Ćwiczenie 1 z metod obliczeniowych w nauce i technice

dla .....

Pobierz z pliku MOwNiT_cw1_dane parametry i, j, p, q, r, s odpowiadające swemu nazwisku.

Parametry i, j wskazują, jakie wybrać funkcje (np. $i = 1,\ \ j = 2$ to $f = \arctan x \cdot \cos x$),

zaś p, q, r, s macierz $\boldsymbol{A}$.

1. Użyj skryptu `lagrange.m` do obliczenia aproksymacji Lagrange'a 4 stopnia wybranej przez siebie funkcji $f(x), f(x) = \ldots\ldots\ldots\ldots\ldots$, na wszystkich częściach odcinka $[a; b]$ o długości $(b-a)/2^n, n = 0, 1, 2, 3$. Narysuj i zachowaj wykres zależności $\log_{10}$(błędu) od $\log_{10}$(liczby odcinków) oraz wykres funkcji $f(x)$ wraz z jej interpolantem na całym (a, b). Odczytaj stopień zbieżności, tj. współczynnik nachylenia krzywej zbieżności. Przez błąd aproksymacji rozumiemy maksymalny błąd na odcinku $[a, b]$, czyli maksymalny z pododcinków.
   Uwaga: Należy dobrać rozsądnie gładką funkcję $f(x)$, aby mogła ona być względnie dokładnie interpolowana na (a, b) np. z dokł. 5%. Do konstrukcji $f(x)$ użyj wskazanych funkcji:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\arctan x$	$\cos x$	$\sin x$	$\tan x$	$\arccos x$	$\arcsin x$	x^3	e^x	$\ln x$	$\cos x$	$\sqrt{x}$	$\sqrt[3]{x}$	x^2	$\cot x$

2. Zastosuj skrypty `inttria.m` oraz `intpara.m` do obliczenia całki oznaczonej wybranej funkcji $f(x)$ jak wyżej, na odcinku $[c; d]$ z podziałami na $n = 10, 20, 40$ i 80 pododcinków. Narysuj i zachowaj wykresy zależności $\log_{10}$ (błąd całkowania) od $\log_{10} n$ (na **jednym** rysunku). Ustal stopnie zbieżności tj. tg kierunkowe otrzymanych wykresów.
   Uwaga: Należy dobrać rozsądnie gładką funkcję, aby mogła być ona względnie dokładnie całkowana. Ścisłą wartość całki można obliczyć stosując np. `intpara.m` dla dużego n, np. $n = 1000$.

3. Zastosuj skrypty `bisect.m`, `siecz.m` oraz `newton.m` do rozwiązania wybranego równania nieliniowego $f(x) = 0$ z dokł. $\epsilon = 10^{-14}$, gdzie $f(x)$ jest jak w zad. 1. Narysuj i zachowaj wykresy zależności $\log_{10} |x - x_n|$ od n, gdzie n jest numerem iteracji, x rozwiązaniem ścisłym, x_n przybliżonym w n-tym kroku (na **jednym** rysunku). Podaj wartości użytych parametrów. (Wartości $|x - x_n|$ drukuje program).

4. Napisz skrypt metody iteracyjnej Jacobiego i Gaussa-Seidela rozwiązywania układu równań liniowych $\boldsymbol{Ax} = \boldsymbol{b}$. Sprawdź zbieżność na wybranym układzie równań 4x4 o symetrycznej macierzy z dominującą dodatnią przekątną (tj. $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$). Narysuj wykresy zależności $\log_{10}$(błędu) od numeru iteracji (na **jednym** wspólnym rysunku). Odczytaj i zweryfikuj z teorią stopnie zbieżności. Porównaj rozwiązanie ścisłe i przybliżone.

$$\boldsymbol{A} = \begin{bmatrix} p & \cdot & \cdot & \cdot \\ \cdot & q & \cdot & \cdot \\ \cdot & \cdot & r & \cdot \\ \cdot & \cdot & \cdot & s \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

5. Używając funkcji `eig(a)` znajdź wartości i wektory własne wybranej macierzy symetrycznej 4x4 (w podwójnej precyzji). Sprawdź poprawność otrzymanego wyniku (tj. czy $\boldsymbol{A}\boldsymbol{q}_i = \lambda_i \boldsymbol{q}_i$), a także wzajemną prostopadłość wektorów własnych (czy $\boldsymbol{q}_i \cdot \boldsymbol{q}_j = 0, i \neq j$). Sprawdź, czy $\sum A_{ii} = \sum \lambda_i$ i $\det \boldsymbol{A} = \lambda_1 \lambda_2 \lambda_3 \lambda_4$.

$$\boldsymbol{A} = \begin{bmatrix} p & \cdot & \cdot & \cdot \\ \cdot & q & \cdot & \cdot \\ \cdot & \cdot & r & \cdot \\ \cdot & \cdot & \cdot & s \end{bmatrix}$$

6. Napisz skrypty do metody potęgowej oraz odwrotnej metody potęgowej (z przesunięciem) do rozwiązywania zagadnienia własnego. Zastosuj te metody do macierzy z zadania poprzedniego: potęgową do największej wartości własnej , odwrotną potęgową do wyznaczenia wszystkich wartości własnych z dokładnością do 4 cyfr znaczących. Podaj przesunięcia i liczby iteracji oraz sprawdzenia: $\boldsymbol{q}_i \cdot \boldsymbol{q}_j = 0, i \neq j$.

Uwaga. W sprawozdaniu należy zawrzeć tylko postać wybranych funkcji czy macierzy oraz wyniki w postaci opisanych liczb, tabel błędów, i wykresów (bez skryptów i wydruków!).

Wypróbuj działanie wybranych funkcji MatLaba:

- Podstawienie:
  > $a = 3$ lub $a = 3;$
  > $b = \sin(a)$
- Definicja macierzy przez wyliczenie jej elementów:
  > $a = [1\ 3\ 8\ 4\ ;\ 5\ 1\ 3\ 2]$
- Definicja macierzy przez wygenerowanie:
  > $a = [1\ :\ 0.1\ :\ 1.5\ ;\ 0\ :\ 0.01\ :\ 0.05]$
- Największy/najmniejszy element wektora (macierzy):
  > $v = [1\ 2\ 6\ -3];$
  > $vmax = max(v)$
- Obliczenie wartości danej funkcji dla elementów macierzy:
  > $a = [1\ :\ 0.1\ :\ 10];$
  > $b = \sin(a)$
- Wyświetlenie 2 wykresów funkcji jednej zmiennej:
  > $x = [0\ :\ pi/20\ :\ 2 * pi];$
  > $y = \sin(x);$
  > $z = \cos(x);$
  > $\mathrm{plot}(x, y)$
  > hold on
  > $\mathrm{plot}(x, z, 'm')$
- Wyświetlenie wykresu funkcji dwu zmiennych:
  > clf
  > $[x, y] = meshgrid(-pi : 0.2\ :\ pi\ ,\ -pi : 0.2 : pi);$
  > $z = \sin(x). * \sin(y). * (x.\hat{}2 + y.\hat{}2);$
  > $mesh(x, y, z)$
- Obliczenie całki oznaczonej
  > $a = quad('sin', -1, 2)$ lub
  > $a = quad8('sin', -1, 2)$
- Znalezienie pierwiastka funkcji:
  > $x0 = fzero('sin', 2.6)$
- Pisanie i wywoływanie skryptu w tzw. m.file.
  Przykładowy skrypt:
  f = input('podaj wzor na f(x)=','s');
  a = input('podaj a=');
  b = input('podaj b=');
  n = input('podaj n=');
  dx = (b-a)/n;
  x = $[a\ :\ dx\ :\ b]$;
  y = sin(x);
  plot(x,y)
  Wywołanie skryptu:
  > script-name
- Rozwiązanie równania różniczkowego zwyczajnego:
  > ts = $[0\ :\ 0.1\ :\ 10]$;
  > y0 = $[1\ ;\ 2]$;
  > ode45('vdpol',ts,y0) lub
  > [t,y] = ode45('vdpol',ts,y0);
  > plot(ts,y)
  gdzie `vdpol.m` jest wcześniej napisanym skryptem postaci:
  function yprime=vdpol(t,y)
  mu=2;
  yprime = $[y(2)\ ;\ mu * (1 - y(1)\hat{}2) * y(2) - y(1)]$;

Wypróbuj działanie wybranych funkcji MatLaba:

- Dodawanie macierzy:
  $>\ a = [1\ 3\ 8\ 4\ ;\ 5\ 1\ 3\ 2]$
  $>\ b = [2\ 5\ 1\ 6\ ;\ 3\ 4\ 6\ 1]$
  $>\ a + b$
- Mnożenie macierzy:
  $>\ a = [1\ 3\ 5\ \ :\ 2\ 1\ 3]$
  $>\ b = [1\ 1\ 4\ \ :\ 2\ 2\ 1\ \ :\ 3\ \ 2\ 1]$
  $>\ a * b$
- Największy/najmniejszy element wektora (macierzy):
  $>\ v = [-1\ 3\ 5\ \ -2];$
  $>\ vmax = max(v)$
  $>\ vmin = min(v)$
- Funkcje macierzowe:
  $>\ a = [1\ 2\ 3\ ,\ 0\ 9\ 1\ ,\ 3\ 4\ 7];$
  Odwracanie:
  $>\ inv(a)$
  Transponowanie:
  $>\ a'$
  Wyznacznik:
  $>\ det(a)$
  Wartości własne:
  $>\ eig(a)$
  Przekątna macierzy:
  $>\ diag(a)$

%lagrange.m

```
f=input('podaj wzor f(x)=','s');
c=input('podaj c=');
d=input('podaj d=');
%Lagrange polynomials of 4th order
w1='(x-.25).*(x-.50).*(x-.75).*(x-1 )/(24/256)';
w2='(x- 0).*(x-.50).*(x-.75).*(x-1 )/(-6/256)';
w3='(x- 0).*(x-.25).*(x-.75).*(x-1 )/(4/256)';
w4='(x- 0).*(x-.25).*(x-.50).*(x-1 )/(-6/256)';
w5='(x- 0).*(x-.25).*(x-.50).*(x-.75)/(24/256)';
%find values of f(x) at sampling points:
x=[0:.25:1];
x=c+(d-c)*x;
a=eval(f);
%find values of Lagrangian interpolant of f(x):
x=[0:.01:1];
y=a(1).*eval(w1)+a(2).*eval(w2)+a(3).*eval(w3)...
+a(4).*eval(w4)+a(5).*eval(w5);
%find values of exact function f(x):
x=c+(d-c)*x;
z=eval(f);
%draw pictures:
plot(x,y)
hold on
plot(x,z,'m')
%find maximum error
y=abs(y-z);
max(y)
```

%bisect.m

```
f=input('podaj wzor na f(x)=','s');
a=input('podaj a=');
b=input('podaj b=');
eps=input('podaj eps=');
x=a;
f1=eval(f);
x=b;
f2=eval(f);
del=1000000;
while del > eps
x=(b+a)/2;
f3=eval(f);
if f1 * f3 < 0
b=x;
f2=f3;
else
a=x;
f1=f3;
end
del=b-a
end
x=x
```

%intpara.m

```
n=input('podaj n:');
a=input('podaj a:');
b=input('podaj b:');
f=input('podaj wzor f(x)=','s');
h=(b-a)/n;
s=0;
for i=1:n
x=a+(i-1)*h;
y=eval(f);
x=x+0.5*h;
t=eval(f);
x=x+0.5*h;
z=eval(f);
s=s+(y/6+2/3*t+z/6)*h;
end
s=s
```

%inttria

```
n=input('podaj n:');
a=input('podaj a:');
b=input('podaj b:');
f=input('podaj wzor f(x)=','s');
h=(b-a)/n;
s=0;
for i=1:n
x=a+(i-1)*h;
y=eval(f);
x=x+h;
z=eval(f);
s=s+(y+z)*0.5*h;
end
s=s
```

%newton.m

```
%newton
f=input('podaj wzor f(x)=','s');
df=input('podaj wzor na df/dx=','s');
x=input('podaj x0=');
eps=input('podaj eps=');
del=1000000;
while del > eps
x1=x-eval(f)/eval(df);
del=abs(x1-x)
x=x1;
end
format long
x=x
```

%siecz.m

```
f=input('podaj wzor f(x)=','s');
x1=input('podaj x1=');
x2=input('podaj x2=');
eps=input('podaj eps=');
x=x1;
f1=eval(f);
x=x2;
f2=eval(f);
del=1000000;
while del > eps
x=(x1*f2-x2*f1)/(f2-f1);
f3=eval(f);
if f1 * f3 < 0
del=abs(x2-x)
x2=x;
f2=f3;
else
del=abs(x1-x)
x1=x;
f1=f3;
end
end
x=x
```

% **jacobi.m**

```
clear
n=input('podaj n=');
a(1:n,1:n)=0
b(1:n,1:1)=0
display('podaj macierz a wierszami')
for i=1:n
for j=1:n
a(i,j)=input(' ');
end
end
display('podaj wektor b')
for i=1:n
b(i,1)=input(' ');
end
% pomocnicze macierze
c(1:n,1:n) = 0;
d(1:n,1:n) = 0;
for i=1:n
for j=1:n
if i==j
d(i,j)=a(i,j);
else
c(i,j)=a(i,j);
end
end
end
d=inv(d);
x(1:n,1:1)=10000000;
y(1:n,1:1)=0;
step=0;
% WLASCIWE ITERACJE
while(max(abs(x-y)) > 0.0000001);
max(abs(x-y))
x=y;
y=d*(b-c*x);
step=step+1
y
pause
end
```

%powermet.m

```
clear
n=input('podaj n=');
a(1:n,1:n)=0
x(1:n,1:1)=1
y(1:n,1:1)=0
display('podaj macierz a wierszami')
for i=1:n
for j=1:n
a(i,j)=input(' ');
end
end
lam1=0;
lam2=1;
while(abs(lam1-lam2)>0.0001)
lam1=lam2;
y=a*x;
lam2=x'*y
norm2=y'*y;
norm=sqrt(norm2);
x=y/norm;
pause
end
```

%gaussseidel.m

```
clear
n=input('podaj n=');
a(1:n,1:n)=0
b(1:n,1:1)=0
display('podaj macierz a wierszami')
for i=1:n
for j=1:n
a(i,j)=input(' ');
end
end
display('podaj wektor b')
for i=1:n
b(i,1)=input(' ');
end
% pomocnicze macierze
c(1:n,1:n) = 0;
d(1:n,1:n) = 0;
for i=1:n
for j=1:n
if i>=j
d(i,j)=a(i,j);
else
c(i,j)=a(i,j);
end
end
end
d=inv(d);
x(1:n,1:1)=10000000;
y(1:n,1:1)=0;
step=0;
% WLASCIWE ITERACJE
while(max(abs(x-y)) > 0.0000001);
max(abs(x-y))
x=y;
y=d*(b-c*x);
step=step+1
y
pause
end
```

%inverse shifted power method

```
clear
n=input('podaj n=');
a(1:n,1:n)=0
x(1:n,1:1)=1
y(1:n,1:1)=0
display('podaj macierz a wierszami')
for i=1:n
for j=1:n
a(i,j)=input(' ');
end
end
shift=input('pdaj shift=');
a=a-shift*eye(n);
a=inv(a);
lam1=0;
lam2=1;
while(abs(lam1-lam2)>0.0001)
lam1=lam2;
y=a*x;
lam2=x'*y;
lam = 1/lam2 + shift
norm2=y'*y;
norm=sqrt(norm2);
x=y/norm;
pause
end
```