Ensimmäisen kertaluvun yhtälön numeerinen ratkaiseminen

Eulerin menetelmä alkaurvoprobleeman $y' = f(x, y)$, $y(x_0) = y_0$ ratkaisemiseksi voidaan ohjelmoida Mathematicalle `euler`-nimiseksi funktioksi seuraavalla tavalla (solu on ajettava, jotta määritelmä tulee voimaan):

```
euler[f_, x0_, y0_, h_, xend_] := Module[{X, Y, kend},
  X[0] = x0;
  X[k_] := X[k] = X[k - 1] + h;
  Y[0] = y0;
  Y[k_] := Y[k] = Y[k - 1] + h f[X[k - 1], Y[k - 1]];
  kend = Round[(xend - x0) / h];
  Table[{X[k], Y[k]}, {k, 0, kend}]]
```

```
General::spell1 :
 Possible spelling error: new symbol name "kend" is similar to existing symbol "xend". More…
```

Funktion ensimmäinen argumentti on differentiaaliyhtälön oikean puolen funktio f Mathematican funktioksi määriteltynä, kaksi seuraavaa määrittävät alkuehdon, neljäs on askelpituus. Laskenta tapahtuu välillä `[x0,xend]`.

Funktion runko muodostuu `Module`-komennosta, jonka sisällä on ensin aaltosuluissa lueteltu paikalliset muuttujat, jotka eivät näy funktion ulkopuolelle. Seuraavilla neljällä rivillä on kaavat, joilla paikallisten funktioiden `X` ja `Y` arvoiksi määritellään numeerisen ratkaisun hilapisteiden koordinaatit. Toiseksi viimeisellä rivillä lasketaan tarvittavien askeleiden lukumäärä. Viimeisellä rivillä tapahtuu varsinainen laskenta: Hilapisteiden koordinaatit todella lasketaan ja sijoitetaan taulukkoon, joka tulee `Module`-kutsun arvoksi ja myös `euler`-funktion arvoksi.

Rivit

`X[k_]:=X[k]=X[k-1]+h;` ja
`Y[k_]:=Y[k]=Y[k-1]+h f[X[k-1],Y[k-1]];`

ovat hieman erikoiset: niissä näyttäisi olevan kaksi sijoituskäskyä tehtynä symboleilla `:=` ja `=` . Kyseessä on rerkursiivinen määrittely (symboli `:=`), jossa esimerkiksi arvon `X[k]` laskeminen palautetaan arvon `X[k-1]` laskemiseen, tämä edelleen arvon `X[k-2]` laskemiseen jne., kunnes päädytään arvoon `X[0]`, joka on aiemmin asetettu. Jotta jokaista arvoa — esimerkiksi seuraavaa arvoa `X[k+1]` — laskettaessa ei uudelleen tarvitsisi laskea kaikkia arvoja alkuarvoon `X[0]` saakka, laskettu arvo `X[k]` talletetaan funktion `X` arvoksi operaattorilla `=` . (Vrt. arvon antamiseen komennossa `X[0]=x0;` .)

`euler`-funktion käyttö tapahtuu antamalla sille tarvittavat argumentit, jolloin funktio palauttaa taulukon, jossa on hilapisteiden koordinaatit. Argumentit voidaan tallettaa muillekin nimille kuin funktion määrittelyssä käytetyille tai antaa myös suoraan numeroina (esimerkiksi `askel= 0.1, euler[Function[{x,y},2 x y],0,1,askel,2]`).

```
f[x_, y_] := 2 x y
```

```
h = 0.1
```

```
0.1
```

```
x0 = 0
```

```
0
```

```
y0 = 1
```

```
1
```

```
xend = 2
```

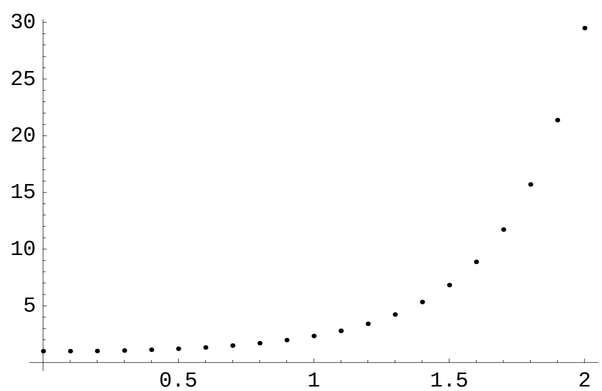
```
2
```

```
eulerdata = euler[f, x0, y0, h, xend]
```

```
{{0, 1}, {0.1, 1}, {0.2, 1.02}, {0.3, 1.0608}, {0.4, 1.12445}, {0.5, 1.2144},
 {0.6, 1.33584}, {0.7, 1.49615}, {0.8, 1.70561}, {0.9, 1.9785}, {1., 2.33463},
 {1.1, 2.80156}, {1.2, 3.4179}, {1.3, 4.2382}, {1.4, 5.34013}, {1.5, 6.83537},
 {1.6, 8.88598}, {1.7, 11.7295}, {1.8, 15.7175}, {1.9, 21.3758}, {2., 29.4986}}
```

Datan perusteella voidaan piirtää kuva ratkaisua approksimoivista pisteistä.

```
eulerkuva = ListPlot[eulerdata]
```




```
- Graphics -
```

Samaan tapaan voidaan ohjelmoida parannettu Eulerin menetelmä, Rungen – Kuttan menetelmä ja Adamsin – Bashforthin menetelmä:

```
pareuler[f_, x0_, y0_, h_, xend_] := Module[{X, Y, kend},
  X[0] = x0;
  X[k_] := X[k] = X[k - 1] + h;
  Y[0] = y0;
  Y[k_] :=
   Y[k] = Y[k - 1] + h / 2 (f[X[k - 1], Y[k - 1]] + f[X[k], Y[k - 1] + h f[X[k - 1], Y[k - 1]]]);
  kend = Round[(xend - x0) / h];
  Table[{X[k], Y[k]}, {k, 0, kend}]]
```

```
rungekutta[f_, x0_, y0_, h_, xend_] := Module[{X, Y, kend},
  X[0] = x0;
  X[k_] := X[k] = X[k - 1] + h;
  Y[0] = y0;
  Y[k_] := Y[k] = Y[k - 1] + 1 / 6 Module[{k1, k2, k3, k4}, k1 = h f[X[k - 1], Y[k - 1]];
       k2 = h f[X[k - 1] + h / 2, Y[k - 1] + k1 / 2];
       k3 = h f[X[k - 1] + h / 2, Y[k - 1] + k2 / 2];
       k4 = h f[X[k - 1] + h, Y[k - 1] + k3];
       k1 + 2 k2 + 2 k3 + k4];
  kend = Round[(xend - x0) / h];
  Table[{X[k], Y[k]}, {k, 0, kend}]]
```

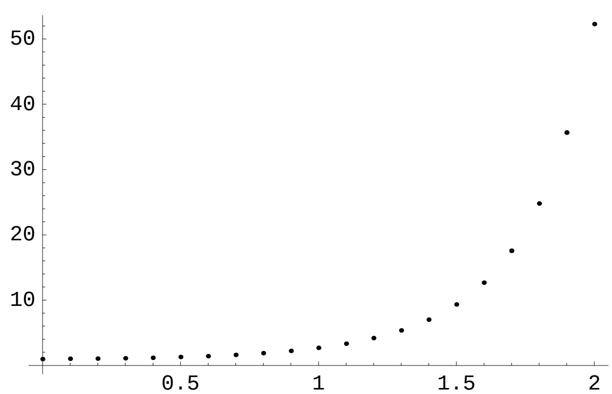
```
adamsbashforth[f_, x0_, y0_, h_, xend_] := Module[{X, Y, kend},
  X[0] = x0;
  X[k_] := X[k] = X[k - 1] + h;
  Y[0] = y0;
  Y[k_ /; k <= 3] :=
   Y[k] = Y[k - 1] + 1 / 6 Module[{k1, k2, k3, k4}, k1 = h f[X[k - 1], Y[k - 1]];
       k2 = h f[X[k - 1] + h / 2, Y[k - 1] + k1 / 2];
       k3 = h f[X[k - 1] + h / 2, Y[k - 1] + k2 / 2];
       k4 = h f[X[k - 1] + h, Y[k - 1] + k3];
       k1 + 2 k2 + 2 k3 + k4];
  Y[k_ /; k > 3] := Y[k] = Y[k - 1] + h / 24 (55 f[X[k - 1], Y[k - 1]] -
        59 f[X[k - 2], Y[k - 2]] + 37 f[X[k - 3], Y[k - 3]] - 9 f[X[k - 4], Y[k - 4]]);
  kend = Round[(xend - x0) / h];
  Table[{X[k], Y[k]}, {k, 0, kend}]]
```

Näiden avulla muodostetut ratkaisut ja vastaavat kuvat:

```
pareulerdata = pareuler[f, x0, y0, h, xend]
```

```
{{0, 1}, {0.1, 1.01}, {0.2, 1.0407}, {0.3, 1.09399}, {0.4, 1.17319}, {0.5, 1.28347},
 {0.6, 1.43236}, {0.7, 1.63059}, {0.8, 1.89345}, {0.9, 2.2426}, {1., 2.70906},
 {1.1, 3.33756}, {1.2, 4.19331}, {1.3, 5.37247}, {1.4, 7.01859}, {1.5, 9.34876},
 {1.6, 12.6956}, {1.7, 17.5758}, {1.8, 24.803}, {1.9, 35.6766}, {2., 52.3019}}
```

```
pareulerkuva = ListPlot[pareulerdata]
```

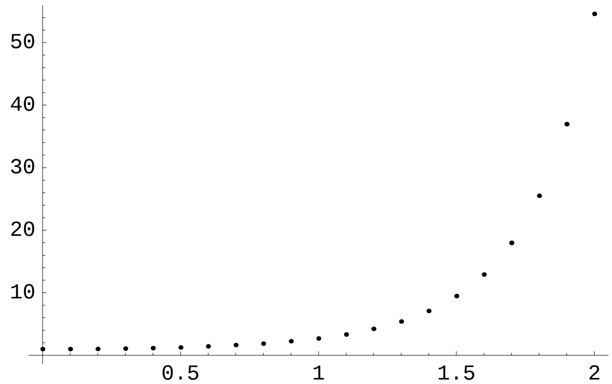


- Graphics -

```
rungekuttadata = rungekutta[f, x0, y0, h, xend]
```

```
{{0, 1}, {0.1, 1.01005}, {0.2, 1.04081}, {0.3, 1.09417}, {0.4, 1.17351}, {0.5, 1.28403},
 {0.6, 1.43333}, {0.7, 1.63232}, {0.8, 1.89648}, {0.9, 2.2479}, {1., 2.71827},
 {1.1, 3.35346}, {1.2, 4.22065}, {1.3, 5.41938}, {1.4, 7.09912}, {1.5, 9.48734},
 {1.6, 12.935}, {1.7, 17.9918}, {1.8, 25.5307}, {1.9, 36.9601}, {2., 54.5863}}
```

```
rungekuttakuva = ListPlot[rungekuttadata]
```

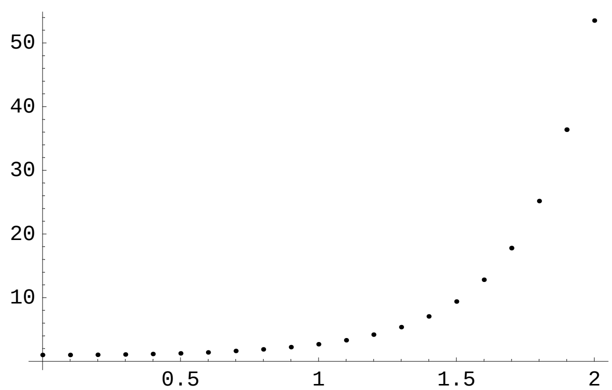



```
- Graphics -
```

```
adamsbashforthdata = adamsbashforth[f, x0, y0, h, xend]
```

```
{{0, 1}, {0.1, 1.01005}, {0.2, 1.04081}, {0.3, 1.09417}, {0.4, 1.17342}, {0.5, 1.28376},
 {0.6, 1.43279}, {0.7, 1.63131}, {0.8, 1.89473}, {0.9, 2.24495}, {1., 2.71334},
 {1.1, 3.34529}, {1.2, 4.20711}, {1.3, 5.39685}, {1.4, 7.06139}, {1.5, 9.42361},
 {1.6, 12.8263}, {1.7, 17.8044}, {1.8, 25.2039}, {1.9, 36.3831}, {2., 53.5546}}
```

```
adamsbashforthkuva = ListPlot[adamsbashforthdata]
```



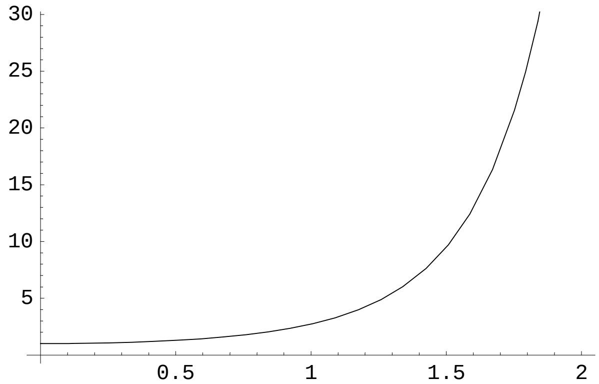

```
- Graphics -
```

Esimerkkinä oleva alkuarvoprobleema on myös ratkaistavissa algebrallisesti, jolloin voidaan verrata eri menetelmien tarkkuutta toisiinsa ja myös tarkkaan ratkaisuun:

```
tarkkaratkaisu = y /. First[DSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y, x]]
```

Function$\left[\{x\}, e^{x^2}\right]$

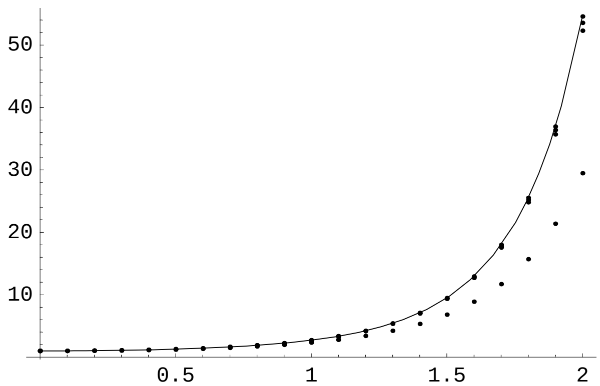
```
tarkkakuva = Plot[tarkkaratkaisu[x], {x, 0, xend}]
```




```
- Graphics -
```

Kaikki ratkaisut samassa kuvassa:

```
kuva = Show[eulerkuva, pareulerkuva,
  rungekuttakuva, adamsbashforthkuva, tarkkakuva, PlotRange -> All]
```




```
- Graphics -
```

Lopuksi eri menetelmillä saadut arvot tarkasteluvälin loppupisteessä:

```
vertailu = {Last[eulerdata][[2]], Last[pareulerdata][[2]],
  Last[rungekuttadata][[2]], Last[adamsbashforthdata][[2]], tarkkaratkaisu[xend] // N}
```

```
{29.4986, 52.3019, 54.5863, 53.5546, 54.5982}
```

Edellä olevia syötteitä voi muuntaa ja tämän jälkeen ajaa koko muistikirjan yhdellä kerralla (valikko *Kernel / Evaluation*).

Linkit

ensimmäisen kertaluvun yhtälön numeerinen ratkaiseminen
Eulerin menetelmä
parannettu Eulerin menetelmä
Rungen–Kuttan menetelmä
Adamsin–Bashforthin menetelmä

SKK 30.04.2001