AKADEMIA GÓRNICZO-HUTNICZA

IM. STANISŁAWA STASZICA W KRAKOWIE



Wydział Inżynierii Metali i Informatyki Przemysłowej



PROJEKT INŻYNIERSKI

pt.

„Realizacja frontalnego solwera MES z wykorzystaniem technologii OpenCL”

Imię i nazwisko dyplomanta: **Paweł Wal**

Kierunek studiów: **Informatyka Stosowana**

Nr albumu: **240202**

Opiekun: dr inż. Łukasz Rauch

Podpis dyplomanta: Podpis opiekuna:

Kraków 2014

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszy projekt inżynierski wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

Kraków, dnia ………….… Podpis dyplomanta…………….

Spis treści

[1 WSTĘP 3](#_Toc375322091)

[2 WPROWADZENIE TEORETYCZNE 5](#_Toc375322092)

[2.1 METODY ROZWIĄZYWANIA UKŁADÓW RÓWNAŃ LINIOWYCH 5](#_Toc375322093)

[2.2 METODA ELIMINACJI GAUSSA 6](#_Toc375322094)

[2.2.1 ELIMINACJA W PRZÓD 6](#_Toc375322095)

[2.2.2 PODSTAWIANIE WSTECZ 9](#_Toc375322096)

[2.2.3 POTENCJAŁ ZRÓWNOLEGLENIA METODY GAUSSA 9](#_Toc375322097)

[2.3 MACIERZE 9](#_Toc375322098)

[2.3.1 CHARAKTERYSTYKA MACIERZY MES 9](#_Toc375322099)

[2.3.2 METODY PRZECHOWYWANIA DANYCH 9](#_Toc375322100)

[3 Praktyka jakaś 9](#_Toc375322101)

[4 WNIOSKI 9](#_Toc375322102)

[5 BIBLIOGRAFIA 9](#_Toc375322103)

# WSTĘP

Ponad czterdzieści lat temu, w roku 1970, Bruce Irons opublikował swoją koncepcję programu rozwiązującego układy równań w postaci macierzy rzadkich metodą frontalną. Jego główną motywacją do opracowania tej techniki była nie tylko wydajność, ale również zużycie pamięci[4]. Irons pracował z komputerem ICT 1905, który w najlepszym razie mógł mieć 96 kilobajtów pamięci operacyjnej w postaci 32,768 słów po 24 bity. Nic więc w tym dziwnego, że poszukiwał rozwiązania jak najbardziej optymalnego pamięciowo.

Dziś co prawda pamięć jest znacznie mniej ograniczonym zasobem – przeciętny komputer biurowy dysponuje na ogół kilkoma gigabajtami, większość kart które wykorzystuje się do obliczeń nie będzie posiadała mniej niż gigabajt. Jak jednakże mówi przysłowie – apetyt rośnie w miarę jedzenia. Wraz ze wzrostem pamięci i mocy obliczeniowej wzrastają również rozmiary problemów – w tym tych z zakresu metody elementów skończonych - na które zasadzają się badacze.

Wraz z biegiem lat technologia obliczeniowa ewoluowała, kulminując obecnie w wielordzeniowych procesorach i koprocesorach, oraz – z samej swojej natury przystosowanych do operacji zmiennoprzecinkowych – kartach graficznych wykorzystywanych w zagadnieniach GPGPU. Szczególnie te ostatnie stanowią interesującą materię. Składają się z jednego lub więcej procesorów strumieniujących, z których każdy dysponuje na ogół dużą ilością (powyżej stu) rdzeni obliczeniowych, mogących wykonywać obliczenia już nie tylko równolegle, ale w sposób masowo równoległy, niedostępny dla urządzeń opartych na klasycznych procesorach. Dla kodu obliczeniowego uruchamianego na takich urządzeniach istnieje jednak szereg ograniczeń[3]: transfery z pamięci dostępu ogólnego (pamięci hosta) do pamięci urządzenia są kosztowne czasowo gdyż wymuszają okresy w których ani CPU, ani GPU nie wykonują obliczeń, operacje atomowe na pamięci globalnej są powolne, problem stanowi również nadmierna dywergencja wątków.

Niniejsza praca jest poświęcona implementacji solwera podążającego za ideą solwera frontalnego zaprezentowanego w 1970 roku przez Ironsa, rozumianej jako rozłożenie złożonego problemu na serię częściowo tylko od siebie zależnych, lżejszych pamięciowo i znacznie prostszych obliczeniowo pod-problemów. Celem przyświecającym tej pracy jest również stworzenie oprogramowania wykorzystującego w najlepszy dostępny sposób możliwości masowej równoległości oferowane przez nowoczesne, wysokowydajne urządzenia obliczeniowe.

Mimo iż oprogramowanie podąża za myślą Ironsa, postulowana przez niego metoda rozwiązywania układów równań liniowych nie została dosłownie zastosowana. Zasada matematyczna na której opiera się stworzone oprogramowanie jest wyprowadzona ze zmodyfikowanej metody eliminacji Gaussa. Modyfikacje mają na celu wykorzystanie mocnych stron zastosowanych urządzeń obliczeniowych, jednocześnie omijając ich ograniczenia i potencjalne słabości.

Do realizacji samego oprogramowania została wykorzystana architektura i zestaw bibliotek OpenCL. Każdy z wiodących producentów sprzętu obliczeniowego ma zazwyczaj swoją własną architekturę którą obsługują jego urządzenia – jest to na przykład oprogramowanie CUDA dla urządzeń firmy NVIDIA czy Stream od firmy ATI. W odróżnieniu od nich OpenCL jest otwartym standardem, którego implementacje dla konkretnych urządzeń należą wprawdzie do ich producentów, lecz jego zastosowanie jest możliwe na urządzeniach NVIDIA, ATI, Intel, a nawet na niektórych procesorach w architekturze ARM pod kontrolą systemu Android[8]. Standard OpenCL dostarcza warstwę abstrakcji, dzięki której urządzenie na którym wykonywany jest kod nie ma znaczenia dla twórcy oprogramowania, tak długo jak implementacja OpenCL dla tej platformy jest z nim zgodna.

Stworzony kod został przetestowany pod kątem wydajności dla szeregu macierzy o różnych rozmiarach i charakterystykach. [TODO: wymienić urządzenia, GT550Ti, GT540M, Tesla M2090, Intel(R) Xeon(R) CPU X5650 @ 2.67GHz, xeon phi?]

# WPROWADZENIE TEORETYCZNE

## METODY ROZWIĄZYWANIA UKŁADÓW RÓWNAŃ LINIOWYCH

Koniunkcję pewnej liczby równań liniowych w skład których wchodzi ten sam zestaw zmiennych nazywa się układem równań liniowych. Układ takowy można zapisać w postaci równania macierzowego:

Bądź jego ogólnej reprezentacji macierzowej:

Ze względu na wygodę, w przykładach umieszczanych w literaturze często stosowana jest również reprezentacja w postaci macierzy rozszerzonej:

Ze względu na czytelność, powyższa forma będzie stosowana w pozostałej części tego dokumentu.

Zostało zaproponowanych wiele metod ich rozwiązywania. Oprogramowanie komputerowe służące do ich rozwiązywania nazywane jest solwerami.

Metody służące do rozwiązywania układów równań liniowych można podzielić na dwie główne kategorie[5]:

1. Metody iteracyjne
2. Metody bezpośrednie

Do metod bezpośrednich zaliczana jest między innymi metoda eliminacji Gaussa oraz metoda faktoryzacji (dekompozycji) LU.

## METODA ELIMINACJI GAUSSA

Metoda eliminacji Gaussa składa się z dwóch zasadniczych faz.

1. Faza eliminacji w przód
2. Faza podstawiania wstecz

### ELIMINACJA W PRZÓD

Pierwsza z nich opiera się na liniowej operacji algebraicznej[6], czyli na wiedzy, iż każde równanie w układzie równań liniowych może zostać zastąpione równaniem powstałym z połączenia tego równania z dowolnym innym występującym w tym układzie.

Przedstawiając układ równań w postaci macierzowej, daje to podstawy do stosowania jednej z macierzowych operacji elementarnych, czyli dodawania lub odejmowania od siebie wierszy bądź ich wielokrotności.

Celem fazy eliminacji w przód jest doprowadzenie układu równań w postaci macierzowej do górnej macierzy trójkątnej (znanej również jako macierz schodkowa). Dokonuje się tego eliminując - przy pomocy operacji elementarnych – pewną liczbę niewiadomych z poszczególnych równań, reprezentowanych przez wiersze w macierzy [1,6].

Prześledźmy przykład fazy eliminacji w przód za [1]. Na rysunku 1 dany jest układ równań w postaci macierzy rozszerzonej:

Rysunek 1. Układ równań liniowych w postaci macierzowej

Jest to układ równań wygenerowany przez prosty problem metody elementów skończonych. Na pierwszy rzut oka widać że jest symetryczna oraz pasmowa – mimo że pasmo to jest bardzo szerokie. Są to dwie przydatne właściwości macierzowych postaci równań opisujących problemy metody elementów skończonych.

Pierwszym krokiem eliminacji w przód jest wyeliminowanie zmiennych oraz przy pomocy pierwszego wiersza macierzy pomnożonego przez odpowiednią liczbę. Po przeprowadzeniu tej operacji, macierz wygląda jak na rysunku 2.

Rysunek 2. Układ równań liniowych w postaci macierzowej po pierwszym kroku eliminacji w przód

Następnym krokiem jest postąpienie analogicznie w stosunku do zmiennych oraz , co zostało uwidocznione na rysunku 3.

Rysunek 3. Układ równań liniowych w postaci macierzowej po drugim kroku eliminacji w przód

Ostatnim krokiem w przypadku tej przykładowej macierzy jest zredukowanie wyrazu , jak zostało to uwidocznione na rysunku 4.

Rysunek 4. Układ równań liniowych w postaci macierzowej po zakończeniu eliminacji w przód

Analizując przebieg tego przykładu można wysnuć dwa pomocne wnioski. Dla każdego równania, czyli wiersza macierzy, można – przy założeniu wykonywania operacji w sposób sekwencyjny – wyznaczyć ilość operacji elementarnych które są konieczne do doprowadzenia go do pożądanej postaci. W powyższym przykładzie dla wiersza 1 było to zero operacji elementarnych, wiersz numer 2 wymagał jednej operacji elementarnej, zaś wiersze 3 i 4 – po dwie. Wychodząc z powyższego wniosku postawić można kolejny - iż dla danej macierzy można szybko wyznaczyć maksymalną ilość operacji elementarnych konieczną do sprowadzenia jej do macierzy schodkowej przypadającej na wiersz. Jak łatwo zauważyć, ta ilość operacji jest związana bezpośrednio z szerokością pasma macierzy: .

Drugi wniosek wyprowadzić można z faktu, iż eliminacja w przód sprowadza macierz do postaci macierzy schodkowej. Cechą charakterystyczną tej macierzy jest to, iż pierwszy wyraz niezerowy w danym wierszu musi znajdować się na diagonali macierzy; bezpośrednio oznacza to, że żadne dwa wiersze nie mogą mieć identycznego pierwszego wyrazu niezerowego.

## PODSTAWIANIE WSTECZ

asdasasd

### POTENCJAŁ ZRÓWNOLEGLENIA METODY GAUSSA

## MACIERZE

asdasdasda

### CHARAKTERYSTYKA MACIERZY MES

aasdasdasd

### METODY PRZECHOWYWANIA DANYCH

Asdasdasdasd

## ARCHITEKTURA OPENCL

asdasdasdasd

# Praktyka jakaś

asdasdasdasd

# WNIOSKI

asdasdasdasd

# BIBLIOGRAFIA

[1] Bathe, K. J.: Finite element procedures in engineering analysis. *Englewood Cliffs: Prentice-Hall*. 1982.

[2] Bernstein, A.J.: Analysis of Programs for Parallel Processing. *IEEE Transactions on Electronic Computers.* 1966 Vol. EC-15, no. 5, pp. 757-763

[3] Cook, S.: CUDA Programming. A Developer’s Guide to Parallel Computing with GPUs. *Waltham: Elsevier.* 2013.

[4] Irons, B.: A Frontal Solution Program For Infinite Element Analysis. *International Journal for Numerical Methods in Engineering.* 1970 Vol. 2, pp. 5-32

[5] Jamil, N.: A Comparison of Direct and Indirect Solvers for Linear Systems of Equations. *International Journal of Emerging Sciences.* 2012 Vol. 2, no. 2, pp. 310-321

[6] Rońda J., Oliver G.J., Introduction to numerical methods with Matlab procedures, 2010.

[7] ICT 1900 Series Central Processors 1904, 1905, *ICT Press release* (ICT), 1964 p. 4. [dostęp: 2013-11-02], Dostępny w Internecie:

<<http://bitsavers.trailing-edge.com/pdf/ict_icl/1900/brochures/1904_Central_Processor_Sep64.pdf>>

[8] Conformant Products [online], *Khronos Group*, [dostęp: 2013-12-11], Dostępny w internecie:

<<http://www.khronos.org/conformance/adopters/conformant-products#opencl>>