Egzamin z obliczeń naukowych 2008 gr 1

Uwagi wstępne

Wyniki zadań teoretycznych max. jedna strona A4. Na początku macie Państwo max 35 min na część teoretyczną w tym czasie nie korzystacie Państwo z komputerów, swoich notatek itp Po oddaniu reszta czasu na zadania komputerowe.

Wyniki wszystkich zadań z octave'a w jednym skrypcie i m-plikach.

Wyniki zadania z Bibliotek: pliki w C - i Makefile - komenda *make zadanie* ma kompilować program.

Wszystkie pliki spakowane do zgzipowanego archiwum tar wysyłacie Państwo mailem na adres: `L.Marcinkowski@mimuw.edu.pl` (pakujemy komendą: `tar zcvf Nazwisko.tar.gz plik1 plik2... plikM`).

Niech **K** będzie sumą z zadań z Octave'a i Bibliotek komputerowych a **T** sumą punktów z zadań teoretycznych. Propozycję oceny będzie zależeć od $\mathbf{K}+\mathbf{T}$. Warunkiem dopuszczenia do egzaminu ustnego czyli możliwości poprawy oceny jest $\mathbf{K} \geq \mathbf{10}$.

Octave

Rozpatrzmy zagadnienie drgania struny wiolonczeli o długości $L = 1$ w czasie $[0, T]$ - znamy jej pozycje i prędkośc dla $t = 0$. Opisuje je równanie hiperboliczne

$$u_{tt} = u_{xx} - 0.2 * u_t, \quad u(t,0) = u(t,1) = 0, \quad u(0,x) = u0(x), \quad u_t(0,x) = v0(x)$$

gdzie $u0(x), v0(x)$ dane funkcje.

W wyniku dyskretyzacji tego problemu względem zmiennej x na siatce równoodległej $x_k = k * h,\ k = 0, .., N+1$ dla $h = \frac{1}{N+1}$ dla danego N otrzymujemy układ równań zwyczajnych dla $u_k(t)$ z $k = 1, \ldots, N, \quad t > 0$:

$$\frac{d^2u_k}{dt^2} = h^{-2} * (u_{k-1} - 2 * u_k + u_{k+1}) - \frac{1}{5} * \frac{du_k}{dt} \qquad t > 0, \quad k = 1, \ldots N, \tag{1}$$

$$u_k(0) = u0(x_k) \qquad \frac{du_k}{dt}(0) = v0(x_k), \tag{2}$$

przy czym mamy $u_0(t) = u_{N+1}(t) = 0$ dla dowolnego $t > 0$. $u_k(t)$ przybliża wartość rozwiązania $u(t, x_k)$.

Wsk: równanie różniczkowe (1) można zapisać jako $\frac{d^2}{dt^2}\vec{u} = h^{-2} * A_N * \vec{u} - \frac{1}{5} * \frac{d\vec{u}}{dt}$ dla macierzy A_N *trójdiagonalnej* z $\vec{u} = (u_1, \ldots, u_N)^T$.

Interesuje nas rozwiązanie dla czasów równoodległych $t_n = n * \tau$ dla $n = 0, \ldots, M$ z $\tau = \frac{T}{M}$.

Zadanie 1 (13pkt) Napisz m-plik z funkcją `U=struna()`, której parametrami jest $N, \vec{u0}, \vec{v0}, T, M$ gdzie

$$\vec{u0} = (u0(x_1), \ldots; u0(x_N))^T \ i \ \vec{v0} = (v0(x_1), \ldots, v0(x_N))^T,$$

która zwróci jako rozwiązanie U - macierz $(M+1) \times N$ taką że n-ty wiersz jest długości N i zawiera przybliżenie rozwiązania dla czasu $t_n = n * \tau$ tzn $u_k(t_n)$.

W skrypcie zastosuj odpowiednio powyższą funkcje dla $N = 49$,$T = 5$ dla $u0(x) = 0.2 * (0.5 - |x - 0.5|)$ i $v0(x) = 0$ dla $x \in [0, 1]$ tak aby:

- Narysować wykres rozwiązań dla wszystkich czasów $t_k = k * \tau \in [4, 5]$ dla $\tau = 0.1$ względem x na jednym wykresie.
- Narysować wykres rozwiązania dla ustalonych 2 punktów siatki $x_2 = 2 * h = 0.04$ oraz $x_{40} = 40 * h = 0.8$ względem czasu.
- Podać maksymalną amplitudę określonego punktu struny (maximum modułu wartości odpowiedniego $u(t, x_k)$) dla $x_k = 0.6$.

Biblioteki

Zadanie 2 (10pkt) Napisać funkcję w osobnym pliku źródłowym, która korzystając z funkcji `DGEMM` z BLASów i `DGESV` z Lapacka obliczy macierz X rozwiązanie równania $A * X = B * F$ z danymi macierzami A, B, F wymiaru $N \times N$. W przypadku gdy nie ma rozwiązania funkcja powinna zwrócić odpowiednią informacje na ekran.

Parametry wejściowe funkcji: macierze A, B, F, i wymiar N. Na wyjściu wynik czyli macierz X zwrócona w macierzy F.

Zastosować w programie do macierzy:

$$A = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \qquad F = \begin{pmatrix} 2 & -1 \\ 0 & 1 \end{pmatrix}$$

Wynik wyprowadzić na ekran. W `Makefilu` włącz sensowne opcje optymalizacji. Program ma się kompilować po wywołaniu komendy `make zadanie`.

Teoria

Zadanie 3 (1pkt) Czy układ RRZ $y' + Ay = 0;\ y(t_0) = y0$ dla A macierzy wymiaru $N \times N$ ortogonalnej może być sztywnym?

Który ze schematów lepiej zastosować do rozwiązywania tego układu: zmodyfikowany schemat Eulera czy schemat trapezów (ze względu na koszt)?

Zadanie 4 (2pkt) Rozpatrzmy równanie nieliniowe w N wymiarach:

$$F(\vec{x}) = (y_1^2, \ldots, y_N^2)^T - \vec{1}$$

dla $\vec{y} = A * \vec{x}$ gdzie macierz $A = A^T > 0$ jest $N \times N$ i $\vec{1} = (1, \ldots, 1)^T$. Jeśli zastosujemy metodę Newtona do znalezienie jakiegoś rozwiązania to czy otrzymamy zbieżność lokalną a jeśli tak to czy rząd zbieżnosci będzie co najmniej kwadratowy? (w świetle teorii którą Państwo znacie.)

Zadanie 5 (1pkt) Zapisać macierz F z zadania bibliotek w formacie współrzędnych (tzn zapisać wszystkie dane potrzebne w tym formacie)

Egzamin z obliczeń naukowych 2008 gr 2

Uwagi wstępne

Wyniki zadań teoretycznych max. jedna strona A4. Na początku macie Państwo max 35 min na część teoretyczną w tym czasie nie korzystacie Państwo z komputerów, swoich notatek itp Po oddaniu reszta czasu na zadania komputerowe.

Wyniki wszystkich zadań z octave'a w jednym skrypcie i m-plikach.

Wyniki zadania z Bibliotek: pliki w C - i Makefile - komenda *make zadanie* ma kompilować program.

Wszystkie pliki spakowane do zgzipowanego archiwum tar wysyłacie Państwo mailem na adres: `L.Marcinkowski@mimuw.edu.pl` albo `przykry@mimuw.edu.pl` w zależności od pilnującego, (pakujemy komendą: `tar zcvf Nazwisko.tar.gz plik1 plik2... plikM`).

Niech **K** będzie sumą z zadań z Octave'a i Bibliotek komputerowych a **T** sumą punktów z zadań teoretycznych. Propozycję oceny będzie zależeć od $\mathbf{K}+\mathbf{T}$. Warunkiem dopuszczenia do egzaminu ustnego czyli możliwości poprawy oceny jest $\mathbf{K} \geq \mathbf{10}$.

Octave

Rozpatrzmy metodę iteracyjną znalezienia rozwiązania układu równań nieliniowych

$$F(\vec{x}^*) = 0 \qquad F : G \subset R^N \to R^N$$

zwaną metodą cięciw w której kolejne iteracyjne przybliżenie x_{n+1} jest określone równaniem

$$DF(\vec{x}_0)(\vec{x}_{n+1} - \vec{x}_n) + F(\vec{x}_n) = 0 \qquad n \geq 0$$

gdzie x_0 dane przybliżenie startowe. Metoda jest dobrze zdefiniowana o ile $DF(\vec{x}_0)$ jest macierzą nieosobliwą.

Interesuje nas implementacja powyższej metody w której $DF(\vec{x}_0)$ zostanie zastapione przez $A = (a_{ij})$ przybliżony Jakobian zdefiniowany jako

$$a_{ij} = \delta^{-1}(F_i(\vec{x}_0 + \delta * \vec{e}_j) - F_i(\vec{x}_0)) \qquad i, j = 1, \ldots, N$$

dla zadanego parametru $\delta = 1e-7$. Tutaj F_i odpowiednia i-ta składowa funkcja F tzn $F(\vec{x}) = (F_1(\vec{x}), \ldots, F_N(\vec{x}))^T$ oraz $\vec{e}_j$ j-ty wersor np $\vec{e}_1 = (1, 0, 0, \ldots, 0)^T$.

Warunkiem zatrzymania metody jest aby spełniony był nastepujący warunek stopu:

$$\|F(\vec{x}_n)\|_2 \leq atol + rtol * \|F(\vec{x}_0)\|_2 \tag{3}$$

gdzie $atol, rtol > 0$ zadane parametry. W przypadku gdy $n > 100$ i warunek stopu nie jest spełniony metoda też się zatrzymuje ale uznajemy że nie ma zbieżności.

Zadanie 1 (13pkt) Napisz m-plik z funkcją `cieciwa(nazwaF,X0,rtol,atol)` której parametrami są: `nazwaF` - nazwa funkcji octave'a postaci `[Y]=F(X)`, która dla danego wektora X obliczy $Y = F(X)$, wektor startowy $X0$, oraz tolerancje stopu $rtol$ i $atol$ dwa jako parametry opcjonalne, por. (3), z domyślnymi wartościami: $atol = 10^{-8}$ i $rtol = 10^{-6}$.

Funkcja ma zwrócić $\vec{x}_n$ n-te przybliżenie iteracyjne - pierwsze dla którego zachodzi (3), uzyskane przy pomocy metody cięciw opisanej powyżej (Jakobian dla $\vec{x}_0$ obliczony w sposób przybliżony jak wyżej) oraz $info$ w którym zawarty zostanie kod wyniku: zero jeśli został spełniony warunek stopu (3), jeden jeśli warunek stopu (3) nie jest spełniony ale ilośc iteracji przekroczyło 100 tzn gdy $n = 101$ - wtedy zwrócone ma zostać $\vec{x}_{101}$

Przetestuj w skrypcie funkcje `cieciwa()` na przykładzie nastepującej funkcji $F : R^{100} \to R^{100}$:

$$F(\vec{x}) = 10^4 * A * \vec{x} + g(\vec{x}) - \vec{1}$$

gdzie $\vec{1} = (1, \ldots, 1)^T$, $A = A^T$ macierz trójdiagonalna 100×100 taka, że $a_{ii} = 2$ a $a_{ij} = -1$ o ile $|i - j| = 1$, $a_{ij} = 0$ o ile $|i - j| > 1$, oraz $g(\vec{x}) = (x_1^2, x_2^2, \ldots, x_{100}^2)^T$. Za przybliżenie startowe proszę wziąść $\vec{x}_0 = \vec{1} = (1, \ldots, 1)^T$ i domyślne tolerancje błędu. Na ekran proszę wydrukować komunikat czy metoda zbiegła i $\|F(\vec{x})\|_2 / \|F(\vec{x}_0)\|_2$ dla $\vec{x}_n$ zwróconego przez Państwa funkcje `cieciwa()`.

Biblioteki

Zadanie 2 (10pkt) Napisać funkcję w *osobnym* pliku źródłowym C, która korzystając z funkcji `dsyev` z Lapacka obliczy najmniejszą wartość własną macierzy symetrycznej: A wymiaru $N \times N$ (tzn funkcja ma zwrócić $\min_n \lambda_n$ dla λ_n wartości własnych A).

Parametry wejściowe funkcji: macierze A w formacie takim jaki wymaga Lapack i wymiar N.

Zastosować w programie do macierzy

$$A = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$$

Wynik wyprowadzić na ekran.

W `Makefilu` włącz sensowne opcje optymalizacji. Program ma się kompilować po wywołaniu komendy `make zadanie`.

Teoria

Zadanie 3 (1pkt) Do zagadnienia początkowego skalarnego z $f \in C^\infty(R)$

$$y' = f(y) \qquad y(0) = y0$$

które ma rozwiązanie gładkie prostej rzeczywistej zastosowano schemat Heuna rzędu dwa z krokiem $h = 10^{-2}$ i otrzymano wektor $(y_0, \ldots, y_{1000})$, który z y_k przybliża najlepiej $y(1)$ (wg teorii schematów)? Proszę podać rząd oszacowania błedu $|y_k - y(1)|$ jako odpowiednie $O(h^p)$.

Zadanie 4 (2pkt) Mamy równanie $F(\vec{x}) = A * \vec{x} + g(\vec{x}) - \vec{f}$ dla dowolnej macierzy A wymiaru 9×9 symetrycznej, funkcji $g(\vec{x}) = (x_1^3, \ldots, x_9^3)^T$, zadanego wektora $\vec{f}$. Załóżmy że istnieje $\vec{x}^*$ rozwiązanie dla odp. $\vec{f}$. Czy możemy być pewni że metoda Newtona zastosowana do znalezienia przybliżenia $\vec{x}^*$ będzie zbieżna lokalnie kwadratowo (na podstawie wiedzy z wykładu)? Uzasadnić.

Zadanie 5 (1pkt) Opisać krótko w jakiej strukturze danych należy trzymać macierz aby wywołać z poziomu programu C funkcję z biblioteki LAPACK, której argumentem jest ta macierz.