Derivare si Integrare
Numerica

1 Derivare

1.1 Introducere

1.1.1 Notiuni generale

Fie o functie $f : [a, b] \to \mathbb{R}$. Dorim sa calculam, in mod aproximativ, derivatele functiei $f(x)$ in anumite puncte din intervalul $[a, b]$, cunoscand valorile functiei intr-un numar redus de puncte $(x_i, f(x_i))$, $i = \overline{0, n}$.

1.1.2 Motivatie

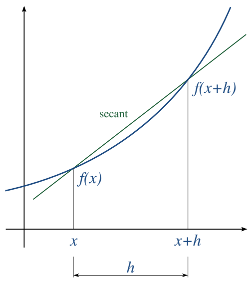
- In Computer Science se lucreaza in general cu date discrete si nu se cunoaste forma explicita a functiei. Astfel, trebuie aproximata valoarea derivatei in punctele de interes.
- Ne reamintim din capitolul de interpolare spline-urile cubice de clasa C^2 tensionate. Ultimele 2 conditii necesita cunoasterea derivatelor $f'(x_0)$ si $f'(x_n)$. De cele mai multe ori, aceste valori nu se cunosc, deci trebuie aproximate.

1.1.3 Idee

Pornim de la definita derivatei: $\boxed{f'(x_0) \overset{\text{def}}{=} \lim\limits_{h \to 0} \frac{f(x_0+h)-f(x_0)}{h}}$

Apoi, ne folosim de *interpolarea Lagrange* pentru a gasi niste aproximari mai bune.

Vom avea in vedere si *eroarea de aproximare* care intervine pe parcursul calculului.



1.1.4 Moduri de aproximare abordate

- Prima derivata:
  - $\boxed{\textbf{Two-Point Formulas}}$
  - $\boxed{\textbf{Three-Point Formulas}}$
- A doua derivata:
  - $\boxed{\textbf{Second Derivative Midpoint Formula}}$

1.1.5 Aplicatie care foloseste derivarea numerica

Edge Detection

Daca avem o *imagine* si dorim sa extragem *contururile* obiectelor continute in ea, trebuie sa calculam derivata imaginii[1]. Aceasta tehnica presupune aplicarea unui filtru (*kernel*) asupra imaginii originale. Ideea din spatele *matricei de convolutie* este *gradientul de imagine*[2].

Pentru a intelege mai bine unde este folosita derivarea numerica in cadrul procesului de extragere a contururilor, puteti viziona urmatorul clip (**Edge Detection Using Laplacian**): 🎞

[1] https://en.wikipedia.org/wiki/Image_derivatives

[2] https://en.wikipedia.org/wiki/Image_gradient#Computer_vision

1.1.6 Pregatirea terenului

Inainte de a discutia despre fiecare mod de aproximare a derivatelor, prezentam cadrul discutiei:

- Pornim de la o functie continua $f : [a,b] \to \mathbb{R}$ careia ii cunoastem valorile intr-un numar redus de puncte.
- Dorim sa gasim o aproximatie cat mai buna pentru valorile derivatelor functiei $f(x)$.

1.1.7 Estimarea erorii interpolarii polinomiale

Fie $f(x)$ cu $x \in [a,b]$ si un suport al interpolarii $S = [(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))]$.
Fie $P_n(x)$ polinomul de grad n care *interpoleaza* functia $f(x)$ pe $[a,b]$, adica $P_n(x_i) = f(x_i), \forall\ i = \overline{0,n}$.

Functia eroare[3] se defineste astfel: $\boxed{e(x) = f(x) - P(x)}$, $x \in [a,b]$ si $\begin{cases} f(x) = valoarea\ reala \\ P_n(x) = valoarea\ aproximata \end{cases}$

De acum inainte, tot ce este marcat cu rosu, face parte din termenii *erorii de aproximatie.*

Teorema: Fie $f(x)$ o functie de clasa C^{n+1} pe intervalul $[a,b]$ si $P_n(x)$ un polinom de grad cel mult n care interpoleaza functia f in $n+1$ puncte distincte din $[a,b]$. Atunci, pentru orice $x \in [a,b]$, exista un punct $\xi_x \in [a,b]$, astfel incat eroare de aproximare se poate exprima.

Matematic, aceasta are forma: $\boxed{e(x) = \frac{1}{(n+1)!} \cdot f^{(n+1)}(\xi_x) \cdot \prod_{i=0}^{n} (x - x_i)}$

1.2 Prima derivata

Asa cum am vazut in capitolul de interpolare, polinoamele au avantajul ca se pot deriva (si integra) foarte usor. Asadar, putem gasi niste aproximari ale derivatei $f'(x)$, ajutandu-ne de o tehnica de interpolare.

Inainte de a prezenta moduri de aproximare ale derivatei, reamintim faptul ca atunci cand gradul polinomului de interpolare este mare, apar oscilatii in capete si eroarea de aproximare creste foarte mult.

Deci, trebuie sa mentinem gradul polinomului de interpolare mic. In cele ce urmeaza, vom trata cazurile in care suportul interpolarii este format din 2 sau 3 puncte.

1.2.1 2 puncte (n=1)

Modul de determinare

Inainte de toate, pentru a aproxima $f'(x_0)$, facem ipotezele: $\begin{cases} x_0 \in (a,b) \\ f \in C^2[a,b] \\ x_1 = x_0 + h, \text{ cu } h \text{ suficient de mic, a.î. } x_1 \in (a,b) \end{cases}$

Asadar, suportul interpolarii arata astfel: $S = [(x_0, f(x_0)), (x_1, f(x_1))]$
Aproximam functia $f(x)$ cu ajutorul polinomului de interpolare *Lagrange* (de grad $n = 1$):
$f(x) \simeq P_1(x)$, $x \in (x_0, x_1)$ (Punem conditia ca x sa se afle intre x_0 si x_1 pentru ca interpolarea facuta are *valabilitate*[4] doar intre cele 2 puncte $\equiv$ doar in suportul interpolarii)

Interpoland Lagrange intre punctele x_0 si x_1, obtinem: $P_1(x) = f(x_0) \cdot \frac{x-x_1}{x_0-x_1} + f(x_1) \cdot \frac{x-x_0}{x_1-x_0}$
Derivam expresia obtinuta: $P_1'(x) = f(x_0) \cdot \frac{1}{x_0-x_1} + f(x_1) \cdot \frac{1}{x_1-x_0}$
Tinem cont de faptul ca $x_1 = x_0 + h$ (ipoteza initiala) si substituim in expresia de mai sus.
Astfel, gasim ca: $P_1'(x) = f(x_0) \cdot \frac{-1}{h} + f(x_0+h) \cdot \frac{1}{h} \iff \boxed{P_1'(x) = \frac{f(x_0+h)-f(x_0)}{h}}$

Pornim de la functia eroare $\boxed{e(x) = f(x) - P_1(x)}$ si ne folosim de *teorema prezentata anterior*:
$f(x) = P_1(x) + \frac{1}{2!} \cdot f''(\xi(x)) \cdot (x-x_0)(x-x_1)\ |()' \Rightarrow f'(x) = P_1'(x) + \frac{d}{dx}[\frac{(x-x_0)(x-x_1)}{2} \cdot f''(\xi(x))]$
Dar, $P_1'(x) = \frac{f(x_0+h)-f(x_0)}{h}$ (reamintim ca $x_1 = x_0+h$) $\Rightarrow f'(x) = \frac{f(x_0+h)-f(x_0)}{h} + \frac{d}{dx}[\frac{(x-x_0)(x-x_0-h)}{2} \cdot f''(\xi(x))]$
$\Rightarrow \boxed{f'(x) = \frac{f(x_0+h)-f(x_0)}{h} + \frac{2(x-x_0)-h}{2} \cdot f''(\xi(x)) + \frac{(x-x_0)(x-x_0-h)}{2} \cdot \frac{d}{dx}[f''(\xi(x))]}$ **(1)**

[3] https://en.wikipedia.org/wiki/Polynomial_interpolation#Interpolation_error
[4] https://stats.stackexchange.com/a/418898

Eliminand termenii care contin expresia $\xi(x)$ (termenii scrisi cu rosu), obtinem urmatoarea aproximatie: $\boxed{f'(x) \simeq \frac{f(x_0+h)-f(x_0)}{h},\ \forall x \in [x_0, x_0+h]}$

O problema cu formula **(1)** este ca nu avem informatii despre $\frac{d}{dx}[f''(\xi(x))]$ si astfel nu putem sa estimam eroarea de aproximare.

Dar, cand facem $x = x_0$, coeficientul lui $\frac{d}{dx}[f''(\xi(x))]$ devine 0 si formula se reduce la:

Two-Point Formula

$\boxed{f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{h}{2} \cdot f''(\xi)}$ (ξ se afla intre x_0 si x_0+h) **(*)**

Formula **(*)** este o metoda de **ordin 1** [5] si poarta denumirile: $\begin{cases} \textbf{forward-difference formula},\ h > 0 \\ \textbf{backward-difference formula},\ h < 0 \end{cases}$

Pentru *valori mici* ale lui h, aproximatia $f'(x_0) \simeq \frac{f(x_0+h)-f(x_0)}{h}$ se face cu o eroare de ordinul $M \cdot \frac{|h|}{2}$, unde M este maximul atins de functia $|f''(\xi(x))|$ pe intervalul $[x_0, x_0+h]$.

Exemplu numeric

Fie functia $f(x) = ln(x)$. Aproximati $f'(1.8)$ folosind, pe rand, pasul $\begin{cases} h = 0.1 \\ h = 0.05 \\ h = 0.01 \end{cases}$
Determinati si limitele erorii de aproximare.

Folosim *forward-difference formula*: $f'(x_0) \simeq \frac{f(x_0+h)-f(x_0)}{h} \iff f'(1.8) \simeq \frac{f(1.8+h)-f(1.8)}{h}$

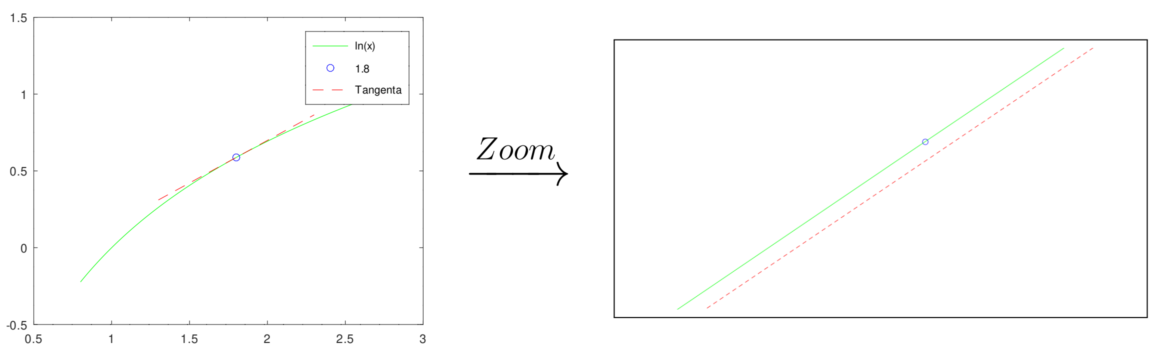
Prezentam rezolvarea completa pentru pasul $h = 0.1$:
$f'(1.8) \simeq \frac{f(1.8+0.1)-f(1.8)}{0.1} \iff f'(1.8) \simeq \frac{ln(1.9)-ln(1.8)}{0.1} \iff \boxed{f'(1.8) \simeq 0.5406722}$

Pentru limitele erorii de aproximare, avem: $f''(x) = -\frac{1}{x^2}$ si $1.8 < \xi < 1.9 \xRightarrow{(*)}$
$\Rightarrow err_{h=0.1} = \frac{|h \cdot f''(\xi)|}{2} = \frac{|h|}{2\xi^2} < \frac{0.1}{2 \cdot (1.8)^2} \Rightarrow \boxed{err_{h=0.1} < 0.0154321}$

Analog se procedeaza si pentru $h = 0.05$ si $h = 0.01$
Rezultatele sunt prezentate in tabelul alaturat:

Pasul	Aproximatia	Limita erorii
h	$\frac{f(1.8+h)-f(1.8)}{h}$	$\frac{\|h\|}{2 \cdot (1.8)^2}$
0.1	0.5406722	0.0154321
0.05	0.5479795	0.0077160
0.01	0.5540180	0.0015432

Derivata functiei $f(x) = ln(x)$ este $f'(x) = \frac{1}{x}$, $\forall x > 0$, deci valoarea exacta a derivatei functiei $f(x)$ in punctul 1.8 este $f'(1.8) = \frac{1}{1.8} = 0.55\overline{5}$. In acest caz, am obtinut aproximatii foarte bune, chiar si cu cel mai mare pas. Insa, cu cat pasul h este mai mic, cu atat vom obtine aproximatii mai bune.



Ce putem imbunatati?

Pana acum, am lucrat cu 2 puncte in suportul interpolarii. Teoretic, cu cat avem mai multe informatii, cu atat putem sa aproximam mai bine. In continuare, ne indreptam atentia catre formule de aproximare care implica 3 puncte in suportul de interpolare.

[5] https://en.wikipedia.org/wiki/Order_of_accuracy

1.2.2 3 puncte (n=2)

Modul de determinare

In cele ce urmeaza, lucram cu 3 puncte in suportul interpolarii. Formulele pe care le vom deduce sunt foarte utile daca cele 3 puncte sunt *echidistante*.

Asadar, vom considera: $\begin{cases} x_0 = x_0 \\ x_1 = x_0 + h \quad\quad (h \neq 0) \\ x_2 = x_0 + 2h \end{cases}$

Modeland putin *teorema privind estimarea erorii*, obtinem urmatoarea formula *generala* de aproximare a derivatei I:

$$\boxed{f'(x_j) = \sum_{k=0}^{n} f(x_k)L'_k(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \prod_{\substack{k=0 \\ k\neq j}}^{n} (x_j - x_k)}$$ (s.n. **(n+1)-point formula**)

In general, folosind mai multe puncte pentru a aproxima $f'(x_j)$ in ecuatia dedusa anterior, vom obtine o acuratete mai mare, cu toate ca pentru un numar mare de evaluari (multe puncte), eroarea de rotunjire[6] creste.

Observatie: Cele mai utilizate formule sunt cele care implica 1, 3 sau 5 puncte. In continuare, vom trata cazul cu 3 puncte:

Inainte de toate, pentru a aproxima $f'(x_0)$, facem ipotezele: $\begin{cases} x_0 \in (a,b) \\ f \in C^2[a,b] \\ x_1 = x_0 + h, \text{ cu } h \text{ suficient de mic, a.î. } x_1 \in (a,b) \\ x_2 = x_0 + 2h, \text{ cu } h \text{ suficient de mic, a.î. } x_2 \in (a,b) \end{cases}$

Asadar, suportul interpolarii arata astfel: $S = [(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))]$

Aproximam functia $f(x)$ cu ajutorul polinomului de interpolare *Lagrange* (de grad $n = 2$):

$f(x) \simeq P_2(x)$ cu $x \in (x_0, x_2)$ si $P_2(x) = f(x_0) \cdot L_0(x) + f(x_1) \cdot L_1(x) + f(x_2) \cdot L_2(x)$

Pornim de la *multiplicatorii Lagrange*: $\begin{cases} L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \\ L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \\ L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \end{cases} \overset{()'}{\Longrightarrow} \begin{cases} L'_0(x) = \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} \\ L'_1(x) = \frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)} \\ L'_2(x) = \frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)} \end{cases}$

Derivand polinomul de interpolare Lagrange, obtinem:

$P'_2(x) = f(x_0) \cdot L'_0(x) + f(x_1) \cdot L'_1(x) + f(x_2) \cdot L'_2(x)$

Particularizam **(n+1)-point formula** pentru $n = 2$:

$$f'(x_j) = f(x_0)\cdot[\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}] + f(x_1)\cdot[\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}] + f(x_2)\cdot[\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}] + \frac{f^{(3)}(\xi_j)}{3!} \prod_{\substack{k=0 \\ k\neq j}}^{2} (x_j - x_k)$$

Am obtinut formula generala pentru 3 puncte in suportul interpolarii.

Reamintim ipotezele legate de puncte ($x_j = x_0,\ x_1 = x_0 + h$ si $x_2 = x_0 + 2h$) si substituim in expresia obtinuta anterior. $\boxed{f'(x_0) = \frac{1}{h}[-\frac{3}{2}f(x_0) + 2f(x_0 + h) - \frac{1}{2}f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0)}$

Analog pentru $x_j = x_1$: $\boxed{f'(x_0 + h) = \frac{1}{h}[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0 + 2h)] - \frac{h^2}{6}f^{(3)}(\xi_1)}$

Analog pentru $x_j = x_2$: $\boxed{f'(x_0 + 2h) = \frac{1}{h}[\frac{1}{2}f(x_0) - 2f(x_0 + h) + \frac{3}{2}f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_2)}$

In a doua ecuatie, se face schimbarea de variabila $x_0 + h \to x_0$, pentru a determina o aproximare a derivatei in punctul x_0. Analog in ultima ecuatie: $x_0 + 2h \to x_0$.

In final, se obtin urmatoarele 3 formule: $\begin{cases} f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0) \\ f'(x_0) = \frac{1}{2h}[-f(x_0 - h) + f(x_0 + h)] - \frac{h^2}{6}f^{(3)}(\xi_1) \\ f'(x_0) = \frac{1}{2h}[f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)] + \frac{h^2}{3}f^{(3)}(\xi_2) \end{cases}$

Este usor de observat ca daca substituim h cu $-h$ in prima ecuatie, obtinem ultima ecuatie.

[6] https://bit.ly/3k1hJas

Practic, raman doar 2 formule:

Three-Point Endpoint Formula

$\boxed{f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0+h) - f(x_0+2h)] + \frac{h^2}{3}f^{(3)}(\xi)}$ (ξ se afla intre x_0 si x_0+2h) **(**)**

Three-Point Midpoint Formula

$\boxed{f'(x_0) = \frac{1}{2h}[f(x_0+h) - f(x_0-h)] - \frac{h^2}{6}f^{(3)}(\xi)}$ (ξ se afla intre x_0-h si x_0+h) **(***)**

Cu toate ca ambele formule **(**)** si **(***)** sunt metode de **ordin 2**, eroarea din ecuatia (***) este aproximativ *jumatate* din eroarea provenita in ecuatia (**), deoarece formula (***) foloseste informatii din ambele parti ale punctului x_0, pe cand (**) foloseste puncte doar dintr-o parte.

Observatie: Ecuatia (**) se dovedeste ceva mai utila atunci cand se doreste aproximarea derivatei in apropierea capetelor intervalului, deoarece informatii despre functia f in afara intervalului, de obicei, nu sunt cunoscute.

Exemplu numeric

Fie functia $f(x) = xe^x$. Aproximati $f'(2)$, cunoscand urmatorul suport de interpolare:

x	f(x)
1.8	10.889365
1.9	12.703199
2.0	**14.778112**
2.1	17.148957
2.2	19.855030

Suportul de interpolare ne permite sa aproximam $f'(2)$ in 4 moduri diferite:

- Folosind **Three-Point Endpoint (**)** cu pasul: $\begin{cases} h = 0.1 \\ \text{sau} \\ h = -0.1 \end{cases}$
- Folosind **Three-Point Midpoint (***)** cu pasul: $\begin{cases} h = 0.1 \\ \text{sau} \\ h = 0.2 \end{cases}$

Observatie: Pentru limitele erorii de aproximare, tinem cont de faptul ca $f^{(3)}(x) = (x+2)e^x$

Folosind formula **(**)** cu pasul $\mathbf{h = 0.1}$, obtinem:

$f'(2) = \frac{1}{2\cdot 0.1}[-3f(2) + 4f(2+0.1) - f(2+2\cdot 0.1)] \Rightarrow \boxed{f'(2) = 22.032310}$ ($\xi \in (2.0, 2.2)$)

$err(**)_{h=0.1} = \frac{|h^2 \cdot f^{(3)}(\xi)|}{3} = \frac{|h^2| \cdot (\xi+2)e^\xi}{3} < \frac{(0.1)^2 \cdot (2.2+2)e^{2.2}}{3} \Rightarrow \boxed{err(**)_{h=0.1} < 0.126}$

Folosind **(**)** cu pasul $\mathbf{h = -0.1}$, obtinem:

$f'(2) = \frac{1}{2\cdot -0.1}[-3f(2) + 4f(2-0.1) - f(2-2\cdot 0.1)] \Rightarrow \boxed{f'(2) = 22.054525}$ ($\xi \in (1.8, 2.0)$)

$err(**)_{h=-0.1} = \frac{|h^2 \cdot f^{(3)}(\xi)|}{3} = \frac{|h^2| \cdot (\xi+2)e^\xi}{3} < \frac{(-0.1)^2 \cdot (2.0+2)e^{2}}{3} \Rightarrow \boxed{err(**)_{h=-0.1} < 0.098}$

Folosind formula **(***)** cu pasul $\mathbf{h = 0.1}$, obtinem:

$f'(2) = \frac{1}{2\cdot 0.1}[f(2+0.1) - f(2-0.1)] \Rightarrow \boxed{f'(2) = 22.228790}$ ($\xi \in (1.9, 2.1)$)

$err(***)_{h=0.1} = \frac{|h^2 \cdot f^{(3)}(\xi)|}{6} = \frac{(0.1)^2 \cdot (2.1+2)e^{2.1}}{6} \Rightarrow \boxed{err(***)_{h=0.1} < 0.055}$

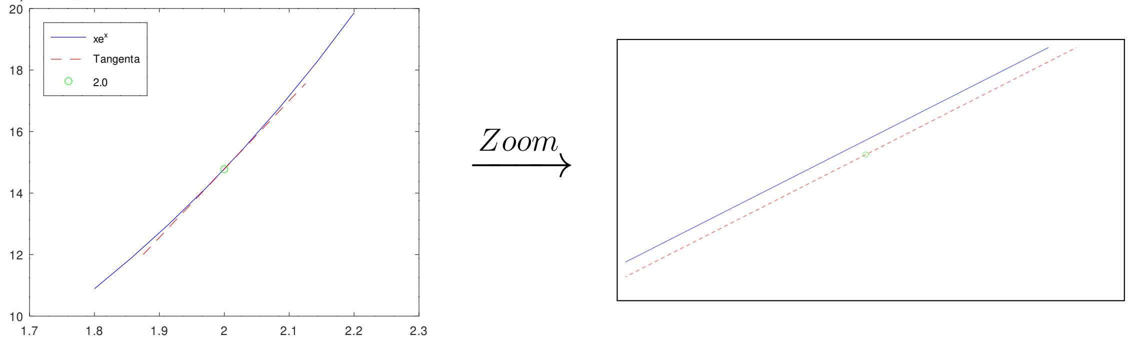
Folosind formula **(***)** cu pasul $\mathbf{h = 0.2}$, obtinem:

$f'(2) = \frac{1}{2\cdot 0.2}[f(2+0.2) - f(2-0.2)] \Rightarrow \boxed{f'(2) = 22.414163}$ ($\xi \in (1.8, 2.2)$)

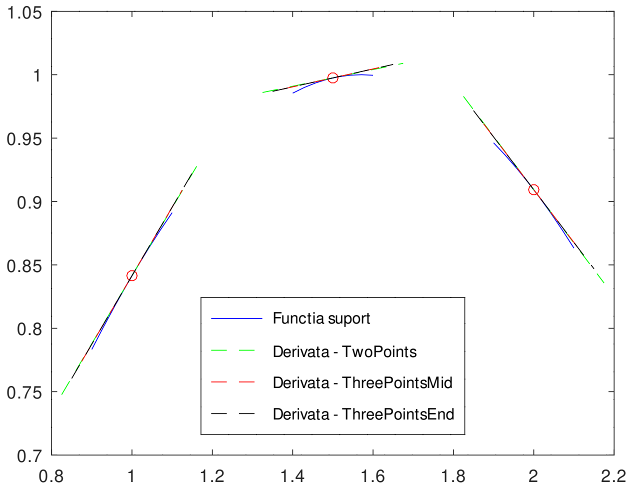
$err(***)_{h=0.2} = \frac{|h^2 \cdot f^{(3)}(\xi)|}{6} = \frac{(0.1)^2 \cdot (2.2+2)e^{2.2}}{6} \Rightarrow \boxed{err(***)_{h=0.2} < 0.063}$

Derivata functiei $f(x) = xe^x$ este $f'(x) = (x+1)e^x$, deci valoarea *exacta* a derivatei functiei $f(x)$ in punctul 2.0 este $f'(2.0) = (2+1)e^2 = \boxed{22.167168}$

Comparand cele 4 aproximatii, observam ca cea mai apropiata de valoarea exacta este formula (***) cu pasul $h = 0.1$



1.2.3 Concluzii prima derivata



In figura alaturata se pot observa interpretarile grafice ale primei derivate obtinute prin cele 3 formule determinate anterior: Two-Point(*), Three-Point-Endpoint(**), Three-Point-Midpoint(***)

Privind *graficul*, pare ca cele 3 metode conduc la aceeasi tangenta, insa daca analizam lucrurile din punct de vedere *numeric*, observam ca metodele au diferite grade de acuratete.

$$\text{Erorile de aproximatie: } \begin{cases} (*) : 7.95 \cdot 10^{-3} \\ (**) : 2.28 \cdot 10^{-5} \\ (***) : 1.14 \cdot 10^{-5} \end{cases}$$

Asadar, pentru functia suport, metoda (***) are cea mai buna acuratete.

1.3 A doua derivata

Modul de determinare

In continuare, dorim sa aproximam $f''(x_0)$. Pornim de la dezvoltarea in serie Taylor[7] a functiei $f(x)$ in jurul punctului x_0: $f(x) = f(x_0) + (x-x_0)f'(x_0) + \frac{(x-x_0)^2}{2!}f''(x_0) + \frac{(x-x_0)^3}{3!}f^{(3)}(x_0) + \ldots$

Pastram din seria Taylor, primii 3 termeni:
$f(x) = f(x_0) + (x-x_0)f'(x_0) + \frac{(x-x_0)^2}{2!}f''(x_0) + \frac{(x-x_0)^3}{3!}f^{(3)}(x_0) + \frac{(x-x_0)^4}{4!}f^{(4)}(\xi)$

Facand, pe rand, $\boxed{x \to x_0 + h}$ si $\boxed{x \to x_0 - h}$, obtinem:

$$\begin{cases} f(x_0+h) = f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2 + \frac{1}{6}f^{(3)}(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_1)h^4 \\ f(x_0-h) = f(x_0) - f'(x_0)h + \frac{1}{2}f''(x_0)h^2 - \frac{1}{6}f^{(3)}(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_{-1})h^4 \end{cases}$$

Observatie: $x_0 - h < \xi_{-1} < x_0 < \xi_1 < x_0 + h$

Insumam cele 2 relatii membru cu membru si obtinem:
$f(x_0+h) + f(x_0-h) = 2f(x_0) + \boxed{f''(x_0)}h^2 + \frac{1}{24}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]h^4$

$$\Longrightarrow \boxed{f''(x_0) = \frac{1}{h^2}[f(x_0-h) - 2f(x_0) + f(x_0+h)] - \frac{h^2}{24}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]}$$

Presupunem ca $f^{(4)}$ este continua pe $[x_0-h, x_0+h]$.
Este evident ca $\frac{1}{2}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]$ se afla intre $f^{(4)}(\xi_1)$ si $f^{(4)}(\xi_{-1})$.

[7] https://en.wikipedia.org/wiki/Taylor_series#Definition

Conform Toeremei de Medie[8], exista o valoare ξ intre ξ_{-1} si ξ_1 si implicit in intervalul $[x_0 - h, x_0 + h]$, a.î. $f^4(\xi) = \frac{1}{2}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]$. Astfel, formula de aproximare a derivatei a II-a, se reduce la:

Second Derivative Midpoint Formula

$$\boxed{f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{12} f^{(4)}(\xi)}, \text{ cu } \xi \in (x_0 - h, x_0 + h) \ \textbf{(****)}$$

Astfel, am determinat formula cu care putem aproxima derivata a II-a a unei functii f. Privind puterea lui h, tragem concluzia ca aceasta este o metoda de **ordin 2**.

Exemplu numeric

Fie functia $f(x) = xe^x$. Aproximati $f''(2)$, cunoscand urmatorul suport de interpolare:

x	f(x)
1.8	10.889365
1.9	12.703199
2.0	**14.778112**
2.1	17.148957
2.2	19.855030

Suportul de interpolare ne permite sa aproximam $f''(2)$ in 2 moduri diferite:

Folosind **Second Derivative MidPoint (****)** cu pasul: $\begin{cases} h = 0.1 \\ \quad \text{sau} \\ h = 0.2 \end{cases}$

Observatie: Pentru limitele erorii de aproximare, tinem cont de faptul ca $f^{(4)}(x) = (x + 3)e^x$

Folosind pasul $\mathbf{h = 0.1}$, obtinem:

$$f''(2) = \frac{1}{(0.1)^2}[f(2 - 0.1) - 2f(2) + f(2 + 0.1)] \Rightarrow \boxed{f''(2) = 29.593200} \ (\xi \in (1.9, 2.1))$$

$$err(****)_{h=0.1} = \frac{|h^2 \cdot f^{(4)}(\xi)|}{12} = \frac{|h^2| \cdot (\xi+3)e^{\xi}}{12} < \frac{(0.1)^2 \cdot (2.1+3)e^{2.1}}{12} \Rightarrow \boxed{err(****)_{h=0.1} < 0.034}$$

Folosind pasul $\mathbf{h = 0.2}$, obtinem:

$$f''(2) = \frac{1}{(0.2)^2}[f(2 - 0.2) - 2f(2) + f(2 + 0.2)] \Rightarrow \boxed{f''(2) = 29.704275} \ (\xi \in (1.8, 2.2))$$

$$err(****)_{h=0.2} = \frac{|h^2 \cdot f^{(4)}(\xi)|}{12} = \frac{|h^2| \cdot (\xi+3)e^{\xi}}{12} < \frac{(0.2)^2 \cdot (2.2+3)e^{2.2}}{12} \Rightarrow \boxed{err(****)_{h=0.2} < 0.156}$$

Derivata II a functiei $f(x) = xe^x$ este $f''(x) = (x + 2)e^x$, deci valoarea *exacta* a derivatei II a functiei $f(x)$ in punctul 2.0 este $f''(2.0) = (2+2)e^2 = \boxed{29.556224}$. Comparand cele 2 aproximatii, observam ca cea mai apropiata de valoarea exacta este formula (****) cu pasul $h = 0.1$

1.4 Rezumat formule derivare

Two-Point (*)

$$\boxed{f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} \cdot f''(\xi)} \ (\xi \text{ se afla intre } x_0 \text{ si } x_0 + h)$$

Three-Point-Endpoint (**)

$$\boxed{f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3} f^{(3)}(\xi)} \ (\xi \text{ se afla intre } x_0 \text{ si } x_0 + 2h)$$

Three-Point-Midpoint (***)

$$\boxed{f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6} f^{(3)}(\xi)} \ (\xi \text{ se afla intre } x_0 - h \text{ si } x_0 + h)$$

Second Derivative Midpoint Formula (****)

$$\boxed{f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{12} f^{(4)}(\xi)}, \text{ cu } \xi \in (x_0 - h, x_0 + h)$$

[8] https://en.wikipedia.org/wiki/Intermediate_value_theorem#Theorem

2 Integrare

2.1 Introducere

2.1.1 Notiuni generale

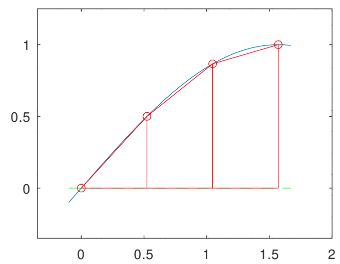
Fie o functie $f : [a, b] \to \mathbb{R}$. Dorim sa calculam, in mod aproximativ, integrala $\int_a^b f(x)\,dx$, cunoscand valorile functiei intr-un numar redus de puncte $(x_i, f(x_i))$, $i = \overline{0, n}$.

2.1.2 Motivatie

Unele integrale sunt dificil de calculat folosind metode analitice. Astfel, introducem integrarea numerica prin **cuadraturi**.[9]

2.1.3 Idee

- Impartim intervalul $[a, b]$ in subintervale.
- Pe fiecare subinterval, aproximam functia $f(x)$ cu un polinom $P_i(x)$, obtinut prin *interpolare*.
- Integram $P_i(x)$ pe fiecare subinterval si insumam rezultatele.



2.1.4 Moduri de aproximare abordate

- **Regula trapezelor**
- **Regula Simpson**
- **Cuadraturi adaptive**

2.1.5 Aplicatii care folosesc integrarea numerica

Aplicatii precum calcularea de *arii*, *volume*, *centrul de masa* sunt evidente pentru integrarea numerica. Vom prezenta, pe scurt, utilizarea integrarii numerica in studiul *probabilitatilor*.

Studiul probabilitatilor

Dupa cum stiti, in studiul probabilitatilor, integralele definite sunt foarte intalnite.

De exemplu, pentru a determina *functia de repartitie*,[10] se poate calcula direct integrala definita.

Pentru mai multe detalii, puteti lectura materialul:

2.1.6 Pregatirea terenului

Inainte de a discutia despre fiecare mod de aproximare a integralei $\int_a^b f(x)\,dx$, prezentam cadrul discutiei:

- Pornim de la o functie continua $f : [a, b] \to \mathbb{R}$ careia ii cunoastem valorile intr-un numar redus de puncte.
- Dorim sa gasim o aproximatie cat mai buna a integralei definite $\int_a^b f(x)\,dx$.

Alegem $(n + 1)$ puncte *echidistante* (pentru metodele *Newton-Cotes*) in intervalul $[a, b]$:

$$\boxed{x_i = a_i + i \cdot h}, \begin{cases} i = 0 : n \\ h = \frac{b-a}{n} \text{ (lungimea subintervalului)} \end{cases}$$

Ideea de cuadratura numerica este data de aproximarea integralei definite printr-o *suma*:

$$\boxed{\int_a^b f(x)\,dx \simeq \sum_{i=0}^{n} a_i f(x_i)}$$

[9] https://en.wikipedia.org/wiki/Numerical_integration

[10] https://en.wikipedia.org/wiki/Cumulative_distribution_function

Ne propunem sa determinam coeficientii a_i. Aproximam functia $f(x)$ cu polinomul de interpolare Lagrange, metinand gradul n cat mai mic: $f(x) \simeq P_n(x)$

$P_n(x) \overset{\text{def}}{=} \sum_{i=0}^{n} f(x_i) L_i(x)$.

Asadar, facand aproximatia $f(x) \simeq P_n(x)$ si integrand in ambii membri, obtinem:

$\int_a^b f(x)\,dx \simeq \int_a^b \sum_{i=0}^{n} f(x_i) L_i(x)\,dx \iff \int_a^b f(x)\,dx \simeq \sum_{i=0}^{n} f(x_i) \cdot \boxed{\int_a^b L_i(x)\,dx}$

Dar, $\int_a^b f(x)\,dx \simeq \sum_{i=0}^{n} f(x_i) \cdot \boxed{a_i}$

$\xRightarrow[identificare]{Prin} \boxed{a_i = \int_a^b L_i(x)\,dx}, \; i = 0 : n$

Asadar, pentru a aproxima $\int_a^b f(x)\,dx$, alegem coeficientii a_i astfel: $a_i = \int_a^b L_i(x)\,dx$

Reamintim faptul ca $L_i(x)$ s.n. *multiplicatori Lagrange* si se calculeaza astfel: $\boxed{L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \; j = \overline{0, n}}$

In continuare, vom trata pe rand cazurile $\begin{cases} n = 1 \text{ (Trapez)} \\ n = 2 \text{ (Simpson)} \end{cases}$

2.2 Metode Newton-Cotes

2.2.1 Regula simpla a trapezelor (n=1)

Modul de determinare

Dorim sa aproximam $\int_a^b f(x)\,dx$, folosind o interpolare polinomiala Lagrange de grad $n = 1$ (interpolare liniara). Impartim intervalul $[a, b]$ intr-un subinterval ($n = 1$).

Atunci, abscisele vor fi: $\boxed{x_0 = a}$ si $\boxed{x_1 = b}$, iar lungimea subintervalului: $\boxed{h = \frac{b-a}{1} = b - a}$

Asa cum am discutat, aproximam functia f printr-un polinom de grad $n = 1$: $f(x) \simeq P_1(x)$

$P_1(x) = f(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + f(x_1) \cdot \frac{x - x_0}{x_1 - x_0}$

$\xRightarrow{|\int_a^b} \int_a^b f(x)\,dx \simeq \int_a^b P_1(x)\,dx \iff \int_a^b f(x)\,dx \simeq \int_a^b [f(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + f(x_1) \cdot \frac{x - x_0}{x_1 - x_0}]\,dx \overset{(\dots)}{=} \frac{h}{2}[f(a) + f(b)]$

Astfel, am determinat formula pentru *Regula Simpla a Trapezelor*:

Regula Simpla a Trapezelor

$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{2}[f(a) + f(b)]}$ **(*)** (reamintim faptul ca $h = b - a$)

Interpretarea geometrica

Cand functia f ia valori pozitive pe $[a, b]$, integrala $\int_a^b f(x)\,dx$ este aproximata cu *aria unui trapez*. De aici si denumirea de *Regula Trapezelor*.

Formula (*) poate fi determinata si geometric - Aproximam integrala cu aria trapezului:

$\int_a^b f(x)\,dx \simeq \frac{[Baza_{mare} + Baza_{mica}] \cdot Inaltimea}{2} \iff \int_a^b f(x)\,dx \simeq \frac{[f(x_1) + f(x_0)] \cdot [x_1 - x_0]}{2}$

Exemplu numeric

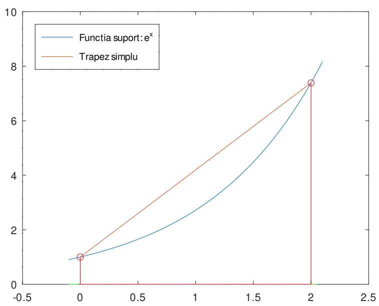
Fie functia $f(x) = e^x$, $x \in [0, 2]$. Aproximati $\int_0^2 f(x)\,dx$, folosind *Regula Simpla a Trapezelor*.

Pur si simplu, aplicam formula (*): $\int_0^2 f(x)\,dx \simeq \frac{h}{2} \cdot [f(0) + f(2)]$, unde $h = b - a = 2 - 0 = 2$.

Deci, $\int_0^2 f(x)\,dx \simeq \frac{2}{2} \cdot [e^0 + e^2] \Longrightarrow \boxed{\int_0^2 f(x)\,dx \simeq 8.389}$

Observatie: Valoarea exacta a integralei este $\int_0^2 f(x)\,dx = 6.389$

Asadar, aproximatia cu trapez simplu nu este cea mai buna.



Ce putem imbunatati?

In loc sa integram pe tot intervalul, putem sa impartim problema in probleme mai mici, aplicand regula simpla a trapezelor pe subintervale.

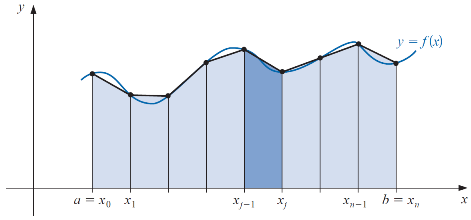
2.2.2 Regula compusa a trapezelor

Modul de determinare

Regula simpla a trapezelor discutata anterior, este utila atunci cand intergram o singura data, pe tot intervalul $[a, b]$.

Pentru a ne apropia de o aproximatie mai buna a integralei $\int_a^b f(x)\,dx$, trebuie sa divizam problema in sub-probleme:

Impartim intervalul $[a, b]$ in n subintervale de *lungimi egale*: $x_i = a + i \cdot h,\ i = 0 : n$ si $h = \frac{b-a}{n}$



Atunci, $\int_a^b f(x)\,dx = \int_{x_0}^{x_n} f(x)\,dx = \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)\,dx$

Apoi, aplicam regula simpla a trapezelor pe fiecare subinterval.

Astfel, integrala se poate aproxima: $\int_a^b f(x)\,dx \simeq \sum\limits_{i=0}^{n-1} \frac{h}{2} \cdot [f(x_i) + f(x_{i+1})]$

$\Rightarrow \int_a^b f(x)\,dx \simeq \frac{h}{2} \cdot [f(x_0) + f(x_1) + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + f(x_n)]$

$\Rightarrow \int_a^b f(x)\,dx \simeq \frac{h}{2} \cdot [f(x_0) + f(x_n) + 2\sum\limits_{i=1}^{n-1} f(x_i)]$

Astfel, am determinat formula pentru *Regula Compusa a Trapezelor*:

Regula Compusa a Trapezelor

$$\boxed{\int_a^b f(x)\,dx \simeq h \cdot [\frac{f(x_0)+f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i)] \overset{\text{not}}{=} T(f;h)},\ h = \frac{b-a}{n} \quad \textbf{(**)}$$

Avantajul gruparii termenilor in aceasta forma este ca se poate **vectoriza**.[11]

Analiza erorii

Facem urmatoarele notatii: $\begin{cases} T(f;h) \equiv \text{aproximatia integralei folosind regula compusa a trapezelor} \\ I(f) \equiv \text{valoarea exacta a integralei} \\ E_T(f;h) \equiv \text{eroarea trapezoidala pentru functia } f \text{ cu pas } h \end{cases}$

Eroarea totala se poate scrie astfel: $E_T(f;h) = I(f) - T(f;h) = \sum\limits_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} [f(x) - P_1(x)]\,dx$

$\Rightarrow E_T(f;h) = \sum\limits_{i=0}^{n-1} E_{T,i}(f;h)$, unde cu $E_{T,i}(f;h)$ am notat eroarea pe subintervalul $[x_i, x_{i+1}]$.

Practic, este eroarea provenita in urma interpolarii polinomiale pe subintervalul i.

Confrom *teoremei de estimare a erorii*:

$E_{T,i}(f;h) = \int_{x_i}^{x_{i+1}} \frac{1}{2} f''(\xi_i)(x - x_i)(x - x_{i+1})\,dx = \frac{1}{2} f''(\xi_i) \cdot \int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1})\,dx$

$\xRightarrow{(\ldots)} E_{T,i}(f;h) = -\frac{1}{12} h^3 \cdot f''(\xi_i),\ \xi_i \in (x_i, x_{i+1})$

Insumam erorile *locale* si obtinem eroarea *globala*: $E_T(f;h) = \sum\limits_{i=0}^{n-1} E_{T,i}(f;h) = \sum\limits_{i=0}^{n-1} -\frac{h^3}{12} \cdot f''(\xi_i)$

Completam cu $\frac{1}{n}$: $E_T(f;h) = -\frac{h^3}{12} \cdot [\sum\limits_{i=0}^{n-1} f''(\xi_i)] \cdot \frac{1}{n} \cdot n$

Tinem cont ca f'' continua, deci exista o medie a valorilor: $E_T(f;h) = -\frac{h^3}{12} \cdot f''(\xi) \cdot n,\ \xi \in (a, b)$

Reamintim ca $n = \frac{b-a}{h}$: $E_T(f;h) = -\frac{b-a}{12} \cdot h^2 \cdot f''(\xi),\ \xi \in (a, b)$

$\xRightarrow{|\,|} \boxed{|E_T(f;h)| \le \frac{b-a}{12} \cdot h^2 \cdot \max\limits_{x \in (a,b)} |f''(x)|}$

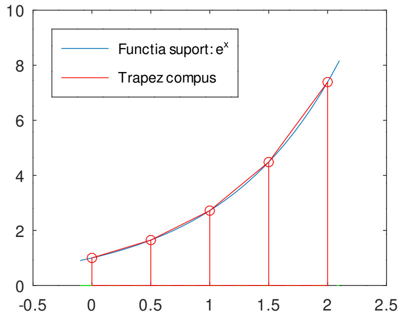
Astfel, am determinat limitele erorii totale pentru *Regula Compusa a Trapezelor*.

Exponentul lui h ne indica faptul ca formula este o metoda de **ordin 2**.

[11] https://stackoverflow.com/questions/1422149/what-is-vectorization

Exemplu numeric

Fie functia $f(x) = e^x$, $x \in [0, 2]$. Aproximati $\int_0^2 f(x)\,dx$ cu *Regula Compusa a Trapezelor*, folosind $n = 4$ subintervale.



Pur si simplu, aplicam formula (**), pentru $n = 4$:
$\int_0^2 f(x)\,dx \simeq h \cdot [\frac{f(0)+f(2)}{2} + f(0.5) + f(1) + f(1.5)]$, unde $h = \frac{b-a}{n} = \frac{2-0}{4} = 0.5$.

Deci, $\int_0^2 f(x)\,dx \simeq 0.5 \cdot [\frac{e^0+e^2}{2} + e^{0.5} + e^1 + e^{1.5}] \Longrightarrow \boxed{\int_0^2 f(x)\,dx \simeq 6.521}$

Observatie: Valoarea exacta a integralei este $\int_0^2 f(x)\,dx = 6.389$
Asadar, am obtinut o aproximatie mult mai buna fata de trapez simplu.

Pentru a intelege mai bine cum lucreaza *Regula Compusa a Trapezelor*, puteti urmari aceasta Animatie

Cerinta suplimentara: Care este numarul minim de subintervale prin care putem aproxima integrala $\int_0^2 e^x\,dx$, folosind *Regula Compusa a Trapezelor*, cu o eroare mai mica de $0.5 \cdot 10^{-4}$?

Ne folosim de inegalitatea: $|E_T(f;h)| \leq \frac{b-a}{12} \cdot h^2 \cdot \max_{x\in(a,b)} |f''(x)|$ si tinem cont ca $\begin{cases} f''(x) = e^x,\ f'' \text{ continua} \\ a = 0 \\ b = 2 \end{cases}$

Inlocuim $\max_{x\in(0,2)} |f''(x)| = e^2$ in inegalitatea de mai sus si obtinem:
$|E_T(f;h)| \leq \frac{2-0}{12} \cdot h^2 \cdot e^2 \leq 0.5 \cdot 10^{-4} \iff h^2 \leq 0.5 \cdot 10^{-4} \cdot 6 \cdot e^{-2} \simeq 4.06 \cdot 10^{-5}$
Dar, $h = \frac{2-0}{n}$. Atunci, inegalitatea devine: $\frac{2-0}{n} \leq \sqrt{4.06 \cdot 10^{-5}} \Rightarrow \boxed{n \geq 314}$

Asadar, avem nevoie de cel putin 314 subintervale (315 puncte) pentru a aproxima integrala cu o eroare mai mica de $0.5 \cdot 10^{-4}$. In concluzie, *foarte multe puncte*.

Ce putem imbunatati?
Putem creste gradul polinomului de interpolare Lagrange de la $n = 1$ la $n = 2$. Astfel, introducem in discutie *Regula Simpson*.

2.2.3 Regula simpla Simpson (n=2)

Modul de determinare
De data aceasta, suportul interpolarii este format din $n + 1 = 2 + 1 = 3$ puncte (x_0, x_1, x_2).
Asadar, mai avem un punct x_1 in mijlocul intervalului $[x_0, x_2] \Rightarrow$ Aproximatie mai buna a functiei, implicit a integralei.

Putem determina *Formula Simpla Simpson* in 2 moduri:

- Interpolam Lagrange (Analog trapezelor)
- Pornim de la dezvoltarea in serie Taylor a functiei f in punctul x_1, pastrand primii 3 termeni si folosind aproximatia derivatei a II-a, invatate anterior.

In cele ce urmeaza, vom exemplifica varianta cu interpolare Lagrange:

Impartim intervalul $[a, b]$ in $n = 2$ subintervale.
Atunci, abscisele vor fi: $\boxed{x_0 = a}$ si $\boxed{x_1 = \frac{a+b}{2}}$, $\boxed{x_2 = b}$, iar lungimea subintervalului: $\boxed{h = \frac{b-a}{2}}$

Polinomul de interpolare Lagrange de grad $n = 2$:
$P_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)$

Asadar, pornim de la aproximatia $f(x) \simeq P_2(x)$ si intergram in ambii membri:
$\int_a^b f(x)\,dx \simeq \int_{x_0}^{x_2} [\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)]\,dx$

$$\int_a^b f(x)\,dx \simeq \frac{f(x_0)}{2h^2}\cdot\int_{x_0}^{x_2}(x-x_1)(x-x_2)\,dx - \frac{f(x_1)}{h^2}\cdot\int_{x_0}^{x_2}(x-x_0)(x-x_2)\,dx + \frac{f(x_2)}{2h^2}\cdot\int_{x_0}^{x_2}(x-x_0)(x-x_1)\,dx$$

$$\Rightarrow \int_a^b f(x)\,dx \simeq \frac{f(x_0)}{2h^2}\cdot\frac{2h^3}{3} - \frac{f(x_1)}{h^2}\cdot\frac{-4h^3}{3} + \frac{f(x_2)}{2h^2}\cdot\frac{2h^3}{3} \iff \int_a^b f(x)\,dx \simeq f(x_0)\cdot\frac{h}{3} + f(x_1)\cdot\frac{4h}{3} + f(x_2)\cdot\frac{h}{3}$$

Astfel, am determinat formula pentru *Regula Simpla Simpson*:

Regula Simpla Simpson

$$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{3}\cdot[f(a)+4f(\tfrac{a+b}{2})+f(b)]},\ h=\frac{b-a}{2}\quad (***)$$

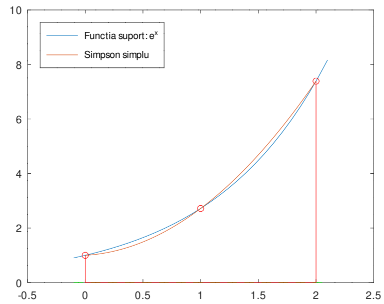
Exemplu numeric

Fie functia $f(x)=e^x$, $x\in[0,2]$. Aproximati $\int_0^2 f(x)\,dx$, folosind *Regula Simpla Simpson*.

Pur si simplu, aplicam formula (***):
$\int_0^2 f(x)\,dx \simeq \frac{h}{3}\cdot[f(0)+4f(\frac{0+2}{2})+f(2)]$, unde $h=\frac{b-a}{2}=\frac{2-0}{2}=1$.

Deci, $\int_0^2 f(x)\,dx \simeq \frac{1}{3}\cdot[e^0+4\cdot e^1+e^2] \Longrightarrow \boxed{\int_0^2 f(x)\,dx \simeq 6.420}$

Observatie: Valoarea exacta a integralei este $\int_0^2 f(x)\,dx = 6.389$
Asadar, metodele incep sa se apropie de valoarea exacta.

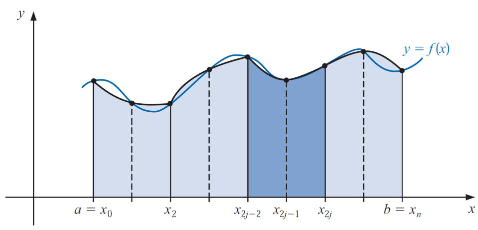


Ce putem imbunatati?
La fel ca la trapeze, in loc sa integram pe tot intervalul, impartim problema in probleme mai mici, aplicand regula simpla Simpson pe subintervale.

2.2.4 Regula compusa Simpson

Modul de determinare

Regula simpla Simpson discutata anterior, este utila atunci cand intergram o singura data, pe tot intervalul $[a,b]$.
Pentru a ne apropia de o aproximatie si mai buna a integralei $\int_a^b f(x)\,dx$, trebuie sa divizam problema in sub-probleme:
Impartim intervalul $[a,b]$ in $(2n)$ subintervale de *lungimi egale*: $x_i = a + i\cdot h$, $i = 0:2n$ si $h=\frac{b-a}{2n}$



Apoi, aplicam regula simpla Simpson pe fiecare subinterval.
Astfel, integrala se poate aproxima: $\int_a^b f(x)\,dx \simeq \sum_{i=1}^{n}\frac{h}{3}\cdot[f(x_{2i-2})+4f(x_{2i-1})+f(x_{2i})]$

$$\Rightarrow \int_a^b f(x)\,dx \simeq \frac{h}{3}[f(x_0)+2\cdot\sum_{i=1}^{n-1}f(x_{2i})+4\cdot\sum_{i=1}^{n}f(x_{2i-1})+f(x_{2n})]$$

Astfel, am determinat formula pentru *Regula Compusa Simpson*:

Regula Compusa Simpson

$$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{3}\cdot[f(x_0)+f(x_{2n})+4\cdot\sum_{i=1}^{n}f(x_{2i-1})+2\cdot\sum_{i=1}^{n-1}f(x_{2i})] \overset{\text{not}}{=} S(f;h)},\ h=\frac{b-a}{2n}\quad (****)$$

La fel ca la trapeze, *avantajul* gruparii termenilor in aceasta forma este ca se poate **vectoriza**.

Analiza erorii

Facem urmatoarele notatii: $\begin{cases} S(f;h) \equiv \text{aproximatia integralei folosind regula compusa Simpson} \\ I(f) \equiv \text{valoarea exacta a integralei} \\ E_S(f;h) \equiv \text{eroarea Simpson pentru functia } f \text{ cu pas } h \\ E_{S,i}(f;h) \equiv \text{eroarea Simpson pe subintervalul } i \end{cases}$

Se procedeaza in mod analog *erorii trapezelor* si se gaseste ca eroarea pe subintervalul i este:
$E_{S,i}(f;h) = -\frac{h^5}{90} \cdot f^{(4)}(\xi_i),\ \xi_i \in (x_{2i}, x_{2i+2})$

Insumam erorile *locale* si obtinem eroarea *totala*: $E_S(f;h) = I(f) - S(f;h)$

$\Rightarrow E_S(f;h) = -\frac{1}{90} \cdot h^5 \cdot [\sum_{i=0}^{n-1} f^{(4)}(\xi_i)] \cdot \frac{1}{n} \cdot n$ (am completat cu n)

Tinem cont ca $f^{(4)}$ continua, deci exista o medie a valorilor:

$\Rightarrow E_S(f;h) = -\frac{h^5}{90} \cdot f^{(4)}(\xi) \cdot \frac{b-a}{2h},\ \xi \in (a,b)$ (reamintim ca $n = \frac{b-a}{2h}$)

$\overset{|\,|}{\Longrightarrow} \boxed{|E_S(f;h)| \le \frac{b-a}{180} \cdot h^4 \cdot \max_{x\in(a,b)} |f^{(4)}(x)|}$

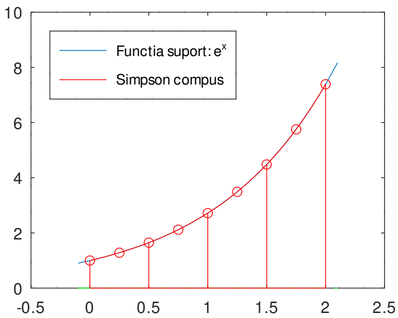
Astfel, am determinat limitele erorii totale pentru *Regula Compusa Simpson*.
Exponentul lui h ne indica faptul ca formula este o metoda de **ordin 4**.

Exemplu numeric

Fie functia $f(x) = e^x$, $x \in [0,2]$. Aproximati $\int_0^2 f(x)\,dx$ cu *Regula Compusa Simpson*, folosind $n = 4$ subintervale.

Pur si simplu, aplicam formula (****), pentru $n = 4$:
$\int_0^2 f(x)\,dx \simeq \frac{h}{3} \cdot \{f(x_0) + f(x_8) + 4 \cdot [f(x_1) + f(x_3) + f(x_5) + f(x_7)] + 2 \cdot [f(x_2) + f(x_4) + f(x_6)]\}$, unde $h = \frac{b-a}{2n} = \frac{2-0}{8} = 0.25$.

Deci, $\int_0^2 f(x)\,dx \simeq \frac{0.25}{3} \cdot [e^0 + e^2 + 4 \cdot (e^{0.25} + e^{0.75} + e^{1.25} + e^{1.75}) + 2 \cdot (e^{0.5} + e^1 + e^{1.5})] \Longrightarrow \boxed{\int_0^2 f(x)\,dx \simeq 6.38919}$



Observatie: Valoarea exacta a integralei este $\int_0^2 f(x)\,dx = 6.38906$
Asadar, am obtinut o aproximatie foarte buna, considerand doar $n = 4$.

Pentru a intelege mai bine cum lucreaza *Regula Compusa Simpson*, puteti urmari aceasta Animatie

Cerinta suplimentara: Care este numarul minim de subintervale prin care putem aproxima integrala $\int_0^2 e^x\,dx$, folosind *Regula Compusa Simpson*, cu o eroare mai mica de $0.5 \cdot 10^{-4}$?

Ne folosim de inegalitatea: $|E_S(f;h)| \le \frac{b-a}{180} \cdot h^4 \cdot \max_{x\in(a,b)} |f^{(4)}(x)|$ si tinem cont ca $\begin{cases} f^{(4)}(x) = e^x,\ f^{(4)} \text{ continua} \\ a = 0 \\ b = 2 \end{cases}$

Inlocuim $\max_{x\in(0,2)} |f^{(4)}(x)| = e^2$ in inegalitatea de mai sus si obtinem:

$|E_S(f;h)| \le \frac{2-0}{180} \cdot h^4 \cdot e^2 \le 0.5 \cdot 10^{-4} \iff h^4 \le 0.5 \cdot 10^{-4} \cdot \frac{180}{2e^2} \simeq 6.09 \cdot 10^{-4}$

Dar, $h = \frac{2-0}{2n}$. Atunci, inegalitatea devine: $\frac{2-0}{2n} \le \sqrt[4]{6.09 \cdot 10^{-4}} \Rightarrow \boxed{n \ge 7}$

Asadar, avem nevoie de cel putin 7 subintervale ($2 \cdot 7 + 1 = 15$ puncte) pentru a aproxima integrala cu o eroare mai mica de $0.5 \cdot 10^{-4}$.

Daca va aduceti aminte, la *Trapez Compus* (ordin 2) era nevoie de 315 puncte, pe cand la *Simpson Compus* (ordin 4) sunt necesare doar 15 puncte, pentru aceeasi aproximatie (si aceeasi complexitate).

Ce putem imbunatati?

Pana acum, am lucrat cu metode care implica divizarea intervalului $[a,b]$ in subintervale de *lungimi egale*.

In continuare, ne vom ocupa de o tehnica de aproximare a integralelor care nu necesita puncte echidistante.

2.3 Cuadraturi adaptive

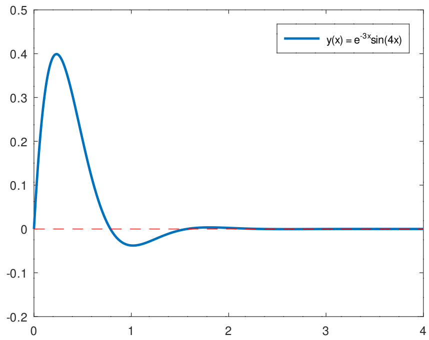
2.3.1 Notiuni generale

Eroarea locala pentru aproximarea unei integrale este mare acolo unde functia se schimba mult (derivatele $f^{(k)}$ sunt mari).

Folosind subintervale egale (ca pana acum), irosim efort in zonele in care functia nu se schimba foarte mult. Regulile compuse discutate anterior sunt foarte utile, insa au de suferit pentru ca necesita puncte echidistante.

2.3.2 Motivatie

Sa consideram urmatoarea ecuatie diferentiala: $y'' + 6y' + 25 = 0$, avand urmatoarele conditii initiale $\begin{cases} y(0) = 0 \\ y'(0) = 4 \end{cases}$



Rezolvand analitic ecuatia diferentiala, obtinem solutia unica $y(x) = e^{-3x} \cdot sin(4x)$

Functii de acest tip sunt foarte intalnite in *ingineria mecanica* (descriu anumite caracteristici ale resorturilor, amortizoarelor) si in *ingineria electrica* (sunt solutii pentru circuite electrice).

Graficul functiei $y(x) = e^{-3x} \cdot sin(4x)$ este ilustrat in figura alaturata.

Sa presupunem ca vrem sa calculam integrala $\int_0^4 y(x)\,dx$. Graficul indica faptul ca pe intervalul $[3, 4]$, integrala (aria) este foarte aproape de 0, iar pe $[2, 3]$ nu ne asteptam sa fie o valoare foarte mare. Insa, pe intervalul $[0, 2]$ are loc o variatie semnificativa a functiei si nu mai este clar ce se intampla cu integrala.

Aceasta este o situatie in care *Regulile Compuse* nu sunt foarte folositoare, deoarece folosesc acelasi efort atat pe portiunile in care f variaza, cat si in regiunile in care f nu variaza atat de mult.

Astfel, apare intrebarea: *Ce tehnica trebuie folosita pe fiecare interval de intergrare si ce acuratete trebuie sa aiba?*

2.3.3 Idee

Introducem conceptul de **cuadratura adaptiva**.[12]
Ne concentram atentia pe zonele in care functia $f(x)$ variaza mai mult.
Altfel spus, micsoram (*injumatatim*) pasul h in regiunile in care functia variaza mai mult.

2.3.4 Modul de determinare

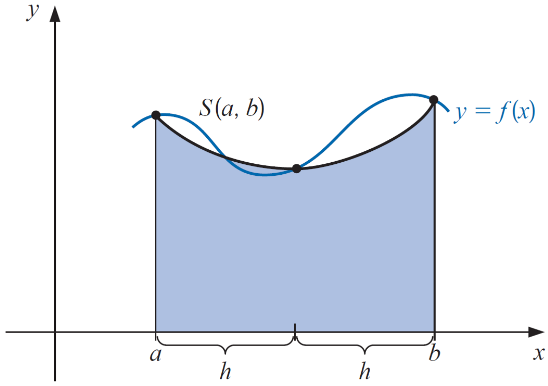
Notiunea de cuadratura adaptiva poate fi implementata cu orice metoda (trapeze, Simpson).
Vom exemplifica folosind Simpson:

Sa presupunem ca vrem sa aproximam $\int_a^b f(x)\,dx$ cu o anumita toleranta $\epsilon > 0$.

Primul pas este sa aplicam *Simpson* cu pasul $h = \frac{b-a}{2}$

Conform Simpson: $S_1[a, b] = \frac{b-a}{6} \cdot [f(a) + 4f(\frac{a+b}{2}) + f(b)]$
Eroarea Simpson: $E_1[a, b] = -\frac{h^5}{90} \cdot f^{(4)}(\xi),\ \xi \in (a, b)$

Atunci, $I(f)[a, b] = S_1[a, b] + E_1[a, b]$



[12] https://en.wikipedia.org/wiki/Adaptive_quadrature

Al doilea pas este sa *injumatatim* intervalul $[a, b]$

Astfel, obtinem punctul din mijloc $\boxed{c = \frac{a+b}{2}}$

Atunci, $I(f)[a,b] = I(f)[a,c] + I(f)[c,b]$

Dar, $\begin{cases} I(f)[a,c] = S_1[a,c] + E_1[a,c] \\ I(f)[c,b] \ = S_1[c,b] + E_1[c,b] \end{cases}$

Notam $\begin{cases} S_2[a,b] \overset{\text{not}}{=} S_1[a,c] + S_1[c,b] \\ E_2[a,b] \overset{\text{not}}{=} E_1[a,c] + E_1[c,b] \end{cases}$

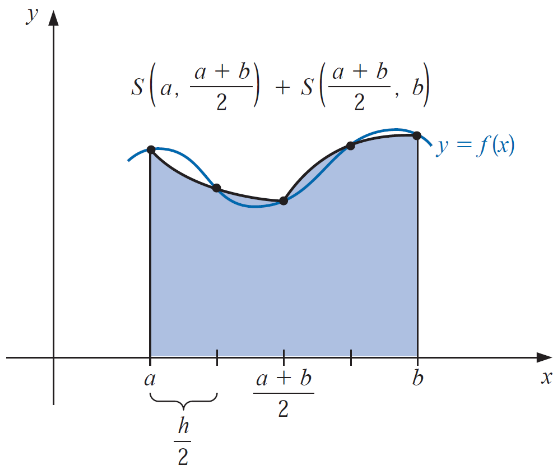
Atunci, $I(f)[a,b] = S_2[a,b] + E_2[a,b]$

Dar, $E_2[a,b] = -\frac{1}{90} \cdot (\frac{h}{2})^5 \cdot [f^{(4)}(\xi_1) + f^{(4)}(\xi_2)]$, $\begin{cases} \xi_1 \in (a,c) \\ \xi_2 \in (c,b) \end{cases}$

Daca $f^{(4)}$ nu se schimba mult, $E_1[a,c] \simeq E_1[c,b]$

$\Rightarrow E_2[a,b] \simeq 2 \cdot \frac{1}{2^5} \cdot -\frac{h^5}{90} \cdot f^{(4)}(\xi) \Rightarrow \boxed{E_2[a,b] \simeq \frac{1}{2^4} \cdot E_1[a,b]}$

Reamintim faptul ca $\begin{cases} E_2[a,b] \equiv \text{eroarea cu pas } \frac{h}{2} \\ E_1[a,b] \equiv \text{eroarea cu pas } h \end{cases}$



Concluzionand, daca integram pe tot intrevalul $[a, b]$, obtinem: $I(f) = S_1 + E_1$, iar daca integram cu punctul c la mijloc, obtinem: $I(f) = S_2 + E_2$, unde $E_2 = \frac{1}{2^4} \cdot E_1$

Egaland cele 2 ecuatii, obtinem: $S_2 - S_1 = 15E_2$. Asadar, putem aproxima $\boxed{E_2 = \frac{1}{15} \cdot (S_2 - S_1)}$

Asta inseamna ca aproximatia cu pas $\frac{h}{2}$ este de 15 ori mai buna decat aproximatia cu pas h.

Reamintim faptul ca $\begin{cases} S_2 \equiv \text{aproximatia Simpson cu pas } \frac{h}{2} \\ S_1 \equiv \text{aproximatia Simpson cu pas } h \end{cases}$

Deci, cand $f^{(4)}$ NU variaza mult, putem aproxima $\boxed{I = S_2 + \frac{S_2 - S_1}{15}}$

Practic, aproximam si eroarea E_2, astfel ne apropiem mult mai repede de rezultatul dorit.

2.3.5 Pseudocod

$function\ [S] = adaptive\_simpson(f, st, dr, tol)$
  $mij = st + \frac{dr-st}{2}$ (echivalent cu: $\frac{st+dr}{2}$) [13]
  Compute S1 = Simpson pe $[st, dr]$
  Compute S2 = Simpson pe $[st, mij]$ + Simpson pe $[mij, dr]$

  Daca $|S_2 - S_1| < 15 \cdot tol$ (functia $f^{(4)}$ nu se schimba mult)
    $S = S_2 + \frac{S_2 - S_1}{15}$
  Altfel (functia $f^{(4)}$ variaza mult)
    $mij = st + \frac{dr-st}{2}$
    $left = adaptive\_simpson(f, st, mij, \frac{tol}{2})$
    $right = adaptive\_simpson(f, mij, dr, \frac{tol}{2})$
    $S = left + right$
$endfunction$

Observatie: Se apeleaza recursiv *left* si *right* cu tolerantele $\frac{tol}{2}$ pentru ca suma lor va avea cu siguranta o toleranta mai mica decat *tol*.

2.3.6 Animatie

Foarte utila in intelegerea modului de lucru al algoritmului, este urmatoarea Animatie

[13] https://cs.stackexchange.com/questions/80415/why-is-binary-search-using-this-weird-thing-to-calculate-middle

2.4 Rezumat formule integrare

Regula Simpla a Trapezelor (*)

$$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{2}[f(a)+f(b)]}\ (h=b-a)$$

Regula Compusa a Trapezelor (**)

$$\boxed{\int_a^b f(x)\,dx \simeq h\cdot[\frac{f(x_0)+f(x_n)}{2}+\sum_{i=1}^{n-1} f(x_i)] \overset{\text{not}}{=} T(f;h)},\ h=\frac{b-a}{n}$$

Regula Simpla Simpson (***)

$$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{3}\cdot[f(a)+4f(\frac{a+b}{2})+f(b)]},\ h=\frac{b-a}{2}$$

Regula Compusa Simpson (****)

$$\boxed{\int_a^b f(x)\,dx \simeq \frac{h}{3}\cdot[f(x_0)+f(x_{2n})+4\cdot\sum_{i=1}^{n} f(x_{2i-1})+2\cdot\sum_{i=1}^{n-1} f(x_{2i})] \overset{\text{not}}{=} S(f;h)},\ h=\frac{b-a}{2n}$$

3 Probleme Propuse

3.1 Problema 1 - Derivata I

Completati urmatoarele programe Octave pentru a calcula **derivatele I** ale functiilor in punctele de interes.

Descarca scriptul:

```
function [df pf p2points] = TwoPoints(f, x0, h)
  % Input:
  %       Functia f
  %       Punctul (x0) in care trebuie sa aproximez prima derivata a lui f
  %       Pasul h
  % Output:
  %        Valoarea aproximata (y0) a derivatei lui f in x0
  %        2 ploturi pentru legenda


  % TODO: df = Formula TwoPoints: df = ...

```

Descarca scriptul:

```
function [df p3pointsMid] = ThreePoints_Mid(f, x0, h)
  % Input:
  %       Functia f
  %       Punctul (x0) in care trebuie sa aproximez prima derivata a lui f
  %       Pasul h
  % Output:
  %        Valoarea aproximata (y0) a derivatei lui f in x0
  %        1 plot pentru legenda


  % TODO: df = Formula ThreePointsMid: df = ...

```

Descarca scriptul:

```
function [df p3pointsEnd] = ThreePoints_End(f, x0, h)
  % Input:
  %       Functia f
  %       Punctul (x0) in care trebuie sa aproximez prima derivata a lui f
  %       Pasul h
  % Output:
  %        Valoarea aproximata (y0) a derivatei lui f in x0
  %        1 plot pentru legenda


  % TODO: df = Formula ThreePointsEnd: df = ...
```

Pentru testarea programelor, puteti folosi urmatoarea functie:
Descarca scriptul:

```
function [] = testare_derivate()

  clf;

  a = 2;
  b = 3;
  x0 = [a : 0.5 : b]; % Abscisele unde se vor aproxima derivatele

  h = 0.01; % Pasul

  f = @(x) exp(2.*x); % Functie de derivat
  df = @(x) 2.*exp(2.*x); % Derivata functiei

```

3.2 Problema 2 - Derivata II

Completati urmatoarea functie Octave pentru a calcula **derivatele II** ale functiilor in punctele de interes.
Descarca scriptul:

```
function d2f = ThreePoints(f, x0, h)
  % Input:
  %       Functia f
  %       Punctul (x0) in care trebuie sa aproximez a doua derivata a lui f
  % Output:
  %        Valoarea aproximata (y0) a derivatei de ordin 2 a lui f in x0

  % TODO: d2f = Second Derivative Midpoint Formula: d2f = ...


```

Pentru testarea programului, puteti folosi urmatoarea functie:
Descarca scriptul:

```
function [] = testare_derivate()

  a = 2;
  b = 3;
  x0 = [a : 0.5 : b]; % Abscisele unde se vor aproxima derivatele II

  h = 0.1; % Pasul

  f = @(x) x .* exp(x); % Functie de derivat
  d2f = @(x) (x+2) .* exp(x); % Derivata de ordin 2 a functiei

```

3.3 Problema 3 - Integrare

Completati urmatoarele functii Octave pentru a calcula **integralele definite** ale functiilor pe intervalele de interes.
Descarca scriptul:

```
function aprox = trapez_simplu(f, a, b)
  % Input:
  %       Functia f
  %       Intervalul de integrare [a, b]
  % Output:
  %       Valoarea aproximata a integralei

  % Trapez simplu: Interpolare liniara intre f(a) si f(b)
  % Deci, se foloseste polinomul de interpolare Lagrange de grad 1
  % Apoi se aproximeaza integrala cu aria trapezului

  % TODO: Pasul h - Trapez Simplu: h = ...


  % TODO: Aproximatie - Aplicare Formula: aprox = ...

```

Descarca scriptul:

```
function aprox = trapez_compus(f, a, b, n)
  % Input:
  %       Functia f
  %       Intervalul de integrare [a, b]
  %       Numarul de puncte echidistante (n)
  % Output:
  %       Valoarea aproximata a integralei


  % TODO: Pasul h - Trapez Compus: h = ...


  % TODO: n puncte echidistante: x = ...


  % TODO: Aproximatie - Aplicare Formula: aprox = ...

```

Descarca scriptul:

```
function aprox = Simpson_simplu(f, a, b)
  % Input:
  %       Functia f
  %       Intervalul de integrare [a, b]
  % Output:
  %       Valoarea aproximata a integralei

  % Simpson simplu: Interpolare patratica intre f(a) si f(b)
  % Partea de sus a trapezului ('capacul') este inlocuita cu o parabola
  % Deci, se foloseste polinomul de interpolare Lagrange de grad 2

  % TODO: Pasul h - Simpson Simplu: h = ...


  % TODO: Aproximatie - Aplicare Formula: aprox = ...

```

Descarca scriptul:

```
function aprox = Simpson_compus(f, a, b, n)
  % Input:
  %       Functia f
  %       Intervalul de integrare [a, b]
  %       Numarul de puncte echidistante (2n)
  % Output:
  %       Valoarea aproximata a integralei


  % TODO: Pasul h - Simpson Compus: h = ...


  % TODO: 2n puncte echidistante: x = ...


  % TODO: Aproximatie - Aplicare Formula: aprox = ...

```

Pentru testarea programului, puteti folosi urmatoarea functie:
Descarca scriptul:

```
function [] = testare_Newton_Cotes()

  format long;

  % Functia de integrat
  f = @(x) exp(x);

  % Intervalul de integrat
  a = 1;
  b = 3;

  % Numarul de subintervale
  n = 4;
```

Pentru testarea programului, veti avea nevoie si de functia care aplica interpolarea Lagrange pe un set de puncte (vezi *Laborator Interpolare*).

Descarca scriptul:

```
function y0 = lagrange(x, y, x0)
  % Input:
  %        Coordonatele punctelor din suportul de interpolare (x, y)
  %        Punctul (x0) in care dorim sa aflam valoarea dupa interpolare
  % Output:
  %        Valoarea (y0) in punctul dorit (x0)

```

3.4 Extra

In incheiere, puteti lectura un material referitor la utilizarea integrarii numerice intr-o aplicatie din viata reala (**Reverse Osmosis Model**):