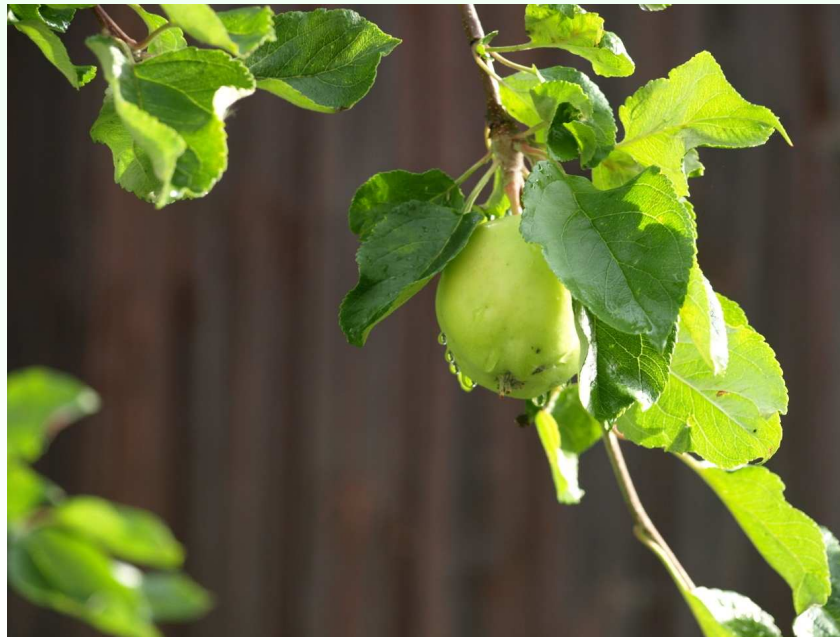


Ohjelmointiesimerkkejä TI-laskimille

Hannu Lehto
Lahden Lyseon lukio



N-kertoman (n!) laskeminen

- N-kertoman (n!) laskeminen

- 2-asteen yhtälön ratkaiseminen

- π :n likiarvo iteroimalla

- Haarukointimenetelmä

- Newtonin menetelmä

- Puolisuunnikassääntö

- Simpsonin sääntö

Seuraava ohjelma ratkaisee n-kertoman, kun $n \in \mathbb{Z}$

N-kertoman (n!) laskeminen

- N-kertoman (n!) laskeminen

- 2-asteen yhtälön ratkaiseminen

- π :n likiarvo iteroimalla

- Haarukointimenetelmä

- Newtonin menetelmä

- Puolisuunnikassääntö

- Simpsonin sääntö

Seuraava ohjelma ratkaisee n-kertoman, kun $n \in \mathbb{Z}$

Prompt N

1 \rightarrow T

1 \rightarrow M

While $M \leq N$

T * M \rightarrow T

M + 1 \rightarrow M

End

ClrHome

Output(1,1,N)

Output(1,4,"kertoma on")

Output(2,8,T)

2-asteen yhtälön ratkaiseminen

- N-kertoman (n!) laskeminen
- **2-asteen yhtälön ratkaiseminen**
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikassääntö
- Simpsonin sääntö

Seuraava ohjelma ratkaisee muotoa $ax^2 + bx + c = 0$ olevan toisen asteen yhtälön.

2-asteen yhtälön ratkaiseminen

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikassääntö
- Simpsonin sääntö

Seuraava ohjelma ratkaisee muotoa $ax^2 + bx + c = 0$ olevan toisen asteen yhtälön.

```
Prompt A,B,C
 $B^2 - 4 * A * C \rightarrow D$ 
If D<0
Then
Disp "EI RATKAISUJA"
Else
 $(-B + \sqrt{(D)}/(2 * A) \rightarrow X$ 
 $(-B - \sqrt{(D)}/(2 * A) \rightarrow Y$ 
Disp X,Y
End
```

π :n likiarvo iteroimalla

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikassääntö
- Simpsonin sääntö

Likiarvo lasketaan yksikköympyrän sisään piirretyn säännöllisen n-kulmion piiristä: $2\pi \cdot 1 \approx na_n$, missä a_n on n-kulmion sivun pituus.

π :n likiarvo iteroimalla

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikkasääntö
- Simpsonin sääntö

Likiarvo lasketaan yksikköympyrän sisään piirretyn säännöllisen n-kulmion piiristä: $2\pi \cdot 1 \approx na_n$, missä a_n on n-kulmion sivun pituus.

Iteroitava lauseke on $a_{2n} = \frac{a_n}{\sqrt{2 + \sqrt{4 - a_n^2}}}$, joka ilmoittaa, miten

säännöllisen 2n-kulmion sivun pituus a_{2n} saadaan säännöllisen n-kulmion sivun pituudesta a_n .

π :n likiarvo iteroimalla

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikkasääntö
- Simpsonin sääntö

Likiarvo lasketaan yksikköympyrän sisään piirretyn säännöllisen n-kulmion piiristä: $2\pi \cdot 1 \approx na_n$, missä a_n on n-kulmion sivun pituus.

Iteroitava lauseke on $a_{2n} = \frac{a_n}{\sqrt{2 + \sqrt{4 - a_n^2}}}$, joka ilmoittaa, miten

säännöllisen 2n-kulmion sivun pituus a_{2n} saadaan säännöllisen n-kulmion sivun pituudesta a_n . Ohjelmassa tehdään 20 iterointia ja liikkeelle lähdetään säännöllisestä 6-kulmiosta, jonka sivun pituus on ympyrän säde 1.

π :n likiarvo iteroimalla

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikkasääntö
- Simpsonin sääntö

Likiarvo lasketaan yksikköympyrän sisään piirretyn säännöllisen n-kulmion piiristä: $2\pi \cdot 1 \approx na_n$, missä a_n on n-kulmion sivun pituus.

Iteroitava lauseke on $a_{2n} = \frac{a_n}{\sqrt{2 + \sqrt{4 - a_n^2}}}$, joka ilmoittaa, miten

säännöllisen 2n-kulmion sivun pituus a_{2n} saadaan säännöllisen n-kulmion sivun pituudesta a_n . Ohjelmassa tehdään 20 iterointia ja liikkeelle lähdetään säännöllisestä 6-kulmiosta, jonka sivun pituus on ympyrän säde 1.

```
ClrHome
1 → A
6 → N
For(I,1,20)
A/√(2 + √(4 - A²)) → A
2 * N → N
N * A/2 → P
Disp P
End
```

Haarukointimenetelmä

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla

Haarukointimenetelmä

- Newtonin menetelmä
- Puolisuunnikkasääntö
- Simpsonin sääntö

Ohjelma etsii funktion tunnetulla välillä sijaitsevan nollakohdan likiarvon. Funktio on ennen ohjelman suorittamista tallennettava funktiomuuttujaan Y_1

```
ClrHome
Disp "Tarkkuus"
Prompt T
Disp "Vali"
Prompt A,B
0 → C
Repeat abs(A - B) < T or Y1(C) = 0
(A + B)/2 → C
If Y1(A) * Y1(C) < 0
C → B
If Y1(A) * Y1(C) > 0
C → A
End
ClrHome
Disp "ratkaisu"
Disp C
```

Newtonin menetelmä

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- **Newtonin menetelmä**
- Puolisuunnikassääntö
- Simpsonin sääntö

Ohjelma etsii funktion nollakohdan likiarvon. Funktio on ennen ohjelman suorittamista tallennettava funktiomuuttujaan Y_1 .

Newtonin menetelmä

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- **Newtonin menetelmä**
- Puolisuunnikassääntö
- Simpsonin sääntö

Ohjelma etsii funktion nollakohdan likiarvon. Funktio on ennen ohjelman suorittamista tallennettava funktiomuuttujaan Y_1 . Nollakohdan likiarvon antava kaava $x - \frac{f(x)}{f'(x)}$ on talletettava funktiomuuttujaan Y_2 .

Newtonin menetelmä

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- **Newtonin menetelmä**
- Puolisuunnikassääntö
- Simpsonin sääntö

Ohjelma etsii funktion nollakohdan likiarvon. Funktio on ennen ohjelman suorittamista tallennettava funktiomuuttujaan Y_1 .

Nollakohdan likiarvon antava kaava $x - \frac{f(x)}{f'(x)}$ on talletettava funktiomuuttujaan Y_2 .

Ohjelman on puutteellinen: se voi juuttua silmukkaan tai keskeytyä nolalla jakoon.

Newtonin menetelmä

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- **Newtonin menetelmä**
- Puolisuunnikassääntö
- Simpsonin sääntö

Ohjelma etsii funktion nollakohdan likiarvon. Funktio on ennen ohjelman suorittamista tallennettava funktiomuuttujaan Y_1 .

Nollakohdan likiarvon antava kaava $x - \frac{f(x)}{f'(x)}$ on talletettava funktiomuuttujaan Y_2 .

Ohjelman on puutteellinen: se voi juuttua silmukkaan tai keskeytyä nolalla jakoon.

```
ClrHome
Disp "Nollakohta"
Disp "Newtonin"
Disp "menetelmalla"
Disp "Alkuarvaus"
Prompt X
While  $Y_1(X) > 1 * 10^{(-8)}$ 
 $Y_2(X) \rightarrow X$ 
Disp X
End
```

Puolisuunnikassääntö

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikassääntö
- Simpsonin sääntö

Ohjelma laskee likiarvon integraalille $I = \int_a^b f(x)dx$ käyttäen kaavaa $I \approx \frac{h}{2} (y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n)$. Funktio on ennen ohjelman suoritusta talletettava funktiomuuttujaan Y_1 .

```
ClrHome
Disp "Rajat?"
Prompt A,B
Disp "Valit?"
Prompt N
1  $\rightarrow$  L
 $Y_1(A) + Y_1(B) \rightarrow S$ 
 $(B - A)/N \rightarrow H$ 
While  $L < N$ 
 $S + 2 * Y_1(A + L * H) \rightarrow S$ 
 $L + 1 \rightarrow L$ 
End
 $(H/2) * S \rightarrow I$ 
Disp I
```

Simpsonin sääntö

- N-kertoman (n!) laskeminen
- 2-asteen yhtälön ratkaiseminen
- π :n likiarvo iteroimalla
- Haarukointimenetelmä
- Newtonin menetelmä
- Puolisuunnikassääntö
- **Simpsonin sääntö**

Ohjelma laskee likiarvon integraalille $I = \int_a^b f(x)dx$ käyttäen kaavaa $I \approx \frac{h}{3} [y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + 4y_{n-1} + y_n]$. Funktio on ennen ohjelman suoritusta talletettava funktiomuuttujaan Y_1 .

```
ClrHome
Disp "Rajat?"
Prompt A,B
Repeat  $N/2 = \text{int}(N/2)$ 
Disp "Valit(parillinen)?"
Prompt N
End
 $1 \rightarrow L$ 
 $Y_1(A) + Y_1(B) \rightarrow S$ 
 $(B - A)/N \rightarrow H$ 
```

```
While  $L < N$ 
If  $L/2 = \text{int}(L/2)$ 
Then
 $S + 2 * Y_1(A + L * H) \rightarrow S$ 
Else
 $S + 4 * Y_1(A + L * H) \rightarrow S$ 
End
 $L + 1 \rightarrow L$ 
End
 $(H/3) * S \rightarrow I$ 
Disp "Integraali"
Disp I
```