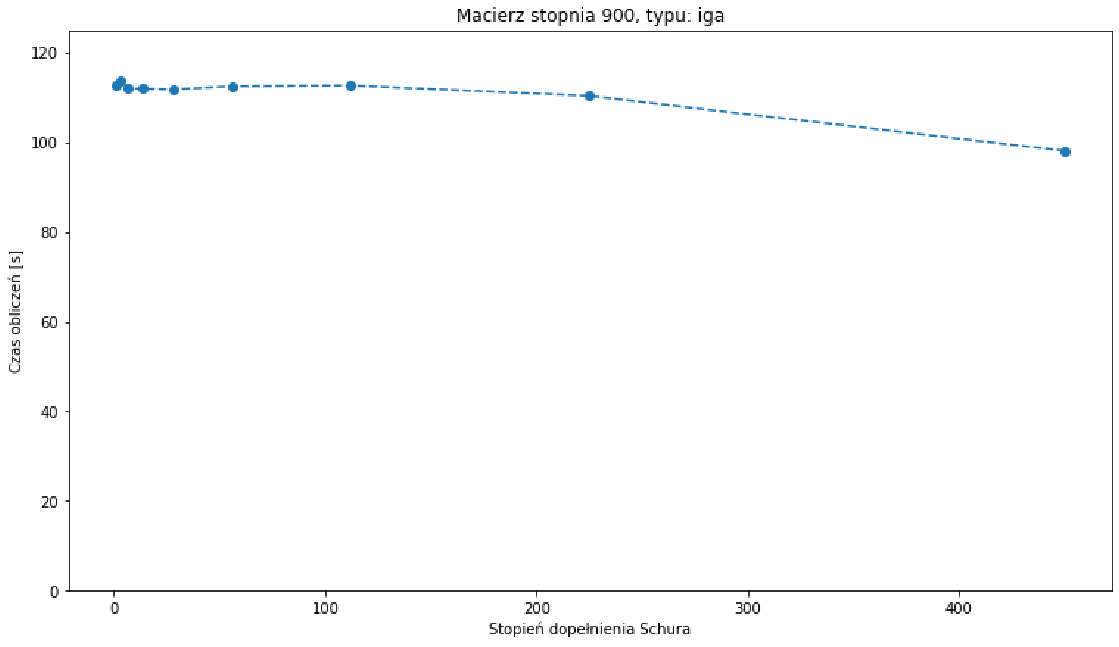


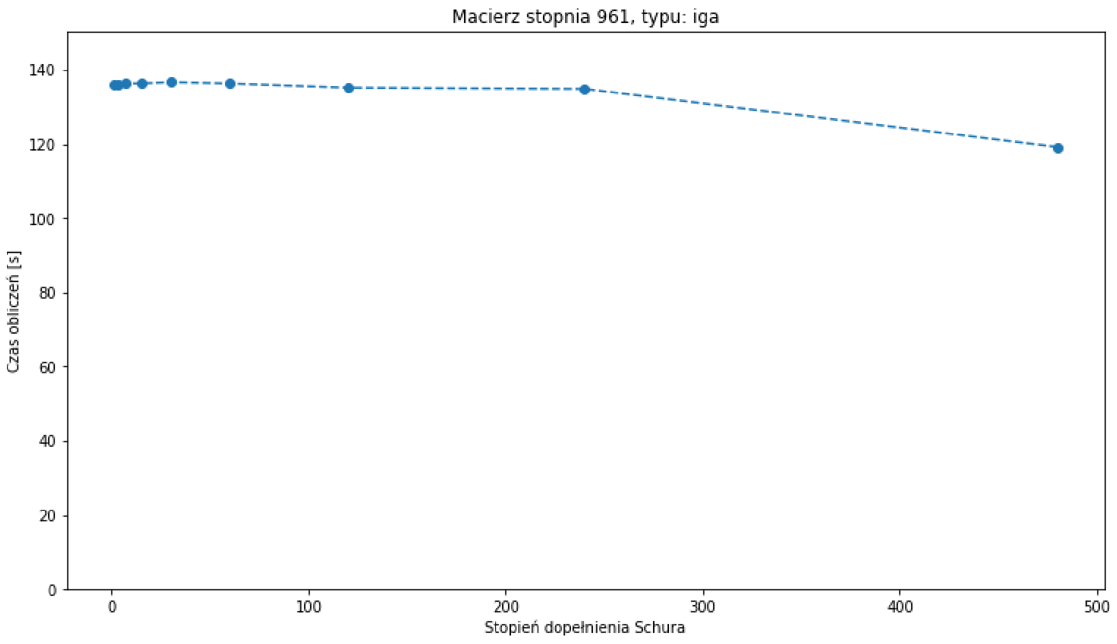
Macierz stopnia 676, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura

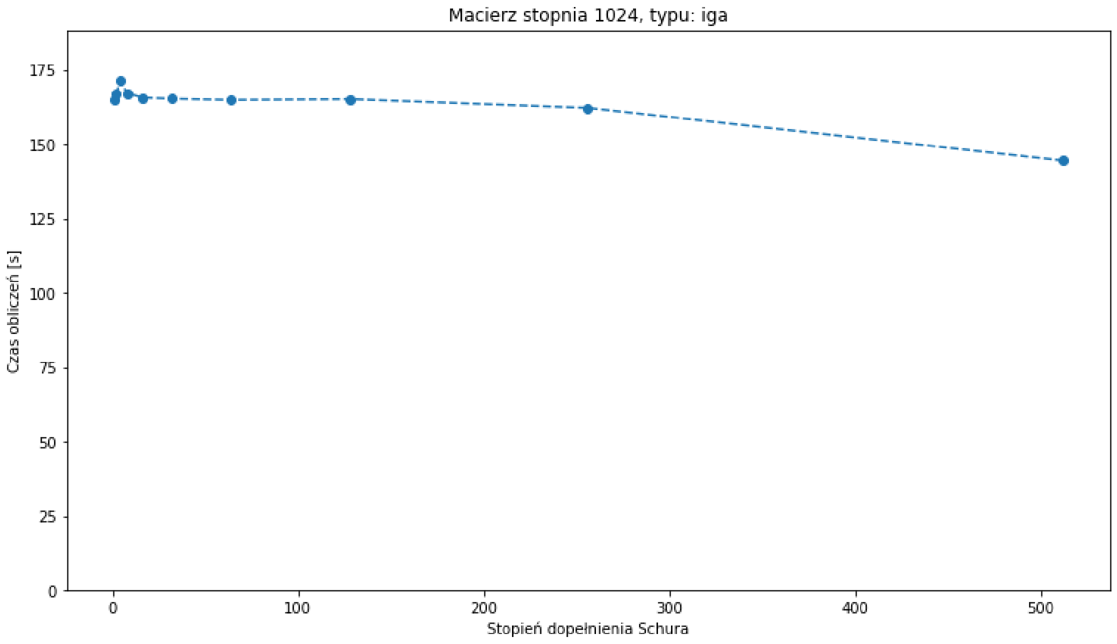

Macierz stopnia 729, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
10
20
30
40
50
60
0
50
100
150
200
250
300
350

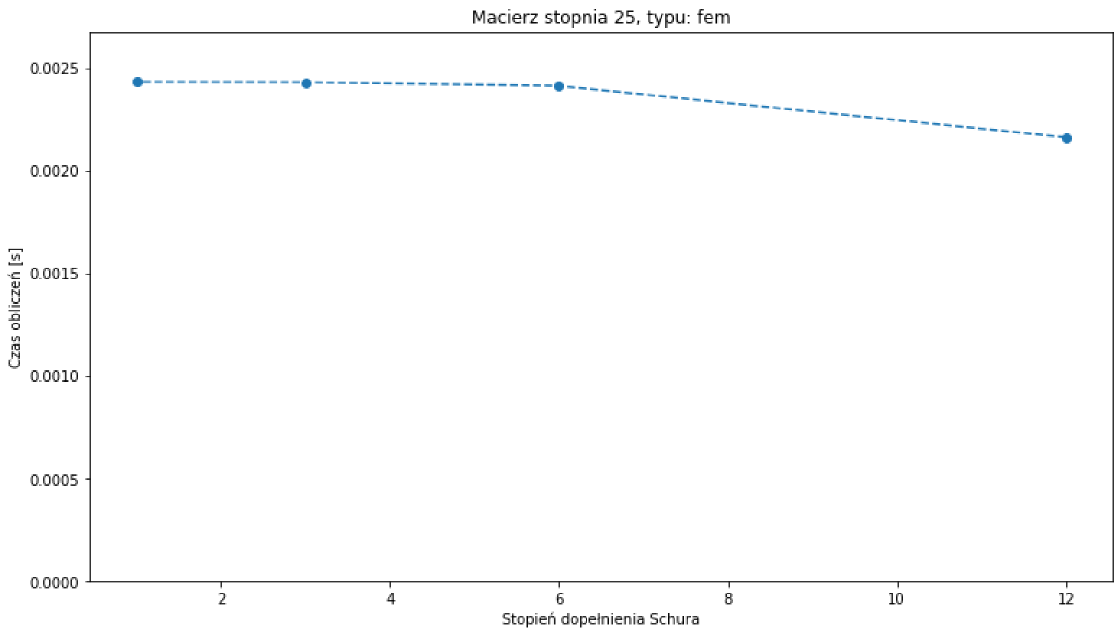

Macierz stopnia 784, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
10
20
30
40
50
60
70
80
0
50
100
150
200
250
300
350
400

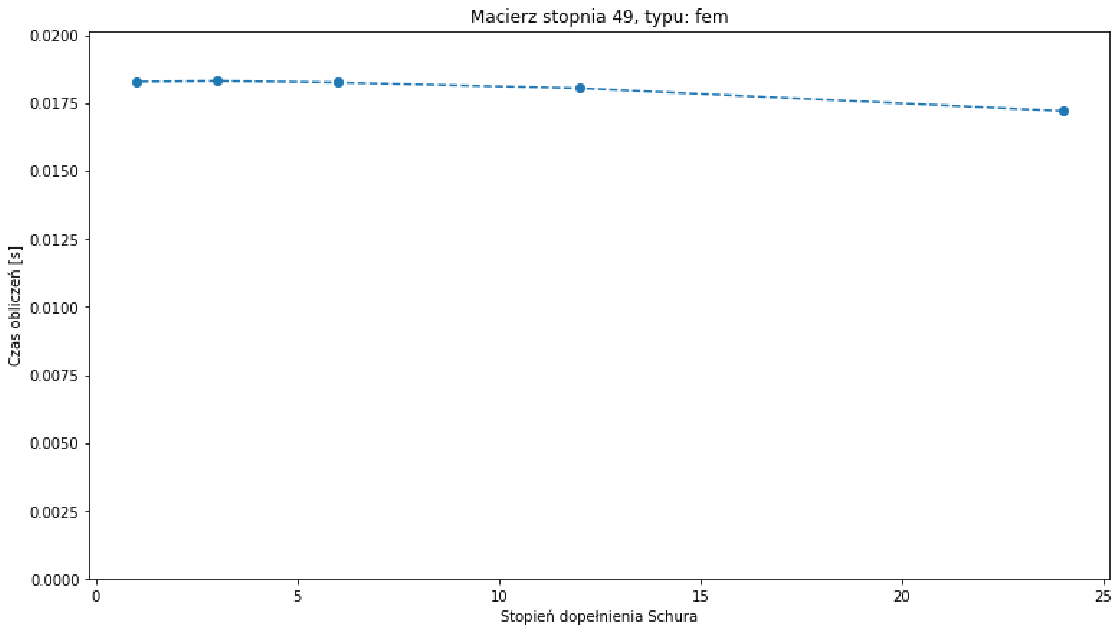

Macierz stopnia 841, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
20
40
60
80
100
0
100
200
300
400

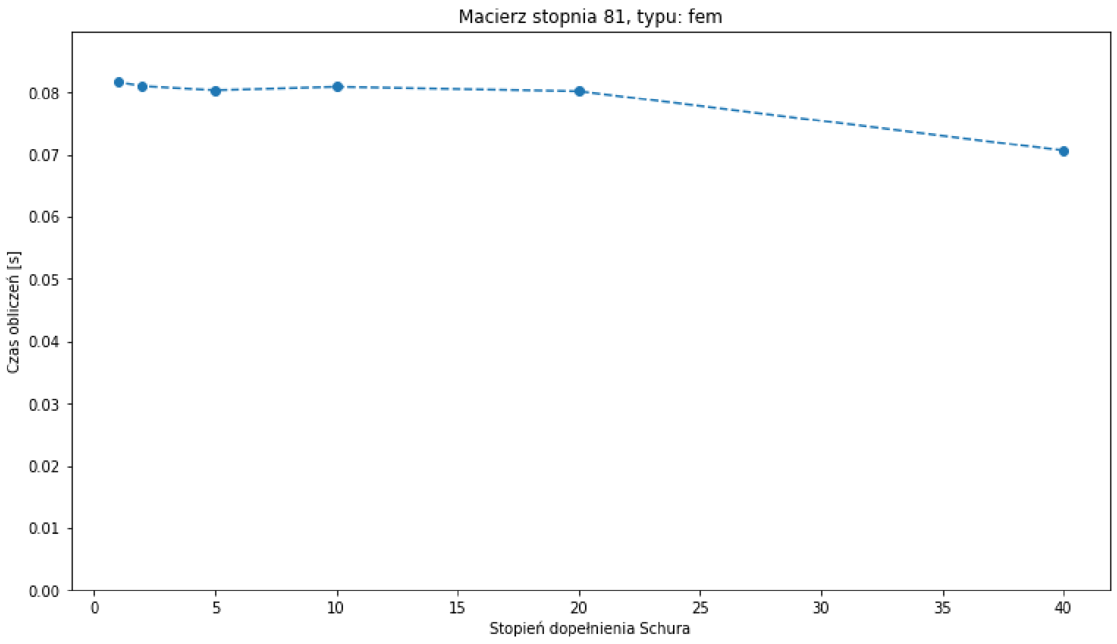

Macierz stopnia 900, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
20
40
60
80
100
120
0
100
200
300
400

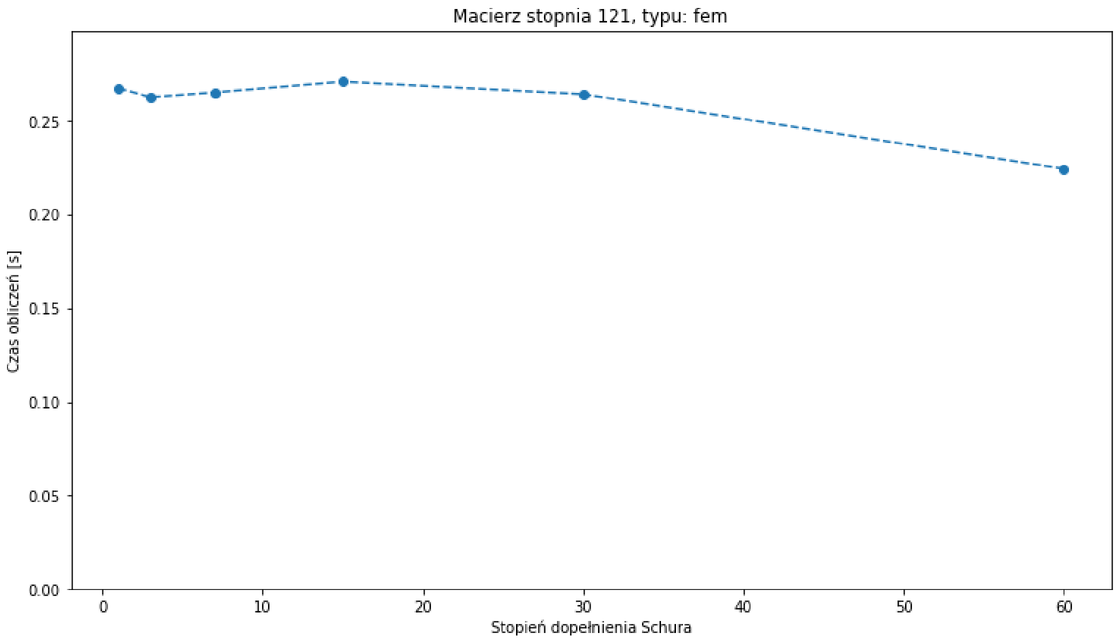

Macierz stopnia 961, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
20
40
60
80
100
120
140
0
100
200
300
400
500

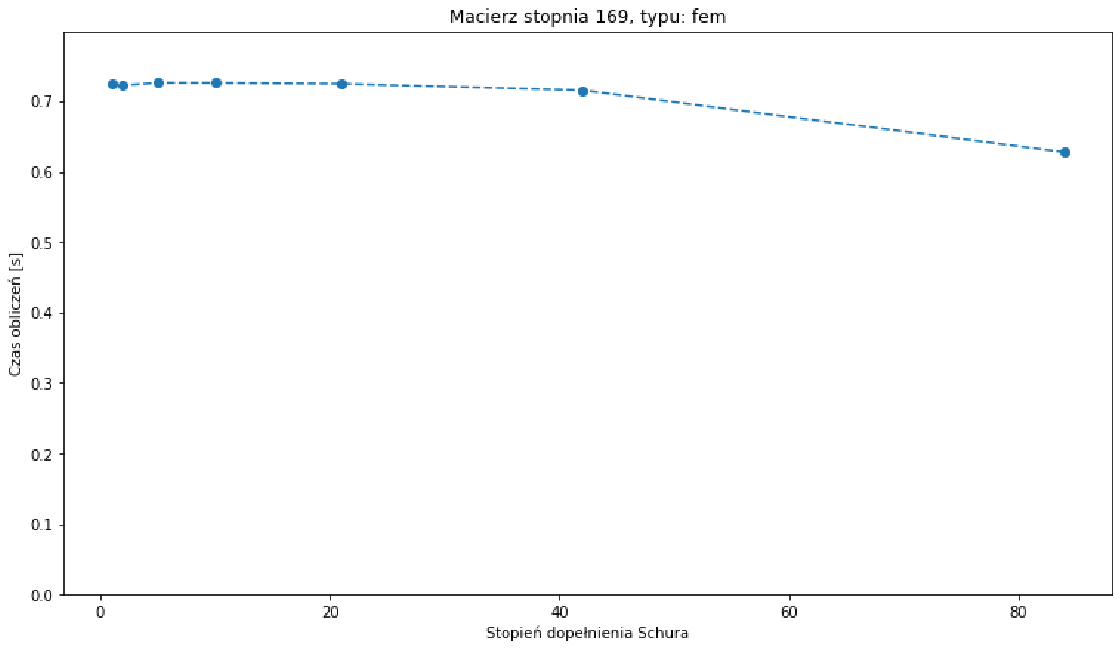

Macierz stopnia 1024, typu: iga
Czas obliczeń [s]
Stopień dopełnienia Schura
0
25
50
75
100
125
150
175
0
100
200
300
400
500

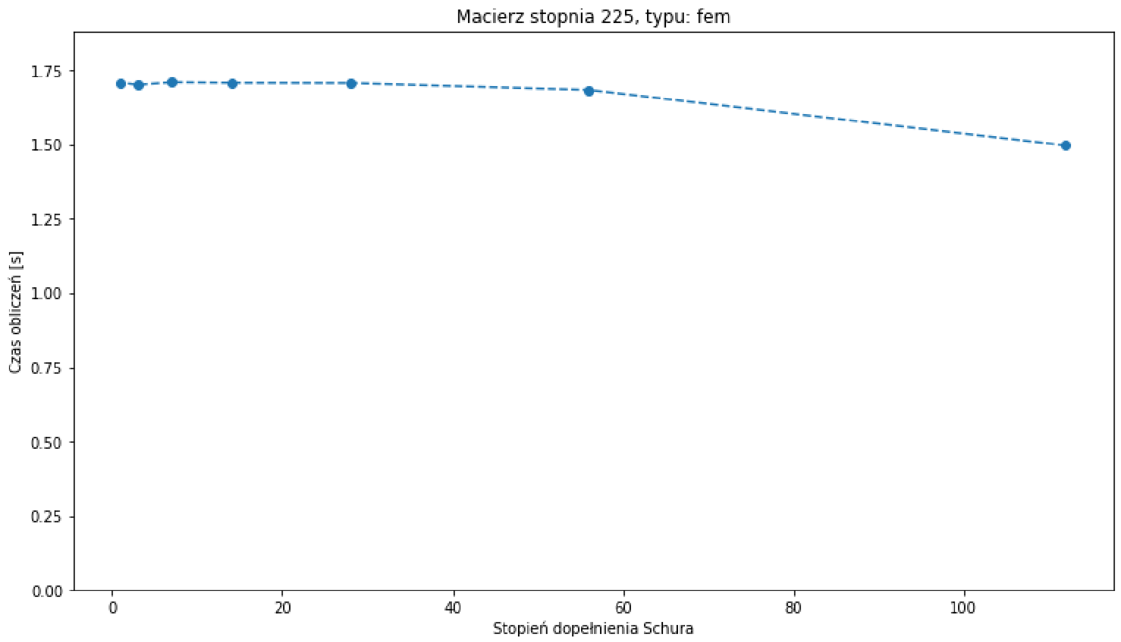

Macierz stopnia 25, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
2
4
6
8
10
12

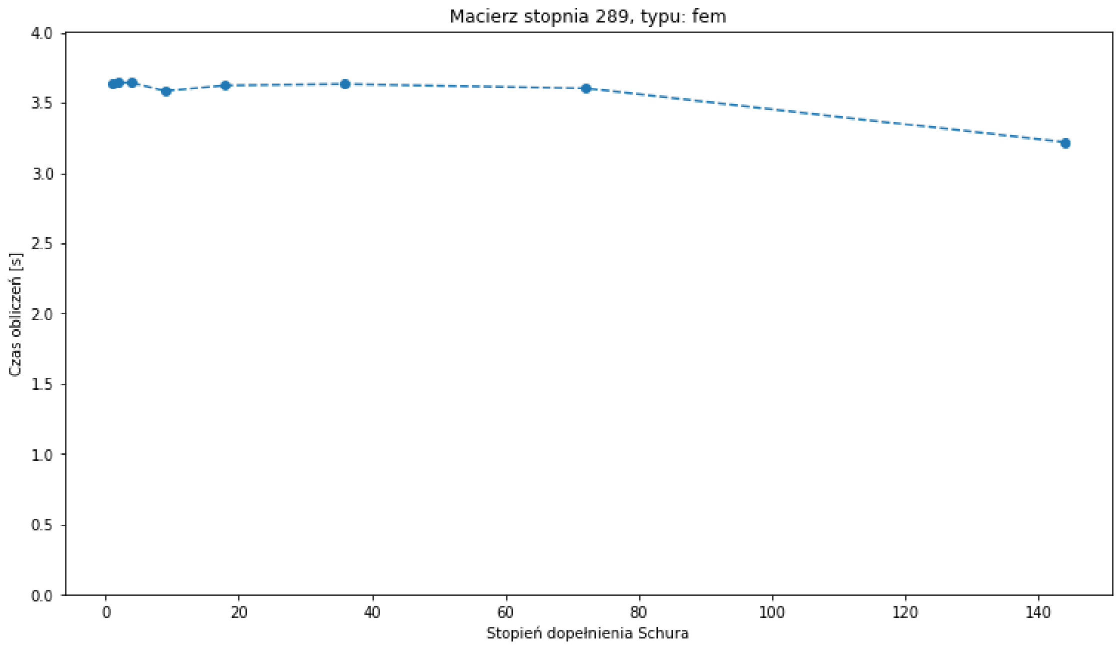

Macierz stopnia 49, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
0
5
10
15
20
25

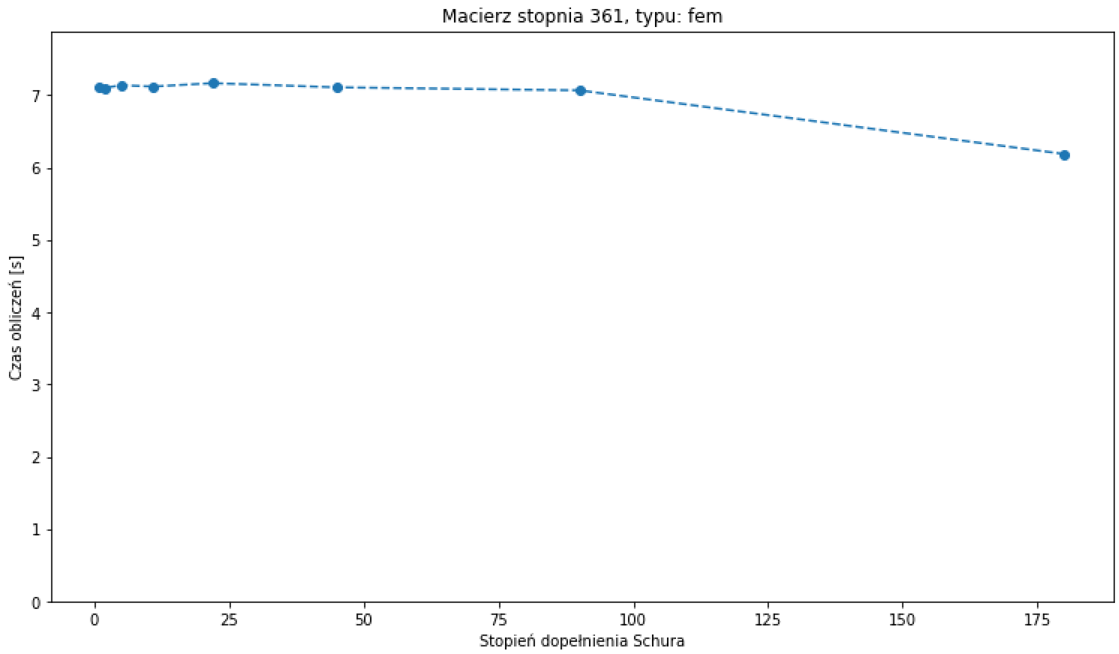

Macierz stopnia 81, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0
5
10
15
20
25
30
35
40

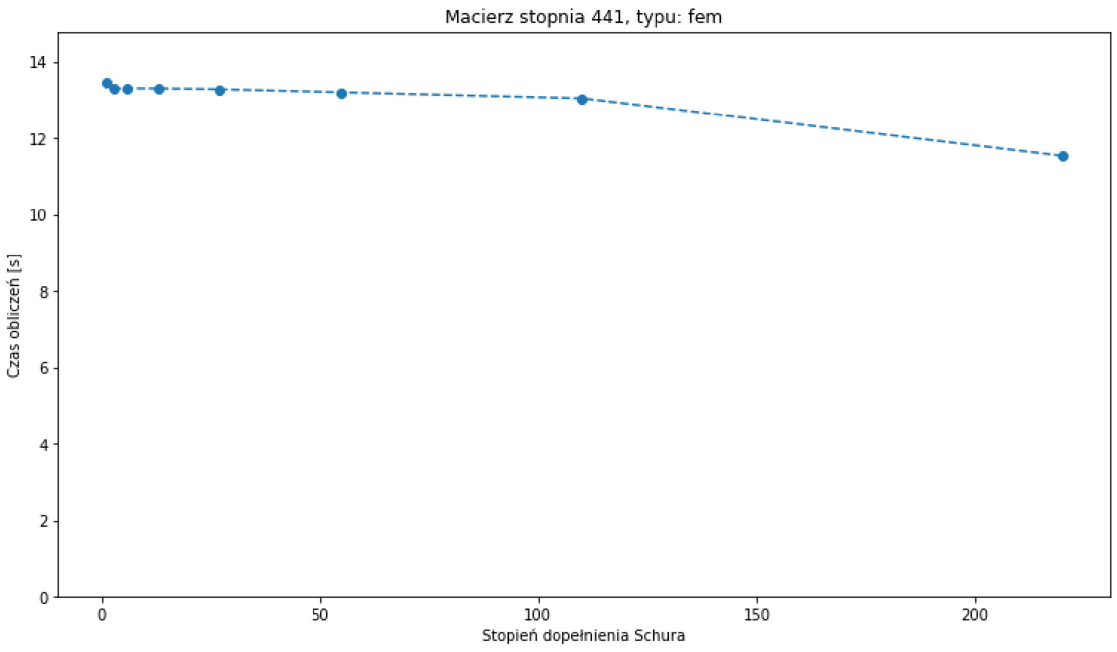

Macierz stopnia 121, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.05
0.10
0.15
0.20
0.25
0
10
20
30
40
50
60

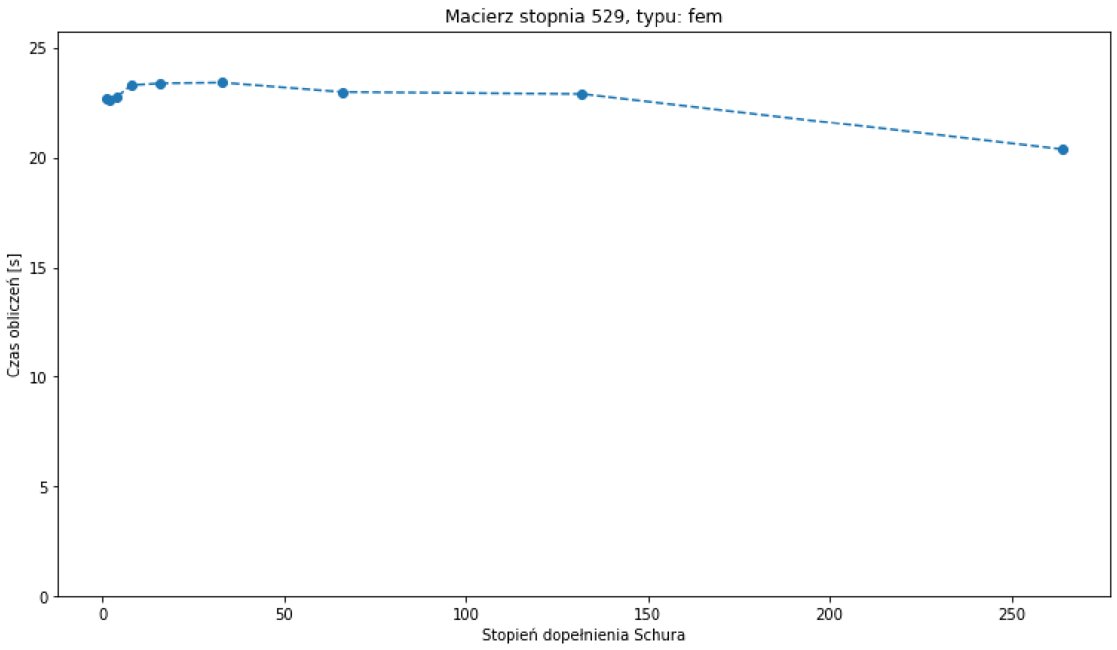

Macierz stopnia 169, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80

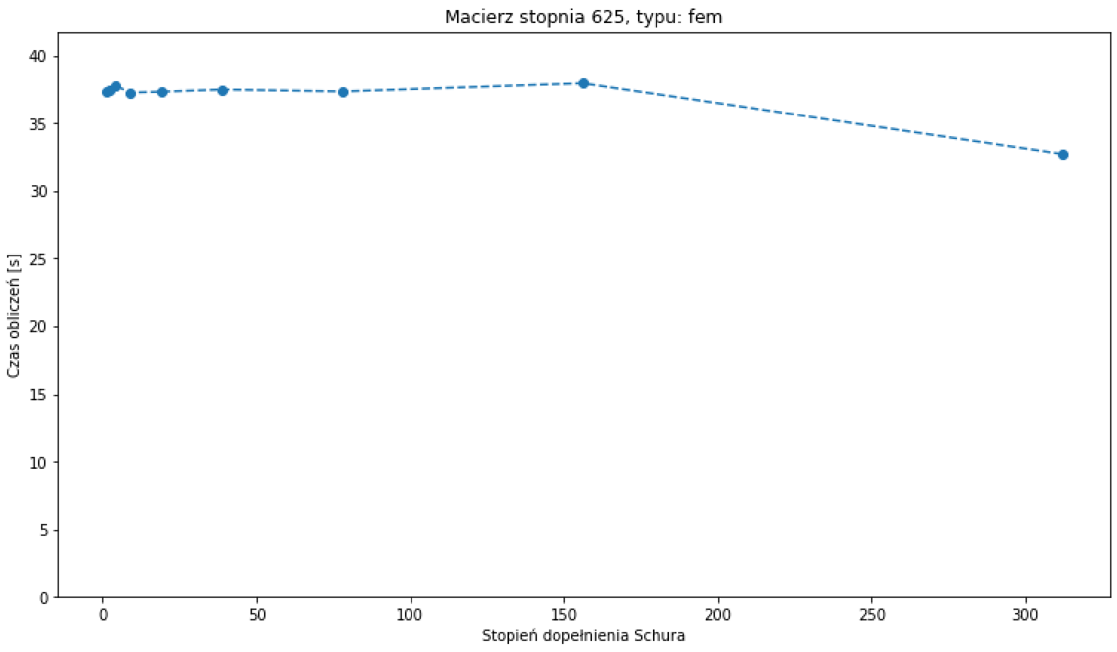

Macierz stopnia 225, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0
20
40
60
80
100

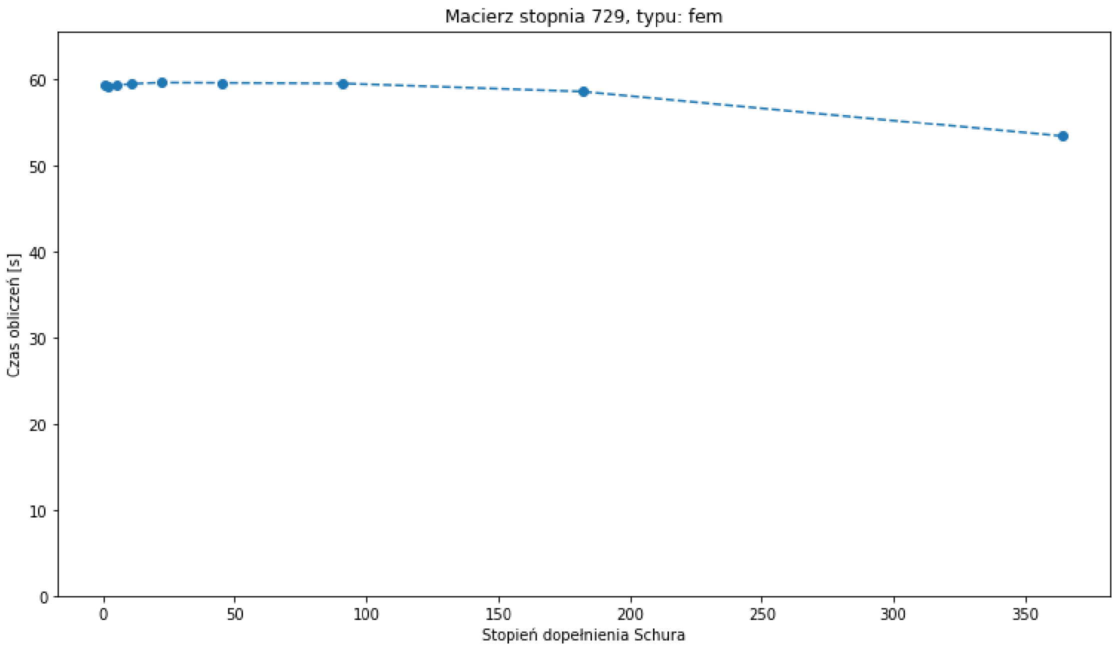

Macierz stopnia 289, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
120
140

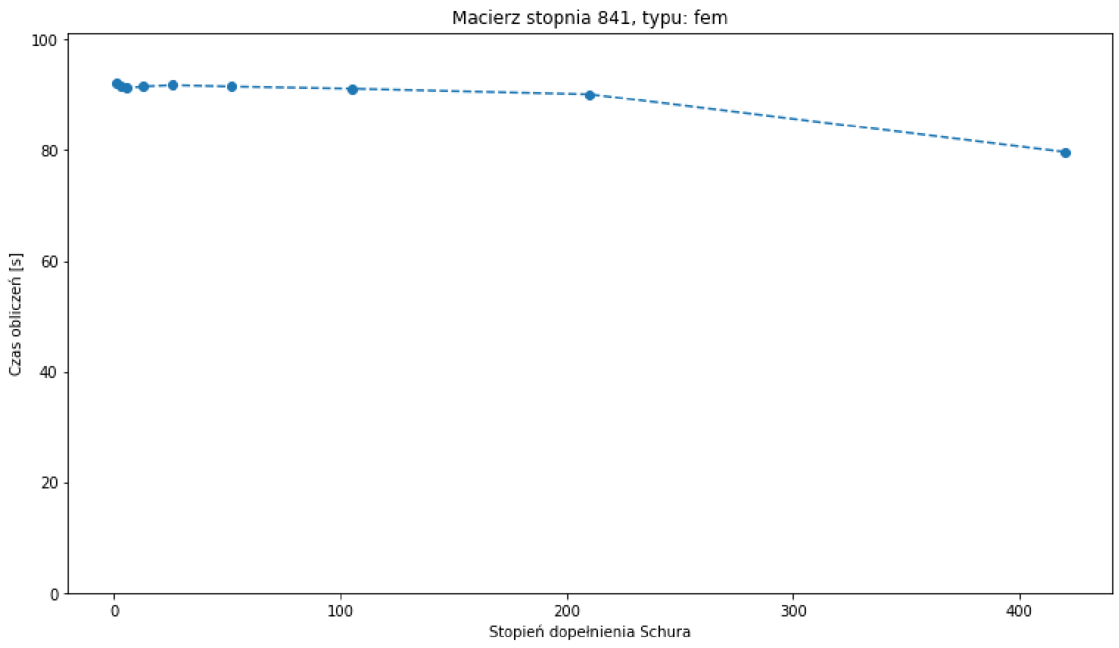

Macierz stopnia 361, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
7
6
5
4
3
2
1
0
0
25
50
75
100
125
150
175

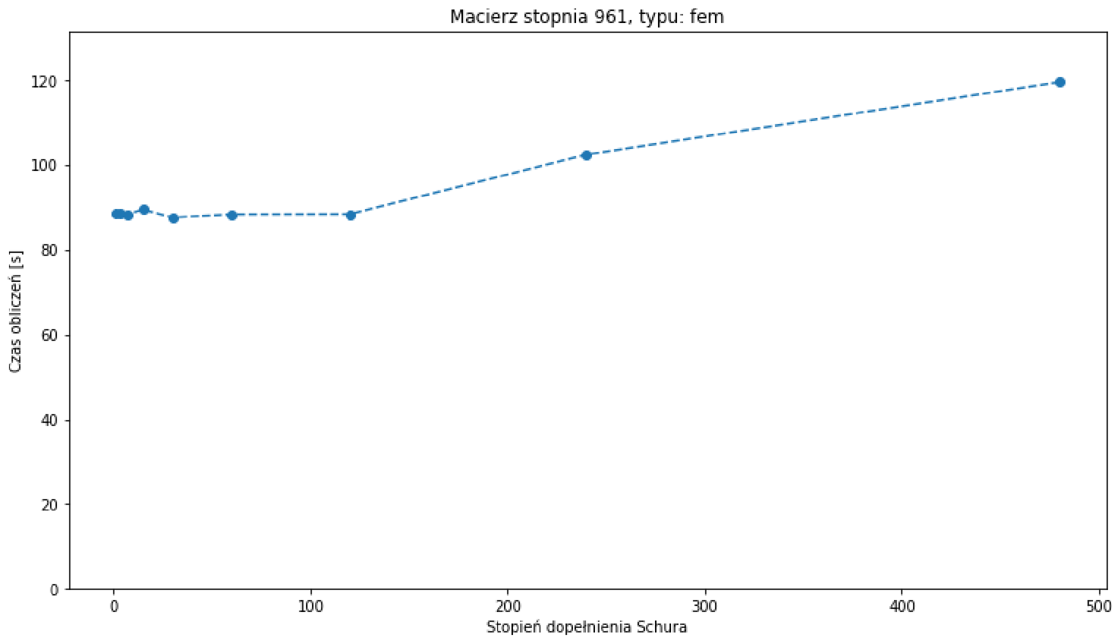

Macierz stopnia 441, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
2
4
6
8
10
12
14
0
50
100
150
200

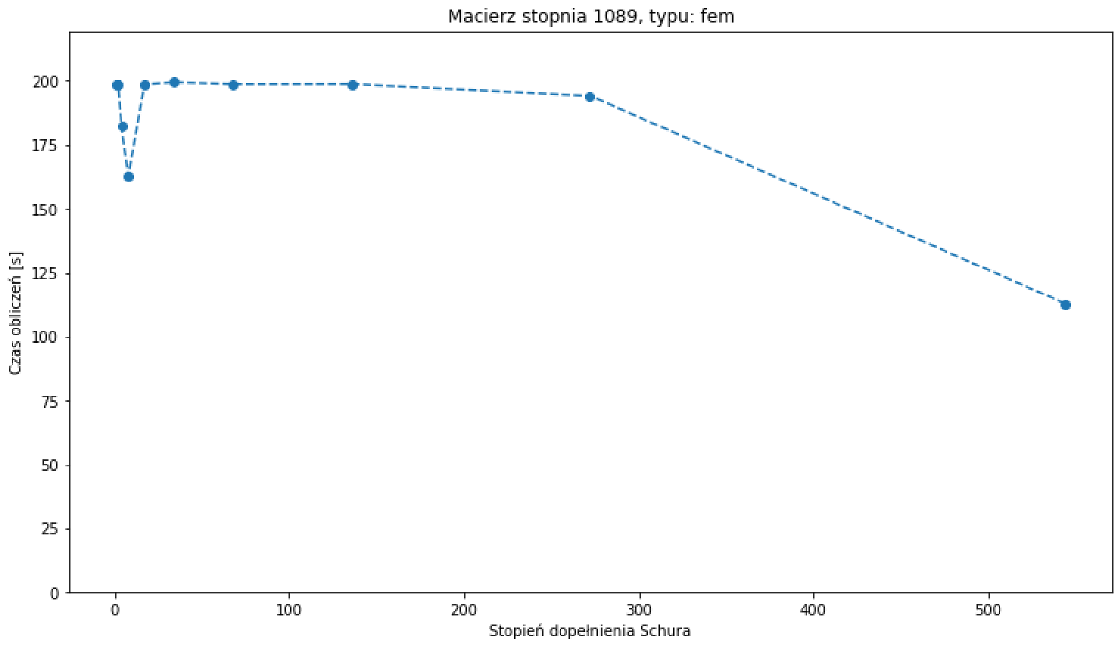

Macierz stopnia 529, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
5
10
15
20
25
0
50
100
150
200
250


Macierz stopnia 625, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
5
10
15
20
25
30
35
40
0
50
100
150
200
250
300


Macierz stopnia 729, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
10
20
30
40
50
60
0
50
100
150
200
250
300
350


Macierz stopnia 841, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
20
40
60
80
100
0
100
200
300
400


Macierz stopnia 961, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
20
40
60
80
100
120
0
100
200
300
400
500


Macierz stopnia 1089, typu: fem
Czas obliczeń [s]
Stopień dopełnienia Schura
0
25
50
75
100
125
150
175
200
0
100
200
300
400
500

```
[ ]:
```