Problemă-Model-Algoritm-Soluţie (T1)

1. Problema

Abordarea oricărei probleme presupune următoarele etape preliminare:

i. Identificarea problemei (procese observate în realitatea și viaţa de zi cu zi, în rutina profesională sau situații virtuale, ipotetice);
ii. Descrierea problemei (explicitarea, argumentarea, exprimarea folosind terminologia adecvată);
iii. Formularea problemei (sintetizarea cerinţelor, a obiectivului, precizarea datelor şi a ipotezelor).

Orice problemă tehnică sau de altă natură necesită o abordare științifică pentru rezolvarea ei.

Problemă → ... **d e m e r s *de rezolvare*** ... → Soluţie

Demersul de rezolvare presupune următoarele etape: elaborarea MODELULUI problemei, întocmirea ALGORITMULUI de rezolvare şi PROGRAMAREA – adică implementarea (software) a algoritmului.

2. Modelarea

Procesul generării unui **obiect (S′)** care înlocuieşte **sistemul real (S)** în vederea rezolvării problemei. Acest obiect se numeşte **model** şi poate fi *fizic* sau *formal* (matematic, logic, grafo-analitic).

În sens formal, *modelul este un înlocuitor* **conceptual** şi **abstract** al sistemului real.

Bazat pe legi şi principii

Bazat pe un set de simboluri şi operatori

Modelul (**S′**) este un echivalent al sistemului real (S) care poate fi studiat mai uşor. Echivalarea **S′⇔S** poate fi exactă sau aproximativă. De exemplu, modelele geometrice pot fi riguros echivalente cu sistemele reale pe care le reprezintă. În alte situaţii, modelul poate fi o construcţie teoretică cu ajutorul căruia reuşim să aproximăm o parte a realităţii. Redarea construcţiei teoretice prin relaţii matematice conduce la obţinerea unui *model matematic* al sistemului studiat.
Studiul oricărui model se poate face experimental prin *simulare numerică*.

Etape de modelare:
1. Identificarea **fenomenelor** şi a legilor şi principiilor care le guvernează.
2. Formalizarea matematică propriuzisă.
3. Detalierea, rafinarea și simplificarea modelului prin stabilirea *ipotezelor simplificatoare*.

3. Algoritmul

Reprezintă etapa de formalizare a fazelor şi paşilor de rezolvare a modelului problemei.
Instrumentele de descrierea a algoritmilor sunt: schema logică (organigrama), pseudocodul, precum și instrumente standardizate în ingineria software (de exemplu UML -Unified Modeling Language).

Implementarea software a algoritmului permite obţinerea soluţiei prin simulare.

Exemplu de problemă

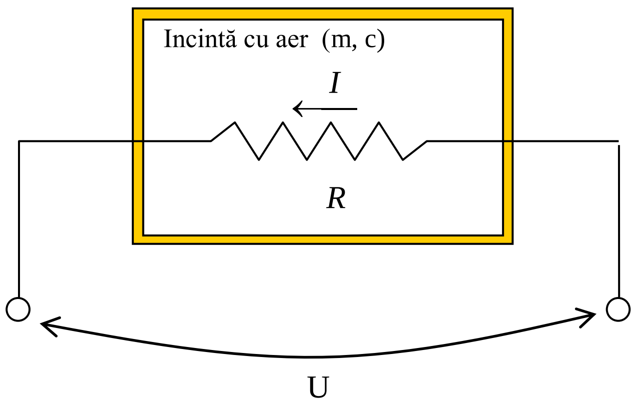
A) Etape preliminare

i) Identificarea problemei:

Problema încălzirii unei camere (incinte) cu ajutorul unui radiator electric.

ii) Descrierea problemei

Încălzirea are loc prin efect Joule, cu ajutorul unui circuit electric (pur rezistiv) având rezistenţa *R*.
Schema generică a problemei este următoarea:



iii) Formularea problemei

Se cere să se determine cum evoluează temperatura în incintă, cunoscând parametrii constructivi ai radiatorului (puterea nominală – P_n și tensiunea nominală – U_n) și tensiunea de alimentare (U) a acestuia precum și dimensiunile incintei și proprietățile fizice ale mediului care trebuie încălzit – adică aerul. Elementul rezistiv al radiatorului (cu rezistența R) se încălzește datorită trecerii unui curent electric (I) prin acesta (efect Joule).
Ipoteze (simplificatoare):
- incinta este perfect izolată, deci nu schimbă căldură cu exteriorul,
- se consideră că aerul se încălzește uniform în volumul incintei.

B) Demersul de rezolvare

1) Modelul problemei:

- energia electrică (W) se exprimă în funcție de puterea curentă absorbită de radiator (P), în care curentul se determină cu legea lui Ohm (I=U/R).

$$W = P \cdot t = U \cdot I \cdot t = \frac{U^2}{R} t$$

- rezistența electrică R este un parametru constructiv dimensionat cu relația:

$$R = \frac{U_n^2}{P_n},$$

deci energia electrică de exprimă sub forma $W = P_n \left(\frac{U}{U_n} \right)^2 t$.

- *legea conservării energiei* pentru efectul termoelectric al curentului electric (Joule), în ipoteza că nu sunt pierderi de căldură presupune că toată energia electrică se transformă în căldură transmisă mediului din incintă:

$$W \cong Q = m \cdot c \cdot \Delta\theta$$

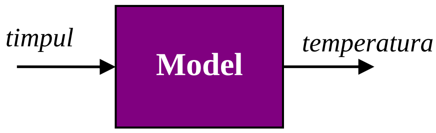
$$P_n \left(\frac{U}{U_n} \right)^2 t = m \cdot c(\theta - \theta_0)$$

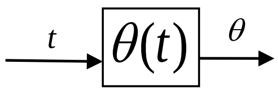
- explicitarea masei de aer care se încălzește se face pe baza volumului incintei și a densității aerului, presupunând ca forma incintei este paralelipipedică $V = l \times w \times h$, rezultă: $m = \rho \cdot V$.

Rezultă **modelul matematic al problemei** ca o funcție liniară de o singură variabilă, reprezentând un sistem cu o intrare și o ieșire:

$$\theta(t) = \frac{P_n}{\rho V c} \left(\frac{U}{U_n} \right)^2 t + \theta_0$$

Reprezentări structurale de tip bloc:





<u>Modelul se poate simula numeric prin furnizarea datelor necesare, astfel</u>:
- pentru condiția inițială dată, de exemplu la momentul $t_0=0$, temperatura inițială $\theta_0=20^oC$ (în grade Kelvin: 273,15+20=293,15K);
- pentru aer: densitatea se consideră $\rho=1Kg/m^3$, căldura specifică c=837 J//Kg K, volumul se alege;
- Puterea și tensiunea nominală se aleg, de ex. 2000W și 230V, iar tensiunea de alimentare *U* se poate considera eventual cu o valoare mai mică (220V).
- se alege timpul maxim de simulare (timpul final), de exemplu $t_f=180$ secunde.

Terminologie. Semantica termenilor în limbaj matematic:

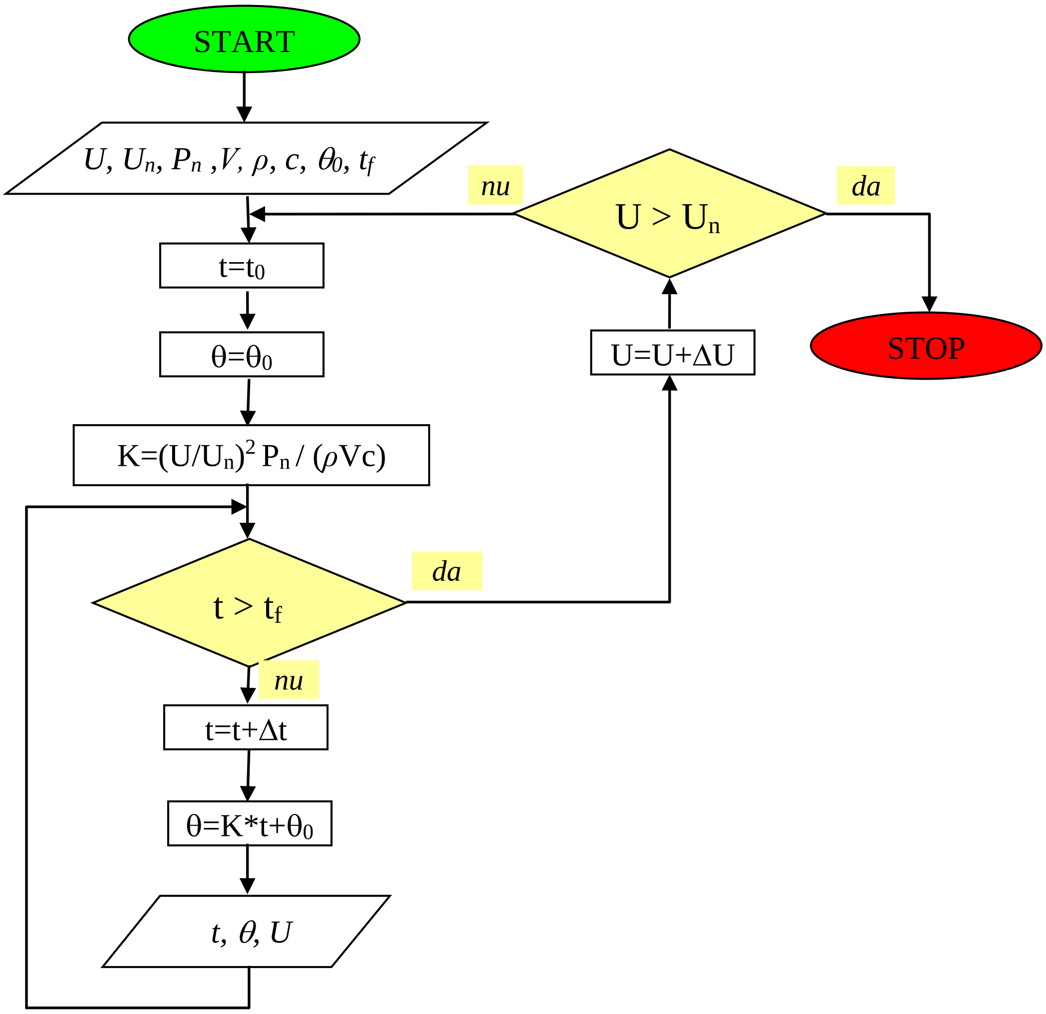
$\theta(t)$ este o funcţie, unde t este *argumentul* funcţiei,

θ este o variabilă *dependentă*, iar t este variabila *independentă*.

Toate celelalte mărimi care apar în modelul matematic se numesc *parametrii*.

În general orice funcţie este considerată variabilă dependentă de argumentul sau argumentele sale, care sunt considerate în acest caz variabile independente.

2) Algoritmul



Se observă că parametrul U – tensiunea de alimentare face obiectul unui ciclu de calcul distinct pentru a simula evoluția temperaturii atunci când sistemul este alimentat cu tensiuni mai mici decât cea nominală. Se poate alege de exemplu, o tensiune inițială de alimentare U=200V și pasul ΔU=10V.

3) Să se realizeze un *program software* pentru implementarea algoritmului problemei de mai sus. Alegeți diferite valori pentru parametrii problemei și temperatura inițială.

Modelarea simulativa în ştiinţă şi inginerie: metode „*soft-computing*” (T2)

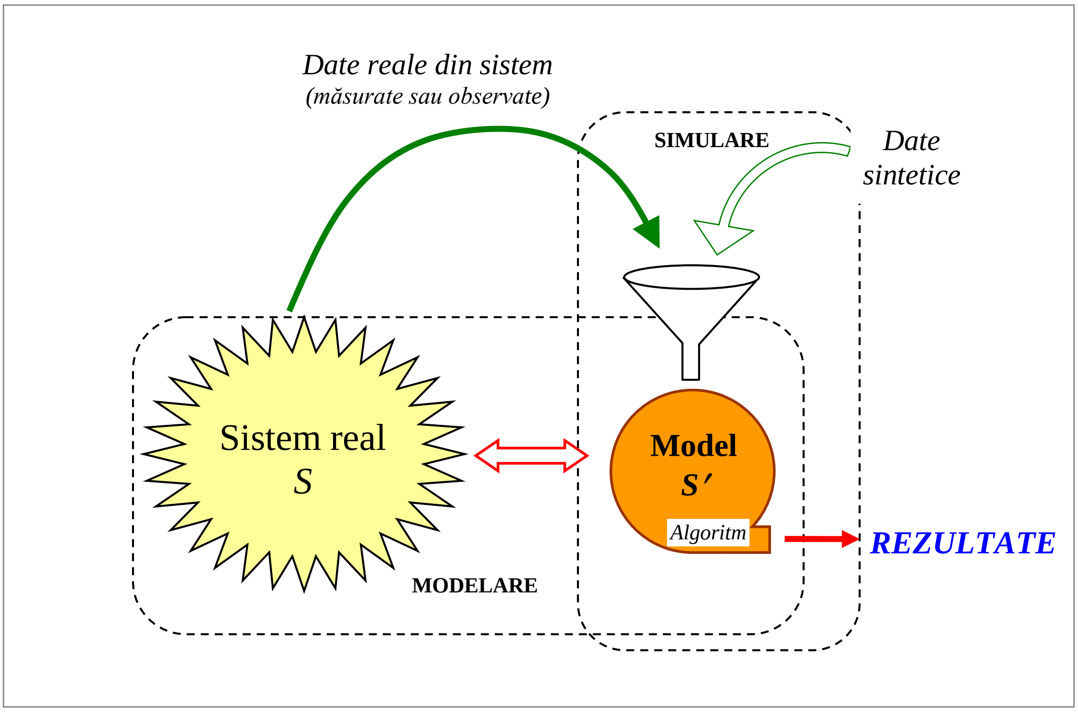
Studiul oricărui sistem se poate face prin *simulare numerică*. Simularea numerică presupune:

MODEL MATEMATIC (*formal*) + ALGORITM DE REZOLVARE +
PROGRAM DE CALCUL (*implementare software*) + DATE

Datele folosite în simulare pot fi reale sau sintetice.
Date reale: se obțin prin achiziția directă din sistemul fizic, prin măsurare sau observare.
Date sintetice: obţinute sau generate prin alte mijloace decât măsurare.

Fundamentul simulării: Calculul Numeric.

Schema sugestivă ilustrând principiul simulării numerice pe baza modelului formal:

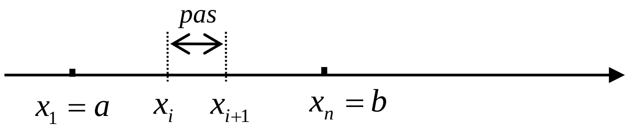


Concepte de bază în Calculul Numeric:

- discretizarea,
- diferenţă finită,
- iteraţie,

Discretizarea – trecerea din domeniul continuu infinit, în discret finit:

$$[a,b] \in \Re \rightarrow \{x_1 = a, x_2, x_3, ..., x_i, ..., x_n = b\}$$



Discretizarea unui domeniu numeric continuu se poate face cu un *pas* de lungime constantă sau variabilă (pas adaptiv).

Se defineşte pasul de discretizare sub forma diferenţei finite de ordinul *1* : $\Delta x_i = x_i - x_{i-1}$, pentru $i = 1,\ldots,n$.

Diferenţe finite de ordin superior

$\Delta x_i = x_i - x_{i-1}$ (diferența finită de ordinul 1),

$\Delta^{(2)} x_i = \Delta x_i - \Delta x_{i-1} = (x_i - x_{i-1}) - (x_{i-1} - x_{i-2}) = x_i - 2x_{i-1} + x_{i-2}$,

$\Delta^{(3)} x_i = \Delta^{(2)} x_i - \Delta^{(2)} x_{i-1}$,

$\ldots$

$\Delta^{(n)} x_i = \Delta^{(n-1)} x_i - \Delta^{(n-1)} x_{i-1}$.

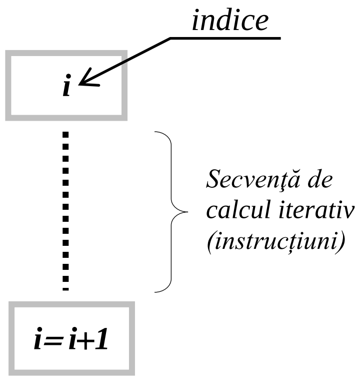
Exemplu. Orice domeniu (interval numeric) pe mulțimea numerelor reale este continuu și conține o infinitate de elemente. Calculatoarele numerice nu pot opera cu astfel de entități în maniera în care o face mintea umană. Prin urmare, calculatoarele lucrează cu mulțimi discrete și finite. Astfel, se poate opera cu tablouri numerice de date organizate sub formă vectorială și matriceală. De pildă generarea unui interval numeric [-1, 3] cu pasul 0.5 constă în definirea unui vector de date, astfel:

```
X= -1 : 0.5 : 3

X =

   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000
```

Iteraţie - secvenţă de calcul unică, prin care se parcurge un set de instrucţiuni de calcul cu date iniţializate o singură dată, în cadrul unui proces repetitiv. În general, aceasta corespunde unui pas de discretizare a variabilelor pe domeniul de lucru.



În limbajul curent se foloseşte termenul de *incrementare* a valorii indicelui (trecerea la valoarea următoare) sau *decrementare* (trecerea la valoarea anterioară) în cazul calculelor repetitive.
Secvenţa de calcul iterativ poate include de pildă, calculul valorilor succesive ale unei variabile pe un domeniu numeric discretizat cu un pas constant:

$$x_i = x_{i-1} + pas\_x\,,$$

precum şi calculul unor funcţii ce depind de variabilei respective.

Notă.
Foarte multe modele matematice întâlnite în problemele tehnice sunt modele diferenţiale. Acestea se rezolvă numeric prin discretizare şi utilizarea diferenţelor finite de diferite ordine. Astfel:

$$\underbrace{f'(x)=\frac{df}{dx}}_{\text{Forma analitică, pe domeniul infinit}} \cong \underbrace{\frac{\Delta f}{\Delta x}}_{\text{Aproximarea, pe domeniu discretizat}} \Rightarrow \underbrace{\Delta f = f'(x)\cdot\Delta x}_{\text{Modelul discretizat rezolvabil numeric}}$$

Aflarea funcţiei *f(x)* se face prin integrare $\int f'(x)dx$ fie prin metode exacte (numite şi metode analitice – bazate pe aflarea primitivei funcţiei de integrat), fie aproximativ – prin calcul numeric iterativ (repetitiv) pe baza formulei de forma:

$$f_i(x_i) = f_{i-1}(x_{i-1}) + f'(x_{i-1})\Delta x\,,$$

cunoscând *condiţiile iniţiale* (x_0, f_0), şi $f'(x_0)$.

Conceptul de recursivitate.

Un alt concept bazat pe calculul iterativ este *recursivitatea*. Recursivitatea este un model de evoluție în care valoarea unei variabile la un moment dat depinde de valoarea sa la momentul anterior. Cu alte cuvinte starea unui sistem de la pasul anterior (*k-1*) se regăsește explicit în expresia ce descrie starea curentă (*k*). De exemplu:

$$z_k = f(z_{k-1}) + c\,,$$

unde *f* este o funcție, iar *c* reprezintă o constantă.

Modelele recursive pot fi utilizate pentru sisteme dinamice de exemplu cele de tip fractal, pe mulțimea numerelor complexe.

Bibliografie recomandată:

S.Ionita, P.Anghelescu, T.A.Stanescu, *Calcul Numeric Ingineresc. Mediul Matlab,* Ed. MatrixRom, Buc., 2007. (Vezi capitolele incarcate pe platforma de e-learning)