

Renata Chrobak*

OBLICZENIA METODĄ ELEMENTÓW SKOŃCZONYCH NA SIECI STACJI ROBOCZYCH I NA KOMPUTERZE RÓWNOLEGLYM

1. Wprowadzenie

Uważa się obecnie, że sieć stacji roboczych stanowić może konkurencyjne rozwiązanie w stosunku do maszyn wektorowych, jak i maszyn o wysokim stopniu równoległości. Podstawę rozwoju tego nowego narzędzia informatyki stanowią różnorodne środowiska programowe [1], różniące się modelem przetwarzania, umożliwiające wykorzystanie sieci jako systemu równoległego, gdyż efektywne wykorzystanie dużej mocy *wirtualnego superkomputera* zależy od właściwego powiązania cech charakterystycznych architektury, dostępnych narzędzi programowych oraz algorytmu obliczeniowego.

Metoda elementów skończonych (MES) jest znana i stosowana od wielu lat w różnych dziedzinach nauk podstawowych i technicznych, między innymi w obliczeniach inżynierskich dużej skali złożoności. Obliczenia metodą elementów skończonych są czasochłonne i wymagają dostępu do znacznych ilości pamięci komputera. Znacząca część zainstalowanej w świecie mocy komputerowej wykorzystywana jest na potrzeby obliczeń metodą MES lub metodami pokrewnymi, w których istotnym, lecz nie jedynym elementem jest efektywne rozwiązywanie bardzo dużych układów równań liniowych o macierzach rzadkich.

Rozwój sieci komputerowych, pojawienie się środowisk programowania rozproszonego oraz wieloprocesorowych komputerów z pamięcią rozproszoną spowodowały gwałtowny wzrost zainteresowania problematyką obliczeń rozproszonych w odniesieniu do metody elementów skończonych [2–9]. Możliwość rozproszenia obliczeń MES posiada potencjalnie oczywiste zalety: skrócenie czasu obliczeń i dostęp do dużej (rozproszonej) pamięci operacyjnej. Jednak mimo zaangażowania w tę problematykę wielu znanych ośrodków naukowych i przemysłowych na świecie, wydaje się, że nie został do tej pory stworzony algorytm i struktury danych o charakterze uniwersalnym i należy wątpić w możliwość ich

* Wyższa Szkoła Pedagogiczna, Rzeszów

powstania. Wynika to z różnorodności i złożoności zjawisk symulowanych metodami MES i wypływających stąd różnych wymagań stawianych przed algorytmami i strukturami danych. Również w zastosowaniu do symulacji określonych procesów technologicznych trwają intensywne badania w zakresie równoległych algorytmów i ich efektywnej komputerowej implementacji na różnorodnym sprzęcie komputerowym.

W pracy przedstawione są wyniki badań nad wykorzystaniem sieci komputerów do modelowania metodą MES całości przebiegu procesu spęcznienia – jednego z ważnych procesów przeróbki plastycznej metali. MES jest obecnie podstawową metodą do numerycznego modelowania procesów przeróbki plastycznej metali [10]. Metoda ta w połączeniu ze sztywno-plastycznym modelem odkształcanego ciała pozwala analizować duże odkształcenia plastyczne, jakim ulega próbka podczas procesów przeróbki plastycznej, i pozwala pomijać odkształcenia sprężyste. W pracy zastosowano rozwiązanie polegające na wykorzystaniu rachunku wariacyjnego.

2. Model teoretyczny

Szczegółowy opis termomechanicznego modelu procesów plastycznej przeróbki metali zawarty jest w pracy [10]. Zastosowany w niej termomechaniczny model procesu spęcznienia składa się z dwóch części, z których pierwsza opisuje zjawiska mechaniczne. Na jej podstawie można wyznaczyć pola prędkości, odkształcenia, prędkości odkształcenia i naprężenie oraz ciepło generowane w odkształcanym metalu, zaś druga opisuje zjawiska termiczne i umożliwia wyznaczenie rozkładu temperatur w czasie odkształcenia i chłodzenia.

2.1. Model teoretyczny zjawisk mechanicznych

Przy założonym sztywno-plastycznym modelu odkształcanego ciała o objętości V , którego powierzchnia składa się z dwóch części tj. powierzchni swobodnej S_u i powierzchni tarcia S_f , rzeczywiste pole prędkości wyznaczone jest z warunku minimum funkcjonału mocy odkształcenia plastycznego

$$M = \int_V \sigma_i \dot{\epsilon}_i dV + \int_V \lambda \dot{\epsilon}_v dV - \int_{S_f} \underline{\mathbf{f}}^T \underline{\mathbf{v}} dS$$

gdzie:

- $\dot{\epsilon}_i$ – intensywność prędkości odkształcenia,
- σ_i – intensywność naprężenia,
- λ – mnożnik Lagrange'a,
- $\dot{\epsilon}_v$ – prędkość odkształcenia objętościowego,
- $\underline{\mathbf{f}}$ – naprężenie zewnętrzne na powierzchni S_f ,
- $\underline{\mathbf{v}}$ – wektor prędkości.

Zakłada się, że ośrodek jest materiałem sztywno-plastycznym spełniającym warunek płynięcia Huberta-Misesa.

Po uwzględnieniu dyskretyzacji obszaru i wyznaczeniu pochodnych względem składowych prędkości w węzłach oraz względem mnożników Lagrange'a otrzymuje się układ nieliniowych równań. Układ ten rozwiązuje się metodą Newtona-Raphsona. Rozwiązanie układu uzyskuje się iteracyjnie. W każdej iteracji:

- rozwiązuje się układ liniowych równań z niewiadomymi przyrostami prędkości węzłowych i mnożnikami Lagrange'a,
- wykonuje się korekty pola prędkości poprzez uwzględnienie obliczonych przyrostów prędkości.

2.2. Modelowanie zjawisk termicznych

W procesach przeróbki plastycznej metali występuje cały szereg zjawisk termomechanicznych, takich jak generowanie ciepła w wyniku pracy odkształcenia plastycznego, generowanie ciepła w wyniku pracy sił tarcia czy odprowadzanie ciepła w wyniku kontaktu odkształcanego metalu z narzędziem. Równocześnie następuje przepływ ciepła wewnątrz materiału w wyniku przewodzenia. W procesach przeróbki plastycznej metali temperatura jest ważnym parametrem, gdyż decyduje o oporze odkształcenia, warunkach tarcia oraz o strukturze i własnościach metalu poddawanego przeróbce. W stanie niestacjonarnym zjawiska wymiany ciepła opisuje równanie

$$\nabla' (k \nabla t) + Q - c_p \rho \frac{\partial t}{\partial t} = 0$$

gdzie:

- k – współczynnik przewodzenia ciepła,
- T – temperatura,
- Q – prędkość generowania ciepła wewnątrz materiału,
- ρ – gęstość materiału,
- c_p – ciepło właściwe materiału,
- t – czas.

W celu określenia temperatury T po czasie $t + \delta t$ przy znanym polu temperatur T_0 w czasie t , zastosowano metodę Galerkina, prowadzącą do układu liniowych równań.

3. Opis modelu procesu spęczania

Proces spęczania jest podzielony na kilka stopni. W każdym ze stopni próbka:

- pod wpływem narzędzia poddawana jest odkształceniu aż do osiągnięcia założonej wielkości odkształcenia,
- następnie jest chłodzona.

Modelowanie odkształcenia próbki w czasie kontaktu z narzędziem obejmuje wyznaczenie pola prędkości, pola prędkości odkształcenia, pola odkształcenia, wyznaczanie naprężeń oraz wyznaczanie rozkładu temperatury w próbce podczas odkształcenia. Podczas modelowania procesu spęczania należy uwzględnić zależność własności mechanicznych od temperatury, jak również zależność rozkładu temperatury od własności mechanicznych.

Proces spęczania jest procesem niestacjonarnym, gdyż prędkości węzłowe zmieniają się w czasie trwania odkształcenia. W związku z tym do modelowania tego procesu metodą elementów skończonych stosuje się rozwiązanie quasi-stacjonarne. Cały proces dzieli się na szereg kroków czasowych i w każdym kroku wyznaczone są parametry mechaniczne odkształcanego materiału, a następnie wyznaczany jest rozkład temperatury. W kolejnych krokach czasowych wykonywana jest aktualizacja siatki elementów ze względu na duże odkształcenia, jakim ulega próbka w czasie spęczania.

W każdym kroku czasowym w trakcie wyznaczania własności mechanicznych wymagane jest rozwiązanie układu nieliniowych równań, natomiast dla wyznaczenia rozkładu temperatur konieczne jest rozwiązanie układu liniowych równań. Modelowanie procesu spęczania przebiega według schematu przedstawionego na rysunku 1.

```
inicjowanie podstawowych zmiennych
```

```
for stopień = 1 to n do
```

```
    inicjowanie zmiennych
```

```
    repeat
```

```
        wyznaczenie węzłów będących w kontakcie z narzędziem
```

```
        wyznaczenie pola prędkości i pola prędkości odkształcenia
```

```
        wyznaczenie naprężeń
```

```
        wyznaczenie ciepła generowanego w wyniku odkształcenia plastycznego
```

```
        wyznaczenie ciepła generowanego w wyniku tarcia
```

```
        modyfikowanie kroku czasowego
```

```
        wyznaczenie rozkładu temperatury
```

```
        modyfikowanie współrzędnych węzłów
```

```
        modyfikowanie wartości odkształceń
```

```
        aktualizacja siatki elementów
```

```
    until osiągnięto założoną wielkość odkształcenia
```

```
    wyznaczenie rozkładu temperatury w czasie chłodzenia
```

```
od;
```

Rys. 1. Schemat obliczeń (sekwencyjnych) dla modelowaniu procesu spęczania (n określa liczbę stopni spęczania)

4. Algorytm równoległy

4.1. Ogólna charakterystyka

Dobór metody zrównoleglenia obliczeń do modelowania procesu spęczania zależy od kilku elementów, takich jak: niestacjonarność procesu spęczania, konieczność uwzględniania zarówno zjawisk mechanicznych, jak i termicznych, nieliniowość w opisie zjawisk mechanicznych, zależność własności mechanicznych próbki od temperatury oraz zależność roz-

kładu temperatury od własności mechanicznych. Dla opracowania równoległego algorytmu modelowania procesu spęczania przeprowadzono szczegółową analizę algorytmu sekwencyjnego. Stwierdzono, że:

- znaczący wpływ na czas obliczeń mają zarówno obliczenia mechaniczne, jak i termiczne; rozwiązywanie układu równań jest najbardziej czasochłonnym fragmentem obliczeń MES,
- tylko niektóre etapy obliczeń (np. wyznaczanie ciepła i modyfikacja kroku czasowego) mogą być realizowane współbieżnie; jednak czas obliczeń tych etapów jest niewielki w porównaniu z całkowitym czasem obliczeń; pozostałe etapy obliczeń muszą być realizowane sekwencyjnie, np. najpierw modelowanie odkształcenia próbki, a dopiero później modelowanie chłodzenia,
- dla zrównoleglenia obliczeń przy modelowaniu procesu spęczania konieczne jest zastosowanie dekompozycji danych.

Siatkę elementów dzieli się na podsiatki rozdzielone węzłami granicznymi. W każdej podsiatce występują dwa rodzaje węzłów: węzły lokalne i węzły graniczne. Każdą podsiatkę przypisuje się do jednego procesu. Wówczas konieczna jest synchronizacja obliczeń i komunikacja pomiędzy procesami. Sposób dekompozycji siatki elementów decyduje o złożoności komunikacyjnej, pamięciowej i obliczeniowej algorytmu. Dekompozycja siatki powinna zapewniać minimalizację narzutu czasowego związanego z komunikacją i synchronizacją obliczeń.

W pracy uwzględniono jedynie dekompozycję siatki elementów wzdłuż jednej krawędzi na pasy elementów. Przy podziale siatki wymagana jest synchronizacja obliczeń i komunikacja pomiędzy wszystkimi procesami podczas:

- obliczeń mechanicznych (rozwiązywania układu nieliniowych równań),
- obliczeń temperaturowych (rozwiązywania układu liniowych równań),
- wyznaczania kroku czasowego,
- aktualizacji siatki elementów (generacji siatki elementów na jednym procesorze).

Komunikacja tylko pomiędzy procesami przypisanymi do sąsiednich pasów konieczna jest podczas wyznaczania ciepła generowanego w wyniku odkształcenia plastycznego i podczas aktualizacji siatki elementów. Dla przyspieszenia obliczeń równoległych zastosowano asynchroniczną komunikację tam, gdzie było to możliwe.

Efektywność obliczeń równoległych metodą elementów skończonych zależy przede wszystkim od efektywności rozwiązywania dużych układów liniowych równań. W pracy ograniczono się do dokładnych metod rozwiązywania układu liniowych równań. Wybór metody rozwiązywania układu równań determinuje numeracja węzłów, gdyż decyduje ona o rozmieszczeniu niezerowych wartości w macierzy sztywności.

W pracy zastosowano trzy metody numeracji, które minimalizują liczbę operacji podczas rozwiązywania układu liniowych równań. Uzyskano 3 równoległe algorytmy modelowania procesu spęczania, różniące się tylko na etapie tworzenia i rozwiązywania układu liniowych równań. Każdy z algorytmów ma inną złożoność obliczeniową, komunikacyjną i pamięciową oraz różną optymalną dekompozycję siatki elementów. Szczegółowy opis zastosowanych algorytmów znajduje się w pracy [13].

4.2. Algorytm I – metoda dekompozycji domenowej

Zastosowano następującą numerację węzłów w siatce: najpierw ponumerowano wszystkie węzły lokalne w kolejnych pasach, a następnie węzły graniczne. Węzły lokalne numerowane są tak, by zminimalizować szerokość półpasma w odpowiednich podmacierzach. Macierz sztywności ma wtedy postać „strzałki

$$\begin{bmatrix} \mathbf{A}_1 & & & & \mathbf{B}_1 \\ & \mathbf{A}_2 & & & \mathbf{B}_1 \\ & & \ddots & & \vdots \\ & & & \mathbf{A}_p & \mathbf{B}_p \\ \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_p & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_p \\ \mathbf{c} \end{bmatrix}$$

gdzie:

- $\mathbf{A}_j, \mathbf{b}_j, \mathbf{x}_j$ – macierz sztywności, wektor niewiadomych i wektor sił dla węzłów lokalnych,
- $\mathbf{K}, \mathbf{y}, \mathbf{c}$ – macierz sztywności, wektor niewiadomych i wektor sił dla węzłów granicznych,
- $\mathbf{B}_j, \mathbf{G}_j$ – macierze określające zależność między węzłami lokalnymi a granicznymi,
- p – liczba procesorów.

Układ równań liniowych rozwiązywany jest równoległe blokową metodą eliminacji Gaussa. Sekwencyjnie metodami dokładnymi rozwiązywany jest stosunkowo niewielki układ Schura [11]

$$\hat{\mathbf{K}}\mathbf{y} = \hat{\mathbf{c}}$$

gdzie:

$$\hat{\mathbf{K}} = \mathbf{K} - \sum_{j=1}^p \mathbf{G}_j \mathbf{A}_j^{-1} \mathbf{B}_j$$

$$\hat{\mathbf{c}} = \mathbf{c} - \sum_{j=1}^p \mathbf{G}_j \mathbf{A}_j^{-1} \mathbf{b}_j$$

4.3. Algorytm II – zmodyfikowana metoda dekompozycji domenowej

Węzły lokalne podzielono na dwa rodzaje:

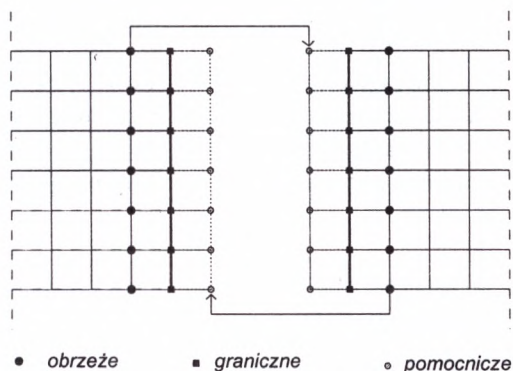
- 1) węzły obrzeża,
- 2) węzły wewnętrzne (rys. 2).

Zastosowano następującą numerację węzłów lokalnych: najpierw numerowano węzły wewnętrzne a następnie węzły obrzeża. Macierz sztywności ma wtedy postać „strzałki”, gdzie:

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \end{bmatrix} \quad \mathbf{b}_j = \begin{bmatrix} \mathbf{d}_j \\ \mathbf{g}_j \end{bmatrix} \quad \mathbf{A}_j = \begin{bmatrix} \mathbf{P}_j & \mathbf{R}_j \\ \mathbf{R}_j^T & \mathbf{S}_j \end{bmatrix} \quad \mathbf{B}_j = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{V}_j & 0 \end{bmatrix}$$

gdzie:

- P_j, d_j, u_j – podmacierz sztywności, wektor sił i wektor niewiadomych dla węzłów wewnętrznych,
- S_j, g_j, v_j – podmacierz sztywności, wektor sił i wektor niewiadomych dla węzłów obrzeża,
- R_j – określa zależność pomiędzy węzłami wewnętrznymi i obrzeża,
- V_j – określa zależność pomiędzy węzłami obrzeża i granicznymi.

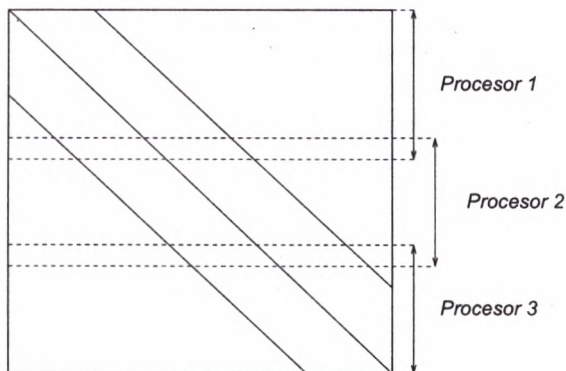


Rys. 2. Rodzaje węzłów w siatce

Układ liniowych równań rozwiązywano analogicznie jak w algorytmie I.

4.4. Algorytm III – pasmowa metoda dekompozycji

Zastosowano następującą numerację węzłów w siatce: w każdym pasie węzły (lokalne i graniczne) numerowano wzdłuż krótszej krawędzi siatki. Macierz sztywności była wówczas macierzą pasmową (rys. 3). Rozwiązywanie układu liniowych równań wykonano na jednym procesorze.



Rys. 3. Podział macierzy sztywności na podmacierze – algorytm III

5. Analiza efektywności obliczeń rozproszonych

Podstawą metody elementów skończonych jest podzielenie analizowanego obszaru na elementy. Do obliczeń zastosowano elementy czworokątne. Obliczenia mechaniczne wykonywano na elementach z czterema węzłami. Dla obliczeń temperaturowych do każdego elementu dodawano pomocnicze węzły, tak że obliczenia wykonywano na elementach z dwunastoma węzłami. Użycie większej liczby węzłów w elemencie dla obliczeń rozkładu temperatury było konieczne ze względu na występowanie znacznych gradientów temperatur w pobliżu styku metalu z narzędziem.

Dla spęczania próbek osiowosymetrycznych początkowa siatka elementów ma kształt prostokąta. Dla takiej siatki liczbę elementów określa liczba węzłów (w obliczeniach mechanicznych) wzdłuż poziomej krawędzi siatki (n_h) i pionowej krawędzi siatki (n_v). Kierunek poziomy i pionowy określony jest w stosunku do narzędzia. W pracy podzielono siatki elementów na trzy typy:

- 1) siatki kwadratowe,
- 2) siatki wąskie,
- 3) siatki pośrednie – pozostałe siatki.

Analizę złożoności pamięciowej i obliczeniowej przedstawiono w pracy [13]. Analiza ta jest zagadnieniem złożonym, gdyż modelowanie procesu spęczania jest problemem nieliniowym i niestacjonarnym. Czas obliczeń i wymagana pamięć zależą od:

- danych fizycznych próbki i otoczenia,
- parametrów numerycznych.

Parametrami numerycznymi są między innymi liczba węzłów wzdłuż krawędzi poziomych i pionowych (względem narzędzia) w siatce elementów oraz sposób numeracji węzłów. Fizyczne parametry to: rozmiar próbki, wielkość i prędkość odkształcenia, temperatura początkowa próbki i temperatura otoczenia, a także kształt narzędzia.

Liczba węzłów wzdłuż krawędzi poziomych i pionowych określa liczbę elementów oraz liczbę węzłów, liczbę równań, szerokość półpasma macierzy sztywności (dla obliczeń mechanicznych i temperaturowych). Wielkości te wpływają na złożoność obliczeniową i pamięciową. Ponadto czas obliczeń uzależniony jest od:

- liczby kroków czasowych wykonywanych podczas odkształcenia i chłodzenia,
- liczby iteracji wykonywanych podczas rozwiązywania układu nieliniowych równań w obliczeniach mechanicznych.

W pracy podjęto badanie zależności czasu obliczeń od liczby węzłów wzdłuż krawędzi poziomych i pionowych (względem narzędzia) przy ustalonych wartościach pozostałych parametrów. Do badań wybrano siatki z ustaloną liczbą węzłów (podczas obliczeń mechanicznych). Różne siatki uzyskuje się, zmieniając liczbę węzłów wzdłuż poziomej krawędzi siatki. Do badań wybrano siatki z 2304 węzłami (w obliczeniach mechanicznych) ze względu na możliwość utworzenia wielu różnych siatek z taką liczbą węzłów, w tym siatki kwadratowej.

5.1. Metodyka badań

Dokonano implementacji komputerowej równoległych algorytmów modelowania procesu spęczania. Zastosowano środowisko Parallel Virtual Machine (PVM). Równoległe programy to zestawy: jeden program nazywany dalej farmer i kopie programu tzw. worker [1].

Przeprowadzono cykl eksperymentów komputerowych, których celem była analiza i porównanie efektywności obliczeń dla poszczególnych algorytmów. Częściowe rezultaty opublikowano w pracy [12]. Szczegółową analizę efektywności obliczeń przeprowadzono dla algorytmu II. Uznano go za najbardziej efektywny. Zestawienie komputerów, na których realizowano badania, podano w tabeli 1. Pomiar czasu wykonano, posługując się systemową procedurą `gettimeofday`. Obliczenia zrealizowano na:

- homogenicznej (izolowanej) sieci stacji roboczych,
- heterogenicznej (izolowanej) sieci stacji roboczych,
- komputerze równoległym Convex Exemplar SPP1000.

Tabela 1

Zestawienie komputerów użytych w badaniach

Komputer (nazwa)	Zegar (MHz)	RAM (MB)	OS poziom optymalizacji
SUN SPARCstation 2 (SS2)	40	32	SunOS v4.1.1; -O3
IBM RS/6000-520H(RS65)	20	32	AIX v3.2; -O
IBM RS/6000-320 (RS63)	20	32	AIX v3.2; -O
HP 9000-712/80 (HP80)	80	128	HP-UX 9.05; -O2
HP 9000-712/60 (HP60)	60	32	HP-UX 9.05; -O2

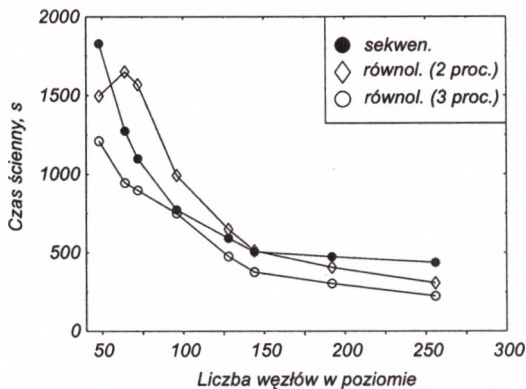
Dla zbadania efektywności czasowych algorytmów dla każdego z nich przeprowadzono serię eksperymentów komputerowych. Dla danej siatki elementów i wybranych komputerów eksperyment obejmował kilkakrotnie:

- określenie dekompozycji siatki elementów,
- wykonanie pomiaru czasu ściennego.

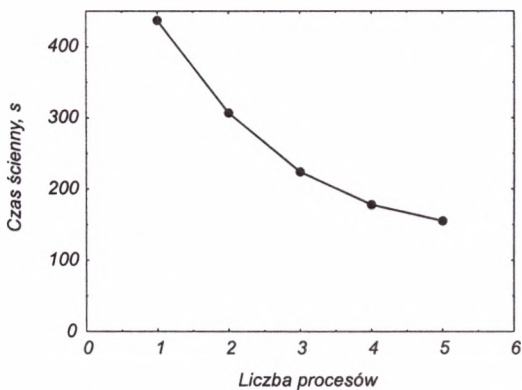
5.2. Obliczenia na homogenicznej sieci stacji roboczych

Na homogenicznej sieci stacjach roboczych IBM RS/6000 – 320 wykonano obliczenia:

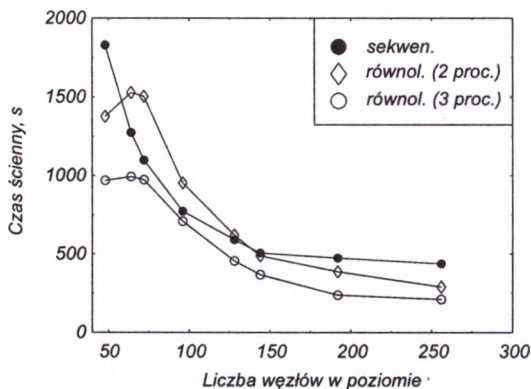
- dla siatek z 2304 węzłami; otrzymane czasy obliczeń na dwóch i trzech stacjach przedstawiono dla algorytmu I na rysunku 4, dla algorytmu II – na rysunku 6, dla algorytmu III – na rysunku 8,
- dla siatki z 256×9 węzłami (siatka wąska); otrzymaną redukcję czasu wykonania w funkcji liczby komputerów przedstawiono dla algorytmu I na rysunku 5, dla algorytmu II – na rysunku 7, dla algorytmu III – na rysunku 9.



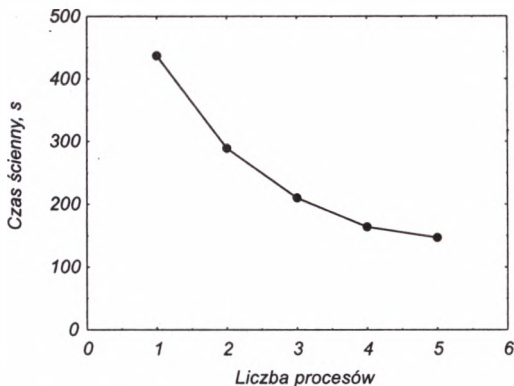
Rys. 4. Czas wykonania programu (algorytm I) na sieci IBM RS/6000-320; siatki z 2304 węzłami



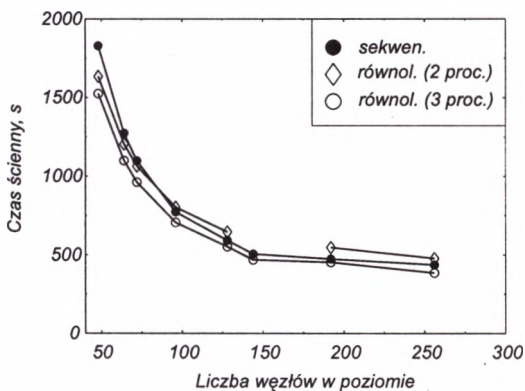
Rys. 5. Czas wykonania programu (algorytm I) na sieci IBM RS/6000-320 w funkcji liczby procesorów; siatka: 256×9 węzłów



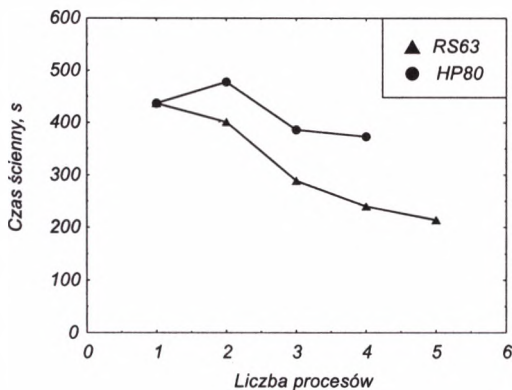
Rys. 6. Czas wykonania programu (algorytm II) na sieci IBM RS/6000-320; siatki z 2304 węzłami



Rys. 7. Czas wykonania programu (algorytm II) na sieci IBM RS/6000-320 w funkcji liczby procesów; siatka: 256 × 9 węzłów



Rys. 8. Czas wykonania programu (algorytm III) na sieci stacji roboczych IBM RS/6000-320; siatki z 2304 węzłami



Rys. 9. Czas wykonania programu (algorytm III); siatka: 256 × 9 węzłów; farmer: HP 9000-712/80, IBM RS/6000-320

Dla algorytmu I i algorytmu II kształt krzywej opisującej zależność czasu wykonania od liczby węzłów wzdłuż poziomej krawędzi siatki (przy ustalonej liczbie węzłów w siatce) jest inny w przypadku obliczeń sekwencyjnych i równoległych. Jest to spowodowane zastosowaniem odmiennych metod rozwiązywania układu liniowych równań w programie sekwencyjnym i równoległym:

- w programie sekwencyjnym rozwiązywany jest układ równań z macierzą pasmową,
- w programie równoległym zastosowano równoległy algorytm rozwiązywania układu równań metodą dekompozycji domenowej.

Czasy wykonania obliczeń przy zastosowaniu algorytmu III są zbliżone do czasu obliczeń sekwencyjnych. Dla algorytmu I i algorytmu II relacja pomiędzy czasem wykonania programu sekwencyjnego a czasem wykonania programu równoległego zależy od typu siatki. Dla siatek z 2304 węzłami:

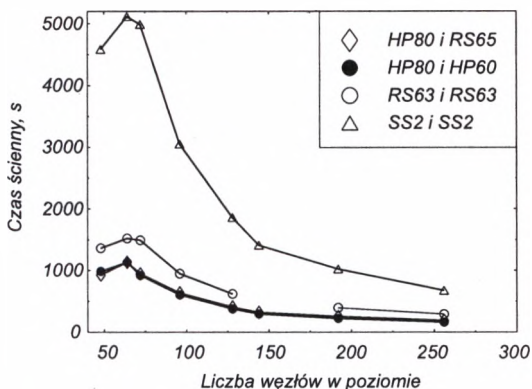
- dla siatek wąskich otrzymano czas obliczeń równoległych mniejszy niż czas obliczeń sekwencyjnych,
- dla siatki kwadratowej otrzymano czas obliczeń równoległych mniejszy od czasu wykonania obliczeń sekwencyjnych,
- dla pozostałych siatek czas wykonania programu równoległego był większy od czasu wykonania obliczeń sekwencyjnych.

5.3. Sieć heterogeniczna

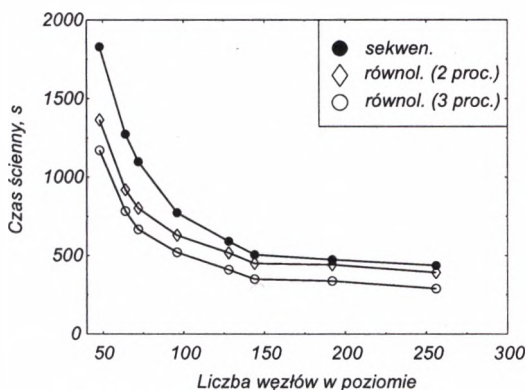
Przeprowadzono analizę efektywności obliczeń na heterogenicznej sieci komputerów. W obliczeniach na heterogenicznej sieci komputerów sposób dekompozycji siatki ma większy wpływ na czas obliczeń niż w obliczeniach na sieci homogenicznej. Zbadano dekompozycję siatki na 2 pasy elementów. Na rysunku 10 przedstawiono czasy wykonania (algorytm II) na heterogenicznej sieci komputerów dla siatek z 2304 węzłami. Program farmer wykonano na stacji HP 9000-712/80. Programy worker wykonywano na stacjach:

- IBM RS/6000-520H i HP 9000-712/80,
- HP 9000-712/60 i HP 9000-712/80,
- dwie stacje IBM RS/6000-320,
- dwie stacje SUN SPARCstation 2.

Na rysunku 11 przedstawiono czasy wykonania dla algorytmu III, w przypadku gdy program farmer wykonywany jest na stacji roboczej HP 9000-712/80, a program worker na stacjach IBM RS/6000-320. Na rysunku tym przedstawiono dla porównania czas wykonania dla programu sekwencyjnego na IBM RS/6000-320. Uzyskano krótszy czas obliczeń na sieci w porównaniu z czasem obliczeń programu sekwencyjnego (na stacji IBM RS/6000-320). Największe skrócenie czasu obliczeń uzyskano dla siatek kwadratowych i pośrednich, najmniejsze – dla siatek wąskich. Jest to związane między innymi ze znacznym wpływem czasu rozwiązywania układu równań na całkowity czas obliczeń. Spośród siatek o ustalonej liczbie węzłów czas rozwiązywania układu równań jest najdłuższy dla siatek kwadratowych, a najkrótszy dla siatek wąskich. Stąd rozwiązywanie układu równań na szybszym komputerze daje większą redukcję czasu obliczeń dla siatek kwadratowych niż wąskich.



Rys. 10. Czas wykonania programu (algorytm II) na homogenicznej i heterogenicznej sieci; podział: 2 pasy; siatki: 2304 węzły



Rys. 11. Czas wykonania programu (algorytm III); worker – IBM RS/6000-320, farmer – HP 9000-712/80; siatki z 2304 węzłami

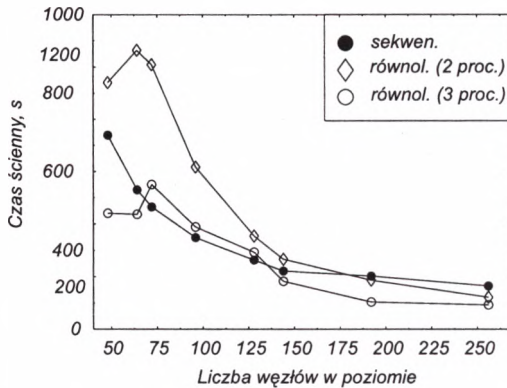
5.4. Obliczenia na komputerze równoległym

Dla algorytmu II dla siatek z 2304 węzłami przeprowadzono obliczenia na dedykowanych tzw. subkompleksach komputera Convex Exemplar SPP1000. Wówczas na czas wykonania programu nie wpływają inne programy, jak ma to miejsce w przypadku programów uruchamianych na sieci stacji roboczych. Otrzymana na komputerze równoległym (rys. 12) wydajność obliczeń jest podobna do wydajności na homogenicznej sieci stacjach roboczych IBM RS/6000-320. Na komputerze Convex Exemplar SPP1000 relacja pomiędzy czasem wykonania programu sekwencyjnego a czasem wykonania programu równoległego zależy od typu siatki:

- dla siatek wąskich, w obliczeniach równoległych otrzymano czas wykonania mniejszy niż w obliczeniach sekwencyjnych;

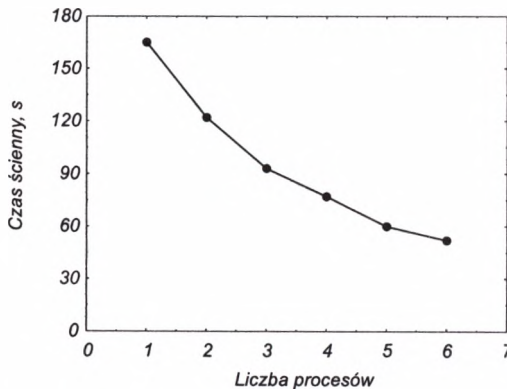
– dla siatki kwadratowej otrzymano:

- czas wykonania programu równoległego dla dwóch procesorów większy od czasu wykonania obliczeń sekwencyjnych,
- czas wykonania programu równoległego dla trzech procesorów mniejszy od czasu wykonania obliczeń sekwencyjnych,
- dla pozostałych siatek czas wykonania programu równoległego większy od czasu wykonania obliczeń sekwencyjnych.



Rys. 12. Czas wykonania programu (algorytm II) na Exemplar SPP1000; siatki z 2304 węzłami

Na rysunku 13 przedstawiono otrzymaną redukcję czasu wykonania w funkcji liczby procesorów dla siatki z 256×9 węzłami. Obliczenia przeprowadzono dla algorytmu II na komputerze Convex Exemplar SPP1000.



Rys. 13. Czas wykonania programu (algorytm II) na Exemplar SPP1000 w funkcji liczby procesorów; siatka: 256×9 węzłów

6. Wnioski

- Opracowane algorytmy umożliwiają uzyskanie istotnego czasu obliczeń dla większości siatek elementów.
- Żaden z proponowanych algorytmów nie jest algorytmem uniwersalnym. Efektywność obliczeń dla danego algorytmu zależy między innymi od typu siatki, od sposobu podziału siatki i od wyboru komputerów, na których wykonywane są obliczenia.
- Algorytmy I i II nadają się do obliczeń na homogenicznej i heterogenicznej sieci komputerów, a także na komputerach równoległych. Użycie algorytmu III jest korzystne w przypadku sieci wysoce heterogenicznej, gdy jeden komputer ma znacznie większą moc obliczeniową niż pozostałe komputery.
- Sposób dekompozycji siatki elementów jest szczególnie ważny dla obliczeń na heterogenicznej sieci komputerów i wymaga zbadania wydajności poszczególnych komputerów. Dla obliczeń na homogenicznej sieci komputerów sposób dekompozycji siatki elementów jest istotny, zwłaszcza dla algorytmu I i II, szczególnie zaś dla siatek kwadratowych.
- Dekompozycja siatki elementów na pasy nie jest optymalnym sposobem dekompozycji, zwłaszcza dla siatek kwadratowych.

Literatura

- [1] Zieliński K. (red.): *Środowiska programowania rozproszonego w sieciach komputerowych*. Kraków, Księgarnia Akademicka 1994
- [2] Synn S.Y., Falton R.E.: *Practical strategy for concurrent substructure analysis*. Computer & Structures, **54** (1995), 939–944
- [3] Fahmy M.W., Namini A.H.: *Survey of parallel nonlinear dynamic analysis methodologies*. Computer & Structures, **53** (1994), 1033–1043
- [4] Mathur K.K.: *Parallel algorithm for large scale simulation in materials processing*. in: Proceedings of the fifth International Conference on Numerical Methods in Industrial Forming Processes – NUMIFORM'95, Ithaca, 18–21 June 1995, Simulation of Materials Processing: Theory, Methods and Applications, Shen S., Dawson P.R. (eds.), A.A. Balkema/Rotterdam/Brookfield, 1995
- [5] Nelly J.R., Couch R., Futral S., Dube E.: *A parallel 3D ALE code for metal forming analyses*. In: Proceedings of the Fifth International Conference on Numerical Methods in Industrial Forming Processes – NUMIFORM'95, Ithaca, 18–21 June 1995, Simulation of Materials Processing: Theory, Methods and Applications, Shen S., Dawson P.R. (eds.), A.A. Balkema/Rotterdam/Brookfield, 1995
- [6] Płażek J., Banaś K., Kitowski J.: *Finite Element Message Passing/DSM Simulation Algorithm for Parallel Computers*. In: Hertzberger, B., Sloot, P., (Eds.), Proc. Int. Conf. High Performance Computing and Networking, Amsterdam, April 1998, Lecture Notes in Computer Science 1401, pp. 878–880, Springer 1998
- [7] Johnson A.A., Tezduyar T.E.: *Parallel computing methods for large-scale 3D simulation of fluid-object interactions*. In: Idelsohn S., Oñate E. and E. Dvorkin (Eds.), COMPUTATIONAL MECHANICS. New Trends and Application, Barcelona, CIMNE 1998

- [8] Shioya R., Ragawa G.: *Parallel finite elements of ten-million dots based on domain decomposition method*. In: Idelsohn S., Ofiate E. and E. Dvorkin (Eds.), COMPUTATIONAL MECHANICS. New Trends and Application, Barcelona, CIMNE 1998
- [9] Correla da Silva E., Devloo P.R.B., Slhessarenko Filho L., Menezes F.A.: *An object oriented environment for the development of parallel finite element applications*. In: Idelsohn S., Ofiate E. and E. Dvorkin (Eds.), COMPUTATIONAL MECHANICS. New Trends and Application, Barcelona, CIMNE 1998
- [10] Kabayashi S., Oh S.I., Altan T.: *Metal Forming and the Finite Element Method*. New York-Oxford, Oxford University Press 1989
- [11] Ortega J.M.: *Introduction to parallel and vector solution of linear system*. New York NY, Plenum Press 1988
- [12] Chrobak R., Bubak M., Kitowski J., Mościński J.: *Parallel FEM Simulation of Forging Processes on Workstations and HP/Convex Exemplar*. In: Hertzberger, B., Sloop, P., (Eds.), Proc. Int. Conf. High Performance Computing and Networking, Vienna, April 28–30, 1997, Lecture Notes in Computer Science 1225, pp. 1018–1021, Springer, 1997
- [13] Chrobak R.: *Zastosowanie heterogenicznej sieci komputerów do modelowania procesu spęcznienia metodą elementów skończonych*. Praca doktorska Wydział AEIIE AGH, Kraków 1997

Recenzent

prof. dr hab. inż. Jacek Mościński