פרויקט: **סימולציה של מעבר חום במוט**

שיטה: Crank-Nicholson

<u>מטרת הפרויקט:</u>

- לפתח שיטה נומרית – שיטת Crank-Nicholson לחישוב טמפרטורה של מוט בזמן ו במקום מסויים
- להוכיח יציבות של שיטה
- להוכיח עיקביות של שיטה
- להוכיח היתכנסות של שיטה
- לפתח ביטוי לשגיאה בשיטה הזאת
- לכתוב תוכנה בשפת C לחישוב טמפרטורה במוט
- לחשב טמפררורה במוט עשוי מצינק עבור

  Initial Condition $f(x) = 40\tan(0.1x)$

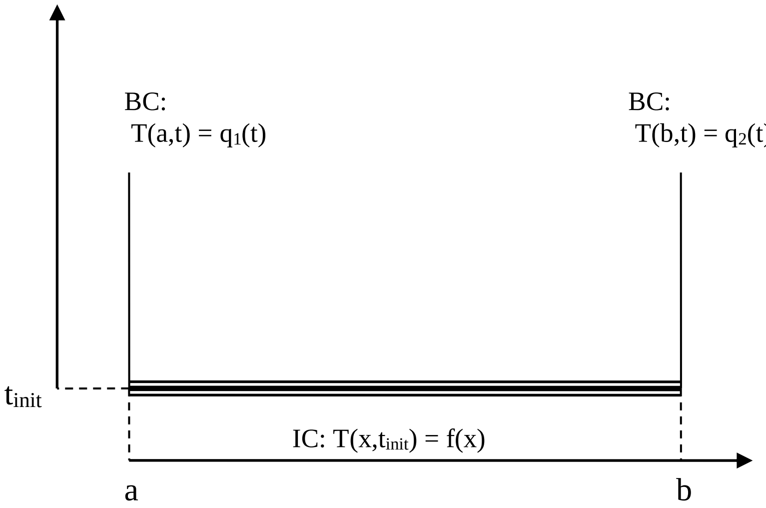
  Left Boundary Condition $q_1(t) = 10\tanh(t)$

  Right Boundary Condition $q_2(t) = 20\sin(5t)$

  בגבולות מ-5 עד 5.4 לפי שיטה הנ"ל
- לבצעה בדיקת תוכנה

<u>מבוא</u>

כדי לבנות מודל סימולציה של מעבר חום נסתקל על הגדרה פיסיקלית של מודל:
יש מוט מבודד, ידוע טמפרטורה על שני קצבות של מוט (Boundary Conditions), ידוע חומר של מוט, ידוע מצב התחלתי (Initial Conditions) – טמפרטורה בכל מקום המוט.



BC - Boundary Conditions – תנאי השפה (טמפרטורה של גוף בנקודות a ו-b)
IC - Initial Condition – תנאי התחלה (טמפרטורה של גוף בהתחלת תהליך)
ניקח כי בסיס של מודל את חוק שימור אנרגיה. ניקח שני זמנים $t+\delta t$ ו- t ואיזשהו גוף שמסתו לא משתנה

$$c\rho T(x,t+\delta t)A_0\delta x = c\rho T(x,t)A_0\delta x + q(x,t)A_0\delta t - q(x+\delta x,t)A_0\delta t \Rightarrow$$

איפו ש:
$q(x,t)A_0\delta t$ - אנרגיה נכנסת
$q(x+\delta x,t)A_0\delta t$ - אנרגיה יוצאת
A_0 - שטח חתך של המוט
c - specific heat, $\left[\frac{kJ}{kg-{}^0C}\right]$, $c=\frac{\Delta E}{m\Delta T}$
$\Delta T = T_0 - T(t)$ איפו
T_0 - טמפרטורה מיסביב הגוף (טמפרטורה של סביבה)
$T(t)$ - טמפרטורה של הגוף
ρ - צפיפות הגוף, $\left[\frac{gr}{cm^3}\right]$
$g(x,t)$ - flux, $g(x,t) = -kT_x(x,t)$ איפו
k - Thermal conductivity, $\left[\frac{Watt}{m-{}^0C}\right]$

$$\Rightarrow c\rho T(x,t+\delta t) = c\rho T(x,t)\delta x + q(x,t)\delta t - q(x+\delta x,t)\delta t \Rightarrow$$
$$\Rightarrow c\rho\left[T(x,t+\delta t) - T(x,t)\right]\delta x = -\left[q(x+\delta x,t) - q(x,t)\right]\Rightarrow$$
$$\Rightarrow c\rho\frac{T(x,t+\delta t)-T(x,t)}{\delta t} = -\frac{q(x+\delta x,t)-q(x,t)}{\delta x} \Rightarrow c\rho T_t = -q_x$$

נניח כי $q=-kT_x$ אז $q_x=-kT_{xx}$ לכן $c\rho T_t = kT_{xx}$

מקבלים משוואה המתאימה למודל של מעבר חום: $T_t(x,t)=\frac{k}{c\rho}T_{xx}(x,t) \Leftrightarrow T_t=\frac{k}{c\rho}T_{xx} \Leftarrow$

$\Leftarrow$ $\boxed{T_t(x,t)=\alpha T_{xx}(x,t)}$, כאשר $\alpha=\frac{k}{c\rho}$

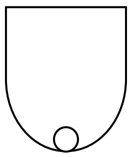
הגדרה של נגזרת ראשונה לפי **הפרש קדמי**: $T_t=\lim_{\Delta t\to 0}\frac{T(t+\Delta t)-T(t)}{\Delta t}+O(\Delta t)$

הגדרה של נגזרת ראשונה לפ **הפרש אחורי**: $T_t=\lim_{\Delta t\to 0}\frac{T(t)-T(t+\Delta t)}{\Delta t}+O(\Delta t)$

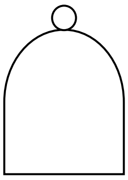
הגדרה של נגזרת שניה לפ **הפרש מרכזי**: $T_t=\lim_{\Delta t\to 0}\frac{T(t-\Delta t)-2T(t)+T(t+\Delta t)}{\Delta t^2}+O(\Delta t^2)$

בעיה נקראת **מוגדרת היטב** אם מתקיימים תנאים הבאים:
א) קיים פתרון
ב) פתרון יחיד
ג) תלות רציפה של פתרון המשוואה בנתוני הקלט

עיקרון maximum – מטבע הדברים אנו יודעים שעם הזמן טמפרטורה של גוף לא תעלה על מקסימום בין הטמפרטורות שלו וסביבתו , וכמו כן לא תירד מתחת למינימום בין אותם טמפרטורות .
יצינות - מערכת נקראת יציבה אם היא שומרת על עיקרון המקסימום ואינה רגישה לשינויים קלים בקלט.
דוגמא מין החיים:



הכדור יציב בכל דחיפה קטנה



הכדור לא יציב בכל דחיפה קטנה

עיקביות – פתרון של בעיה נכונה. שיטה נומרית שאנו משתמשים בה לא מניאה אותנו לפתרון של בעיה אחרת!
משפט Lax – שיטה נקראת מתכנסת אם היא יציבה ועיקבית

דרישות הלקוח

- לבנות מודל לסימולציה של מעבר חום.
- שיטה תחפשת פיתרון בצורה מדויקת
- חישוב מתבצעה זמן קצר
- שיטה אמינה
- לכתוב תוכנה קלה ונוכה לשימוש

קלט של תוכנה:

- הגדרת גבולות
- הגדרת Boundary Condition
- הגדירת Initial Condition
- פרמטרים שכשורים להחומר שממנו עסוי גוף (c, k, ρ)
- דיוק הנדרש

פלט של תוכנה:

- וקטור טמפרטורות של גוף בזמן שנידרש
- וקטור טמפרטורות של גוף במקום הנידרש
- טמפרטורה במקום הטוף מסויים בזמן כלשהו

שיטה Crank-Nicholson

מטרה של שיטה לפתור משוואה דיפרנציאלית חלקית מין הצורה $U_t = \alpha U_{xx}$

כאשר $\alpha = \dfrac{k}{c\rho}$

בסיס לשיטת חישוב של משוואת החום.

$$T_t(x,t) = \frac{T(x,t+\delta t) - T(x,t)}{\delta t} + O(\delta t)$$

$$T_{xx}(x,t) = \frac{T(x+\delta x,t) - 2T(x,t) + T(x-\delta x,t)}{\delta x^2} + O(\delta x^2)$$

משתי המשוואות הנ"ל נקבל קירוב

$$\boxed{\frac{T(x,t+\delta t) - T(x,t)}{\delta t} \approx \alpha \frac{T(x-\delta x,t) - 2T(x,t) + T(x+\delta x,t)}{\delta x^2}} \quad \mathbf{(1)}$$

נגדיר רשת של נקודות (x_n, t_m) **כך ש**

נגדיר N - מספר חלוקות של קטע $[a,b]$

מי פו $\delta x = \dfrac{b-a}{N}$ כך ש:

$$x_0 = a$$
$$x_1 = a + \delta x$$
$$\dots$$
$$x_i = a + i\delta x$$
$$x_N = a + N\delta x = b$$

נגדיר גם $\delta t > 0$ כך ש:

$$t_0 = t_{init}$$
$$t_1 = t_{init} + \delta t$$
$$\dots$$
$$t_j = t_{init} + j\delta t$$

נגדיר קירוב נומרי ל $T(x_n, t_m) \Leftarrow T(x_i, t_j) \approx u_{i,j}$

נגדיר $x = x_i, t = t_j$

נחליף בנוסחה (1) ערכים של T ע"י קירובים של u:

$$\boxed{\frac{u_{i,j+1} - u_{i,j}}{\delta t} = \alpha \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\delta x^2}}$$

נחשב ערך של פונקציה u בנקודה $\left\{ i\delta x, (j+\frac{1}{2})\delta t \right\}$ ונחליף ביטוי $\dfrac{\partial^2 u}{\partial x^2}$ ע"י ממוצע של קירובים ברמה j ו-$j+1$ כמו בציור 1.

ז.א. $\left(\dfrac{\partial u}{\partial t}\right)_{i,j+\frac{1}{2}} = \alpha \left(\dfrac{\partial^2 u}{\partial x^2}\right)_{i,j+\frac{1}{2}}$, נקבל

$$\frac{u_{i,j+1} - u_{i,j}}{\delta t} = \frac{\alpha}{2} \left\{ \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{\delta x^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\delta x^2} \right\}$$

מפה נקבל

$$(*) \quad \boxed{-\lambda u_{i-1,j+1} + (2+2\lambda) u_{i,j+1} - \lambda u_{i+1,j+1} = \lambda u_{i-1,j} + (2-2\lambda) u_{i,j} + \lambda u_{i+1,j}}$$

כאשר $\lambda = \alpha \delta t / \delta x^2$

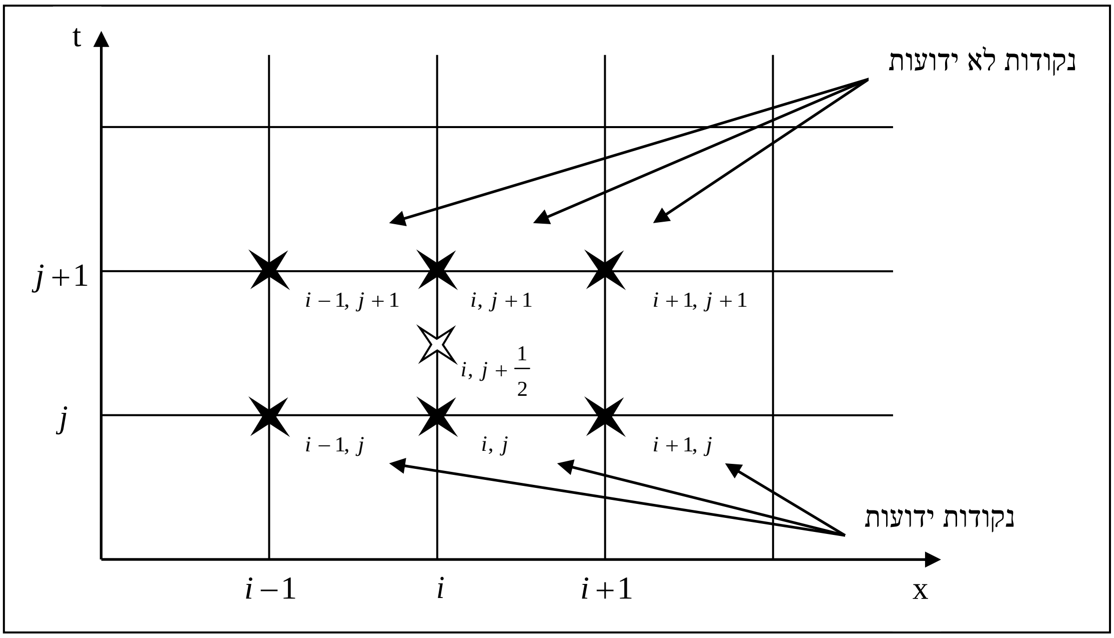
קיבלנו אלגוריתם $\boxed{u_{i,j+1}=\frac{1-\lambda}{1+\lambda}u_{i,j}+\frac{\lambda}{2+2\lambda}\left(u_{i-1,j}+u_{i+1,j}+u_{i-1,j+1}+u_{i+1,j+1}\right)}$

כאשר $i=1,2,\dots,N-1$ ו- $j=0,1,2,\dots$

$u_{i,0}=f(x_i)$ - קירוב לתנאי התחלה

$\begin{cases}u_{0,j}=g_1(t_j)\\ u_{N,j}=g_2(t_j)\end{cases}$ - קירוב לתנאי שפה

ציור 1



חילקנו גוף ל- N קטעים לפי רוחב אז יש לנו N נקודות ידועות ברמה כאשר $j=0$ ו- $i=1,2,\dots,N$. זה נותן $N-1$ משוואות לפי (*) בשביל $N-1$ ערכים לא ידועים ברמה כאשר $j=1$. ניתן למצוא u בכל נקודה ברמה $j=1$. בדיוק באותה צורה ניתן למצוא בל N נקודות ברמה כאשר $j=2$ ע"י $N-1$ משוואות שניבנאו ע"ס נקודות של רמה כאשר $j=1$, וכו'.

בדיקת יציבות של השיטה:

את היציבות של השיטה נבדוק לפי שיטה של **von Neumann** :

נחליף כל נקודה בביטוי $\varepsilon\ u_m^n=\xi^n e^{i\Delta xm\gamma}$ כאשר:

γ - תדר

ξ - גובה של תדר

ε - גורם קטן כלשהו

$i=\sqrt{-1}$

ונדרוש : $|\xi|\le 1$ כדי ששיטה תהיה יציבה.

$$\varepsilon\xi^{n+1}e^{im\gamma\Delta x}=\frac{1-\lambda}{1+\lambda}\varepsilon\xi^n e^{im\gamma\Delta x}+\frac{\lambda}{2+2\lambda}\left(\varepsilon\xi^n e^{i(m+1)\gamma\Delta x}+\varepsilon\xi^n e^{i(m-1)\gamma\Delta x}+\varepsilon\xi^{n+1}e^{i(m+1)\gamma\Delta x}+\varepsilon\xi^{n+1}e^{i(m-1)\gamma\Delta x}\right)\Rightarrow$$

$$\xi=\frac{1-\lambda}{1+\lambda}+\frac{\lambda}{2+2\lambda}\left(e^{i\gamma\Delta x}+e^{-i\gamma\Delta x}+\xi e^{i\gamma\Delta x}+\xi e^{-i\gamma\Delta x}\right)\Rightarrow\xi=\frac{1-\lambda}{1+\lambda}+\frac{\lambda}{1+\lambda}\left(\left(1+\xi\right)\cos(\gamma\delta x)\right)\Rightarrow$$

$$\xi=\frac{1-\lambda}{1+\lambda}+\frac{\lambda}{1+\lambda}\left((1+\xi)\cos(\gamma\delta x)\right)\Rightarrow\xi(1+\lambda-\lambda\cos(\gamma\delta x))=1-\lambda+\lambda\cos(\gamma\delta x)\Rightarrow$$

$$\boxed{\xi=\frac{1-\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}}$$

לפי שיטת von Neumann נדרוש $-1\le\frac{1-\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}\le 1$

1) $\frac{1-\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}\le 1$ מתקיים תמיד כי מונה תמיד קטן ממחנה

2) $\frac{1-\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}\ge -1$ $\Leftarrow$ בגלל שמחנה תמיד חיובי לכן $\frac{1-\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}+1\ge 0$ $\Leftarrow$

$\frac{1-\lambda(1-\cos(\gamma\delta x))+1+\lambda(1-\cos(\gamma\delta x))}{1+\lambda(1-\cos(\gamma\delta x))}\ge 0 \Leftarrow \frac{2}{1+\lambda(1-\cos(\gamma\delta x))}\ge 0 \Leftarrow$ מחנה תמיד חיובי

לכן תנאי מתקיים עבור כל λ

ע"ס 1 ו-2 מקבלים $|\xi|\le 1$, לכן שיטה יציבה.

<u>בדיקת עיקביות של השיטה:</u>

לפי טור טיילור

$$T(x,t+\delta t)=T(x,t)+\delta t T_t(x,t)+\frac{\delta t^2}{2}T_{tt}(x,t)+\frac{\delta t^3}{3!}T_{ttt}(x,t)+\ldots\Rightarrow$$

$$\frac{T(x,t+\delta t)-T(x,t)}{\delta t}=T_t(x,t)+\frac{\delta t}{2}T_{tt}(x,t)+\frac{\delta t^2}{3!}T_{ttt}(x,t)+\ldots\Rightarrow$$

$$\boxed{\frac{T(x,t+\delta t)-T(x,t)}{\delta t}=T_t(x,t)+O(\delta t)}$$

$$T(x+\delta x,t)=T(x,t)+\delta x T_x(x,t)+\frac{\delta x^2}{2}T_{xx}(x,t)+\frac{\delta x^3}{3!}T_{xxx}(x,t)+\ldots;$$

$$T(x-\delta x,t)=T(x,t)-\delta x T_x(x,t)+\frac{\delta x^2}{2}T_{xx}(x,t)-\frac{\delta x^3}{3!}T_{xxx}(x,t)+\ldots$$

$$T(x+\delta x,t)+T(x-\delta x,t)=2T(x,t)+\delta x^2 T_{xx}(x,t)+\frac{2\delta x^4}{4!}T_{xxxx}(x,t)+\ldots$$

$$\frac{T(x+\delta x,t)-2T(x,t)+T(x-\delta x,t)}{\delta x^2}=T_{xx}(x,t)+\frac{2\delta x^2}{4!}T_{xxxx}(x,t)+\frac{2\delta x^6}{6!}T_{xxxxxx}(x,t)+\ldots$$

$$\boxed{T_{xx}(x,t)=\frac{T(x+\delta x,t)-2T(x,t)+T(x-\delta x,t)}{\delta x^2}+O(\delta x^2)}$$

נדרוש שהפרש בין פתרון מדוייק ופיתרון נומרי ישאוף לאפס

$$T_t(x,t)-\alpha T_{xx}(x,t)-\frac{T(x,t+\delta t)-T(x,t)}{\delta t}+\alpha\frac{T(x+\delta x,t)-2T(x,t)+T(x-\delta x,t)}{\delta x^2}\to 0\Rightarrow$$

$$O(\delta t^2)+O(\delta x^2)\to 0$$

זה מתקיים תמיד בגלל ש- $\delta t,\delta x\to 0$

לכן שיטה עיקבית

<u>בדיקת התכנסות של השיטה:</u>

לפי משפט Lax שיטה מתכנסת ע"ס יציבותה ועיקביותה.

אלגוריתם של תוכנה

1) הקלטת נתונים של המשתמש (תחילת וסוף של מוט, זמני התחלה וסיום של מדידה, פרמטרים של חומר שממנו עשוי מוט, דיוק הנדרש של חישובים).
2) תוכנית מוצאת מספר חלוקות אופטימלי באורך המוט שמתאים לדיוק של לקוח.
3) בניית טבלה של טמפרטורה, לפי מקום במוט וזמן, בקובץ **"heat_temp.txt"** עבור מספר אופטימלי של חלוקות מחלק ראשון של התוכנית.

כל החישובים שנעשו בתוכנית מתבצעים עבור $\lambda = 1$ **אז:**

1) $$u_{i,j+1} = \frac{1}{4}\left(u_{i-1,j} + u_{i+1,j} + u_{i-1,j+1} + u_{i+1,j+1}\right)$$

2) $$dDT = \frac{(dDX)^2}{dALFA}$$

משתנים:

dC – Specific heat
dR – Consistence of material
dK – Thermal conductivity
power - חזקה של 2 נותנת מספר חלוקות של מוט לפי אורך
dDX - אורך קטה אחת לפי חלוקת מוט
dDT - אורך קטה אחת של זמן לפי נוסחה
dA – התחלתית של מוט
dB – סוף של מוט
dTinit – זמן התחלתי של מדידה
dTend – זמן סיום של מדידה
dTOL – דיוק של חישוב טמפרטורה

פונקציות:

double leftBoundaryCondition(double)
double rightBoundaryCondition(double)
double initialCondition(double)
שלושת הפונקציות מאועדות להגדיר תנאי התחלה של טמפרטורה בזמן התחלה של מדידה.

void vNextVector(double *, int, double) – פונקציה בונה וקטור של טמפרטורה (פתרון N משוואות עם N ניילמים) ברמת זמן מסויימת ע"ס וקטור של רמה קודמת

void vNextIteration(double *, double, double, int, double, double, double, double) – פונקציה ע"ס תנאי התחלה בונה וקטור של טמפרטורה בזמן התחלתי של מדידה ובעזרת פונקציה vNextVector() בונה עבור כל רמה של זמן וקטורים של טמפרטורה בנקודות המוט לפי חלוקה המתאימה

bool bCheckTolerance(double *, double *, double , double , double , double) – פונקציה בונה באופן אקראי 10 נקודות על המוט, מחפסת הפרש מקסימלי בין כל 10 נקודות של איטרציה נוכחית וקודמת ואם הפרש הזה קטן דיוק מחזירה ערך **true**, אחרת **false**.

double dTemperatureInX(double *, double, double, double, double) - פונקציה מחזירה ערך של טמפרטורה במקום כלשהו במוט בזמן סופי של מדידה

double dMAX(double *, int) - פונקציה מחזירה מספר מקסימלי מוקטור שהיא מקבלת

void vfInsertIntoFile(char *, double, double, double, double, int, double, double) - פונקציה פותחת קובץ ומחניסה תוצאות של מדידה בצורה של טבלה

char cfMenu(void) - פונקציה שמאפשרת לבחור כמה דרכים של חישובים:

1) לבדיקת טמפרטורה בזמן מסויים בנקודת גוף מסויימת ללחוץ 1
2) ליצירת וקטור של טמפרטורות בזמן מסויים לפי כל אורך של מוט, בחר 2
3) ליצירת וקטור של טמפרטורות בנקודה של המוט מסויימת לפי קטות של זמן, בחר 3
4) אם זיקרון של מחשב מאפשר אז לבנות קובץ כטבלה של טמפרטורות לפי רשת שבונה תוכנית

void vfTemperatureInPointXT(double, double, double, double, double, int) - פונקציה מחפשת טמפרטורמ בזמן מסויים בנקודת מוט מסויימת

void vfTemperatureInTimeT(char *, double, double, double, double, double, int) - פונקציה בונה ומחשבת וקטור של טמפרטורות לפי זמן הנקלט ומחניסה אותו לקובץ ששמו נקלט ע"י פונקציה **main()**. בניית הוקטור נקבעה ע"ס משתנה **power** שקובעה מספר חלוקות של מוט לפי ערכו (2^{power}).

void vfTemperatureInPointX(char *, double, double, double, double, double, int) - פונקציה מחשבת טמפרטורה במוט לפי מקום הנקלט ומחניסה אותה לקובץ ששמו נקלט ע"י פונקציה **main()**.

מספר קטות של זמן נקבע ע"י משתנה **power** $\left(\frac{t-T_{end}}{\delta t}\right)+w$

אם $\left(t-T_{end}\right)\%\delta t=0 \Leftarrow w=0$, אחרת $w=1$

<u>בדיקות של תוכנה</u>

Initial Condition – f(x)
Left Boundary Condition – $g_1(t)$
Right Boundary Condition – $g_2(t)$

כל החישובים נעשו עבור גוף עשוי מצינק שפרמטריו הם: $\boxed{\alpha = 4.13518\times10^{-5}}$

1) נציב $f(x)=10°C, g_1(t)=10°C, g_2(t)=10°C$, dTOL = 0.000001
טמפרטורה קבוה ושווה בכל מקום במוט לאורך כל הזמן, לכן פטרון אנליטי $T(x,t)=10°C$,
כאשר גוף נמצא בתחום [5.0,5.4]

נקודה	ערך מתוכנה	ערך מדויק	שגיאה
(5.2300, 12953.5600)	10	10	0
(5.3800, 14569.8900)	10	10	0
(5.3154, 13786.2534)	10	10	0

רואים שחישוב מתבצעה עם דיוק מצויין עבור מספר איטרציות מינימלי

2) נציב $f(x)=x, g_1(t)=0°C, g_2(t)=0.4°C$, dTOL = 0.001
פטרון אנליטי $T(x,t)=x$, כאשר גוף נמצא בתחום [0.0,0.4]

נקודה	ערך מתוכנה	ערך מדויק	שגיאה
(0.2300, 12953.5600)	.230407	.230000	4.07×10^{-4}
(0.3800, 14569.8900)	.380742	.380000	7.42×10^{-4}
(0.3154, 13786.2534)	.316013	.315400	6.13×10^{-4}

רואים שחישוב מתבצעה עם דיוק מצויין עבור מספר פעולות קטן מאוד

3) נציב $f(x)=2\cos(0.1x)+5\sin(0.1x), g_1(t)=2e^{-0.01\alpha t}\ g_2(t)=e^{-0.01\alpha t}(2\cos(0.04)+5\sin(0.04))$,
פתרון אנליטי מהצורה $T(x,t)=e^{-0.01\alpha t}(2\cos(0.1x)+5\sin(0.1x))$, dTOL = 0.001,
כאשר גוף נמצא בתחום [0.0,0.4]

נקודה	ערך מתוכנה	ערך מדויק	שגיאה
(0.2300, 12953.5600)	2.104690	2.103165	1.53×10^{-3}
(0.3800, 14569.8900)	2.176090	2.175365	7.25×10^{-4}
(0.3154, 13786.2534)	2.145110	2.144419	6.91×10^{-4}
(0.3895, 14854.4700)	2.180500	2.179753	7.47×10^{-4}

רואים שחישוב מתבצעה עם דיוק מצויין עבור מספר פעולות קטן מאוד

סימולציה של מעבר חום במוט עם פרמטרים הבאים:

חומר של מוט: **צינק**

גבולות של מוט: [5.0,5.4]

קואפיציאנטים: $\rho = 7.133\left[\frac{gr}{cm^3}\right], c = .3831\left[\frac{Joul}{kg - °C}\right], k = 1.13\left[\frac{Joul}{cm - °C - \sec}\right]$

Left Boundary Condition - $10\tanh(t)$

Right Boundary Condition - $20\sin(5t)$

Initial Condition - $40\tan(.01x)$

קלט של תוכנית:

Enter left bound of rod, [m], **a = 5.0**
Enter right bound of rod, [m], **b = 5.4**
Enter tolerance of culculation, [Cel], **TOL = 0.01**
Enter cpecific heat, [Joul/kg-kelvin], **C = 0.3831**
Enter thermal conductivity, [Joul/cm-kelvin-sec], **K = 1.13**
Enter consistence of stuff, [gr/cm^3], **R = 7.133**
What are free memory space [GByte], **freeMemory = 1.0**

דיוק: **0.01**

נקודה	ערך מתוכנה
(0.2300, 12953.5600)	.5942360
(0.3800, 14569.8900)	-.0954487
(0.3154, 13786.2534)	.2212840
(0.3895, 14854.4700)	.0315553

דיוק: **0.001**

נקודה	ערך מתוכנה
(0.2300, 12953.5600)	.5962200
(0.3800, 14569.8900)	-.0286842
(0.3154, 13786.2534)	.2237180
(0.3895, 14854.4700)	.8558350

ספרי עזר

1) מדריך במתמטיקה גבוהה של ויגודסקי.
2) *G.D.Smith, Numerical Solution of Partial Differential Equations*