



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Radu Fechete

Dumitrița C. Moldovan

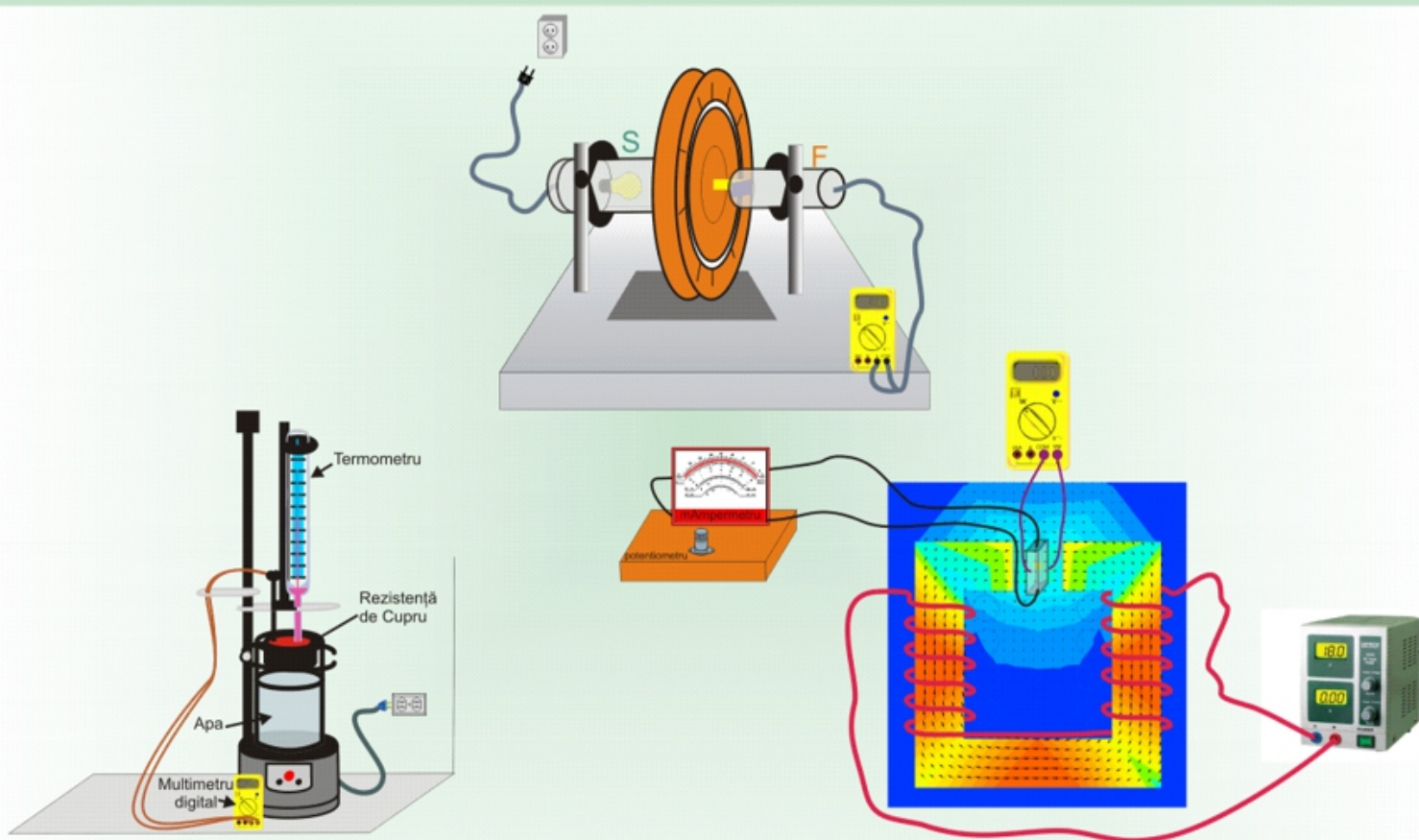
Ramona I. Chelcea

Lidia Pop

Maria Boșca

FIZICĂ

Îndrumător de lucrări virtuale de laborator



UTPRESS
Cluj-Napoca, 2021
ISBN 978-606-737-519-0



Radu Fechete, Dumitrița C. Moldovan, Ramona I. Chelcea, Lidia Pop, Maria Boșca

Fizică

Îndrumător de lucrări virtuale de laborator



UTPRESS

Cluj - Napoca, 2021

ISBN 978-606-737-519-0



Editura U.T.PRESS
Str. Observatorului nr. 34
400775 Cluj-Napoca
Tel.:0264-401.999
e-mail: utpress@biblio.utcluj.ro
<http://biblioteca.utcluj.ro/editura>

Director: ing. Călin Câmpean

Recenzia: Prof. dr. Eugen Culea
 Şef I. dr. Marius Nicolae Bîrlea

Coperta: Dumitrița C. Moldovan

Copyright © 2021 Editura U.T.PRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii U.T.PRESS.

ISBN 978-606-737-519-0

Bun de tipar: 17.06.2021

Cuprins

I. Introducere	5
II Mecanica.....	6
II. 1 Determinarea modulului longitudinal de elasticitate.....	6
II.2 Determinarea constantei elastice a unui resort	9
II.3 Determinarea accelerației gravitaționale cu ajutorul unui pendul fizic	13
II.4 Determinarea timpului de ciocnire a două bile.....	15
II.5 Studiul undelor staționare transversale în corzi vibrante	18
III. Termodinamica și mecanica fluidelor	21
III.1 Determinarea raportului căldurilor molare ale gazelor.....	21
III.2 Studiul unei pompe de căldură	24
III.3 Determinarea coeficientului de vâscozitate al lichidelor	27
III.4 Determinarea coeficientului de vâscozitate dinamică al aerului.....	30
IV Electricitate și Magnetism	33
IV.1 Studiul conductibilității electrice a metalelor	33
IV.2 Studiul efectului Hall.....	36
V Optică	39
V.1 Etalonarea unui spectroscop. Analiză spectrală calitativă	39
V.2 Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină.....	42
V.3 Studiul efectului fotoelectric și determinarea constantei lui Planck.....	46
V.4 Studiul polarizării luminii	49
VI Fizica solidului.....	51
VI.1 Studiul efectului termoelectric	51
VI.2 Determinarea energiei de activare a unui semiconductor.....	53
VI.3 Determinarea conductibilității termice printr-o metodă staționară	56
VI.4 Verificarea experimentală a legii lui Ștefan-Boltzmann	59
VI.5 Determinarea temperaturii Curie a materialelor feromagnetice.....	62
VI.6 Studiul comportării materialelor feromagnetice	65
VII. Codul sursa Java Script.....	68
1. Determinarea modulului longitudinal de elasticitate.....	68
2. Determinarea constantei elastice a unui resort	73
3. Determinarea accelerației gravitaționale cu ajutorul pendulului fizic	81

4.	Determinarea timpului de ciocnire a două bile.....	89
5.	Studiul undelor staționare in corzi vibrante.....	98
6.	Determinarea raportului căldurilor molare ale gazelor.....	103
7.	Studiul unei pompe de căldură.....	115
8.	Determinarea coeficientului de vâscozitate al lichidelor.....	121
9.	Determinarea coeficientului de vâscozitate dinamică a aerului.....	127
10.	Studiul conductibilității electrice a metalelor.....	136
11.	Studiul efectului Hall.....	143
12.	Etalonarea unui spectroscop. Analiză spectrală calitativă.....	150
13.	Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină.....	162
14.	Studiul efectului fotoelectric și determinarea constantei lui Planck.....	173
15.	Studiul polarizării luminii.....	183
16.	Studiul efectului termoelectric.....	196
17.	Determinarea energiei de activare a unui semiconductor.....	202
18.	Determinarea conductibilității termice printr-o metodă staționară.....	210
19.	Verificarea Experimentală a legii lui Stefan-Boltzmann.....	217
20.	Determinarea temperaturii Curie a materialelor feromagnetice.....	223
21.	Studiul comportării materialelor feromagnetice.....	232
VIII.	Bibliografie.....	238

I. Introducere

Experiență practică! Un lucru extrem de important pentru formarea unui inginer, a unui om de știință, sau a oricărui om din secolul XXI. Experiența practică este ceea ce ne-a lipsit cel mai mult în aceste vremuri de izolare fizică, atunci când cursurile, seminariile dar și laboratoarele sunt efectuate online. O experiență cât mai apropiată de experiența practică este ceea ce vă propune acest îndrumător de laborator virtual. Cu atât mai mult dacă este vorba despre Fizică, știința fundamentală care studiază fenomenele naturii în direct contact cu aceasta. Iar dacă experiența directă nu mai este posibilă, iar realitatea virtuală este încă departe de a fi accesibilă tuturor, un colectiv de patru cadre didactice, Șef de Lucrări la Universitatea Tehnică din Cluj-Napoca sub coordonarea unui Profesor Universitar au depus eforturi considerabile pentru a crea simulări virtuale ale unui număr de 21 de Lucrări de Laborator de Fizică importante, care se găsesc on site în Laboratoarele de Fizică ale UTCN.

Lucrările de laborator virtuale, pot să fie accesate online de către studenții, și nu numai, din orice colț al lumii. Pentru aceasta s-a folosit limbajul de programare JavaScript, ocazie cu care autorii doresc să îi acorde mulțumirile lor lui *Daniel Shiffman* și a lui serie extraordinară de lecții de programare *The Coding Train*. Acest îndrumător de lucrări de laborator virtuale apare imediat după o experiență de un an de lucru cu studenții UTCN de la aproape toate secțiile care au în programa analitică Laboratorul de Fizică. Autorii pot să spună cu mulțumire că efortul lor a fost apreciat de către studenți. Astfel, aceștia au putut individual să efectueze lucrările de laborator și să prelucreze datele experimentale, iar experiența a fost cât se poate de apropiată de realitate. Simulările interactive ale lucrărilor de laborator permit nu numai observarea în detaliu a fenomenului fizic, dar câteodată o astfel de simulare numerică (de exemplu la Studiul Efectului Fotoelectric) este mai ofertantă chiar decât experiența practică. Astfel studenții au ocazia, scriind direct valori numerice sau prin modificarea poziției unor cursoare sau apăsând pe butoane virtuale, să pornească încălzirea unui vas cu apă sau a unui conductor, să modifice intensitatea curentului electric, să modifice frecvența curentului prin conductor, să închidă și să deschidă robinete, să rotească polarizori, sau să măsoare spectrul atomic de emisie, să încarce sau să descarce condensatori, să cionească două bile sau să pună în oscilație un pendul fizic. Lucrările de laborator virtuale au fost folosite cu succes pentru orele de laborator dar și pentru orele de curs. În speranța că prin aceasta Fizica a devenit mai accesibilă și mai frumoasă pentru toată lumea, vă dorim Succes!

Cluj-Napoca, Iunie 2021
Autorii

II Mecanica

II. 1 Determinarea modului longitudinal de elasticitate

Introducere

Un corp solid supus unei solicitări la întindere se deformează. Deformarea poate fi:

- elastică, dacă în urma încetării acțiunii forței exterioare, corpul revine la starea inițială.
- plastică, dacă în urma încetării acțiunii forței exterioare, corpul nu revine la forma inițială.

În cazul unei *deformări elastice*, în interiorul corpului deformat ia naștere o forță elastică care este direct proporțională cu constanta elastică, k , și alungirea corpului, Δl , și se opune deformării. În cazul particular al unui fir, care are secțiunea normală S și lungimea inițială l_0 această forță se opune solicitării exterioare reprezentate de o forță, F , care acționează la capătul acestuia. Astfel, din legea lui Hooke:

$$\frac{F}{S} = E \frac{\Delta l}{l_0}$$

se exprimă modulul longitudinal de elasticitate E sau modulul lui Young. Acesta este o constantă de material. Constanta elastică depinde de natura materialului prin modulul de elasticitate, precum și de geometria firului (lungimea și aria secțiunii acestuia).

Simularea numerică a lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Modulul%20longitudinal/>

Prin accesarea link-ului se deschide pagina:

Laboratorul de Fizica: Determinarea modului longitudinal de elasticitate

Documentație laborator Student: _____ Grupa: _____

Experiment

Material	l [m]	d [mm]	S [10 ⁻⁶ m ²]	m [kg]	Δl [10 ⁻⁵ m]	E [10 ⁸ N/m ²]	E [10 ⁸ N/m ²]	ΔE/E [%]	ΔE [10 ⁸ N/m ²]
Oțel	1.03	1.0		1					
				3					
				5					
				7					
				9					
				11					
				13					

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

Dispozitivul experimental

Pentru determinarea modului longitudinal de elasticitate, în laborator, este folosit dispozitivul experimental prezentat în fotografia din stânga. Simularea fidelă a acestuia este prezentată în dreapta. Firul al cărui modul de elasticitate urmează a fi măsurat este fixat în partea sa superioară. În partea inferioară a firului este fixată o placă metalică pe care acționează un ceas comparator care poate măsura alungirea, Δl , cu o precizie foarte mare. Lungimea inițială a firului se măsoară cu un metru metalic care are precizia de 1 mm. Diametrul firului s-a măsurat cu un micrometru care are precizia de 0.01 mm. Forța exterioară depinde de valoarea greutăților puse pe placa metalică (taler).

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

Laboratorul de Fizica: Determinarea modului longitudinal de elasticitate

Student: _____ Grupa: _____

Material	l [m]	d [mm]	S [10 ⁻⁶ m ²]	m [kg]	Δl [10 ⁻⁵ m]	E [10 ⁹ N/m ²]	E [10 ⁹ N/m ²]	$\Delta E/E$ [%]	ΔE [10 ⁹ N/m ²]
Oțel	1.03	1.0		1					
				3					
				5					
				7					
				9					
				11					
				13					

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pașii pentru înregistrarea datelor constau în:

- i. Se încarcă masa, indicată în tabel, pe taler cu ajutorul cursorului: .
- ii. Se citește alungirea, în *mm*, pe ceasul comparator . Valoarea citită se trece în tabel, cu două zecimale, ținând cont de ordinul de mărime a lui Δl : $10^{-5}m$.
- iii. Pentru a determina modulul longitudinal de elasticitate (Young), se folosește formula:

$$E = \frac{mg l_0}{S \Delta l}$$

Se dau:

- Accelerația gravitațională: $g = 9.807 \text{ m/s}^2$;
- Lungimea firului: $l_0 = 1.03 \text{ m}$;

- Diametrul firului: $d = 1 \text{ mm}$;
 - Suprafața firului: $S = \frac{\pi d^2}{4}$.
- iv. După determinarea valorilor modulului longitudinal de elasticitate în fiecare caz, se va calcula media aritmetică a valorilor obținute. Aceasta se trece în tabel.
- v. Eroarea relativă de calcul a modulului longitudinal de elasticitate se calculează cu formula:

$$\frac{\Delta E}{E} = \frac{\Delta m}{m} + \frac{\Delta l_0}{l_0} + \frac{\Delta \pi}{\pi} + 2 \frac{\Delta d}{d} + \frac{\Delta(\Delta l)}{\Delta l}$$

- unde Δm , Δl_0 , $\Delta \pi$, Δd și $\Delta(\Delta l)$ sunt erorile absolute ale mărimilor fizice care intră în formula de calcul a modulului longitudinal de elasticitate. Valoarea obținută prin calcul se înmulțește cu 100 pentru ca eroarea relativă să aibă unitatea de măsură în procente așa cum se cere în tabel.
- vi. Eroarea absolută se determină din:

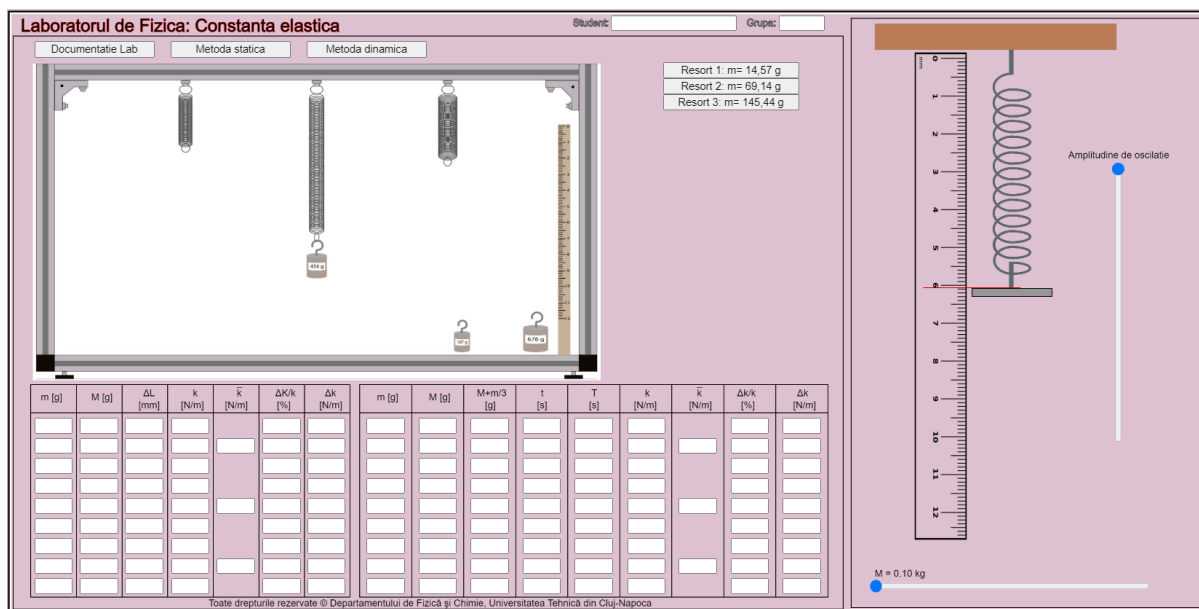
$$\Delta E = E \cdot \frac{\Delta E}{E}$$

unde $\frac{\Delta E}{E}$ este valoarea adimensională a erorii relative.

Vă urăm succes!

- ix. Valoarea medie a constantei elastice, \bar{k} se calculează făcând media aritmetică a celor trei valori ale constantei elastice pentru fiecare resort: $\bar{k} = \frac{k_1+k_2+k_3}{3}$.
 - x. Eroarea relativă se calculează cu relația: $\frac{\Delta k}{k} = \frac{\Delta M}{M} + \frac{\Delta g}{g} + \frac{\Delta(\Delta L)}{\Delta L}$;
 - xi. Eroarea absolută se calculează cu relația: $\Delta k = ak$, deoarece $\frac{\Delta k}{k} = a$.
 - xii. Toate rezultatele obținute se trec în tabel.
4. Pentru a efectua experimentul virtual prin metoda dinamică se apasă butonul

Metoda dinamica



5. Pașii pentru înregistrarea datelor constau în:
- i. Se notează în cel de-al doilea tabel în prima coloană cele trei mase diferite pentru cele trei resorturi, notate cu

Resort 1: m= 14,57 g
Resort 2: m= 69,14 g
Resort 3: m= 145,44 g

.
 - ii. Se apasă butonul

Resort 1: m= 14,57 g

 și mutând cursorul

M = 0.00 kg

 spre dreapta se alege trei mase diferite care se trec în tabel în cea de-a doua coloană.
 - iii. Cu ajutorul relației $M + \frac{m}{3}$ se calculează masa redusă, iar valorile obținute se trec în tabel.
 - iv. Se mută cursorul

M = 0.00 kg

 spre dreapta pentru prima masă aleasă și se alege amplitudinea dorită cu ajutorul cursorului

Amplitudine de oscilatie

, iar apoi se cronometrează durata t în care sistemul efectuează un număr $n = 20$ de oscilații complete.
 - v. Se fac câte trei măsurători pentru fiecare resort.

- vi. Se repetă măsurătorile pentru celelalte două resorturi.
- vii. Cu ajutorul relației $T = \frac{t}{n}$ se calculează perioada de timp a unei oscilații, iar valorile obținute se trec în tabel.
- viii. Pentru a determina constanta elastică prin metoda dinamică, se folosește relația:

$$k = \frac{4\pi^2}{T^2} \left(M + \frac{m}{3} \right)$$

- ix. Constanta elastică medie \bar{k} se calculează făcând media aritmetică a celor trei valori ale constantei elastice pentru fiecare resort: $\bar{k} = \frac{k_1 + k_2 + k_3}{3}$.

- xiii. Eroarea relativă se calculează cu relația: $\frac{\Delta k}{k} = 2 \frac{\Delta \pi}{\pi} + 2 \frac{\Delta T}{T} + \frac{\Delta M + \Delta(\frac{m}{3})}{M + \frac{m}{3}}$,

$$\text{unde: } \frac{\Delta T}{T} = \frac{\Delta t}{t} + \frac{\Delta n}{n}.$$

- xiv. Eroarea absolută se calculează cu relația: $\Delta k = ak$, deoarece $\frac{\Delta k}{k} = a$.
- xv. Toate rezultatele obținute se trec în tabel.

Vă urăm succes!

II.3 Determinarea accelerației gravitaționale cu ajutorul unui pendul fizic

Introducere

Pendulul fizic este un corp solid (rigid) montat în așa fel încât să poată oscila liber într-un plan vertical, sub acțiunea forței gravitaționale, \vec{G} , în jurul unei axe care nu trece prin centrul său de masă, C. Dacă se scoate pendulul din poziția de echilibru, acesta efectuează o mișcare oscilatorie în jurul acestei poziții.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Pendulul%20Fizic/>

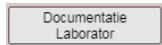
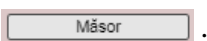
Prin accesarea link-ului se deschide pagina:

nr crt	m [Kg]	d [cm]	n	t [s]	T [s]	dT ² [cm ²]	d ² [cm ²]	g [m/s ²]	I _C [Kg·m ²]
1		10							
2		20							
3		30							
4		40							
5	10	10							
6		20							
7		30							
8		40							

Dispozitivul experimental

În fotografia din partea centrală a lucrării este prezentată o platbandă metalică, lungă de aproximativ 1 m, care este prevăzută în lungul ei cu opt orificii echidistante, dispuse simetric de o parte și de alta, a centrului de masă. Platbanda poate fi așezată, cu fiecare orificiu al ei, pe un suport fixat în perete, prevăzut cu un cuțit, din oțel de înaltă duritate, cu muchia în sus, muchie ce constituie axa de rotație. Așezând platbanda pe muchia cuțitului, folosind unul dintre orificii, se obține un pendul fizic de o anumită lungime d și un anumit moment de inerție I . Schimbând orificiul, se modifică atât d cât și I , astfel că pendulul va avea o altă perioadă de oscilație.

Instrucțiuni pentru simularea numerică:

1. Referatul de laborator al lucrării se accesează apăsând butonul  .
2. Pentru a efectua experimentul virtual se apasă butonul  .

The screenshot shows a virtual laboratory window titled "Laboratorul de Fizica: Determinarea accelerației gravitaționale cu pendulul fizic". It features a control panel with buttons for "Oscileaza", "Stop", "Sus", and "Jos", a data table, and a diagram of the pendulum setup. The diagram shows a pendulum bob suspended from a pivot point, with a coordinate system and a force vector G pointing downwards. The data table has columns for "nr crt", "m [Kg]", "d [cm]", "n", "t [s]", "T [s]", "d² [cm²]", "d² [cm²]", "g [m/s²]", and "I_G [kg·m²]".

nr crt	m [Kg]	d [cm]	n	t [s]	T [s]	d ² [cm ²]	d ² [cm ²]	g [m/s ²]	I _G [kg·m ²]
1		10							
2		20							
3		30							
4		40							
5	10	10							
6		20							
7		30							
8		40							

3. Pașii pentru înregistrarea datelor constau în:

- i. Se alege numărul de oscilații complete pe care să le efectueze pendulul fizic.
- ii. Se alege poziția pendulului fizic: cu orificiile negre sus sau jos, apăsând butoanele:



- iii. Se alege orificiul pe care să se sprijine bara modificând poziția cursorului:



- iv. Se pune bara în oscilație, apăsând butonul

- v. Când se începe numărarea oscilațiilor complete pe care le efectuează pendulul fizic se pornește și un cronometru pentru a determina timpul de oscilație.

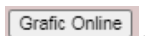
- vi. Când numărătoarea este completă se oprește cronometru și oscilația pendulului, apăsând butonul:

- vii. Se trece în tabelul de valori timpul obținut pentru oscilația pendulului pentru primul orificiu ales.

- viii. Se repetă pașii (iii-vii) până sunt efectuate determinările de timp pentru toate orificiile prevăzute pe pendulul fizic.

- ix. Se completează tabelul de valori și apoi se reprezintă grafic: $d\bar{T}^2 = f(d^2)$.

- x. Opțional, se poate utiliza și graficul realizat de program, apăsând butonul:



- xi. Se aproximează datele experimentale cu o funcție liniară.

- xii. Se determină panta dreptei obținute ($tg \theta$) și apoi se calculează accelerația gravitațională utilizând formula: $tg \theta = \frac{4\pi^2}{g}$.

Vă urăm succes!

II.4 Determinarea timpului de ciocnire a două bile

Introducere

Ciocnirea este un proces de interacțiune prin contact între două sau mai multe corpuri. Timpul în care corpurile sunt în contact se numește *timp de ciocnire*. Ciocnirea elastică durează un timp relativ scurt, iar înainte și după aceasta corpurile nu interacționează. La ciocnire corpurile se deformează în zona de contact, iar o parte din energia lor cinetică se transformă în energie potențială de deformare elastică.

Timpul de ciocnire variază de la secunde, pentru sfere mari din material elastic, la ordinul a 10^{-4} s pentru ciocniri între corpurile dure. Pentru ciocnirile particulelor elementare timpul de ciocnire ajunge la ordinul 10^{-18} s. În această lucrare se urmărește ciocnirea elastică a două bile. Deoarece timpul de ciocnire este foarte scurt, pentru o măsurătoare exactă, bilele sunt plasate într-un circuit electric compus dintr-o capacitate electrică C în serie cu o rezistență R . Pe durata ciocnirii (timpului de ciocnire) bilele metalice fiind în contact permit condensatorului C să se descarce prin rezistența R . Variația sarcinii electrice a condensatorului este proporțională cu timpul de ciocnire. Măsurând tensiunea pe condensator înainte de ciocnire (când este încărcat cu sarcină) și după ciocnire (când s-a descărcat parțial) putem calcula timpul de ciocnire.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Timp%20ciocnire/>

Prin accesarea link-ului se deschide pagina:

The screenshot shows a virtual laboratory interface with the following components:

- Title:** Laboratorul de Fizică: Determinarea timpului de ciocnire a două bile
- Navigation:** Documentație Laborator, Ciocnire oțel - oțel, Experiment
- Data Tables:**
 - Ciocnire oțel - oțel:** Table with 10 rows and 6 columns: C [μF], R [Ω], RC [μs], U_i [V], U_c [V], t*10⁻⁵ [s].
 - Ciocnire oțel - plumb:** Table with 5 rows and 6 columns: C [μF], R [Ω], RC [μs], U_i [V], U_c [V], t*10⁻⁵ [s].
 - Ciocnire oțel - cupru:** Table with 5 rows and 6 columns: C [μF], R [Ω], RC [μs], U_i [V], U_c [V], t*10⁻⁵ [s].
- Footer:** Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

Dispozitivul experimental

Dispozitivul experimental constă dintr-un cadru pe care sunt fixate firele de suspensie a două bile metalice. Circuitul electric atașat este compus dintr-o rezistență electrică reglabilă, voltmetru, un condensator cu capacitatea electrică $C = 22\mu F$ și o baterie de încărcare a condensatorului. Comutatorul principal, aflat central în față, pe poziția "Încarcă" determină încărcarea condensatorului de la sursa de tensiune, iar pe poziția "Măsoară" se citește descărcarea condensatorului. Apăsând butonul "Cădere" se ciocnesc bilele deoarece se decuplează electromagnetul care ține bila și se activează celălalt electromagnet care prinde bila ciocnită. Ciocnirile vor fi între două bile de oțel, o bilă de oțel și una de plumb, iar apoi o bilă de oțel și alta de cupru.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se deschide apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

3. Pentru înregistrarea datelor:

- i. Se apasă butonul .
- ii. Se poziționează rezistența din circuit pe valoarea dorită cu ajutorul cursorului.
- iii. Pentru a încărca cu sarcină electrică condensatorul C, se poziționează comutatorul pe poziția . Încărcarea condensatorului e maximă când bateria roșie de lângă el indică acest lucru.

- iv. După încărcarea condensatorului se mută comutatorul pe poziția , moment în care se citește de pe voltmetru valoarea tensiunii de încărcare. Aceasta se notează în tabel, în coloana corespunzătoare, U_i .
- v. În momentul citirii tensiunii U_i , se apasă butonul pentru a dezactiva electromagnetul de care e prins bila și astfel, ciocnirea să aibă loc.
- vi. Se citește valoarea tensiunii condensatorului obținută în momentul ciocnirii, U_c , și se trece în tabel, în coloana corespunzătoare.
- vii. Pentru a determina timpul de ciocnire al bilelor, se folosește formula:

$$\tau = RC \ln \frac{U_i}{U_c},$$

Se dau:

- Rezistențele: $R_1 = 5 \Omega$ și $R_2 = 10 \Omega$;
 - Corespondența: $\mu = 10^{-6}$.
- viii. Se repetă ciocnirile de 5 ori pentru fiecare rezistență.
- ix. Determinarea timpului de ciocnire pentru bile diferite: din oțel și respectiv plumb, se obține prin apăsarea butonului , și se urmează pașii **ii-viii**.
- x. Determinarea timpului de ciocnire pentru bile din oțel și respectiv din cupru, se obține prin apăsarea butonului , și se urmează pașii **ii-viii**.

Vă urăm succes!

II.5 Studiul undelor staționare transversale în corzi vibrante

Introducere

Dacă direcția de oscilație a particulelor dintr-un mediu elastic este perpendiculară pe direcția de propagare a undei se numește undă transversală. Undele transversale se propagă numai în mediile solide elastice uni, bi-dimensionale și la suprafața lichidelor. Câteva exemple de unde transversale sunt:

- valurile produse de o piatra aruncată în apă;
- ciupirea unui instrument muzical cu coarde;
- lovirea membranei unei tobe.

Simularea numerică a lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/UndeStationare/>

Prin accesarea link-ului se deschide pagina de web, de mai jos:

The screenshot shows a web application titled "Laborator de Fizica - Studiul undelor stationare transversale in corzile vibrante". It features a data table for recording experimental and calculated values, and a photograph of the physical experimental setup.

m [kg]	T [N]	v [m/s]	Niu [Hz]	1	2	3	4	5	Lamda [m]	1	2	3	4	5
			calculat						calculat					
			experimental						experimental					
			calculat						calculat					
			experimental						experimental					

The photograph shows the experimental setup: a string fixed at one end to a wall and attached to a vibrator at the other. The vibrator is a GWINSTEK SFG-1013 function generator. Two metal weights are suspended from the string. A whiteboard in the background displays the formula $f = \frac{v}{\lambda}$ and the value $\lambda = 0.45 \text{ m}$.

Dispozitivul experimental

Undele staționare transversale în corzile vibrante pot fi studiate online folosind o simulare numerică a unui dispozitiv experimental existent fizic, on site, la Departamentul de Fizică și Chimie a UTCN. Dispozitivul experimental este prezentat în cele trei poze, din dreapta, din pagina *web*. Pe un stativ de plexiglas este montat un fir de Cu care este mediul solid elastic de-a lungul căruia se propaga undele transversale mecanice. Capetele acestui fir sunt conectate printr-un amplificator la un generator de audio - frecvență. La capătul din stânga sunt suspendate două corpuri cu mase diferite iar capătul din dreapta este fix. Prin fir

trece un curent electric de audio- frecvența provenit de la generator. Iar aproape de capătul din dreapta a stativului de plexiglas este montat un magnet permanent. Firul trece printre polii magnetului permanent. În urma interacțiunii dintre curentul electric care trece prin fir și câmpul magnetic constant produs de magnetul permanent apare o forță oscilatorie cu aceeași frecvență ca a curentului electric care realizează excitarea corzii.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se deschide apăsând butonul: .
2. Pentru a efectua experimentul virtual se apasă butonul: .

Laborator de Fizica - Studiul undelor stationare transversale in corzile vibrante

Document Masuratori Student: Grupa:

Masa [Kg] 0.445

Amplificator

Generator 25.0 Hz

niu = 25.0 [Hz]

m [kg]	T [N]	v [m/s]	Niu [Hz]	1	2	3	4	5	Lamda [m]	1	2	3	4	5
			calculat						calculat					
			experimental						experimental					
			calculat						calculat					
			experimental						experimental					

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

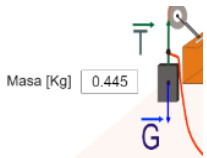

3. Etapele pentru înregistrarea datelor sunt:

- Se folosesc două corpuri de masă diferită pentru a tensiona firul de cupru (poza din dreapta). Valorile maselor celor două corpuri se trec, în tabel, în kg, în prima coloană.
- Se calculează forța de tensionare a firului, în tabel coloana numărul doi.
- Se calculează viteza de propagare a undei transversale.
- Se calculează frecvențele și lungimile de undă conform documentației de laborator.

Se dau:

- Lungimea firului de Cupru, $l = 1 \text{ m}$.
- Densitatea liniară, $\mu = 1.25 \text{ g/m}$.
- Accelerația gravitațională, $g = 9.807 \text{ m/s}^2$.
- Diametrul firului de Cupru, $\varphi = 0.45 \text{ mm}$.



- v. În momentul în care este completat tabelul cu toate datele calculate se scrie valoarea masei, în kg, în secțiunea din imaginea alăturată.
- 
- vi. Se modifică frecvența citită pe generator cu ajutorul cursorului din imaginea alăturată,  luându-se ca reper valoarea calculată a fiecărei frecvențe în funcție de n, astfel încât să se observe armonica fundamentală și armonicile superioare n=1, 2, 3, 4, 5. Notând în tabel valoarea experimentală.
- vii. Pentru a determina lungimea de undă experimentală, se folosește formula:
- $$\lambda_{exp} = \frac{v}{\nu_{exp}}$$
- viii. După determinarea valorilor frecvenței experimentale și calculul lungimilor de undă, valorile obținute ale acestora se trec în tabel.

Vă urăm succes!

- vi. Pentru a determina raportul căldurilor molare sau exponentul adiabatic, se folosește formula:

$$\gamma = \frac{h_1}{h_1 - h_2}$$

- vii. Se calculează media aritmetică a raportului căldurilor molare și se trece în tabel în coloana corespunzătoare.

- viii. Eroarea relativă se determină cu relația:

$$\frac{\Delta\gamma}{\gamma} = \frac{(h_1 + h_2)\Delta h}{h_1(h_1 - h_2)}$$

unde Δh este eroarea absolută de măsurare a denivelării de pe manometru. Se va înmulți rezultatul care se obține din calcul cu 100, pentru ca valoarea erorii obținută să fie exprimată în %.

- ix. Eroarea absolută se determină cu relația:

$$\Delta\gamma = \frac{\Delta\gamma}{\gamma}\gamma$$

unde $\frac{\Delta\gamma}{\gamma}$ este adimensional.

Vă urăm succes!

III.2 Studiul unei pompe de căldură

Introducere

Prin studierea pompei de căldură se înțelege funcționarea unor sisteme de încălzire și răcire cum ar fi: frigidere, congelatoare, aparate de aer condiționat și invertoare de căldură. Frigiderele și pompele de căldură diferă între ele prin faptul că frigiderele întrețin într-o încălț o temperatură T_f mai coborâtă decât temperatura T_c a mediului înconjurător, în timp ce pompele de căldură transferă unei încălț aflată la o temperatură mai mare decât cea a mediului înconjurător cantități de căldură suplimentare preluate de la surse mai reci. În cele ce urmează se poate observa și măsura energia, temperatura apoi calcula puterea unui încălzitor electric și a unei pompei de căldură.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Studiul%20pompei%20de%20caldura/>

Prin accesarea link-ului se deschide pagina:

Laborator de Fizica: Studiul unei pompei de caldura

Student: Grupa:

Document

motor termic
 $\eta = \frac{L}{Q_1}$

frigidere
pompa de caldura
 $\varepsilon = \frac{Q_1}{L}$

Tabel	Incalzitor electric			Pompa de caldura		
	Contor [kWh]	Temperatura [°C]	timp [s]	Contor [kWh]	Temperatura [°C]	timp [s]
Initial	<input type="text"/>	<input type="text"/>	0	<input type="text"/>	<input type="text"/>	0
Final	<input type="text"/>	<input type="text"/>	4x60	<input type="text"/>	<input type="text"/>	8x60
Valori	<input type="text"/>	<input type="text"/>	$t_0 = 240$	<input type="text"/>	<input type="text"/>	$t = 480$
Puterea	<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>	

Toate drepturile rezervate © Departamentul de Fizica si Chimie, Universitatea Tehnica din Cluj-Napoca

Dispozitivul experimental

În laborator, pompa de căldură a fost realizată folosind instalația de producere a frigului de la un frigider cu compresor. Evaporatorul a fost desfăcut și plasat în aer pe un stativ vertical, iar condensatorul a fost înlocuit cu o țevă de Cupru în serpentină, introdusă într-un calorimetru. Compresorul, alimentat de la rețeaua de 220 V printr-un contor electric, consumă

o energie electrică ΔE necesară pentru ca uleiul din calorimetru să primească o anumită cantitate de energie sub formă de căldură.

Instrucțiuni pentru simularea numerică:

1. Documentația lucrării se deschide apăsând butonul: .

2. Pentru a efectua experimentul virtual se apasă butonul: .

Laborator de Fizica: Studiul unei pompe de caldura

Student: _____ Grupa: _____

sursa calda

Q_1

Q_2

sursa rece

motor termic

$\eta = \frac{L}{Q_1}$

sursa calda

Q_1

Q_2

sursa rece

frigider
pompa de caldura

$\varepsilon = \frac{Q_1}{L}$

Tabel	Incalzitor electric			Pompa de caldura		
Citire	Contor [kWh]	Temperatura [°C]	timp [s]	Contor [kWh]	Temperatura [°C]	timp [s]
Initial	<input type="text"/>	<input type="text"/>	0	<input type="text"/>	<input type="text"/>	0
Final	<input type="text"/>	<input type="text"/>	4x60	<input type="text"/>	<input type="text"/>	8x60
Valori	<input type="text"/>	<input type="text"/>	$t_0 = 240$	<input type="text"/>	<input type="text"/>	$t = 480$
Puterea	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Toate drepturile rezervate © Departamentul de Fizica și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Etapele pentru înregistrarea datelor sunt:

i. Se citește temperatura și poziția contorului de energie, se apasă butonul .

ii. Se trec valorile în tabel pe rândul cu datele inițiale.

iii. Se apasă butonul: , iar după 4 minute se oprește apăsând butonul

.

iv. Se citește temperatura și indicația contorului de energie.

v. Se trec valorile în tabel pe rândul cu datele finale.

vi. Se repetă măsurătoarea pentru pompa de căldură, se apasă butonul .

vii. Se apasă butonul: , iar după 8 minute se oprește apăsând butonul

.

viii. Se citește temperatura și indicația contorului de energie.

ix. Se trec valorile în tabel pe rândul cu datele finale.

x. Se determină puterile consumată folosind relațiile:

$$P_0 = \frac{\Delta E_0}{t_0} \quad \text{și} \quad P = \frac{\Delta E}{t}$$

xi. Valorile calculate se trec în tabel.

xii. Se determină capacitatea calorică a uleiului:

$$C = \frac{\Delta E_0}{\Delta T_0}.$$

xiii. Se determină eficiența, ε a pompei de căldură:

$$\varepsilon = \frac{\Delta E_0}{\Delta T_0} \cdot \frac{\Delta T}{\Delta E},$$

$$\varepsilon = C \cdot \frac{\Delta T}{\Delta E}.$$

xiv. Se determină căldura cedată de pompă ($Q_{\text{ced}} = \varepsilon \Delta E$).

xv. Se determină eficiența ciclului cu ajutorul relației:

$$\varepsilon_0 = \frac{\varepsilon}{\eta},$$

unde $\eta = 0,6$ este randamentul compresorului.

Vă urăm succes!

III.3 Determinarea coeficientului de vâscozitate al lichidelor

Introducere

Vâscozitatea reprezintă proprietatea fluidelor de a se opune deformațiilor (de a opune rezistență la schimbarea formei). Aceasta se manifestă la fluidele în mișcare, prin apariția unor eforturi tangențiale, fiind datorată frecării dintre straturile alăturate de fluid, care se deplasează unele față de altele. Vâscozitatea se mai poate defini ca fiind o proprietate comună tuturor fluidelor, prin care, cu forțe (F) suficient de mici, se pot produce deformații oricât de mari, cu viteze de deformare mici:

$$F = \eta S \frac{dv}{dr},$$

care este legea frecării interne dată de Newton și care arată că forța de frecare internă este direct proporțională cu aria suprafeței straturilor fluide și cu gradientul vitezei pe direcția normală a vitezei. Mărimea η caracterizează proprietatea de vâscozitate a fluidului și se numește coeficient de vâscozitate dinamică sau vâscozitate dinamică. Sensul fizic al acestei mărimi este acela de tensiune tangențială care se dezvoltă în interiorul unui fluid omogen, când gradientul vitezei este unitar. Unitatea de măsură a vâscozității dinamice în Sistemul Internațional este $[N \text{ s} / \text{m}^2]$.

Pentru a măsura vâscozitatea unui lichid se va folosi metoda lui Stokes. Această metodă constă în măsurarea vitezei limită de cădere a unei sfere (bile) într-un lichid.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/CoeficientulVascozitateLichide/>

Prin accesarea link-ului se deschide pagina:

ρ_{bile} [kg/m ³]	ρ_{lich} [kg/m ³]	r [mm]	h [m]	t [s]	t_{mediu} [s]	v_{lim} [m/s]	η [Ns/m ²]	η [m ² /s]	Δt [s]	Δh [m]	Δt [s]	$\Delta \eta$ [%]
			0.4									

Totale drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

Dispozitivul experimental

Pentru determinarea coeficientului de vâscozitate al lichidelor, în laborator, este folosit dispozitivul experimental din imaginea din dreapta. Cele două tuburi sunt umplute cu două tipuri de ulei: hidraulic și de motor, care au densități diferite. În cele două tuburi sunt introduse două bile metalice de aceeași masă și diametru, care sunt cunoscute. Bilele sunt lăsate să cadă pe o înălțime dată cronometrând timpul de cădere al lor, astfel se poate determina viteza de cădere a bilelor, de care este nevoie pentru a determina coeficientul de vâscozitate utilizând metoda lui Stokes. Simulările reproduc procedeul fizic din laborator.

Instrucțiuni pentru simularea numerică:

1. Pentru accesarea părții teoretice a lucrării se apăsă butonul Documentatie Lab
2. Pentru a efectua experimentul virtual se apăsă butonul Masor.

Laboratorul de Fizica: Determinarea coeficientului de vâscozitate al lichidelor

Student: _____ Grupa: _____

Documentatie Laborator

Vasozimetru

Măsur

Start Reset

ρ_{ulei} [Kg/m ³]	ρ_{bila} [Kg/m ³]	r [mm]	h [m]	t [s]	t_{mediu} [s]	v_s [m/s]	η [Ns/m ²]	ϕ [m ² /s]	Δr [m]	Δh [m]	Δt [s]	$\Delta \eta/\eta$ [%]
800.0		3.1	0.4									

$\rho_{\text{ulei}} = 800.0 \text{ kg/m}^3$

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pentru ca în tubul cilindric să apară bila a cărei viteză de cădere se determină, în tabel trebuie scrisă valoarea razei bilei, dată în referatul de laborator (sau conform indicațiilor cadrului didactic).

Laboratorul de Fizica: Determinarea coeficientului de vâscozitate al lichidelor

Student: _____ Grupa: _____

Documentatie Laborator

Vasozimetru

Măsur


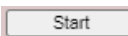
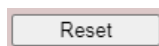
Start Reset

ρ_{ulei} [Kg/m ³]	ρ_{bila} [Kg/m ³]	r [mm]	h [m]	t [s]	t_{mediu} [s]	v_s [m/s]	η [Ns/m ²]	ϕ [m ² /s]	Δr [m]	Δh [m]	Δt [s]	$\Delta \eta/\eta$ [%]
800.0		3.1	0.4									

$\rho_{\text{ulei}} = 800.0 \text{ kg/m}^3$

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

4. Pentru realizarea măsurătorilor se procedează astfel:

- i. Se alege densitatea dorită în intervalul $\rho = (800 - 950) \text{ kg/m}^3$, pentru uleiul studiat cu ajutorul cursorului .
- ii. Se apasă butonul , iar bila va începe să cadă.
- iii. Se cronometrează timpul de cădere al bilei pe distanța de 40 cm, între reperele 50 și 10 de pe scala gradată a tubului cilindric (pentru ușurință se observă și apariția unei linii roșii în dreptul celor două repere).
- iv. Înainte de a efectua o nouă măsurătoare se apasă pe butonul .
- v. Se repetă măsurătorile de cinci ori pentru fiecare densitate de ulei aleasă, valoarea fiecărui timp se trece în tabel.
- vi. Se calculează media aritmetică a timpilor obținuți (care se trece în tabel pentru fiecare tip de ulei cu densitate diferită), iar cu aceasta se calculează viteza de cădere a bilei pentru cele două tipuri de ulei considerate.
- vii. Se calculează densitatea bilei (ρ_0), cunoscând masa (0,92 g) și raza ei (3.1 mm).
- viii. Se calculează coeficientul de vâscozitate dinamică utilizând relația:

$$\eta = \frac{2}{9} \cdot \frac{g(\rho_0 - \rho)r^2}{v}$$

- ix. Se calculează coeficientul de vâscozitate cinematică utilizând relația:

$$\vartheta = \frac{\eta}{\rho}$$

- x. În final se vor calcula erorile relative care s-au făcut în cazul determinării coeficientului de vâscozitate dinamică utilizând relația:

$$\frac{\Delta\eta}{\eta} = \frac{\Delta h}{h} + \frac{\Delta t}{t} + 2 \cdot \frac{\Delta r}{r},$$

unde: Δh , Δt , Δr - sunt erori absolute și se vor considera în momentul efectuării laboratorului de către cadrul didactic împreună cu studenții.

- xi. Valorile obținute pentru aceste erori se trec în tabel, exprimate în procente.

Vă urăm succes!

III.4 Determinarea coeficientului de vâscozitate dinamică al aerului

Introducere

Dinamica fluidelor (hidrodinamica) este acea parte a fizicii care, în general se ocupă cu studiul mișcării fluidelor. Spre deosebire de solide, la lichidele în mișcare, pe lângă densitate, trebuie să ținem cont și de deplasarea relativă a *straturilor de fluid* unele față de altele. Aceasta implică forțe de frecare interne, respectiv viteze de curgere diferite. Acest comportament al fluidelor se numește vâscozitate.

În prezent există mai multe metode de determinare a coeficientului de vâscozitate atât pentru lichide cât și pentru gaze. Câteva din aceste metode sunt:

- amortizarea oscilațiilor unui pendul de torsiune;
- căderea unei bile în lichid;
- curgerea fluidului printr-un tub capilar.

Pentru determinarea coeficientului de vâscozitate dinamică al aerului vom studia curgerea fluidului-aer printr-un tub capilar.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/VasozitateAer/>

Prin accesarea link-ului se deschide pagina:

Laboratorul de Fizică: Determinarea coeficientului de vâscozitate dinamică a aerului

Student: _____ Grupa: _____

Documentație laborator

Experiment



Densitatea apei = 996.84 [kg/m³]

T _{camera} [°C]	R [10 ⁻³ m]	l [10 ⁻³ m]	V [10 ⁻³ m ³]	Δt [s]	Δh [10 ⁻³ m]	η [10 ⁻³ Ns/m ²]	η _{mediu} [Ns/m ²]
22	4	65	100				
			200				
			300				
			400				
			500				
			600				
			700				
			800				

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

Dispozitivul experimental

Pentru determinarea coeficientului de vâscozitate dinamică a aerului, în laborator, este folosit dispozitivul experimental prezentat în cele două fotografii din dreapta. Vasul de sus este

umplut cu apă. La deschiderea robinetului apa va curge din vasul de sus în vasul de jos, iar volumul de apă care curge este înlocuit cu aerul ce se absoarbe din atmosferă prin tubul capilar. Datorită forțelor de frecare internă ale aerului, presiunea de la capetele capilarului nu va fi aceeași, iar diferența de presiune Δp o vom obține măsurând denivelarea Δh din manometrul cu apă.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

Laboratorul de Fizică: Determinarea coeficientului de vâscozitate dinamică a aerului

Student: _____ Grupa: _____

Documentație laborator

Experiment

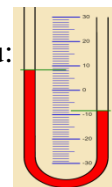
Densitatea apei = 996.84 [kg/m³]

T _{camera} [°C]	R [10 ⁻³ m]	l [10 ⁻³ m]	V [10 ⁻³ m ³]	Δt [s]	Δh [10 ⁻³ m]	η [10 ⁻² N·s/m ²]	η_{mediu} [10 ⁻² N·s/m ²]
22	4	65	100				
			200				
			300				
			400				
			500				
			600				
			700				
			800				

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pașii pentru înregistrarea datelor constau în:

- i. Se va deschide robinetul mutând cursorul spre dreapta. Se lasă să curgă din vas un volum de apă variabil conform indicațiilor din tabel. Vasul cu apă are capacitatea totală de 1000 ml.
- ii. După deschiderea robinetului apare o diferență de nivel Δh pe manometru: Denivelarea citită în mm, se va trece în tabel transformată în metri.
- iii. În momentul deschiderii robinetului se pornește un cronometru.
- iv. Cronometrul se oprește în momentul în care din vasul gradat a curs volumul de apă corespunzător, și se mută cursorul spre stânga pentru a închide robinetul.
- v. Intervalul de timp înregistrat, Δt , se trece în tabel în dreptul volumului de apă corespunzător.
- vi. Înainte de a efectua o nouă măsurătoare se apasă butonul (cu robinetul închis).



vii. Pentru a determina coeficientul de vâscozitate dinamică a aerului, se folosește formula:

$$\eta = \frac{\pi \rho g \Delta h \Delta t R^4}{8 V l}.$$

Se dau:

- Densitatea apei (este dependentă de temperatură) se poate citi deasupra tabelului.
Astfel, la temperatura de 22°C, aceasta este: Densitatea apei = 996.84 [kg/m³].
- Corespondența: 1 l = 1 dm³.
- Accelerația gravitațională: $g = 9.807 \text{ m/s}^2$.
- Raza tubului capilar: $R = 0.4 \text{ mm}$;
- Lungimea tubului capilar: $l = 65 \text{ mm}$.

viii. După determinarea valorilor vâscozității în fiecare caz, se va calcula media aritmetică a valorilor obținute. Aceasta se trece în ultima coloana din tabel.

Vă urăm succes!

IV Electricitate și Magnetism

IV.1 Studiul conductibilității electrice a metalelor

Introducere

Curentul electric reprezintă o mișcare dirijată a particulelor încărcate electric, fiind forma de manifestare a efectului acțiunii unui câmp electric asupra sarcinilor electrice care se găsesc libere în diverse materiale. Astfel, mișcarea ordonată a purtătorilor de sarcină se suprapune peste mișcarea haotică de agitație termică care există implicit în corpurile aflate la temperaturi mai mari de 0 K. În solide, curenții electrice sunt datorati mișcării ordonate a electronilor. La metale purtătorii de sarcină, care participă la procesul de conducție, sunt electronii din banda de valență parțial ocupată. Valoarea conductibilității electrice a solidelor este determinată în esență de concentrația purtătorilor de sarcină care pot participa la procesul de conducție și de împrăștierea pe care aceștia o suferă în mișcarea lor prin solid, adică de mobilitatea lor μ . În cazul metalelor, unde concentrația purtătorilor rămâne practic constantă, creșterea temperaturii duce la creșterea rezistivității electrice prin scăderea mobilității purtătorilor de sarcină electrică, ca rezultat al creșterii împrăștierii electronilor de către vibrațiile rețelei cristaline.

Simularea lucrării de laborator se găsește la adresa:


<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/ConductibilitateElectrica/>

Prin accesarea link-ului se deschide pagina:

Laboratorul de Fizica: Studiul conductibilității electrice a metalelor
Student: _____
Grupa: _____

Documentatie Laborator

Experiment



Grafic Online
l = 384 m
φ = 0,132 mm

t [°C]	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
R [Ω]															
ρ															
[10 ⁻³ Ω·m]															
σ															
[10 ⁻³ Ω ⁻¹ ·m ⁻¹]															
μ															
[10 ⁻³ m ² /Vs]															

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

Dispozitivul experimental

Instalația experimentală constă dintr-un fir de cupru bobinat, introdus într-un vas termorezistent care este așezat pe o plită electrică. Pentru a fi încălzită în mod omogen, bobina care conține firul din cupru este scufundată în apă. Se urmărește modificarea rezistenței electrice a firului odată cu creșterea temperaturii apei. Rezistența firului de cupru se măsoară cu ajutorul unui multimetru, iar temperatura cu un termometru (sau termocuplu) care se găsește în apropierea bobinei.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

t [°C]	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
R [Ω]															
ρ															
σ															
σ [10 ⁸ Ω ⁻¹ m ⁻¹]															
σ [10 ⁸ Ω ⁻¹ m ⁻¹]															
σ [10 ⁸ Ω ⁻¹ m ⁻¹]															

3. Pașii pentru înregistrarea datelor constau în:

- i. Se notează în tabel valoarea rezistenței electrice a firului la temperatura de 20°C.
- ii. Se pornește încălzirea firului de cupru prin apăsarea butonului .
- iii. Se notează din 5 în 5 °C valoarea rezistenței electrice a firului de cupru.
- iv. Pentru a determina rezistivitatea, se folosește formula:

$$\rho = R \frac{S}{l}$$

Se dau:

- Lungimea firului: .
- Diametrul firului: .

v. Pentru determinarea conductibilității electrice se utilizează formula:

$$\sigma = \frac{1}{\rho}$$

vi. Se calculează mobilitatea purtătorilor de sarcină, cu formula:

$$\mu = \frac{\sigma}{n \cdot e}.$$

Se dau:

- Numărul purtătorilor de sarcină din cupru: $n = 8.4 \cdot 10^{28} \text{ e}^-/\text{m}^3$;
- Sarcina electronului: $e = 1.6 \cdot 10^{-19} \text{ C}$.

vii. Se trasează pe hârtie milimetrică sau în programul Excel dependența $R = f(t)$.

Vă urăm succes!

IV.2 Studiul efectului Hall

Introducere

În domeniul măsurătorilor electrotehnice ca și în domeniul calculatoarelor electronice și automată, efectul Hall este utilizat la măsurarea intensității câmpului magnetic și a inducției magnetice, la măsurarea pierderilor de energie în fier, a curenților continui foarte intenși, la măsurarea puterii în rețelele de curent continuu și alternativ, a defazajului și a factorului de putere.

Efectul Hall constă în apariția unei diferențe de potențial pe laturile unei probe atunci când aceasta se găsește într-un câmp magnetic și este străbătută de curent.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/EfectulHall/>

Prin accesarea link-ului se deschide pagina:

I [A]	B [T]	Is [mA]	UB [mV]	RH [VmA]	$n \cdot 10^{20}$ [e/m ³]	σ [$\Omega^{-1} m^{-1}$]	μ [m ² Vs]
1.5	0.20	2.0					
		4.0					
		6.0					
		8.0					
2	0.25	2.0					
		4.0					
		6.0					
		8.0					
2.5	0.28	2.0					
		4.0					
		6.0					
		8.0					

Dispozitivul experimental

Dispozitivul experimental (vezi poza din dreapta sus a paginii web) se compune dintr-un electromagnet confecționat din tole ARMCO în formă de cadru, pentru a concentra cât mai bine câmpul magnetic. Fiecare bobină are un număr mare de spire, valoarea maximă a curentului de alimentare, a bobinelor electromagnetului fiind de 3A.

Placă semiconductoare de Germaniu care se studiază, este introdusă între polii electromagnetului și este alimentată de la un redresor, cu un curent de comandă, măsurat cu miliampermetrul, mA. Curentul maxim admis prin plăcuță este 10 mA. Tensiunea Hall se măsoară între laturile plăcuței de Germaniu cu ajutorul unui multimetru digital.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării în format pdf. se deschide apăsând butonul: Document .

2. Pentru a efectua experimentul virtual se apasă butonul: Masuratori .

I [A]	B [T]	Is [mA]	IB/a [AT/m]	UH [mV]	RH [Vn/AT]	$n \cdot 10^{22}$ [e/m ³]	σ [Ω·m] ⁻¹	μ [m ² /Vs]
1.5	0.20	2.0						
		4.0						
		6.0						
		8.0						
2	0.25	2.0						
		4.0						
		6.0						
		8.0						
2.5	0.28	2.0						
		4.0						
		6.0						
		8.0						

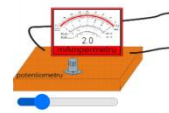
3. Etapele pentru înregistrarea datelor sunt:

i. Se citește valoarea diferenței de potențial în absența câmpului magnetic (intensitatea curentului de alimentare a electromagneților $I = 0$). Existența unei diferențe de potențial se datorează asimetriei contactelor laterale față de axa orizontală a plăcii semiconductoare și este egală cu căderea de tensiune pe placă între aceste contacte.

ii. Se fixează valoarea curentului $I = 1.5 A$ prin întreg sistemul, prin scrierea valorii 1.5 pe generator.



iii. Se modifică curentul I_s prin plăcuța semiconductoare de studiat cu ajutorul cursorului din 2 mA în 2mA până la 8 mA (vezi figura din partea dreaptă), apoi se citește valoarea tensiunii electrice de pe multimetrul digital.



ATENȚIE! Se ține seama că tensiunea indicată de voltmetru este suma tensiunii de asimetrie și a tensiunii Hall. Se repetă măsurătorile pentru alte două valori ale curentului din electromagneți $I = 2 A$ și $I = 2.5 A$.

iv. Se calculează pentru fiecare dintre cele 12 valori experimentale ale lui U_H obținute pentru un anumit câmp B și un anumit curent I raportul $\frac{IB}{a}$.

v. Se trec datele în tabel.

vi. Se trasează graficul $U_H = f\left(\frac{IB}{a}\right)$ pe foaia milimetrică sau în Excel, apoi se calculează panta acestuia care reprezintă constanta Hall, R_H .

ATENȚIE la unitățile de măsură! Se trece valoarea în tabel.

vii. Se calculează concentrația de purtători de sarcină cu relația:

$$n = \frac{1}{R_H e};$$

viii. Se calculează conductibilitatea electrică cu relația:

$$\sigma = \frac{c}{R \cdot a \cdot b},$$

unde: R este rezistența semiconductorului cu valoarea de 400 Ω , iar a, b, c, sunt dimensiunile plăcuței cu grosimea a = 1.5 mm, lățimea b=2.0 mm și lungimea c=12 mm.

ix. Mobilitatea se calculează cu relația:

$$\mu = R_H \sigma.$$

x. Datele obținute se trec în tabelul.

Vă urăm succes!

V Optică

V.1 Etalonarea unui spectroscop. Analiză spectrală calitativă

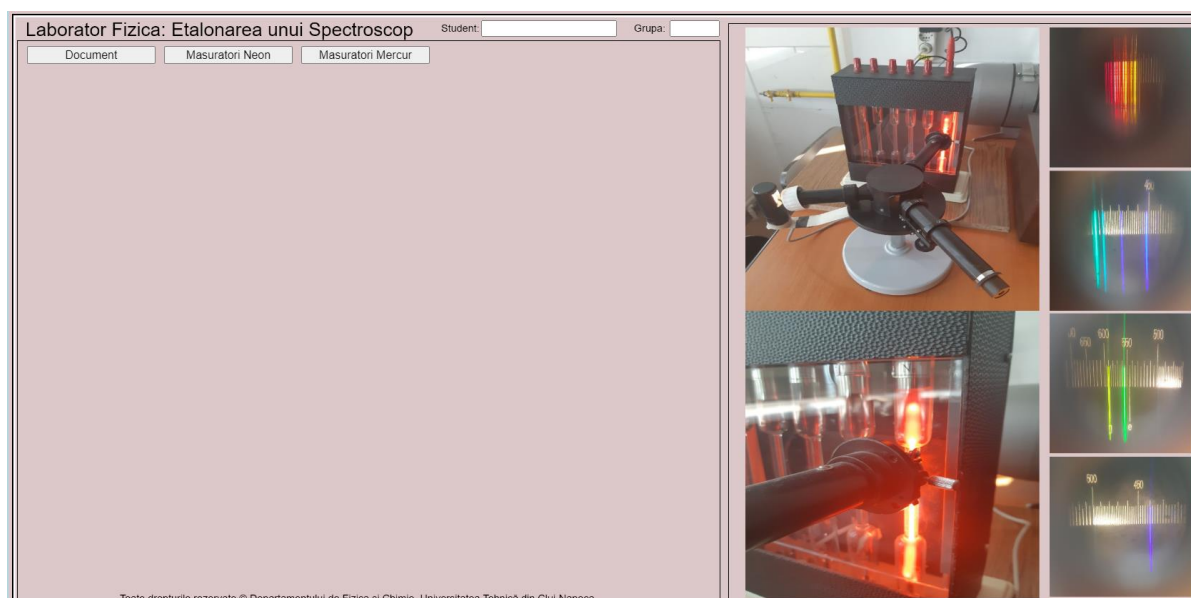
Introducere

Spectroscopia optică (caracterizată pentru domeniile vizibil, ultraviolet și infraroșu) permite identificarea elementelor componente din materiale și substanțe. De asemenea poate fi utilizată pentru determinarea structurii unui material sau stabilirea concentrației unui element specific dintr-un compus. Utilitatea spectroscopiei este demonstrată și aplicată în aproape toate domeniile de activitate cum ar fi: fizică, astrofizică, chimie, biologie, medicină, protecția mediului, industria alimentară.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Spectroscopul/>

Prin accesarea link-ului se deschide pagina:



Dispozitivul experimental

În laborator analiza spectrală se efectuează cu ajutorul spectroscopului. Acesta este format din patru elemente principale: **colimator** al fantei prevăzut cu o fantă reglabilă, **prisma** care separă lumina de diferite lungimi de undă din fasciculul incident este piesa cea mai importantă a spectroscopului. Deoarece deviația produsă de prismă crește cu scăderea lungimii de undă, lumina violet este deviată cel mai mult, iar lumina roșie este deviată cel mai puțin. **Luneta**: reglată pentru infinit (care permite observarea spectrului produs de prismă), formată din lentilă în al cărui plan focal se formează imaginile fantei, adică spectrul luminii studiate și ocularul care permite observarea imaginilor date de lentilă, **colimator** al scării micrometrice:

care suprapune peste imaginea spectrului studiat imaginea unui sistem de fire reticulare, necesare pentru stabilirea liniilor spectrale. De obicei pentru stabilirea poziției liniilor spectrale se utilizează dispozitive micrometrice.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării în format pdf. se deschide apăsând butonul: Document.
2. Efectuarea experimentului virtual are două părți:
 - i. Observarea și măsurarea spectrului de Neon. Accesând butonul Masuratori Neon se deschide următoarea pagină:

Laborator Fizica: Etalonarea unui Spectroscop Student: _____ Grupa: _____

Document Masuratori Neon Masuratori Mercur

00 div. 400 div. 500 div. 600 div. 700 div.

Date Neon	deviere = 0								
lungimea de unda [Å]	4964	5031	5330	5400	5760	5852	5945	6143	6402
pozitia liniei [div.]									

Grafic Online

Toate drepturile rezervate © Departamentul de Fizica și Chimie, Universitatea Tehnică din Cluj-Napoca

- ii. Observarea și măsurarea spectrului de Mercur. Accesând butonul Masuratori Mercur se deschide următoarea pagină:

Laborator Fizica: Etalonarea unui Spectroscop Student: _____ Grupa: _____

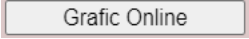
Document Masuratori Neon Masuratori Mercur

400 div. 500 div. 600 div. 700 div.

Date Mercur									
lungimea de unda [Å]									
pozitia liniei [div.]									

Grafic Online

Toate drepturile rezervate © Departamentul de Fizica și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Etapele pentru înregistrarea datelor sunt:
 - i. La măsurătorile de Neon se alege devierea în intervalul $[-50; 50]$ cu ajutorul cursorului de sub spectrul Neonului din pagina web. Apoi se identifică culorile corespunzătoare lungimilor de undă din tabel și se citește valoarea corespunzătoare poziției liniei.
 - ii. Aceste valori se trec în tabel.
 - iii. Se desenează graficul $\lambda (\text{Å}) = f(\text{poziția}(\text{div.}))$ pe foaia milimetrică sau într-un program de grafică precum Excel. Datele obținute se fitează cu o funcție liniară. Prin apăsarea butonului  se poate vizualiza graficul etalon.
 - iv. La măsurătorile de Mercur trebuie citită poziția liniei corespunzătoare culorii alese. Se completează tabelul cu datele. Apoi din graficul etalon al Neonului se identifică valoarea lungimii de undă corespunzătoare poziției liniei citite anterior din scala spectrului.
 - v. Datele obținute se trec în tabel.

Vă urăm succes!

V.2 Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină

Introducere

Fotometria este acea parte a Fizicii care se ocupă cu metodele și mijloacele de măsurare ale mărimilor caracteristice radiațiilor din domeniul vizibil. Lumina este un stimul capabil să impresioneze ochiul și produce în creier senzații specifice, astfel:

- *senzația de intensitate* care este dependentă de energia luminoasă care cade în unitatea de timp pe unitatea de suprafață a retinei și variază în funcție de energia izvorului luminos și distanța până la acesta;
- *senzația de culoare* care se manifestă prin faptul că ochiul prezintă o sensibilitate diferită în funcție de lungimea de undă a radiației incidente.

Mărimile fotometrice legate de senzația luminoasă sunt:

- fluxul luminos, Φ , măsurat în lm (lumeni),
- intensitatea luminoasă a sursei, I, măsurată în cd (candela),
- iluminare unei suprafețe, E, măsurată în lx (lux).

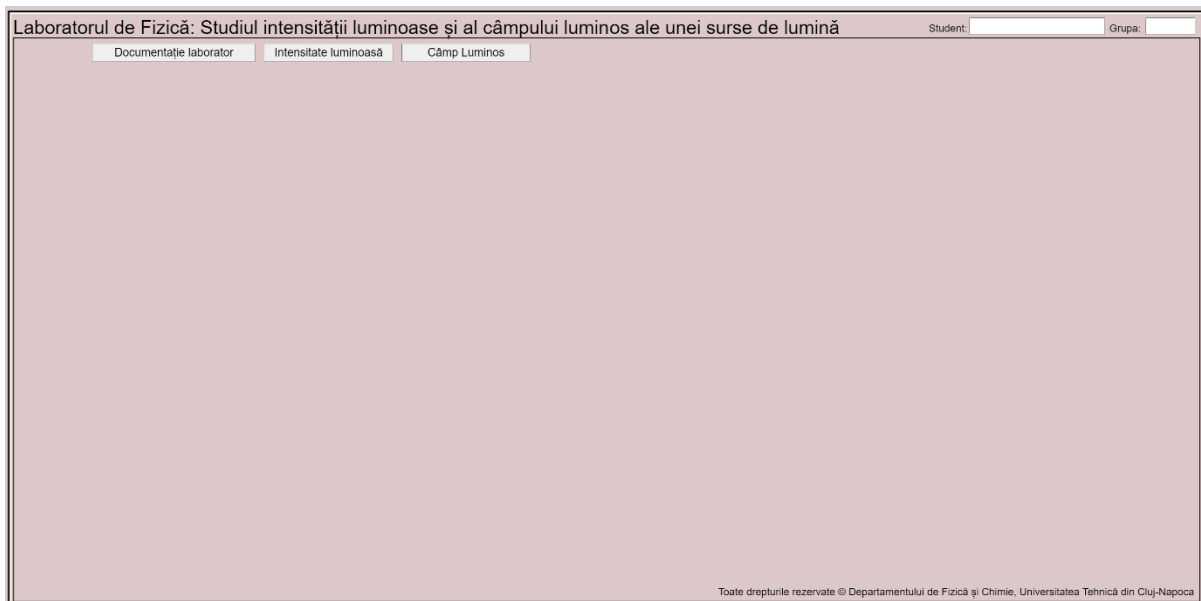
Lucrarea își propune determinarea intensității luminoase a unei surse necunoscute și studiul distribuției intensității luminoase în jurul unui bec cu filament. Din punct de vedere practic se consideră două surse de lumină între care se află un fotometru, cu ajutorul căruia se măsoară iluminarea a două suprafețe orientate perpendicular pe cele două surse de lumină. Cunoscând intensitatea luminoasă a uneia dintre surse și măsurând distanțele dintre surse și fotometru, pentru o iluminare egală a celor două suprafețe, se poate determina intensitatea luminoasă a sursei necunoscute.

Distribuția intensității luminoase în jurul unui bec cu filament depinde de direcția de observare, întrucât filamentul este un corp tridimensional asimetric necentrat în bec. Prin rotirea becului se modifică suprafața sub care se observa filamentul, și deci intensitatea sursei de lumina. Din punct de vedere practic se urmărește modificarea distanței becului cu filament față de un fotometru pentru a se obține o iluminare constantă a acestuia pentru unghiuri de observare cuprinse între 0° și 360° .

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Intensitatea%20Luminoasa/>

Prin accesarea link-ului se deschide pagina:



Dispozitivul experimental

1. Intensitatea luminoasă

Ca surse de lumină se folosesc două becuri electrice, unul ales ca etalon, de intensitate cunoscută (12 cd). Tensiunea electrică pe cel de-al doilea bec este variabilă, permițând astfel, studiul variației intensității luminoase în funcție de tensiunea aplicată. Între cele două surse de lumina pe un suport vertical care se poate glisa de-a lungul unui banc optic sunt doi senzori de lumina (câte unul orientat perpendicular pe fiecare sursă) conectați la un microcontroler de tip Arduino. Microcontrolerul are rolul de a măsura iluminarea provenită de la fiecare sursă și de a o afișa pe un ecran:

- E_0 de la sursa cu intensitate luminoasă cunoscută.
- E_x de la sursa cu intensitate luminoasă necunoscută.

De-a lungul bancului optic este montată o riglă, cu ajutorul căreia se va citi distanța dintre senzorii de lumină și surse atunci când se măsoară o iluminare egală (în limita erorilor experimentale) pentru ambele surse ($E_0 \cong E_x$).

2. Câmpul luminos

Studiul câmpului luminos al unei surse de lumină se face cu ajutorul unui senzor de lumină conectat la un microcontroler Arduino care indică în mod direct iluminarea. De-a lungul direcției pe care se fac măsurătorile se află o riglă pe care se citește distanța dintre becul cu filament și senzorul de lumină.

Instrucțiuni pentru simularea numerică:

1. Referatul de laborator al lucrării se accesează apăsând butonul Documentație laborator.
2. Pentru determinarea intensității luminoase a unei surse de lumină necunoscute prima dată se apasă butonul Intensitate luminoasă.

Pagina web se va popula ca în figura de mai jos:

Laboratorul de Fizică: Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină

Student: _____ Grupa: _____

Documentație laborator Intensitate luminoasă Câmp Luminos

2

$E_x = 0.08$ $E_0 = 0.53$

I_0 [Cd]	U [V]	R_x [cm]	R_0 [cm]	I_x [Cd]
12	4			
	6			
	8			
	10			
	12			
	14			
	16			
	18			
	20			

Grafic Online

Dacă avem o iluminare egală a două câmpuri $E_x = E_0$, atunci conform relației (10), se obține:

$$\frac{I_x}{I_0} = \frac{R_0^2}{R_x^2} \quad (11)$$

I_0 și I_x fiind intensitățile luminoase a două surse, R_0 și R_x distanțele de la surse până la fotometru. Cunosând intensitatea luminoasă I_0 a unei surse și măsurându-se R_0 și R_x , se poate determina intensitatea I_x a celei de-a doua surse:

$$I_x = I_0 \frac{R_0^2}{R_x^2} \quad (12)$$

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pentru înregistrarea datelor se urmează pașii:
 - i. Se aplică pe filamentul becului de intensitate necunoscută tensiunile din tabel scriindu-le direct pe voltmetru.
 - ii. Se deplasează perechea de senzori de lumină până în poziția în care iluminările celor două surse sunt egale: $E_x \cong E_0$.
 - iii. Se notează în tabel distanțele: dintre perechea de senzori și sursa cu intensitate luminoasă cunoscută, R_0 , și dintre senzorii de lumină și sursa cu intensitate luminoasă necunoscută, R_x .
 - iv. Se reiau măsurătorile pentru fiecare valoare a tensiunii dată în tabel.
 - v. Pentru determinarea intensității luminoase a sursei de lumină, se folosește formula:

$$I_x = I_0 \frac{R_0^2}{R_x^2},$$

unde: intensitatea sursei cunoscute $I_0 = 12 \text{ cd}$.

- vi. După determinarea valorilor intensității luminoase a sursei necunoscute, în fiecare caz, se apasă butonul Grafic Online : și se reprezintă grafic $I_x = f(U)$ pe hârtie milimetrică (Excel, Origin, etc.).

4. Pentru determinarea câmpului luminos al unei surse de lumină se apasă butonul

Câmp Luminos

Laboratorul de Fizică: Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină

Documentație laborator Intensitate luminoasă Câmp Luminos

Student: Grupa:

Crește 30 grade
Unghi = 0
Scade 30 grade

Grafic 1 Online

Unghi [grade]	0	30	60	90	120	150	180	210	240	270	300	330	360
Distanța [cm]													

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

5. Pașii de urmat pentru înregistrarea datelor constau în:

- Se alege o valoare a iluminării și se notează distanța la care se află senzorul de lumina față de sursă.
- Se modifică unghiul și se păstrează în fiecare caz aceeași iluminare, prin modificarea corespunzătoare a distanței de la bec la senzor care se trece în tabel.
- După înregistrarea datelor se apasă butonul: [Grafic 1 Online](#) și se reprezintă grafic $Distanța = f(unghi)$, în coordonate polare.

Vă urăm succes!

V.3 Studiul efectului fotoelectric și determinarea constantei lui Planck

Introducere

Efectul fotoelectric constă în emisia de electroni de la suprafața unui corp (metalic sau oxidic) sub acțiunea luminii. Albert Einstein folosește o ipoteză prin care admite că lumina este emisă în mod discontinuu sub formă de cuante de energie numite fotoni. Legătura cantitativă dintre energia cinetică a fotoelectronilor și frecvența luminii incidente este dată de ecuația lui Einstein:

$$h\nu = L_e + \frac{mv^2}{2},$$

unde: $h\nu$ este energia fotonilor, luminii incidente; L_e este lucrul mecanic de ieșire (extracție) al electronului prin bariera de potențial de la suprafața metalului; $\frac{mv^2}{2}$ este energia cinetică a fotoelectronilor emiși.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Planck/>

Prin accesarea link-ului se deschide pagina:

Laboratorul de Fizica: Studiul efectului fotoelectric și determinarea constantei lui Planck.

Student:

Grupa:

Documentație Laborator

Experiment

$\lambda_1=6250 \text{ \AA}$

$\lambda_2=6896 \text{ \AA}$

Grafic Online

	U[V]	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	S ₁	S ₂	U _{f1}	U _{f2}	h	Δh/h
λ ₁	I [div]								25						[10 ⁻⁸ A/V]	[10 ⁻⁸ A/V]	[V]	[V]	[10 ⁻³⁴ Js]	[%]
λ ₂	U[V]	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	I [div]					
									15											

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca.

Dispozitivul experimental

În fotografia din partea dreaptă a lucrării este prezentată instalația experimentală din laborator. Celula fotoelectrică formată dintr-un catod fotosensibil de Cesium și un anod este alimentată prin intermediul unei rezistențe variabile (un potențiomtru) de la o sursă de curent continuu. Diferența de potențial între electrozi se măsoară cu un voltmetru. O diafragmă reglabilă permite stabilirea unui fluxul luminos variabil care cade pe catodul fotocelulei. Pentru

a avea un fascicul incident monocromatic, lumina provenită de la o lampă de cuarț trece printr-un filtru de lumină (galben sau roșu). Pentru măsurarea curentului fotoelectric se folosește un galvanometru cu spot luminos de mare sensibilitate ($1 \text{ diviziune} = 9 \cdot 10^{-9} \text{ A}$).

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se deschide apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul , iar apoi se apasă butonul .

Laboratorul de Fizica: Studiul efectului fotoelectric si determinarea constantei lui Planck.

Student: _____
 Grupa: _____

$\lambda_1 = 6250 \text{ \AA}$
 $\lambda_2 = 6896 \text{ \AA}$

Reglează Fanta

Catod Anod

0.0 V

0.0 div

1 div = $9 \cdot 10^{-9} \text{ A}$

R

Reglează Tensiunea

	U [V]	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	S_1	S_2	U_{f1}	U_{f2}	h	$\Delta h/h$
λ_1	I [div]								25											
λ_2	U [V]	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5						
	I [div]								15											

Grafic Online

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pașii pentru înregistrarea datelor constau în:

- i. Se mută cursorul în poziția în care galvanometrul indică aproximativ 25 diviziuni.
- ii. Se modifică tensiunea aplicată între catod și anod prin intermediul potențiometrului, de la cursorul notându-se valorile intensității (în diviziuni) pentru fiecare din tensiunile (pozitive, respectiv negative) din tabel.
- iii. Se schimbă filtrul apăsând butonul .
- iv. Când voltmetrul indică se mută cursorul până în poziția în care galvanometrul indică aproximativ 15 diviziuni.
- v. Se repetă operația descrisă la punctul **ii**.
- vi. Se reprezintă pe hârtie milimetrică (sau în Excel) caracteristica curent-tensiune ($I = f(U)$) pentru ambele filtre, pe același sistem de axe.
- vii. Se trasează panta porțiunii celei mai abrupte a graficului $I = f(U)$, pentru a determina sensibilitățile celor două filtre (S_1 și S_2), respectiv tensiunile de frânare

(U_{f1} și U_{f2}). Sensibilitățile celor filtrelor se află, determinând panta celor două grafice și ținând cont că pe galvanometru 1 *diviziune* = $9 \cdot 10^{-9}$ A. Tensiunile de frânare reprezintă tensiunile pentru care intensitatea curentului este zero.

viii. Pentru a determina constanta lui Planck, se folosește formula:

$$h = \frac{e \cdot \lambda_1 \lambda_2}{c \cdot \lambda_2 - \lambda_1} (U_{f1} - U_{f2}).$$

Se dau:

- Sarcina electronului: $e = 1.6 \cdot 10^{-19}$ C;
- Viteza luminii: $c = 3 \cdot 10^8$ m/s;
- Lungimea de undă pentru filtrul: galben: $\lambda_1 = 6250 \text{ \AA} = 6250 \cdot 10^{-10}$ m, roșu: $\lambda_2 = 6896 \text{ \AA} = 6896 \cdot 10^{-10}$ m.

Vă urăm succes!

V.4 Studiul polarizării luminii

Introducere

Lumina emisă de diferite surse (cum ar fi Soarele sau un bec) este total nepolarizată. Acest lucru înseamnă că cele două componente magnetice și electrice, oscilează în toate direcțiile perpendiculare pe direcția de propagare. Când direcția pe care oscilează componenta electrică este unică, vorbim de o undă electromagnetică total polarizată. Dacă frecvența acestei oscilații se află în spectrul vizibil, atunci vorbim de lumină polarizată.

Lumina vizibilă este formată din unde electromagnetice cu lungimea de undă cuprinsă între 3800 Å și 7600 Å ($1 \text{ Å} = 10^{-10} \text{ m}$). Efectul de *lumină* este creat de acțiunea componentei electrice a câmpului electromagnetic asupra anumitor substanțe aflate în celulele (conuri și bastonașe) de pe retina ochiului. Polarizarea luminii este fenomenul care apare atunci când unda electromagnetică care constituie lumina vizibilă oscilează într-o direcție preferențială.

Simularea lucrării de laborator se găsește la adresa:

https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Polarizarea_luminii/

Prin accesarea link-ului se deschide pagina:

The screenshot shows a virtual laboratory interface titled 'Laboratorul de Fizica: Polarizarea luminii'. It features a student and group input field, a 'Documentatie Laborator' and 'Masor' button, a schematic diagram of the experimental setup, and a data table.

θ [°]	0	30	60	90	120	150	180	210	240	270	300	330	360
I_e (μA)													
I_t (μA)													

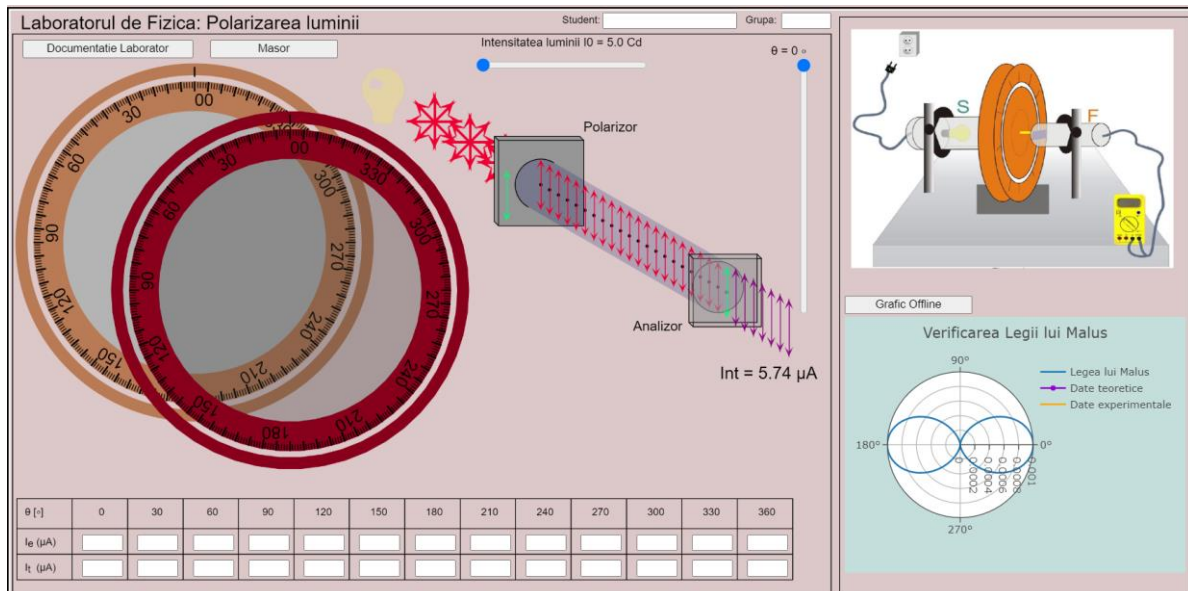
Dispozitivul experimental

Pentru a studia polarizarea luminii, în laborator, este folosit dispozitivul experimental prezentat schematic în imaginea de mai sus. În lucrarea de față, lumina naturală provenind de la un bec este trecută printr-un prim polarizor, apoi trece prin al doilea polarizor (ca dispozitiv) cu rol de analizor. Trecând prin polarizor, lumina devine total polarizată. Unghiul θ este

unghiul dintre direcțiile de polarizare ale celor doi polarizori. Acesta poate fi măsurat datorită goniometrului pe care sunt montați analizorul și polarizorul, și care permite rotirea analizorului. Intensitatea razei reflectate este măsurată cu ajutorul unei fotocelule F, care funcționează pe baza efectului fotoelectric.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .



3. Pașii pentru înregistrarea datelor constau în:

- i. Cu ajutorul cursorului valoarea maximă pentru intensitatea fasciculului care intră în analizor.
- ii. Cu ajutorul cursorului se rotește analizorul cu câte 30°, de la 0° la 360°, și se măsoară intensitatea experimentală a curentului electric $I_e(\mu A)$. Valorile măsurate se scriu în tabel.
- iii. Cu datele obținute se va trasa variația $I_e = f(\theta)$ în coordonate polare.
- iv. Valoarea maximă obținută a curentului notată cu I_{max} , se va utiliza pentru calculul valorilor teoretice, conform legii lui Malus:
$$I_t = I_{max} \cos^2 \theta,$$
 pentru fiecare unghi θ considerat experimental.
- v. Se reprezintă alături de curba experimentală și curba teoretică $I_t = f(\theta)$.
- vi. Toate rezultatele obținute se trec în tabel.

Vă urăm succes!

VI Fizica solidului

VI.1 Studiul efectului termoelectric

Introducere

Efectul termoelectric se explică prin fenomenele fizice care apar la contactul metal-metal. Astfel, punând în contact două metale diferite, între ele apare o diferență de potențial de contact. Aceasta se datorează tendinței de egalare a potențialelor chimice ale celor două metale. Valoarea tensiunii depinde de natura metalelor și are în general valori mici (milivolți – volți). Aceasta nu depinde de forma și de dimensiunile metalelor aflate în contact, dar depinde de puritatea acestora.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/EfectulTermoelectric/>

Prin accesarea link-ului se deschide pagina:

The screenshot shows a web-based virtual laboratory interface. At the top, it says "Laboratorul de Fizica: Studiul Efectului Termoelectric" with fields for "Student" and "Grupa". Below this are two buttons: "Documentatie Laborator" and "Măsur". The main area is a large empty space. At the bottom left, there is a table with two rows: "t [C]" and "E [mV]". The "t [C]" row contains values from 25.0 to 95.0 in increments of 5.0. The "E [mV]" row contains empty input boxes. At the bottom right, there is a photograph of the experimental setup, showing a thermocouple connected to a voltmeter, with one junction in a beaker of water on a hot plate and the other in a Dewar flask. Below the photo is a button labeled "Grafic Online". At the very bottom, there is a small copyright notice: "Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca".

t [C]	25.0	30.0	35.0	40.0	45.0	50.0	55.0	60.0	65.0	70.0	75.0	80.0	85.0	90.0	95.0
E [mV]															

Dispozitivul experimental

Dispozitivul experimental folosit este prezentat în fotografia din dreapta sus în pagina web. Metalele utilizate pentru studiul efectului termoelectric sunt Cupru și Constantan. Unul dintre contactele celor două metale (sudura A) este menținut într-un vas cu apă pus pe o plită, iar celălalt contact (sudura B) este menținut într-un vas Dewar în gheață ($T_B \cong 0^\circ\text{C}$). Diferența de temperatură dintre cele două contacte este dată de temperatura contactului A (T_A), care crește treptat.

Instrucțiuni pentru simularea numerică:

1. Referatul de laborator al lucrării se accesează apăsând butonul .

2. Pentru a efectua experimentul virtual se apasă butonul .

The screenshot shows a software interface for a thermoelectric experiment. The main window displays a schematic of a thermocouple circuit. One junction is immersed in a hot reservoir at $T_A = 25.0^\circ\text{C}$, while the other is in a cold reservoir at T_B . A voltmeter in the circuit shows a reading of 1.00 mV . Below the schematic is a data table with two rows: $t [^\circ\text{C}]$ and $E [\text{mV}]$. The t row contains values from 25.0 to 95.0 in increments of 5.0. The E row contains empty input boxes. To the right of the simulation is a photograph of the physical experimental setup, including a thermocouple, a voltmeter, and a glass of water. A 'Grafic Online' button is located below the photograph.

$t [^\circ\text{C}]$	25.0	30.0	35.0	40.0	45.0	50.0	55.0	60.0	65.0	70.0	75.0	80.0	85.0	90.0	95.0
$E [\text{mV}]$															

3. Pașii pentru înregistrarea datelor constau în:

- i. Se notează în tabel valoarea tensiunii electromotoare corespunzătoare temperaturii de 25°C .
- ii. Pentru creșterea temperaturii T_A , se apasă butonul .
- iii. Se notează în tabel valorile corespunzătoare pentru tensiunea electromotoare din 5°C în 5°C .
- iv. Orientativ, reprezentarea grafică se poate observa online apăsând butonul .
- v. Se reprezintă grafic $E = f(t)$.
- vi. Se aproximează datele experimentale cu o funcție liniară.
- vii. Se calculează panta dreptei obținute. Aceasta reprezintă coeficientul Seebeck al termocuplului utilizat.

Vă urăm succes!

VI.2 Determinarea energiei de activare a unui semiconductor

Introducere

Semiconductorii sunt considerați ca niște conductori electrici cu valori ale conductibilității electrice cuprinse între 10^{-12} - 10^6 ($\Omega \cdot m$)⁻¹, valori mai mici decât la metale, dar mai mari decât la izolatori. Prin aplicarea teoriei cuantice la mișcarea electronilor din rețeaua cristalină s-a stabilit că energiile pe care le poate lua un electron într-un cristal sunt grupate în zone (benzi) permise separate de zone (benzi) interzise. Modul de distribuire al electronilor pe nivelele energetice aflate în interiorul unei benzi permise este determinat de starea termică în care se află cristalul. La temperatura de 0 K electronii vor fi repartizați pe cele mai joase nivele de energie, astfel încât benzile vor fi umplute integral începând cu cele inferioare până la o anumită bandă care poate fi ocupată parțial sau total. Ultimul nivel energetic populat cu electroni la 0 K se numește nivel Fermi.

Energia de activare ΔE a unui semiconductor se definește ca fiind cantitatea minimă de energie necesară pentru trecerea unui electron din banda de valență în banda de conducție.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/EnergiaActivare/>

Prin accesarea link-ului se deschide pagina:



The screenshot shows a web-based laboratory simulation interface. At the top, it reads 'Laboratorul de Fizica: Determinarea energiei de activare a unui semiconductor'. Below this, there are two buttons: 'Documentatie Laborator' and 'Măsur'. The main area contains a data table with the following columns: 't [°C]', 'T [K]', '10³/T [K⁻¹]', 'R [Ω]', 'lg(R) [-]', and 'ΔE [eV]'. The table has 18 rows, with the first row containing the value 22.0 and the rest being empty input fields. To the right of the table is a photograph of the experimental setup, showing a semiconductor sample in a beaker of water on a magnetic stirrer, with a multimeter connected to it. Below the photo is a 'Grafic Online' button. At the bottom of the interface, there is a small copyright notice: 'Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca'.

t [°C]	T [K]	10 ³ /T [K ⁻¹]	R [Ω]	lg(R) [-]	ΔE [eV]
22.0					
25.0					
30.0					
35.0					
40.0					
45.0					
50.0					
55.0					
60.0					
65.0					
70.0					
75.0					
80.0					
85.0					
90.0					
95.0					

Dispozitivul experimental

Pentru determinarea energiei de activare a unui semiconductor, în laborator, poate fi folosit dispozitivul experimental prezentat în poza din dreapta sus. Semiconductorul este introdus într-un vas cu apă. Acesta este așezat pe o plită electrică. Temperatura se citește cu

ajutorul unui senzor de temperatură introdus în vasul cu apă. Semiconductorul este legat la un multimetru cu ajutorul căruia se determină rezistența la diferite temperaturi.

O altă metodă pentru a determina variația rezistenței cu temperatura unui semiconductor este introdus într-un balon de sticlă. În interiorul balonului există un filament prin care va trece curent electric de o anumită intensitate, astfel încât să simuleze un mic cuptor. Temperatura din interiorul lui fiind citită cu ajutorul unui termocuplu. Semiconductorul este legat la un multimetru sau ohmmetru pentru a-i măsura rezistența la temperaturi diferite.

Instrucțiuni pentru simularea numerică

1. Referatul de laborator al lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

t [°C]	T [K]	$10^3/T$ [K ⁻¹]	R [kΩ]	lg(R) [-]	ΔE [eV]
22.0					
25.0					
30.0					
35.0					
40.0					
45.0					
50.0					
55.0					
60.0					
65.0					
70.0					
75.0					
80.0					
85.0					
90.0					
95.0					

Interfața de simulare prezintă un titlu "Laboratorul de Fizică: Determinarea energiei de activare a unui semiconductor" și un buton "Măsur" în partea de sus. În centru este o imagine 3D a aparaturii experimentale: un balon de sticlă pe un suport, un termocuplu și un rezistor conectat la un multimetru. În partea de jos stânga este un tabel cu coloane pentru temperatură (t [°C]), temperatură în Kelvin (T [K]), inversul temperaturii în Kelvin (10³/T [K⁻¹]), rezistență (R [kΩ]), logaritmul rezistenței (lg(R) [-]) și energia de activare (ΔE [eV]). În partea de jos dreapta este un buton "Grafic Online".

3. Pașii pentru înregistrarea și prelucrarea datelor constau în:
 - i. Se notează prima valoare a rezistenței semiconductorului la temperatura camerei în tabelul de valori.
 - ii. Se apasă butonul .
 - iii. Se așteaptă creșterea temperaturii în balonul de sticlă și se notează valorile rezistențelor semiconductorului pentru toate temperaturile indicate în tabelul de valori.
 - iv. Se completează coloana corespunzătoare temperaturii exprimate în K.
 - v. Se calculează $\frac{10^3}{T}$ [K⁻¹] și lgR.
 - vi. Se reprezintă grafic $\lg R = f\left(\frac{10^3}{T}\right)$. Orientativ, reprezentarea grafică se poate realiza și online apăsând butonul .
 - vii. Se aproximează datele experimentale cu o funcție liniară.

viii. Se determină panta dreptei obținute ($tg\theta$). Înmulțind panta dreptei cu factorul 0.2 se obține energia de activare pentru semiconductorul considerat.

Vă urăm succes!

VI.3 Determinarea conductibilității termice printr-o metodă staționară

Introducere

Dacă două corpuri A și B, cu temperaturi diferite, se găsesc în contact cu un același corp conductor C, atunci căldura se propagă de la corpul mai cald spre corpul mai rece, prin corpul C (Fig. 1). Schimbul de căldură se face prin conducție termică de-a lungul corpului C, de la particulă la particulă. Cantitatea de căldură, Q transmisă în unitatea de timp printr-o secțiune S este direct proporțională cu diferența de temperatură ΔT dintre corpurile A și B, cu secțiunea S a corpului C și invers proporțională cu lungimea L a corpului C.

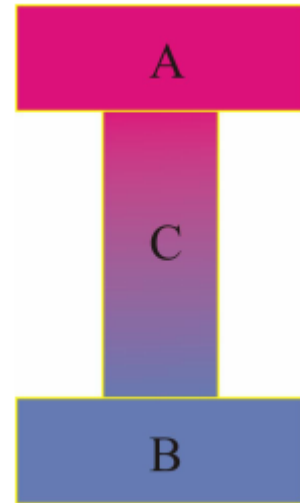


Figura 1

$$Q = \lambda S \frac{\Delta T}{L}$$

Factorul de proporționalitate, λ este coeficientul de conductibilitate termică al corpului C: $\lambda = \frac{Q}{S \frac{\Delta T}{L}} = \frac{Q}{S \Delta T} L$ fiind numeric egal cu cantitatea de căldură transportată în unitatea de timp prin unitatea de secțiune, datorită unui gradient de temperatură unitar.

Coeficientul de conductivitate termică, λ este o mărime specifică fiecărui material și depinde de proprietățile fizice ale acestuia, cum sunt: temperatura, porozitatea, densitatea, umiditatea, structura materialului etc.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/ConductibilitateaTermica/>

Prin accesarea link-ului se deschide pagina:

Documentatie Laborator

$$\lambda = \frac{Q_1 + Q_2 + Q_3 - Q_4}{S \cdot \text{grad}(t)}$$

$$Q_1 = \frac{m_1 c_1 (T_1 - T_2)}{t} \quad (W) \quad m_1 = 0,805 \text{ kg}, c_1 = 894,52 \frac{J}{\text{kg} \cdot \text{grad}}$$

$$Q_2 = \frac{(T_1 - T_2) m_2 c_2}{t} \quad (W) \quad c_2 = 4185 \frac{J}{\text{kg} \cdot \text{grad}}, c_3 = 383,3 \frac{J}{\text{kg} \cdot \text{grad}}$$

$$\text{grad}(t) = \frac{\sum T_2 - \sum T_1}{t}$$

nr.	t [min]	T ₁ [°C]	T ₂ [°C]	T [°C]	θ [°C]	λ [W/m grad]
1	0					
2	2					
3	4				22	
4	6					
5	8					
6	10					

Q₂ = f((ΣT₁ - Σθ) / n)

Q₁ = f((Σθ - ΣT) / n)

Dispozitivul experimental

Pentru determinarea coeficientului de conductibilitate termică, în laborator, este folosit dispozitivul experimental format din: i) corpul A – cald, rolul lui este jucat de vaporii de apă produși prin încălzirea apei dintr-un balon de sticlă, de la un reșou; ii) corpul care preia căldura (corpul B) este apa dintr-un calorimetru; iii) corpul prin care se transmite căldura (corpul C) este un cilindru plin din metal, cu un capac la partea superioară la care se leagă un furtun de conducere a vaporilor de apă.

Instrucțiuni pentru simularea numerică:

1. Referatul de laborator al lucrării se accesează apăsând butonul .
2. Pentru a efectua experimentul virtual se apasă butonul .

Laboratorul de Fizica: Determinarea conductibilitatii termice printr-o metoda stationara

Student: _____ Grupa: _____

Documentatie Lab Masor GraficO

$$\lambda = \frac{Q_1 + Q_2 + Q_3 - Q_4}{S \cdot \text{grad}(t)}$$

$$Q_1 = \frac{m_1 c_1 (T_1 - T_2)}{t} \quad (W) \quad m_1 = 0,805 \text{ kg}, c_1 = 894,52 \frac{J}{\text{kg grad}}$$

$$Q_3 = \frac{(T_2 - T) m_2 c_2}{t} \quad (W), c_2 = 4185 \frac{J}{\text{kg grad}}, c_p = 3883,3 \frac{J}{\text{kg grad}}$$

$$\text{grad}(t) = \frac{\sum T_2 - \sum T_1}{t}$$

Start Stop

Scilindru= 12,56*10^-4 m^2
L= 0,12 m

T₂ = 63,0 °C
T₁ = 43,0 °C
T = 23,0 °C

masa vas= 137g
masa apa= 621g

Q₂ (cal/s) Q₄ (cal/s)

Q₂ = f $\left(\frac{\sum T_1 - \sum \theta}{n} \right)$
Q₄ = f $\left(\frac{\sum \theta - \sum T}{n} \right)$

nr.	t [min]	T ₁ [°C]	T ₂ [°C]	T [°C]	θ [°C]	A [W/(m grad)]
1	0					
2	2					
3	4					22
4	6					
5	8					
6	10					


3. Se notează temperaturile T_1 , T_2 , T , θ (temperatura camerei) la momentul inițial de timp ($n_1 = 0$).
4. Pașii pentru înregistrarea și prelucrarea datelor constau în:
 - i. Se apasă butonul , iar apa din balonul de sticlă va începe să fiarbă, ducând la creșterea temperaturilor.
 - ii. Concomitent cu apăsarea butonului de se pornește și un cronometru.
 - iii. Valorile celor trei temperaturi T- temperatura apei din calorimetru, T_1 - temperatura din partea de jos a cilindrului metalic, T_2 - temperatura din partea de sus a cilindrului metalic se citesc din 2 în 2 minute și se trec în tabel.
 - iv. După completarea tabelului cu valorile experimentale ale temperaturilor se calculează cantitatea de căldură (Q) care trece în unitatea de timp prin secțiunea de la mijlocul probei, care se compune din:

$$Q = Q_1 + Q_2 + Q_3 - Q_4,$$

unde:

- Q_1 este cantitatea de căldură transportată într-o secundă pentru încălzirea jumătății de jos a cilindrului, dată de relația:

$$Q_1 = \frac{m_c \cdot c_c}{2} \cdot \frac{T_1^{(\tau)} - T_1^{(0)}}{\tau}.$$

- Q_2 este cantitatea de căldură cedată într-o secundă mediului înconjurător de către jumătatea de jos a cilindrului. Pentru determinarea ei s-au făcut măsurători suplimentare ale căror rezultate sunt date în diagrama anexă (care poate fi accesată apăsând butonul ). Căldura cedată este proporțională cu diferența de temperatură dintre cilindru și mediu. Pentru a determina căldura Q_2 se calculează temperatura medie a jumătății de jos a cilindrului $\frac{\sum T_1}{n}$ și a mediului înconjurător $\frac{\sum \theta}{n}$. De pe diagrama $Q_2 = f\left(\frac{\sum T_1}{n} - \frac{\sum \theta}{n}\right)$ se citește Q_2 corespunzător diferenței medii de temperatură.

- Q_3 este căldura absorbită de calorimetru și apa din el în unitatea de timp:

$$Q_3 = \frac{(T^{(\tau)} - T^{(0)}) \cdot (m_v c_v + m_a c_a)}{\tau}.$$

- Q_4 este căldura absorbită de calorimetru din mediul înconjurător, iar pentru determinarea ei se folosește o diagramă asemănătoare cu aceea pentru Q_2 . Q_4 se determină în funcție de diferențele corespunzătoare.

$$Q_4 = f\left(\frac{\sum \theta}{n} - \frac{\sum T}{n}\right).$$

- ∇T este gradientul de temperatură și se determină cu ajutorul relației:

$$\nabla T = \frac{\frac{\sum T_2 - \sum T_1}{n}}{L},$$

unde $L = 0.12$ m este distanța dintre cele două termometre T_1 și T_2 .

- v. În final se calculează coeficientul de conductibilitate termică utilizând relația:

$$\lambda = \frac{Q_1 + Q_2 + Q_3 - Q_4}{S \cdot \nabla T},$$

unde $S = 12.56 \cdot 10^{-4} \text{ m}^2$ pentru cilindrul utilizat. Valoare obținută pentru conductibilitate se trece în tabelul de valori.

Vă urăm succes!

VI.4 Verificarea experimentală a legii lui Ștefan-Boltzmann

Introducere

Toate corpurile emit și absorb energie din/sau mediul înconjurător sub formă de radiație electromagnetică datorită oscilațiilor sarcinilor elementare care se găsesc în toate corpurile aflate la o temperatură mai mare de zero Kelvin. Această radiație electromagnetică depinde de natura corpurilor astfel, corpurile solide au un spectru continuu și se observă toate frecvențele. O astfel de radiație se numește radiație termică, are un spectru continuu, este nepolarizată, iar spectrul acesteia depinde numai de temperatura la care se găsește corpul radiant.

Scopul lucrării este acela de a verifica legea lui Ștefan-Boltzmann pentru energia radiantă de un corp negru. Sursa radiantă este un bec cu incandescență. Se va măsura intensitatea și tensiunea electrică, determinându-se apoi puterea și temperatura. Se va evalua grafic dacă se respectă legea lui Ștefan-Boltzmann determinându-se panta dreptei $\ln P = f(\ln T)$, a cărei valoare trebuie să fie aproximativ 4.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Stefann-Boltzmann/>

Prin accesarea link-ului se deschide pagina:

The screenshot shows a virtual laboratory interface titled "Laborator de Fizica - Verificarea Experimentală a legii lui Ștefan-Boltzmann". It features a data table, a graph, and two photographs of the experimental setup.

U [V]	I [A]	P [W]	ln P	R [Ω]	T [K]	ln T
6						
8						
10						
12						
14						
16						
18						
20						

The graph displays the spectral radiance $\rho(\lambda, T)$ in $\text{W}/(\text{m}^2 \cdot \text{nm})$ versus wavelength λ in nm for five different temperatures: 5500K, 5000K, 4500K, 4000K, and 3500K. The curves show a peak that shifts to shorter wavelengths and increases in height as the temperature increases.

Two photographs on the right show the experimental setup: the top one shows a digital multimeter (GEM) connected to a power source, and the bottom one shows a light bulb connected to a power source.

Dispozitivul experimental

Pentru verificarea legii lui Ștefan-Boltzmann se folosește un bec de 24 V și 0.1 A, o sursă de alimentare de 24 V, un reostat, un ampermetru și un voltmetru legate ca în montajul din poza din pagina web din partea dreapta jos.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării poate fi accesată apăsând butonul: .

2. Pentru a efectua experimentul virtual se apasă butonul: .

Laborator de Fizica - Verificarea Experimentală a legii lui Stefan-Boltzmann

Document Masuratori Student: Grupa:

U [V]	I [A]	P [W]	ln P	R [Ω]	T [K]	ln T
6						
8						
10						
12						
14						
16						
18						
20						

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pașii pentru înregistrarea și prelucrarea datelor constau în:

i. Se modifică tensiunea electrică pe bec între limitele 6 V și 20 V, ajustând valoarea citită de pe voltmetru cu ajutorul cursorului localizat sub acesta. Pentru fiecare valoare a tensiunii electrice se măsoară valorile corespunzătoare intensității curentului electric indicate pe ampermetru .

ii. Se trec valorile în tabel.

iii. Se determină puterea consumată folosind relația:

$$P = U \cdot I .$$

iv. Se calculează logaritmul natural al puterii și se trece în tabel.

v. Se determină rezistența filamentului cu relația:

$$R = \frac{U}{I} .$$

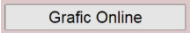
vi. Se calculează temperatura filamentului din relația de mai jos:

$$R = R_0 \cdot (1 + \alpha \cdot t) \Rightarrow T = T_0 + \frac{R - R_0}{\alpha \cdot R_0} ,$$

Se dau:

- $\alpha = 0.0053 \text{ } \text{grad}^{-1}$.
- $T_0 = 300 \text{ K}$.

- $R_0 = 0,9 \Omega$.

- vii. Se calculează logaritmul natural al temperaturii și se trece în tabel.
- viii. Se reprezintă grafic $\ln P = f(\ln T)$ pe foaia milimetrică sau într-un program de tip Excel. Orientativ, reprezentarea grafică se poate realiza și online apăsând butonul  .
- ix. Se aproximează datele experimentale cu o funcție liniară.
- x. Se aleg două puncte (cât mai îndepărtate dar în domeniu de reprezentare) de pe funcție liniară (nu din tabel) și se calculează panta graficului.
- xi. Rezultatul se compară cu valoarea teoretică 4, verificând-se legea lui Ștefan-Boltzmann.

Vă urăm succes!

VI.5 Determinarea temperaturii Curie a materialelor feromagnetice

Introducere

Atunci când este introdusă într-un câmp magnetic orice substanță devine activă magnetic, adică ea este atrasă sau respinsă de acesta. Substanțele care sunt atrase pot fi *paramagnetice*, *feromagnetice*, *ferimagnetice* sau *antiferomagnetice*, iar substanțele care sunt respinse se numesc *diamagnetice*.

La materialele magnetice se întâlnesc două tipuri de magnetizări:

- magnetizație permanentă: materialul este magnetizat intrinsec;
- magnetizație temporară: materialul dobândește proprietăți magnetice sub acțiunea unui câmp magnetic extern.

Există două tipuri de materiale magnetice:

- materiale magnetice liniare: la care apare noțiunea de *susceptibilitate magnetică*. Aici întâlnim materialele diamagnetice, paramagnetice.
- materiale magnetice neliniare: la care apare noțiunea de *curbă de histerezis*. În această categorie întâlnim materialele ferimagnetice, feromagnetice și antiferomagnetice.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Determinarea%20temperaturii%20Curie/>

Prin accesarea link-ului se deschide pagina:

Table 1: Data for Ferita

T(°C)	200	250	300	320	330	340	350	360	370	380	400	420	450
I(μA)													

Table 2: Data for Nichelita

T(°C)	320	340	360	380	400	410	420	425	430	435	450	470	500
I(μA)													

Toate drepturile rezervate © Departamentul de Fizică și Chimie, UTCN

Dispozitivul experimental

Pentru determinarea temperaturii Curie, în laborator, este folosit dispozitivul experimental prezentat în imaginea de mai sus. Cele două probe folosite (corpuri de studiat) din Fe și Ni, sunt așezate pe rând în cuptorul bobinat cu fir de kantal care este alimentat de la un autotransformator cu curent alternativ. Creșterea treptată a temperaturii duce la scăderea permeabilității magnetice, astfel scade valoarea intensității curentului indus.

Instrucțiuni pentru simularea numerică:

1. Partea teoretică a lucrării se deschide apăsând butonul .
2. Pentru a efectua experimentul virtual în care studiem proba de nichel se apasă butonul.

Laboratorul de fizica: Temperatura Curie

Student: _____ Grupa: _____

Documentatie Laborator Masor Ni Masor Fe

Start Stop

$I = 8.49 \mu A$ $T = 320.0 \text{ } ^\circ C$

Nichelita

Ferita	
T(°C)	200 250 300 320 330 340 350 360 370 380 400 420 450
I(μA)	

Nichelita	
T(°C)	320 340 360 380 400 410 420 425 430 435 450 470 500
I(μA)	

Grafic Offline

Temp. Curie Nichel

I (μA)

t (°C)

Toate drepturile rezervate © Departamentul de Fizică și Chimie, UTCN

3. Pașii pentru înregistrarea datelor constau în:

- i. Se apasă butonul .
- ii. Se mută cursorul spre dreapta dacă se dorește ca viteza cu care se citesc măsurătorile să fie mai mare.
- iii. Pentru fiecare valoare a temperaturii T [°C] din tabel se citește valoarea curentului indus I [μA].
- iv. Se apasă butonul pentru a opri măsurătorile. Valorile obținute se trec în tabel.
- v. Se reprezintă grafic intensitatea curentului indus în funcție de temperatură. Din grafic se determină temperatura Curie ca fiind temperatura la care intensitatea curentului scade brusc, punctul de inflexiune al curbei.

4. Pentru a efectua experimentul virtual în care studiem proba de fier se apasă butonul

Ferita													
T(°C)	200	250	300	320	330	340	350	360	370	380	400	420	450
I(μA)													

Nichelita													
T(°C)	320	340	360	380	400	410	420	425	430	435	450	470	500
I(μA)													

Toate drepturile rezervate © Departamentul de Fizică și Chimie, UTCN

5. Pașii pentru înregistrarea datelor constau în:
- i. Se apasă butonul .
 - ii. Se mută cursorul spre dreapta dacă se dorește ca viteza cu care se citesc măsurătorile să fie mai mare.
 - iii. Pentru fiecare valoare a temperaturii T [°C] din tabel se citește valoarea curentului indus I [μA].
 - iv. Se apasă butonul pentru a opri măsurătorile. Valorile obținute se trec în tabel.
 - v. Se reprezintă grafic intensitatea curentului indus în funcție de temperatură. Din grafic se determină temperatura Curie ca fiind temperatura la care intensitatea curentului scade brusc, punctul de inflexiune al curbei.
 - vi. Reprezentarea grafică pentru ambele probe se face pe același grafic.

Vă urăm succes!

VI.6 Studiul comportării materialelor feromagnetice

Introducere

În prezența câmpurilor magnetice exterioare corpurile se magnetizează. Inducția magnetică B reprezintă măsura câmpului magnetic în orice mediu. În cazul substanțelor feromagnetice și paramagnetice *inducția magnetică scade* odată cu *creșterea temperaturii*, depinzând de aceasta.

Momentele magnetice ale materialelor feromagnetice sub influența unui câmp magnetic exterior se vor orienta în același sens, asta înseamnă că *permeabilitatea magnetică* va avea valori mari. La reprezentarea inducției magnetice în funcție de intensitatea câmpului magnetic nu se va obține o dependență liniară ci o curbă de histerezis. Dintre materialele cu comportare feromagnetică avem: fierul, cobaltul, nichelul, gadoliniul și aliajele lor.

Simularea lucrării de laborator se găsește la adresa:

<https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Studiul%20materialelor%20feromagnetice/>

Prin accesarea link-ului se deschide pagina:

I (A)	-5.0	-4.0	-3.0	-2.0	-1.0	-0.5	0.0	0.5	1.0	2.0	3.0	4.0	5.0	4.0	3.0	2.0	1.0	0.5	0.0	-0.5	-1.0	-2.0	-3.0	-4.0	-5.0	
Φ (mWb)																										
H (A/m)																										
B (T)																										

Dispozitivul experimental

Pentru studierea comportării materialelor feromagnetice, în laborator, este folosit dispozitivul experimental reprezentat schematic în imaginea de mai jos. Proba feromagnetică de forma unui tor este confecționată din oțel. La trecerea curentului prin spirele solenoidului bobinat pe probă, aceasta se magnetizează.

Instrucțiuni pentru simularea numerică :

1. Partea teoretică a lucrării se deschide apăsând butonul .

2. Pentru a efectua experimentul virtual se apasă butonul Masor.

I (A)	-5.0	-4.0	-3.0	-2.0	-1.0	-0.5	0.0	0.5	1.0	2.0	3.0	4.0	5.0	4.0	3.0	2.0	1.0	0.5	0.0	-0.5	-1.0	-2.0	-3.0	-4.0	-5.0	
Φ'(mWb)																										
H(A/m)																										
B(T)																										

Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca

3. Pașii pentru înregistrarea datelor constau în:

- i. Cu ajutorul cursorului se măsoară valorile corespunzătoare fluxului ϕ' (mWb) pentru fiecare valoare a curentului I (A) prin solenoid.
- ii. Se calculează intensitatea câmpului magnetic extern cu ajutorul relației:

$$H = \frac{NI}{l}.$$

- iii. Se calculează inducția câmpului magnetic în miezul de fier cu ajutorul relației:

$$B' = \frac{\Phi'}{nS}.$$

Se dau:

- Numărul de spire: $N = 180$ spire.
 - Numărul de spire al bobinei cuplată la tor: $n = 4$ spire.
 - Secțiunea bobinei: $S = 9 \text{ cm}^2$.
 - Lungimea solenoidului: $l = \pi d_{\text{mediu}}$.
 - Diametrul mediu al solenoidului: $d_{\text{mediu}} = 25 \text{ cm}$.
- iv. Valorile obținute se trec în tabel și e reprezintă grafic dependența $B' = f(H)$.
 - v. În această reprezentare grafică centrul curbei de histerezis are coordonatele: $H = 0$ și $B' = \frac{B'_2 - B'_1}{2}$ (vezi Fig.3 - din referatul de laborator).
 - vi. Valoarea inducției de remanență se calculează cu relația:

$$B_r = \frac{B'_2 - B'_1}{2}$$

-
- vii. Câmpul coercitiv $\pm H$ se citește de pe grafic fiind abscisa punctului de pe curba de histerezis care are ordonata $\frac{B_2' + B_1'}{2}$.

Vă urăm succes!

VII. Codul sursa Java Script

1. Determinarea modului longitudinal de elasticitate

```
let img1,img2,img3;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let MarimeFizical = [];
let masa = [0, 1, 3, 5, 7, 9, 11, 13];
let Err = [];
let diametru;
let suprafata;
let lungime;
let modul = [];
let modulmediu;
let eroarerel = [];
let eroareabs = [];
let ModulYoung;
let Student, Grupa;

let TipMasuratoare;

function preload()
{
    img1 = loadImage("Img/2.jpg");
    img2 = loadImage("Img/Comparator.jpg");
    img3 = loadImage("Img/Stand.jpg");
}

function setup()
{
    createCanvas(1400, 700);
    background(195, 260, 200);
    NrMasuratori = masa.length-1;
    ModulYoung = 180000.0+random(-9000,9000);
    console.log(ModulYoung);
    dist = 25;
    Err[0] = 0.0;
    for (let i=0; i<NrMasuratori; i++)
    {
        let MF1 = new Input(430, height-(NrMasuratori+1)*dist+i*dist+9, "");
        MarimeFizical.push(MF1);
        let MF2 = new Input(500, height-(NrMasuratori+1)*dist+i*dist+9, "");
        modul.push(MF2);
        let MF3 = new Input(630, height-(NrMasuratori+1)*dist+i*dist+9, "");
        eroarerel.push(MF3);
        let MF4 = new Input(690, height-(NrMasuratori+1)*dist+i*dist+9, "");
        eroareabs.push(MF4);
        Err[i+1] = 0.01*random(-1,1);
    }

    diametru = new Input(MarimeFizical[0].x-190,MarimeFizical[0].y+75,"1.0");
    suprafata = new Input(MarimeFizical[0].x-125,MarimeFizical[0].y+75, "");
    lungime = new Input(MarimeFizical[0].x-255,MarimeFizical[0].y+75,"1.03");
    modulmediu = new Input(MarimeFizical[0].x+135,MarimeFizical[0].y+75, "");
```

```
//MarimeFizica1[1].hide();

Student = createInput("");
Student.position(0.5*width+10, 50);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+55, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

suprafata = createInput("");
suprafata.position(0.5*width-190, 463);
suprafata.style('width', '10px');
suprafata.style('font-size', '7px');
suprafata.style('text-align', 'center');

modul = createInput("");
modul.position(0.5*width, 462);
modul.style('width', '10px');
modul.style('font-size', '7px');
modul.style('text-align', 'center');

modulmediu = createInput("");
modulmediu.position(0.5*width+65, 462);
modulmediu.style('width', '10px');
modulmediu.style('font-size', '7px');
modulmediu.style('text-align', 'center');

eroareabs = createInput("");
eroareabs.position(0.5*width+195, 462);
eroareabs.style('width', '10px');
eroareabs.style('font-size', '7px');
eroareabs.style('text-align', 'center');

button0 = createButton("Documentație laborator");
button0.position(290, 60);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '180px');
button0.mouseClicked(Documentatie);

buttonMasor = createButton("Experiment");
buttonMasor.position(button0.x, button0.y+30);
buttonMasor.addClass("button button-primary button-border-thick button-box");
buttonMasor.style('width', '140px');
buttonMasor.mouseClicked(FunctieButon);

rSlider = createSlider(0, NrMasuratori, 0, 1);
rSlider.position(0.7*width+30, height-40);
rSlider.style('width', '350px');
rSlider.style('colour', 'red');
rSlider.hide();

}

function draw()
{
```

```

background(180, 230, 185);
image(img1, width*0.015, 50, width*0.18, height*0.90);
fill(0);
noStroke();
textSize(12);
text("Student: ", Student.x-55, Student.y+5);
text("Grupa: ", Grupa.x-50, Grupa.y+5);

if (Masor)
{

  image(img2, width*0.85+50, 0.6*height, width*0.08, height*0.20);
  image(img3, width*0.69, 40, width*0.2, height*0.80);

  let NrCorpuri = rSlider.value();
  let alungire = NrCorpuri*1;
  let dia = diametru.MyInput.value();

  let alungireCalc = masa[int(NrCorpuri)]*9.81*1.03*4/(ModulYoung*3.1415*dia*dia);
  console.log(alungireCalc);

  fill(0);
  text("m = "+masa[int(NrCorpuri)]+" kg", width*0.72, height-65);
  textSize(16);
  fill(25,0,0);
  text(nf(alungireCalc*1000+Err[NrCorpuri],1,5), width*0.88+35, 0.7*height+17);

  stroke(185,122,85);
  strokeWeight(dia+1);
  line(0.8*width+9, 0.1*height+50, 0.8*width+9, 0.6*height+alungire);
  stroke(20,90,40);
  fill(20,90,40);
  rect(0.8*width+9-50, 0.83*height+alungire, 100, 5);

  strokeWeight(3);
  line(0.8*width+9, 0.625*height+50, 0.8*width+9, 0.83*height+alungire);

  noStroke();
  fill(185,122,85);
  rect(0.8*width+9, 0.5*height+65+alungire, 180, 7);
  fill(100,60,30);
  rect(0.8*width-1, 0.2*height-29, 20, 20);
  if(NrCorpuri>0)
  {
    fill(150);
    stroke(0);
    strokeWeight(1);

    for(let i = 0; i<NrCorpuri; i++)
    {
      rect(0.8*width+9-30, 0.83*height+alungire-8-i*8, 60, 8);
    }
  }
}
Tabel();
}

function Tabel()

```

```

{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de Fizica: Determinarea modului longitudinal de elasticitate", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-45);
  rect(0.698*width, 15,0.3*width-10, height-25);
  for (var i=0; i<11; i++)
  {
    line(MarimeFizica1[0].x-145+i*65,MarimeFizica1[0].y-70,MarimeFizica1[0].x-145+i*65,height-48);
  }
  line(MarimeFizica1[0].x-145,MarimeFizica1[0].y-70,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70);
  line(MarimeFizica1[0].x-145,height-48,MarimeFizica1[0].x+505,height-48);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+64,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+64);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+89,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+89);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+114,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+114);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+139,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+139);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+164,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+164);
  line(MarimeFizica1[0].x-145+260,MarimeFizica1[0].y-70+189,MarimeFizica1[0].x+310,MarimeFizica1[0].y-70+189);

  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+64,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+64);
  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+89,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+89);
  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+114,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+114);
  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+139,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+139);
  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+164,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+164);
  line(MarimeFizica1[0].x-145+520,MarimeFizica1[0].y-70+189,MarimeFizica1[0].x+505,MarimeFizica1[0].y-70+189);
  line(MarimeFizica1[0].x-145,MarimeFizica1[0].y-31,MarimeFizica1[0].x+505,MarimeFizica1[0].y-31);

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentul de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca",330, 685);

  fill(0);
  noStroke();
  textSize(12);
  // Capul de tabel
  text("Material",MarimeFizica1[0].x-135,MarimeFizica1[0].y-50);
  text("Oțel",MarimeFizica1[0].x-125,MarimeFizica1[0].y+60);

```



```

text("1",MarimeFizical[0].x-81+33,MarimeFizical[0].y-55);
text("[m]",MarimeFizical[0].x-88+33,MarimeFizical[0].y-35);

text("d",MarimeFizical[0].x-50+65,MarimeFizical[0].y-55);
text("[mm]",MarimeFizical[0].x-60+65,MarimeFizical[0].y-35);

text("S",MarimeFizical[0].x-50+2*65,MarimeFizical[0].y-57);
text("[10 m ]",MarimeFizical[0].x-75+2*65,MarimeFizical[0].y-35);
text("2",MarimeFizical[0].x-28+2*65,MarimeFizical[0].y-40);

text("m",MarimeFizical[0].x-50+3*65,MarimeFizical[0].y-55);
text("[kg]",MarimeFizical[0].x-55+3*65,MarimeFizical[0].y-35);

text("Δl",MarimeFizical[0].x-40+4*62,MarimeFizical[0].y-57);
text("[10 m]",MarimeFizical[0].x-66+4*65,MarimeFizical[0].y-35);
text("-5",MarimeFizical[0].x-64+4*69,MarimeFizical[0].y-40);

text("E",MarimeFizical[0].x-50+5*65,MarimeFizical[0].y-57);
text("[10 N/m ]",MarimeFizical[0].x-79+5*65,MarimeFizical[0].y-35);
text("2",MarimeFizical[0].x-23+5*65,MarimeFizical[0].y-40);

text("Ē",MarimeFizical[0].x-52+6*65,MarimeFizical[0].y-57);
text("[10 N/m ]",MarimeFizical[0].x-78+6*65,MarimeFizical[0].y-35);
text("2",MarimeFizical[0].x+42+5*65,MarimeFizical[0].y-40);

text("ΔE/E",MarimeFizical[0].x-60+7*65,MarimeFizical[0].y-55);
text("[%]",MarimeFizical[0].x-55+7*65,MarimeFizical[0].y-35);

text("ΔE",MarimeFizical[0].x-55+8*65,MarimeFizical[0].y-57);
text("[10 N/m ]",MarimeFizical[0].x-87+8*66,MarimeFizical[0].y-35);
text("2",MarimeFizical[0].x+167+5*66,MarimeFizical[0].y-40);

for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(masa[i+1],1),MarimeFizical[0].x-70+215,MarimeFizical[0].y+i*dist-12);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/SIM/2010-2011/7.%20Modulul%20longitudinal.pdf';
}

function FunctieButon()
{
  Masor = true;
  rSlider.show();
}

```

2. Determinarea constantei elastice a unui resort

```
let img1,img2,img3;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let Metoda,Resort;
let NrMasuratori, dist;
let Student, Grupa;
let DeltaL = [];
let Delta = [];
let masa1 = [];
let masa2 = [];
let masa11 = [];

let masa22 = [];
let masa3 = [];
let ConstE11 = [];
let ConstE12 = [];
let timp = [];
let timpt = [];
let Perioada = [];
let ConstantaElastica;
let ConstantaElastica1 = [];
let EroareRel1 = [];
let EroareAbs1 = [];
let ConstantaElastica2,ConstantaElastica3, ConstantaElastica4;
let EroareAbs2 = [];
let EroareRel2 = [];
var masa = [0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09];
let alungire;
var Contor0, Contor;
let VariatieK, MasaResort;

function preload()
{
  img1 = loadImage("Img/StandExperimental.jpg");
  img2 = loadImage("Img/Resort.jpg");
  img3 = loadImage("Img/Liniar.webp");
}

function setup()
{
  createCanvas(1510, 760);
  background(280, 270, 270);
  VariatieK = random(-1,1);
  MasaResort = 0.01457;
```

```
Resort = 1;

NrMasuratori = 9;
Metoda = 1;
dist = 25;

for (let i=0; i<NrMasuratori; i++)
{
    let MF1 = new Imput(45, height-(NrMasuratori+1)*dist+i*dist+10);
    masa1.push(MF1);
    let MF2 = new Imput(102, height-(NrMasuratori+1)*dist+i*dist+10);
    masa2.push(MF2);
    let MF3 = new Imput(158, height-(NrMasuratori+1)*dist+i*dist+10);
    DeltaL.push(MF3);
    let MF4 = new Imput(216, height-(NrMasuratori+1)*dist+i*dist+10);
    ConstE11.push(MF4);
    let MF5 = new Imput(330, height-(NrMasuratori+1)*dist+i*dist+10);
    EroareRel1.push(MF5);
    let MF6 = new Imput(386, height-(NrMasuratori+1)*dist+i*dist+10);
    EroareAbs1.push(MF6);
}
ConstantaElastica1 = new Imput(273,ConstE11[1].y);
ConstantaElastica11 = new Imput(273,ConstE11[4].y);
ConstantaElastica12 = new Imput(273,ConstE11[7].y);

for (let i=0; i<NrMasuratori; i++)
{
    let MF7 = new Imput(460, height-(NrMasuratori+1)*dist+i*dist+10);
    masa11.push(MF7);
    let MF8 = new Imput(525, height-(NrMasuratori+1)*dist+i*dist+10);
    masa22.push(MF8);
    let MF9 = new Imput(590, height-(NrMasuratori+1)*dist+i*dist+10);
    masa3.push(MF9);
    let MF10 = new Imput(655, height-(NrMasuratori+1)*dist+i*dist+10);
    timp.push(MF10);
    let MF11 = new Imput(720, height-(NrMasuratori+1)*dist+i*dist+10);
    timpt.push(MF11);
    let MF12 = new Imput(786, height-(NrMasuratori+1)*dist+i*dist+10);
    ConstE12.push(MF12);
    let MF13 = new Imput(915, height-(NrMasuratori+1)*dist+i*dist+10);
    EroareRel2.push(MF13);
    let MF14 = new Imput(980, height-(NrMasuratori+1)*dist+i*dist+10);
    EroareAbs2.push(MF14);
}
ConstantaElastica2 = new Imput(850,ConstE12[1].y);
ConstantaElastica3 = new Imput(850,ConstE12[4].y);
ConstantaElastica4 = new Imput(850,ConstE12[7].y);
```

```
//DeltaL[3].hide();

Student = createInput("");
Student.position(0.5*width+10, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'left');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentatie Lab");
button0.position(45, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentatie);

buttonMS = createButton("Metoda statica");
buttonMS.position(button0.x+button0.width+20, button0.y);
buttonMS.addClass("button button-primary button-border-thick button-box");
buttonMS.style('width', '150px');
buttonMS.mouseClicked(FuncieButon);

buttonMD = createButton("Metoda dinamica");
buttonMD.position(buttonMS.x+buttonMS.width+20, buttonMS.y);
buttonMD.addClass("button button-primary button-border-thick button-box");
buttonMD.style('width', '150px');
buttonMD.mouseClicked(Dinamic);

buttonResort1 = createButton("Resort 1: m= 14,57 g");
buttonResort1.position(0.55*width, 0.1*height);
buttonResort1.addClass("button button-primary button-border-thick button-box");
buttonResort1.style('width', '170px');
buttonResort1.mouseClicked(Resort1);

buttonResort2 = createButton("Resort 2: m= 69,14 g");
buttonResort2.position(buttonResort1.x, buttonResort1.y+20);
buttonResort2.addClass("button button-primary button-border-thick button-box");
buttonResort2.style('width', '170px');
buttonResort2.mouseClicked(Resort2);

buttonResort3 = createButton("Resort 3: m= 145,44 g");
buttonResort3.position(buttonResort1.x, buttonResort2.y+20);
buttonResort3.addClass("button button-primary button-border-thick button-box");
buttonResort3.style('width', '170px');
```

```

buttonResort3.mouseClicked(Resort3);

rSlider = createSlider(0, NrMasuratori, 0, 1);
rSlider.position(width*0.7+30, height-40);
rSlider.style('width', '350px');
rSlider.style('colour', 'red');
rSlider.hide();

d = createDiv();
d.style('padding-left: 50px;');
d.style('transform-origin: 10% 50% 0');
d.style('transform: rotate(' + (90) + 'deg);');
d.position(0.9*width, 180);
dSlider = createSlider(0, 2.5, 0, 0.1);
dSlider.style('width', '350px');
d.child(dSlider);
dSlider.hide();
}

function draw()
{
background(221, 193, 208);
switch (Resort) {
case 2:
ConstantaElastica = 16.7*(1+0.05*VariatieK);
break;
case 3:
ConstantaElastica = 45.0*(1+0.05*VariatieK);
break;
default:
ConstantaElastica = 30.7*(1+0.05*VariatieK);
break;
}
//console.log("Const el."+ConstantaElastica);

text("Student: ", Student.x-55, Student.y+5)
text("Grupa: ", Grupa.x-48, Grupa.y+5)

image(img1, button0.x-10, 70, width*0.47, height*0.52);
Tabel();
if (Masor)
{
if (Metoda == 1)
{
MetodaStatica();
}
else

```

```

    {
      MetodaDinamica();
    }
  }
}
function MetodaStatica()
{
  let NrCorpuri = rSlider.value();
  text("M = "+nf(masa[int(NrCorpuri)],1,2)+" kg",width*0.7+30,height-50);
  alungire = 9.81*masa[int(rSlider.value())]/ConstantaElastica*5000*0.862;
  console.log(alungire);

  image(img2, 0.8*width, 50, 100, height*0.4+alungire);
  fill(185,122,86);
  noStroke();
  rect(0.72*width,20,0.2*width,33);
  push();
  translate(0.85*width-30,0.15*height-58);
  rotate(Math.PI*0.5);
  image(img3, 0, -30, height*0.8, height*0.3);
  pop();
  stroke(200,0,0);
  line(0.76*width,45+height*0.4+alungire,0.84*width,45+height*0.4+alungire);
  if(NrCorpuri>0)
  {
    fill(150);
    stroke(0);
    for(let i=0; i<NrCorpuri; i++)
    {
      rect(width*0.8,45+height*0.4+alungire+i*10+1,100,10);
    }
  }
}
function MetodaDinamica()
{
  let MasaCorp = round(100*(0.01+masa[int(rSlider.value())])*0.1);
  let MasaM = MasaCorp+MasaResort/3;
  let niu = Math.sqrt(ConstantaElastica/(MasaM));
  //console.log("T = "+2*Math.PI/niu);
  //console.log("w = "+niu);

  angleMode(RADIANS);
  let AmpOsc = 10*dSlider.value();
  let AlungireVar = AmpOsc*Math.sin(niu*(millis()-Contor0)*0.001); // Aici
  let NrCorpuri = rSlider.value()+1;
  text("M = "+nf(MasaCorp,1,2)+" kg",width*0.7+30,height-50);
  text("Amplitudine de oscilatie",width*0.9-30,0.3*height-40);
  alungire = 9.81*masa[int(rSlider.value())]/ConstantaElastica*5000*0.862;

```

```

console.log(alungire);
image(img2, 0.8*width, 50, 100, height*0.4+alungire+AlungireVar);

fill(185,122,86);
noStroke();
rect(0.72*width,20,0.2*width,33);
push();
  translate(0.85*width-30,0.15*height-58);
  rotate(Math.PI*0.5);
  image(img3, 0, -30, height*0.8, height*0.3);
pop();
stroke(200,0,0);
line(0.76*width,45+height*0.4+alungire+AlungireVar,0.84*width,45+height*0.4+alungire+AlungireVar);
if(NrCorpuri>0)
{
  fill(150);
  stroke(0);
  for(let i=0; i<NrCorpuri; i++)
  {
    rect(width*0.8,45+height*0.4+alungire+AlungireVar+i*10+1,100,10);
  }
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  //noStroke();
  stroke(200,0,0);
  text("Laboratorul de Fizica: Constanta elastica", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-45);
  rect(0.7*width, 13,0.3*width-10, height-25);
  for (var i=0; i<8; i++)
  {
    line(masa1[0].x-13+i*57,masa1[0].y-50,masa1[0].x-13+i*57,height-25);
  }

  for (var i=0; i<10; i++)
  {
    line(masa11[0].x-17+i*65,masa11[0].y-50,masa11[0].x-17+i*65,height-25);
  }
}

```

```

}
line(masa1[0].x-13,masa1[0].y-50,masa1[0].x+385,masa1[0].y-50);
line(masa1[0].x-13,height-25,masa1[0].x+385,height-25);
line(masa1[0].x-13,masa1[0].y-15,masa1[0].x+385,masa1[0].y-15);
line(masa11[0].x-17,masa11[0].y-50,masa11[0].x+568,masa11[0].y-50);
line(masa11[0].x-17,height-25,masa11[0].x+568,height-25);
line(masa11[0].x-17,masa11[0].y-15,masa11[0].x+568,masa11[0].y-15);

fill(0);
noStroke();
 textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",width*0.16+13,height-13);

```

// Capul de tabel 1

```

text("m [g]",masa1[0].x-0+0*65,masa1[0].y-30);
text("M [g]",masa1[0].x-2+1*65,masa1[0].y-30);
text("ΔL",masa1[0].x-2+2*65,masa1[0].y-35);
text("[mm]",masa1[0].x-8+2*65,masa1[0].y-20);
text("k",masa1[0].x-5+3*65,masa1[0].y-35);
text("[N/m]",masa1[0].x-20+3*65,masa1[0].y-20);
text("k",masa1[0].x-15+4*65,masa1[0].y-35);
text("_",masa1[0].x-15+4*65,masa1[0].y-48);
text("[N/m]",masa1[0].x-30+4*65,masa1[0].y-20);
text("ΔK/k",masa1[0].x-32+5*65,masa1[0].y-35);
text("[%]",masa1[0].x-30+5*65,masa1[0].y-20);
text("Δk",masa1[0].x-35+6*65,masa1[0].y-35);
text("[N/m]",masa1[0].x-42+6*65,masa1[0].y-20);

```

// Capul de tabel 2

```

text("m [g]",masa11[0].x-60+1*65,masa11[0].y-30);
text("M [g]",masa11[0].x-60+2*65,masa11[0].y-30);
text("M+m/3",masa11[0].x-0+2*65,masa11[0].y-35);
text("[g]",masa11[0].x-55+3*65,masa11[0].y-20);
text("t",masa11[0].x-50+4*65,masa11[0].y-35);
text("[s]",masa11[0].x-55+4*65,masa11[0].y-20);
text("T",masa11[0].x-52+5*65,masa11[0].y-35);
text("[s]",masa11[0].x-55+5*65,masa11[0].y-20);
text("k",masa11[0].x-50+6*65,masa11[0].y-35);
text("[N/m]",masa11[0].x-64+6*65,masa11[0].y-20);
text("k",masa11[0].x-50+7*65,masa11[0].y-35);
text("_",masa11[0].x-50+7*65,masa11[0].y-48);
text("[N/m]",masa11[0].x-60+7*65,masa11[0].y-20);
text("Δk/k",masa11[0].x-0+7*65,masa11[0].y-35);
text("[%]",masa11[0].x-60+8*65,masa11[0].y-20);
text("Δk",masa11[0].x-55+9*65,masa11[0].y-35);
text("[N/m]",masa11[0].x-60+9*65,masa11[0].y-20);

```



```
for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(masa[i],1,2),masa1[0].x-105,masa1[0].y+i*dist+7);
  text(nf(masa[i]+0.01,1,2),masa1[0].x-60,masa1[0].y+i*dist+7);
}

}

function Resort1()
{
  Resort = 1;
  MasaResort = 0.01457;
}

function Resort2()
{
  Resort = 2;
  MasaResort = 0.06914;
}

function Resort3()
{
  Resort = 3;
  MasaResort = 0.14544;
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Constanta%20elastica/Doc/L1_Determinarea%20constantei%20elastice%20a%20unui%20resort_Petru.pdf';
}

function FunctieButon()
{
  rSlider.show();
  dSlider.hide();
  Masor = true;
  Metoda = 1;
}

function Dinamic()
{
  dSlider.show();
  rSlider.show();
  Masor = true;
  Contor0 = millis();
  Metoda = 2;
}
```

3. Determinarea accelerației gravitaționale cu ajutorul pendulului fizic

```

let img1, img2, img3, img4;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let Oscileaza = false;
var timp0, Contor;
let hMicuratori, dist;
let Unghi, UnghiMaxim;
let Hasul = [];
let Perioada = [];
let hT2 = [];
let hT2i0;
let hPatrat = [];
let Timp = [];
let Masa, Acceleratia, IZero;
var hMic = [10, 20, 30, 40, 10, 20, 30, 40];
let SaltGrafic = [62, 113, 165, 222];
let SaltGrafic1 = [63, 114, 166, 223];
let CorectieImg = [0, 0, 0, -5];
let panta, intersectia, eroarea;
let TipMasuratoare;
let Student, Grupa;
let IZeroUL, Niu, PerioadaT;
let Pozitie, Salt;
let nroscilatii;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};

function preload()
{
  img1 = loadImage("Img/Fig1.png");
  img2 = loadImage("Img/Capture.jpg");
  img3 = loadImage("Img/PendululFizicSus1.jpg");
  img4 = loadImage("Img/PendululFizicJos1.jpg");
}

function setup()
{
  createCanvas(1700, 700);
  background(220, 200, 200);
  hMicuratori = hMic.length;
  IZeroUL = 1*(1+0.05*random(-1,1));
  UnghiMaxim = 5;
  Pozitie = true;
  dist = 25;

```

```

Student = createInput("");
Student.position(0.5*width+10, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

for (let i=0; i<hMicuratori; i++)
{
  let MF1 = new Imput(165,height-(hMicuratori+1)*dist+i*dist+5,nf(hMic[i]));
  Hasul.push(MF1);
  let MF2 = new Imput(355,height-(hMicuratori+1)*dist+i*dist+5, "");
  Perioada.push(MF2);
  let MF3 = new Imput(420,height-(hMicuratori+1)*dist+i*dist+5, "");
  hT2.push(MF3);
  let MF4 = new Imput(485,height-(hMicuratori+1)*dist+i*dist+5, "");
  hPatrat.push(MF4);
  let MF5 = new Imput(290,height-(hMicuratori+1)*dist+i*dist+5, "");
  Timp.push(MF5);
}

Masa = new Imput(100,height-130,"10");
nrosilatii = new Imput(230,height-130,"");
Acceleratia = new Imput(555,height-130,"");
IZero = new Imput(620,height-130,"");

button0 = createButton("Documentatie Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentatie);

buttonGB = createButton("MăSOR");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(Hasul[0].x+0.645*width, Hasul[0].y-120);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width-10', '150px');
buttonGraf.mouseClicked(ArataGrafic);

buttonStart = createButton("Oscileaza");
buttonStart.position(button0.x, button0.y+60);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");

```

```

buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

buttonSus = createButton("Sus");
buttonSus.position(button0.x, button0.y+100);
buttonSus.addClass("button button-primary button-border-thick button-box");
buttonSus.style('width', '100px');
buttonSus.mouseClicked(Sus);
buttonSus.hide();

buttonJos = createButton("Jos");
buttonJos.position(buttonSus.x+buttonSus.width+20, buttonSus.y);
buttonJos.addClass("button button-primary button-border-thick button-box");
buttonJos.style('width', '100px');
buttonJos.mouseClicked(Jos);
buttonJos.hide();

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
Graf.style('height', '300px');

rSlider = createSlider(0, 3, 0, 1);
rSlider.position(Hasul[0].x-100, Hasul[0].y-80);
rSlider.style('width', '350px');
rSlider.style('colour', 'red');
rSlider.hide();

}

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.75, 25, width*0.20, height*0.45);
  image(img2, width*0.245, button0.y-10, width*0.20, height*0.55);
  Tabel();

  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);
  if (Masor)
  {
    let hpart = 0.01*hMic[int(rSlider.value())];
    let SaltG = SaltGrafic[int(rSlider.value())];
    let SaltG1 = SaltGrafic1[int(rSlider.value())];
    let CorImg= CorectieImg[int(rSlider.value())];
    PerioadaT = 2*Math.PI*sqrt(IZeroUL/Masa.MyInput.value()/9.81/hpart+hpart/9.81);
    Niu = 1.0/PerioadaT;
    angleMode(DEGREES);
    Unghi = 0;
    if(Oscileaza)
    {

```

```

    let t = (millis()-timp0)*0.001;
    Unghi = UnghiMaxim*Math.sin(2*Math.PI*t*Niu);
  }
  if(Pozitie)
  {
    text("d = "+nf(hpart*100,1,0)+" cm",rSlider.x,rSlider.y-10);
  }else{
    text("d = "+nf(hpart*100,1,0)+" cm",rSlider.x,rSlider.y-10);
  }
}

push();
translate(0.5*width+60,0.5*height-SaltG);
push();
rotate(Unghi);
if(Pozitie)
{
  image(img4, -0.025*width+7, -0.4*height+SaltG-7+CorImg, width*0.05, height*0.8);
  fill(220,200,200);
  noStroke();
  rect(-50,0.4*height+SaltG-10,100,50);
}else{
  image(img3, -0.025*width+7, -0.4*height+SaltG1-7+CorImg, width*0.05, height*0.8);
  fill(220,200,200);
  noStroke();
  rect(-50,0.4*height+SaltG-10,100,50);
}
pop();
fill(136,0,27);
strokeWeight(1);
stroke(0);
triangle(0,0,-5,20,5,20);

pop();

//textSize(16);
//text(nf(Unghi,1,6),0.43*width,0.585*height);
//text(nf(PerioadaT,1,6),0.43*width,0.585*height+20);
for (let i=0; i<hMicuratori; i++)
{
  trace1.x[i] = Perioada[i].MyInput.value();
  trace1.y[i] = hPatrat[i].MyInput.value();
}
LinFit(trace1.x,trace1.y,hMicuratori-1);
var Minim = 0.9*min(trace1.x);
var Maxim = 1.1*max(trace1.x);
var NrFit = 101;
var dx = (Maxim-Minim)/(NrFit-1.0);
for (var i = 0; i< NrFit; i++)
{
  trace2.x[i] = Minim+i*dx;
  trace2.y[i] = intersectia+panta*trace2.x[i];
}
}
Tabel();
if(ImgArataGrafic == 'Grafic Offline')
{
  Graf.show();
  Grafic();
}

```

```

}
else
{
  Graf.hide();
}
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de Fizica: Determinarea acceleratiei gravitationale cu pendulul fizic ", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-45);
  rect(0.7*width, 15,0.3*width-10, height-25);
  for (var i=0; i<11; i++)
  {
    line(Hasul[0].x-150+i*65,Hasul[0].y-50,Hasul[0].x-150+i*65,height-25);
  }
  line(Hasul[0].x-150,Hasul[0].y-50,Hasul[0].x+500,Hasul[0].y-50);
  line(Hasul[0].x-150,height-25,Hasul[0].x+500,height-25);
  line(Hasul[0].x-150,Hasul[0].y-15,Hasul[0].x+500,Hasul[0].y-15);

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",330, 686);

  fill(0);
  noStroke();
  textSize(12);
  // Capul de tabel
  text("nr.crt",Hasul[0].x-130,Hasul[0].y-30);
  text("m",Hasul[0].x-60,Hasul[0].y-35);
  text("[Kg]",Hasul[0].x-66,Hasul[0].y-20);
  text("d",Hasul[0].x-60+1*65,Hasul[0].y-35);
  text("[cm]",Hasul[0].x-68+1*65,Hasul[0].y-20);
  text("n",Hasul[0].x-60+2*65,Hasul[0].y-35);
  text("t",Hasul[0].x-57+3*65,Hasul[0].y-35);
  text("[s]",Hasul[0].x-62+3*65,Hasul[0].y-20);
  text("T",Hasul[0].x-60+4*65,Hasul[0].y-35);
  text("[s]",Hasul[0].x-62+4*65,Hasul[0].y-20);
  text("dT",Hasul[0].x-58+5*65,Hasul[0].y-35);
  text("[ cms ]",Hasul[0].x-68+5*65,Hasul[0].y-20);
  text("d",Hasul[0].x-60+6*65,Hasul[0].y-35);
  text("[cm ]",Hasul[0].x-68+6*65,Hasul[0].y-20);
  text("g",Hasul[0].x+10+6*65,Hasul[0].y-35);
  text("[m/s ]",Hasul[0].x+6*65,Hasul[0].y-20);
  text("T",Hasul[0].x+75+6*65,Hasul[0].y-35);
  text("[Kg*m ]",Hasul[0].x+55+6*65,Hasul[0].y-20);

```

```

for (let i = 0; i < hMicuratori; i++)
{
  text(nf(i+1,1,0),Hasul[0].x-130,Hasul[0].y+i*dist+7);
}
textSize(9);
text("2",Hasul[0].x-45+5*65,Hasul[0].y-40);
text("2",Hasul[0].x-53+6*65,Hasul[0].y-40);
text("0",Hasul[0].x+80+6*65,Hasul[0].y-30);
text(" 2",Hasul[0].x-40+5*65,Hasul[0].y-25);
text(" 2",Hasul[0].x-50+6*65,Hasul[0].y-25);
text(" 2",Hasul[0].x+22+6*65,Hasul[0].y-25);
text(" 2",Hasul[0].x+85+6*65,Hasul[0].y-25);
}

```

```

function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

```

```

function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Pendulul Fizic',
    xaxis:
    {
      title:'d^2 [cm^2]',
      visible: 1
    },
    yaxis:
    {
      title:'dT^2 [cm s^2]'
    },
    paper_bgcolor: "#c4dedc"
  };

  var data = [trace1,trace2];
  //let data = [trace1];

```

```
Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Pendulul Fizic/Doc/01%20Pendulul%20Fizic.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
  buttonSus.show();
  buttonJos.show();
  rSlider.show();
}

function Start()
{
  Oscileaza = true;
  timp0 = millis();
}

function Stop()
{
  Oscileaza = false;
}

function Sus()
{
  Pozitie = true;
}

function Jos()
{
  Pozitie = false;
}

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
  }
}
```

```
    sx += x[i]*f;
    sxx+= x[i]*x[i]*f;
    sy += y[i]*f;
    sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
    eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

4. Determinarea timpului de ciocnire a două bile

```

let img1;
let Masor = false;
let NrMasuratori, dist;
let MarimeFizica1 = [];
let MarimeFizica2 = [];
let MarimeFizica3 = [];
let Xuri = [1, 2, 3, 4, 5, 1, 2, 3, 4, 5];
let Student, Grupa;
let Capacitate1, Rezistenta1, RC1;
let Capacitate2, Rezistenta2, RC2, U1, U2, t;
let Capacitate3, Rezistenta3, RC3, U01, U02, t0;
let TensIncarc, Qmax, TensMas, DELTAQ;
let ciocnire;
let ComutatorK;
let CapacitateCondensator, Qcond;
let timp0, timp1;
let Pas;
let Rm1, Vm1, Am1;
let Rm2, Vm2, Am2;
let RB1, VB1, AB1;
let RB2, VB2, AB2;
let unghiB0;
let unghiB1, unghiB2;
let TipMasuratoare;

function preload()
{
    img1 = loadImage("Img/Circuit.jpg");
}

function setup()
{
    createCanvas(1400, 800);
    background(220, 200, 200);
    NrMasuratori = Xuri.length;
    ComutatorK = true;
    CapacitateCondensator = 22;
    Qcond = 0;
    DELTAQ = 0;
    TensMas = Qcond/CapacitateCondensator;
    unghiB0 = 26;
    Pas = 0;
    TipMasuratoare = 0;
    dist = 25;

    for (let i=0; i<NrMasuratori; i++)
    {
        let MF1 = new Input(100,height-(NrMasuratori+1)*dist+i*dist+17, "");
        MarimeFizica1.push(MF1);
        let MF2 = new Input(165,height-(NrMasuratori+1)*dist+i*dist+17, "");
        MarimeFizica2.push(MF2);
        let MF3 = new Input(230,height-(NrMasuratori+1)*dist+i*dist+17, "");
        MarimeFizica3.push(MF3);

        let U1 = new Input(100,height-40-(NrMasuratori+1)*dist+i*dist-313, "");
        MarimeFizica1.push(U1);
    }
}

```

```

let U2 = new Input(165,height-40-(NrMasuratori+1)*dist+i*dist-313,"");
MarimeFizica2.push(U2);
let t = new Input(230,height-40-(NrMasuratori+1)*dist+i*dist-313,"");
MarimeFizica3.push(t);

let U01 = new Input(515,height-(NrMasuratori+1)*dist+i*dist+17,"");
MarimeFizica1.push(U01);
let U02 = new Input(580,height-(NrMasuratori+1)*dist+i*dist+17,"");
MarimeFizica2.push(U02);
let t0 = new Input(645,height-(NrMasuratori+1)*dist+i*dist+17,"");
MarimeFizica3.push(t0);
}

Capacitate1 = new Input(MarimeFizica1[0].x-195,MarimeFizica1[0].y-250," ");
Rezistenta1 = new Input(MarimeFizica1[0].x-128,MarimeFizica1[0].y-320," ");
RC1 = new Input(MarimeFizica1[0].x-65,MarimeFizica1[0].y-320," ");
Rezistenta1 = new Input(MarimeFizica2[0].x-195,MarimeFizica1[0].y-195," ");
RC1 = new Input(MarimeFizica2[0].x-130,MarimeFizica1[0].y-195," ");

Capacitate2 = new Input(MarimeFizica1[0].x-195,MarimeFizica1[0].y+110," ");
Rezistenta2 = new Input(MarimeFizica1[0].x-128,MarimeFizica1[0].y+50," ");
RC2 = new Input(MarimeFizica1[0].x-65,MarimeFizica1[0].y+50," ");
Rezistenta2 = new Input(MarimeFizica2[0].x-195,MarimeFizica1[0].y+175," ");
RC2 = new Input(MarimeFizica2[0].x-130,MarimeFizica1[0].y+175," ");

Capacitate3 = new Input(MarimeFizica1[0].x+220,MarimeFizica1[0].y+110," ");
Rezistenta3 = new Input(MarimeFizica1[0].x+285,MarimeFizica1[0].y+50," ");
RC3 = new Input(MarimeFizica1[0].x+350,MarimeFizica1[0].y+50," ");
Rezistenta3 = new Input(MarimeFizica1[0].x+285,MarimeFizica1[0].y+175," ");
RC3 = new Input(MarimeFizica1[0].x+350,MarimeFizica1[0].y+175," ");

Student = createInput("");
Student.position(0.8*width, 20);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+55, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentație Laborator");
button0.position(400, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '200px');
button0.mouseClicked(Documentatie);

buttonM = createButton("Experiment");
buttonM.position(button0.x+30, button0.y+button0.width-170);
buttonM.addClass("button button-primary button-border-thick button-box");
buttonM.style('width', '150px');
buttonM.mouseClicked(FunctieButon);

buttonoo = createButton("Ciocnire oțel - oțel");
buttonoo.position(30, 80);
buttonoo.addClass("button button-primary button-border-thick button-box");

```

```
buttonoo.style('width', '150px');
buttonoo.mouseClicked(OtelOtel);

buttonop = createButton("Ciocnire oțel - plumb");
buttonop.position(buttonoo.x, buttonoo.y+buttonoo.width+235);
buttonop.addClass("button button-primary button-border-thick button-box");
buttonop.style('width', '150px');
buttonop.mouseClicked(OtelPlumb);

buttonoc = createButton("Ciocnire oțel - cupru");
buttonoc.position(buttonop.x+buttonop.width+265, buttonop.y);
buttonoc.addClass("button button-primary button-border-thick button-box");
buttonoc.style('width', '150px');
buttonoc.mouseClicked(OtelCupru);

rSlider = createSlider(1, 15, 4, 0.1);
rSlider.position(0.7*width-25, 0.1*height-5);
rSlider.style('width', '140px');
rSlider.style('colour', 'red');
rSlider.hide();

buttonMAS = createButton("Măsoară");
buttonMAS.position(rSlider.x-140,0.35*height-0);
buttonMAS.addClass("button button-primary button-border-thick button-box");
buttonMAS.style('width', '80px');
buttonMAS.mouseClicked(ButonMAS);
buttonMAS.hide();

buttonINC = createButton("Încarcă");
buttonINC.position(rSlider.x+20,0.35*height-0);
buttonINC.addClass("button button-primary button-border-thick button-box");
buttonINC.style('width', '80px');
buttonINC.mouseClicked(ButonINC);
buttonINC.hide();

TensIncarc = new Input(buttonINC.x-113,buttonINC.y-76,"5.8");
TensIncarc.hide();

buttonCADERE = createButton("Cădere");
buttonCADERE.position(rSlider.x-50,0.5*height-0);
buttonCADERE.addClass("button button-primary button-border-thick button-box");
buttonCADERE.style('width', '100px');
buttonCADERE.mouseClicked(ButonCADERE);
buttonCADERE.hide();

buttonRes = createButton("Reset");
buttonRes.position(buttonCADERE.x+300,0.5*height-0);
buttonRes.addClass("button button-primary button-border-thick button-box");
buttonRes.style('width', '100px');
buttonRes.mouseClicked(ButonRes);
buttonRes.hide();

}

function draw()
{
  background(130, 165, 140);
  fill(0);
  noStroke();
```

```

textSize(12);
text("Student: ", Student.x-55,Student.y+5);
text("Grupa: ", Grupa.x-45,Grupa.y+5);

if (Masor)
{
  CineSeCiocneste();
  Qmax = CapacitateCondensator*TensIncarc.MyInput.value();
  fill(180,230,185);
  noStroke();
  rect(0.6*width-5, 0.1*height-35,width*0.4-20,height*0.9+5);
  image(img1, rSlider.x-337, rSlider.y-30, width*0.4, height*0.40);
  let Rez = rSlider.value();
  fill(200,0,0);
  textSize(14);
  text("R = "+nf(Rez,1,1)+" Ω",rSlider.x+40,rSlider.y+27);

  strokeWeight(4);
  stroke(0);
  line(rSlider.x-3,rSlider.y+2,rSlider.x+(Rez-1)*9,rSlider.y+2);
  if(ComutatorK)
  {
    line(rSlider.x-25,0.35*height-8,rSlider.x-52,0.35*height-25);
    Qcond = Qcond*Math.exp(-0.000000001*(millis()-timp0));
    TensMas = Qcond/CapacitateCondensator;
  }
  else
  {
    line(rSlider.x-25,0.35*height-8,rSlider.x+2,0.35*height-25);
    Qcond = Qmax*(1-exp(-0.00001*(millis()-timp0)))+Qcond*exp(-0.00001*(millis()-timp0));
    TensMas = 0;
    Pas = 0;
  }
  noStroke();
  text("Q",buttonMAS.x-93,buttonMAS.y-130);
  fill(0,0,200);
  text(nf(TensMas,1,2)+" V",buttonMAS.x-30,buttonMAS.y-60);
  strokeWeight(1);
  stroke(0);
  fill(200);
  let xC = 0.515*width;
  let yC = 0.2*height;
  rect(xC,yC,11,100);
  noStroke();
  fill(200,0,0);
  let BaraQcond = map(Qcond,0,Qmax,0,100);
  rect(xC+1,yC+101-BaraQcond,10,BaraQcond-1);
  Ciocnire();
  fill(0);
  text("V",buttonINC.x+55,buttonINC.y-69);
}
Tabel();
}

function Ciocnire()
{
  push();
  translate(0.77*width+1,0.41*height+5);
  angleMode(DEGREES);

```

```
switch (Pas) {
  case 1:
    Rm1 = 17; Vm1 = 111; Am1 = 21;
    Rm2 = 132; Vm2 = 255; Am2 = 9;
    if(unghiB1>0.1)
    {
      unghiB1 = unghiB1-1;
    }else{
      unghiB1 = 0;
      Pas = 2;
    }
    unghiB2 = 0;
    break;
  case 2:
    Rm1 = 17; Vm1 = 111; Am1 = 21;
    Rm2 = 132; Vm2 = 255; Am2 = 9;
    unghiB1 = 0;
    unghiB2 = 0;
    SeCiocneste();
    if(Qcond-DELTAQ>0)
    {
      Qcond -= DELTAQ;
    }else{
      Qcond = 0;
    }
    Pas = 3
    break;
  case 3:
    Rm1 = 17; Vm1 = 111; Am1 = 21;
    Rm2 = 132; Vm2 = 255; Am2 = 9;
    unghiB1 = 0;
    if(unghiB2<unghiB0)
    {
      unghiB2 += 1;
    }else{
      unghiB2 = unghiB0;
      Pas = 4
    }
    break;
  case 4:
    Rm1 = 17; Vm1 = 111; Am1 = 21;
    Rm2 = 132; Vm2 = 255; Am2 = 9;
    unghiB2 = unghiB0;
    unghiB1 = 0;
    break;
  default:
    Rm1 = 132; Vm1 = 255; Am1 = 9;
    Rm2 = 17; Vm2 = 111; Am2 = 21;
    unghiB1 = unghiB0;
    unghiB2 = 0;
    console.log(unghiB1);
}
push();
rotate(unghiB1);
stroke(185,182,86);
strokeWeight(2);
line(0,0,0,400);
noStroke();
fill(RB1,VB1,AB1);
ellipse(0,400,25,25);
```

```

    fill(0);
    ellipse(0,0,5,5);
    pop();
    push();
    translate(23,0);
    rotate(-unghiB2);
    stroke(185,182,86);
    strokeWeight(2);
    line(0,0,0,400);
    noStroke();
    fill(RB2,VB2,AB2);
    ellipse(0,400,25,25);
    fill(0);
    ellipse(0,0,5,5);
    pop();
    pop();
    stroke(0);
    strokeWeight(1);
    fill(Rm1,Vm1,Am1);
    rect(0.6*width,0.85*height,50,20);
    fill(Rm2,Vm2,Am2);
    rect(0.922*width-1,0.85*height,50,20);
}

function CineSeCiocneste()
{
  switch (TipMasuratoare) {
    case 1:
      RB1 = 92; VB1 = 92; AB1 = 92;
      RB2 = 92; VB2 = 92; AB2 = 152;
      break;
    case 2:
      RB1 = 92; VB1 = 92; AB1 = 92;
      RB2 = 185; VB2 = 120; AB2 = 90;
      break;
    default:
      RB1 = 92; VB1 = 92; AB1 = 92;
      RB2 = 92; VB2 = 92; AB2 = 92;
  }
}

function SeCiocneste()
{ let TimpCiocnire;
  let Rezi = rSlider.value();
  switch (TipMasuratoare) {
    case 1:
      TimpCiocnire = 170*(1+0.05*random(-1,1));
      break;
    case 2:
      TimpCiocnire = 260*(1+0.05*random(-1,1));
      break;
    default:
      TimpCiocnire = 50*(1+0.05*random(-1,1));
  }
  DELTAQ = Qcond*(1-exp(-(TimpCiocnire/CapacitateCondensator/Rezi)));
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);

```

```

noStroke();
text("Laboratorul de Fizică: Determinarea timpului de ciocnire a două bile", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,width-20, height-45);

for (var i=0; i<7; i++)
{
  line(MarimeFizical[0].x-85+i*65,MarimeFizical[0].y-425,MarimeFizical[0].x-85+i*65,height-385);
  line(MarimeFizical[0].x-85+i*65,MarimeFizical[0].y-50,MarimeFizical[0].x-85+i*65,height-15);
  line(MarimeFizical[0].x+330+i*65,MarimeFizical[0].y-50,MarimeFizical[0].x+330+i*65,height-15);
}
line(MarimeFizical[0].x-85,MarimeFizical[0].y-425,MarimeFizical[0].x+305,MarimeFizical[0].y-425);
line(MarimeFizical[0].x-85,height-385,MarimeFizical[0].x+305,height-385);
line(MarimeFizical[0].x-20,MarimeFizical[0].y-256,MarimeFizical[0].x+305,MarimeFizical[0].y-256);
line(MarimeFizical[0].x-85,MarimeFizical[0].y-385,MarimeFizical[0].x+305,MarimeFizical[0].y-385);

line(MarimeFizical[0].x-85,MarimeFizical[0].y-50,MarimeFizical[0].x+305,MarimeFizical[0].y-50);
line(MarimeFizical[0].x-85,height-15,MarimeFizical[0].x+305,height-15);
line(MarimeFizical[0].x-85,MarimeFizical[0].y-15,MarimeFizical[0].x+305,MarimeFizical[0].y-15);
line(MarimeFizical[0].x-20,MarimeFizical[0].y+114,MarimeFizical[0].x+305,MarimeFizical[0].y+114);

line(MarimeFizical[0].x+330,MarimeFizical[0].y-50,MarimeFizical[0].x+720,MarimeFizical[0].y-50);
line(MarimeFizical[0].x+330,height-15,MarimeFizical[0].x+720,height-15);
line(MarimeFizical[0].x+330,MarimeFizical[0].y-15,MarimeFizical[0].x+720,MarimeFizical[0].y-15);
line(MarimeFizical[0].x+395,MarimeFizical[0].y+114,MarimeFizical[0].x+720,MarimeFizical[0].y+114);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",828, 785);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("C",MarimeFizical[0].x-57, MarimeFizical[0].y-412);
text("[μF]",MarimeFizical[0].x-63, MarimeFizical[0].y-395);
text("R",MarimeFizical[0].x+10,MarimeFizical[0].y-412);
text("[Ω]",MarimeFizical[0].x+5, MarimeFizical[0].y-395);
text("RC",MarimeFizical[0].x+70,MarimeFizical[0].y-412);
text("[μs]",MarimeFizical[0].x+69, MarimeFizical[0].y-395);
text("U",MarimeFizical[0].x+140,MarimeFizical[0].y-412);
text("i",MarimeFizical[0].x+148,MarimeFizical[0].y-407);
text("[V]",MarimeFizical[0].x+138, MarimeFizical[0].y-395);
text("U",MarimeFizical[0].x+200,MarimeFizical[0].y-412);
text("c",MarimeFizical[0].x+208,MarimeFizical[0].y-407);
text("[V]",MarimeFizical[0].x+200, MarimeFizical[0].y-395);
text("t*10",MarimeFizical[0].x+260,MarimeFizical[0].y-412);
text("-5",MarimeFizical[0].x+281,MarimeFizical[0].y-416);
text("[s]",MarimeFizical[0].x+265, MarimeFizical[0].y-395);

```



```

text("C",MarimeFizical[0].x-57, MarimeFizical[0].y-35);
text("[μF]",MarimeFizical[0].x-63, MarimeFizical[0].y-20);
text("R",MarimeFizical[0].x+10,MarimeFizical[0].y-35);
text("[Ω]",MarimeFizical[0].x+5, MarimeFizical[0].y-20);
text("RC",MarimeFizical[0].x+70,MarimeFizical[0].y-35);
text("[μs]",MarimeFizical[0].x+69, MarimeFizical[0].y-20);
text("U",MarimeFizical[0].x+140,MarimeFizical[0].y-35);
text("i",MarimeFizical[0].x+148,MarimeFizical[0].y-32);
text("[V]",MarimeFizical[0].x+138, MarimeFizical[0].y-20);
text("U",MarimeFizical[0].x+200,MarimeFizical[0].y-35);
text("c",MarimeFizical[0].x+208,MarimeFizical[0].y-32);
text("[V]",MarimeFizical[0].x+200, MarimeFizical[0].y-20);
text("t*10",MarimeFizical[0].x+260,MarimeFizical[0].y-35);
text("-5",MarimeFizical[0].x+281,MarimeFizical[0].y-38);
text("[s]",MarimeFizical[0].x+265, MarimeFizical[0].y-20);

```

```

text("C",MarimeFizical[0].x+357, MarimeFizical[0].y-35);
text("[μF]",MarimeFizical[0].x+353, MarimeFizical[0].y-20);
text("R",MarimeFizical[0].x+425,MarimeFizical[0].y-35);
text("[Ω]",MarimeFizical[0].x+420, MarimeFizical[0].y-20);
text("RC",MarimeFizical[0].x+485,MarimeFizical[0].y-35);
text("[μs]",MarimeFizical[0].x+483, MarimeFizical[0].y-20);
text("U",MarimeFizical[0].x+550,MarimeFizical[0].y-35);
text("i",MarimeFizical[0].x+558,MarimeFizical[0].y-32);
text("[V]",MarimeFizical[0].x+547, MarimeFizical[0].y-20);
text("U",MarimeFizical[0].x+617,MarimeFizical[0].y-35);
text("c",MarimeFizical[0].x+625,MarimeFizical[0].y-32);
text("[V]",MarimeFizical[0].x+615, MarimeFizical[0].y-20);
text("t*10",MarimeFizical[0].x+673,MarimeFizical[0].y-35);
text("-5",MarimeFizical[0].x+695,MarimeFizical[0].y-38);
text("[s]",MarimeFizical[0].x+678, MarimeFizical[0].y-20);

```

```

}

```

```

function Documentatie()

```

```

{

```

```

    window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2018-2019/lucrari/1.Timpul%20de%20ciocnire.pdf';

```

```

}

```

```

function FunctieButon()

```

```

{

```

```

    Masor = true;
    rSlider.show();
    buttonMAS.show();
    buttonINC.show();
    timp0 = millis();
    TensIncarc.show();
    buttonCADERE.show();
    buttonRes.show();

```

```

}

```

```

function ButonMAS()

```

```

{

```

```

    ComutatorK = true;
    timp0 = millis();

```

```

}

```

```

function ButonINC()

```

```
{
  ComutatorK = false;
  timp0 = millis();
}

function ButonCADERE()
{
  Pas = 1;
  timp1 = millis();
}

function ButonRes()
{
  Pas = 0;
  timp1 = millis();
}

function OtelOtel()
{
  TipMasuratoare = 0;
}

function OtelPlumb()
{
  TipMasuratoare = 1;
}

function OtelCupru()
{
  TipMasuratoare = 2;
}
```

5. Studiul undelor staționare in corzi vibrante

```

let img1;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Masa,Masa1,Masa2,T1, T2, v1, v2;
let NiuC1 = [];
let NiuC2 = [];
let NiuE1 = [];
let NiuE2 = [];
let LamdaC1 = [];
let LamdaC2 = [];
let LamdaE1 = [];
let LamdaE2 = [];
let TipMasuratoare;
let Student, Grupa;
var Xuri = [1, 2, 3, 4, 5];
let panta, intersectia, eroarea;

function preload()
{

  img1 = loadImage("Img/WavesSuport.jpg");
  img2 = loadImage("Img/Generator.jpg");
  img3 = loadImage("Img/Greutati.jpg");
  img4 = loadImage("Img/Stand.jpg");
}

function setup()
{
  createCanvas(1400, 700);
  background(220, 200, 200);
  NrMasuratori = Xuri.length;
  dist = 65;

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(300+i*dist,height-150);
    NiuC1.push(MF1);
  }
  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(300+i*dist,height-120);
    NiuE1.push(MF1);
  }
  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(300+i*dist,height-80);
    NiuC2.push(MF1);
  }
  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(300+i*dist,height-50);
    NiuE2.push(MF1);
  }
  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(690+i*dist,height-150);
    LamdaC1.push(MF1);
  }
}

```

```

}
for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Imput(690+i*dist,height-120);
  LamdaE1.push(MF1);
}
for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Imput(690+i*dist,height-80);
  LamdaC2.push(MF1);
}
for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Imput(690+i*dist,height-50);
  LamdaE1.push(MF1);
}

Masa1 = new Imput(40,height-135);
Masa2 = new Imput(40,height-65);
T1 = new Imput(105,height-135);
T2 = new Imput(105,height-65);
v1 = new Imput(170,height-135);
v2 = new Imput(170,height-65);

Student = createInput("");
Student.position(0.5*width+10, 50);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

Masa = createInput('0.445');
Masa.position(220, 265);
Masa.style('width', '40px');
Masa.style('font-size', '12px');
Masa.style('text-align', 'center');
Masa.hide();

button0 = createButton("Document");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentation);

buttonGB = createButton("Masuratori");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

rSlider = createSlider(20, 300, 25, 0.1);
rSlider.position(NiuC1[0].x, NiuC1[0].y-70);
rSlider.style('width', '500px');
rSlider.style('colour', 'red');

```

```

rSlider.hide();

}

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.73, 30, width*0.263, height*0.3);
  image(img2, width*0.73, 250, width*0.263, height*0.3);
  image(img3, width*0.73, 470, width*0.263, height*0.3);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+6);
  text("Grupa: ", Grupa.x-47, Grupa.y+6);

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizica si Chimie, Universitatea Tehnică din Cluj-
Napoca", 230, 689);

  Tabel();
  if (Masor)
  {
    image(img4, 64, 100, width*0.625, height*0.5);

    let niu = rSlider.value();
    textSize(12);
    text('niu = '+nf(niu,1,1)+' [Hz]', NiuC1[0].x, NiuC1[0].y-78);
    textSize(12);
    text('Masa [Kg]', Masa.x-70, Masa.y+5);

    text("Hz", NiuC1[0].x+215, NiuC1[0].y-180);
    textSize(20);
    text(nf(niu,1,1), NiuC1[0].x+180, NiuC1[0].y-157);

    stroke(185, 122, 87);

    strokeWeight(2);
    let g = 9.81;
    let miu = 1.25*0.001;

    let x0 = NiuC1[0].x-10;
    let y0 = NiuC1[0].y-349;
    let frectAmp = 2*PI*frameCount*rSlider.value()*0.001;
    let niu1 = 0.5*Math.sqrt(Masa.value()*g/miu);
    let niu2 = 2*niu1;
    let niu3 = 3*niu1;
    let niu4 = 4*niu1;
    let niu5 = 5*niu1;
    console.log(niu1);

    for(let i=1; i< 500; i++)
    {
      let y11 = 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu1),2))*sin(1*2*PI*0.001*(i-1));
      y11 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu2),2))*sin(2*2*PI*0.001*(i-1));
      y11 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu3),2))*sin(3*2*PI*0.001*(i-1));
      y11 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu4),2))*sin(4*2*PI*0.001*(i-1));
      y11 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu5),2))*sin(5*2*PI*0.001*(i-1));
    }
  }
}

```

```

let y12 = -20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu1),2))*sin(2*PI*0.001*i)
y12 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu2),2))*sin(2*2*PI*0.001*i);
y12 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu3),2))*sin(3*2*PI*0.001*i);
y12 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu4),2))*sin(4*2*PI*0.001*i);
y12 += 20*sin(frectAmp)*exp(-Math.pow(0.5*(niu-niu5),2))*sin(5*2*PI*0.001*i);
line(x0+i-1,y0-y11,x0+i,y0-y12);
}

}

}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laborator de Fizica - Studiul undelor stationare transversale in corzile vibrante", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.73*width-20, height-45);
  rect(0.73*width, 15,0.27*width-10, height-25);
  for (var i=0; i<16; i++)
  {
    line(NiuC1[0].x-280+i*65,NiuC1[0].y-50,NiuC1[0].x-280+i*65,height-25);
  }
  line(NiuC1[0].x-280,NiuC1[0].y-50,NiuC1[0].x+695,NiuC1[0].y-50);
  line(NiuC1[0].x-280,height-25,NiuC1[0].x+695,height-25);
  line(NiuC1[0].x-85,height-135,NiuC1[0].x+695,height-135);
  line(NiuC1[0].x-280,height-100,NiuC1[0].x+695,height-100);
  line(NiuC1[0].x-85,height-65,NiuC1[0].x+695,height-65);
  line(NiuC1[0].x-280,NiuC1[0].y-15,NiuC1[0].x+695,NiuC1[0].y-15);
  fill(0);
  noStroke();
  textSize(13);
  // Capul de tabel
  text("m [kg]",NiuC1[0].x-265,NiuC1[0].y-30);
  text("T [N]",NiuC1[0].x-195,NiuC1[0].y-30);
  text("v [m/s]",NiuC1[0].x-135,NiuC1[0].y-30);
  text("Niu [Hz]",NiuC1[0].x-75,NiuC1[0].y-30);
  text("Lamda [m]",NiuC1[0].x+308,NiuC1[0].y-30);

  textSize(9);
  text("calculat",NiuC1[0].x-80,NiuC1[0].y+5);
  text("experimental",NiuC1[0].x-80,NiuC1[0].y+35);

  text("calculat",NiuC1[0].x-80,NiuC1[0].y+70);
  text("experimental",NiuC1[0].x-80,NiuC1[0].y+105);

  text("calculat",NiuC1[0].x+308,NiuC1[0].y+5);
  text("experimental",NiuC1[0].x+308,NiuC1[0].y+35);

  text("calculat",NiuC1[0].x+308,NiuC1[0].y+70);
  text("experimental",NiuC1[0].x+308,NiuC1[0].y+105);

```

```
for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(Xuri[i],1,0),NiuC1[0].x+i*dist+7,NiuC1[0].y-30);
  text(nf(Xuri[i],1,0),NiuC1[0].x+i*dist+400,NiuC1[0].y-30);
}

}

function Documentation()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/IM+MTR/L2.pdf';
}

function FunctieButon()
{
  rSlider.show();
  Masa.show();
  Masor = true;
}

}
```

6. Determinarea raportului căldurilor molare ale gazelor

```

let img1, img2;
let Masor = false;
let ApasMasor = true;
let NrMasuratori;
let dist;
let MarimeFizica1 = [];
let MarimeFizica2 = [];
let MarimeFizica3 = [];
let MarimeFizica4 = [];
let MarimeFizica5 = [];
let Student, Grupa;
let denivelare;
let raportmediu, diferenta, eroaremedie, aprox;
let xPiston, yPiston, xRobinet2, yRobinet2;
let DxR2,DyR1,DxPiston;
let yNivel;
let AparBile,AmCrescutPresiunea,SeEgalizeazaPresiunea;
let CrestePresiuneaH2;
let AmCrescutPresiuneaH2;
let Gamma, H1, H2;
let Temperatura;
let NrApasari;
let Contor0,Contor1,timp;

let Xuri = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

// module aliases
var Engine = Matter.Engine,
    World = Matter.World,
    Bodies = Matter.Bodies,
    Body = Matter.Body,
    Constraint = Matter.Constraint;

var engine;
var world;
var boxes = [];
var circles = [];
var Aer = [];
var boundaries = [];
var stergatoare = [];
var constringeri = [];

function preload()
{
    img1 = loadImage("Img/schita.jpg");
    img2 = loadImage("Img/OmImpingator.jpg");
}

function setup()
{
    createCanvas(1400, 700);
    background(249, 240, 204);
    engine = Engine.create();
    world = engine.world;
    Engine.run(engine);
    DyR1 = 0;
    DxR2 = 0;
    DxPiston = -5;

```



```

xRobinet1 = 0.5*width-8;
yRobinet1 = 190;
xRobinet2 = 0.2*width-40;
yRobinet2 = 140;
yNivel = 0;
NrApasari=0;
AparBile = true;
AmCrescutPresiunea = false;
SeEgalizeazaPresiunea = true;
CrestePresiuneaH2 = false;
AmCrescutPresiuneaH2 = true;
timp = 0;
world.gravity.y = 0;

NrMasuratori = Xuri.length;
xPiston = 0.75*width-53;
yPiston = 180;
dist = 25;

for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Imput(5,height-(NrMasuratori)*dist+i*dist*1.1-45);
  MarimeFizica1.push(MF1);
  let MF2 = new Imput(70,height-(NrMasuratori)*dist+i*dist*1.1-45);
  MarimeFizica2.push(MF2);
  let MF3 = new Imput(135,height-(NrMasuratori)*dist+i*dist*1.1-45);
  MarimeFizica3.push(MF3);
  let MF4 = new Imput(330,height-(NrMasuratori)*dist+i*dist*1.1-45);
  MarimeFizica4.push(MF4);
  let MF5 = new Imput(395,height-(NrMasuratori)*dist+i*dist*1.1-45);
  MarimeFizica5.push(MF5);
}

raportmediu = new Imput(MarimeFizica1[0].x-105,MarimeFizica1[0].y+120," ");
diferenta = new Imput(MarimeFizica1[0].x-40,MarimeFizica1[0].y+120," ");
eroaremedie = new Imput(MarimeFizica1[0].x+155,MarimeFizica1[0].y+120," ");
aprox = new Imput(MarimeFizica1[0].x+220,MarimeFizica1[0].y+120,"");
Temperatura = new Imput(MarimeFizica1[0].x-105,MarimeFizica1[0].y-80,"23");

Student = createInput("");
Student.position(0.77*width+10,20);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+55, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentație Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '200px');
button0.clicked(Documentatie);

buttonR1 = createButton('Deschide / Închide Robinet 1');
buttonR1.position(645, 50);

```

```

buttonR1.addClass("button button-primary button-border-thick button-box");
buttonR1.style('width', '200px');
buttonR1.mouseClicked(SetR);
buttonR1.hide();

buttonR2 = createButton("Deschide / Închide Robinet 2");
buttonR2.position(25, 100);
buttonR2.addClass("button button-primary button-border-thick button-box");
buttonR2.style('width', '200px');
buttonR2.mouseClicked(SetR);
buttonR2.hide();

buttonP = createButton("Pompează aer");
buttonP.position(880, 90);
buttonP.addClass("button button-primary button-border-thick button-box");
buttonP.style('width', '120px');
buttonP.mouseClicked(SetR);
buttonP.hide();

buttonE = createButton("Experiment");
buttonE.position(button0.x+button0.width+20, button0.y);
buttonE.addClass("button button-primary button-border-thick button-box");
buttonE.style('width', '150px');
buttonE.mouseClicked(FunctieButon);

buttonStart = createButton("Reset");
buttonStart.position(buttonE.x+buttonE.width+20, buttonE.y);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '90px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

var d = createDiv();
d.style('padding-left: 50px;');
d.style('transform-origin: 10% 50% 0');
d.style('transform: rotate(' + (90) + 'deg);');
d.position(0.5*width+30, 25);
R1Slider = createSlider(-55, 0, DyR1, 1);
R1Slider.style('width', '100px');
d.child(R1Slider);
R1Slider.hide();

R2Slider = createSlider(-50, 0, DxR2, 1);
R2Slider.position(50, 125);
R2Slider.style('width', '100px');
R2Slider.style('colour', 'red');
R2Slider.hide();

PistonSlider = createSlider(-152, DxPiston+5, DxPiston, 1);
PistonSlider.position(xPiston-150, yPiston-65);
PistonSlider.style('width', '170px');
PistonSlider.style('colour', 'red');
PistonSlider.hide();

}

function draw()
{
background(249, 240, 204);
image(img1, width*0.885, 40, width*0.10, height*0.20);

```

```

fill(0);
noStroke();
textSize(12);
text("Student: ", Student.x-55,Student.y+5);
text("Grupa: ", Grupa.x-50,Grupa.y+5);
textSize(16);
text("T = ", 950,333);
text("C", 1032,333);
textSize(10);
text("o", 1028,325);
//text(mouseX+"."+mouseY,mouseX,mouseY);
if (Masor)
{

var Temp = Temperatura.MyInput.value();
Gamma = 1.4-0.000005*(Temp+273.15);
if(R1Slider.value()>-5 && R2Slider.value()<-40)
{
  if(SeEgalizeazaPresiunea)
  {
    H1 = yNivel*1;           //scala
    H2 = H1*(1-1/Gamma);
    SeEgalizeazaPresiunea = false;
    Contor0=millis();
  }
  timp=(millis()-Contor0)*0.001;
  yNivel = H1*Math.exp(-20.0*timp);
  if(yNivel<2)
  {
    CrestePresiuneaH2 = true;

  }
}
if(CrestePresiuneaH2 && R2Slider.value()>-2)
{
  if(AmCrescutPresiuneaH2)
  {
    Contor1=millis();
    AmCrescutPresiuneaH2 = false;
  }
  timp = (millis()-Contor1)*0.001;
  yNivel = H2*(1-exp(-0.5*timp));
}
DyR1 = R1Slider.value();
DxR2 = R2Slider.value();
DxPiston = PistonSlider.value();
if(DxPiston>=0 && AparBile)
{
  SaAparaBile();
}
if(DxPiston<=-152 && !AmCrescutPresiunea && R2Slider.value()>-2)
{
  SaCreascaPresiunea();
}

console.log(DxR2);
Body.setPosition(boxes[0].body, {x: 0.4*width+50, y: 455+yNivel});
Body.setPosition(boxes[1].body, {x: xPiston +DxPiston, y: 207});
Body.setPosition(boxes[2].body, {x: xPiston+105+DxPiston, y: 207});
Body.setPosition(boxes[3].body, {x: xRobinet2+ 2*DxR2 , y: yRobinet2});

```

```

Body.setPosition(boxes[4].body, {x: xRobinet1 , y: yRobinet1 + DyR1});

image(img2, xPiston+185+DxPiston, yPiston-5, width*0.05, height*0.1);
for (var i=0; i<boundaries.length; i++)
{
  boundaries[i].show();
}
fill(249, 240, 204);
rect(0.52*width,180, 100,55);
for (var i=0; i<boxes.length; i++)
{
  boxes[i].show();
}
for (var i=0; i<constringeri.length; i++)
{
  constringeri[i].show();
}

for (var i=0; i<circles.length; i++)
{
  circles[i].show();
  //circles[i].arataCM();
  if (circles[i].isOffScreen())
  {
    circles[i].RemoveFromWorld();
    circles.splice(i,1);
    i--;
  }
  px = circles[i].body.position.x;
  py = circles[i].body.position.y;
  var Vx = circles[i].body.velocity.x;
  var Vy = circles[i].body.velocity.y;
  if (abs(Vx) < 0.5)
  {
    Matter.Body.setVelocity( circles[i].body, {x: Vx*2 , y: Vy});
  }
  if (abs(Vx) > 5)
  {
    Matter.Body.setVelocity( circles[i].body, {x: Vx/2 , y: Vy});
  }
  if (abs(Vy) < 0.5)
  {
    Matter.Body.setVelocity( circles[i].body, {x: Vx , y: Vy*2});
  }
  if (abs(Vy) > 5)
  {
    Matter.Body.setVelocity( circles[i].body, {x: Vx , y: Vy*0.5});
  }

  if ((px>565 && px<800 && py <170) || (px>800 && py < 145) || ((px<150 || px> 330) && py<145))
  {
    circles[i].RemoveFromWorld();
    circles.splice(i,1);
    i--;
  }
  if ((px>615 && px<800 && py > 245) || (px>800 && py > 270))
  {
    circles[i].RemoveFromWorld();
    circles.splice(i,1);
    i--;
  }
}

```

```

}
if(py > height - 60)
{
  Matter.Body.setVelocity( circles[i].body, {x: random(-2,2) , y: random(-5,0)});
}
if(py < 160 && (px < 0.2*width-20 || px > 0.2*width+20))
{
  Matter.Body.setVelocity( circles[i].body, {x: random(-2,2) , y: random(0,5)});
}
if(py > 445+yNivel && (px > 550 && px < 615))
{
  Matter.Body.setVelocity( circles[i].body, {x: random(-1,1) , y: random(-5,-3)});
}
if(px < 30)
{
  Matter.Body.setVelocity( circles[i].body, {x: random(0,5) , y: random(-2,2)});
}
if((px > 0.4*width-10 && px < 0.4*width+10) && (py < 180 || py > 240))
{
  Matter.Body.setVelocity( circles[i].body, {x: random(-5,0) , y: random(-2,2)});
}
if(px > xPiston+DxPiston-20)
{
  Matter.Body.setVelocity( circles[i].body, {x: random(-5,0) , y: random(-1,1)});
}
if((py < 480 || py > 430) && (px > 0.4*width+115 && px < 0.4*width+170) )
{
  //Matter.Body.setVelocity( circles[i].body, {x: random(-5,2) , y: random(-5,0)});
}
//console.log(circles.length, world.bodies.length);
}
for (var i=0; i<Aer.length; i++)
{
  Aer[i].show();
  //circles[i].arataCM();
  if (Aer[i].isOffScreen())
  {
    Body.setPosition(Aer[i].body, {x: 750, y: 215});
  }
  px = Aer[i].body.position.x;
  py = Aer[i].body.position.y;
  var Vx = Aer[i].body.velocity.x;
  var Vy = Aer[i].body.velocity.y;
  if (abs(Vx) < 0.5)
  {
    Matter.Body.setVelocity( Aer[i].body, {x: Vx*2 , y: Vy});
  }
  if (abs(Vx) > 5)
  {
    Matter.Body.setVelocity( Aer[i].body, {x: Vx/2 , y: Vy});
  }
  if (abs(Vy) < 0.5)
  {
    Matter.Body.setVelocity( Aer[i].body, {x: Vx , y: Vy*2});
  }
  if (abs(Vy) > 5)
  {
    Matter.Body.setVelocity( Aer[i].body, {x: Vx , y: Vy*0.5});
  }
}

```

```

if ((px>565 && px<800 && py <170) || (px>800 && py < 145)|| (px> xPiston+DxPiston))
{
  Matter.Body.setPosition(Aer[i].body, {x: 750, y: 215});
} else if ((px>615 && px<800 && py > 245) || (px>800 && py > 270))
{
  Matter.Body.setPosition(Aer[i].body, {x: 750, y: 215});
}
if(py> height - 60)
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-2,2) , y: random(-5,0)});
}
if(py< 160 && (px< 0.2*width-20 || px> 0.2*width-30))
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-5,5) , y: random(-2,2)});
}
if(py> 445+yNivel && (px> 550 && px< 615))
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-1,1) , y: random(-5,-3)});
}
if(px < 30)
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-5,5) , y: random(-2,2)});
}
if((px> 0.6*width-10 && px<0.6*width+10) && (py<180 || py> 200))
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-5,0) , y: random(-2,2)});
}
if(px> xPiston+DxPiston-20)
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-2,0) , y: random(-2,0)});
}
if((px > 0.75*width-20 && px<0.75*width+10) && (py<160 && py> 167))
{
  Matter.Body.setVelocity( Aer[i].body, {x: random(-2,0) , y: random(-2,0)});
}

//console.log(circles.length, world.bodies.length);
}
//circles[50].arataViteza();
stroke(0);
strokeWeight(2);
line(0.4*width+34, 246, 0.4*width+34, 610);
fill (250, 0, 0);
ellipse(0.4*width+109, 610, 150, 150);
fill (249, 240, 204);
ellipse(0.4*width+109, 610, 110, 110);
noStroke();
rect (0.4*width+35, 534, 200, 76);
stroke(0);
strokeWeight(2);
line(0.4*width+54, 246, 0.4*width+54, 610);
line(0.4*width+184, 266, 0.4*width+184, 610);
line(0.4*width+164, 266, 0.4*width+164, 610);
fill (249, 240, 204);
ellipse(0.4*width+174, 266, 20, 20);
noStroke();
rect (0.4*width+165, 266, 18, 25);
fill (250, 0, 0);
rect (0.4*width+35, 450+yNivel, 18, 170-yNivel);
rect (0.4*width+165, 450-yNivel, 18, 170+yNivel);

```

```

    Scala(1);
    stroke(0,250,0);
    strokeWeight(1);
    line(0.4*width+35, 450+yNivel,0.4*width+135, 450+yNivel);
    line(0.4*width+85, 450-yNivel,0.4*width+185, 450-yNivel);
  }

  Tabel();
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de Fizică: Determinarea raportului căldurilor molare ale gazelor", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.995*width-15, height-45);
  //rect(0.7*width, 15,0.3*width-10, height-25);
  for (var i=0; i<10; i++)
  {
    line(MarimeFizica1[0].x+470+i*65,MarimeFizica1[0].y-60,MarimeFizica1[0].x+470+i*65,height-25);
  }
  line(MarimeFizica1[0].x+470,MarimeFizica1[0].y-60,MarimeFizica1[0].x+1055,MarimeFizica1[0].y-60);
  line(MarimeFizica1[0].x+470,height-25,MarimeFizica1[0].x+1055,height-25);
  line(MarimeFizica1[0].x+470,MarimeFizica1[0].y-20,MarimeFizica1[0].x+1055,MarimeFizica1[0].y-20);

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
  Napoca",795, 688);

  fill(0);
  noStroke();
  textSize(12);
  // Capul de tabel
  text("h",MarimeFizica1[0].x+497,MarimeFizica1[0].y-40);
  text("l",MarimeFizica1[0].x+505,MarimeFizica1[0].y-35);
  text("[cm]",MarimeFizica1[0].x+492,MarimeFizica1[0].y-25);
  text("h",MarimeFizica2[0].x+497,MarimeFizica2[0].y-40);
  text("2",MarimeFizica2[0].x+505,MarimeFizica2[0].y-35);
  text("[cm]",MarimeFizica2[0].x+492,MarimeFizica2[0].y-25);
  text("γ",MarimeFizica3[0].x+500,MarimeFizica3[0].y-40);
  text("γ",MarimeFizica3[0].x+565,MarimeFizica3[0].y-40);
  text("_",MarimeFizica3[0].x+565,MarimeFizica3[0].y-50);
  text("Δh",MarimeFizica3[0].x+627,MarimeFizica3[0].y-40);
  text("[cm]",MarimeFizica3[0].x+623,MarimeFizica3[0].y-25);
  text("Δγ/γ",MarimeFizica4[0].x+490,MarimeFizica4[0].y-40);
  text("[%]",MarimeFizica4[0].x+492,MarimeFizica4[0].y-25);
  text("Δγ",MarimeFizica5[0].x+490,MarimeFizica5[0].y-40);
  text("Δγ",MarimeFizica5[0].x+561,MarimeFizica5[0].y-40);

```

```

text("_",MarimeFizica5[0].x+568,MarimeFizica5[0].y-50);
text("γ ±",MarimeFizica5[0].x+618,MarimeFizica5[0].y-40);
text("_",MarimeFizica5[0].x+618,MarimeFizica5[0].y-50);
text(" Δγ",MarimeFizica5[0].x+628,MarimeFizica5[0].y-40);
text("_",MarimeFizica5[0].x+642,MarimeFizica5[0].y-50);

for (let i = 0; i<NrMasuratori; i++)
{
  //text(MarimeFizica1[0].x-65,MarimeFizica1[0].y+i*dist+7);
}

}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/SIM/2019-2020/Laboratoare-noi/L5_Determinarea%20raportului%20caldurilor%20molare%20ale%20gazelor.pdf';
}

function FunctieButon()
{
  Masor = true;
  if(ApasMasor)
  {
    R1Slider.show();
    R2Slider.show();
    PistonSlider.show();
    buttonR1.show();
    buttonR2.show();
    buttonP.show();
    buttonStart.show();
    var PereteCutieSt1 = new Boundary(20,0.5*height+70,10, 530, 0, 220, 186, 166);
    var PereteCutieDr1 = new Boundary(0.4*width,0.5*height+110,10, 450, 0, 220, 186, 166);
    var PereteCutieDr2 = new Boundary(0.4*width,165,10, 30, 0, 220, 186, 166);
    var FundCutie = new Boundary(0.25*width-60,height-50,540, 10, 0, 220, 186, 166);
    var CapacCutie1 = new Boundary(0.125*width-35,150,250, 10, 0, 220, 186, 166);
    var CapacCutie2 = new Boundary(0.25*width+85,150,260, 10, 0, 220, 186, 166);
    var SuportDopCutie = new Boundary(0.2*width-40,130,50, 10, 0,220, 186, 166);
    var FurtunSus1 = new Boundary(0.5*width-80,175,130, 10, 0, 160, 112, 84);
    var FurtunSus2 = new Boundary(0.5*width+50,175,110, 10, 0, 160, 112, 84);
    var FurtunInSus1 = new Boundary(0.5*width-18,155,8, 30, 0, 160, 112, 84);
    var FurtunInSus2 = new Boundary(0.5*width+2,155,8, 30, 0, 160, 112, 84);
    var FurtunJos1 = new Boundary(0.4*width+15,240,40, 10, 0, 160, 112, 84);
    var FurtunJos2 = new Boundary(0.5*width+8,240,190, 10, 0, 160, 112, 84);
    var FurtunInJos1 = new Boundary(0.4*width+30,445,10, 400, 0, 249, 240, 204);
    var FurtunInJos2 = new Boundary(0.4*width+60,445,10, 400, 0, 249, 240, 204);

    var SiringaPerStg1 = new Boundary(0.6*width-29,167,10, 50, 45, 165, 210, 225);
    var SiringaPerStg2 = new Boundary(0.6*width-29,167+80,10, 50, -45, 165, 210, 225);
    var SiringaPerSus = new Boundary(0.7*width-55,151,200, 10, 0, 165, 210, 225);
    var SiringaPerJos = new Boundary(0.7*width-55,151+112,200, 10, 0, 165, 210, 225);
    var SiringaPerDr1 = new Boundary(0.75*width-30,172,10, 40, 0, 165, 210, 225);
    var SiringaPerDr2 = new Boundary(0.75*width-30,172+70,10, 40, 0, 165, 210, 225);
    var SiringaSuportPerDrSus = new Boundary(0.75*width-15,172+52, 40, 10, 0, 165, 210, 225);
    var SiringaSuportPerDrJos = new Boundary(0.75*width-15,172+18, 40, 10, 0, 165, 210, 225);

    boundaries.push(PereteCutieSt1);
    boundaries.push(PereteCutieDr1);
    boundaries.push(CapacCutie1);
    boundaries.push(CapacCutie2);
  }
}

```



```

boundaries.push(SuportDopCutie);
boundaries.push(PereteCutieDr2);
boundaries.push(FurtunSus1);
boundaries.push(FurtunSus2);
boundaries.push(FurtunJos1);
boundaries.push(FurtunJos2);
boundaries.push(FurtunInJos1);
boundaries.push(FurtunInJos2);
boundaries.push(FundCutie);
boundaries.push(SiringaPerStg1);
boundaries.push(SiringaPerStg2);
boundaries.push(SiringaPerSus);
boundaries.push(SiringaPerJos);
boundaries.push(SiringaPerDr1);
boundaries.push(SiringaPerDr2);
boundaries.push(FurtunInSus1);
boundaries.push(FurtunInSus2);
boundaries.push(SiringaSuportPerDrSus);
boundaries.push(SiringaSuportPerDrJos);

var limitaManometruSt = new Box(0.4*width+50,440,60, 10, 0, 249, 240, 204);
var Piston = new Box(xPiston,172+35,35, 102, 0, 60, 112, 184);
var tijaPiston = new Box(xPiston+250,172+35,180, 25, 0, 60, 112, 84);
var Robinet1 = new Box(xRobinet1,yRobinet1,15, 90, 0, 60, 112, 84);
var Robinet2 = new Box(xRobinet2,yRobinet2,180, 10, 0, 60, 112, 84);
Body.setDensity(Robinet1, 200);
Body.setDensity(Piston, 4000);
Body.setDensity(tijaPiston, 4000);

boxes.push(limitaManometruSt);
boxes.push(Piston);
boxes.push(tijaPiston);
boxes.push(Robinet2);
boxes.push(Robinet1);

//var legatura = new Legaturi(boxes[1].body, boxes[2].body, 1, 15,0,0,-80,0);
//constringeri.push(legatura);

for(var i=0;i<300;i++)
{
    var bila = new Circle(0.25*width+random(-300,205), 0.5*height+random(-190,250), 4, 50,50,250);
    Matter.Body.setVelocity( bila.body, {x: random(-5,5) , y: random(-5,5)});
    circles.push(bila);
}
}
ApasMasor = false;
}

function SaAparaBile()
{
    AparBile = false;
    AmCrescutPresiunea = false;
    for(var i=0;i<20;i++)
    {
        var bila = new Circle(0.65*width+random(-50,50), yPiston+25+random(-20,20), 4, 230,100,200);
        Matter.Body.setVelocity( bila.body, {x: random(-5,5) , y: random(-5,5)});
        Aer.push(bila);
    }
}
}

```

```

function SaCreascaPresiunea()
{
  if (R1Slider.value()<-30)
  {
    AmCrescutPresiunea = true;
    AparBile = true;
    yNivel += 5;
  }
}

function Scala(yCorr)
{
  var x = 642;
  var y = 470;
  dy0 = 6;
  dy1 = 5*dy0;
  dy2 = 10*dy0;
  let salt = -20;
  strokeWeight(1);
  //stroke(220,0,220);
  //line(100,0.5*height+yCorr,200,0.5*height+yCorr);
  stroke(0,0,220);
  for(let i=-30; i<30; i++)
  {
    line(x,y+yCorr-i*dy0+salt,x+28,y+yCorr-i*dy0+salt);
  }

  for(let i=-6; i<6; i++)
  {
    line(x,y+yCorr-i*dy1+salt,x+38,y+yCorr-i*dy1+salt);
  }
  strokeWeight(2);
  for(let i=-3; i<=3; i++)
  {
    line(x,y+yCorr-i*dy2+salt,x+48,y+yCorr-i*dy2+salt);
  }
  noStroke();
  fill(0);
  for(let i=-3; i<=3; i++)
  {
    text(nf(10*i,1,0),x+53,y+yCorr-i*dy2+3+salt);
  }
}

function SetR()
{
  Robinet = 1;
  console.log(robinet);
}

function Start()
{
  for (var i=Aer.length-1; i>=0; i--)
  {
    Aer[i].RemoveFromWorld();
    Aer.splice(i,1);
  }
  var NAer = circles.length;
  for(var i=NAer;i<300;i++)
  {
    var bila = new Circle(0.25*width+random(-300,205), 0.5*height+random(-190,250), 4, 50,50,250);
  }
}

```

```
Matter.Body.setVelocity( bila.body, {x: random(-5,5), y: random(-5,5)});
circles.push(bila);
}

DyR1 = 0;
DxR2 = 0;
DxPiston = -5;
yNivel = 0;
NrApasari=0;
AparBile = true;
AmCrescutPresiunea = false;
SeEgalizeazaPresiunea = true;
CrestePresiuneaH2 = false;
AmCrescutPresiuneaH2 = true;
timp = 0;
Masor = true;

}
```

7. Studiul unei pompe de căldură

```

let img1, img2, img3, img4, img5;
let ImgArataGrafic, ImgArataGrafic1;
var Contor0, Contor1;
let CresteTemp = false;
let Masor = false;
let NrMasuratori, dist;
let timp = [];
let T1mas = [];
let T2mas = [];
let Tmas = [];
let theta;
let lanbda = 205;
let Temp;
let Bule = [];
let Apa = [];
let ModIncalzitor;
let TempUlei, TempAmb;
let Putere;
let PutereIncalzitor = 862.5;
let PuterePompaC = 179.25;
let CitireContor;
let EnergieConsumata;
let VariatieTemp, DeltaT;
let CapacitateCalUlei;
var Xuri = [1, 2, 3, 4, 5, 6];

let panta, intersectia, eroarea;

let TipMasuratoare;
let Student, Grupa;

function ExpT2(t,r)
{
  return r+60+0.03382*t-0.0000771605*t*t+0.0000000857339*t*t*t;
}

function ExpT1(t,r)
{
  return r+40+0.02134*t-0.0000170855*t*t+0.0000000107167*t*t*t;
}

function ExpT(t,r)
{
  return r+20+0.01503*t-0.00000409226*t*t;
}

function preload()
{
  img1 = loadImage("Img/pompa de caldura.jpg");
  img2 = loadImage("Img/contor.jpg");
  img3 = loadImage("Img/Termoplonjon.gif");
  img4 = loadImage("Img/Bilant energetic.jpg");
  img5 = loadImage("Img/Formule.jpg");
}

```

```

function setup()
{
  createCanvas(1300, 700);
  background(200, 200, 200);
  NrMasuratori = Xuri.length;
  CitireContor = round(random(100000,100000000));
  console.log(CitireContor);
  CitireContor *= 0.0001;
  EnergieConsumata = 0;
  VariatieTemp = 0;
  DeltaT = 0;
  Contor1 = millis();
  CapacitateCalUlei = 0.0071*(1+0.02*random(-1,1));
  TempAmb = 25;
  TempUlei = TempAmb;
  ModIncalzitor = true;
  Putere = PutereIncalzitor;
  lanbda = 205+10*random(-1,1);
  theta = 1;
  dist = 25;

  Student = createInput("");
  Student.position(0.01*width+70, 48);
  Student.style('width', '150px');
  Student.style('font-size', '12px');
  Student.style('text-align', 'center');

  Grupa = createInput("");
  Grupa.position(Student.x+Student.width+60, Student.y);
  Grupa.style('width', '50px');
  Grupa.style('font-size', '12px');
  Grupa.style('text-align', 'center');
  /*******imputuri */
  let MF1 = new Input(100,height-(NrMasuratori-0.5)*dist+10,"");
  timp.push(MF1);

  ContorIncalzitorF = new Input(timp[0].x,timp[0].y+(dist+5), "");
  ContorIncalzitor = new Input(timp[0].x,timp[0].y+2*(dist+5), "");
  TemperaturaI_IncalzitorI = new Input(timp[0].x+70,timp[0].y+0*(dist+5), "");
  TemperaturaI_IncalzitorF = new Input(timp[0].x+70,timp[0].y+1*(dist+5), "");
  TemperaturaI_Incalzitor = new Input(timp[0].x+70,timp[0].y+2*(dist+5), "");
  Putere_Incalzitor = new Input(timp[0].x+70,timp[0].y+3*(dist+4), "");
  ContorPompaI = new Input(timp[0].x+205,timp[0].y, "");
  ContorPompaF = new Input(timp[0].x+205,timp[0].y+(dist+5), "");
  ContorPompa = new Input(timp[0].x+205,timp[0].y+2*(dist+5), "");
  TemperaturaI_PompaI = new Input(timp[0].x+287,timp[0].y, "");
  TemperaturaI_PompaF = new Input(timp[0].x+287,timp[0].y+(dist+5), "");
  TemperaturaI_Pompa = new Input(timp[0].x+287,timp[0].y+2*(dist+5), "");
  Putere_Pompa = new Input(timp[0].x+287,timp[0].y+3*(dist+4), "");
  /*******butoane */
  button0 = createButton("Document");
  button0.position(30, 80);
  button0.addClass("button button-primary button-border-thick button-box");
  button0.style('width', '150px');
  button0.mouseClicked(Documentatie);

  buttonGB = createButton("Masuratori");
  buttonGB.position(button0.x+button0.width+20, button0.y);
  buttonGB.addClass("button button-primary button-border-thick button-box");

```

```

buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

buttonStart = createButton("Start");
buttonStart.position(button0.x+700, button0.y+50);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+30, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

buttonIncalzitor = createButton("Incalzitor");
buttonIncalzitor.position(button0.x+700, button0.y+80);
buttonIncalzitor.addClass("button button-primary button-border-thick button-box");
buttonIncalzitor.style('width', '100px');
buttonIncalzitor.mouseClicked(Incalzitor);
buttonIncalzitor.hide();

buttonPompa = createButton("Pompa");
buttonPompa.position(button0.x+830, button0.y+80);
buttonPompa.addClass("button button-primary button-border-thick button-box");
buttonPompa.style('width', '100px');
buttonPompa.mouseClicked(Pompa);
buttonPompa.hide();
}

function draw()
{
  background(241, 227, 228);

  Tabel();

  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);
  textSize(16);
  text("motor termic", 120, 450);
  text("frigider sau pompa de caldura", 280, 450);
  image(img4, width*0.04, 100, width*0.35, height*0.4);
  image(img5, width*0.04, 390, width*0.35, height*0.13);

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizica si Chimie, Universitatea Tehnică din Cluj-
Napoca", 650, 689);

  if(Masor)
  {
    let Timp = 0;

```

```

image(img1, width*0.460, 80, width*0.50, height*0.85);
fill (178,178,178);
noStroke();
rect(0.58*width, 0.5*height, 102, 50);
fill (0);
textSize(10);
text("kWh ",0.58*width+75, 0.5*height+45);
if (ModIncalzitor)
{
  image(img3, width*0.8, 250, width*0.1, height*0.25);
}
if(CresteTemp)
{
  Timp = (millis()-Contor0)*0.001;
  EnergieConsumata = 0.000001*Putere*Timp/3.6; // in kWh
  VariatieTemp = EnergieConsumata/CapacitateCalUlei;
  if(Timp > 500)
  {
    CresteTemp = false;
  }
}
else
{
  let TimpRacire = (millis()-Contor1)*0.001;
  VariatieTemp = DeltaT*exp(-TimpRacire*0.001);
}
TempUlei = TempAmb + VariatieTemp;

fill(0);
noStroke();
textSize(18);
text(nf(CitireContor+EnergieConsumata,4,4),0.6*width-18,0.5*height+30);
fill(0,0,120,100);
stroke(0);
strokeWeight(1);
// rect(0.5*width,0.556*height+6,5.2,185);
noStroke();
fill(220,0,0);
rect(0.76*width-1,0.46*height-(TempUlei-25)*5+82,6,105+(TempUlei-25)*5);
stroke(0);
strokeWeight(6);
line(0.755*width, 352, 0.9*width, 350);
textSize(16);
noStroke();
fill(220,10,10);
text(" "+nf(TempUlei,1,2)+" °C",0.76*width-75,0.46*height-(TempUlei-25)*5+82);
}
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laborator de Fizica: Studiul unei pompe de caldura", 25, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
}

```

```

stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.42*width-20, height-45);
rect(0.42*width, 15,0.58*width-10, height-25);

line(timp[0].x-85+0.001*65,timp[0].y-80,timp[0].x-85+0.001*65,height-25);
line(timp[0].x-85+1*65,timp[0].y-80,timp[0].x-85+1*65,height-25);
line(timp[0].x-85+2*65,timp[0].y-50,timp[0].x-85+2*65,height-53);
line(timp[0].x-85+3.35*65,timp[0].y-50,timp[0].x-85+3.35*65,height-53);
line(timp[0].x-85+4.2*65,timp[0].y-80,timp[0].x-85+4.2*65,height-25);
line(timp[0].x-85+5.2*65,timp[0].y-50,timp[0].x-85+5.2*65,height-53);
line(timp[0].x-85+6.7*65,timp[0].y-50,timp[0].x-85+6.7*65,height-53);
line(timp[0].x-85+7.7*65,timp[0].y-80,timp[0].x-85+7.7*65,height-25);

line(timp[0].x-85,timp[0].y-80,timp[0].x+415,timp[0].y-80);
line(timp[0].x-85,timp[0].y-50,timp[0].x+415,timp[0].y-50);

line(timp[0].x-85,height-25,timp[0].x+415,height-25);
line(timp[0].x-85,timp[0].y-15,timp[0].x+415,timp[0].y-15);
line(timp[0].x-85,timp[0].y+15,timp[0].x+415,timp[0].y+15);
line(timp[0].x-85,timp[0].y+45,timp[0].x+415,timp[0].y+45);
line(timp[0].x-85,timp[0].y+75,timp[0].x+415,timp[0].y+75);

fill(0);
noStroke();
textSize(14);
// Capul de tabel
text("Tabel",30,timp[0].y-60);
text("Incalzitor electric",135,timp[0].y-60);
text("Pompa de caldura",345,timp[0].y-60);

text("Citire",timp[0].x-68,timp[0].y-30);
text("Initial",timp[0].x-68,timp[0].y+5);
text("Final",timp[0].x-68,timp[0].y+35);
text("Valori",timp[0].x-68,timp[0].y+65);
text("Puterea",timp[0].x-75,timp[0].y+95);

text("Contor",timp[0].x-5,timp[0].y-35);
text("[kWh]",timp[0].x-2,timp[0].y-20);

text("Temperatura",timp[0].x+50,timp[0].y-35);
text("[°C]",timp[0].x+75,timp[0].y-20);

text("timp",timp[0].x+2.3*65,timp[0].y-35);
text("[s]",timp[0].x+2.4*65,timp[0].y-20);
text("0",timp[0].x+2.4*65,timp[0].y+10);
text("4x60",timp[0].x+2.2*65,timp[0].y+40);
text("t = 240",timp[0].x+2.1*65,timp[0].y+65);

text("Contor",timp[0].x+6+3*65,timp[0].y-35);
text("[kWh]",timp[0].x+3.15*65,timp[0].y-20);
text("Temperatura",timp[0].x+4.1*65,timp[0].y-35);
text("[°C]",timp[0].x+4.5*65,timp[0].y-20);
text("timp",timp[0].x+5.7*65,timp[0].y-35);
text("[s]",timp[0].x+5.8*65,timp[0].y-20);
text("0",timp[0].x+5.8*65,timp[0].y+10);
text("8x60",timp[0].x+5.7*65,timp[0].y+40);
text("t = 480",timp[0].x+5.6*65,timp[0].y+65);

```



```
    textSize(12);
    text("0",timp[0].x+2.15*65,timp[0].y+70);
}

function Documentatie()
{
    window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2019-2020/lucrari/Semi/12.%20Studiul%20unei%20pompe%20de%20caldura.pdf';
}

function FunctieButon()
{
    Masor = true;
    buttonStart.show();
    buttonStop.show();
    buttonIncalzitor.show();
    buttonPompa.show();
    //Temp = temperatura[0];
}

function Start()
{
    CresteTemp = true;
    Contor0 = millis();
    EnergieConsumata = 0;
}

function Stop()
{
    CresteTemp = false;
    CitireContor += EnergieConsumata;
    EnergieConsumata = 0;
    Contor1 = millis();
    DeltaT=VariatieTemp;
}

function Incalzitor()
{
    ModIncalzitor = true;
    Putere = PutereIncalzitor;
}

function Pompa()
{
    ModIncalzitor = false;
    Putere = PuterePompaC;
}
```

8. Determinarea coeficientului de vâscozitate al lichidelor

```
let img1, img2, img3;
let Masor = false;
let cade = false;
let NrMasuratori, dist;
let MarimeFizica1 = [];
let MarimeFizica2 = [];
var Xuri = [800,895];
let timpMediu1, timpMediu2;
let viteza01, viteza02;
let rho01, rho02;
let xBila, yBila, inaltime;
let raza;
var Contor0, Contor;
let densUlei;
let vitBila;
let coefVasc;
let Student, Grupa;

let TipMasuratoare;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};

function preload()
{
  img1 = loadImage("Img/SuportVascozitate.jpg");
  img2 = loadImage("Img/Vascozitate.jpg");
  img3 = loadImage("Img/Vascozitate2.jpg");
  img4 = loadImage("Img/Vascozimetru.jpg");
}

function setup()
{
  createCanvas(1650, 700);
  background(220, 200, 200);

  NrMasuratori = 5;
  xBila = 0.651*width;
  yBila = 0.3*height;
  inaltime = 0;
  dist = 25;
  Contor0 = 0;
```

```

Contor = 0;

Student = createInput("");
Student.position(0.5*width+63, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Input(290,0.8*height-(NrMasuratori+1)*dist+i*dist+5);
  MarimeFizica1.push(MF1);
}

for (let i=0; i<NrMasuratori; i++)
{
  let MF2 = new Input(290,height-(NrMasuratori+1)*dist+i*dist+5,"");
  MarimeFizica2.push(MF2);
}

raza = new Input(MarimeFizica1[2].x-130,MarimeFizica1[2].y+70,"");
//raza.MyInput.html() = 10;

timpMediu1 = new Input(MarimeFizica1[2].x+65,MarimeFizica1[2].y,"");
timpMediu2 = new Input(MarimeFizica2[2].x+65,MarimeFizica2[2].y,"");
viteza01 = new Input(timpMediu1.x+65,timpMediu1.y,"");
viteza02 = new Input(timpMediu2.x+65,timpMediu2.y,"");
rho01 = new Input(timpMediu1.x-325,timpMediu1.y,"");
rho02 = new Input(timpMediu2.x-325,timpMediu2.y,"");
eta01 = new Input(timpMediu1.x+135,timpMediu1.y,"");
eta02 = new Input(timpMediu2.x+135,timpMediu2.y,"");
miu01 = new Input(timpMediu1.x+200,timpMediu1.y,"");
miu02 = new Input(timpMediu2.x+200,timpMediu2.y,"");
deltaR = new Input(timpMediu1.x+265,timpMediu1.y+70,"");
deltaH = new Input(timpMediu2.x+330,timpMediu1.y+70,"");
deltaT = new Input(timpMediu2.x+390,timpMediu1.y+70,"");
deltaeta01 = new Input(timpMediu1.x+455,timpMediu1.y,"");
deltaeta02 = new Input(timpMediu2.x+455,timpMediu2.y,"");
densitBila = new Input(MarimeFizica2[2].x-190,MarimeFizica1[4].y+20,"");

button0 = createButton("Documentatie Laborator");
button0.position(width*0.02, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.clicked(Documentatie);

```

```

buttonGB = createButton("Măsor");
buttonGB.position(890, 50);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

buttonVR = createButton("Vascozimetru");
buttonVR.position(button0.x+button0.width+75, button0.y+10);
buttonVR.addClass("button button-primary button-border-thick button-box");
buttonVR.style('width', '150px');
buttonVR.mouseClicked(ButonVR);

buttonStart = createButton("Start");
buttonStart.position(buttonGB.x, buttonGB.y+30);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Reset");
buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

rSlider = createSlider(800, 950, 800, 0.1);
rSlider.position(buttonGB.x, height-30);
rSlider.style('width', '350px');
rSlider.style('colour', 'red');
rSlider.hide();

}

function draw()
{
  let A =-93.4666;
  let B = 0.1198;

  background(220, 200, 200);
  image(img1, width*0.78, 40, width*0.2, height*0.9);
  image(img4, width*0.15, 50, width*0.3, height*0.4);

  //image(img3, 1600, 150, width*0.1, height*0.75);
  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55,Student.y+5);
  text("Grupa: ", Grupa.x-50,Grupa.y+5);

```

```

Tabel();
if(Masor)
{
  image(img2, buttonGB.x+100, 100, width*0.1, height*0.75);
  //text(nf(mouseY,1,1),mouseX,mouseY);
  fill(200, 0, 0);
  let razaB = raza.MyInput.value()*4;
  densUlei = rSlider.value();
  coefVasc = A + B*densUlei;
  vitBila = 2.0/9.0*9.81*(980-densUlei)*razaB*razaB*0.000001/coefVasc;
  //console.log(vitBila);
  fill(0);
  text('ρ_ulei = '+nf(densUlei,1,1)+ " kg/m^3",rSlider.x,rSlider.y-15);
  //text('coefV = '+nf(coefVasc,1,3),rSlider.x+200,rSlider.y-15);

  if(cade)
  {
    Contor = millis();
    inaltime = (Contor-Contor0)*vitBila/1.56;
  }
  if (inaltime>385-0.5*razaB)
  {
    cade = false;
  }
  ellipse(xBila, yBila+inaltime,int(razaB), int(razaB));

  let culoare = map(densUlei,800,950,50,225);

  fill(227,191,95,culoare);
  rect(xBila-30,yBila-9,68,height*0.56);

  let senzor1 = 260; //522
  if(abs(yBila+inaltime-senzor1)<5)
  {
    stroke(240,0,0);
    line(xBila-30,senzor1,xBila+35,senzor1);
  }
  let senzor2 = 522;
  if(abs(yBila+inaltime-senzor2)<5)
  {
    stroke(240,0,0);
    line(xBila-30,senzor2,xBila+35,senzor2);
  }
}
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);

```

```

noStroke();
text("Laboratorul de Fizica:Determinarea coeficientului de vascozitate al lichidelor ", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.76*width-20, height-45);
rect(0.76*width, 15,0.24*width-10, height-25);
for (var i=0; i<14; i++)
{
  line(MarimeFizical[0].x-275+i*65,MarimeFizical[0].y-50,MarimeFizical[0].x-275+i*65,height-25);
}
line(MarimeFizical[0].x-275,MarimeFizical[0].y-50,MarimeFizical[0].x+570,MarimeFizical[0].y-50);
line(MarimeFizical[0].x-275,height-25,MarimeFizical[0].x+570,height-25);
line(MarimeFizical[0].x-275,MarimeFizical[0].y-15,MarimeFizical[0].x+570,MarimeFizical[0].y-15);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",300, 690);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("p ulei",MarimeFizical[0].x-265,MarimeFizical[0].y-35);
text("p bila",MarimeFizical[0].x-265+65,MarimeFizical[0].y-35);
text("[Kg/m ]",MarimeFizical[0].x-265,MarimeFizical[0].y-20);
text("[Kg/m ]",MarimeFizical[0].x-265+65,MarimeFizical[0].y-20);
text("r",MarimeFizical[0].x-250+2*65,MarimeFizical[0].y-35);
text("[mm]",MarimeFizical[0].x-260+2*65,MarimeFizical[0].y-20);
text("h",MarimeFizical[0].x-245+3*65,MarimeFizical[0].y-35);
text("[m]",MarimeFizical[0].x-250+3*65,MarimeFizical[0].y-20);
text("0.4",MarimeFizical[0].x-250+3*65,MarimeFizical[4].y+30);
text("t",MarimeFizical[0].x-250+4*65,MarimeFizical[0].y-35);
text("[s]",MarimeFizical[0].x-255+4*65,MarimeFizical[0].y-20);
text("t mediu",MarimeFizical[0].x-256+5*65,MarimeFizical[0].y-35);
text("[s]",MarimeFizical[0].x-250+5*65,MarimeFizical[0].y-20);
text("v",MarimeFizical[0].x-245+6*65,MarimeFizical[0].y-35);
text("[m/s]",MarimeFizical[0].x-190+5*65,MarimeFizical[0].y-20);
text("η",MarimeFizical[0].x-190+6*65,MarimeFizical[0].y-35);
text("[Ns/m ]",MarimeFizical[0].x-205+6*65,MarimeFizical[0].y-20);
text("g",MarimeFizical[0].x-190+7*65,MarimeFizical[0].y-35);
text("[m /s]",MarimeFizical[0].x-198+7*65,MarimeFizical[0].y-20);
text("Δ r",MarimeFizical[0].x-190+8*65,MarimeFizical[0].y-35);
text("[m]",MarimeFizical[0].x-188+8*65,MarimeFizical[0].y-20);
text("Δ h",MarimeFizical[0].x-190+9*65,MarimeFizical[0].y-35);
text("[m]",MarimeFizical[0].x-185+9*65,MarimeFizical[0].y-20);

```

```

text("Δ t",MarimeFizica1[0].x-190+10*65,MarimeFizica1[0].y-35);
text("[s]",MarimeFizica1[0].x-185+10*65,MarimeFizica1[0].y-20);
text("Δη/η",MarimeFizica1[0].x-186+11*64,MarimeFizica1[0].y-35);
text("[%]",MarimeFizica1[0].x-190+11*65,MarimeFizica1[0].y-20);

//text(nf(Placute, 1,0)+" placute",MarimeFizica1[0].x-80,MarimeFizica1[0].y-80);
//text("x = "+nf(0.5*Placute, 1,1)+" [mm]",MarimeFizica1[0].x-80,MarimeFizica1[0].y-60);

for (let i = 0; i<Xuri.length; i++)
{
  text(nf(Xuri[i],1,1),MarimeFizica1[3].x-255,MarimeFizica1[2].y+i*dist*5.6+7);
}
textSize(9);
text("3",MarimeFizica1[0].x-234,MarimeFizica1[0].y-25);
text("3",MarimeFizica1[0].x-166,MarimeFizica1[0].y-25);
text("0",MarimeFizica1[0].x+150,MarimeFizica1[0].y-29);
text("2",MarimeFizica1[0].x+215,MarimeFizica1[0].y-25);
text("2",MarimeFizica1[0].x+270,MarimeFizica1[0].y-25);

}
function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2019-2020/lucrari/SemI/L15Ro_Coeficientul%20de%20vascozitate.pdf';
}

function ButonVR()
{
  window.location='https://www.pce-instruments.com/english/slot/2/download/358232/manual-viscometer-pce-rvi2.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
  rSlider.show();
}

function Start()
{
  cade = true;
  Contor = Contor0 = millis();
}

function Stop()
{
  cade = false;
  inaltime = 0;
}

```

9. Determinarea coeficientului de vâscozitate dinamică a aerului

```

let img1,img2,img3, img4;
let Masor = false;
let NrMasuratori, dist;
let MasuratoareNr;
let MarimeFizical = [];
let Volume = [100, 200, 300, 400, 500, 600, 700, 800];
let razaCapilar, lungCapilar, rhoApa, gCluj,constK, vascozitateAer;
let DELTAH, eroareH,Hmax,NivelApa;
let Robinet;
let volum = [];
let timpT0, timpT1,timpT2,timpT3;
let denivelare = [];
let vascozitate = [];
let Apa = [];
let temperatura;
let vascozitatemedie;
let fluid;
let Student, Grupa;
let yNivel, yNivel0 ,yCor,yCor0;

let TipMasuratoare;

function preload()
{
  img1 = loadImage("Img/img1.jpg");
  img2 = loadImage("Img/img2.jpg");
  img3 = loadImage("Img/img3.jpg");
  img4 = loadImage("Img/img4.jpg");
}

function setup()
{
  createCanvas(1400, 700);
  background(175, 150, 150);
  gCluj = 9.807;
  NrMasuratori = Volume.length;
  MasuratoareNr = 1;
  dist = 25;
  yNivel = 0;
  yNivel0= 0;
  timpT1 = 0;
  yCor = 0;
  yCor0= 0;
  Robinet = 0;
  vascozitateAer = 1.0;
  DELTAH = 0.18;
  Hmax = 50;
  eroareH = Hmax*0.05*random(-1,+1);
  NivelApa = 100;

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(430, height-NrMasuratori*dist+i*dist+9, "");
    MarimeFizical.push(MF1);
    let MF2 = new Input(368, height-NrMasuratori*dist+i*dist+9,nf(Volume[i],3,0));
    volum.push(MF2);
    let MF3 = new Input(500, height-NrMasuratori*dist+i*dist+9, "");
  }

```



```
denivelare.push(MF3);
let MF4 = new Input(563, height-NrMasuratori*dist+i*dist+9,"");
denivelare.push(MF4);
}

razaCapilar = new Input(MarimeFizical[0].x-190,MarimeFizical[0].y+85,"4");
lungCapilar = new Input(MarimeFizical[0].x-125,MarimeFizical[0].y+85,"65");
temperatura = new Input(MarimeFizical[0].x-255,MarimeFizical[0].y+85,"22");
vascozitate = new Input(MarimeFizical[0].x+198,MarimeFizical[0].y+85,"");

//MarimeFizical[1].hide();

Student = createInput("");
Student.position(0.8*width+10, 20);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+55, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

volum = createInput("");
volum.position(0.5*width+377, 452);
volum.style('width', '10px');
volum.style('font-size', '7px');
volum.style('text-align', 'center');

denivelare = createInput("");
denivelare.position(0.5*width+509, 452);
denivelare.style('width', '10px');
denivelare.style('font-size', '7px');
denivelare.style('text-align', 'center');

vascozitate = createInput("");
vascozitate.position(0.5*width+563, 452);
vascozitate.style('width', '10px');
vascozitate.style('font-size', '7px');
vascozitate.style('text-align', 'center');

vascozitate = createInput("");
vascozitate.position(0.5*width+630, 452);
vascozitate.style('width', '10px');
vascozitate.style('font-size', '7px');
vascozitate.style('text-align', 'center');

button0 = createButton("Documentație laborator");
button0.position(20, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '180px');
button0.mouseClicked(Documentatie);

buttonMasor = createButton("Experiment");
buttonMasor.position(button0.x, button0.y+30);
buttonMasor.addClass("button button-primary button-border-thick button-box");
buttonMasor.style('width', '140px');
buttonMasor.mouseClicked(FunctieButon);
```

```

buttonReset = createButton("Reset");
buttonReset.position(0.48*width-20, height-380);
buttonReset.addClass("button button-primary button-border-thick button-box");
buttonReset.style('width', '140px');
buttonReset.mouseClicked(Reset);
buttonReset.hide();

```

```

buttonR = createButton("Deschide robinet");
buttonR.position(0.48*width-20, height-325);
buttonR.addClass("button button-primary button-border-thick button-box");
buttonR.style('width', '140px');
buttonR.hide();

```

```

rSlider = createSlider(0, 1, Robinet, 1);
rSlider.position(0.48*width-5, height-300);
rSlider.style('width', '100px');
rSlider.style('colour', 'red');
rSlider.hide();

```

```

}

```

```

function draw()
{
  background(245, 230, 190);
  image(img1, width*0.87, 40, width*0.1095, height*0.55);
  image(img4, width*0.76, 40, width*0.095, height*0.45);
  Tabel();
  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);

```

```

if(Robinet == 0 && rSlider.value() == 1)
{
  Start();
}

```

```

if(Robinet == 1 && rSlider.value() == 0)
{
  Robinet = 0;
  yCor0 = yCor;
  yNivel0 = yNivel;
  timpT1 = millis();
}

```

```

if (Masor)
{
  image(img2, width*0.3, 190, width*0.25, height*0.45);
  image(img3, width*0.5, 480, width*0.07, height*0.22);
  CurgeApa();
  AnsambluExperimental();
  Scala(1);
  ScalaApa(2);

  fill(55,95,240);
  stroke(0);
  strokeWeight(0);

```

```

beginShape();
  vertex(0.5*width-191, 0.35*height+80+yCor);
  vertex(0.5*width-191+125, 0.35*height+80+yCor);
  vertex(0.5*width-191+123, 0.55*height+75);
  vertex(0.5*width-189, 0.55*height+75);
endShape(CLOSE);
ellipse(0.5*width-128.5, 0.55*height+75,121,15);

beginShape();
  vertex(0.55*width-67, 0.75*height-24+100-yCor*0.8);
  vertex(0.55*width+25, 0.75*height-24+100-yCor*0.8);
  vertex(0.55*width+24, 0.75*height-24+120);
  vertex(0.55*width-66, 0.75*height-24+120);
endShape(CLOSE);
ellipse(0.55*width-21, 0.75*height+95,90,24);

stroke(220,50,50);
strokeWeight(1);
line(0.5*width-250, 0.35*height+80+yCor,0.5*width-150, 0.35*height+80+yCor)

}
}
function Start()
{
  Robinet = 1;
  timpT0 = millis();
}

function Reset()
{
  yNivel = 0;
  yCor = 0;
  yCor0 = 0;
  eroareH = Hmax*0.05*random(-1,+1);
  for(let i = Apa.length-1; i>=0 ;i-- )
  {
    Apa.splice(i,1);
  }
}

function CurgeApa()
{
  if(Robinet == 1)
  {
    let tsec = (millis()-timpT0)*0.001;
    yNivel = Hmax*(1-exp(-tsec*5))+eroareH;
    if(yCor < 100)
    {
      yCor = yCor0+0.00023*constK*Hmax*tsec/vascozitateAer; //Lidia cauta constanta
      SaCurgeApa();
      yNivel0 = yNivel;
      timpT2 = millis();
    }
  }
  else
  {
    timpT3 = millis();
    yNivel = yNivel0*Math.exp(-(timpT3-timpT2)*0.005);
  }
}
}

```

```

else{
  timpT0 = millis();
  yNivel = yNivel*Math.exp(-(timpT0-timpT1)*0.005);
  //console.log(timpT0);
}
for(let i = 0; i< Apa.length;i++)
{
  Apa[i].move();
  Apa[i].show();
  if (Apa[i].isOffScreen())
  {
    Apa[i].vx = random(-0.1,0.1);
    Apa[i].vy = random(-0.01,0.01);
  }
}
if (Apa.length>=200)
{
  Apa.splice(1,1);
}
}

function AnsambluExperimental()
{
  stroke(0);
  strokeWeight(2);
  line(0.4*width-434, 226, 0.4*width-434, 610);
  fill (250, 0, 0);
  ellipse(0.4*width-359, 610, 150, 150);
  fill (245, 230, 190);
  ellipse(0.4*width-359, 610, 110, 110);
  noStroke();
  rect (0.4*width-413.3, 534, 200, 76);
  stroke(0);
  strokeWeight(2);
  line(0.4*width-414.5, 230, 0.4*width-414.5, 610);
  line(0.4*width-284, 266, 0.4*width-284, 610);
  line(0.4*width-304.4, 266, 0.4*width-304, 610);
  fill (245, 230, 190);
  //ellipse(0.4*width-294.5, 266, 20.5, 20.5);
  noStroke();
  rect (0.4*width-303.6, 265, 18.5, 37);
  fill (250, 0, 0);
  rect (0.4*width-303, 445+yNivel, 18.2, 170-yNivel);
  rect (0.4*width-433, 445-yNivel, 18, 170+yNivel);
  stroke(0,130,0);
  line(120,445-yNivel,200,445-yNivel);
  line(210,445+yNivel,280,445+yNivel);

  stroke(0);
  strokeWeight(2);
  line(0.05*width-34, 300, 0.05*width-34, 205);
  line(0.05*width-40, 300, 0.05*width-40, 190);
  line(0.05*width-34, 205, 0.05*width+40, 205);
  line(0.05*width-40, 190, 0.05*width+40, 190);

  line(0.44*width-50, 210, 0.05*width+250, 210);
  line(0.44*width-35, 190, 0.05*width+250, 190);
  line(0.44*width-35, 218, 0.44*width-35, 190);

```

```

line(0.44*width-50, 218, 0.44*width-50, 210);

line(0.44*width-35, 296, 0.44*width-35, 270);
line(0.44*width-50, 296, 0.44*width-50, 270);

fill (245, 230, 190);
ellipse(0.45*width-420, 200, 245, 75);
noStroke();
rect (0.45*width-550, 191, 50, 13);
rect (0.45*width-330, 191, 50, 18);
rect (0.45*width-503, 218, 17.5, 20);
}

function Tabel()
{
  if (rSlider.value()==0)
  {
    buttonR.html("Deschide robinet");
  }
  else
  {
    buttonR.html("Inchide robinet");
  }
}

rhoApaT(temperatura.MyInput.value());
let RC = razaCapilar.MyInput.value();
let LC = lungCapilar.MyInput.value();
constK = Math.PI*rhoApa*gCluj*Math.pow(RC,4)/8.0/LC*0.001;
console.log(constK);
noStroke();
fill(0,0,0);
textSize(12);
text("Densitatea apei = "+nf(rhoApa,3,2)+" [kg/m ]",0.5*width+220, 415);
textSize(8);
text("3",0.5*width+386, 410);
text("-4",0.5*width+245, 455);
text("-3",0.5*width+312, 455);
text("3",0.5*width+395, 455);
text("2",0.5*width+597, 455);
text("2",0.5*width+663, 455);
textSize(22);
text("Laboratorul de Fizică: Determinarea coeficientului de vâscozitate dinamică a aerului", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,1*width-20, height-45);

for (var i=0; i<9; i++)
{
  line(MarimeFizica1[0].x+420+i*65,MarimeFizica1[0].y-80,MarimeFizica1[0].x+420+i*65,height-30);
}
line(MarimeFizica1[0].x+420,MarimeFizica1[0].y-80,MarimeFizica1[0].x+940,MarimeFizica1[0].y-80);
line(MarimeFizica1[0].x+420,MarimeFizica1[0].y-41,MarimeFizica1[0].x+940,MarimeFizica1[0].y-41);
line(MarimeFizica1[0].x+420,height-30,MarimeFizica1[0].x+940,height-30);
line(MarimeFizica1[0].x+615,MarimeFizica1[0].y-80+64,MarimeFizica1[0].x+875,MarimeFizica1[0].y-80+64);

```

```

line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+89,MarimeFizical[0].x+875,MarimeFizical[0].y-80+89);
line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+114,MarimeFizical[0].x+875,MarimeFizical[0].y-80+114);
line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+139,MarimeFizical[0].x+875,MarimeFizical[0].y-80+139);
line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+164,MarimeFizical[0].x+875,MarimeFizical[0].y-80+164);
line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+189,MarimeFizical[0].x+875,MarimeFizical[0].y-80+189);
line(MarimeFizical[0].x+615,MarimeFizical[0].y-80+214,MarimeFizical[0].x+875,MarimeFizical[0].y-80+214);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-Napoca",830, 685);

fill(0);
noStroke();
textSize(12);
// Capul de tabel

text("T",MarimeFizical[0].x+420+10,MarimeFizical[0].y-65);
text("cameră",MarimeFizical[0].x+420+15,MarimeFizical[0].y-60);
text("[°C]",MarimeFizical[0].x+415+33,MarimeFizical[0].y-45);

text("R",MarimeFizical[0].x+450+65,MarimeFizical[0].y-65);
text("[10 m]",MarimeFizical[0].x+500,MarimeFizical[0].y-45);

text("I",MarimeFizical[0].x+450+2*65,MarimeFizical[0].y-67);
text("[10 m]",MarimeFizical[0].x+435+2*65,MarimeFizical[0].y-45);

text("V",MarimeFizical[0].x+450+3*65,MarimeFizical[0].y-67);
text("[10 m ]",MarimeFizical[0].x+428+3*65,MarimeFizical[0].y-45);
//text("3",MarimeFizical[0].x+470+3*65,MarimeFizical[0].y-50);

text("Δt",MarimeFizical[0].x+447+4*65,MarimeFizical[0].y-65);
text("[s]",MarimeFizical[0].x+447+4*65,MarimeFizical[0].y-45);

text("Δh",MarimeFizical[0].x+460+5*62,MarimeFizical[0].y-67);
text("[10 m]",MarimeFizical[0].x+430+5*65,MarimeFizical[0].y-45);

text("η",MarimeFizical[0].x+450+6*65,MarimeFizical[0].y-67);
text("[10 Ns/m ]",MarimeFizical[0].x+420+6*65,MarimeFizical[0].y-45);

text("η",MarimeFizical[0].x+435+7*65,MarimeFizical[0].y-67);
text("[10 Ns/m ]",MarimeFizical[0].x+421+7*65,MarimeFizical[0].y-45);
text("mediu",MarimeFizical[0].x+442+7*65,MarimeFizical[0].y-64);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/SIM/2016-2017/Laboratoare/Determinarea%20coeficientului%20de%20vascozitate%20dinamica%20a%20aerului.pdf';
}

function FunctieButon()

```

```

{
  Masor = true;
  buttonReset.show();
  rSlider.show();
  buttonR.show();
}

function Scala(yCorr)
{
  var x = 177;
  var y = 464;
  dy0 = 6;
  dy1 = 5*dy0;
  dy2 = 10*dy0;
  let salt = -20;
  strokeWeight(1);
  stroke(0,0,220);
  for(let i=-30; i<30; i++)
  {
    line(x,y+yCorr-i*dy0+salt,x+28,y+yCorr-i*dy0+salt);
  }

  for(let i=-6; i<6; i++)
  {
    line(x,y+yCorr-i*dy1+salt,x+38,y+yCorr-i*dy1+salt);
  }
  strokeWeight(2);
  for(let i=-3; i<=3; i++)
  {
    line(x,y+yCorr-i*dy2+salt,x+48,y+yCorr-i*dy2+salt);
  }
  noStroke();
  fill(0);
  for(let i=-3; i<=3; i++)
  {
    text(nf(10*i,1,0),x+53,y+yCorr-i*dy2+3+salt);
  }
}

function ScalaApa(yCorr)
{
  var x = 500;
  var y = 463.5;
  dy0 = 12;
  dy1 = 5*dy0;
  dy2 = 10*dy0;
  let salt = -20;
  strokeWeight(1);
  stroke(0,0,220);
  for(let i=0; i<10; i++)
  {
    line(x-28,y+yCorr-i*dy0+salt,x,y+yCorr-i*dy0+salt);
  }

  for(let i=0; i<20; i++)
  {
    line(x-20,y+yCorr-i*dy0*0.5+salt,x,y+yCorr-i*dy0*0.5+salt);
  }
}

```

```
for(let i=0; i<2; i++)
{
  line(x-38,y+yCorr-i*dy1+salt,x,y+yCorr-i*dy1+salt);
}
strokeWeight(2);
for(let i=0; i<=1; i++)
{
  line(x-48,y+yCorr-i*dy2+salt,x,y+yCorr-i*dy2+salt);
}
noStroke();
fill(0);
textAlign(RIGHT);
for(let i=0; i<=5; i++)
{
  text(nf(200*i,1,0),x-53,y+yCorr-i*dy0*2+3+salt);
}
textAlign(LEFT);
}

function rhoApaT(Temp)
{
  rhoApa = 1001.56878-0.15612*Temp-0.00266*Temp*Temp;
}

function SaCurgaApa()
{
  if(random(0,1)<0.5)
  {
    let pozX = 0.55*width-18+random(-1,1);
    let pozY = 0.75*height-54;
    let picatura = new Circle(pozX,pozY,random(3,6),55,95,240);
    if(Apa.length < 200)
    {
      Apa.push(picatura);
    }
  }
}
```


10. Studiul conductibilității electrice a metalelor

```

let img1, img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Rezystenta = [];
let Rezystivitate = [];
let Conductibilitate = [];
let Mobilitate = [];
let Temperatura = [20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90];
let Student, Grupa;
var Contor0, Contor;
let CresteTemp = false;
let Temp;

let panta, intersectia, eroarea;

let TipMasuratoare;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};

function preload()
{
  img1 = loadImage("Img/1.jpg");
  img2 = loadImage("Img/Conductibilitatea_Apa.jpg");
}

function setup() {
  createCanvas(1400, 700);
  background(230, 150, 110);
  NrMasuratori = Temperatura.length;
  dist = 50;

  for (let i = 0; i < NrMasuratori; i++) {
    let MF1 = new Input((NrMasuratori + 1) * dist + i * dist * 1.189 - 400, height - 202);
    Rezystenta.push(MF1);
  }
  for (let i = 0; i < NrMasuratori; i++) {
    let MF1 = new Input((NrMasuratori + 1) * dist + i * dist * 1.189 - 400, height - 170);
    Rezystivitate.push(MF1);
  }
  for (let i = 0; i < NrMasuratori; i++) {
    let MF1 = new Input((NrMasuratori + 1) * dist + i * dist * 1.189 - 400, height - 135);
    Conductibilitate.push(MF1);
  }
}

```

```
for (let i = 0; i < NrMasuratori; i++) {
  let MF1 = new Input((NrMasuratori + 1) * dist + i * dist * 1.189 - 400, height - 102);
  Mobilitate.push(MF1);
}

for (let i = 0; i < NrMasuratori; i++) {
  trace1.x[i] = Temperatura[i];
}
Temp = Temperatura[0];

Student = createInput("");
Student.position(0.5 * width + 20, 20);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x + Student.width + 55, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

Rezistivitate = createInput("");
Rezistivitate.position(0.5 * width - 658, 581);
Rezistivitate.style('width', '9px');
Rezistivitate.style('font-size', '7px');
Rezistivitate.style('text-align', 'center');

Conductibilitate = createInput("");
Conductibilitate.position(0.5 * width - 662, 615);
Conductibilitate.style('width', '9px');
Conductibilitate.style('font-size', '7px');
Conductibilitate.style('text-align', 'center');

Mobilitate = createInput("");
Mobilitate.position(0.5 * width - 663, 649);
Mobilitate.style('width', '9px');
Mobilitate.style('font-size', '7px');
Mobilitate.style('text-align', 'center');

button0 = createButton("Documentatie Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '200px');
button0.mouseClicked(Documentatie);

buttonMasor = createButton("Experiment");
buttonMasor.position(button0.x, button0.y + 30);
buttonMasor.addClass("button button-primary button-border-thick button-box");
buttonMasor.style('width', '200px');
buttonMasor.mouseClicked(FunctieButon);

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(0.71 * width, 0.5 * height);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);
```

```

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y + 30);
Graf.style('width', '400px');
Graf.style('height', '300px');

buttonStart = createButton("Start");
buttonStart.position(button0.x, button0.y + 60);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '90px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x + buttonStart.width + 20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '90px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

data = createButton("l = 384 m");
data.position(button0.x + 1116, button0.y + 300);
data.addClass("button button-primary button-border-thick button-box");
data.style('width', '120px');

data = createButton("Ø = 0,132 mm");
data.position(button0.x + 1238, button0.y + 300);
data.addClass("button button-primary button-border-thick button-box");
data.style('width', '126px');
}

```

```

function draw() {
  background(250, 210, 190);
  image(img1, width * 0.77, 20, width * 0.17, height * 0.45);
  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x - 55, Student.y + 5);
  text("Grupa: ", Grupa.x - 50, Grupa.y + 5);

```

```
Tabel();
```

```

if (Masor) {
  image(img2, button0.x + button0.width + 10, 50, width * 0.5, height * 0.65);
  for (let i = 0; i < NrMasuratori; i++) {
    trace1.y[i] = Rezistenta[i].MyInput.value();
  }
  LinFit(trace1.x, trace1.y, NrMasuratori - 1);
  var Minim = 0.9 * min(trace1.x);
  var Maxim = 1.1 * max(trace1.x);
  var NrFit = 101;
  var dx = (Maxim - Minim) / (NrFit - 1.0);
  for (var i = 0; i < NrFit; i++) {
    trace2.x[i] = Minim + i * dx;
    trace2.y[i] = intersectia + panta * trace2.x[i];
  }
  noFill();
  stroke(0);
  strokeWeight(1);
  rect(0.5 * width - 11, 0.1 * height, 10, 0.35 * height);

```

```

Grafic();
fill(200, 0, 220);
if (CresteTemp) {
  Temp = Temperatura[0] + (frameCount - Contor0) * 0.002;
  //console.log(Temp);

  if (Temp > 95) {
    CresteTemp = false;
  }
}
let R0 = 443.34066;
let alfa = 1.75934;

rect(0.5 * width - 11, 0.40 * height - 10 - Temp * 1.8, 10, 110 + Temp * 1.8);
noStroke();
textSize(16);
text(nf(Temp, 1, 1), 0.5 * width + 15, 0.40 * height - 7 - Temp * 1.8);
fill(200);
noStroke();
rect(0.40 * width + 12, 0.65 * height - 40, 60, 20);
fill(240, 0, 0);
stroke(0, 0, 120);
text(nf(R0 + alfa * Temp, 1, 2), 0.40 * width + 20, 0.65 * height - 25);
noStroke();
fill(0);
textSize(14);
text("Ω", 0.40 * width + 25, 0.65 * height - 5);
}

if (ImgArataGrafic == 'Grafic Offline') {
  Graf.show();
  Grafic();
}
else {
  Graf.hide();
}
}

function Tabel() {
  textSize(22);
  fill(0, 0, 0);
  noStroke();
  text("Laboratorul de Fizica: Studiul conductibilității electrice a metalelor", 15, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7, width - 10, height - 10);
  stroke(0);
  rect(5, 5, width - 10, height - 10);
  strokeWeight(1);
  rect(10, 35, 0.7 * width - 15, height - 45);
  rect(0.7 * width, 15, 0.3 * width - 10, height - 25);
  for (var i = 0; i < 17; i++) {
    line(Rezistenta[0].x - 385 + i * 59.5, Rezistenta[0].y + 10, Rezistenta[0].x - 385 + i * 59.5, height - 33);
  }
  line(Rezistenta[0].x - 385, Rezistenta[0].y + 10, Rezistenta[0].x + 567, Rezistenta[0].y + 10);
  line(Rezistenta[0].x - 385, Rezistenta[0].y + 35, Rezistenta[0].x + 567, Rezistenta[0].y + 35);
  line(Rezistenta[0].x - 385, Rezistenta[0].y + 65, Rezistenta[0].x + 567, Rezistenta[0].y + 65);
  line(Rezistenta[0].x - 385, Rezistenta[0].y + 101, Rezistenta[0].x + 567, Rezistenta[0].y + 101);
}

```

```

line(Rezistenta[0].x - 385, Rezistenta[0].y + 135, Rezistenta[0].x + 567, Rezistenta[0].y + 135);
line(Rezistenta[0].x - 385, height - 33, Rezistenta[0].x + 567, height - 33);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca", 220, 685);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("t [°C]", Rezistenta[0].x - 365, Rezistenta[0].y + 27);
text("R [Ω]", Rezistenta[0].x - 365, Rezistenta[0].y + 55);
text("p", Rezistenta[0].x - 355, Rezistenta[0].y + 73);
text("[10 Ωm]", Rezistenta[0].x - 379, Rezistenta[0].y + 98);
text("σ", Rezistenta[0].x - 355, Rezistenta[0].y + 110);
text("[10 Ω m ]", Rezistenta[0].x - 384, Rezistenta[0].y + 132);
text("μ", Rezistenta[0].x - 355, Rezistenta[0].y + 143);
text("[10 m2/Vs]", Rezistenta[0].x - 384, Rezistenta[0].y + 164);
textSize(8);
text("-1", Rezistenta[0].x - 350, Rezistenta[0].y + 127);
text("-1", Rezistenta[0].x - 335, Rezistenta[0].y + 127);

textSize(12);
for (let i = 0; i < NrMasuratori; i++) {
  text(nf(Temperatura[i], 1), Rezistenta[0].x - 300 + i * dist * 1.188, Rezistenta[0].y + 28);
}
}

function Start() {
  CresteTemp = true;
  Contor0 = frameCount;
}

function Stop() {
  CresteTemp = false;
}

function ArataGrafic() {
  if (ImgArataGrafic == 'Grafic Online') {
    ImgArataGrafic = 'Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

function Grafic() {
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =

```

```

{
  margin:
  {
    t: 50,
    b: 50,
    l: 50,
    r: 50
  },
  title: 'Variatia rezistentei cu Temp',
  xaxis:
  {
    title: 't [C]',
    visible: 1
  },
  yaxis:
  {
    title: 'R [ $\Omega$ ]'
  },
  paper_bgcolor: "#c4dedc"
};

var data = [trace1, trace2];
//let data = [trace1];

Plotly.newPlot(GRAFIC, data, layout);
}

function Documentatie() {
  window.location = 'https://phys.utcluj.ro/resurse/Facultati/AR+ITT/L1.pdf';
}

function FunctieButon() {
  Masor = true;
  buttonStart.show();
  buttonStop.show();
}

function LinFit(x, y, N) {
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N + 1];
  for (let i = 1; i <= N; i++) {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for (let i = 1; i <= N; i++) {
    f = 1.0 / (sigmy[i] * sigmy[i]);
    s += f;
    sx += x[i] * f;
    sxx += x[i] * x[i] * f;
    sy += y[i] * f;
    sxy += x[i] * y[i] * f;
  }
  f = 1.0 / (s * sxx - sx * sx);
  panta = (s * sxy - sx * sy) * f;
  intersectia = (sy * sxx - sx * sxy) * f;
}

```

```
eroarea = 0.0;
for (let i = 1; i <= N; i++) {
  eroarea += pow((y[i] - panta * x[i] - intersectia) / sigmy[i], 2);
}
}
```

11. Studiul efectului Hall

```

let img1, img2, img3;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let IBper_a = [];
let UH = [];
let Int = [];
let CpMagnetic = [];
let RH, Rh, NrElectroni, Conductivitate, Mobilitate;
let TipMasuratoare;
var Is = [2, 4, 6, 8, 2, 4, 6, 8, 2, 4, 6, 8];
let IntB0;
let panta, intersectia, eroarea;
let Studenti, Grup;
let latime;
let ConstantaHall;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};

function preload()
{
  img1 = loadImage("Img/Constanta Hall.jpg");
  img2 = loadImage("Img/Fig1.jpg");
  img3 = loadImage("Img/Hall.jpg");
}

function B0(I)
{
  if(I>=1)
  {
    return -0.07+0.24*I-0.04*I*I;
  }
  else{
    return 0.0;
  }
}

function corectie(I)
{
  return 0.00714+2.69048*I-0.27321*I*I+0.01042*I*I*I;
}

function setup()
{

```



```

createCanvas(1400, 700);
background(220, 200, 200);
NrMasuratori = Is.length;
dist = 22;
Rh = 0.01741+0.00087*random(-1,1);
console.log(Rh);
latime = 1.5;

//LatimeScala = 40;
//InaltimeScala = 10;
//Scala(LatimeScala,InaltimeScala);

for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Input(230,height-(NrMasuratori+1)*dist+i*dist+15, "");
  IBper_a.push(MF1);
  let MF2 = new Input(295,height-(NrMasuratori+1)*dist+i*dist+15, "");
  UH.push(MF2);
}

for (let i=0; i<3; i++)
{
  let MF1 = new Input(230-195,IBper_a[1].y+(0.5+4*i)*dist,1.5+i*0.5, "");
  Int.push(MF1);
  let M1 = MF1.MyInput.value();
  let MF2 = new Input(230-130,IBper_a[1].y+(0.5+4*i)*dist,nf(round((B0(M1))*100)*0.01,1,2));
  CpMagnetic.push(MF2);
}

RH = new Input(UH[0].x+1*65,Int[1].y, "");
IntB0 = new Input(0.455*width,0.32*height+43, "");
IntB0.hide();
NrElectroni = new Input(UH[0].x+2*65,Int[1].y, "");
Conductivitate = new Input(UH[0].x+3*65,Int[1].y, "");
Mobilitate = new Input(UH[0].x+4*65,Int[1].y, "");

button0 = createButton("Document");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentation);

buttonGB = createButton("Masuratori");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

Student = createInput("");
Student.position(0.5*width+10, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);

```

```

Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

rSlider = createSlider(0, 10, 2, 0.1);
rSlider.position(IBper_a[0].x-80, IBper_a[0].y-210);
rSlider.style('width', '100px');
rSlider.style('colour', 'red');
rSlider.hide();

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(IBper_a[0].x+0.28*width, IBper_a[0].y-45);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '350px');
Graf.style('height', '280px');

}

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.71, 30, width*0.27, height*0.35);
  image(img2, width*0.73, 340, width*0.23, height*0.46);

  noStroke();
  //fill(200,0,0);
  //stroke(200,0,0);
  //strokeWeight(1);
  textSize(12);
  text("Student: ", Student.x-53, Student.y+5);
  text("Grupa: ", Grupa.x-45, Grupa.y+5);
  Tabel();
  if (Masor)
  {
    image(img3, width*0.10, 70, width*0.47, height*0.43);
    let Intensitate = rSlider.value();
    let IB0 = IntB0.MyInput.value();
    text('B0 = '+nf(round(B0(IB0)*100)*0.01, 1, 2)+' [T]', 0.35*width-28, 0.25*height+33);
    text(nf(Intensitate, 1, 1), IBper_a[0].x-22, IBper_a[0].y-265);
    stroke(185, 122, 87);
    let corectie = correctie(Intensitate);
    let TensHall = 1000*(Intensitate*B0(IB0)*Rh/latime)+corectie;
    fill(150);
    noStroke();
    rect(0.31*width-10, 0.12*height-10, 35, 15);
    fill(0);
    textSize(12);
    text(nf(TensHall, 1, 2), 0.31*width-5, 0.12*height+3);
    textSize(10);
  }
}

```

```

    text("mV",0.31*width-0,0.12*height+15);
}

if(ImgArataGrafic == 'Grafic Offline')
{
    for(let i = 0; i<NrMasuratori; i++)
    {
        trace1.x[i] = IBper_a[i].MyInput.value();
        trace1.y[i] = UH[i].MyInput.value();
    }
    LinFit(trace1.x,trace1.y,NrMasuratori-1);
    var Minim = 0.9*min(trace1.x);
    var Maxim = 1.1*max(trace1.x);
    var NrFit = 101;
    var dx = (Maxim-Minim)/(NrFit-1.0);
    for (var i = 0; i< NrFit; i++ )
    {
        trace2.x[i] = Minim+i*dx;
        trace2.y[i] = intersectia+panta*trace2.x[i];
    }
    Graf.show();
    Grafic();
}
else
{
    Graf.hide();
}

}

function Tabel()
{
    textSize(22);
    fill(0,0,0);
    noStroke();
    text("Laborator de Fizica - Studiul efectului Hall ", 20, 30);

    fill(0);
    noStroke();
    textSize(12);
    text("Toate drepturile rezervate © Departamentului de Fizica si Chimie",1010, 679);
    text("Universitatea Tehnică din Cluj-Napoca",1070, 689);
    noFill();

    strokeWeight(2);
    stroke(255);
    rect(7, 7,width-10, height-10);
    stroke(0);
    rect(5, 5,width-10, height-10);
    strokeWeight(1);
    rect(10, 35,0.7*width-20, height-45);
    rect(0.7*width, 15,0.3*width-10, height-25);
    for (var i=0; i<10; i++)
    {
        line(IBper_a[0].x-82-130+i*65,IBper_a[0].y-50,IBper_a[0].x-82-130+i*65,height-15);
    }
    line(IBper_a[0].x-212,IBper_a[0].y-50,IBper_a[0].x+372,IBper_a[0].y-50);
    line(IBper_a[0].x-212,IBper_a[4].y-9,IBper_a[0].x+112,IBper_a[4].y-9);
}

```

```

line(IBper_a[0].x-212,IBper_a[8].y-9,IBper_a[0].x+112,IBper_a[8].y-9);
line(IBper_a[0].x-212,height-15,IBper_a[0].x+497-125,height-15);
line(IBper_a[0].x-212,IBper_a[0].y-15,IBper_a[0].x+372,IBper_a[0].y-15);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("I [A]",IBper_a[0].x-190,IBper_a[0].y-30);
text("B [T]",IBper_a[0].x-130,IBper_a[0].y-30);
text("Is [mA]",IBper_a[0].x+60-130,IBper_a[0].y-30);
text("IB/a",IBper_a[0].x,IBper_a[0].y-35);
text("[AT/m]",IBper_a[0].x+125-130,IBper_a[0].y-20);
text("UH",IBper_a[0].x+70,IBper_a[0].y-35);
text("[mV]",IBper_a[0].x+65,IBper_a[0].y-20);
text("RH",IBper_a[0].x+137,IBper_a[0].y-35);
text("[Vm/AT]",IBper_a[0].x+127,IBper_a[0].y-20);
text("n·10",IBper_a[0].x+198,IBper_a[0].y-35);
text("[e/m ]",IBper_a[0].x+325-130,IBper_a[0].y-20);

text("σ",IBper_a[0].x+272,IBper_a[0].y-35);
text("[Ω·m]",IBper_a[0].x+260,IBper_a[0].y-20);
text("μ",IBper_a[0].x+335,IBper_a[0].y-35);
text("[m /Vs]",IBper_a[0].x+322,IBper_a[0].y-20);

for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(Is[i],1,1),IBper_a[0].x-55,IBper_a[0].y+i*dist+7);
}

textSize(8);
text("3",IBper_a[0].x+218,IBper_a[0].y-25);
text("20",IBper_a[0].x+222,IBper_a[0].y-40);
text("-1",IBper_a[0].x+289,IBper_a[0].y-28);
text("2",IBper_a[0].x+335,IBper_a[0].y-25);

}

function Documentation()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/EfectulHall/Doc/Studiul%20efectului%20Hall.pdf';
}

function FunctieButon()
{
  rSlider.show();
  Masor = true;
  IntB0.show();
}

function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else

```

```

{
  ImgArataGrafic = 'Grafic Online';
  buttonGraf.html(ImgArataGrafic);
}
}

```

```

function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Efectul Hall',
    xaxis:
    {
      title:'IB/a [AT/m]',
      visible: 1
    },
    yaxis:
    {
      title:'U_H [mV]'
    },
    paper_bgcolor: "#c4dedc"
  };

  var data = [trace1,trace2];
  //let data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

```

```

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
    sx += x[i]*f;

```

```
sxx+= x[i]*x[i]*f;
sy += y[i]*f;
sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

12. Etalonarea unui spectroscop. Analiză spectrală calitativă

```

var img1, img2, img3, img4, img5, img6, img7, img8, imgScala;
var LatimeScala, InaltimeScala;
var Culori = [];
var r_o, g_o, b_o;
var inaltime;
var ImgArataGrafic;
var PondereSmooth = [];
var x = [];
var y = [];
var lambdaNe = [4964, 5031, 5330, 5400, 5760, 5852, 5945, 6143, 6402];
var TE = [4964, 5031, 5330, 5400, 5760, 5852, 5945, 6143, 6402];
var panta, intersectia, eroarea;
var ArataSpNe, ArataSpHg;
var CorectieScala;

```

```

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'scatter',
  name: 'Date Exp.'
};

```

```

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};

```

```

function preload()
{
  img1 = loadImage("Img/AnsambluExperimental.jpg");
  img2 = loadImage("Img/Neon.jpg");
  img3 = loadImage("Img/Ne-rosu.jpg");
  img4 = loadImage("Img/Ne-albastru.jpg");
  img5 = loadImage("Img/Hg-Galben.jpg");
  img6 = loadImage("Img/Hg-Violet.jpg");
  img7 = loadImage("Img/Ne-spectruOK.jpg");
  img8 = loadImage("Img/Hg-spectru.jpg");

```

```

}

```

```

function setup()
{
  createCanvas(1400, 700);
  background(220, 200, 200);

  LatimeScala = 40;
  InaltimeScala = 10;
  //Scala(LatimeScala, InaltimeScala);

  ImgArataGrafic = 'Grafic Online';

```

```
button0 = createButton("Document");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentatie);

button1 = createButton(ImgArataGrafic);
button1.position(30, 0.45*height);
button1.addClass("button button-primary button-border-thick button-box");
button1.style('width', '150px');
button1.mouseClicked(ArataGrafic);

buttonNe = createButton("Masuratori Neon");
buttonNe.position(button0.x+button0.width+10, button0.y);
buttonNe.addClass("button button-primary button-border-thick button-box");
buttonNe.style('width', '150px');
buttonNe.mouseClicked(Neon);

buttonHg = createButton("Masuratori Mercur");
buttonHg.position(buttonNe.x+buttonNe.width+10, buttonNe.y);
buttonHg.addClass("button button-primary button-border-thick button-box");
buttonHg.style('width', '150px');
buttonHg.mouseClicked(Mercur);

input1 = createInput();
input1.position(button1.x+160, button1.y-30);
//input1.value(TE[0]);
input1.style('width', '40px');
input1.style('font-size', '12px');

input2 = createInput();
input2.position(input1.x+input1.width+10, input1.y);
//input2.value(TE[1]);
input2.style('width', '40px');
input2.style('font-size', '12px');

input3 = createInput();
input3.position(input2.x+input2.width+10, input1.y);
//input3.value(TE[2]);
input3.style('width', '40px');
input3.style('font-size', '12px');

input4 = createInput();
input4.position(input3.x+input3.width+10, input1.y);
//input4.value(TE[3]);
input4.style('width', '40px');
input4.style('font-size', '12px');

input5 = createInput();
input5.position(input4.x+input4.width+10, input1.y);
//input5.value(TE[4]);
input5.style('width', '40px');
input5.style('font-size', '12px');

input6 = createInput();
input6.position(input5.x+input5.width+10, input1.y);
//input6.value(TE[5]);
input6.style('width', '40px');
input6.style('font-size', '12px');
```



```
input7 = createInput();
input7.position(input6.x+input6.width+10, input1.y);
//input7.value(TE[6]);
input7.style('width', '40px');
input7.style('font-size', '12px');

input8 = createInput();
input8.position(input7.x+input7.width+10, input1.y);
//input8.value(TE[7]);
input8.style('width', '40px');
input8.style('font-size', '12px');

input9 = createInput();
input9.position(input8.x+input8.width+10, input1.y);
//input9.value(TE[8]);
input9.style('width', '40px');
input9.style('font-size', '12px');

input11 = createInput();
input11.position(button1.x+160, button1.y-30);
input11.style('width', '40px');
input11.style('font-size', '12px');

input12 = createInput();
input12.position(input1.x+input1.width+10, input1.y);
input12.style('width', '40px');
input12.style('font-size', '12px');

input13 = createInput();
input13.position(input2.x+input2.width+10, input1.y);
input13.style('width', '40px');
input13.style('font-size', '12px');

input14 = createInput();
input14.position(input3.x+input3.width+10, input1.y);
input14.style('width', '40px');
input14.style('font-size', '12px');

input15 = createInput();
input15.position(input4.x+input4.width+10, input1.y);
input15.style('width', '40px');
input15.style('font-size', '12px');

input16 = createInput();
input16.position(input5.x+input5.width+10, input1.y);
input16.style('width', '40px');
input16.style('font-size', '12px');

input17 = createInput();
input17.position(input6.x+input6.width+10, input1.y);
input17.style('width', '40px');
input17.style('font-size', '12px');

input18 = createInput();
input18.position(input7.x+input7.width+10, input1.y);
input18.style('width', '40px');
input18.style('font-size', '12px');

input19 = createInput();
input19.position(input8.x+input8.width+10, input1.y);
```

```
input19.style('width', '40px');
input19.style('font-size', '12px');

input21 = createInput();
input21.position(button1.x+160, button1.y-60);
input21.style('width', '40px');
input21.style('font-size', '12px');

input22 = createInput();
input22.position(input1.x+input1.width+10, input21.y);
input22.style('width', '40px');
input22.style('font-size', '12px');

input23 = createInput();
input23.position(input2.x+input2.width+10, input21.y);
input23.style('width', '40px');
input23.style('font-size', '12px');

input24 = createInput();
input24.position(input3.x+input3.width+10, input21.y);
input24.style('width', '40px');
input24.style('font-size', '12px');

input25 = createInput();
input25.position(input4.x+input4.width+10, input21.y);
input25.style('width', '40px');
input25.style('font-size', '12px');

input26 = createInput();
input26.position(input5.x+input5.width+10, input21.y);
input26.style('width', '40px');
input26.style('font-size', '12px');

input27 = createInput();
input27.position(input6.x+input6.width+10, input21.y);
input27.style('width', '40px');
input27.style('font-size', '12px');

input28 = createInput();
input28.position(input7.x+input7.width+10, input21.y);
input28.style('width', '40px');
input28.style('font-size', '12px');

input29 = createInput();
input29.position(input8.x+input8.width+10, input21.y);
input29.style('width', '40px');
input29.style('font-size', '12px');

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(button1.x, height-360);
Graf.style('width', '700px');
Graf.style('height', '340px');

rSlider = createSlider(-50, 50, 0, 1);
rSlider.position(Graf.x+180, Graf.y-115);
rSlider.style('width', '350px');
rSlider.hide();

Student = createInput("");
```

```

Student.position(0.4*width, 20);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+70, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

for (var i = 0; i<lanbdaNe.length; i++)
{
  trace1.y[i] = lanbdaNe[i];
  y[i] = lanbdaNe[i];
}

//Grafic();
ArataSpNe = 0;
ArataSpHg = 0;
button1.hide();
input1.hide();
input2.hide();
input3.hide();
input4.hide();
input5.hide();
input6.hide();
input7.hide();
input8.hide();
input9.hide();
input11.hide();
input12.hide();
input13.hide();
input14.hide();
input15.hide();
input16.hide();
input17.hide();
input18.hide();
input19.hide();
input21.hide();
input22.hide();
input23.hide();
input24.hide();
input25.hide();
input26.hide();
input27.hide();
input28.hide();
input29.hide();

}

function draw()
{
  background(220, 200, 200);
  //textSize(15);
  //text('Nume Fisier',input.x-7, input.y-15);

  image(img1, width*0.62-15, 20, height*0.5, height*0.5-20);
  image(img2, width*0.62-15, height*0.5, height*0.5, height*0.5-15);
  image(img3, width-height*0.25-10, 20, height*0.24, height*0.24-5);
}

```

```

image(img4, width-height*0.25-10, height*0.25+12, height*0.24, height*0.24-5);
image(img5, width-height*0.25-10, height*0.50+3, height*0.24, height*0.24-5);
image(img6, width-height*0.25-10, height*0.75-5, height*0.24, height*0.24-5);

if (ArataSpNe == 1)
{
  image(img7, 20, height*0.05+40, width*0.5, height*0.20);
  fill(0,0,0);
  rect(20, height*0.15+30, width*0.5-5, height*0.10+10);
  noStroke();
  textSize(14);
  CorectieScala = rSlider.value();
  text("deviere = "+nf(CorectieScala,1,0),Graf.x+530, Graf.y-110);
  ScalaNe(-CorectieScala);
  fill(0,255,0);
  ellipse(0.2*width+32,height*0.05+47,4,4);
  ellipse(0.2*width+39,height*0.05+47,4,4);
  ellipse(0.2*width+82,height*0.05+47,4,4);
  ellipse(0.2*width+92,height*0.05+47,4,4);
  ellipse(0.2*width+150,height*0.05+47,4,4);
  ellipse(0.2*width+165,height*0.05+47,4,4);
  ellipse(0.2*width+180,height*0.05+47,4,4);
  ellipse(0.2*width+202,height*0.05+47,4,4);
  ellipse(0.2*width+243,height*0.05+47,4,4);

  noFill();
  stroke(0);
  rect(button1.x-5,button1.y-75,button1.width+lanbdaNe.length*(input1.width+10)+5,60);
  rect(button1.x-5,button1.y-75,button1.width+lanbdaNe.length*(input1.width+10)+5,30);
  rect(button1.x-5,button1.y-95,button1.width,80);

  for (var i = 0; i<lanbdaNe.length; i++)
  {
    fill(0);
    noStroke();
    textSize(12);
    text(nf(lanbdaNe[i],0,0),input1.x+i*(input1.width+10),input1.y-25);
  }
  text("lungimea de unda [Å]",button1.x+20,input1.y-25);
  text("pozitia liniei [div.]",button1.x+20,input1.y+5);
  text("Date Neon",button1.x+20,input1.y-50);

  x[0] = trace1.x[0] = input1.value();
  x[1] = trace1.x[1] = input2.value();
  x[2] = trace1.x[2] = input3.value();
  x[3] = trace1.x[3] = input4.value();
  x[4] = trace1.x[4] = input5.value();
  x[5] = trace1.x[5] = input6.value();
  x[6] = trace1.x[6] = input7.value();
  x[7] = trace1.x[7] = input8.value();
  x[8] = trace1.x[8] = input9.value();

  LinFit(x,y,8);
  var N = 801;
  var Lmin = 350;
  var Lmax = 750;
  var dl = (Lmax-Lmin)/N;
  for (var i = 0; i<N; i++)
  {
    trace2.x[i] = Lmin+i*dl;
  }

```

```

    trace2.y[i] = intersectia+panta*trace2.x[i];
    fill(0);
    //
  }
  Grafic();
}

if (ArataSpHg == 1)
{
  image(img8, 20, height*0.05+40, width*0.5, height*0.20);
  fill(0,0,0);
  rect(20, height*0.15+30, width*0.5+5, height*0.10+10);
  rect(20, height*0.05+40, 70, height*0.20);
  rect(width*0.5-20, height*0.05+40, 70, height*0.20);
  ScalaHg(-CorectieScala*1.1);

  noFill();
  stroke(0);
  rect(button1.x-5,button1.y-75,button1.width+lanbdaNe.length*(input1.width+10)+5,60);
  rect(button1.x-5,button1.y-75,button1.width+lanbdaNe.length*(input1.width+10)+5,30);
  rect(button1.x-5,button1.y-95,button1.width,80);

  fill(0);
  noStroke();
  textSize(12);
  text("lungimea de unda [Å]",button1.x+20,input1.y-25);
  text("pozitia liniei [div.]",button1.x+20,input1.y+5);
  text(" Date Mercur",button1.x+20,input1.y-50);
}

fill(220, 200, 200);
noStroke();
rect(0,0,20,height);
rect(0.5*width+20,0,0.1*width,height);
fill(0);
noStroke();
textSize(12);
text("Student: ", Student.x-55,Student.y+5);
text("Grupa: ", Grupa.x-50,Grupa.y+5);
Tabel();

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizica si Chimie, Universitatea Tehnică din Cluj-  
Napoca",130, 689);

if(ImgArataGrafic == 'Grafic Offline')
{
  //input1.hide();
  Graf.show();
}
else
{
  //input1.show();
  Graf.hide();
}
fill(0);

```

```
}

```

```
function ScalaNe(yCorr)
{
  push();
  translate(402,0.31*height);
  rotate(Math.PI*0.5);
  translate(-478,24);
  dy0 = 15;
  dy1 = 5*dy0;
  dy2 = 10*dy0;
  dy3 = 0.5*dy0;
  let salt = -20;
  strokeWeight(1);
  //stroke(220,0,220);
  //line(100,0.5*height+yCorr,200,0.5*height+yCorr);
  stroke(255);
  for(let i=-15; i<40; i++)
  {
    line(402,0.31*height+yCorr-i*dy0+salt,430,0.31*height+yCorr-i*dy0+salt);
  }

  for(let i=-30; i<80; i++)
  {
    line(402,0.31*height+yCorr-i*dy3+salt,420,0.31*height+yCorr-i*dy3+salt);
  }

  for(let i=-8; i<9; i++)
  {
    line(402,0.31*height+yCorr-i*dy1+salt,440,0.31*height+yCorr-i*dy1+salt);
  }
  strokeWeight(2);
  for(let i=-2; i<5; i++)
  {
    line(402,0.31*height+yCorr-i*dy2+salt,450,0.31*height+yCorr-i*dy2+salt);
  }
  noStroke();
  fill(255);
  textSize(14);
  rotate(-Math.PI*0.5);
  for(let i=-1; i<4; i++)
  {
    text(nf(100*i+400,1,0)+" div.",i*dy2-220-yCorr,250+0.31*height);
  }

  pop();
}

```

```
function ScalaHg(yCorr)
{
  push();
  translate(402,0.31*height);
  rotate(Math.PI*0.5);
  translate(-478,98);
  dy0 = 17.4;
  dy1 = 5*dy0;

```

```

dy2 = 10*dy0;
dy3 = 0.5*dy0;
let salt = -20;
strokeWeight(1);
//stroke(220,0,220);
//line(100,0.5*height+yCorr,200,0.5*height+yCorr);
stroke(255);
for(let i=-15; i<40; i++)
{
  line(402,0.31*height+yCorr-i*dy0+salt,430,0.31*height+yCorr-i*dy0+salt);
}

for(let i=-30; i<80; i++)
{
  line(402,0.31*height+yCorr-i*dy3+salt,420,0.31*height+yCorr-i*dy3+salt);
}

for(let i=-8; i<9; i++)
{
  line(402,0.31*height+yCorr-i*dy1+salt,440,0.31*height+yCorr-i*dy1+salt);
}
strokeWeight(2);
for(let i=-2; i<5; i++)
{
  line(402,0.31*height+yCorr-i*dy2+salt,450,0.31*height+yCorr-i*dy2+salt);
}
noStroke();
fill(255);
textSize(14);
rotate(-Math.PI*0.5);
for(let i=-1; i<4; i++)
{
  text(nf(100*i+400,1,0)+" div.",i*dy2-220-yCorr,250+0.31*height);
}

pop();
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laborator Fizica: Etalonarea unui Spectroscop", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.6*width-20, height-45);
  rect(0.6*width, 15,0.4*width-10, height-25);
}

function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {

```

```

    ImgArataGrafic='Grafic Offline';
    button1.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    button1.html(ImgArataGrafic);
  }
}

function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Etalonare spectroscop (Ne): De reprezentat pe hartie milimetrica',
    xaxis:
    {
      title:'Pozitie [div.]',
      visible: 1
    },
    yaxis:
    {
      title:' $\lambda$  [Å]'
    },
    paper_bgcolor: "#c4dedc"
  };

  var data = [trace1,trace2];
  //var data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

function LinFit(x,y,N)
{
  var f, s, sx, sy, sxx, sxy;
  var sigmy = [N+1];
  for (var i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(var i=1; i<=N; i++)
  {

```



```

    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
    sx += x[i]*f;
    sxx+= x[i]*x[i]*f;
    sy += y[i]*f;
    sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(var i=1; i<=N; i++)
{
    eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}

}

function Culoare(x)
{
    r_o = int(130*exp(-pow(abs(x-14)/15.94,2.6))+50*exp(-pow(abs(x+6.375)/9.56,2.6))+115.0/(1+exp((x-235.875)/9.56))+255.0/(1.0+exp((114.75-x)/6.375))-115.0);
    g_o = int(255*exp(-pow(abs(x-102)/51.0,6)));
    b_o = int(255*exp(-pow(abs(x-38.25)/38.25,8)));
    //console.log(r_o);
}

function Neon()
{
    ArataSpNe = 1;
    ArataSpHg = 0;
    button1.show();
    input1.show();
    input2.show();
    input3.show();
    input4.show();
    input5.show();
    input6.show();
    input7.show();
    input8.show();
    input9.show();

    rSlider.show();

    input11.hide();
    input12.hide();
    input13.hide();
    input14.hide();
    input15.hide();
    input16.hide();
    input17.hide();
    input18.hide();
    input19.hide();

    input21.hide();
    input22.hide();
    input23.hide();
    input24.hide();
    input25.hide();
}

```

```
input26.hide();
input27.hide();
input28.hide();
input29.hide();
```

```
}
```

```
function Mercur()
```

```
{
  ArataSpNe = 0;
  ArataSpHg = 1;
  input11.show();
  input12.show();
  input13.show();
  input14.show();
  input15.show();
  input16.show();
  input17.show();
  input18.show();
  input19.show();
```

```
input21.show();
input22.show();
input23.show();
input24.show();
input25.show();
input26.show();
input27.show();
input28.show();
input29.show();
```

```
input1.hide();
input2.hide();
input3.hide();
input4.hide();
input5.hide();
input6.hide();
input7.hide();
input8.hide();
input9.hide();
```

```
rSlider.hide();
```

```
}
```

```
function Documentatie()
```

```
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Spectroscopul/Doc/Analiza%20Spectra
la.pdf';
}
```

13.Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină

```

let img1, img2, img3, img4, img5, img6, img7, img8, img9, img10, img11, img12, img13, imgScala;
let img14, img15, img16, img17, img18;
let ImgArataGrafic, ImgArataGrafic1;
let PondereSmooth = [];
var Volti = [ 4, 6, 8, 10, 12, 14, 16, 18, 20];
let Grade = [ 0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360];
let V_Grd = [14, 18, 21, 20, 17, 11, 8, 16, 18, 20, 17, 12, 14];
let Grad;
let NrMasuratoriInt, NrMasuratoriCamp;
let TipMasuratoare;
let BecIntensity;
let DistBecToBec, Ex, E0;
let dist = 30;
let Rx = [];
let R0 = [];
let R_CpL = [];
let Ix;
let IntNec = [];

```

```

let panta, intersectia, eroarea;

```

```

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};
/*
var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};
*/

```

```

var trace2 = {
  r: [],
  theta: [],
  mode: 'lines',
  name: 'Cardioid',
  line: {color: 'darkviolet'},
  type: 'scatterpolar'
};

```

```

function preload()
{
  img1 = loadImage("Img/ripla.jpg");
  img2 = loadImage("Img/BecAprins.jpg");
  img3 = loadImage("Img/BecAprinsInvers.jpg");
  img4 = loadImage("Img/Voltmetru.jpg");
  img5 = loadImage("Img/SenzLumina.jpg");

```

```

img6 = loadImage("Img/AnsambluExperimental.jpg");
img7 = loadImage("Img/AE1.jpg");
img8 = loadImage("Img/AE2.jpg");
img9 = loadImage("Img/AE3.jpg");
img10 = loadImage("Img/AE4.jpg");
img11 = loadImage("Img/AE5.jpg");
img12 = loadImage("Img/AE6.jpg");
img13 = loadImage("Img/Calcul.jpg");
img14 = loadImage("Img/CpL0gr.jpg");
img15 = loadImage("Img/CpL30gr_n.jpg");
img16 = loadImage("Img/CpL30gr_c.jpg");
img17 = loadImage("Img/CpL180gr.jpg");
img18 = loadImage("Img/BecTop.jpg");

}

function setup()
{

  createCanvas(1400, 700);
  background(220, 200, 200);
  NrMasuratoriInt = Volti.length;
  NrMasuratoriCamp = Grade.length;
  BecIntensity = 200;
  Grad = 0;

  for (let i=0; i<NrMasuratoriInt; i++)
  {
    let DistX = new Imput(200,height-NrMasuratoriInt*dist+i*dist-5);
    DistX.hide();
    Rx.push(DistX);
    let Dist0 = new Imput(270,height-NrMasuratoriInt*dist+i*dist-5);
    Dist0.hide();
    R0.push(Dist0);
    let IntX = new Imput(340,height-NrMasuratoriInt*dist+i*dist-5);
    IntX.hide();
    IntNec.push(IntX);
  }

  for (let i=0; i<NrMasuratoriCamp; i++)
  {
    let DistCamp = new Imput(112+50*i, height-50);
    DistCamp.hide();
    R_CpL.push(DistCamp);
  }

  Student = createInput("");
  Student.position(0.8*width+10, 20);
  Student.style('width', '150px');
  Student.style('font-size', '12px');
  Student.style('text-align', 'center');

  Grupa = createInput("");
  Grupa.position(Student.x+Student.width+55, Student.y);
  Grupa.style('width', '50px');
  Grupa.style('font-size', '12px');
  Grupa.style('text-align', 'center');

  ImgArataGrafic = 'Grafic Online';
  ImgArataGrafic1 = 'Grafic 1 Online';

```

```
button0 = createButton("Documentație laborator");
button0.position(110, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '190px');
button0.mouseClicked(Documentatie);

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(IntNec[0].x+100, IntNec[0].y-dist*1.5-5);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);
buttonGraf.hide();

buttonGraf1 = createButton(ImgArataGrafic1);
buttonGraf1.position(IntNec[0].x+400, IntNec[0].y+dist*1.5);
buttonGraf1.addClass("button button-primary button-border-thick button-box");
buttonGraf1.style('width', '150px');
buttonGraf1.mouseClicked(ArataGrafic1);
buttonGraf1.hide();

buttonInt = createButton("Intensitate luminoasă");
buttonInt.position(button0.x+button0.width+10, button0.y);
buttonInt.addClass("button button-primary button-border-thick button-box");
buttonInt.style('width', '150px');
buttonInt.mouseClicked(Intensitate);

buttonCamp = createButton("Câmp Luminos");
buttonCamp.position(buttonInt.x+buttonInt.width+10, buttonInt.y);
buttonCamp.addClass("button button-primary button-border-thick button-box");
buttonCamp.style('width', '150px');
buttonCamp.mouseClicked(CampLuminos);

buttonPlus = createButton("Creste 30 grade");
buttonPlus.position(buttonInt.x-50, R_CpL[2].y-200);
buttonPlus.addClass("button button-primary button-border-thick button-box");
buttonPlus.style('width', '150px');
buttonPlus.mouseClicked(Creste30);
buttonPlus.hide();

buttonMinus = createButton("Scade 30 grade");
buttonMinus.position(buttonPlus.x, buttonPlus.y+40);
buttonMinus.addClass("button button-primary button-border-thick button-box");
buttonMinus.style('width', '150px');
buttonMinus.mouseClicked(Scade30);
buttonMinus.hide();

InputVolti = createInput(2);
InputVolti.position(button0.x-76, button0.y+25);
InputVolti.style('width', '50px');
InputVolti.style('font-size', '12px');
InputVolti.style('text-align', 'center');
InputVolti.hide();

console.log(Rx);

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x-10, buttonGraf.y+25);
Graf.style('width', '395px');
```

```

Graf.style('height', '295px');
Graf.hide();

Graf1 = createDiv('Grafic 1 aici');
Graf1.id('Graf1');
Graf1.position(buttonGraf1.x+150, buttonGraf1.y-3.8*dist);
Graf1.style('width', '400px');
Graf1.style('height', '300px');
Graf1.hide();

DistBecToBec = 30;
rSlider = createSlider(0, DistBecToBec, 0.5*DistBecToBec, 0.001*DistBecToBec);
rSlider.position(Rx[0].x-120, Rx[0].y-150);
rSlider.style('width', '500px');
rSlider.style('colour', 'red');
rSlider.hide();

for (let i = 0; i<NrMasuratoriInt; i++)
{
  trace1.x[i] = Volti[i];
  trace1.y[i] = Volti[i];
  //x [i] = trace1.x[i];
}

for (let i = 0; i<NrMasuratoriCamp; i++)
{
  trace2.r[i] = Grade[i];
  trace2.theta[i] = Grade[i];
}

console.log(trace1);
TipMasuratoare = 0;
}

function draw()
{
  background(220, 200, 200);
  textAlign (LEFT)
  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55,Student.y+5);
  text("Grupa: ", Grupa.x-50,Grupa.y+5);
  text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",820, 680);
  textAlign (LEFT, CENTER)
  textSize(22);
  text("Laboratorul de Fizică: Studiul intensității luminoase și al câmpului luminos ale unei surse de lumină", 10,
25);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,1*width-20, height-45);

  if (TipMasuratoare == 1)

```

```

{
  MasorIntNecunoscuta();
  if(ImgArataGrafic == 'Grafic Offline')
  {
    for (let i = 0; i<NrMasuratoriInt; i++)
    {
      trace1.y[i] = IntNec[i].MyInput.value();
      //trace1.y[0] = Volti[i];
    }
    Graf.show();
    Grafic();
  }
  else
  {
    Graf.hide();
  }
  Graf1.hide();
}

if (TipMasuratoare == 2)
{
  MasorCampLuminos();
  if(ImgArataGrafic1 == 'Grafic 1 Offline')
  {
    for (let i = 0; i<NrMasuratoriCamp; i++)
    {
      trace2.r[i] = R_CpL[i].MyInput.value();
      //trace1.y[0] = Volti[i];
    }
    Graf1.show();

    //LinFit(x,y,8);

    Grafic1();
  }
  else
  {
    Graf1.hide();
  }
  Graf.hide();
}

fill(0);
}

function MasorIntNecunoscuta()
{
  image(img6, width*0.5, 50, width*0.20, height*0.4-10);
  image(img7, width*0.69, 50, width*0.15, height*0.4-10);
  image(img8, width*0.84, 50, width*0.15, height*0.4-10);
  image(img9, width*0.5, height*0.42, width*0.25, height*0.275);
  image(img10, width*0.74, height*0.42, width*0.25, height*0.275);
  image(img11, width*0.5, height*0.68, width*0.25, height*0.275);
  image(img12, width*0.74, height*0.68, width*0.25, height*0.275);
  buttonGraf.show();
  rSlider.show();
  strokeWeight(3);
  InputVolti.show();
}

```

```

let R_x = rSlider.value();
let R_0 = DistBecToBec - R_x;

BecIntensity=map(InputVolti.value(),0,20,256,0);
Ex = 10*IntX(InputVolti.value())/R_x/R_x;
E0 = 120/R_0/R_0;
noFill();
stroke(0);
rect(Rx[0].x-185,Rx[0].y-dist*1.5-10,400,dist*(NrMasuratoriInt+1.5));
strokeWeight(2);
rect(Rx[0].x-185,Rx[0].y-dist*1.5-10,400,dist*1.25);
rect(Rx[0].x-100,Rx[0].y-dist*1.5-10,70,dist*(NrMasuratoriInt+1.5));

image(img1, rSlider.x-27, rSlider.y+10, rSlider.width+45, 60);
image(img2, rSlider.x-69, rSlider.y-80, 90, 60);
image(img3, rSlider.x-80+rSlider.width+45, rSlider.y-80, 100, 60);
image(img4, button0.x-95, button0.y+10, 80, 135);

fill(220, 200, 200,BecIntensity);
noStroke();
rect(rSlider.x-69, rSlider.y-80, 100, 60);

fill(220, 200, 200,map(10,0,20,256,0));
rect(rSlider.x-80+rSlider.width+45, rSlider.y-80, 100, 60);

fontSize(16);
fill(0);
textAlign(LEFT,CENTER);
text("I [Cd]",Rx[0].x-165,Rx[0].y-dist*1.2);
text("U [V]",Rx[0].x-80,Rx[0].y-dist*1.2);
text("R [cm]",Rx[0].x-5,Rx[0].y-dist*1.2);
text("R [cm]",R0[0].x-5,R0[0].y-dist*1.2);
text("I [Cd]",IntNec[0].x-5,IntNec[0].y-dist*1.2);
let pozCursor = R_x/DistBecToBec*rSlider.width;
fontSize(14);
text("12",Rx[0].x-150,Rx[0].y+dist*0.45*NrMasuratoriInt);
fontSize(10);
text("0",Rx[0].x-160,Rx[0].y-dist*0.98);
text("0 ",Rx[0].x+77,Rx[0].y-dist*0.98);
text("x ",Rx[0].x+8,Rx[0].y-dist*0.98);
text("x ",Rx[0].x+140,Rx[0].y-dist*0.98);
fontSize(12);
text("E = "+nf(Ex,1,2),rSlider.x+pozCursor-55,rSlider.y-100);
text("E = "+nf(E0,1,2),rSlider.x+pozCursor+5,rSlider.y-100);
fontSize(9);
text("x ",rSlider.x+pozCursor-47,rSlider.y-95);
text("0 ",rSlider.x+pozCursor+13,rSlider.y-95);

image(img5, rSlider.x+pozCursor-30,rSlider.y-90,20,80);
image(img5, rSlider.x+pozCursor,rSlider.y-90,20,80);
image(img13, Graf.x,Graf.y+10,350,230);

for (let i=0; i<NrMasuratoriInt; i++)
{
  Rx[i].show();
  R0[i].show();
  IntNec[i].show();
  textSize(14);
  text(nf(Volti[i],1,0),Rx[0].x-75,height-NrMasuratoriInt*dist+i*dist+0);
}

```



```

}

function IntX(x)
{
  return 2+x*x*x*x/1000;
}

function MasorCampLuminos()
{
  image(img14, width*0.55, 50, width*0.2, height*0.3-10);
  image(img15, width*0.78, 50, width*0.2, height*0.3-10);
  if(ImgArataGrafic1 == 'Grafic 1 Online')
  {
    image(img16, width*0.55,height*0.36, width*0.2, height*0.3-10);
    image(img17, width*0.78,height *0.36 , width*0.2, height*0.3-10);
  }

  image(img1, rSlider.x-28, rSlider.y+10, rSlider.width+45, 60);
  let R_x = rSlider.value();
  BecIntensity=map(V_Grd[Grad],0,20,256,0);
  Ex = 10*IntX(V_Grd[Grad])/R_x/R_x;
  let pozCursor = R_x/DistBecToBec*rSlider.width;
  image(img5, rSlider.x+pozCursor-15,rSlider.y-90,20,80);
  fill(0);
  textSize(12);
  noStroke();
  textAlign(CENTER,CENTER);
  text("E = "+nf(Ex,1,2),rSlider.x+pozCursor,rSlider.y-100);
  textSize(9);
  text("x ",rSlider.x+pozCursor-10,rSlider.y-95);
  textSize(14);
  text("Unghi = "+nf(Grade[Grad],1,0),buttonPlus.x+0.5*buttonPlus.width,buttonPlus.y+25);

  for (let i=0; i<NrMasuratoriCamp; i++)
  {
    text(nf(Grade[i],1,0),R_CpL[i].x+15, R_CpL[i].y-38);
  }

  strokeWeight(2);
  stroke(0);
  noFill();
  rect(R_CpL[0].x-95,R_CpL[0].y-dist*1.5-15,740,80);
  rect(R_CpL[0].x-95,R_CpL[0].y-dist*1.5-15,740,40);
  rect(R_CpL[0].x-95,R_CpL[0].y-dist*1.5-15,80,80);
  fill(0);
  textSize(12);
  noStroke();
  text("Unghi [grade]",R_CpL[0].x-55, R_CpL[0].y-40);
  text("Distanța [cm]",R_CpL[0].x-55, R_CpL[0].y);

  push();
  translate(rSlider.x-10, rSlider.y-50);
  rotate(Grade[Grad]*3.1415/180.0);
  imageMode(CENTER);
  image(img18, 5, 0 , 80, 80);
  pop();
}

function ArataGrafic()

```

```

{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

function ArataGrafic1()
{
  if (ImgArataGrafic1 == 'Grafic 1 Online')
  {
    ImgArataGrafic1='Grafic 1 Offline';
    buttonGraf1.html(ImgArataGrafic1);
  }
  else
  {
    ImgArataGrafic1 = 'Grafic 1 Online';
    buttonGraf1.html(ImgArataGrafic1);
  }
}

function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Ix = f(U) : De reprezentat in Excel',
    xaxis:
    {
      title:'Tensiune pe bec [V]',
      visible: 1
    },
    yaxis:
    {
      title:'Ix [Cd]'
    },
    paper_bgcolor: "#c4dedc"
  };

  let data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

```

```
function Grafic1()
{
  Graf1.html("");

  GRAFIC = document.getElementById('Graf1');

  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title: 'Curba de egala iluminare',
    xaxis:
    {
      title: 'grade',
      visible: 1
    },
    yaxis:
    {
      title: 'Dist. [cm]'
    },
    paper_bgcolor: "#c4dedc"
  };

  let data = [trace2];

  Plotly.newPlot( GRAFIC, data, layout);
}
```

```
function Intensitate()
{
  TipMasuratoare = 1;
  for (let i=0; i<NrMasuratoriInt; i++)
  {
    Rx[i].show();
    R0[i].show();
    IntNec[i].show();
  }
  for (let i=0; i<NrMasuratoriCamp; i++)
  {
    R_CpL[i].hide();
  }
  InputVolti.show();
  buttonGraf1.hide();
  buttonGraf.show();
  buttonPlus.hide();
  buttonMinus.hide();
}
```

```
function CampLuminos()
{
  TipMasuratoare = 2;
  for (let i=0; i<NrMasuratoriCamp; i++)
  {
```

```

    R_CpL[i].show();
  }
  for (let i=0; i<NrMasuratoriInt; i++)
  {
    Rx[i].hide();
    R0[i].hide();
    IntNec[i].hide();
  }
  InputVolti.hide();
  buttonGraf.hide();
  buttonGraf1.show();
  rSlider.show();
  buttonPlus.show();
  buttonMinus.show();
}

function Creste30()
{
  if (Grad == Grade.length-1)
  {
    Grad = 0;
  }
  Grad++;
}

function Scade30()
{
  if (Grad == 0)
  {
    Grad = Grade.length-1;
  }
  Grad--;
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/SIM/2019-2020/SRTP/Laboratoare/SRTP-L5-
Determinarea Intensitatii Luminoase.pdf';
}

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
    sx += x[i]*f;
    sxx+= x[i]*x[i]*f;
    sy += y[i]*f;
  }
}

```

```
sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

14. Studiul efectului fotoelectric și determinarea constantei lui Planck

```

let img1, img2, img3;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Int1 = [];
let Int2 = [];
let MarimeFizica3 = [];
let Uuri = [-0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5];
let Int0Mas1 = [1, 2, 3, 4, 6, 10, 17, 25, 33, 40, 48, 56, 63];
let Int0Mas2 = [1, 1.25, 1.5, 2, 3, 5, 8, 15, 22, 30, 38, 47, 54];
let IntMas1 = [];
let IntMas2 = [];
let Electroni = [];
let Student, Grupa;
var Contor0, Contor;
let CresteTens = false;
let Temp, LungimeDeUnda, Lumineaza;

```

```

let Uf1, Uf2;
let S1, S2;
let h, eroarerel;

```

```

let panta, intersectia, eroarea;

```

```

let TipMasuratoare;

```

```

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp. λ1'
};

```

```

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara λ1'
};

```

```

var trace3 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp. λ2'
};

```

```

var trace4 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara λ2'
};

```

```

function preload()
{
  img1 = loadImage("Img/1.jpg");
  img2 = loadImage("Img/StandExp.jpg");
  img3 = loadImage("Img/becNou.jpg");
}

function setup()
{
  createCanvas(1400, 700);
  background(240, 240, 140);
  NrMasuratori = Uuri.length;
  LungimeDeUnda = 1;
  Lumineaza = false;
  dist = 40;

  for (let i=0; i<NrMasuratori; i++)
  {
    if(i == 7)
    {
      let MF2 = new Imput((NrMasuratori)*dist+i*dist,height-77,'15');
      Int2.push(MF2);
      let MF1 = new Imput((NrMasuratori)*dist+i*dist,height-127,'25');
      Int1.push(MF1);
    }
    else{
      let MF2 = new Imput((NrMasuratori)*dist+i*dist,height-77,' ');
      Int2.push(MF2);
      let MF1 = new Imput((NrMasuratori)*dist+i*dist,height-127,"");
      Int1.push(MF1);
    }
    trace1.x[i] = Uuri[i];
    trace3.x[i] = Uuri[i];
    IntMas1[i] = Int0Mas1[i]/8+0.05*random(-1,1);
    IntMas2[i] = Int0Mas2[i]/6+0.05*random(-1,1);
  }
  for (let i=5; i<NrMasuratori; i++)
  {
    trace1.x[i-5] = Uuri[i];
    trace3.x[i-5] = Uuri[i];
  }

  S1 = new Imput(Int2[0].x+555,Int2[1].y-15,"");
  S2 = new Imput(Int2[0].x+605,Int2[1].y-15,"");
  Uf1 = new Imput(Int2[0].x+655,Int2[1].y-15,"");
  Uf2 = new Imput(Int2[0].x+705,Int2[1].y-15,"");
  h = new Imput(Int2[0].x+755,Int2[1].y-15,"");
  eroarere1 = new Imput(Int2[0].x+805,Int2[1].y-15,"");

  button0 = createButton("Documentație Laborator");
  button0.position(35, 100);
  button0.addClass("button button-primary button-border-thick button-box");
  button0.style('width', '200px');
  button0.mouseClicked(Documentatie);

  buttonMasor = createButton("Experiment");
  buttonMasor.position(button0.x+25, button0.y+35);
  buttonMasor.addClass("button button-primary button-border-thick button-box");

```

```
buttonMasor.style('width', '150px');
buttonMasor.mouseClicked(FunctieButon);

Student = createInput("");
Student.position(button0.x+50, button0.y-50);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x, Student.y+25);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(Int1[0].x+0.53*width-260, Int2[0].y-270);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

buttonlamda1 = createButton("λ1=6250 Å");
buttonlamda1.position(Int1[0].x-430, Int2[0].y-400);
buttonlamda1.addClass("button button-primary button-border-thick button-box");
buttonlamda1.style('width', '90px');
buttonlamda1.mouseClicked(SetL1);

buttonlamda2 = createButton("λ2=6896 Å");
buttonlamda2.position(Int1[0].x-430, Int2[0].y-370);
buttonlamda2.addClass("button button-primary button-border-thick button-box");
buttonlamda2.style('width', '90px');
buttonlamda2.mouseClicked(SetL2);

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+30);
Graf.style('width', '400px');
Graf.style('height', '300px');

buttonStart = createButton("Start");
buttonStart.position(button0.x, button0.y+70);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '90px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y+0);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '90px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

buttonSens1 = createInput("");
buttonSens1.position(button0.x+637, button0.y+503);
buttonSens1.style('width', '10px');
buttonSens1.style('font-size', '8px');
buttonSens1.style('text-align', 'center');
```



```

buttonSens2 = createInput("");
buttonSens2.position(button0.x+687, button0.y+503);
buttonSens2.style('width', '10px');
buttonSens2.style('font-size', '8px');
buttonSens2.style('text-align', 'center');

buttonh = createInput("");
buttonh.position(button0.x+842, button0.y+503);
buttonh.style('width', '12px');
buttonh.style('font-size', '8px');
buttonh.style('text-align', 'center');

USlider = createSlider(0, NrMasuratori-1, 7, 1);
USlider.position(Int1[0].x-150, Int2[0].y-130);
USlider.style('width', '350px');
USlider.style('colour', 'red');
USlider.hide();

Fanta = createSlider(0, 15, 0, 0.1);
Fanta.position(Int1[0].x-150, 60);
Fanta.style('width', '150px');
Fanta.style('colour', 'red');
Fanta.hide();

}

function draw()
{
  background(240, 200, 110);
  image(img1, width*0.71, 30, width*0.27, height*0.4);
  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);

  if(Masor)
  {
    fill(255);
    noStroke();
    rect(width*0.2, 40, width*0.5-100, 100);
    image(img2, width*0.2, 80, width*0.5-100, height*0.7);
    let TensU = Uuri[int(USlider.value())];
    fill(0);
    noStroke();
    textSize(16);
    fill(220,0,0);
    text(nf(TensU,1,1)+' V',USlider.x+18,USlider.y-127);
    text('Reglează Tensiunea',USlider.x+10, USlider.y-8);

  }

  if (Lumineaza)
  {
    let IntDiv;
    EfFotovoltaic();
    fill(0,0,250);
    textSize(16);
    if (LungimeDeUnda == 1)
    {
      IntDiv = IntMas1[int(USlider.value())]*Fanta.value();
    }
  }
}

```

```

    }
    if (LungimeDeUnda == 2)
    {
        IntDiv = IntMas2[int(USlider.value())]*Fanta.value();
    }
    text(nf(IntDiv,1,1)+'div',0.55*width-36,0.5*height-17);
    text('Reglează Fanta',USlider.x+10, USlider.y-440);
    text("1 div=9*10-9 A",Int1[0].x+255,Int2[0].y-330);
    text("-9 ",Int1[0].x+332,Int2[0].y-340);

    }
    FotoElectroni();
    }
    Tabel();
    for (let i = 5; i < NrMasuratori; i++)
    {
        trace1.y[i-5] = Int1[i].MyInput.value();
        trace3.y[i-5] = Int2[i].MyInput.value();
    }
    LinFit(trace1.x,trace1.y,NrMasuratori-6);
    var Minim = 0.9*min(trace1.x);
    var Maxim = 1.1*max(trace1.x);
    var NrFit = 101;
    var dx = (Maxim-Minim)/(NrFit-1.0);
    for (var i = 0; i < NrFit; i++)
    {
        trace2.x[i] = Minim+i*dx;
        trace2.y[i] = intersectia+panta*trace2.x[i];
    }
    LinFit(trace3.x,trace3.y,NrMasuratori-6);
    var Minim = 0.9*min(trace3.x);
    var Maxim = 1.1*max(trace3.x);
    var NrFit = 101;
    var dx = (Maxim-Minim)/(NrFit-1.0);
    for (var i = 0; i < NrFit; i++)
    {
        trace4.x[i] = Minim+i*dx;
        trace4.y[i] = intersectia+panta*trace4.x[i];
    }
    //Grafic();

    if(ImgArataGrafic == 'Grafic Offline')
    {
        Graf.show();
        Grafic();
    }
    else
    {
        Graf.hide();
    }
    }

function FotoElectroni()
{
    for(let i = 0; i < int(Fanta.value());i++)
    {
        let pozX = 0.35*width+15+random(-10,10);
        let pozY = 0.25*height+10+random(-10,10);
        let el = new Electron(pozX,pozY);
        if(Electroni.length < 200)

```

```

    {
      Electroni.push(e1);
    }

  }

  for(let i = 0; i< Electroni.length;i++)
  {
    Electroni[i].move();
    Electroni[i].vx += Uuri[int(USlider.value())]*0.1;
    Electroni[i].show();
    if (Electroni[i].isOffScreen())
    {
      Electroni.splice(i,1);
      i--;
    }
  }
}

function EfFototelectric()
{
  push();
  translate(0.45*width-10,110);
  rotate(-Math.PI/180.0*132);
  image(img3, 0, 0, 50, 50);
  if (LungimeDeUnda == 1)
  {
    fill(255,202,24,100);
  }
  if (LungimeDeUnda == 2)
  {
    fill(251,69,59,200);
  }
  rect(+25-Fanta.value(),-148,2*Fanta.value(),143);
  stroke(0);
  strokeWeight(1);
  rect(0,-10,50,5);
  fill(50);
  rect(Fanta.value()+25,-15,30,5);
  rect(-Fanta.value()-5,-15,30,5);

  pop();
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de Fizica: Studiul efectului fotoelectric si determinarea constantei lui Planck. ", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-52);
  rect(0.7*width, 15,0.3*width-10, height-33);
  for (var i=0; i<16; i++)

```

```

{
  line(Int1[0].x-500+i*40,Int2[0].y-45,Int1[0].x-500+i*40,height-20);
}
line(Int2[0].x-500,Int2[0].y-45,Int2[0].x+100,Int2[0].y-45);
line(Int2[0].x-500,height-20,Int2[0].x+100,height-20);
line(Int2[0].x-500,Int2[0].y+5,Int2[0].x+100,Int2[0].y+5);
line(Int2[0].x-500+40,Int2[0].y-20,Int2[0].x+100,Int2[0].y-20);
line(Int2[0].x-500+40,Int2[0].y+28,Int2[0].x+100,Int2[0].y+28);

for (var i=0; i<7; i++)
{
  line(Int2[0].x+130+i*50,Int2[0].y-45,Int2[0].x+130+i*50,height-20);
}
line(Int2[0].x+130,Int2[0].y-45,Int2[0].x+430,Int2[0].y-45);
line(Int2[0].x+130,Int2[0].y+5,Int2[0].x+430,Int2[0].y+5);
line(Int2[0].x+130,height-20,Int2[0].x+430,height-20);

fill(0);
noStroke();
textSize(12);

// Capul de tabel
text(" λ ",Int1[0].x-492,Int2[0].y-15);
text(" 1 ",Int1[0].x-485,Int2[0].y-11);
text("U",Int1[0].x-450,Int2[0].y-28);
text("[V]",Int1[0].x-440,Int2[0].y-28);
text("I",Int1[0].x-450,Int2[0].y-2);
text("[div]",Int1[0].x-445,Int2[0].y-2);

text(" λ ",Int1[0].x-492,Int2[0].y+35);
text(" 2 ",Int1[0].x-485,Int2[0].y+39);
text("U",Int1[0].x-450,Int2[0].y+23);
text("[V]",Int1[0].x-440,Int2[0].y+23);
text("I",Int1[0].x-450,Int2[0].y+47);
text("[div]",Int1[0].x-445,Int2[0].y+47);

for (let i = 0; i<NrMasuratori; i++)
{

  text(nf(Uuri[i],1,1),Int1[0].x+i*dist-410,Int2[0].y-28);
  text(nf(Uuri[i],1,1),Int2[0].x+i*dist-410,Int2[0].y+23);

}

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-  
Napoca",833, 693);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text(" S ",Int1[0].x+145,Int2[0].y-30);
text(" 1 ",Int1[0].x+152,Int2[0].y-25);
text("[10 A/V]",Int1[0].x+131,Int2[0].y-5);

```

```

text(" S ",Int1[0].x+198,Int2[0].y-30);
text(" 2 ",Int1[0].x+206,Int2[0].y-25);
text("[10 A/V]",Int1[0].x+181,Int2[0].y-5);
text("U",Int1[0].x+245,Int2[0].y-30);
text(" f1",Int1[0].x+250,Int2[0].y-25);
text("[V]",Int1[0].x+248,Int2[0].y-5);
text("U",Int1[0].x+295,Int2[0].y-30);
text("f2",Int1[0].x+303,Int2[0].y-25);
text("[V]",Int1[0].x+298,Int2[0].y-5);
text("h",Int1[0].x+350,Int2[0].y-30);
text("[10 Js]",Int1[0].x+333,Int2[0].y-5);
text("Δh/h",Int1[0].x+393,Int2[0].y-30);
text("[%]",Int1[0].x+398,Int2[0].y-5);

```

```

}

```

```

function ArataGrafic()

```

```

{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

```

```

function Grafic()

```

```

{
  Graf.html("");

```

```

GRAFIC = document.getElementById('Graf');

```

```

//console.log(trace1[0].y[4]);

```

```

var layout =

```

```

{
  margin:
  {
    t: 50,
    b: 50,
    l: 50,
    r: 50
  },
  title:'I = f(U)',
  xaxis:
  {
    title:'U [V]',
    visible: 1
  },
  yaxis:
  {
    title:'I [div]'
  },
  paper_bgcolor: "#e2d3b0"
};

```

```

var data = [trace1,trace2,trace3,trace4];

```

```
//let data = [trace1];

Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2019-2020/lucrari/SemII/L21Ro_Studiul%20electului%20fotoelectric_Planck.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
  USlider.show();
  Fanta.show();
}

function Start()
{
  Lumineaza = true;
}

function Stop()
{
  Lumineaza = false;
}

function SetL1()
{
  LungimeDeUnda = 1;
  console.log(LungimeDeUnda);
}

function SetL2()
{
  LungimeDeUnda = 2;
  console.log(LungimeDeUnda);
}

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
    sx += x[i]*f;
  }
}
```

```
sxx+= x[i]*x[i]*f;
sy += y[i]*f;
sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
    eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

15.Studiul polarizării luminii

```
let img1,img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Student, Grupa;
let d, s;
let Int, Int0;
let MarimeFizica1 = [];
let MarimeFizica2 = [];
var Iuri = [0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360];
var Yuri = [];
var Eroare = [];

let TipMasuratoare;

var trace1 = {
  r: [],
  theta: [],
  mode: 'lines',
  name: 'Date experimentale',
  line: {color: 'orange'},
  type: 'scatterpolar'
};

var trace2 = {
  r: [],
  theta: [],
  mode: 'scatter',
  name: 'Date teoretice',
  line: {color: 'darkviolet'},
  type: 'scatterpolar'
};

var trace3 = {
  r: [],
  theta: [],
  mode: 'lines',
  name: 'Legea lui Malus',
  line: {color: 'lightred'},
  type: 'scatterpolar'
};
function preload()
{

  img1 = loadImage("Img/Montaj_experimental.jpg");
  img2 = loadImage("Img/Bec.GIF");
```



```
}

function setup()
{
  createCanvas(1400, 700);
  background(220, 200, 200);
  NrMasuratori = Iuri.length;
  dist = 25;

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(100+i*65,height-70);
    MarimeFizica1.push(MF1);
    let MF2 = new Input(100+i*65,height-40);
    MarimeFizica2.push(MF2);
    Yuri[i] = 1;
  }

  for (let i=0; i<=360; i++)
  {
    let Err = random(-1,1);
    Eroare.push(Err);
  }

  Student = createInput("");
  Student.position(0.5*width+10, 18);
  Student.style('width', '150px');
  Student.style('font-size', '12px');
  Student.style('text-align', 'left');

  Grupa = createInput("");
  Grupa.position(Student.x+Student.width+60, Student.y);
  Grupa.style('width', '50px');
  Grupa.style('font-size', '12px');
  Grupa.style('text-align', 'center');

  button0 = createButton("Documentatie Laborator");
  button0.position(30, 50);
  button0.addClass("button button-primary button-border-thick button-box");
  button0.style('width', '200px');
  button0.mouseClicked(Documentatie);

  buttonGB = createButton("Masor");
  buttonGB.position(button0.x+button0.width+20, button0.y);
  buttonGB.addClass("button button-primary button-border-thick button-box");
  buttonGB.style('width', '150px');
  buttonGB.mouseClicked(FunctieButon);
```

```

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(width*0.71, height*0.5);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

Graf = createDiv(' ');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
Graf.style('height', '300px');

d = createDiv();
d.style('padding-left: 50px;');
d.style('transform-origin: 10% 50% 0');
d.style('transform: rotate(' + (90) + 'deg);');
d.position(0.65*width, 45);
s = createSlider(0, 360, 0, 1);
s.style('width', '300px');
d.child(s);

IntSlider = createSlider(5, 25, 5, 0.1);
IntSlider.position(0.4*width, 70);
IntSlider.style('width', '200px');
IntSlider.style('colour', 'red');
IntSlider.hide();

}

function draw()
{
  background(220, 200, 200);
  noTint();
  image(img1, width*0.71, 30, width*0.27, height*0.4);
  fill(0);
  noStroke();
  textSize(12);
  textAlign(LEFT);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);

  if(Masor)
  {
    tint(255, 10*IntSlider.value()); // Apply transparency without changing color
    image(img2, width*0.3, 75, 50, 66);
  }
}

```

```

let teta = Math.PI*s.value()/180.0;
let er = Eroare[int(s.value())]
Int = IntSlider.value()*(0.1+0.05*er+Math.cos(teta)*Math.cos(teta));
textSize(14);
text('Intensitatea luminii I0 = '+nf(IntSlider.value(),1,1)+' Cd',0.4*width,50);
textSize(20);
text('Int = '+nf(Int,1,2)+' μA',0.6*width,0.6*height);
for (let i=0; i<NrMasuratori; i++)
{
  trace1.r[i] = MarimeFizical[i].MyInput.value();
  trace2.r[i] = MarimeFizica2[i].MyInput.value();
  trace1.theta[i] = Iuri[i];
  trace2.theta[i] = Iuri[i];
}
let IntMax = max(trace1.r);
//console.log(trace1.r.length);
if (typeof IntMax !== 'number')
{
  IntMax = 1;
}else{
  IntMax += 0.001;
}
//console.log(typeof IntMax === 'number');
//console.log(IntMax);
angleMode(DEGREES);
for (let ii=0; ii<=360; ii++)
{
  trace3.theta[ii] = ii;
  trace3.r[ii] = IntMax*Math.cos(Math.PI*ii/180.0)*Math.cos(Math.PI*ii/180.0);
}
let Unghiul = s.value();

fill(185,122,86);
noStroke();
ellipse(0.16*width,0.4*height,0.3*width,0.3*width);
fill(220, 200, 200);
ellipse(0.16*width,0.4*height,0.28*width,0.28*width);
fill(185,122,86);
ellipse(0.16*width,0.4*height,0.27*width,0.27*width);
push();
fill(0);
translate(0.16*width,0.4*height);
stroke(0);
strokeWeight(2);
line(0,-195,0,-205);
noStroke();
rotate(Unghiul);

```

```

push();
for (var i = 0; i<12; i++)
{
  text(nf(i*30, 2,0),0,-160);
  rotate(-30);
}
pop();
stroke(0);
strokeWeight(2);
for (var i = 0; i<72; i++)
{
  line(0,-178,0,-188);
  rotate(-5);
}
strokeWeight(1);
for (var i = 0; i<360; i++)
{
  line(0,-182,0,-188);
  rotate(-1);
}
pop();
fill(180);
ellipse(0.16*width,0.4*height,0.22*width,0.22*width);
fill(136,0,27);
//ellipse(0.24*width,0.48*height,0.27*width,0.27*width);
beginShape();
// Exterior part of shape, clockwise winding
let Ncerc = 50;
let dfi = 360/Ncerc;
for(let i = 0; i<Ncerc; i++)
{
  let x = 0.24*width+0.15*width*Math.cos(-3.1415/180*i*dfi);
  let y = 0.48*height+0.15*width*Math.sin(-3.1415/180*i*dfi);
  vertex(x, y);
}
beginContour();
for(let i = 0; i<Ncerc; i++)
{
  let x = 0.24*width+0.14*width*Math.cos(3.1415/180*i*dfi);
  let y = 0.48*height+0.14*width*Math.sin(3.1415/180*i*dfi);
  vertex(x, y);
}
endContour();
endShape(CLOSE);

beginShape();
// Exterior part of shape, clockwise winding
for(let i = 0; i<Ncerc; i++)

```

```

{
  let x = 0.24*width+0.135*width*Math.cos(-3.1415/180*i*dfi);
  let y = 0.48*height+0.135*width*Math.sin(-3.1415/180*i*dfi);
  vertex(x, y);
}
beginContour();
for(let i = 0; i<Ncerc; i++)
{
  let x = 0.24*width+0.11*width*Math.cos(3.1415/180*i*dfi);
  let y = 0.48*height+0.11*width*Math.sin(3.1415/180*i*dfi);
  vertex(x, y);
}
endContour();
endShape(CLOSE);

push();
fill(0);
translate(0.24*width,0.48*height);
stroke(0);
strokeWeight(2);
line(0,-195,0,-205);
noStroke();

push();
for (var i = 0; i<12; i++)
{
  text(nf(i*30, 2,0),0,-160);
  rotate(-30);
}
pop();
stroke(0);
strokeWeight(2);
for (var i = 0; i<72; i++)
{
  line(0,-178,0,-188);
  rotate(-5);
}
strokeWeight(1);
for (var i = 0; i<360; i++)
{
  line(0,-182,0,-188);
  rotate(-1);
}
pop();
let corectie = map(Int,0,IntSlider.value(),0,150);

fill(50,50,50,250-corectie);
ellipse(0.24*width,0.48*height,0.22*width,0.22*width);

```

```
noFill();
stroke(0);
strokeWeight(3);
RotatieVCpE(Unghiul);
Grafic();
noStroke();
fill(0);
textSize(14);
textAlign(LEFT);
text('θ = '+nf(s.value(),1,0)+' °',0.65*width-10,60);
}
```

```
Tabel();
```

```
if(ImgArataGrafic == 'Grafic Offline')
{
  Graf.show();
  Grafic();
}
else
{
  Graf.hide();
}
}
```

```
function RotatieVCpE(u)
{
  let Nr = 21;
  let Nr1 = 4;
  let lung = 240;
  let lat = 60;
  let dl = lung/(Nr-1);
  let du = u/(Nr-1);
  let dy = lat/(Nr-1);
  angleMode(DEGREES);
  push();
  translate(0.45*width,0.3*height);
  stroke(240,0,50);
  push();
  rotate(30);
  for(let i=0; i<Nr1; i++)
  {
    push();
    for(let j = 0; j<4; j++)
    {
      push();
      rotate(j*45-30);
```

```

    triangle(0,-28,-3,-22,0,-25);
    triangle(0,-28, 3,-22,0,-25);
    triangle(0,28,-3,22,0,25);
    triangle(0,28, 3,22,0,25);
    line(0,-25,0,25);
    pop();

}
pop();
translate(-4*dl,0);
}
pop();
fill(0);
textSize(16);
noStroke();
text("Polarizor",50, -60);
fill(150);
stroke(0);
strokeWeight(1);
push();
  rotate(s.value());
  rect(-55,-55,100,100);
  rect(-50,-50,100,100);
  noStroke();
  triangle(45,-55,45,-50,50,-50);
  triangle(-55,45,-50,50,-50,45);
  stroke(0);
  line(-55,-55,-50,-50);
  line(45,-55,50,-50);
  line(-55,45,-50,50);
  translate(-40,15);
  stroke(0,250,120);
  triangle(0,-28,-3,-22,0,-25);
  triangle(0,-28, 3,-22,0,-25);
  triangle(0,28,-3,22,0,25);
  triangle(0,28, 3,22,0,25);
  line(0,-25,0,25);
pop();
translate(10,-15);
rotate(30);
noStroke();
fill(0,0,100,50);
rect(0,-10,lung,60);
arc(0, 20, 60, 60, 90, 270 , PIE);
arc(lung, 20, 60, 60, -90, 90 , PIE);
stroke(0);
noFill();
arc(0, 20, 60, 60, 90, 270);

```

```
ellipse(lung, 20, 60, 60);
fill(0);
for(let i=0; i<Nr; i++)
{
  ellipse(i*dl,20, 3,3);
}
//rotate(-30);
translate(0,20);
stroke(240,0,50);
for(let i=0; i<Nr; i++)
{
  push();
  rotate(s.value()-30);
  triangle(0,-28,-3,-22,0,-25);
  triangle(0,-28, 3,-22,0,-25);
  triangle(0,28,-3,22,0,25);
  triangle(0,28, 3,22,0,25);
  line(0,-25,0,25);
  pop();
  translate(dl,0);
}
//translate(-10,-20);
fill(0);
textSize(16);
noStroke();
rotate(-30);
text("Analizor",-110, 45);
fill(150,150,150,100);
stroke(0);
ellipse(0,0,3,3);
//rotate(s.value());
rect(-40,-40,80,80);
rect(-45,-45,80,80);
stroke(0);
line(-45,-45,-40,-40);
line(35,-45,40,-40);
line(-45,35,-40,40);
stroke(0,250,120);
triangle(0,-28,-3,-22,0,-25);
triangle(0,-28, 3,-22,0,-25);
triangle(0,28,-3,22,0,25);
triangle(0,28, 3,22,0,25);
line(0,-25,0,25);

stroke(136,0,127);
push();
rotate(30);
let Corfact = Int/IntSlider.value();
```

```

for(let i=1; i<=Nr1+3; i++)
{
  translate(dl,0);
  push();
  rotate(-30);
  triangle(0,-28*Corfact,-3,-22*Corfact,0,-25*Corfact);
  triangle(0,-28*Corfact, 3,-22*Corfact,0,-25*Corfact);
  triangle(0,28*Corfact,-3,22*Corfact,0,25*Corfact);
  triangle(0,28*Corfact, 3,22*Corfact,0,25*Corfact);
  line(0,-25*Corfact,0,25*Corfact);
  pop();

}
pop();

pop();
}
function Vizor()
{
  stroke(200,0,0);
  strokeWeight(2);
  push();
  translate(150,0.5*height);
  line(-100,0,100,0);
  pop();

  push();
  translate(275, 0.31*height);
  //stroke(255, 0, 0);
  noStroke();
  fill(150);
  beginShape();
  // Exterior part of shape, clockwise winding
  vertex(-250, -150);
  vertex(250, -150);
  vertex(250, 150);
  vertex(-250, 150);
  // Interior part of shape, counter-clockwise winding
  beginContour();
  let Ncerc = 50;
  let dfi = 360/Ncerc;
  for(let i = 0; i<Ncerc; i++)
  {
    let x = -125+100*Math.cos(3.1415/180*i*dfi);
    let y = -100*Math.sin(3.1415/180*i*dfi);
    vertex(x, y);
  }
  endContour();
}

```

```

beginContour();
  for(let i = 0; i<Ncerc; i++)
  {
    let x = +125+100*Math.cos(3.1415/180*i*dfi);
    let y = -100*Math.sin(3.1415/180*i*dfi);
    vertex(x, y);
  }

endContour();

endShape(CLOSE);
pop();

}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de Fizica: Polarizarea luminii", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-45);
  rect(0.7*width, 15,0.3*width-10, height-25);
  for (var i=0; i<15; i++)
  {
    line(MarimeFizica1[0].x-85+i*65,MarimeFizica1[0].y-50,MarimeFizica1[0].x-85+i*65,height-25);
  }
  line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-50,MarimeFizica1[0].x+825,MarimeFizica1[0].y-50);
  line(MarimeFizica1[0].x-85,height-25,MarimeFizica1[0].x+825,height-25);
  line(MarimeFizica1[0].x-85,height-55,MarimeFizica1[0].x+825,height-55);
  line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-15,MarimeFizica1[0].x+825,MarimeFizica1[0].y-15);
  fill(0);
  noStroke();
  textSize(12);
  // Capul de tabel
  textAlign(LEFT);
  text("θ [°]",MarimeFizica1[0].x-75,MarimeFizica1[0].y-30);
  text("I (μA)",MarimeFizica1[0].x-75,MarimeFizica1[0].y+5);
  text(" e",MarimeFizica1[0].x-75,MarimeFizica1[0].y+7);
  text("I (μA)",MarimeFizica1[0].x-75,MarimeFizica1[0].y+35);
  text(" t",MarimeFizica1[0].x-75,MarimeFizica1[0].y+37);

```

```
textAlign(CENTER);
for (let i=0; i<NrMasuratori; i++)
{
  text(nf(Iuri[i],1,0),115+i*65,height-100);
}
}
function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}
function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Verificarea Legii lui Malus',
    xaxis:
    {
      title:"",
      visible: 1
    },
    yaxis:
    {
      title:"
    },
    paper_bgcolor: "#c4dedc"
  };
```

```
var data = [trace3,trace2,trace1];
//let data = [trace1];

Plotly.newPlot( GRAFIC, data, layout);
}
function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/PolarizationOfLight/Doc/L30_Polarisation of light.pdf';
}

function FunctieButon()
{
  Masor = true;
  IntSlider.show();
}
```

16.Studiul efectului termoelectric

```

let img1, img2, img3, img4, img5;
let img1, img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
var TensElectro = [];
var temperatura = [25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95];
var Contor0, Contor;
let CresteTemp = false;
let Temp, TEM;
let CoefSeebeck = 0.04;
let Student, Grupa;

let panta, intersectia, eroarea;

let TipMasuratoare;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'scatter',
  name:'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name:'fitare liniara'
};

function preload()
{
  img1 = loadImage("Img/SuportEFTErmo.jpg");
  img2 = loadImage("Img/StandExp.jpg");
}

function setup()
{
  createCanvas(1400, 850);
  background(220, 200, 200);
  NrMasuratori = temperatura.length;
  dist = 50;

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(100+i*dist+5,height-50);
    TensElectro.push(MF1);
    trace1.x[i] = temperatura[i];
  }

  Student = createInput("");
  Student.position(0.5*width+10, 18);
  Student.style('width', '150px');

```

```
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentatie Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentatie);

buttonGB = createButton("Măsor");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

buttonStart = createButton("Start");
buttonStart.position(button0.x, button0.y+60);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(0.71*width, 0.62*height);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
Graf.style('height', '320px');

}

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.71, 16, width*0.28, height*0.6);
  fill(0);
  noStroke();
  textSize(12);
```

```

text("Student: ", Student.x-55,Student.y+5);
text("Grupa: ", Grupa.x-50,Grupa.y+5);
  Tabel();
if (Masor)
{
  image(img2, 20, 100, width*0.65, height*0.7);
  for (let i=0; i<NrMasuratori; i++)
  {
    trace1.y[i] = TensElectro[i].MyInput.value();
  }
  LinFit(trace1.x,trace1.y,NrMasuratori-1);
  var Minim = 0.9*min(trace1.x);
  var Maxim = 1.1*max(trace1.x);
  var NrFit = 101;
  var dx = (Maxim-Minim)/(NrFit-1.0);
  for (var i = 0; i< NrFit; i++ )
  {
    trace2.x[i] = Minim+i*dx;
    trace2.y[i] = intersectia+panta*trace2.x[i];
  }
  noFill();
  stroke(0);
  strokeWeight(1);
  rect(0.183*width, 0.24*height, 10, 0.25*height);
  Grafic();
  fill(200,0,0);
  if(CresteTemp)
  {
    Temp = temperatura[0]+(frameCount-Contor0)*0.004;

    if(Temp > 100)
    {
      CresteTemp = false;
    }
  }
  rect(0.183*width+1,0.49*height-10-Temp*2,8,10+Temp*2);
  noStroke();
  textSize(16);
  text(nf(Temp, 1,1),0.14*width+15,0.50*height-7-Temp*2);
  stroke(200,0,0);
  text(nf(Temp*CoefSeebeck, 1,2),0.37*width+8,0.75*height-5);

}

if(ImgArataGrafic == 'Grafic Offline')
{
  Graf.show();
  Grafic();
}
else
{
  Graf.hide();
}

}

function Tabel()
{
  textSize(22);

```

```

fill(0);
noStroke();
text("Laboratorul de Fizica: Studiul Efectului Termoelectric", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.7*width-20, height-45);
rect(0.7*width, 15,0.3*width-10, height-25);
for (var i=0; i<NrMasuratori+2; i++)
{
  line(TensElectro[0].x-63+i*50,TensElectro[0].y-50,TensElectro[0].x-63+i*50,height-25);
}
line(TensElectro[0].x-63,TensElectro[0].y-50,TensElectro[0].x+737,TensElectro[0].y-50);
line(TensElectro[0].x-63,height-25,TensElectro[0].x+737,height-25);
line(TensElectro[0].x-63,TensElectro[0].y-15,TensElectro[0].x+737,TensElectro[0].y-15);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",330, 838);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("t [C]",TensElectro[0].x-50,TensElectro[0].y-30);
text("E",TensElectro[0].x-42,TensElectro[0].y-0);
text("[mV]",TensElectro[0].x-50,TensElectro[0].y+15);

//text(nf(Placute, 1,0)+" placute",TensElectro[0].x-80,TensElectro[0].y-80);
//text("x = "+nf(0.5*Placute, 1,1)+" [mm]",TensElectro[0].x-80,TensElectro[0].y-60);

for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(temperatura[i],1,1),TensElectro[0].x+i*dist,TensElectro[0].y-30);
}

}

function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

```

```
function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Efectul Termoelectric',
    xaxis:
    {
      title:'t [C]',
      visible: 1
    },
    yaxis:
    {
      title:'E [mV]'
    },
    paper_bgcolor: "#c4dedc"
  };

  var data = [trace1,trace2];
  //let data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/Instalatii/lucrari/L4Ro_Efectul%20termoelectric.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
  Temp = temperatura[0];
}

function Start()
{
  CresteTemp = true;
  Contor0 = frameCount;
}

function Stop()
{
  CresteTemp = false;
}
}
```

```
function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
    f = 1.0/(sigmy[i]*sigmy[i]);
    s += f;
    sx += x[i]*f;
    sxx+= x[i]*x[i]*f;
    sy += y[i]*f;
    sxy+= x[i]*y[i]*f;
  }

  f = 1.0/(s*sxx-sx*sx);
  panta = (s*sxy-sx*sy)*f;
  intersectia=(sy*sxx-sx*sxy)*f;
  eroarea = 0.0;
  for(let i=1; i<=N; i++)
  {
    eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
  }
}
```

17.Determinarea energiei de activare a unui semiconductor

```
let img1, img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let CresteTemp = false;
var Contor0, Contor;
let NrMasuratori, dist;
let TemperC;
let TempC = [];
let TempK = [];
let Rez = [];
let Rezi0;
let logR= [];
let deltaE;
let Constanta = 5802.26;
let EnergieDeActivare;

var temperaturaC = [22, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95];

let panta, intersectia, eroarea;

let TipMasuratoare;
let Student, Grupa;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name:'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name:'fitare liniara'
};

function preload()
{
  img1 = loadImage("Img/StandExp.jpg");
  img2 = loadImage("Img/Fig1.jpg");
}

function setup()
```

```
{
  createCanvas(1600, 700);
  background(220, 200, 200);
  NrMasuratori = temperaturaC.length;
  EnergieDeActivare = 0.78*(1+0.02*random(-1,1));
  TemperC = temperaturaC[0];
  Rezi0 = 0.000003723;
  dist = 25;

  Student = createInput("");
  Student.position(0.5*width+50, 22);
  Student.style('width', '150px');
  Student.style('font-size', '12px');
  Student.style('text-align', 'center');

  Grupa = createInput("");
  Grupa.position(Student.x+Student.width+60, Student.y);
  Grupa.style('width', '50px');
  Grupa.style('font-size', '12px');
  Grupa.style('text-align', 'center');

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(100,height-(NrMasuratori+1)*dist+i*dist+5);
    TempC.push(MF1);
    let MF2 = new Input(165,height-(NrMasuratori+1)*dist+i*dist+5);
    TempK.push(MF2);
    let MF3 = new Input(230,height-(NrMasuratori+1)*dist+i*dist+5);
    Rez.push(MF3);
    let MF4 = new Input(295,height-(NrMasuratori+1)*dist+i*dist+5);
    logR.push(MF4);
  }

  deltaE = new Input(358,height-225);

  button0 = createButton("Documentatie Laborator");
  button0.position(30, 50);
  button0.addClass("button button-primary button-border-thick button-box");
  button0.style('width', '150px');
  button0.mouseClicked(Documentatie);

  buttonGB = createButton("Măsur");
  buttonGB.position(button0.x+button0.width+20, button0.y);
  buttonGB.addClass("button button-primary button-border-thick button-box");
  buttonGB.style('width', '150px');
  buttonGB.mouseClicked(FunctieButon);
```

```

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(TempC[0].x+0.645*width, TempC[0].y+85);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width-10', '150px');
buttonGraf.mouseClicked(ArataGrafic);

buttonStart = createButton("Start");
buttonStart.position(button0.x, button0.y+60);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
Graf.style('height', '300px');

}

```

```

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.75, 25, width*0.20*0.85, height*0.55*0.85);
  Tabel();

  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);
  if (Masor)
  {
    image(img2, width*0.275, button0.y+100, width*0.40, height*0.75);
    if(CresteTemp)
    {
      TemperC = temperaturaC[0]+(frameCount-Contor0)*0.001;
      if(TemperC > 95)

```

```

    {
      CresteTemp = false;
    }
  }
  let TemperK = TemperC + 273.15;
  let Rez = Rezi0*Math.exp(Constanta*EnergieDeActivare/TemperK);
  textSize(16);
  text(nf(TemperC,1,2)+" °C",0.43*width,0.585*height-10);
  text(nf(Rez,1,2),0.432*width+15,0.785*height+27);
  text('k',0.455*width+2,0.785*height+7);
  for (let i=0; i<NrMasuratori; i++)

  {
    trace1.x[i] = TempK[i].MyInput.value();
    trace1.y[i] = logR[i].MyInput.value();
  }
  LinFit(trace1.x,trace1.y,NrMasuratori-1);
  var Minim = 0.9*min(trace1.x);
  var Maxim = 1.1*max(trace1.x);
  var NrFit = 101;
  var dx = (Maxim-Minim)/(NrFit-1.0);
  for (var i = 0; i< NrFit; i++)
  {
    trace2.x[i] = Minim+i*dx;
    trace2.y[i] = intersectia+panta*trace2.x[i];
  }
  stroke(200,0,0);
  strokeWeight(2);
  line(0.4*width+(TemperC-22)*1.18+12,0.55*height+10+TemperC*0.05,0.4*width+(TemperC-
22)*1.18+12,0.55*height+20+TemperC*0.05);
  }
  Tabel();
  if(ImgArataGrafic == 'Grafic Offline')
  {
    Graf.show();
    Grafic();
  }
  else
  {
    Graf.hide();
  }
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);

```

```

noStroke();
text("Laboratorul de Fizica: Determinarea energiei de activare a unui semiconductor ", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.7*width-20, height-45);
rect(0.7*width, 15,0.3*width-10, height-25);
for (var i=0; i<7; i++)
{
  line(TempC[0].x-85+i*65,TempC[0].y-50,TempC[0].x-85+i*65,height-25);
}
line(TempC[0].x-85,TempC[0].y-50,TempC[0].x+305,TempC[0].y-50);
line(TempC[0].x-85,height-25,TempC[0].x+305,height-25);
line(TempC[0].x-85,TempC[0].y-15,TempC[0].x+305,TempC[0].y-15);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",330, 686);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("t [°C]",TempC[0].x-75,TempC[0].y-30);
text("T",TempC[0].x-75+80,TempC[0].y-35);
text("[K]",TempC[0].x-63+65,TempC[0].y-20);
text("10 /T",TempC[0].x-68+2*65,TempC[0].y-35);

text("[K ]",TempC[0].x-70+2*65,TempC[0].y-20);
text("R",TempC[0].x-60+3*65,TempC[0].y-35);
text("[kΩ]",TempC[0].x-65+3*65,TempC[0].y-20);
text("lg(R)",TempC[0].x-69+4*65,TempC[0].y-35);
text("[-]",TempC[0].x-64+4*65,TempC[0].y-20);
text("ΔE",TempC[0].x-66+5*65,TempC[0].y-35);
text("[eV]",TempC[0].x-70+5*65,TempC[0].y-20);

//text(nf(Placute, 1,0)+" placute",TempC[0].x-80,TempC[0].y-80);
//text("x = "+nf(0.5*Placute, 1,1)+" [mm]",TempC[0].x-80,TempC[0].y-60);

for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(temperaturaC[i],1,1),TempC[0].x-65,TempC[0].y+i*dist+7);
}

```

```
}
textSize(9);
text("3",TempC[0].x-55+2*65,TempC[0].y-40);
text(" -1",TempC[0].x-60+2*65,TempC[0].y-25);

}
```

```
function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}
```

```
function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Energia de activare',
    xaxis:
    {
      title:'1000/T [1/K]',
      visible: 1
    },
    yaxis:
    {
      title:'lg(R)'
    },
    paper_bgcolor: "#c4dedc"
```



```
};

var data = [trace1,trace2];

Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2019-2020/lucrari/L24Ro_Energia%20de%20activare%20a%20unui%20semiconductor.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
}

function Start()
{
  CresteTemp = true;
  Contor0 = frameCount;
}

function Stop()
{
  CresteTemp = false;
}

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
  sxx = 0.0;
  sxy = 0.0;
  for(let i=1; i<=N; i++)
  {
```

```
f = 1.0/(sigmy[i]*sigmy[i]);
s += f;
sx += x[i]*f;
sxx+= x[i]*x[i]*f;
sy += y[i]*f;
sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
    eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

18. Determinarea conductibilității termice printr-o metodă staționară

```
let img1, img2, img3, img4;
let ImgArataGrafic, ImgArataGrafic1;
var Contor0, Contor;
let CresteTemp = false;
let Masor = false;
let NrMasuratori, dist;
let timp = [];
let T1mas = [];
let T2mas = [];
let Tmas = [];
let theta;
let lanbda = 205;
let Temp;
let Bule = [];
let Apa = [];

var Xuri = [1, 2, 3, 4, 5, 6];

let panta, intersectia, eroarea;

let TipMasuratoare;
let Student, Grupa;

function ExpT2(t,r)
{
  return r+60+0.03382*t-0.0000771605*t*t+0.0000000857339*t*t*t;
}

function ExpT1(t,r)
{
  return r+40+0.02134*t-0.0000170855*t*t+0.0000000107167*t*t*t;
}

function ExpT(t,r)
{
  return r+20+0.01503*t-0.00000409226*t*t;
}

function preload()
{
  img1 = loadImage("Img/FigEtalonare.jpg");
  img2 = loadImage("Img/Stand.jpg");
  img3 = loadImage("Img/Formule.jpg");
  img4 = loadImage("Img/Q.jpg");
```

```
}

function setup()
{
  createCanvas(1700, 700);
  background(220, 200, 200);
  NrMasuratori = Xuri.length;
  lanbda = 205+10*random(-1,1);
  dist = 25;

  Student = createInput("");
  Student.position(0.5*width+70, 18);
  Student.style('width', '150px');
  Student.style('font-size', '12px');
  Student.style('text-align', 'center');

  Grupa = createInput("");
  Grupa.position(Student.x+Student.width+60, Student.y);
  Grupa.style('width', '50px');
  Grupa.style('font-size', '12px');
  Grupa.style('text-align', 'center');

  for (let i=0; i<NrMasuratori; i++)
  {
    let MF1 = new Input(100,height-(NrMasuratori+1)*dist+i*dist+5,nf(i*2,1,0));
    timp.push(MF1);
    let MF2 = new Input(160,height-(NrMasuratori+1)*dist+i*dist+5,"");
    T1mas.push(MF2);
    let MF3 = new Input(230,height-(NrMasuratori+1)*dist+i*dist+5,"");
    T2mas.push(MF3);
    let MF4 = new Input(290,height-(NrMasuratori+1)*dist+i*dist+5,"");
    Tmas.push(MF4);
  }
  theta = new Input(355,height-120,"");
  theta.MyInput.value(22);
  lanbda = new Input(422,height-120,"");

  button0 = createButton("Documentatie Laborator");
  button0.position(32, 70);
  button0.addClass("button button-primary button-border-thick button-box");
  button0.style('width', '150px');
  button0.mouseClicked(Documentatie);

  buttonGB = createButton("Măsor");
  buttonGB.position(button0.x+button0.width+10, button0.y);
  buttonGB.addClass("button button-primary button-border-thick button-box");
  buttonGB.style('width', '150px');
```

```

buttonGB.mouseClicked(FunctieButon);

buttonGB = createButton("GraficQ");
buttonGB.position(button0.x+button0.width+170, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '120px');
buttonGB.mouseClicked(GraficQ);

buttonStart = createButton("Start");
buttonStart.position(button0.x+500, button0.y+60);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '100px');
buttonStart.mouseClicked(Start);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+20, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '100px');
buttonStop.mouseClicked(Stop);
buttonStop.hide();

}

```

```

function draw()
{
  background(220, 200, 200);
  image(img1, width*0.705, 60, width*0.29, height*0.55);

  image(img3, width*0.100-145, 100, width*0.26, height*0.50);
  image(img4, width*0.770, 450, width*0.16, height*0.34);
  Tabel();

  fill(0);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);
  if(Masor)
  {
    let Timp = 0;
    image(img2, width*0.280, 80, width*0.40, height*0.85);
    fill(0);
    noStroke();
    textSize(18);
    text("masa vas= 137g", timp[0].x+9*65, timp[0].y+100);
    text("masa apa= 621g", timp[0].x+9*65, timp[0].y+120);
    text("Scilindru= 12.56*10^-4 m^2", timp[0].x+11*65, timp[0].y-400);
  }
}

```

```

text("L= 0,12 m",timp[0].x+11*65,timp[0].y-380);
if(CresteTemp)
{
  Timp += (millis()-Contor0)*0.001;
  //text(nf(Timp,1,2),0.5*width,0.5*height);
  SaFacemBule();
  SaCurgaApa();

  if(Timp > 700)
  {
    CresteTemp = false;
  }
}
let DifTemp = theta.MyInput.value()-19.0;
let T1 = ExpT1(Timp,DifTemp);
let T2 = ExpT2(Timp,DifTemp);
let T = ExpT(Timp,DifTemp);
textSize(16);
fill(0);
text("T_1 = "+nf(T1,1,1)+" °C",timp[0].x+9.5*65,timp[0].y-100);
text("T_2 = "+nf(T2,1,1)+" °C",timp[0].x+9.5*65,timp[0].y-160);
text("T = "+nf(T,1,1)+" °C",timp[0].x+11.8*65,timp[0].y-100);
fill(0,0,120,100);
stroke(0);
strokeWeight(1);
rect(0.38*width,0.57*height+3,170,5);
rect(0.38*width,0.485*height+3,170,5);
rect(0.5*width,0.556*height+6,5.2,185);

noStroke();
fill(220,0,0);
rect(0.44*width-(T1-46)*5+7,0.57*height+4,64+(T1-46)*5,4);
rect(0.42*width-(T2-64)*4.5+7,0.485*height+4,98+(T2-64)*4.5,4);
rect(0.5*width+1,0.55*height-(T-25)*5+93,4.2,105+(T-25)*5);

}

}

function SaFacemBule()
{
  for(let i = 0; i<10;i++)
  {
    let pozX = 0.625*width+0+random(-15,15);
    let pozY = 0.75*height+10+random(-10,10);
    let vapor = new Vapor(pozX,pozY);
    if(Bule.length < 200)
    {

```

```
    Bule.push(vapor);
  }

}
for(let i = 0; i < Bule.length; i++)
{
  Bule[i].move();
  Bule[i].show();
  if (Bule[i].isOffScreen())
  {
    Bule.splice(i, 1);
    i--;
  }
}
}
function SaCurgaApa()
{
  if(random(0,1)<0.01)
  {
    let pozX = 0.55*width+25+random(-1,1);
    let pozY = 0.75*height+40;
    let vapor = new Lichid(pozX,pozY);
    if(Apa.length < 200)
    {
      Apa.push(vapor);
    }
  }
  for(let i = 0; i < Apa.length; i++)
  {
    Apa[i].move();
    Apa[i].show();
    if (Apa[i].isOffScreen())
    {
      Apa[i].vx = random(-0.1,0.1);
      Apa[i].vy = random(-0.01,0.01);
    }
  }
  if (Apa.length>=200)
  {
    Apa.splice(1,1);
  }
}

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
}
```

```

text("Laboratorul de Fizica:Determinarea conductibilitatii termice printr-o metoda stationara ", 20, 30);
noFill();
strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.7*width-20, height-45);
rect(0.7*width, 15,0.3*width-10, height-25);
for (var i=0; i<8; i++)
{
  line(timp[0].x-85+i*65,timp[0].y-50,timp[0].x-85+i*65,height-25);
}
line(timp[0].x-85,timp[0].y-50,timp[0].x+370,timp[0].y-50);
line(timp[0].x-85,height-25,timp[0].x+370,height-25);
line(timp[0].x-85,timp[0].y-15,timp[0].x+370,timp[0].y-15);

fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",330, 690);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
text("nr.",timp[0].x-68,timp[0].y-30);
text("τ",timp[0].x+10,timp[0].y-30);
text("[min]",timp[0].x,timp[0].y-20);
text("T_1",timp[0].x-10+80,timp[0].y-35);
text("[°C]",timp[0].x+70,timp[0].y-20);
text("T_2",timp[0].x+2.1*65,timp[0].y-35);
text("[°C]",timp[0].x+8+2*65,timp[0].y-20);
text("T",timp[0].x+6+3*65,timp[0].y-35);
text("[°C]",timp[0].x+3*65,timp[0].y-20);
text("θ",timp[0].x+10+4*65,timp[0].y-35);
text("[°C]",timp[0].x+4+4*65,timp[0].y-20);
text("λ",timp[0].x+10+5*65,timp[0].y-35);
text("[W/(m·grd)]",timp[0].x-16+5*65,timp[0].y-20);

for (let i = 0; i<NrMasuratori; i++)
{
  text(nf(Xuri[i],1,0),timp[0].x-65,timp[0].y+i*dist+7);
}
}

```

```
function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/Inginerie%20Electrica/2019-
2020/lucrari/SemI/L13Ro_Conductibilitate%20termica.pdf';
}

function FunctieButon()
{
  Masor = true;
  buttonStart.show();
  buttonStop.show();
}

function Start()
{
  CresteTemp = true;
  Contor0 = millis();
}

function Stop()
{
  CresteTemp = false;
}

function GraficQQ()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/ConductibilitateaTermica/Img/FigEtalo
nare1.jpg';
}
```

19.Verificarea Experimentală a legii lui Stefan-Boltzmann

```
let img1, img2, img3, img4, img5;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Intensitate = [];
let Rezistenta = [];
let Putere = [];
let lnPutere = [];
let Temperatura = [];
let lnTemp = [];
let TipMasuratoare;
var TensU = [6, 8, 10, 12, 14, 16, 18, 20];
let panta, intersectia, eroarea;
let Studenti, Grup;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'polar',
  name: 'Date Exp.'
};

var trace2 = {
  x: [],
  y: [],
  mode: 'lines',
  type: 'scatter',
  name: 'fitare liniara'
};

function TempK(u)
{
  return 530.16667+93.33929*u-1.72917*u*u;
}

function Int(u)
{
  return 0.46917+0.05458*u;
}

function preload()
{
  img1 = loadImage("Img/Fig1.jpg");
  img2 = loadImage("Img/Fig2.jpg");
  img3 = loadImage("Img/Fig3.jpg");
  img4 = loadImage("Img/Stefan-Boltzmann.jpg");
  img5 = loadImage("Img/Bec.GIF");
}

function setup()
{
  createCanvas(1400, 700);
  background(208, 200, 200);
  NrMasuratori = TensU.length;
  dist = 25;
```

```
for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Input(100,height-(NrMasuratori+1)*dist+i*dist+5);
  Intensitate.push(MF1);
  let MF2 = new Input(162,height-(NrMasuratori+1)*dist+i*dist+5);
  Putere.push(MF2);
  let MF3 = new Input(228,height-(NrMasuratori+1)*dist+i*dist+5);
  InPutere.push(MF3);
  let MF4 = new Input(292,height-(NrMasuratori+1)*dist+i*dist+5);
  Rezistenta.push(MF4);
  let MF5 = new Input(357,height-(NrMasuratori+1)*dist+i*dist+5);
  Temperatura.push(MF5);
  let MF6 = new Input(422,height-(NrMasuratori+1)*dist+i*dist+5);
  InTemp.push(MF6);
}

button0 = createButton("Document");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '150px');
button0.mouseClicked(Documentation);

buttonGB = createButton("Masuratori");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

Student = createInput("");
Student.position(0.5*width+70, 50);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'center');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+130, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

rSlider = createSlider(4, 25, 6, 0.1);
rSlider.position(0.17*width, 0.37*height);
rSlider.style('width', '150px');
rSlider.style('colour', 'red');
rSlider.hide();

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(710, 370);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

Graf = createDiv('Grafic aici');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '350px');
```

```

Graf.style('height', '280px');
}

function draw()
{
  background(220, 200, 200);
  noTint();
  image(img1, width*0.500, 70, width*0.20+120, height*0.4);
  image(img2, width*0.82, height*0.5+40, width*0.15, height*0.4);
  image(img3, width*0.82, 30, width*0.15, height*0.4);
  noStroke();
  textSize(12);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-47, Grupa.y+5);
  Tabel();

  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizica si Chimie, Universitatea Tehnică din Cluj-
Napoca", 230, 689);

  if (Masor)
  {
    noTint();
    image(img4, width*0.025, 70, width*0.47, height*0.5);
    let TensiU = rSlider.value();
    tint(255, 10*TensiU); // Apply transparency without changing color
    image(img5, width*0.04, 95, 75, 100, 0);
    text(nf(TensiU, 1, 1), Intensitate[0].x+230, Intensitate[0].y-295);
    stroke(185, 122, 87);
    let Intens = Int(TensiU);
    text(nf(Intens, 1, 2), Intensitate[0].x+169, Intensitate[0].y-103);
  }
  if(ImgArataGrafic == 'Grafic Offline')
  {
    for(let i = 0; i<NrMasuratori; i++)
    {
      trace1.x[i] = lnPutere[i].MyInput.value();
      trace1.y[i] = lnTemp[i].MyInput.value();
    }
    LinFit(trace1.x, trace1.y, NrMasuratori-1);
    var Minim = 0.9*min(trace1.x);
    var Maxim = 1.1*max(trace1.x);
    var NrFit = 101;
    var dx = (Maxim-Minim)/(NrFit-1.0);
    for (var i = 0; i< NrFit; i++)
    {
      trace2.x[i] = Minim+i*dx;
      trace2.y[i] = intersectia+panta*trace2.x[i];
    }
    Graf.show();
    Grafic();
  }
  else
  {
    Graf.hide();
  }
}

```

```

function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laborator de Fizica - Verificarea Experimentala a legii lui Stefan-Boltzmann", 20, 30);
  noFill();

  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.8*width-20, height-45);
  rect(0.8*width, 15,0.2*width-10, height-25);
  for (var i=0; i<8; i++)
  {
    line(Intensitate[0].x-87+i*65,Intensitate[0].y-50,Intensitate[0].x-87+i*65,height-25);
  }

  line(Intensitate[0].x-87,Intensitate[0].y-50,Intensitate[0].x+368,Intensitate[0].y-50);
  line(Intensitate[0].x-87,height-25,Intensitate[0].x+368,height-25);
  line(Intensitate[0].x-87,height-25,Intensitate[0].x+368,height-25);
  line(Intensitate[0].x-85,Intensitate[0].y-15,Intensitate[0].x+368,Intensitate[0].y-15);

  fill(0);
  noStroke();
  textSize(14);
  // Cap tabel
  text("U [V]",Intensitate[0].x-70,Intensitate[0].y-30);
  text("I [A]",Intensitate[0].x,Intensitate[0].y-30);
  text("P [W]",Intensitate[0].x+60,Intensitate[0].y-30);
  text("ln P",Intensitate[0].x+130,Intensitate[0].y-30);
  text("R [Ω]",Intensitate[0].x+190,Intensitate[0].y-30);
  text("T [K]",Intensitate[0].x+255,Intensitate[0].y-30);
  text("ln T",Intensitate[0].x+325,Intensitate[0].y-30);

  for (let i = 0; i<NrMasuratori; i++)
  {
    text(nf(TensU[i], 1,0),Intensitate[0].x-65,Intensitate[0].y+i*dist+7);
  }
}

function Documentation()
{
  window.location='https://phys.utcluj.ro/resurse/Facultati/SIM/2017-2018/Laboratoare/SemII_Petru/Verificarea%20legii%20Stefan-Boltzmann.pdf';
}

function FunctieButon()
{
  rSlider.show();
  Masor = true;
}

```

```
function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}

function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Stefan-Boltzmann',
    xaxis:
    {
      title:'ln (T)',
      visible: 1
    },
    yaxis:
    {
      title:'ln (P)'
    },
    paper_bgcolor: "#c4dedc"
  };

  var data = [trace1,trace2];
  //let data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

function LinFit(x,y,N)
{
  let f, s, sx, sy, sxx, sxy;
  let sigmy = [N+1];
  for (let i=1; i<=N; i++)
  {
    sigmy[i] = 1.0;
  }
  s = 0.0;
  sx = 0.0;
  sy = 0.0;
```

```
sxx = 0.0;
sxy = 0.0;
for(let i=1; i<=N; i++)
{
  f = 1.0/(sigmy[i]*sigmy[i]);
  s += f;
  sx += x[i]*f;
  sxx+= x[i]*x[i]*f;
  sy += y[i]*f;
  sxy+= x[i]*y[i]*f;
}

f = 1.0/(s*sxx-sx*sx);
panta = (s*sxy-sx*sy)*f;
intersectia=(sy*sxx-sx*sxy)*f;
eroarea = 0.0;
for(let i=1; i<=N; i++)
{
  eroarea += pow((y[i]-panta*x[i]-intersectia)/sigmy[i],2);
}
}
```

20. Determinarea temperaturii Curie a materialelor feromagnetice

```
let img1,img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Student, Grupa;
let d, s;
let Int, Int0;

let MarimeFizica1 = [];
let MarimeFizica2 = [];
let MarimeFizica3 = [];
let MarimeFizica4 = [];
var Yuri = [];
var TuriNi = [320, 340, 360, 380, 400, 410, 420, 425, 430, 435, 450, 470, 500];
var TuriFe = [200, 250, 300, 320, 330, 340, 350, 360, 370, 380, 400, 420, 450];
var Eroare = [];
var StartStop;
var Reset;
var TCurieNi, DeltaTNi, IntMaxNi, IntMinNi;
var TCurieFe, DeltaTFe, IntMaxFe, IntMinFe;
var Temperatura;
var TipMasuratoare;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'scatter',
  name: 'Date Exp.'
};
var trace2 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'scatter',
  name: 'Date Exp.'
};

function preload()
{
  img1 = loadImage("Img/Dispozitiv experimental_Curie.jpg");
  img2 = loadImage("Img/Setup_experimental.jpg");
}

function setup()
{
```



```
createCanvas(1400, 700);
background(298, 298, 298);

NrMasuratori = TuriNi.length;
dist = 25;
StartStop = 0;
Reset=0
TCurieNi = 430.0+1*random(-1,1);
IntMinNi = 8.5;
IntMaxNi = 1.5;
DeltaTNi = 4.67;
TCurieFe = 356.7+3*random(-1,1);
IntMinFe = 3.9;
IntMaxFe = 0.3;
DeltaTFe = 17.7;

TipMasuratoare = 0;
if (TipMasuratoare == 0)
{
  Temperatura = TuriNi[0];
} else{
  Temperatura = TuriFe[0];
}

for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Input(100+i*65,height-55,nf(TuriNi[i],3,0));
  MarimeFizica1.push(MF1);
  let MF2 = new Input(100+i*65,height-30, "");
  MarimeFizica2.push(MF2);
  Yuri[i] = 1;
  trace1.x[i]=TuriNi[i];
  trace2.x[i]=TuriFe[i];
  let MF3 = new Input(100+i*65, height-125, nf(TuriFe[i],3,0));
  MarimeFizica3.push(MF3);
  let MF4 = new Input(100+i*65,height-98, "");
  MarimeFizica4.push(MF4);
}

Student = createInput("");
Student.position(0.5*width+10, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'left');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
```

```
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentatie Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '200px');
button0.mouseClicked(Documentatie);

buttonNi = createButton("Masor Ni");
buttonNi.position(button0.x+button0.width+20, button0.y);
buttonNi.addClass("button button-primary button-border-thick button-box");
buttonNi.style('width', '150px');
buttonNi.mouseClicked(FunctieButonNi);

buttonFe = createButton("Masor Fe");
buttonFe.position(buttonNi.x+buttonNi.width+10, buttonNi.y);
buttonFe.addClass("button button-primary button-border-thick button-box");
buttonFe.style('width', '150px');
buttonFe.mouseClicked(FunctieButonFe);

buttonStart = createButton("Start");
buttonStart.position(button0.x+button0.width+20, button0.y+30);
buttonStart.addClass("button button-primary button-border-thick button-box");
buttonStart.style('width', '150px');
buttonStart.mouseClicked(StartMasuratoare);
buttonStart.hide();

buttonStop = createButton("Stop");
buttonStop.position(buttonStart.x+buttonStart.width+10, buttonStart.y);
buttonStop.addClass("button button-primary button-border-thick button-box");
buttonStop.style('width', '150px');
buttonStop.mouseClicked(StopMasuratoare);
buttonStop.hide();

ImgArataGrafic = 'Grafic Online';

buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(width*0.71, height*0.5);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);

Graf = createDiv(' ');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
```

```

Graf.style('height', '300px');
IntSlider = createSlider(0.01, 1, 0.01, 0.01);
IntSlider.position(0.4*width+50, 70);
IntSlider.style('width', '200px');
IntSlider.style('colour', 'red');
IntSlider.hide();

}

function draw()
{
  background(228, 208, 204);
  noTint();
  fill(0);
  noStroke();
  textSize(12);
  textAlign(LEFT);
  text("Student: ", Student.x-55,Student.y+5);
  text("Grupa: ", Grupa.x-50,Grupa.y+5);
  image(img2, 985, 50, 400,220); //defined the image position and size for "setup experimental "

  if(Masor)
  {
    noTint();
    image(img1, width*0.05, 100, width*0.6, height*0.6);
    if(StartStop == 0)
    {
      fill(255);
      rect(width*0.4+50,height*0.4+75,50,50);
      stroke(0);
      //fill(0,0,255)
      strokeWeight(2);
      ellipse(width*0.4+55,height*0.4+90,10,10);
      line(width*0.4+60,height*0.4+90,width*0.4+95,height*0.4+70);
      strokeWeight(1);
    }
    if(StartStop == 1)
    {
      if (Temperatura < 1000)
      {
        var VitCrestTemp = IntSlider.value();
        Temperatura += VitCrestTemp;
      }
    }
    stroke(0);
    strokeWeight(1);
    textSize(14);
  }
}

```

```

fill(0);
text(nf(Temperatura,1,1)+" °C",0.182*width,0.345*height);
fill(200,0,0);
if(TipMasuratoare == 0)
{
  text(nf(IntCurtElectricNi(Temperatura),1,2),0.0755*width,0.415*height);
  noStroke();
  fill(0);
  text("Nichelita",0.32*width+5,0.415*height+8);
}else{
  text(nf(IntCurtElectricFe(Temperatura),1,2),0.0755*width,0.415*height);
  noStroke();
  fill(0);
  text("Ferita",0.32*width+12,0.415*height+8);
}
}

Tabel();
for (let i=0; i<NrMasuratori; i++)
{
  trace1.y[i]=MarimeFizica2[i].MyInput.value();
  trace2.y[i]=MarimeFizica4[i].MyInput.value();
}

if(ImgArataGrafic == 'Grafic Offline')
{
  Graf.show();
  if(TipMasuratoare == 0)
  {
    GraficNi();
  }else{
    GraficFe();
  }
}
else
{
  Graf.hide();
}
}
function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de fizica: Temperatura Curie", 20, 30);
  noFill();

```

```

strokeWeight(2);
stroke(255);
rect(7, 7,width-10, height-10);
stroke(0);
rect(5, 5,width-10, height-10);
strokeWeight(1);
rect(10, 35,0.7*width-20, height-45);
rect(0.7*width, 15,0.3*width-10, height-25);
for (var i=0; i<15; i++)
{
  line(MarimeFizica1[0].x-85+i*65,MarimeFizica1[0].y-15,MarimeFizica1[0].x-85+i*65,height-15);
  line(MarimeFizica1[0].x-85+i*65,MarimeFizica1[0].y-85,MarimeFizica1[0].x-85+i*65,height-85);
}
line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-85,MarimeFizica1[0].x+825,MarimeFizica1[0].y-85);
line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-55,MarimeFizica1[0].x+825,MarimeFizica1[0].y-55);
line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-30,MarimeFizica1[0].x+825,MarimeFizica1[0].y-30);
line(MarimeFizica1[0].x-85,height-15,MarimeFizica1[0].x+825,height-15);
line(MarimeFizica1[0].x-85,height-40,MarimeFizica1[0].x+825,height-40);
line(MarimeFizica1[0].x-85,MarimeFizica1[0].y-15,MarimeFizica1[0].x+825,MarimeFizica1[0].y-15);
fill(0);
noStroke();
textSize(12);
text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, UTCN",width*0.7+10,height-15);

fill(0);
noStroke();
textSize(12);
// Capul de tabel
textAlign(LEFT);

text("T0C",MarimeFizica1[0].x-75,MarimeFizica1[0].y+8)

text("I(μA)",MarimeFizica1[0].x-75,MarimeFizica1[0].y+32);

text("T0C",MarimeFizica1[0].x-75,MarimeFizica1[0].y-65)

text("I(μA)",MarimeFizica1[0].x-75,MarimeFizica1[0].y-40);
text("Nichelita",MarimeFizica1[0].x-85,MarimeFizica1[0].y-17);
text("Ferita",MarimeFizica3[0].x-85,MarimeFizica3[0].y-17);

textAlign(CENTER);
}
function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
}

```

```
}
else
{
  ImgArataGrafic = 'Grafic Online';
  buttonGraf.html(ImgArataGrafic);
}
}
function GraficNi()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title: 'Temp. Curie Nichel',
    xaxis:
    {
      title: 't ( C)',
      visible: 1
    },
    yaxis:
    {
      title: 'I (uA)'
    },
    paper_bgcolor: "#c4dedc"
  };

  let data = [trace1];

  Plotly.newPlot( GRAFIC, data, layout);
}

function GraficFe()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
```

```
var layout =
{
  margin:
  {
    t: 50,
    b: 50,
    l: 50,
    r: 50
  },
  title:'Temp. Curie Ferita',
  xaxis:
  {
    title:'t ( C)',
    visible: 1
  },
  yaxis:
  {
    title:'I (uA)'
  },
  paper_bgcolor: "#c4dedc"
};

let data = [trace2];

Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Determinarea temperaturii Curie/Doc/
Determinarea temperaturii Curie.pdf';
}

function FunctieButonNi()
{
  Masor = true;
  TipMasuratoare = 0;
  Temperatura = TuriNi[0];
  IntSlider.show();
  buttonStart.show();
  buttonStop.show();
}

function FunctieButonFe()
{
  Masor = true;
  TipMasuratoare = 1;
  Temperatura = TuriFe[0];
  IntSlider.show();
}
```

```
    buttonStart.show();
    buttonStop.show();
}

function StartMasuratoare()
{
    StartStop = 1;
}

function StopMasuratoare()
{
    StartStop = 0;
}

function ResetMasuratoare()
{
    StartStop = 1;
}

function IntCurtElectricNi(t)
{
    //return 1;
    return (IntMinNi-IntMaxNi)/(1.0+Math.exp((t-TCurieNi)/DeltaTNi))+IntMaxNi;
}

function IntCurtElectricFe(t)
{
    //return 1;
    return (IntMinFe-IntMaxFe)/(1.0+Math.exp((t-TCurieFe)/DeltaTFe))+IntMaxFe;
}
```


21.Studiul comportării materialelor feromagnetice

```
let img1,img2;
let ImgArataGrafic, ImgArataGrafic1;
let Masor = false;
let NrMasuratori, dist;
let Student, Grupa;
let d, s;
let Int, Int0;

let MarimeFizica1 = [];
let MarimeFizica2 = [];
let MarimeFizica3 = [];
let MarimeFizica4 = [];
var Iuri = [-5,-4,-3, -2, -1, -0.5, 0, 0.5, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0.5, 0, -0.5, -1, -2, -3, -4,-5];
var FluxM = [0, 0.875, 2.1, 3.15, 4.9, 6.5, 8.75, 12.5, 17.5, 26.25, 31.5, 33.6, 35, 34.125, 32.9, 31.85, 30.1, 28.5,
26.25, 22.5,17.5, 8.75, 3.5, 1.4, 0];
var Eroare = [];
var StartStop;
var TCurie, DeltaT;
var IntMax, IntMin;
var Temperatura;

var trace1 = {
  x: [],
  y: [],
  mode: 'markers',
  type: 'scatter',
  name:'Date Exp.'
};

function preload()
{
  img1 = loadImage("Img/Dispozitiv_experimental.jpg");
  img2 = loadImage("Img/Curba_histereza.png");
}

function setup()
{
  createCanvas(1400, 700);
  background(220, 200, 200);

  NrMasuratori = Iuri.length;
  dist = 25;
  StartStop = 0;
  TCurie = 520+5*random(-1,1);
  IntMin = 40;
  IntMax = 20;
```

```
DeltaT = 5;
Temperatura = Iuri[0];

for (let i=0; i<NrMasuratori; i++)
{
  let MF1 = new Imput(75+i*50,height-140,nf(Iuri[i],1,1));
  MarimeFizica1.push(MF1);
  let MF2 = new Imput(75+i*50,height-110, "");
  MarimeFizica2.push(MF2);
  let MF3 = new Imput(75+i*50,height-80, "");
  MarimeFizica3.push(MF3);
  let MF4 = new Imput(75+i*50,height-50, "");
  MarimeFizica4.push(MF4);
  trace1.x[i]=Iuri[i];
}

for (let i=0; i<=360; i++)
{
  let Err = FluxM[i]*0.025*random(-1,1);
  Eroare.push(Err);
}

Student = createInput("");
Student.position(0.5*width+10, 18);
Student.style('width', '150px');
Student.style('font-size', '12px');
Student.style('text-align', 'left');

Grupa = createInput("");
Grupa.position(Student.x+Student.width+60, Student.y);
Grupa.style('width', '50px');
Grupa.style('font-size', '12px');
Grupa.style('text-align', 'center');

button0 = createButton("Documentatie Laborator");
button0.position(30, 50);
button0.addClass("button button-primary button-border-thick button-box");
button0.style('width', '200px');
button0.mouseClicked(Documentatie);

buttonGB = createButton("Masor");
buttonGB.position(button0.x+button0.width+20, button0.y);
buttonGB.addClass("button button-primary button-border-thick button-box");
buttonGB.style('width', '150px');
buttonGB.mouseClicked(FunctieButon);

ImgArataGrafic = 'Grafic Online';
```

```
buttonGraf = createButton(ImgArataGrafic);
buttonGraf.position(width*0.71, height*0.309);
buttonGraf.addClass("button button-primary button-border-thick button-box");
buttonGraf.style('width', '150px');
buttonGraf.mouseClicked(ArataGrafic);
```

```
Graf = createDiv(' ');
Graf.id('Graf');
Graf.position(buttonGraf.x, buttonGraf.y+25);
Graf.style('width', '400px');
Graf.style('height', '300px');
```

```
IntSlider = createSlider(0, 24, 0, 1);
IntSlider.position(0.12*width, 480);
IntSlider.style('width', '200px');
IntSlider.style('colour', 'red');
IntSlider.hide();
```

```
}
```

```
function draw()
{
  background(220, 200, 200);
  fill(0);
  noStroke();
  textSize(12);
  textAlign(LEFT);
  text("Student: ", Student.x-55, Student.y+5);
  text("Grupa: ", Grupa.x-50, Grupa.y+5);
  image(img2, 985, 20, 400, 180); //defined the image position and size
```

```
if(Masor)
{
  noTint();
  image(img1, width*0.05, 100, width*0.6, height*0.6);
  var Pozitie = round(IntSlider.value());

  stroke(0);
  strokeWeight(1);
  textSize(14);
  fill(0);

  text(nf(Iuri[Pozitie], 1, 1), 0.335*width, 0.195*height);
  fill(200, 0, 0);
  text(nf(FluxM[Pozitie]+Eroare[Pozitie], 1, 3), 0.56*width, 0.196*height);
}
```

```
Tabel();
for (let i=0; i<NrMasuratori; i++)
{
  trace1.y[i]=MarimeFizica4[i].MyInput.value();
  trace1.x[i]=MarimeFizica3[i].MyInput.value();
}

if(ImgArataGrafic == 'Grafic Offline')
{
  Graf.show();
  Grafic();
}
else
{
  Graf.hide();
}
}
function Tabel()
{
  textSize(22);
  fill(0,0,0);
  noStroke();
  text("Laboratorul de fizica: Studiul materialelor feromagnetice", 20, 30);
  noFill();
  strokeWeight(2);
  stroke(255);
  rect(7, 7,width-10, height-10);
  stroke(0);
  rect(5, 5,width-10, height-10);
  strokeWeight(1);
  rect(10, 35,0.7*width-20, height-200);
  rect(0.7*width, 15,0.3*width-10, 0.75*height-5);
  for (var i=0; i<27; i++)
  {
    line(MarimeFizica1[0].x-60+i*50,MarimeFizica1[0].y-15,MarimeFizica1[0].x-60+i*50,height-35);
  }
  line(MarimeFizica1[0].x-60,MarimeFizica1[0].y-15,MarimeFizica1[0].x+1240,MarimeFizica1[0].y-15);
  line(MarimeFizica1[0].x-60,MarimeFizica1[0].y+17,MarimeFizica1[0].x+1240,MarimeFizica1[0].y+17);
  line(MarimeFizica1[0].x-60,height-35,MarimeFizica1[0].x+1240,height-35);
  line(MarimeFizica1[0].x-60,height-65,MarimeFizica1[0].x+1240,height-65);
  line(MarimeFizica1[0].x-60,MarimeFizica1[0].y+46,MarimeFizica1[0].x+1240,MarimeFizica1[0].y+46);
  fill(0);
  noStroke();
  textSize(12);
  text("Toate drepturile rezervate © Departamentului de Fizică și Chimie, Universitatea Tehnică din Cluj-
Napoca",width*0.27+10,height-15);
```

```

// Capul de tabel
textAlign(LEFT);

text("I (A)",MarimeFizica1[0].x-45,MarimeFizica1[0].y+5);

text("Φ(mWb)",MarimeFizica1[0].x-59,MarimeFizica1[0].y+35);

text("H(A/m)",MarimeFizica3[0].x-55,MarimeFizica3[0].y+5);

text("B(T)",MarimeFizica4[0].x-50,MarimeFizica4[0].y+5);

textAlign(CENTER);
}
function ArataGrafic()
{
  if (ImgArataGrafic == 'Grafic Online')
  {
    ImgArataGrafic='Grafic Offline';
    buttonGraf.html(ImgArataGrafic);
  }
  else
  {
    ImgArataGrafic = 'Grafic Online';
    buttonGraf.html(ImgArataGrafic);
  }
}
function Grafic()
{
  Graf.html("");

  GRAFIC = document.getElementById('Graf');

  //console.log(trace1[0].y[4]);
  var layout =
  {
    margin:
    {
      t: 50,
      b: 50,
      l: 50,
      r: 50
    },
    title:'Curba de histereză',
    xaxis:
    {
      title:'H (A/m)',
      visible: 1
    },
  },

```

```
yaxis:
{
  title:'B'(T)
},
paper_bgcolor: "#c4dedc"
};

let data = [trace1];

Plotly.newPlot( GRAFIC, data, layout);
}

function Documentatie()
{
  window.location='https://phys.utcluj.ro/resurse/Laboratoare/LabOnline/Studiul materialelor feromagnetice/Do
c/Studiul materialelor feromagnetice.pdf';
}

function FunctieButon()
{
  Masor = true;
  IntSlider.show();
}
```

VIII. Bibliografie

1. I. Cosma, O. Pop, A. Terțan, E. Hanga, I. Boscovitz, I. Lupșa, E. Papp, P. Lucaci, I. Pop, *Fizică: Îndrumător pentru lucrări de laborator*, Editura Institutului Politehnic Cluj-Napoca, 1979.
2. P. Păscuță, L. Pop, M. Boșca, *Fizică. Lucrări practice*, Editura UTPress, Cluj-Napoca, 2013.
3. R. Fechete, R. Chelcea, D. Moldovan, S. Nicoară, I. Coroiu, C. Badea, E. Culea, I. Cosma, N. Șerban, *Fizică: Îndrumător de laborator*, Editura UTPress, Cluj-Napoca, 2014.