ԳԼՈՒԽ 5

Ցանցային և ավտոմատ դիֆերենցման մեթոդների կիրառումը բույսերում ջերմափոխանակության հավասարման լուծման համար

5.1 Ջերմափոխանակության հավասարման թվային լուծում ցանցային մեթոդով

Ենթադրենք, տրված է հետևյալ դիֆերենցիալ հավասարումների համակարգը.

$$\begin{cases} \frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + Q(x,t) \\ u(x,0) = 0\,, \quad 0 \le x \le l \\ u(0,t) = 0\,, \quad 0 \le t \le T \\ u(l,t) = 0\,, \quad 0 \le t \le T \end{cases} \tag{1}$$

որտեղ $u(x,t)$-ն որտեղ ցույց է տալիս ջերմաստիճանը x դիրքում և t ժամանակի պահին, D-ն դիֆուզիայի գործակիցն է, և հաստատուն է: Այն ցույց է տալիս թե որքան արագ է տարածվում ջերմությունը (նյութը, մասնիկները) միջավայրի մեջ: $Q(x,t)$-ն ջերմաքանակն է, այսինքն միավոր ծավալում և միավոր ժամանակում ստացվող (կամ կլանվող) ջերմության քանակն է: Մեր մոդելում, բույսի հյուսվածքը յուրաքանչյուր x-ում և յուրաքանչյուր t-ում ստանում է ջերմաքանակ, հետևյալ հավասարումով.՝

$$Q(x,t) = x(l-x) * \frac{1}{1+t^2}$$

Մագիստրոսական թեզի և գիտական հոդվածների շրջանակներում, վերցրել ենք D-ի հետևյալ արժեքը.

$$D = 10^{-5}\ \text{մ}^2 * \text{վ}$$

Սահմանային և սկզբնական պայմանները սահմանվել են, հաշվի առնելով որոշ ենթադրություններ, և հյուսվածքի ֆիզիկաքիմիական հատկություններ: $u(x,0) = 0$ սկզբնական պայմանը ցույց է տալիս, որ $t = 0$ պահին բույսի ամբողջ հատվածում (միջավայրում) ջերմաստիճանը հավասար է 0-ի, այսինքն ջերմափոխանակությունը դեռ չի սկսվել, իսկ սա իր հերթին նշանակում է, որ բույսի հյուսվածքը (միջավայրը) համասեռ է, այսինքն ունի նույն ջերմային հատկությունները ամբողջ $x_0 = 0$ երկարությամբ, $t = 0$ պահին: $u(0,t) = 0$ և $u(l,t) = 0$ սահմանային պայմանները ցույց են տալիս, որ բույսի մոդելավորված հատվածի ծայրերում ջերմաստիճանը պահվում է հաստատուն, այսինքն եզրային հատվածները համարվում են ջերմաստիճանով (կայուն) հաստատուն հատվածներ:

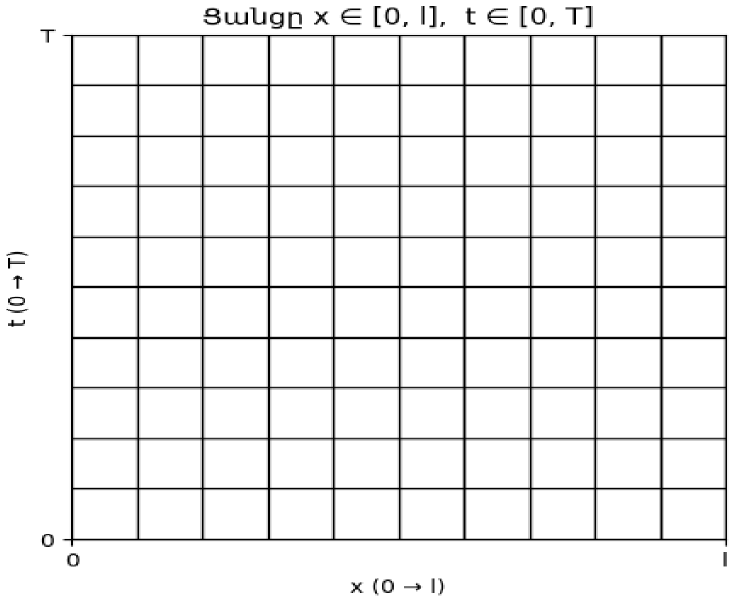
Նախքան խնդրին անցնելը, նախ դիտարկենք ընդհանուր դեպքը և կառուցենք ցանցը:

Քայլ 1.

Առաջին քայլով, տարածությունը և ժամանակը բաժանում ենք համապատասխանաբար M և N հավասար մասերի, Δx և Δt քայլերով (նկ. 5.1).

$$\Delta x = \frac{l}{N} \ , \qquad x_i = i * \Delta x, \qquad i = 0,1,2,\dots,N$$

$$\Delta t = \frac{T}{M} \ , \qquad t_n = n * \Delta t, \qquad n = 0,1,2,\dots,M$$



Նկար 5.1

Քայլ 2.

Երկրորդ քայլով, դիֆերենցիալ հավասարումը մոտարկում ենք տարբերութային հավասարումով.

$$\frac{\partial u}{\partial t} \approx \frac{U_i^{n+1} - U_i^n}{\Delta t} \ , \qquad \frac{\partial^2 u}{\partial x^2} \approx \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2}$$

որտեղ U_i^n-ը ցույց է տալիս ֆունկցիայի արժեքը $u(x,t)$ կետում.՝

$$U_i^n \approx u(x_i, t_n)$$

i-ն տարածական ցանցի ինդեքսն է, իսկ n-ը ժամանակային ցանցի ինդեքսն է: Համաձայն այս նշանակման, ջերմաքանակի համար կստանանք հետևյալ բանաձևը.

$$Q_i^n = x_i(l - x_i) * \frac{1}{1 + {t_n}^2}$$

Քայլ 3.

Երրորդ քայլով, տեղադրելով (1) համակարգի առաջին հավասարման մեջ, կստանանք.

$$\begin{cases} \dfrac{U_i^{n+1} - U_i^n}{\Delta t} = D\dfrac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2} + Q_i^n \\ U_n^0 = 0\,, \quad (\text{ձախ սահման}) \\ U_i^0 = 0\,, \quad (\text{ներքևի սահման}) \\ U_N^n = 0\,, \quad (\text{աջ սահման}) \end{cases}$$

Մեծ հաշվարկներից խուսափելու համար, կատարենք նշանակում.

$$\lambda = \frac{D * \Delta t}{(\Delta x)^2}, \quad => \begin{cases} U_i^{n+1} = U_i^n + \lambda(U_{i+1}^n - 2U_i^n + U_{i-1}^n) + \Delta t Q_i^n \\ U_n^0 = 0 \\ U_i^0 = 0 \\ U_N^n = 0 \end{cases} \tag{2}$$

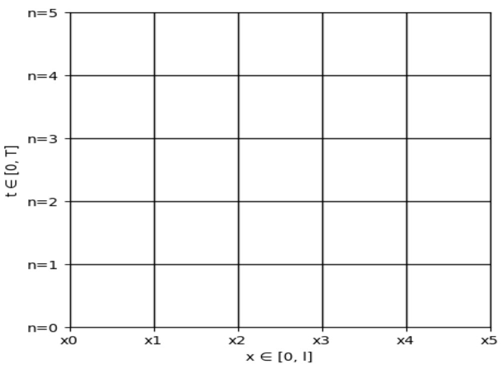
Այժմ դիտարկենք մեր մասնավոր դեպքը: Ենթադրենք մեր ցանցը, ըստ ժամանակի, և ըստ տարածության բաժանված է համապատասխանաբար 5 հավասար մասերի: Ենթադրենք ունենք 10սմ = 0.1մ հյուսվածք, և այն բաժանում ենք $N = 5$ հավասար մասերի, $i = 0,1,2,3,4,5$: Ենթադրենք նաև ունենք 100վ ժամանակ, որը նույնպես բաժանված է $M = 5$ հավասար մասերի, $n = 0,1,2,3,4,5$ (նկ. 5.2 և նկ.5.3).

քայլը.՝ $\Delta x = \frac{l}{N} = \frac{0.1}{5} = 0.02$ մ,

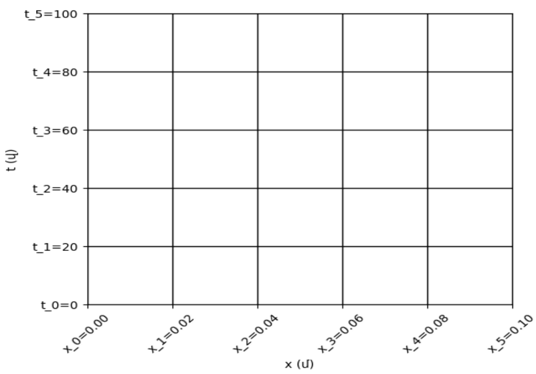
կետերը.՝ $x_0 = 0,\ x_1 = 0.02,\ x_2 = 0.04,\ x_3 = 0.06,\ x_4 = 0.08,\ x_5 = 0.1,$

քայլը.՝ $\Delta t = \frac{T}{M} = \frac{100}{5} = 20$ վ,

կետերը.՝ $t_0 = 0,\ t_1 = 20,\ t_2 = 40,\ t_3 = 60,\ t_4 = 80,\ t_5 = 10.$



Նկար 5.2



Նկար 5.3

Քայլ 1.

Կատարենք առաջին ժամանակային քայլը ($n = 0, \rightarrow\ n = 1$)։ Քանի որ ժամանակի $t = 0$ պահին, ամբողջ հատվածում ջերմաստիճանը 0 է, ապա.

$$U(x,0) = 0, \text{ բոլոր } x\text{-որի համար}$$

որին համապատասխանում է $n = 0$ շերտը ցանցում.

$$n = 0, \quad => U_0^0 = U_1^0 = U_2^0 = U_3^0 = U_4^0 = U_5^0 = 0:$$

(2)-րդ համակարգի առաջին հավասարման մեջ, տեղադրելով $n = 0$, կստանանք հետևյալ բանաձևը.

$$U_i^1 = U_i^0 + \lambda(U_{i+1}^0 - 2U_i^0 + U_{i-1}^0) + \Delta t Q_i^0$$

Քայլ 2.

Այժմ, մինչ $U_1^1, U_2^1, U_3^1, U_4^1$ արժեքներին անցնելը (Աղյուսակ 5.1), հաշվենք λ-ն և Q_i^n արժեքները, քանի որ հետագա բոլոր հաշվարկներում միշտ պետք է գալու.

$$\lambda = \frac{D * \Delta t}{(\Delta x)^2} = \frac{10^{-5} * 20}{(0.02)^2} = 0.5,$$

եթե $n = 0$, => $Q_i^0 = x_i(l - x_i)$

եթե $i = 0$, => $Q_0^0 = x_0(l - x_0) = 0(0.1 - 0) = 0$

եթե $i = 1$, => $Q_1^0 = x_1(l - x_1) = 0.02(0.1 - 0.02) = 0.0016$

եթե $i = 2$, => $Q_2^0 = x_2(l - x_2) = 0.04(0.1 - 0.04) = 0.0024$

եթե $i = 3$, => $Q_3^0 = x_3(l - x_3) = 0.06(0.1 - 0.06) = 0.0024$

եթե $i = 4$, => $Q_4^0 = x_4(l - x_4) = 0.08(0.1 - 0.08) = 0.0016$

եթե $i = 5$, => $Q_5^0 = x_5(l - x_5) = 0.1(0.1 - 0.1) = 0$

Աղյուսակ 5.1

i	0	1	2	3	4	5
Q_i^0	0	0.0016	0.0024	0.0024	0.0016	0

եթե $n = 1$, => $Q_i^1 = x_i(l - x_i)\frac{1}{1+{t_1}^2}$

եթե $i = 0$, => $Q_0^1 = \frac{x_0(l-x_0)}{1+{t_1}^2} = \frac{0(0.1-0)}{1+20^2} = 0$

եթե $i = 1$, => $Q_1^1 = \frac{x_1(l-x_1)}{1+{t_1}^2} = \frac{0.02(0.1-0.02)}{1+20^2} \approx 4 * 10^{-6}$

եթե $i = 2$, => $Q_2^1 = \frac{x_2(l-x_2)}{1+{t_1}^2} = \frac{0.04(0.1-0.04)}{1+20^2} \approx 6 * 10^{-6}$

Եթե $i = 3$, => $Q_3^1 = \frac{x_3(l-x_3)}{1+{t_1}^2} = \frac{0.06(0.1-0.06)}{1+20^2} \approx 6 * 10^{-6}$

Եթե $i = 4$, => $Q_4^1 = \frac{x_4(l-x_4)}{1+{t_1}^2} = \frac{0.08(0.1-0.08)}{1+20^2} \approx 4 * 10^{-6}$

Եթե $i = 5$, => $Q_5^1 = \frac{x_5(l-x_5)}{1+{t_1}^2} = \frac{0.1(0.1-0.1)}{1+20^2} = 0$

Աղյուսակ 5.2

i	0	1	2	3	4	5
Q_i^1	0	$4 * 10^{-6}$	$6 * 10^{-6}$	$6 * 10^{-6}$	$4 * 10^{-6}$	0

Եթե $n = 2$, => $Q_i^2 = x_i(l - x_i)\frac{1}{1+{t_2}^2}$

Եթե $i = 0$, => $Q_0^2 = \frac{x_0(l-x_0)}{1+{t_2}^2} = \frac{0(0.1-0)}{1+40^2} = 0$

Եթե $i = 1$, => $Q_1^2 = \frac{x_1(l-x_1)}{1+{t_2}^2} = \frac{0.02(0.1-0.02)}{1+40^2} \approx 9.9 * 10^{-7}$

Եթե $i = 2$, => $Q_2^2 = \frac{x_2(l-x_2)}{1+{t_2}^2} = \frac{0.04(0.1-0.04)}{1+40^2} \approx 15 * 10^{-7}$

Եթե $i = 3$, => $Q_3^2 = \frac{x_3(l-x_3)}{1+{t_2}^2} = \frac{0.06(0.1-0.06)}{1+40^2} \approx 15 * 10^{-7}$

Եթե $i = 4$, => $Q_4^2 = \frac{x_4(l-x_4)}{1+{t_2}^2} = \frac{0.08(0.1-0.08)}{1+40^2} \approx 9.9 * 10^{-7}$

Եթե $i = 5$, => $Q_5^2 = \frac{x_5(l-x_5)}{1+{t_2}^2} = \frac{0.1(0.1-0.1)}{1+40^2} = 0$

Աղյուսակ 5.3

i	0	1	2	3	4	5
Q_i^2	0	$9.9 * 10^{-7}$	$15 * 10^{-7}$	$15 * 10^{-7}$	$9.9 * 10^{-7}$	0

Եթե $n = 3$, => $Q_i^3 = x_i(l - x_i)\frac{1}{1+{t_3}^2}$

Եթե $i = 0$, => $Q_0^3 = \frac{x_0(l-x_0)}{1+{t_3}^2} = \frac{0(0.1-0)}{1+60^2} = 0$

Եթե $i = 1$, => $Q_1^3 = \frac{x_1(l-x_1)}{1+{t_3}^2} = \frac{0.02(0.1-0.02)}{1+60^2} \approx 4.4 * 10^{-7}$

Եթե $i = 2$, => $Q_2^3 = \frac{x_2(l-x_2)}{1+{t_3}^2} = \frac{0.04(0.1-0.04)}{1+60^2} \approx 6.6 * 10^{-7}$

Եթե $i = 3$, => $Q_3^3 = \frac{x_3(l-x_3)}{1+{t_3}^2} = \frac{0.06(0.1-0.06)}{1+60^2} \approx 6.6 * 10^{-7}$

Եթե $i = 4$, => $Q_4^3 = \frac{x_4(l-x_4)}{1+{t_3}^2} = \frac{0.08(0.1-0.08)}{1+60^2} \approx 4.4 * 10^{-7}$

Եթե $i = 5$, => $Q_5^3 = \frac{x_5(l-x_5)}{1+{t_3}^2} = \frac{0.1(0.1-0.1)}{1+60^2} = 0$

Աղյուսակ 5.4

i	0	1	2	3	4	5
Q_i^3	0	$4.4 * 10^{-7}$	$6.6 * 10^{-7}$	$6.6 * 10^{-7}$	$4.4 * 10^{-7}$	0

եթե $n = 4$, => $Q_i^4 = x_i(l - x_i)\frac{1}{1+{t_4}^2}$

եթե $i = 0$, => $Q_0^4 = \frac{x_0(l-x_0)}{1+{t_4}^2} = \frac{0(0.1-0)}{1+80^2} = 0$

եթե $i = 1$, => $Q_1^4 = \frac{x_1(l-x_1)}{1+{t_4}^2} = \frac{0.02(0.1-0.02)}{1+80^2} \approx 2.4 * 10^{-7}$

եթե $i = 2$, => $Q_2^4 = \frac{x_2(l-x_2)}{1+{t_4}^2} = \frac{0.04(0.1-0.04)}{1+80^2} \approx 3.7 * 10^{-7}$

եթե $i = 3$, => $Q_3^4 = \frac{x_3(l-x_3)}{1+{t_4}^2} = \frac{0.06(0.1-0.06)}{1+80^2} \approx 3.7 * 10^{-7}$

եթե $i = 4$, => $Q_4^4 = \frac{x_4(l-x_4)}{1+{t_4}^2} = \frac{0.08(0.1-0.08)}{1+80^2} \approx 2.4 * 10^{-7}$

եթե $i = 5$, => $Q_5^4 = \frac{x_5(l-x_5)}{1+{t_4}^2} = \frac{0.1(0.1-0.1)}{1+80^2} = 0$

Աղյուսակ 5.5

i	0	1	2	3	4	5
Q_i^4	0	$2.4 * 10^{-7}$	$3.7 * 10^{-7}$	$3.7 * 10^{-7}$	$2.4 * 10^{-7}$	0

եթե $n = 5$, => $Q_i^5 = x_i(l - x_i)\frac{1}{1+{t_5}^2}$

եթե $i = 0$, => $Q_0^5 = \frac{x_0(l-x_0)}{1+{t_5}^2} = \frac{0(0.1-0)}{1+100^2} = 0$

եթե $i = 1$, => $Q_1^5 = \frac{x_1(l-x_1)}{1+{t_5}^2} = \frac{0.02(0.1-0.02)}{1+100^2} \approx 1.5 * 10^{-7}$

եթե $i = 2$, => $Q_2^5 = \frac{x_2(l-x_2)}{1+{t_5}^2} = \frac{0.04(0.1-0.04)}{1+100^2} \approx 2.3 * 10^{-7}$

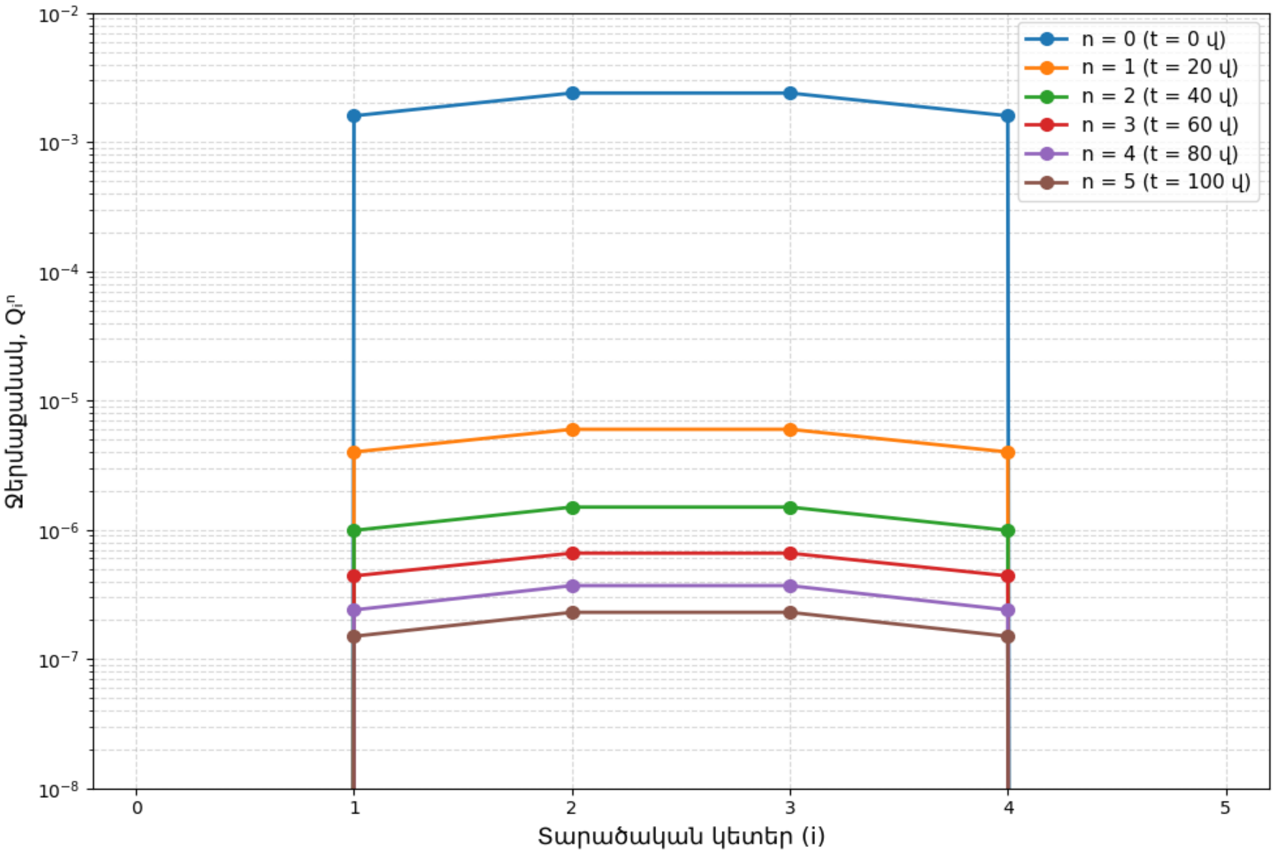
եթե $i = 3$, => $Q_3^5 = \frac{x_3(l-x_3)}{1+{t_5}^2} = \frac{0.06(0.1-0.06)}{1+100^2} \approx 2.3 * 10^{-7}$

եթե $i = 4$, => $Q_4^5 = \frac{x_4(l-x_4)}{1+{t_5}^2} = \frac{0.08(0.1-0.08)}{1+100^2} \approx 1.5 * 10^{-7}$

եթե $i = 5$, => $Q_5^5 = \frac{x_5(l-x_5)}{1+{t_5}^2} = \frac{0.1(0.1-0.1)}{1+100^2} = 0$

Աղյուսակ 5.6

i	0	1	2	3	4	5
Q_i^5	0	$1.5 * 10^{-7}$	$2.3 * 10^{-7}$	$2.3 * 10^{-7}$	$1.5 * 10^{-7}$	0



Նկար 5.4

Պատկերված գրաֆիկը (նկ. 5.4) և (Աղյուսակ 5.1 - Աղյուսակ 5.6) արժեքները ցույց են տալիս որ ժամանակի ընթացքում ջերմաքանակը նվազում է բոլոր կետերում, կենտրոնից դեպի եզրեր տարածվելով և գրեթե հավասարաչափ դառնալով: Դա բնորոշ է ջերմային պրոցեսին, որտեղ սկզբնական էներգիան աստիճանաբար բաշխվում է ամբողջ միջավայրում, մինչև հավասարակշռության վիճակին հասնելը:

Քայլ 3.

Այժմ, կատարենք առաջին ժամանակային քայլը ($n = 0, \rightarrow\ n = 1$): Հաշվենք $U_1^1, U_2^1, U_3^1, U_4^1$ (Աղյուսակ 5.7) արժեքները օգտվելով հետևյալ բանաձից.

$$U_i^1 = U_i^0 + \lambda(U_{i+1}^0 - 2U_i^0 + U_{i-1}^0) + \Delta t Q_i^0$$

եթե $i = 1$, =>
$$U_1^1 = U_1^0 + \lambda(U_2^0 - 2U_1^0 + U_0^0) + \Delta t Q_1^0$$
$$U_1^1 = 0 + 0.5(0 - 2 * 0 + 0) + 0.032 = 0.032$$

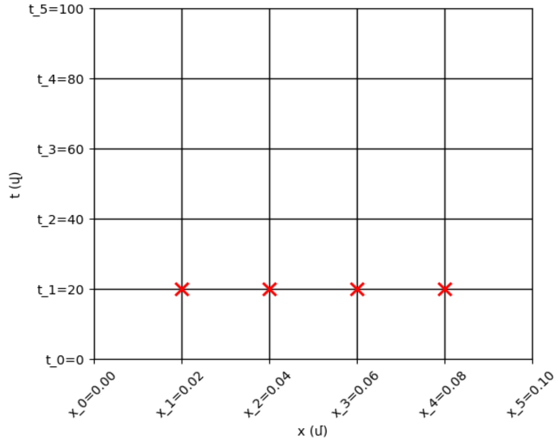
եթե $i = 2$, =>
$$U_2^1 = U_2^0 + \lambda(U_3^0 - 2U_2^0 + U_1^0) + \Delta t Q_2^0$$
$$U_2^1 = 0 + 0.5(0 - 2 * 0 + 0) + 0.048 = 0.048$$

եթե $i = 3$, =>
$$U_3^1 = U_3^0 + \lambda(U_4^0 - 2U_3^0 + U_2^0) + \Delta t Q_3^0$$
$$U_3^1 = 0 + 0.5(0 - 2 * 0 + 0) + 0.048 = 0.048$$

եթե $i = 4$, =>
$$U_4^1 = U_4^0 + \lambda(U_5^0 - 2U_4^0 + U_3^0) + \Delta t Q_4^0$$

$$U_4^1 = 0 + 0.5(0 - 2 * 0 + 0) + 0.032 = 0.032$$

Առաջին ժամանակային քայլից հետո, կստանանք հետևյալ արժեքները (նկ. 5.5).



Նկար 5.5

Աղյուսակ 5.7

i	0	1	2	3	4	5
U_i^1	0	0.032	0.048	0.048	0.032	0

<u>Քայլ 4.</u>

Կատարենք երկրորդ ժամանակային քայլը ($n = 1, \rightarrow\ n = 2$)։ (2)-րդ համակարգի առաջին հավասարման մեջ, տեղադրելով $n = 1$, կստանանք հետևյալ բանաձևը.

$$U_i^2 = U_i^1 + \lambda(U_{i+1}^1 - 2U_i^1 + U_{i-1}^1) + \Delta t Q_i^1։$$

եթե $i = 1$, => $$U_1^2 = U_1^1 + \lambda(U_2^1 - 2U_1^1 + U_0^1) + \Delta t Q_1^1$$

$$U_1^2 = 0.032 + 0.5(0.048 - 2 * 0.032 + 0) + 8 * 10^{-5} = 0.02408$$

եթե $i = 2$, => $$U_2^2 = U_2^1 + \lambda(U_3^1 - 2U_2^1 + U_1^1) + \Delta t Q_2^1$$

$$U_2^2 = 0.048 + 0.5(0.048 - 2 * 0.048 + 0.032) + 12 * 10^{-5} = 0.04012$$

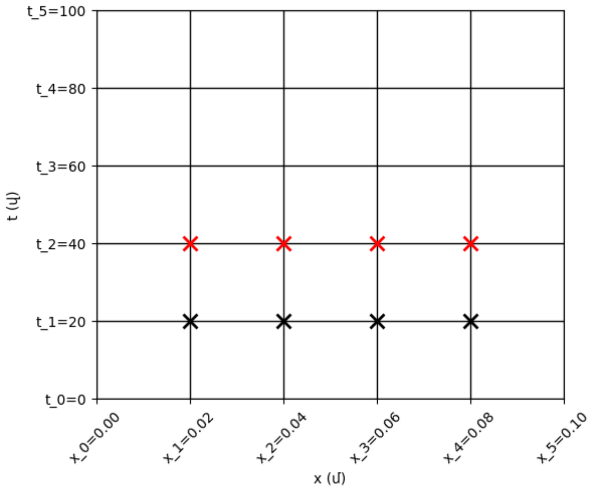
եթե $i = 3$, => $$U_3^2 = U_3^1 + \lambda(U_4^1 - 2U_3^1 + U_2^1) + \Delta t Q_3^1$$

$$U_3^2 = 0.048 + 0.5(0.032 - 2 * 0.048 + 0.048) + 12 * 10^{-5} = 0.04012$$

եթե $i = 4$, => $$U_4^2 = U_4^1 + \lambda(U_5^1 - 2U_4^1 + U_3^1) + \Delta t Q_4^1$$

$$U_4^2 = 0.032 + 0.5(0 - 2 * 0.032 + 0.048) + 8 * 10^{-5} = 0.02408$$

Երկրորդ ժամանակային քայլից հետո, կստանանք հետևյալ արժեքները (նկ. 5.6 և Աղյուսակ 5.8).



Նկար 5.6

Աղյուսակ 5.8

i	0	1	2	3	4	5
U_i^2	0	0.02408	0.04012	0.04012	0.02408	0

Քայլ 5.

Կատարենք երրորդ ժամանակային քայլը ($n = 2, \rightarrow\ n = 3$)։ (2)-րդ համակարգի առաջին հավասարման մեջ, տեղադրելով $n = 2$, կստանանք հետևյալ բանաձևը.

$$U_i^3 = U_i^2 + \lambda(U_{i+1}^2 - 2U_i^2 + U_{i-1}^2) + \Delta t Q_i^2:$$

եթե $i = 1$, => $$U_1^3 = U_1^2 + \lambda(U_2^2 - 2U_1^2 + U_0^2) + \Delta t Q_1^2$$

$$U_1^3 = 0.02408 + 0.5(0.04012 - 2 * 0.02408 + 0) + 19.8 * 10^{-6} = 0.0200798$$

եթե $i = 2$, => $$U_2^3 = U_2^2 + \lambda(U_3^2 - 2U_2^2 + U_1^2) + \Delta t Q_2^2$$

$$U_2^3 = 0.04012 + 0.5(0.04012 - 2 * 0.04012 + 0.02408) + 3 * 10^{-5} = 0.03213$$

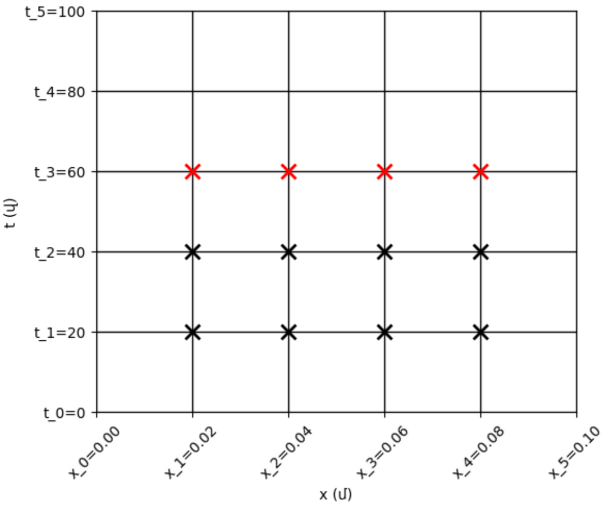
եթե $i = 3$, => $$U_3^3 = U_3^2 + \lambda(U_4^2 - 2U_3^2 + U_2^2) + \Delta t Q_3^2$$

$$U_3^3 = 0.04012 + 0.5(0.02408 - 2 * 0.04012 + 0.04012) + 3 * 10^{-5} = 0.03213$$

եթե $i = 4$, => $$U_4^3 = U_4^2 + \lambda(U_5^2 - 2U_4^2 + U_3^2) + \Delta t Q_4^2$$

$$U_4^3 = 0.02408 + 0.5(0 - 2 * 0.02408 + 0.04012) + 19.8 * 10^{-6} = 0.0200798$$

Երրորդ ժամանակային քայլից հետո, կստանանք հետևյալ արժեքները (նկ. 5.7 և Աղյուսակ 5.9).



Նկար 5.7

Աղյուսակ 5.9

i	0	1	2	3	4	5
U_i^3	0	0.0200798	0.03213	0.03213	0.0200798	0

Քայլ 6.

Կատարենք չորրորդ ժամանակային քայլը ($n = 3, \rightarrow\ n = 4$)։ (2)-րդ համակարգի առաջին հավասարման մեջ, տեղադրելով $n = 3$, կստանանք հետևյալ բանաձևը.

$$U_i^4 = U_i^3 + \lambda(U_{i+1}^3 - 2U_i^3 + U_{i-1}^3) + \Delta t Q_i^3 :$$

եթե $i = 1,\quad =>\qquad U_1^4 = U_1^3 + \lambda(U_2^3 - 2U_1^3 + U_0^3) + \Delta t Q_1^3$

$$U_1^4 = 0.0200798 + 0.5(0.03213 - 2 * 0.0200798 + 0) + 8.8 * 10^{-6} = 0.0160738$$

եթե $i = 2,\quad =>\qquad U_2^4 = U_2^3 + \lambda(U_3^3 - 2U_2^3 + U_1^3) + \Delta t Q_2^3$

$$U_2^4 = 0.03213 + 0.5(0.03213 - 2 * 0.03213 + 0.0200798) + 13.2 * 10^{-6} = 0.0261181$$

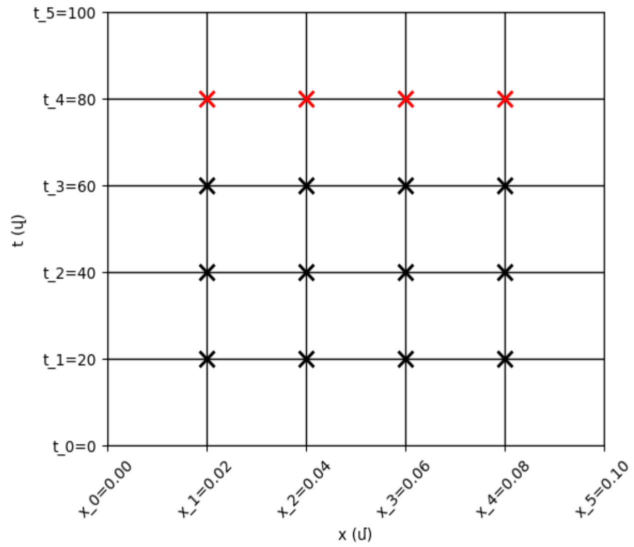
եթե $i = 3,\quad =>\qquad U_3^4 = U_3^3 + \lambda(U_4^3 - 2U_3^3 + U_2^3) + \Delta t Q_3^3$

$$U_3^4 = 0.03213 + 0.5(0.0200798 - 2 * 0.03213 + 0.03213) + 13.2 * 10^{-6} = 0.0261181$$

եթե $i = 4,\quad =>\qquad U_4^4 = U_4^3 + \lambda(U_5^3 - 2U_4^3 + U_3^3) + \Delta t Q_4^3$

$$U_4^4 = 0.0200798 + 0.5(0 - 2 * 0.0200798 + 0.03213) + 8.8 * 10^{-6} = 0.0160738$$

Չորրորդ ժամանակային քայլից հետո, կստանանք հետևյալ արժեքները (նկ. 5.8 և Աղյուսակ 5.10).



Նկար 5.8

Աղյուսակ 5.10

i	0	1	2	3	4	5
U_i^4	0	0.0160738	0.0261181	0.0261181	0.0160738	0

Քայլ 7.

Կատարենք հինգերորդ ժամանակային քայլը ($n = 4, \rightarrow\ n = 5$): (2)-րդ համակարգի առաջին հավասարման մեջ, տեղադրելով $n = 4$, կստանանք հետևյալ բանաձևը.

$$U_i^5 = U_i^4 + \lambda(U_{i+1}^4 -\ 2U_i^4 + U_{i-1}^4) + \Delta t Q_i^4:$$

եթե $i = 1$, $=>$ $U_1^5 = U_1^4 + \lambda(U_2^4 -\ 2U_1^4 + U_0^4) + \Delta t Q_1^4$

$$U_1^5 = 0.0160738 + 0.5(0.0261181 -\ 2 * 0.0160738 + 0) + 4.8 * 10^{-6} = 0.01306385$$

եթե $i = 2$, $=>$ $U_2^5 = U_2^4 + \lambda(U_3^4 -\ 2U_2^4 + U_1^4) + \Delta t Q_2^4$

$$U_2^5 = 0.0261181 + 0.5(0.0261181 -\ 2 * 0.0261181 + 0.0160738) + 7.4 * 10^{-6} = \\ = 0.02110335$$

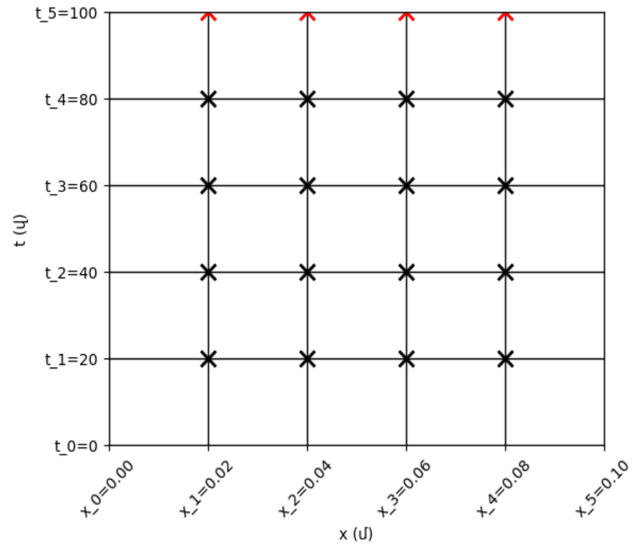
եթե $i = 3$, $=>$ $U_3^5 = U_3^4 + \lambda(U_4^4 -\ 2U_3^4 + U_2^4) + \Delta t Q_3^4$

$$U_3^5 = 0.0261181 + 0.5(0.0160738 -\ 2 * 0.0261181 + 0.0261181) + 7.4 * 10^{-6} = \\ = 0.02110335$$

եթե $i = 4$, $=>$ $U_4^5 = U_4^4 + \lambda(U_5^4 -\ 2U_4^4 + U_3^4) + \Delta t Q_4^4$

$$U_4^5 = 0.0160738 + 0.5(0 -\ 2 * 0.0160738 + 0.0261181) + 4.8 * 10^{-6} = 0.01306385$$

Հինգերորդ ժամանակային քայլից հետո, կստանանք հետևյալ արժեքները (նկ. 5.9 և Աղյուսակ 5.11).



Նկար 5.9

Աղյուսակ 5.11

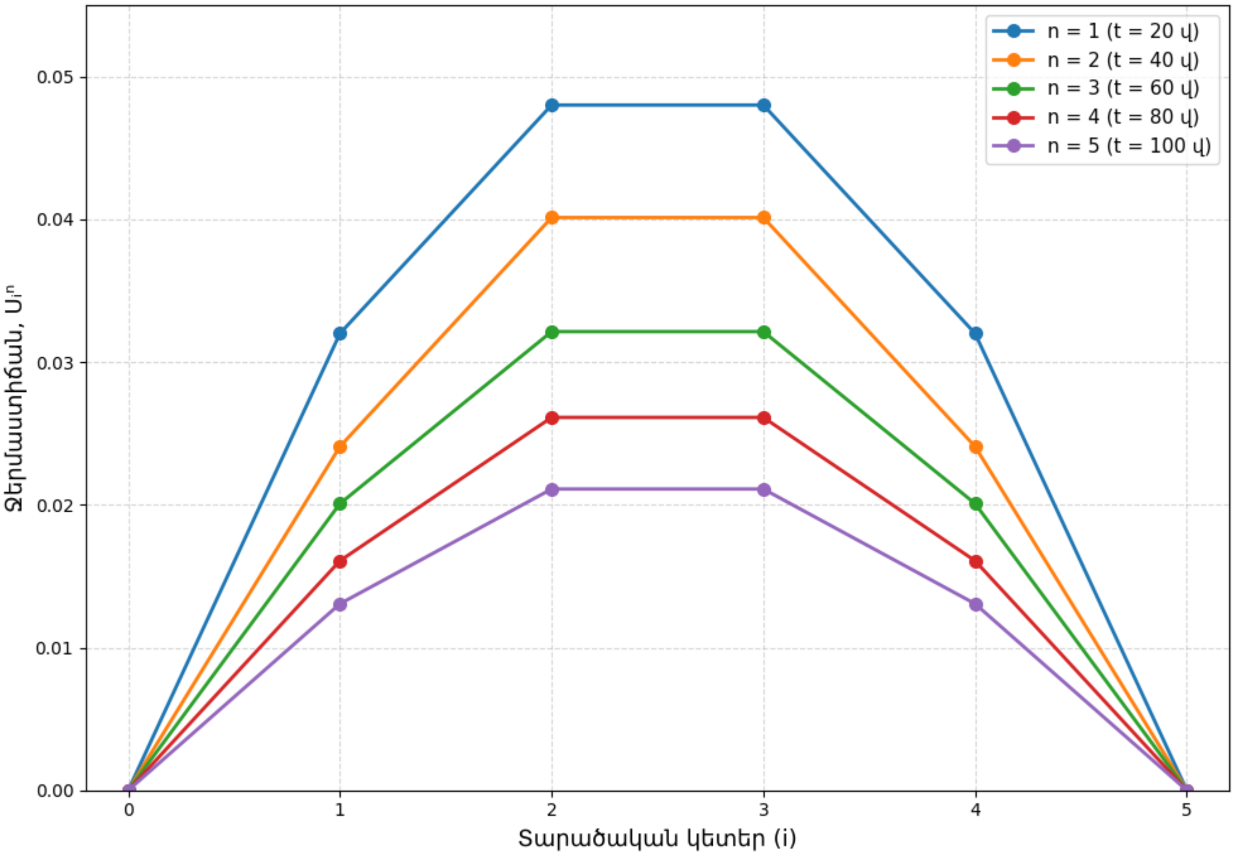
i	0	1	2	3	4	5
U_i^5	0	0.01306385	0.02110335	0.02110335	0.01306385	0

Այսպիսով, ստացվեց որ դիֆերենցիալ հավասարման խնդրի տարբերության մոտարկման արդյունքում, ստացվում է գծային հավասարումների համակարգ, որի լուծման մեթոդները բաժանվում են 2 տեսակի՝ ուղիղ և իտերացիոն:

Ուղիղ են կոչվում այն մեթոդները, որոնց միջոցով կարելի է ստանալ խնդրի ճշգրիտ լուծումը վերջավոր թվով թվաբանական գործողություններից հետո: Օրինակ.՝ մոնոտոն քշման մեթոդը (Թոմասի ալգորիթմ), ոչ մոնոտոն քշման մեթոդը, մատրիցային քշման մեթոդը, և այլն:

Իտերացիոն կոչվում են այն մեթոդները, որոնց միջոցով կարելի է ստանալ խնդրի մոտավոր լուծումը կամայական ճշտությամբ վերջավոր թվով թվաբանական գործողություններից հետո:

Մագիստրոսական թեզում և գիտական հոդվածներում օգտագործվել է իտերացիոն մեթոդը, որի շնորհիվ ստացվել է խնդրի մոտավոր լուծումը: Այժմ հաշվի առնելով Աղյուսակ 5.7 - Աղյուսակ 5.11, և նկ.5.5 – նկ.5.9, կառուցենք մի գրաֆիկ (նկ. 5.10).



Նկար 5.10

Գրաֆիկը ցույց է տալիս ջերմության տարածման դինամիկան բույսի հյուսվածքում, ժամանակի ընթացքում, ջերմաքանակի և հաստատուն դիֆուզիոն գործակցի պայմաններում: Ինչպես տեսնում ենք, սկզբում ջերմաստիճանը բավականին բարձր է, այնուհետև աստիճանաբար նվազում է: Ժամանակի ընթացքում փոփոխությունը դառնում է ավելի աննշան և մոտենում է կայուն մակարդակի: Ֆիզիկայի տեսանկյունից դա նշանակում է որ բույսի հյուսվածքում ջերմաստիճանը այլևս չի նվազում, այսինքն ջերմության փոփոխությունը դադարում է զգալիորեն փոփոխվել:

5.2 Ջերմափոխանակության հավասարման լուծում ավտոմատ մեթոդով

Ենթադրենք, նորից տրված է հետևյալ դիֆերենցիալ հավասարումների համակարգը.

$$\begin{cases} \dfrac{\partial u}{\partial t} = D\dfrac{\partial^2 u}{\partial x^2} + Q(x,t) \\ u(x,0) = 0\,, \quad 0 \le x \le l \\ u(0,t) = 0\,, \quad 0 \le t \le T \\ u(l,t) = 0\,, \quad 0 \le t \le T \end{cases}$$

որը, ինչպես տեսանք նախորդ ենթագլխում ընդունում էր հետևյալ տեսքը.

$$
\begin{cases}
U_i^{n+1} = U_i^n + \lambda\left(U_{i+1}^n - 2U_i^n + U_{i-1}^n\right) + \Delta t Q_i^n \\
U_n^0 = 0 \\
U_i^0 = 0 \\
U_N^n = 0
\end{cases} \tag{3}
$$

<u>Քայլ 1.</u>

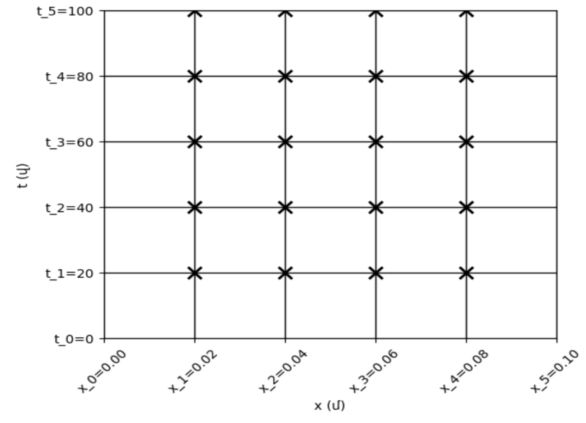
Առաջին քայլով, սահմանում ենք բոլոր ֆիզիկական պարամետրերը, և կատարում նշանակումը.

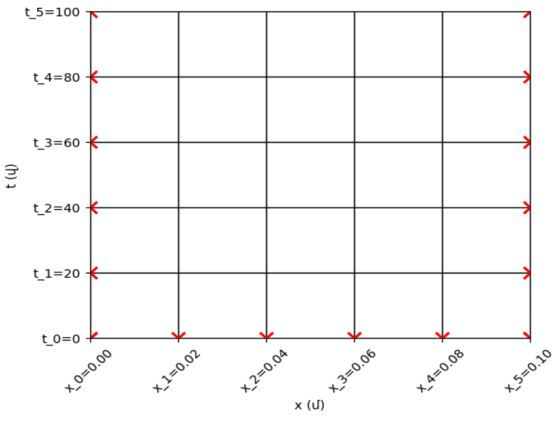
D = 1e-5   N = 5   dt = 20.0   lam = D * dt / dx**2

L = 0.1   M = 5   dx = L / N

<u>Քայլ 2.</u>

Համաձային առաջին քայլին, մենք ունենք $(M+1)$ ժամանակային կետ $(t_0, t_1, t_2, t_3, t_4, t_5)$, և $(N+1)$ հատ տարածական կետ $(x_0, x_1, x_2, x_3, x_4, x_5)$: Մենք ի սկզբանե չգիտենք $u(x,t)$ արժեքները, այդ պատճառով վերցնում ենք (գեներացնում ենք) պատահական բնական թվեր, օգտվելով Գաուսյան բախշումից, հաշվի առնելով սկզբնական և սահմանային պայմանները: Այսինքն մեր մոդելը «սովորելու» է ներքին կետերը $(x_1 t_i, x_2 t_i, x_3 t_i, x_4 t_i,\ i = 1,2,\dots,5)$ (ներքին հանգույցները) (նկ. 5.11), քանի որ ըստ մեր սահմանային և սկզբնական պայմանների՝ $U_0^0 = U_1^0 = U_2^0 = U_3^0 = U_4^0 = U_5^0 = 0$, $U_0^0 = U_0^1 = U_0^2 = U_0^3 = U_0^4 = U_0^5 = 0$, և $U_5^0 = U_5^1 = U_5^2 = U_5^3 = U_5^4 = U_5^5 = 0$ (նկ. 5.12).





Նկար 5. 11   Նկար 5.12

Այդ դեպքում, մեր ներքին կետերից կազմված մատրիցը (նշանակենք $P(n,i)$-ով), կունենա հետևյալ տեսքը.

$$
P(n,i) = \begin{pmatrix} P(1,1) & \cdots & P(1,5) \\ \vdots & \ddots & \vdots \\ P(5,1) & \cdots & P(5,5) \end{pmatrix}
$$

որի տարրերը առաջին քայլում իրենցից ներկայացնում են պատահական բնական թվեր: Քանի որ torch.randn()-ով գեներացնում ենք պատահական թվեր, Գաուսյան բաշխմամբ, դա նշանակում է որ արժեքները սկզբում կարող են լինել շատ մեծ: Մեր խնդիրն է այդ P –ի արժեքները փոքրացնել, քանի որ մեծ արժեքները կարող են առաջացնել անկայուն օպտիմալացում (unstable optimization): Այդ պատճառով մատրիցի բոլոր տարրերը բազմապատկում ենք 0.001-ով: torch.nn.Parameter-ը P մատրիցը դարձնում է learnable parametr (ուսուցանվող պարամետր), որը optimizer-ը (օպտիմալացման ալգորիթմը) կարող է փոխել:

Քայլ 3.

Երրորդ քայլով, քանի որ մենք ի սկզբանե չգիտենք $u(x,t)$ արժեքները, այդ պատճառով օգտագործում ենք մնացորդային մոտարկումը (residual approach-ը): Ըստ մեր ենթադրության, ճշգրիտ $u(x,t)$ արժեքները այն արժեքներն են, որոնք բավարարում են (3) համակարգին (համակարգի հավասարումներին): Կատարենք նշանակում: Մնացորդը նշանակենք r_i^n-ով.

$$r_i^n = U_i^{n+1} - U_i^n - \lambda(U_{i+1}^n - 2U_i^n + U_{i-1}^n) - \Delta t Q_i^n$$

մնացարդը ցույց է տալիս թե որքան ենք շեղվում (3) համակարգի լուծումից: Եթե մնացորդը շատ մոտ է 0-ին՝ $r_i^n \to 0$, ապա դա նշանակում է որ P մատրիցը արդեն համապատասխանում է (3)-ին և մոտավորապես ճշգրիտ լուծումն ենք ստացել: Եթե $r_i^n = 0$, ապա հաշվարկի արդյունքում ստացված U_i^{n+1}-ն ճշգրիտ լուծումն է (3) համակարգի համար, իսկ եթե $r_i^n \neq 0$, ապա U_i^{n+1}-ն (3) համակարգի համար ճշգրիտ լուծում չէ :

Քայլ 4.

Չորրորդ քայլով, սահմանում ենք կորստի ֆունկցիան (Loss function), որը ցույց է տալիս թե որքանով P-ն չի համապատասխանում (բավարարում) (3) համակարգին.

$$Loss = \sum_{n=1}^{M} \sum_{i=1}^{N-1} (r_i^n)^2 \to min$$

Մեր նպատակն է նվազեցնել և ձգտեցնել այն 0-ի՝ $Loss \to 0$: Այժմ հարց է առաջանում, յուրաքանչյուր $P(n,i)$-ի համար, որ ուղղությամբ շարժվենք և ինչ չափով որ կորստի ֆունկցիան փոքրանա: Դրա համար օգտվենք գրադիենտներից (գրադիենտային վայրէջքի մեթոդից):

Քայլ 5.

Ինչպես նշվեց նախորդ քայլերում, սովորելու ընթացքում $P(n,i)$-ն դառնում է դինամիկ փոփոխական, որը փոփոխվում է քայլ առ քայլ, որպեսզի $Loss \to 0$: Այսինքն, գրադիենտները կարող ենք հաշվարկել $P(n,i)$-ի նկատմամբ (շնորհիվ loss.backward()-ի), և օպտիմալացման ալգորիթմը (գրադիենտային վայրէջքի մեթոդը) կարող է թարմացնել պարամետրերը: Երբ գրադիենտը ստանում ենք դրական՝ $\frac{\partial Loss}{\partial P(n,i)} > 0$, դա նշանակում է որ կորստի ֆունկցիան մեծանում է $P(n,i)$-ի աճի դեպքում, հետևաբար պետք է փոքրացնենք $P(n,i)$-ը: Երբ գրադիենտը ստանում ենք բացասական՝ $\frac{\partial Loss}{\partial P(n,i)} < 0$, դա նշանակում է որ կորստի ֆունկցիան մեծանում է $P(n,i)$-ի նվազման դեպքում, հետևաբար պետք է մեծացնենք $P(n,i)$-ը: Որքան մեծ է $\left|\frac{\partial Loss}{\partial P(n,i)}\right|$ –ն, այնքան ավելի մեծ քայլ պետք է անել այդ ուղղությամբ, որպեսզի արագ հասնենք փոքր $Loss$-ի.

$$\Delta P(n,i) = -\eta \frac{\partial Loss}{\partial P(n,i)}$$

Այժմ հարց է առաջանում, ինչի համար ենք վերցրել մինուս նշանը: Քանի որ գրադիենտը ցույց է տալիս $Loss$-ի աճի ուղղությունը, բայց մեր խնդիրն է փոքրացնել $Loss$-ը, ոչ թե մեծացնել, այդ պատճառով վերցրել ենք մինուս նշանը, որը ցույց է տալիս որ շարժվելու ենք դեպի նվազման ուղղությամբ, ոչ թե աճի: Այստեղ η-ն ուսուցման արագությունն է (learning rate) որը կարգավորում է քայլի չափը: Այն սահմանվում է մեր կողմից (ձեռքով), և ցույց է տալիս թե ինչ չափի քայլ պետք է անենք P-ի յուրաքանչյուր տարրի ուղղությամբ:

Քայլ 6.

Հաջորդ քայլում, մենք օգտագործում ենք optimizer.step()-ը (օպտիմալացման ալգորիթմը), որը մեր մոդելում գրադիենտային վայրէջքի մեթոդն է: Այսինքն optimizer-ը վերցնում է գրադիենտը և թարմացնում $P(n,i)$-ները (պարամետրերը).

$$P(n,i) = P(n,i) - \eta \frac{\partial Loss}{\partial P(n,i)}$$

loss.backward() և optimizer.step() քայլերը կրկնվում են հազարավոր անգամ (մեր մոդելում 4500 անգամ): Դրա շնորհիվ փոքրանում է $Loss$-ը և P-ն «սովորում» է այն արժեքները, որոնք լիովին բավարարում են մեր դիֆերենցիալ հավասարումների համակարգին:

Այժմ, հաշվի առնելով այս մեթոդի շնորհիվ ստացված արդյունքները, պատկերենք գրաֆիկ (նկ. 5.13) և կազմենք աղյուսակ (Աղյուսակ 5.12), որոնք ցույց են տալիս ջերմության տարածման դինամիկան բույսի հյուսվածքում, ժամանակի ընթացքում, ջերմաքանակի և հաստատուն դիֆուզիոն գործակցի պայմաններում.

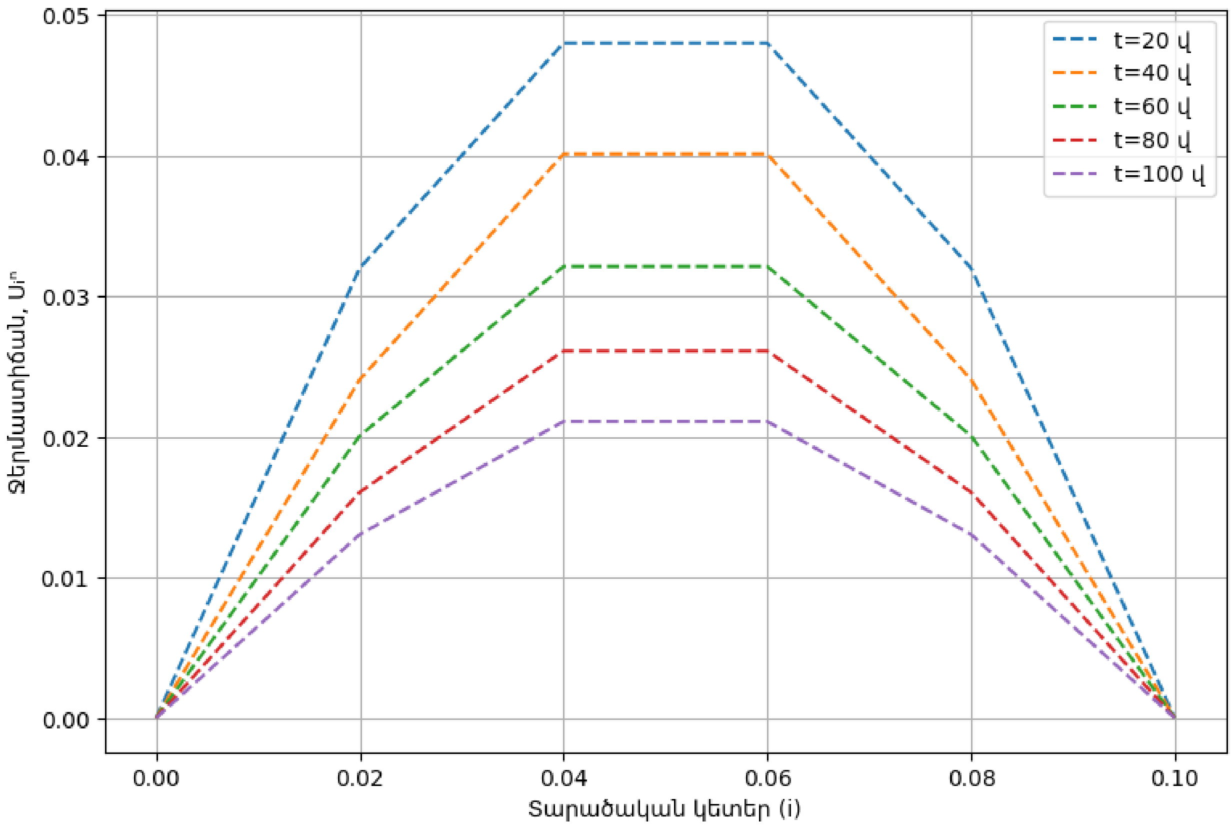
Աղյուսակ 5.12

i	U_i^0	U_i^1	U_i^2	U_i^3	U_i^4	U_i^5
0	0	0	0	0	0	0
1	0	0.032	0.024073	0.020073	0.016068	0.013059
2	0	0.048	0.040109	0.032119	0.026109	0.021096
3	0	0.048	0.040109	0.032119	0.026109	0.021096
4	0	0.032	0.024073	0.020073	0.016068	0.013059
5	0	0	0	0	0	0

(Աղյուսակ 5.13)-ում ներկայացված է յուրաքանչյուր 500 քայլ հետո $Loss$-ի արժեքը, որը գնալով նվազում է, $Loss \to 0$.

Աղյուսակ 5.13

Քայլ	$Loss$
0	$6.901869 * 10^{-3}$
500	$8.916604 * 10^{-5}$
1000	$4.388054 * 10^{-6}$
1500	$2.159585 * 10^{-7}$
2000	$1.062842 * 10^{-8}$
2500	$5.230789 * 10^{-10}$
3000	$2.574339 * 10^{-11}$
3500	$1.266963 * 10^{-12}$
4000	$6.235374 * 10^{-14}$
4500	$3.087285 * 10^{-15}$



Նկար 5.13

Ինչպես տեսնում ենք 2 մեթոդների դեպքում էլ ստացվում է համարյա նույն արժեքները: Բայց այն կարող է շատ տարբերվել երբ ցանցը բաժանում ենք ոչ թե 5*5 մասի, այլ օրինակ 100*100 մասի, կամ loss.backward() և optimizer.step() քայլերը կրկնում ենք 4500-ի փոխարեն 15000 անգամ, կամ օգտագործում ենք այլ օպտիմալացման ալգորիթմներ, օրինակ Adam optimizer, և այլն: Այսինքն մեթոդի կայունությունը և ճկունությունը կախված է բազմաթիվ գործոններից, որոնք թույլ են տալիս ստանալ խնդրի լուծումը տարբեր ճշտությամբ կամ տարբեր հաշվարկային և ժամանակային ռեսուրսներ ծախսելով:

Ջերմափոխանակության հավասարման լուծումը ավտոմատ մեթոդով իրականացվել է Python ծրագրավորման լեզվի միջոցով: Կից ներկայացված է համապատասխան կոդը.

```
import torch
import numpy as np
import matplotlib.pyplot as plt
from time import time
```

```
# --------- 1. Ֆիզիկական պարամետրեր ---------
D = 1e-5          # դիֆուզիոն գործակից
L = 0.1           # հյուսվածքի երկարություն
N = 5             # տարածական բաժանումներ
dx = L / N
dt = 20.0         # վայրկյան
M = 5             # ժամանակային քայլեր

lam = D * dt / dx**2
print(f"dx={dx:.5f} մ, dt={dt:.2f} վ, λ={lam:.5f}")

# --------- 2. Torch  ---------
dtype = torch.float64
device = torch.device("cpu")
torch.manual_seed(0)

P = torch.nn.Parameter(0.001 * torch.randn(M, N-1, dtype=dtype, device=device))
optimizer = torch.optim.SGD([P], lr=1e-2)
num_steps = 4500
print_every = 500

# --------- 3.  Q_i^n ---------
xs = np.linspace(0, L, N+1)
ts = np.linspace(0, M*dt, M+1)

Q = np.zeros((M, N+1))
for n in range(M):
    t_n = ts[n]
    for i in range(N+1):
```

```
        x_i = xs[i]
        Q[n, i] = x_i * (L - x_i) * 1 / (1 + t_n)**2

Q_torch = torch.tensor(Q, dtype=dtype, device=device)

# --------- 4. Loss function = PDE residuals ---------
t0 = time()
for step in range(num_steps):
    optimizer.zero_grad()
    loss = torch.tensor(0.0, dtype=dtype, device=device)

    for n in range(0, M):
        if n == 0:
            u_n = torch.zeros(N+1, dtype=dtype, device=device)
        else:
            u_n = torch.cat([torch.zeros(1, dtype=dtype, device=device), P[n-1], torch.zeros(1, dtype=dtype, device=device)])

        u_np1 = torch.cat([torch.zeros(1, dtype=dtype, device=device), P[n], torch.zeros(1, dtype=dtype, device=device)])
        lap = u_n[2:] - 2*u_n[1:-1] + u_n[:-2]

        resid = u_np1[1:-1] - u_n[1:-1] - lam * lap - dt * Q_torch[n, 1:-1]
        loss += (resid ** 2).sum()

    loss.backward()
    optimizer.step()

    if (step % print_every) == 0 or step == num_steps - 1:
        print(f"Քայլ {step:4d} | կորուստ = {loss.item():.6e}")
```

```
t_elapsed = time() - t0
print(f"\nՎերջ. ժամանակ = {t_elapsed:.2f} վ | Վերջնական կորուստ = {loss.item():.6e}")

# --------- 5. Collect trained solution ---------
U_ad = np.zeros((M+1, N+1), dtype=float)
for n in range(1, M+1):
    U_ad[n, 1:-1] = P.detach().cpu().numpy()[n-1, :]

# --------- 6. Plot result ---------
plt.figure(figsize=(9, 6))
for n in range(1, M+1):
    plt.plot(xs, U_ad[n, :], '--', label=f"t={n*dt:.0f} վ")
plt.xlabel("Տարածական կետեր (i)")
plt.ylabel("Ջերմաստիճան, Uᵢⁿ")
plt.legend()
plt.grid(True)
plt.show()

# --------- 7. Optional print ---------
print("\nU_ad (տողեր՝ ժամանակային շերտեր):")
print(np.round(U_ad, 6))
```