Ensimmäisen kertaluvun yhtälö

Eulerin menetelmä alkuarvoprobleeman $y' = f(x,y)$, $y(x_0) = y_0$, ratkaisemiseksi voidaan ohjelmoida *Maple*lle **`eulernum`**-nimiseksi proseduuriksi seuraavalla tavalla (solu on ajettava, jotta määritelmä tulee voimaan):

```
> eulernum:=proc(f, x0, y0, h, xend)
    local kend, x, y, i;
    kend:= round((xend-x0)/h);
    x[0]:= x0;
    y[0]:= y0;
    for i from 1 to kend do
      x[i]:= x[i-1]+h;
      y[i]:= y[i-1]+h*f(x[i-1], y[i-1]);
    od;
  return [seq([x[k], y[k]], k=0..kend)];
  end:
```

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

Proseduurin toisella rivillä lasketaan tarvittavien askelten lukumäärä **`kend`**. Kolmannella ja neljännellä rivillä määritellään alkuarvot. **`For`**-silmukassa lasketaan numeerisen ratkaisun hilapisteet Eulerin menetelmällä. Viimeisellä rivillä palautetaan vastaus listana koottuna *xy* -pareiksi.

Numeerinen ratkaiseminen tapahtuu antamalla tarvittavat argumentit **`eulernum`**-funktiolle, jolloin tulokseksi saadaan lista hilapisteiden koordinaateista. Argumentit voidaan tallettaa muillekin nimille kuin funktion määrittelyssä käytetyille tai antaa myös suoraan numeroina (esimerkiksi **`askel:= 0.1`**, **`eulernum(f,x0,y0,askel,xend)`** tai **`eulernum(unapply(2*x*y,x,y),0,1,0.1,2)`**.

```
> f:= unapply(2*x*y, x, y);
```

$$f := (x, y) \to 2\, y\, x$$

```
> h:= 0.1;
```

$$h := 0.1$$

```
> x0:= 0;
```

$$x0 := 0$$

```
> y0:= 1;
```

$$y0 := 1$$

```
> xend:= 2;
```

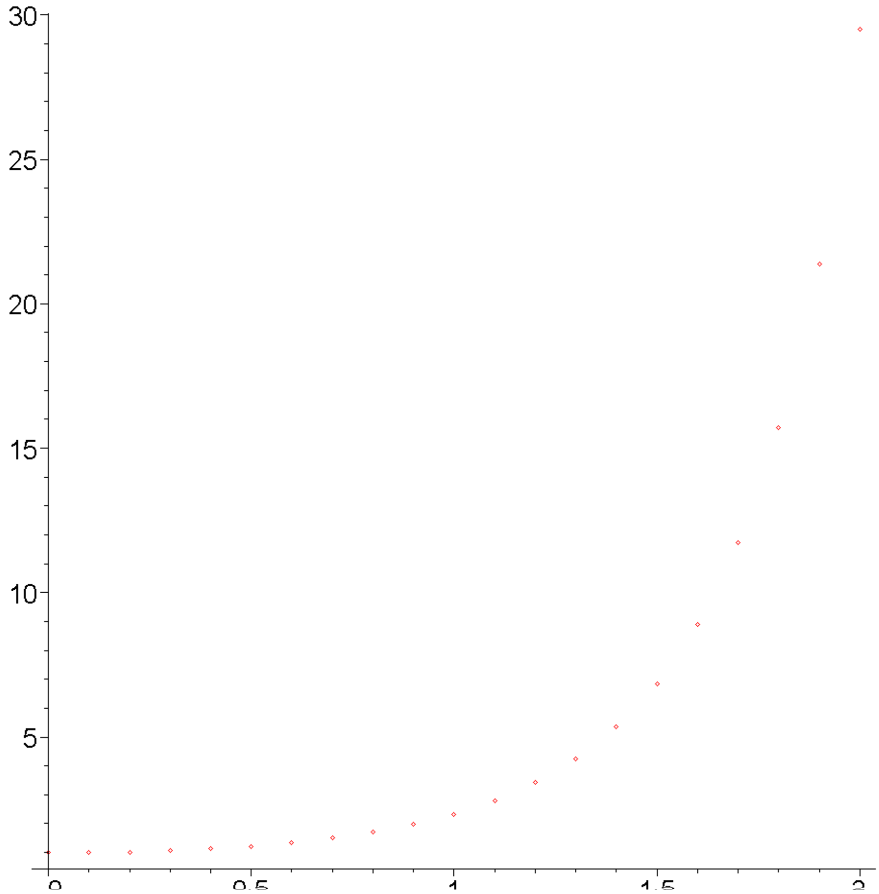
$$xend := 2$$

```
> eulerdata:= eulernum(f, x0, y0, h, xend);
```

eulerdata := [[0, 1], [0.1, 1.], [0.2, 1.02], [0.3, 1.0608], [0.4, 1.124448], [0.5, 1.21440384], [0.6, 1.335844224], [0.7, 1.496145531], [0.8, 1.705605905], [0.9, 1.978502850], [1.0, 2.334633363], [1.1, 2.801560036], [1.2, 3.417903244], [1.3, 4.238200023], [1.4, 5.340132029], [1.5, 6.835368997], [1.6, 8.885979697], [1.7, 11.72949320], [1.8, 15.71752089], [1.9, 21.37582841], [2.0, 29.49864321]]

Datan perusteella voidaan piirtää kuva ratkaisua approksimoivista pisteistä.

```
> eulerkuva:= plot(eulerdata, style=point):
  eulerkuva;
```



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

```
> pareuler:= proc(f, x0, y0, h, xend)
    local kend, x, y, i;
    kend:= round((xend-x0)/h);
    x[0]:= x0;
    y[0]:= y0;
    for i from 1 to kend do
      x[i]:= x[i-1]+h;
      y[i]:= y[i-1]+
        h/2*(f(x[i-1], y[i-1])+
          f(x[i], y[i-1]+h*f(x[i-1], y[i-1])));
```

```
    od;
  return [seq([x[k],y[k]], k=0..kend)];
  end:
```

```
> rungekutta:= proc(f, x0, y0, h, xend)
    local kend, x, y, i, k1, k2, k3, k4;
    kend:= round((xend-x0)/h);
    x[0]:= x0;
    y[0]:= y0;
    for i from 1 to kend do
      x[i]:= x[i-1]+h;
      k1:= h*f(x[i-1], y[i-1]);
      k2:= h*f(x[i-1]+h/2, y[i-1]+k1/2);
      k3:= h*f(x[i-1]+h/2, y[i-1]+k2/2);
      k4:= h*f(x[i-1]+h, y[i-1]+k3);
      y[i]:= y[i-1]+1/6*(k1+2*k2+2*k3+k4);
    od;
  return [seq([x[k], y[k]], k=0..kend)];
  end:
```

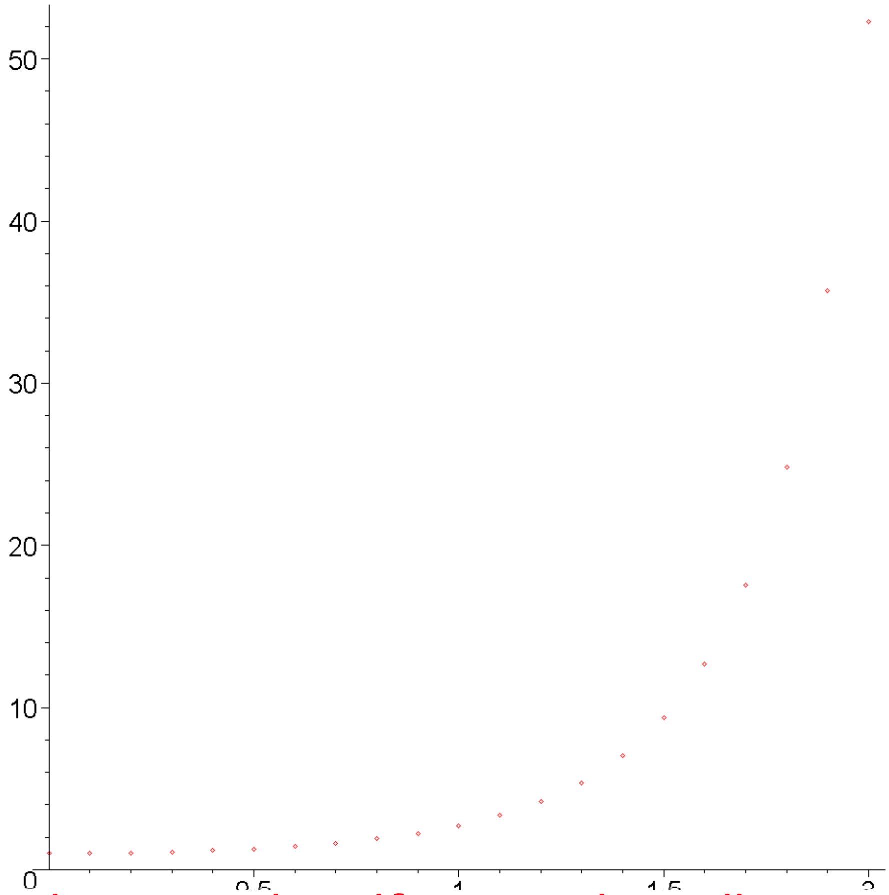
```
> adamsbashforth:= proc(f, x0, y0, h, xend)
    local kend, x, y, i, k1, k2, k3, k4;
    kend:= round((xend-x0)/h);
    x[0]:= x0;
    y[0]:= y0;
    for i from 1 to 3 do
      x[i]:= x[i-1]+h;
      k1:= h*f(x[i-1], y[i-1]);
      k2:= h*f(x[i-1]+h/2, y[i-1]+k1/2);
      k3:= h*f(x[i-1]+h/2, y[i-1]+k2/2);
      k4:= h*f(x[i-1]+h, y[i-1]+k3);
      y[i]:= y[i-1]+1/6*(k1+2*k2+2*k3+k4);
    od;
    for i from 4 to kend do
      x[i]:= x[i-1]+h;
      y[i]:= y[i-1]+h/24*(
        55*f(x[i-1], y[i-1])-
        59*f(x[i-2], y[i-2])+
        37*f(x[i-3], y[i-3])-
        9*f(x[i-4], y[i-4]))
    od;
  return [seq([x[k], y[k]], k=0..kend)];
  end:
```

Näiden avulla muodostetut ratkaisut ja vastaavat kuvat:

```
> pareulerdata:= pareuler(f, x0, y0, h, xend);
```

pareulerdata := [[0, 1], [0.1, 1.010000000], [0.2, 1.040704000], [0.3, 1.093988045], [0.4, 1.173192779], [0.5, 1.283472900], [0.6, 1.432355756], [0.7, 1.630593793], [0.8, 1.893445512], [0.9, 2.242596864], [1.0, 2.709057012], [1.1, 3.337558238], [1.2, 4.193308170], [1.3, 5.372466428], [1.4, 7.018590142], [1.5, 9.348762069], [1.6, 12.69561889], [1.7, 17.57581479], [1.8, 24.80298983], [1.9, 35.67662057], [2.0, 52.30192576]]

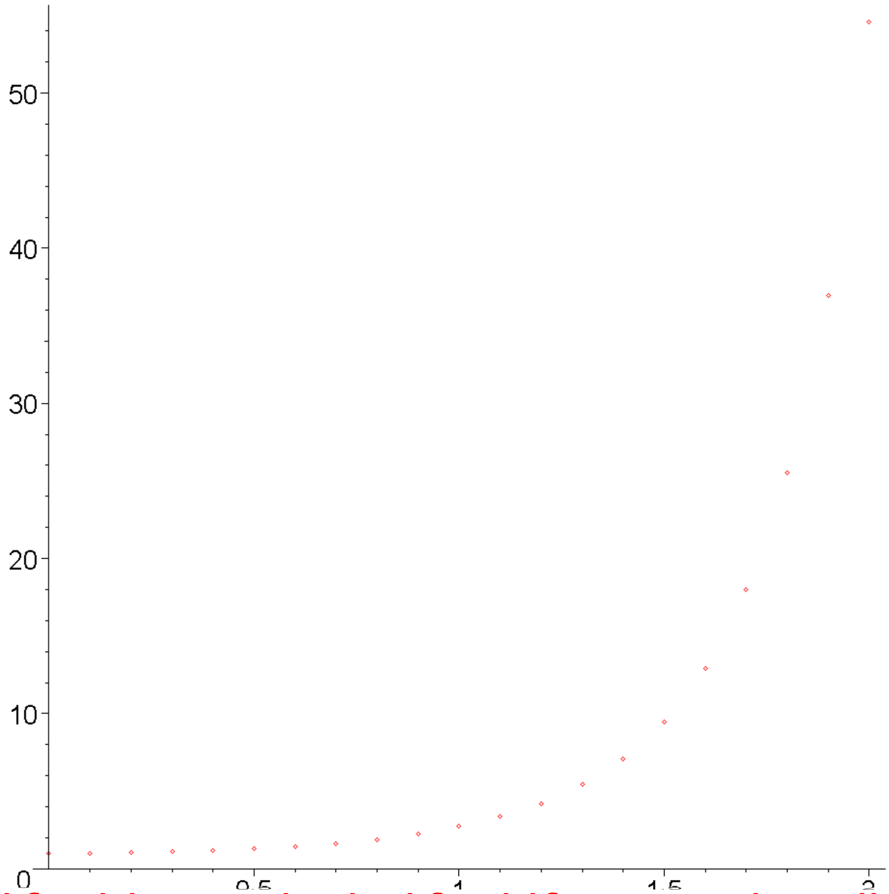
```
> pareulerkuva:= plot(pareulerdata, style=point):
  pareulerkuva;
```




```
> rungekuttadata:= rungekutta(f, x0, y0, h, xend);
```

rungekuttadata := [[0, 1], [0.1, 1.010050166], [0.2, 1.040810769], [0.3, 1.094174264], [0.4, 1.173510813], [0.5, 1.284025255], [0.6, 1.433328994], [0.7, 1.632315187], [0.8, 1.896478467], [0.9, 2.247902589], [1.0, 2.718270175], [1.1, 3.353460191], [1.2, 4.220645591], [1.3, 5.419379283], [1.4, 7.099124725], [1.5, 9.487335476], [1.6, 12.93502908], [1.7, 17.99176113], [1.8, 25.53067937], [1.9, 36.96006234], [2.0, 54.58630867]]

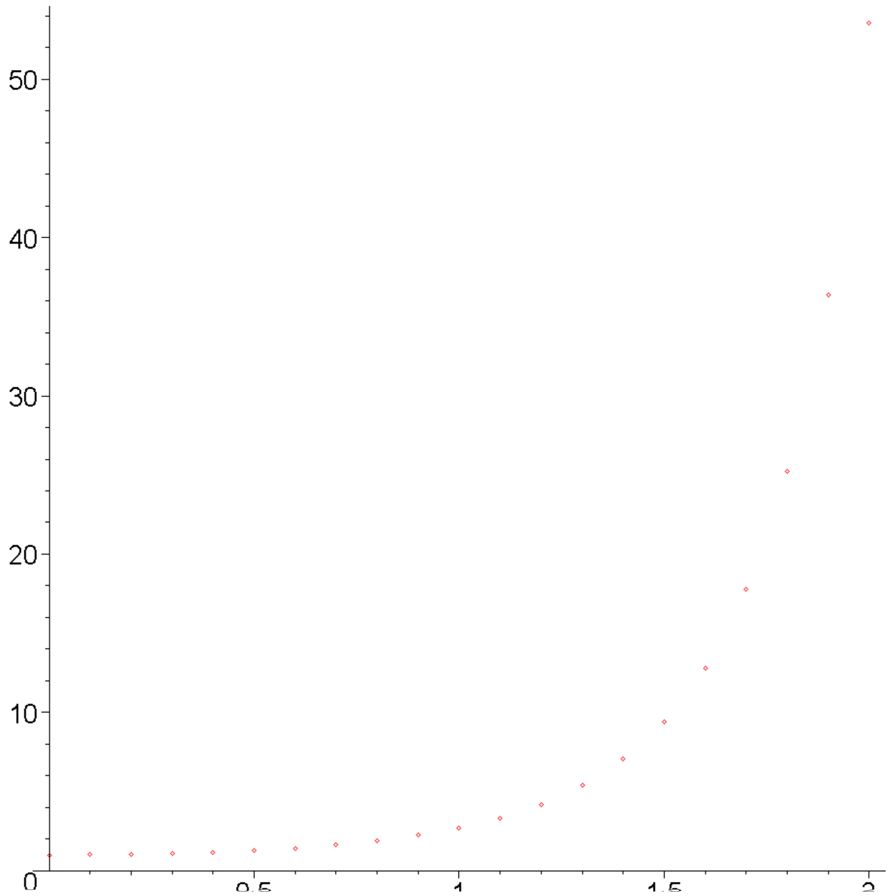
```
> rungekuttakuva:= plot(rungekuttadata, style=point):
  rungekuttakuva;
```




```
> adamsbashforthdata:= adamsbashforth(f, x0, y0, h,xend);
```

adamsbashforthdata := [[0, 1], [0.1, 1.010050166], [0.2, 1.040810769], [0.3, 1.094174264], [0.4, 1.173420046], [0.5, 1.283764305], [0.6, 1.432786640], [0.7, 1.631313792], [0.8, 1.894732637], [0.9, 2.244948474], [1.0, 2.713344710], [1.1, 3.345294745], [1.2, 4.207108940], [1.3, 5.396852546], [1.4, 7.061394527], [1.5, 9.423611254], [1.6, 12.82634935], [1.7, 17.80440109], [1.8, 25.20390923], [1.9, 36.38313185], [2.0, 53.55462781]]

```
> adamsbashforthkuva:= plot(adamsbashforthdata, style=point):
  adamsbashforthkuva;
```

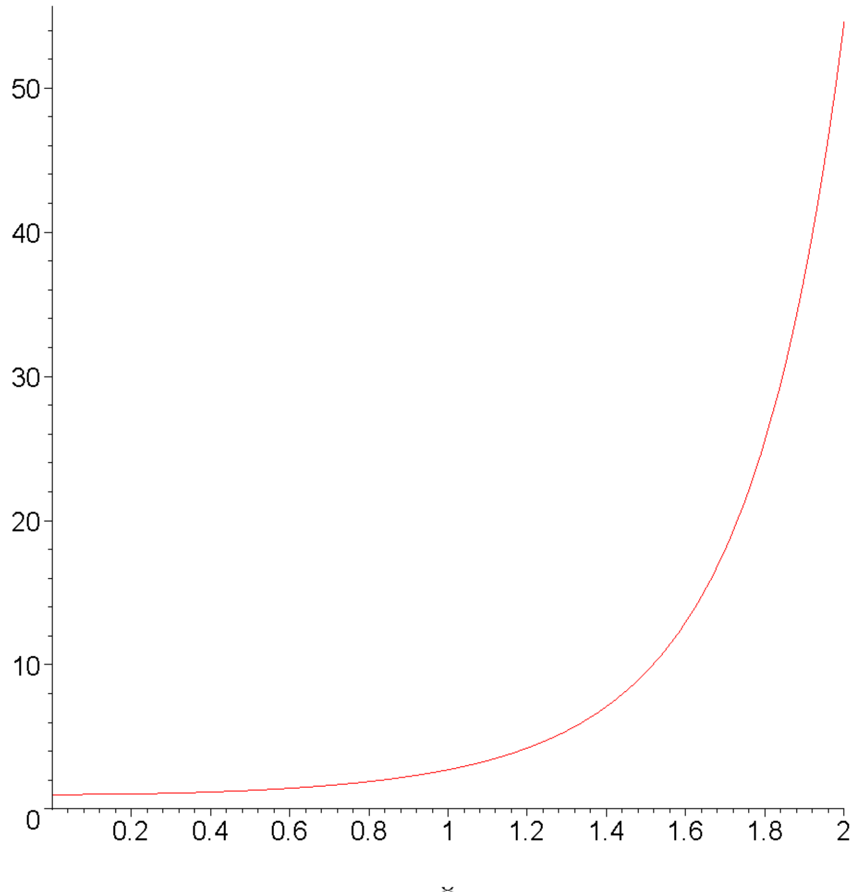


Esimerkkinä oleva alkuarvoprobleema on myös ratkaistavissa algebrallisesti, jolloin voidaan verrata eri menetelmien tarkkuutta toisiinsa ja myös tarkkaan ratkaisuun:

```
> tarkkaratkaisu:= dsolve({diff(y(x), x)=2*x*y(x), y(x0)=y0},
  y(x));
```

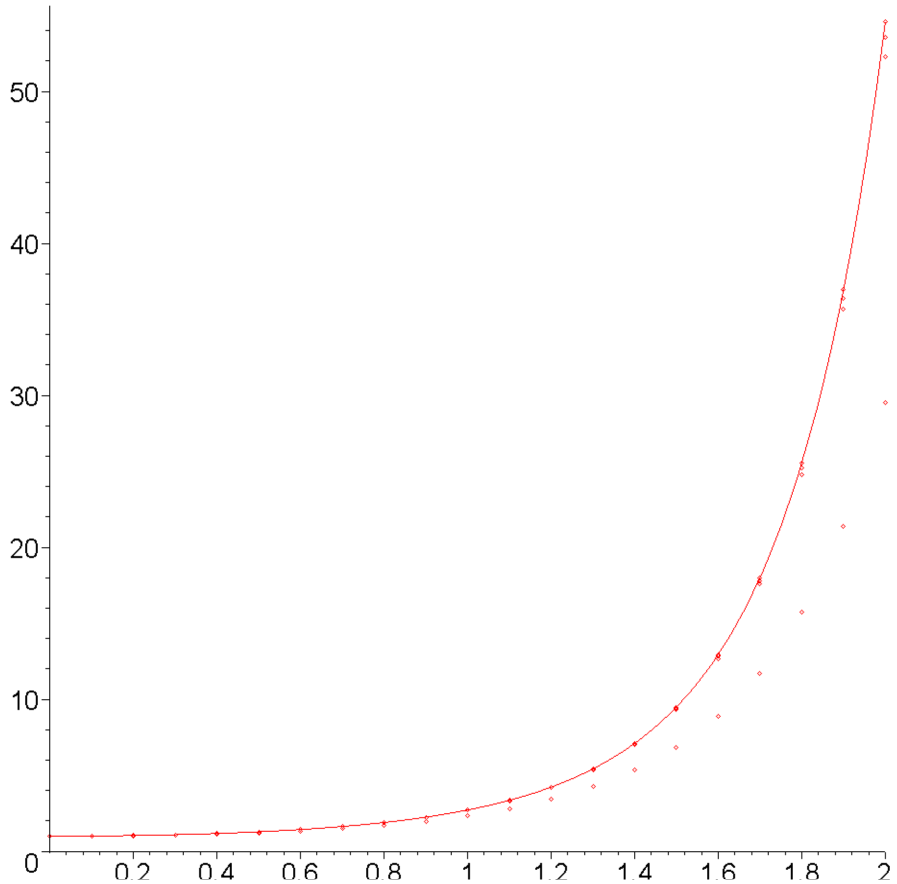
$$tarkkaratkaisu := \mathrm{y}(x) = \mathbf{e}^{(x^2)}$$

```
> tarkkakuva:= plot(rhs(tarkkaratkaisu), x=0..xend):
  tarkkakuva;
```



Kaikki ratkaisut samassa kuvassa:

```
> with(plots):
> display(eulerkuva, pareulerkuva, rungekuttakuva,
  adamsbashforthkuva, tarkkakuva);
```



Lopuksi eri menetelmillä saadut arvot tarkasteluvälin loppupisteessä:

```
> eulerdata[-1][2],
  pareulerdata[-1][2],
  rungekuttadata[-1][2],
  adamsbashforthdata[-1][2],
  evalf(subs(x=2,rhs(tarkkaratkaisu)));
```

$$29.49864321, 52.30192576, 54.58630867, 53.55462781, 54.59815003$$

```
>
```

Edellä olevia syötteitä voi muuntaa ja tämän jälkeen ajaa koko muistikirjan yhdellä kerralla (valikko **Edit** / **Execute** / **Worksheet**).

Linkit

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

SKK & MS 31.05.2001