

Wstęp

Istnieją dwa podstawowe sposoby rozwiązywania układów równań liniowych postaci $\mathbf{Ax}=\mathbf{b}$ – metody bezpośrednie, które badane były podczas poprzednich zajęć, oraz metody iteracyjne. W metodach bezpośrednich, będących w praktyce najczęściej wariantami eliminacji Gaussa, po wykonaniu pewnej liczby operacji otrzymuje się rozwiązanie o dużej dokładności (błędy pojawiają się tylko na skutek skończonej dokładności obliczeń komputerowych i propagacji błędów zaokrągleń – dlatego obliczenia zawsze są przeprowadzane na liczbach podwójnej precyzji, co pozwala uzyskać wyniki z dokładnością co najmniej kilku cyfr znaczących).

Metody iteracyjne uzyskują rozwiązanie w szeregu kroków obliczeniowych (iteracji), zaczynając od pewnego wektora startowego $\mathbf{x}_0$ (*initial guess*) i uzyskując po każdym z kroków kolejne przybliżenia rozwiązania. Miarą zbieżności dla kolejnych rozwiązań $\mathbf{x}_k$ jest norma z wektora residuum układu równań, obliczanego dla $\mathbf{x}_k$ jako: $\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}_k$. Zakłada się, że kiedy residuum układu równań dla kolejnych iteracji zmierza do 0, to także błąd rozwiązania zmierza do zera (jeśli rozwiązanie dokładne oznaczymy jako $\mathbf{x}^*$, a błąd rozwiązania $\mathbf{x}_k$ jako $\mathbf{e}_k = \mathbf{x}^* - \mathbf{x}_k$, to dzięki temu, że $\mathbf{Ax}^* = \mathbf{b}$, otrzymuje się liniową zależność wiążącą błąd z residuum: $\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}_k = \mathbf{Ax}^* - \mathbf{Ax}_k = \mathbf{A}(\mathbf{x}^* - \mathbf{x}_k) = \mathbf{Ae}_k$).

Znając normę residuum, nie znamy normy błędu, jednak z liniowej zależności między oboma wektorami możemy wnioskować, że kiedy norma residuum maleje wielokrotnie, to także norma błędu maleje w podobny sposób. Stąd częstym kryterium zatrzymania iteracji jest albo osiągnięcie przez normę residuum $\|\mathbf{r}_k\|$ określonej małej wartości $\|\mathbf{r}_k\| < \mathbf{e}_{\text{ABS}}$ (absolutna redukcja residuum), albo redukcja o określony czynnik początkowego residuum $\mathbf{r}_0 = \mathbf{Ax}_0 - \mathbf{b}$, $\|\mathbf{r}_k\| / \|\mathbf{r}_0\| < \mathbf{e}_{\text{REL}}$ (względna redukcja residuum).

Najpopularniejszymi obecnie metodami iteracyjnego rozwiązywania układów równań liniowych są tzw. metody podprzestrzeni Kryłowa (rozwiązujące pewien problem optymalizacyjny), których typowymi przykładami są metody sprzężonych gradientów i GMRES. Metody podprzestrzeni Kryłowa zbiegają się tym lepiej im macierz układu jest bliższa macierzy jednostkowej (mającej optymalne uwarunkowanie), stąd często w algorytmie metody pojawiają się dodatkowe operacje tzw. poprawa uwarunkowania macierzy (*preconditioning*).

Jedną z najważniejszych cech decydujących o wydajności i praktycznej użyteczności każdego z iteracyjnych solwerów liniowych (solwer – program rozwiązujący zadany problem), jest szybkość zbieżności – determinująca w ilu iteracjach solwer osiągnie wymaganą redukcję residuum układu. Ostateczna wydajność zależy także od czasu realizacji pojedynczej iteracji oraz od czasu przygotowania struktur danych do wykonywania iteracji (ten ostatni czas może być istotny dla wszelkich bardziej złożonych metod, np. z wyrafinowanymi *preconditionerami*, takimi jak algorytmy *multigrid* albo algorytmy niekompletnego rozkładu LU (*incomplete LU factorization*),

Do wyrażenia szybkości zbieżności można użyć liczby iteracji, $\mathbf{N_it}$, prowadzących do osiągnięcia zbieżności (zakładanej redukcji normy residuum), można także obliczyć uśredniony po wszystkich iteracjach stosunek norm kolejnych residuów układu (nazywany dalej uśrednioną redukcją norm): $\text{conv_rate_at_k-th_iteration} = \|\mathbf{r}_{k+1}\| / \|\mathbf{r}_k\|$ (prowadzi to do wzoru: $\|\mathbf{r}_k\| / \|\mathbf{r}_0\| = \text{average_conv_rate}^{\mathbf{N_it}} < \mathbf{e}_{\text{REL}}$, wiążącego szybkość zbieżności z liczbą iteracji $\mathbf{N_it}$. Im współczynnik average_conv_rate bliższy 0, tym zbieżność szybsza, im bliższy 1, tym zbieżność wolniejsza, dla wartości $\text{average_conv_rate} > 1$ iteracje nie zbiegają się, solwer jest rozbieżny dla danego układu równań.

W ramach zajęć będziecie Państwo badać różne metody iteracyjne przedstawione na wykładzie: Jacobiego, Gaussa-Seidla oraz GMRES – jako przykład metody podprzestrzeni Kryłowa, z różnymi wariantami poprawy uwarunkowania macierzy układu równań. Testowane będą szybkość zbieżności normy residuum i czas uzyskiwania rozwiązania. Otrzymane wyniki posłużą do porównania z algorytmem PARDISO bezpośredniego rozwiązywania układu równań, badanym na poprzednich zajęciach.

Zadanie 1.

Stworzenie folderu lab_09, przekopiowanie **problem_heat.dat**, **A2.jk** i **bc_heat.dat** z poprzednich laboratoriów, oraz pobrać plik mkb.dat. Zmiana nazwy problemu w problem_heat.dat oraz nazwy solvera.

```
/* solver input files !!! */

linear_solver_type = 1; // int:
/*      0 - direct solver */
/*      1 - preconditioned GMRES */
/*      2 - multigrid preconditioned GMRES */
/*     10 - standard iterations */
/*     20 - V-cycle multigrid */

//solver_file = "mkb.dat";           // solver_filename
solver_file = "mkb.dat";           // solver_filename
```

Zadanie 2.

Solver - 1 (GMRS)						
dla m = 0						
metoda iteracyjna	zbieżność	osiągnięta norma MAX	liczba iteracji	uśredniona redukcja norm	czas rozwiązania	błąd względny vs. PARDISO
0 - brak	TAK	0,011263265	39	0,477655	0,008653	0,810465677
1 - Jacobiego	TAK	0,009246826	12	0,089924	0,007913	0,826674552
2 - Gaussa - Seidla	TAK	1,71661E-05	16	0,161063	0,008289	0,818438691
4 - niekompletnego rozkładu LU	TAK	2,09808E-05	8	0,020494	0,01291	0,717220835
dla m = 1						
metoda iteracyjna	zbieżność	osiągnięta norma MAX	liczba iteracji	uśredniona redukcja norm	czas rozwiązania	błąd względny vs. PARDISO
0 - brak	TAK	0,035121935	64	0,647243	0,120922	0,098232612
1 - Jacobiego	TAK	1,14441E-05	33	0,418585	0,107924	0,019817267
2 - Gaussa - Seidla	TAK	0,000183105	31	0,40452	0,108208	0,017237934
4 - niekompletnego rozkładu LU	TAK	6,48499E-05	14	0,128312	0,128238	-0,164677674
dla m = 2						
metoda iteracyjna	zbieżność	osiągnięta norma MAX	liczba iteracji	uśredniona redukcja norm	czas rozwiązania	błąd względny vs. PARDISO
0 - brak	TAK	0,107143145	100	0,770318	0,38808	0,217255554
1 - Jacobiego	TAK	1,14441E-05	77	0,698282	0,361969	0,269920572
2 - Gaussa - Seidla	TAK	9,53674E-06	62	0,63688	0,355048	0,283879999
4 - niekompletnego rozkładu LU	TAK	0,000249863	27	0,357098	0,721408	-0,45505593

Najbardziej wydajnościowy w metodzie GMRES jest algorytm z zastosowaniem niekompletnego rozkładu LU, ponieważ za każdym razem uzyskujemy najmniejszą liczbę iteracji pomimo najdłuższego czasu rozwiązania. Najmniej efektywnym algorytmem jest algorytm Jacobiego gdzie dla m = 1 oraz m = 2 liczba iteracji jest największa nie uwzględniając braku metod iteracyjnych jednak czas rozwiązania jest dwukrotnie najmniejszy spośród wszystkich metod.

Solwer - 10						
Jacobi						
Zagęszczenie	zbieżność	osiągnięta norma MAX	liczba iteracji	uśredniona redukcja norm	czas rozwiązań	błąd względny vs. PARDISO
m = 0	TAK	1,90735E-06	66	0,599974	0,014521	0,681933675
m = 1	TAK	5084,023061	100	0,99451	0,116512	-0,058180299
m = 2	TAK	6,71476E+11	100	1,204038	0,650635	-0,312309145
Gauss-Seidl						
Zagęszczenie	zbieżność	osiągnięta norma MAX	liczba iteracji	uśredniona redukcja norm	czas rozwiązań	błąd względny vs. PARDISO
m = 0	TAK	1,90735E-06	22	0,214189	0,010884	0,761598108
m = 1	TAK	1,90735E-06	87	0,681436	0,113059	-0,02681961
m = 2	TAK	0,035399214	100	0,894822	0,639464	-0,289777609

Porównując algorytm Jacobiego w metodach solwera 1 z solwerem 10 można zauważyć, że liczba iteracji jest ponad 3 krotnie większa w solwerze 10, natomiast w algorytmie Gaussa-Seidla prawie 2 krotnie. Jeżeli chodzi o czas rozwiązań można zauważyć, że w metodzie GMRES czas jest znacząco mniejszy.

Najmniejszy czas pojedynczej iteracji w metodzie GMRES można odnotować dla algorytmu Gaussa-Seida dla $m = 0$ i wynosi on 0,000518, natomiast największy dla algorytmu niekompletnego rozkładu LU dla $m = 2$ i wynosi on 0,026719. Jeżeli chodzi o metodę Solwer 10 najniższy czas jest w algorytmie Jacobiego dla $m = 0$ i wynosi on 0,00022, natomiast najwyższy czas można zauważyć w algorytmie Gaussa-Seidla dla $m = 2$, gdzie czas wynosi 0,006506.

Zadanie3.

Zmiana wartości w programie mkb.dat zależnie od Solwera (opisane w pliku).

Dla PARDIO w pliku problem_heat.dat należy ustawić linear_solver_type na 0

```
FINE_LEVEL
1 1 50
0 1 -2
100 0 1.e-12
3 1 1
```

Dla GMRES + GS i GMRES + ILU(0) należy ustawić linear_solver_type na 1.

GMRES+GS

```
FINE_LEVEL
1 1 50
2 1 -2
100 0 1.e-12
3 1 1
```

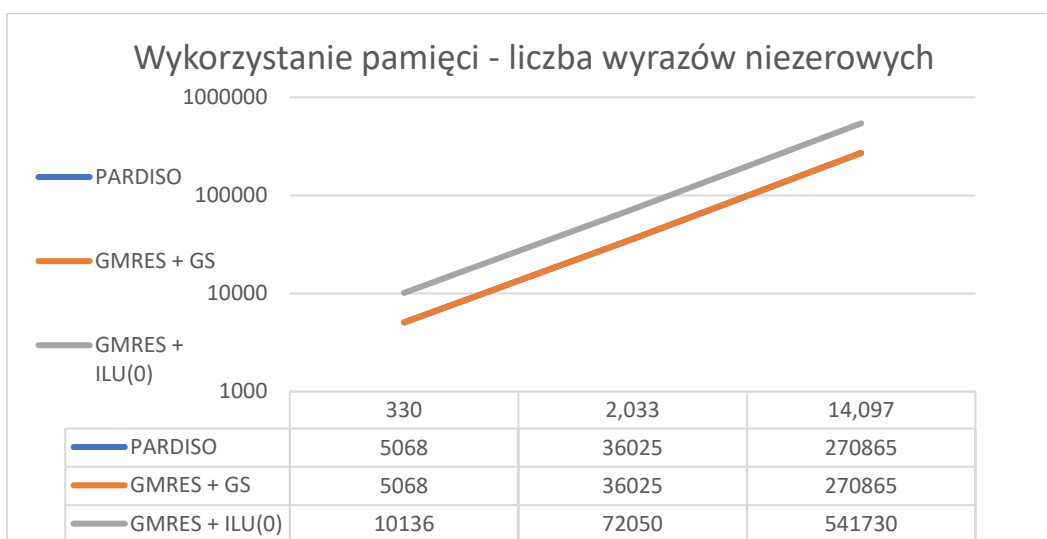
GMRES+ILU(0)

```
FINE_LEVEL
1 1 50
4 1 1
100 0 1.e-12
3 1 1
```

Odpalenie programu ModFEM.

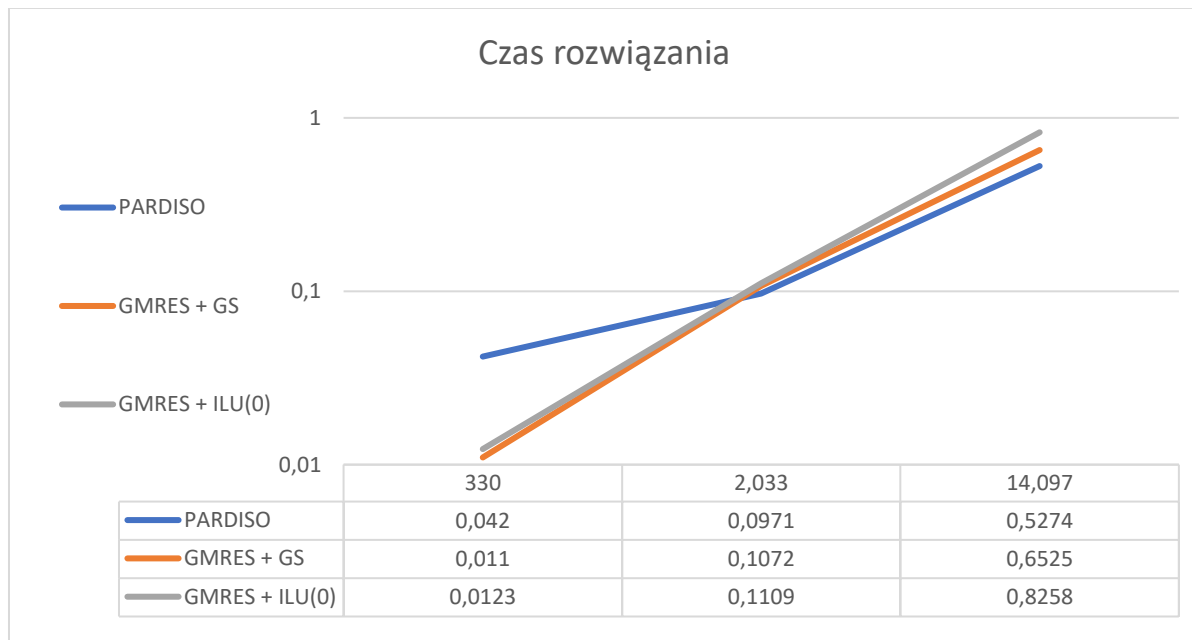
Wykorzystanie pamięci - liczba wyrazów niezerowych				
Solwer	Siatka 1 – bez adaptacji	Siatka 2 – po 1 adaptacji	Siatka 3 – po 2 adaptacjach	Siatka 4 – po 3 adaptacjach
PARDISO	5068	36025	270865	-
GMRES + GS	5068	36025	270865	-
GMRES + ILU(0)	10136	72050	541730	-
nglob	330	2,033	14,097	-

Dla trzykrotnej adaptacji siatki wyskakuje błąd „Not enough space for next edge,h Nred = 347447”, dlatego nie można określić wartości



Wykres zależności badanych parametrów od liczby stopni badanych problemów jest liniowy. Wymagania pamięciowe dla każdego solwera wartości rosną proporcjonalnie w tempie ok. 44 krotnym, jedynie różnią się wartości początkowe wykorzystania pamięci. Można zauważyć, że wartości dla solwera PARDISO pokrywają się z wartościami solwera GMRES + GS, natomiast wartości GMRES + ILU(0) są wyższe od dwóch pozostałych.

Czas rozwiązania				
Solwer	Siatka 1 – bez adaptacji	Siatka 2 – po 1 adaptacji	Siatka 3 – po 2 adaptacjach	Siatka 4 – po 3 adaptacjach
PARDISO	0,042	0,0971	0,5274	-
GMRES + GS	0,011	0,1072	0,6525	-
GMRES + ILU(0)	0,0123	0,1109	0,8258	-



Najszybciej rośnie czas wykonania dla GMRES + ILU(0), gdyż ponad 66-krotnie, dla GMRES + GS prawie 60-krotnie natomiast najwolniej dla PARDISO 12-krotnie. Dla solverów iteracyjnych czas rozwiązania rośnie liniowo, natomiast solver bezpośredni PARDISO ulega w pewnym momencie załamaniu. Solverzy iteracyjne najprawdopodobniej uzyskują czas rozwiązania krótszy niż solver bezpośredni przed momentem załamania się prostej, co widoczne jest na powyższym wykresie. Spowodowane może to być wielkością siatki, gdyż im większa siatka tym mniejszy wpływ czynników zaburzających podstawowe tendencje złożoności obliczeniowej (np. rosnąca wydajność solvera).

Zadanie 4.

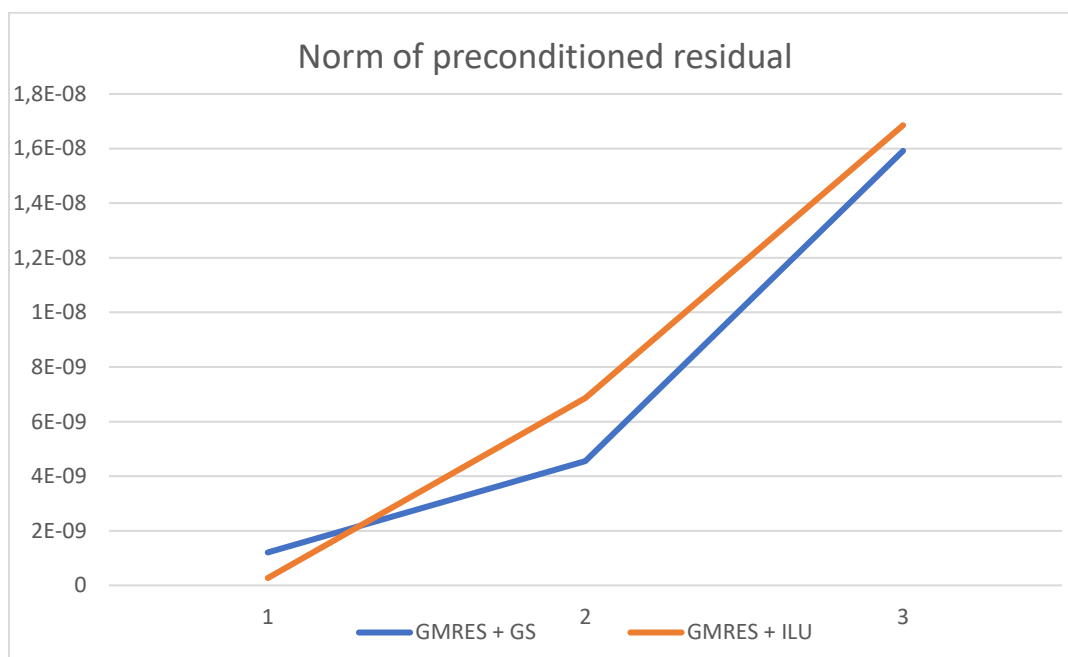
Zmiana opcji tak jak w poprzednim zadaniu dla GMRES+GS i GMRES+ILU, przy odpowiednim ustawieniu iteracji na 50 i 10.

```
GMRES+GS
FINE_LEVEL
1 1 50
2 1 -2
100 0 1.e-12
3 1 1
```

```
GMRES+ILU
FINE_LEVEL
1 1 10
4 1 1
100 0 1.e-12
3 1 1
```

a)

Norm of preconditioned residual				
Solwer	Siatka 1 – bez adaptacji	Siatka 2 – po 1 adaptacji	Siatka 3 – po 2 adaptacjach	Siatka 4 – po 3 adaptacjach
GMRES + GS	1,207E-09	4,547E-09	1,5917E-08	-
GMRES + ILU(0)	2,69E-10	6,865E-09	1,6855E-08	-

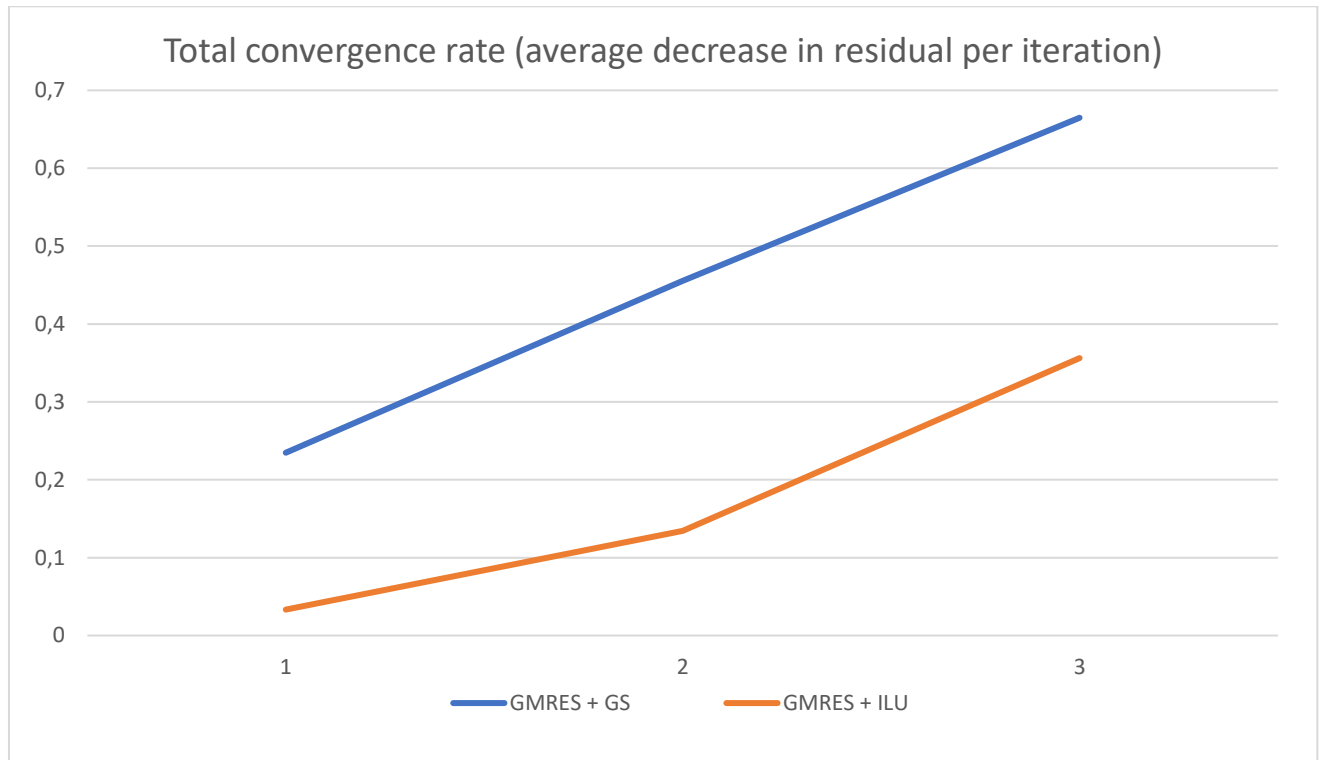


Porównanie otrzymanych wykresów na których widzimy dwa solwery iteracyjne obrazuje normę residuum po poprawie unormowania, gdzie GMRES + ILU(0) pokazuje dla każdej iteracji a GMRES + GS dla co 50. Można zaobserwować, że wartości solwera GMRES + ILU wzrastają szybciej niż dla solwera GMRES + GS oraz załamanie skoku wartości w punkcie 2 nie jest tak duże jak dla solwera GMRES + ILU. Dla solwerów wykonujących dla każdej iteracji norma jest gwałtowniejsza oraz bardziej dokładna. Stąd wynika, że solwer GMRES + ILU okazuje się najszybszy i najbardziej wydajny, gdyż osiągnięto go w mniejszej liczbie iteracji.

GMRES+ILU jest zazwyczaj bardziej dokładny niż GMRES+GS ze względu na zastosowanie techniki niezupełnej faktoryzacji LU (ILU), która pozwala na lepszą aproksymację macierzy pierwotnej.

b)

Total convergence rate (average decrease in residual per iteration)				
Solwer	Siatka 1 – bez adaptacji	Siatka 2 – po 1 adaptacji	Siatka 3 – po 2 adaptacjach	Siatka 4 – po 3 adaptacjach
GMRES + GS	0,234733	0,455182	0,664791	-
GMRES + ILU(0)	0,033304	0,13428	0,356179	-



Powyższy wykres przedstawia uśrednioną redukcję norm dla dwóch solwerów iteracyjnych opisanych w legendzie. Po przeanalizowaniu dochodzimy do konkretnych wniosków

GMRES+GS jest zazwyczaj szybszy niż GMRES+ILU ze względu na niższą złożoność obliczeniową, która wynika z wykorzystania tylko jednego kroku relaksacji Gaussa-Seidla w każdej iteracji.

Zazwyczaj, solwery iteracyjne takie jak GMRES, możemy uznać za bardziej uniwersalne niż bezpośrednio, takie jak np. PARDISO, ze względu na ich zdolność do rozwiązywania dużych, rzadkich układów równań liniowych w sposób efektywny i skalowalny.

W przypadku porównania GMRES+GS i GMRES+ILU:

Jeśli macierz jest dobrze uwarunkowana, to GMRES+GS może być bardziej efektywny ze względu na niższą złożoność obliczeniową i wystarczającą dokładność.

Jednakże, jeśli macierz jest źle uwarunkowana, to GMRES+ILU może być bardziej skuteczny ze względu na lepsze prekondycjonowanie i większą dokładność.

Zadanie (skrótowy opis)	OCENA własna w % (0-100)	OCENA prowadzącego w % (0-100)
Zad. 1 Uruchomienie zadania z iteracyjnym solwerem układów równań liniowych.	100	
Zad. 2 Badanie dokładności i wydajności obliczeń metodami iteracyjnymi.	100	
Zad. 3 Porównanie charakterystyk solwera bezpośredniego PARDISO i wybranych solwerów iteracyjnych zaimplementowanych w ModFEM	100	
Zad. 4 Badanie szybkości zbieżności solwerów iteracyjnych w zależności od rozmiaru zadania	100	
ŁĄCZNIE (400):	400	
OCENA KOŃCOWA:	-----	