Solvery w ViennaCL:

Iteracyjne:  
<http://en.wikipedia.org/wiki/Conjugate_gradient_method>  
<http://en.wikipedia.org/wiki/Biconjugate_gradient_stabilized_method>

<http://en.wikipedia.org/wiki/Generalized_minimal_residual_method>

Bezpośrednie:

Triangular solvers (?) – dzielenie macierzy na trojkatne

<http://en.wikipedia.org/wiki/LU_decomposition>

Od Pawła Wala – już z nim rozmawiałem.

2. Przegląd solwerów układów równań liniowych.

Solwery wykorzystywane do rozwiązania układów równań liniowych można podzielić na iteracyjne i bezpośrednie.

**Solwery iteracyjne**

Solwery iteracyjne były używane w dużym zakresie we wczesnych latach 60’tych ubiegłego wieku, jednak z czasem zaprzestano wykorzystywanie ich do obliczeń. Było to spowodowane odkryciem, że liczba operacji, które były przeprowadzane przez metody iteracyjne przekraczała w znacznym stopniu limity wyznaczone w założeniach teoretycznych. Zastąpione zostały one o wiele bardziej efektywnymi solwerami bezpośrednimi, opartymi o trójkątny rozkład macierzy. Wzrastająca moc obliczeniowa oraz możliwość zrównoleglania obliczeń sprawiła jednak ponowne ożywienie w świecie metod iteracyjnych. Możliwości nowych architektur spowodowały, że solwery iteracyjne były wydajniejsze niż bezpośrednie metody dla bardzo dużych problemów tj. większych niż 100 000 elementów [1]. Kolejną zaletą było o wiele mniejsze zużycie pamięci, ponieważ wymaga on mniej więcej takiej samej ilości pamięci co oryginalne dane [2]. Jest to spowodowane tym, że współczynnik macierzy pozostawiany jest w oryginalnej formie i traktowany jako liniowy operator do wyliczania produktu wynikowego macierzy i wektora. Dodatkowo w przypadku solwera iteracyjnego użytkownik może wykorzystać zrównoleglanie obliczeń .

Iteracyjne metody rozwiązywania układów równań liniowych polegają na krokowym wyznaczaniu rozwiązania. Z każdą kolejną iteracją wynik jest poprawiany, aż do momentu gdy wynik zbliży się wystarczająco do założonego przybliżenia. W związku z tym końcowy wynik solwera iteracyjnego jest zawsze obarczony pewnego stopnia błędem. Teoretycznie używając solwera iteracyjnego można uzyskać idealny wynik przy nieskończonej ilości kroków, jednak w rzeczywistych obliczeniach ilość iteracji jest zawsze wartością skończoną. Złe warunki początkowe mogą jednak spowodować duży błąd w ostatecznych wynikach [5] . Kiedy solver iteracyjny rozwiązuje źle uwarunkowany problem wyniki mogą być wolne lub nie mieć zbieżności. Głównymi operacjami metod iteracyjnych są mnożenia macierzy i wektorów oraz rozwiązywanie liniowych równań algebraicznych dotyczących uwarunkowań wstępnych [4] . Do metod iteracyjnych rozwiązywania układów równań należą między innymi:

- Metoda gradientu sprzężonego - Jest ona jedną z najlepiej znanych metod iteracyjnych wykorzystywaną do rozwiązywania liniowych układów równań z rzadką, symetryczną, dodatnio określoną macierzą [3] . W każdym kolejnym kroku wykonywane jest mnożenie macierzy i wektora. Dla symetrycznej dodatnio określonej macierzy o rozmiarze N x N zbieżność można osiągnąć w N-iteracjach [6].

- Metoda gradientu Bi-sprzężonego BiCG – rozwija metodę gradientów sprzężonych. Umożliwia ona rozwiązanie dla nie-pojedynczej, niesymetrycznej macierzy.

- Ustabilizowana metoda gradientu Bi-Sprzężonego - służy do rozwiązywania liniowych układów równań z niesymetrycznych równań liniowych. Jest stabilniejsza od Metody Gradientu Bi-Sprzężonego.

- Uogólniona Metoda Najmniejszego Residuum GMRES – jest to dosyć popularna metoda do rozwiązywania nieosobliwych, niesymetrycznych macierzy. Badania dowiodły, że w metodzie GMRES stopa zbieżności może poprawiać się wraz z każdą kolejną iteracją [6] [7] .

Popularne metody realizacji uwarunkowania wstępnego: <- opisywać?

* Jacobiego, Gaussa-Seidla, SOR, SSOR i ich blokowe wersje, ICC, ILU, z rzadką aproksymacją macierzy odwrotnej (SPAI), wielomianowa itd.

Nie istnieje jakaś metoda najlepsza, gdyż:

* różnym układom równań najlepiej odpowiadają różne metody,
* pewne metody okazują się bardziej podatne na implementację równoległą niż inne.

**Solwery bezpośrednie**

Podział solwerów bezpośrednich:

- Bezpośrednie klasyczne

- Frontalne –podejście do rozwiązywania układów liniowych szeroko stosowane w analizie elementów skończonych.

Decydującym czynnikiem przy wyborze solver bezpośredniego może być chęć użycia metody, która znacząco zmniejszy ilość niezerowych przejść podczas faktoryzacji macierzy. Dodatkowym atutem jest także możliwość wykorzystania pamięci dyskowej podczas dużych obliczeń na komputerach z małą ilością pamięci RAM i zachowanie przy tym wysokiej wydajności i prędkości podczas przetwarzania danych przechowywanych w RAM. Bezpośrednie metody pozwalają również na wykrywanie geometrycznej niestabilności modelu. Do tej pory najbardziej rozpowszechnionym solwerem bezpośrednim był solwer wielofrontalny. Posiada on wszystkie wymienione powyżej zalety, jednak posiada także nadmierną ilość transferów pamięć – pamięć oraz pamięć – dysk – pamięć. Na wielordzeniowym komputerze w architekturze SMP stanowi to problem uniemożliwiający osiągnięcie maksymalnych osiągnięć i przyśpieszenia z wykorzystaniem dodatkowych procesorów. Powszechnym problemem z bezpośrednimi solwerami jest kwadratowa zależność liczby operacji w wymiarze problemu, przekraczanie ilości dostępnej pamięci RAM oraz zbyt duży czas faktoryzacji macierzy[4] . Do solwerów bezpośrednich zalicza się[[1]](#footnote-1):

- Metodę eliminacji Gaussa – bezpośredni solwer rozwiązujący układy równań pierwszego stopnia.

- Metoda eliminacji Gaussa- Jordana –

- Dekompozycja LU – metoda ta sprowadza się do stworzenia dwóch macierzy trójkątnych dolnej oraz górnej (z ang. Lower – dolna Upper – górna) oraz rozwiązania odpowiedniego równania.

**Porównanie wydajności na podstawie przeprowadzonych do tej pory badań**

[1] Jouglard C. E., Coutinho A.L.G.A., A comparison of iterative multi-level finite element, Solvers, Computers & Structures, 69, 655-670, 1998.

[2] Benqi G., Weiming C., An iterative and parallel solver based on domain decomposition for the h-p version of the finite element method, Journal of Computional and Applied Mathematics, 83, 71-85, 1997.

[3] <http://www.icm.edu.pl/kdm/Metoda_gradient%C3%B3w_sprz%C4%99%C5%BConych_CG>

[4] Fialko S.Y., [Iterative methods for solving large-scale problems of structural mechanics using multi-core computers](http://www.sciencedirect.com/science/article/pii/S1644966513000666), Archives of Civil and Mechanical Engineering, 14, 190-203, 2014

[5] <http://th-www.if.uj.edu.pl/zfs/gora/metnum12/wyklad05.pdf>

[6] Verschoor M, Jalba A. C. Analysis and performance estimation of the Conjugate Gradient method on multiple GPUs, Parallel Computing, 38, 552-575, 2012

[7] Van der Vorst H. A., Vuik C., The superlinear convergence behaviour of GMRES, Journal of Computional and Applied Mathematics, 48, 327-341, 1993

W opisie solwera - > opisac wydajnosc

Benchmarking -> Solwer – Sprzet – Dane

Wyniki -> Czas, Global i Local Worksize -> Na Nvidii zużycie energii

1. Tu kurna maja być linki [↑](#footnote-ref-1)