אנליזה נומרית / תרגיל בית 1#

אריאל סטולרמן

(1)

א. להלן הוכחה כי $x=\sum_{k=0}^{n}\frac{k}{n}B_n^k(x)$ עבור $B_k^n(x)=\binom{n}{k}x^k(1-x)^{n-k}$ (בסיס פולינומי של ברנשטיין): -

ב.

עבור $f(x)=e^x$ מעל $[-1,1]$, לפי משפט הקירוב של Weierstrass נקבל כי הפולינום $p_n(x)=\sum_{k=0}^{n}f\left(\frac{k}{n}\right)B_k^n(x)=\sum_{k=0}^{n}e^{\frac{k}{n}}\binom{n}{k}x^k(1-x)^{n-k}$ הוא קירוב עבור f עם שגיאה קטנה מ-$\varepsilon=0.01$ עבור $n(\varepsilon)$ כלשהו. נמצא n מינימלי המקיים זאת:

תחילה, העתקת הפונקציה אל מעל קטע $[0,1]$ תתן את הפונקציה $g(x)=e^{2x-1}$. כמובן ש-g רציפה מעל $[0,1]$. מעלת הפולינום עבור g מעל קטע $[0,1]$ תהיה אותה מעלה כמו עבור f מעל קטע $[-1,1]$. נמצא את מעלת הפולינום המקיימת $n(\varepsilon)\geq\frac{M}{\delta^2\varepsilon}$ כאשר $|g(x)|\leq M$ עבור $x\in[0,1]$ ולכל $x_1,x_2\in[0,1]$ מתקיים $|x_1-x_2|\leq\delta$: ניתן לבחור את $M=e$ כיוון ש-$\max_{x\in[0,1]}g(x)=e^{2-1}=e$, ואת $\delta=1$ שהוא המינימלי המקיים את התנאי לכל $x_1,x_2\in[0,1]$. לפיכך נקבל:

$$n(0.01)\geq\frac{e}{1\cdot 0.01}\cong\frac{2.718}{0.01}=271.8$$

ומכאן שאם נבחר את מעלת הפולינום להיות $n=272$ (או יותר) נקבל כי $|e^x-p_n(x)|\leq 0.01$.

עבור $f(x)=|x|$ מעל $[-1,1]$, לפי משפט הקירוב נקבל כי פולינום הקירוב הוא: $p_n(x)=\sum_{k=0}^{n}f\left(\frac{k}{n}\right)B_k^n(x)=\sum_{k=0}^{n}\left|\frac{k}{n}\right|\binom{n}{k}x^k(1-x)^{n-k}$. כמו קודם, נעתיק את הפונקציה לקטע $[0,1]$ ונקבל את הפונקציה $g(x)=|2x-1|$, ונבחר את הפרמטרים הבאים: $M=1$ שכן $|g(x)|\leq 1$ לכל $x\in[0,1]$; $\delta=1$, בדיוק כמו קודם. נקבל:

$$n(0.01)\geq\frac{1}{0.01}=100$$

ומכאן שאם נבחר את מעלת הפולינום להיות $n=100$ (או יותר) נקבל כי $\big||x|-p_n(x)\big|\leq 0.01$.

(2)

א. להלן חישוב הסכום $S_N=\sum_{n=0}^{N}\frac{x^n}{n!}$ לכל $N\in\{10,20,\dots,100\}$ עבור $x=10$:

func.m:
```
function[result] = func(x,N)
res = 1
tmp = 1
for n = 1:1:N
    tmp = tmp * (x/n)
    res = res + tmp
end
result = res;
```

Console:
```
format long

N = 10:10:100
N =
    10    20    30    40    50    60    70    80    90   100
x = 10
x =
    10
for i = 1:10
S_N(i) = func(x,N(i))
end
S_N

...

S_N =
  1.0e+004 *
```

```
   1.284230511463845   2.199148202566507   2.202646403625892
   2.202646579480280   2.202646579480672   2.202646579480672
   2.202646579480672   2.202646579480672   2.202646579480672
   2.202646579480672
```

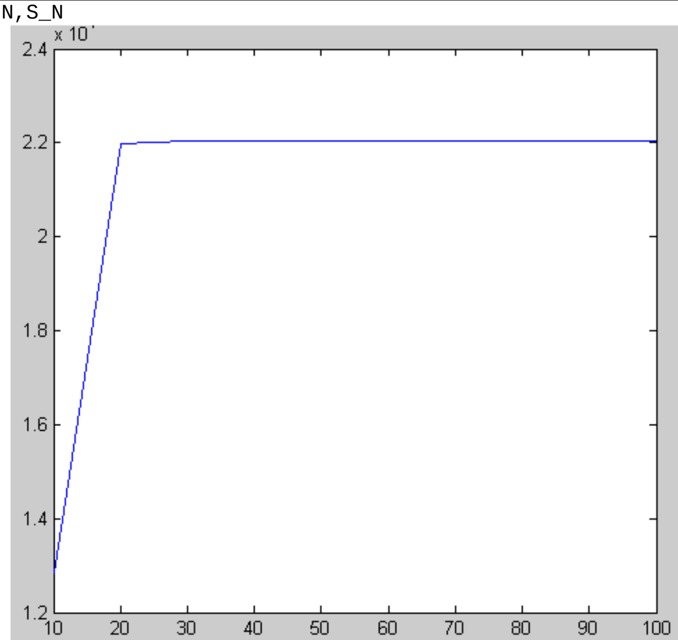
ב. להלן חישוב השגיאה היחסית $R = \left|\frac{e^x - S_N}{e^x}\right|$:

```
R = abs((exp(x)-S_N)/exp(x))
R =
   0.416960249807014   0.001588260661858   0.000000079837947
   0.000000000000178   0.000000000000000   0.000000000000000
   0.000000000000000   0.000000000000000   0.000000000000000
   0.000000000000000
```

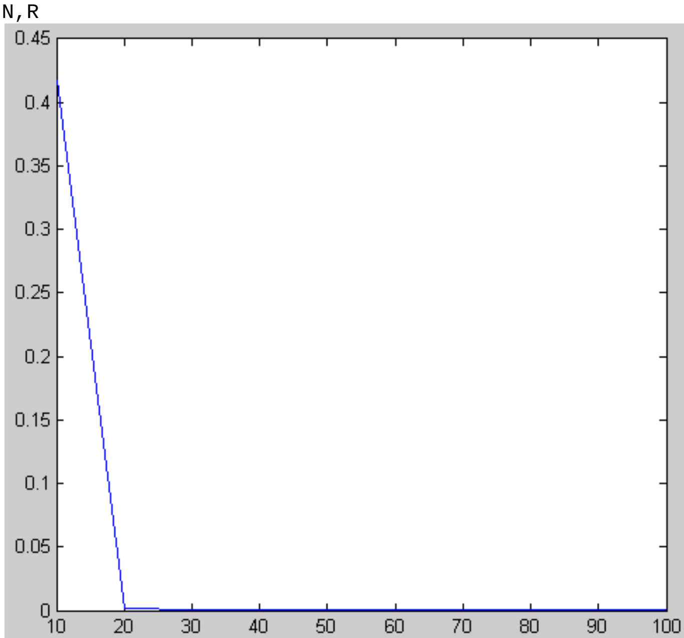
ג. להלן אותו חישוב עבור $x = -10$:

```
x = -10
x =
   -10
for i = 1:10
S_N2(i) = func(x,N(i))
end

...

S_N2
S_N2 =
  1.0e+003 *
   1.342587301587302   0.013396865995696   0.000000970341580
   0.000000045402342   0.000000045399930   0.000000045399930
   0.000000045399930   0.000000045399930   0.000000045399930
   0.000000045399930

R2 = abs((exp(x)-S_N2)/exp(x))
R2 =
  1.0e+007 *
   2.957245227495455   0.029508461061181   0.000002037319562
   0.000000000005313   0.000000000000000   0.000000000000000
   0.000000000000000   0.000000000000000   0.000000000000000
   0.000000000000000

plot(N,S_N)
plot(N,S_N2)
plot(N,R)
plot(N,R2)
```

N,S_N

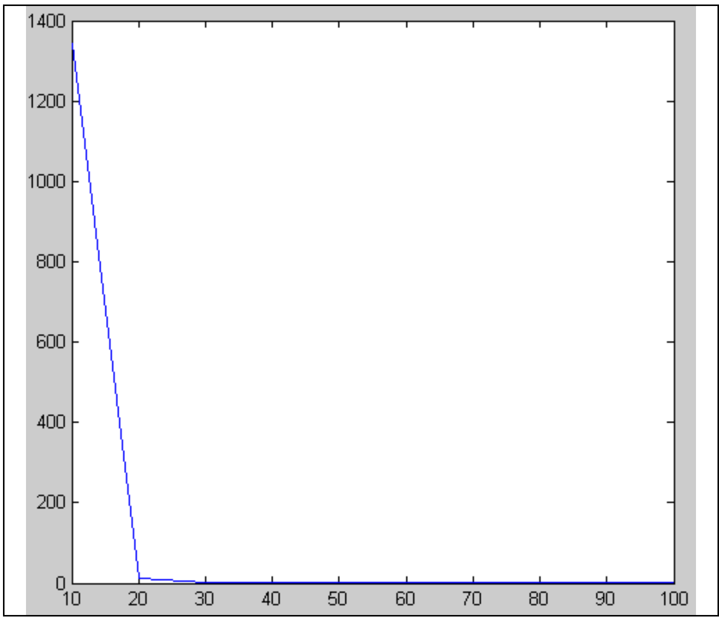


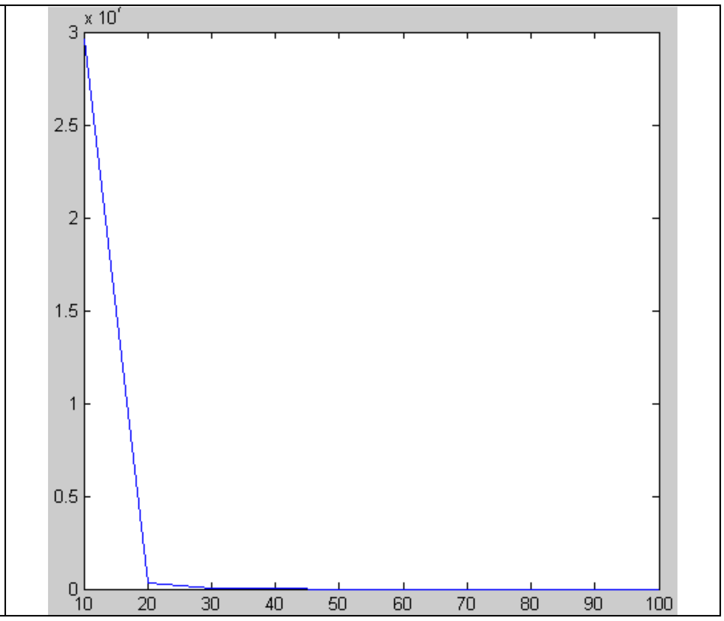
N,R



N,S_N2

N,R2





ד. בשני המקרים הסדרות מתכנסות.

ה. בשני המקרים הסדרות מתכנסות לערך הנכון.

(3)

א. להלן הקוד :

func_q3.m:
```
function[result] = func_q3(k)
n = 10.^k
result = (1 + (1/n)).^n;
```

Console:
```
format long

for k=1:30
res(k) = func_q3(k)
end
res =
   2.593742460100002   2.704813829421529   2.716923932235594   2.718145926824926
   2.718268237192298   2.718280469095753   2.718281694132082   2.718281798347358
   2.718282052011560   2.718282053234788   2.718282053357110   2.718523496037238
   2.716110034086901   2.716110034087023   3.035035206549262   1.000000000000000
   1.000000000000000   1.000000000000000   1.000000000000000   1.000000000000000
   1.000000000000000   1.000000000000000   1.000000000000000   1.000000000000000
   1.000000000000000   1.000000000000000   1.000000000000000   1.000000000000000
   1.000000000000000   1.000000000000000

abs_err = exp(1) - res
abs_err =
   0.124539368359043   0.013467999037517   0.001357896223452   0.000135901634120
   0.000013591266748   0.000001359363292   0.000000134326964   0.000000030111688
  -0.000000223552515  -0.000000224775742  -0.000000224898065  -0.000241667578192
   0.002171794372145   0.002171794372023  -0.316753378090216   1.718281828459046
   1.718281828459046   1.718281828459046   1.718281828459046   1.718281828459046
   1.718281828459046   1.718281828459046   1.718281828459046   1.718281828459046
   1.718281828459046   1.718281828459046   1.718281828459046   1.718281828459046
   1.718281828459046   1.718281828459046

rel_err = abs((exp(1) - res)/exp(1))
rel_err =
   0.045815473235769   0.004954599959619   0.000499542103852   0.000049995417214
   0.000004999947616   0.000000500081808   0.000000049416128   0.000000011077471
   0.000000082240374   0.000000082690374   0.000000082735374   0.000088904533615
   0.000798958499964   0.000798958499919   0.116527055720995   0.632120558828558
   0.632120558828558   0.632120558828558   0.632120558828558   0.632120558828558
   0.632120558828558   0.632120558828558   0.632120558828558   0.632120558828558
   0.632120558828558   0.632120558828558   0.632120558828558   0.632120558828558
   0.632120558828558   0.632120558828558
```

ב.

השגיאה היחסית תהיה מאוד גדולה בשלב כלשהו כיוון שבחישוב $\frac{1}{n}$ עבור n גדול מאוד הערך יחושב ל-0 (כאשר המספר קטן מ-10^{-15}, שכן ניתן לייצג עד 15 ספרות מימין לנקודה). כתוצאה מכך ערך הפונקציה $\left(1+\frac{1}{n}\right)^n$ יחושב ל-1, ואז השגיאה היחסית תהיה $\left|\frac{e-1}{e}\right|$, וזהו ערך גדול מאוד ביחס לתוצאות עם n גדול יותר.

(4)

להלן קוד הבדיקה. בסופו נראה שכבר עבור $n = 3$ מתקבלת תוצאה שונה מ-1:

```
format long
a = 1/3
a =
   0.333333333333333
b = 3
b =
     3
n = 1
n =
     1

while ((a.^n)*(b.^n)) == 1
n = n + 1
end
n =
     2
n =
     3
```