Numeriikkaa

Interpolointi ja funktioiden approksimointi

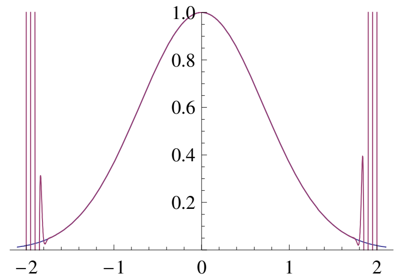
- Motivaatio
  - funktion arvot tunnetaan vain äärellisessä pistejoukossa
  - funktion arvojen laskeminen on työlästä
- Oletetaan, että funktion f arvot tunnetaan pisteissä x_i, $i = 1, \ldots, n$. Merkitään $f(x_i) \equiv f_i$. Pisteitä (x_i, f_i) sanotaan datapisteiksi (esim. mittaustuloksia).
- Interpolaatio tarkoittaa funktion arvojen määrittämistä pisteissä x: $x_i < x < x_{i+1}$, $i = 1, \ldots, n-1$.
  - Jos halutaan laskea $f(x)$, kun $x < x_1$ tai $x > x_n$, kyse on ekstrapolaatiosta. Tällöin yleensä arvataan funktion käyttäytyminen ja sovitetaan arvattu malli dataan "mahdollisimman hyvin" (tästä myöhemmin lisää).
  - Interpolaatiota tai funktioiden approksimointia tarvitaan esim. funktion integraalia laskettaessa, tai yleensäkin jos funktiota tarvitaan jossain laskussa ja joko siitä ei tiedetä tarpeeksi tai se on hankala käsitellä.
- Funktiota voidaan approksimoida polynomilla, esim.

$$p(x) = \sum_{i=1}^{n} f_i \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

  - Huom. yo. polynomi on astetta $n - 1$.
  - *Tehtävä*: osoita, että $p(x_i) = f_i$, $i = 1, \ldots, n$.
  - Polynomi on helppo laskea ja käsitellä.
  - Polynomi on jatkuva ja äärettömän monta kertaa derivoituva.
  - Suurilla n:n arvoilla polynomi oskilloi voimakkaasti

```
In[534]:= a = Table[{x, Exp[-x^2]}, {x, -2, 2, 0.05}] // N;
          b = InterpolatingPolynomial [a, x];
          Plot[{Exp[-x^2], b}, {x, -2.1, 2.1},
           PlotRange → {0, 1}, AxesOrigin → {0, 0}, ImageSize → Small]
```

Out[536]=



– Ei sovellu ekstrapolaatioon

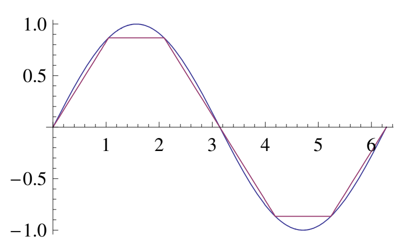
- Paloittainen interpolointi
  - Funktiota f approksimoidaan jokaisella välillä $x_i < x < x_{i+1}$ eri funktioilla, merk. $f(x) \approx q_i(x)$, kun $x_i < x < x_{i+1}$, $i = 1, \ldots, n-1$.
  - Interpoloiva funktio on nyt paloittain määritelty

$$Q(x) = \begin{cases} q_1(x), & x_1 < x < x_2 \\ q_2(x), & x_2 < x < x_3 \\ \cdots \\ q_{n-1}(x), & x_{n-1} < x < x_n \end{cases} \tag{1}$$

  - Funktiot q_i voivat olla esim. polynomeja, yksinkertaisimmillaan vakio- tai lineaarisia funktioita.

```
In[408]:= a = Table[{x, Sin[x]}, {x, 0, 2 π, π / 3}];
          b = Interpolation [a, InterpolationOrder → 1];
          Plot[{Sin[x], b[x]}, {x, 0, 2 π}, ImageSize → Small]
```

Out[410]=

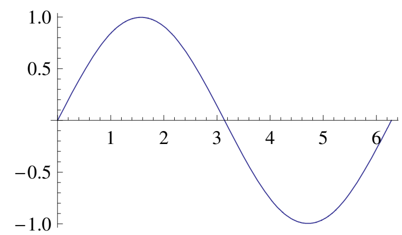


  - Jos q_i ovat suoria, ts. $q_i(x) = a_i(x - x_i) + b_i$, $i = 1, \ldots, n$, kertoimet a_i ja b_i saadaan määrättyä suoraan funktion arvoista: $q_i(x_i) = f_i$.
  - Jos q_i ovat korkeamman asteen polynomeja, tarvitaan lisäehtoja. Usein vaaditaan esim. interpoloivan funktion derivaatan jatkuvuus pisteissä x_i.
  - Paljon käytetty interpolaatio on ns. luonnollinen kuutiollinen spline, jossa kukin q_i on 3. asteen polynomi, ja Q' ja Q'' ovat jatkuvia, kun $x_1 < x < x_n$, ja $Q''(x_1) = Q''(x_n) = 0$.

– Mathematicassa em. spline-sovitus antaa parametrisen käyrän.

```
<< Splines`
s = SplineFit [a, Cubic];
ParametricPlot [s[x], {x, 0, 2 π},
 ImageSize → Small,
 AspectRatio → 1 / GoldenRatio ]
```

Out[604]=



– Paloittain määritelty interpolaatiofunktio käyttäytyy datapisteiden väleissä paremmin kuin yksi polynomi, sillä polynomin aste pysyy kurissa.

–

- Huom. tässä jätetään paljon hienouksia, kuten virheen arviointia, pois. Lisätietoa löytyy muista numeriikan kursseista.

Derivointi

- Jos funktion lauseke on tiedossa, derivaatan likiarvo saadaan suoraan määritelmästä:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

valitsemalla h:n arvo halutun tarkkuuden mukaan.

- Funktion f derivaatoille voidaan johtaa likimääräisiä lausekkeita f:n Taylorin sarjasta

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i$$

- Oletetaan jälleen, että funktion arvot tunnetaan pisteissä x_i, $i = 1, \ldots, n$, $f(x_i) \equiv f_i$. Oletetaan lisäksi, että pisteet x_i ovat tasavälein, ts. $x_{i+1} - x_i = h$, $i = 1, \ldots, n-1$.

  – 1. derivaatalle saadaan tarkempi arvio:

$$f'(x_i) = \frac{f_{i+1} - f_{i-1}}{2h} - \frac{1}{6} f'''(x_i) h^2 + \ldots \tag{2}$$

  – 2. derivaatta

$$f''(x_i) = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} - \frac{1}{12} f^{(4)}(x_i) h^2 - \ldots \tag{3}$$

– 3. derivaatta

$$f'''(x_i) = \frac{f_{i+2} - 2f_{i+1} + 2f_{i-1} - f_{i-2}}{2h^2} - \frac{1}{4} f^{(5)}(x_i)h^2 + \ldots \qquad (4)$$

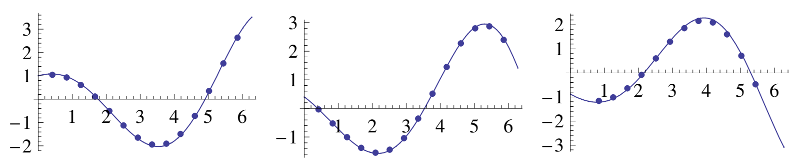
– ja niin edelleen... Funktion n:nnen derivaatan arvon laskemisessa tarvitaan funktion arvo $n+1$ pisteessä.

– *Tehtävä*: Todista yo. kaavat (kynällä ja paperilla) lähtien f:n Taylorin sarjasta.

In[592]:=
```
f[x_] := Exp[0.2 x] Sin[x]; n = 16; Clear[t];
t[1] = 0; t[n] = 2 π;
Do[t[i] = t[1] + (i - 1) (t[n] - t[1])/(n - 1), {i, 2, n - 1}];
h = t[2] - t[1];
df = Table[{t[i], (f[t[i + 1]] - f[t[i - 1]])/(2 h)},
   {i, 2, n - 1}];
ddf = Table[{t[i], (f[t[i + 1]] - 2 f[t[i]] + f[t[i - 1]])/h^2},
   {i, 2, n - 1}];
dddf = Table[{t[i], (f[t[i + 2]] - 2 f[t[i + 1]])/(2 h^3) +
      (2 f[t[i - 1]] - f[t[i - 2]])/(2 h^3)}, {i, 3, n - 2}];
```

In[599]:=
```
g1 = Show[ListPlot[df], Plot[f'[x], {x, 0, 2 π}]];
g2 = Show[ListPlot[ddf], Plot[f''[x], {x, 0, 2 π}]];
g3 = Show[ListPlot[dddf], Plot[f'''[x], {x, 0, 2 π}]];
GraphicsArray[{g1, g2, g3}]
```

Out[602]=



- Derivaatan arvon laskemiseksi datapisteistön reunoilla, $f'(x_1)$ ja $f'(x_n)$, tarvitaan reunaehtoja, esim. $f(x_1 - h) = f(x_n + h) = 0$.
- Derivaattafunktiolle saa approksimaation myös interpoloimalla datapisteitä ja laskemalla interpoloivan funktion derivaatan.

Integrointi

- Tarkastellaan funktion f määrättyä integraalia yli välin $[a, b]$. Tavanomainen (Riemannin) integraali määritellään karkeasti ottaen "Riemannin summien"

$$\sum_{i=1}^{n-1} f(t_i)(x_{i+1} - x_i)$$

raja-arvona, kun välin $[a, b]$ jakoa osaväleihin $[x_i, x_{i+1}]$ tihennetään. Tässä $x_1 = a$, $x_n = b$ ja $t_i \in [x_i, x_{i+1}]$, $i = 1, \ldots, n-1$.

- Olkoon taas $x_{i+1} - x_i = h$, $i = 1, \ldots, n-1$ ja merk. $f(x_i) \equiv f_i$. Yksinkertaisin approksimaatio integraalille saadaan kiinnittämällä yllä $t_i = x_i$:

$$\int_{x_1}^{x_n} f(x)\, dx \approx h \sum_{i=1}^{n-1} f_i$$

- Parempia approksimaatioita määrätylle integraalille saa interpoloimalla funktiota f paloittain määritellyllä funktiolla (ks. kaava (1)).
  - Jos q_i ovat 1. asteen polynomeja, $q_i(x) = a_i(x - x_i) + b_i$, saadaan ns. *puolisuunnikassääntö*:

$$\int_{x_1}^{x_n} f(x)\, dx \approx h \sum_{i=1}^{n-1} \frac{f_i + f_{i+1}}{2} = \frac{f_1}{2} + \sum_{i=2}^{n-1} f_i + \frac{f_n}{2}$$

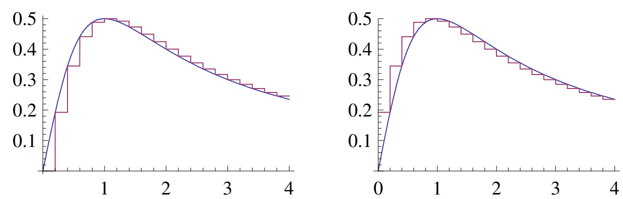
    Tehtävä: Todista yo. kaava!
  - Jos f:ää interpoloidaan ***kahden välin yli*** ulottuvilla toisen asteen polynomeilla, saadaan ns. Simpsonin kaava:

$$\int_{x_1}^{x_n} f(x)\, dx \approx \frac{h}{3}\left(f_1 + 4f_2 + 2f_3 + 4f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n\right)$$

- Numeerinen integrointi on oma taiteenlajinsa. Tämä oli vain alkua.

```
In[392]:= f[x_] := x/(1 + x^2); n = 21; Clear[t];
          t[1] = 0; t[n] = 4;
          Do[t[i] = t[1] + (i - 1) (t[n] - t[1])/(n - 1), {i, 2, n - 1}];
          h = t[2] - t[1];

In[396]:= g[x_] := (For[i = 1; gx = 0, i ≤ n, i ++,
             If[x ≥ t[i] && x ≤ t[i + 1], gx = f[t[i]];
              Break]]; gx)

In[397]:= j = Interpolation[Table[{t[i], f[t[i]]}, {i, 1, n}],
             InterpolationOrder → 0];

In[401]:= g1 = Plot[{f[x], g[x]}, {x, t[1], t[n]}];
          g2 = Plot[{f[x], j[x]}, {x, t[1], t[n]},
             PlotRange → {0, 0.5}];
          GraphicsArray[{g1, g2}]
```

Out[403]=



- Jostain syystä Mathematica ei laske paloittain määritellyn funktion integraalia oikein.

```
k[x_] := (For[i = 0; kx = 0, i ≤ n, i ++,
    If[x ≥ t[i] && x ≤ t[i + 1],
      kx = (f[t[i + 1]] - f[t[i]])/h (x - t[i]) + f[t[i]];
      Break]]; kx)

In[431]:= h NSum[(f[t[i]] + f[t[i + 1]])/2, {i, 1, n - 1}]
          h NSum[f[t[i]], {i, 1, n - 1}]
          NIntegrate[j[x], {x, 0, 4}]

Out[431]= 1.40235

Out[432]= 1.35529

Out[433]= 1.44941
```

Funktion sovittaminen dataan

- Motivaatio
  - etsiä parametreille sellaiset arvot, että teoreettinen malli vastaa havaintoaineistoa
- Esim. mitataan jonkin elektronisen komponentin läpi kulkeva virta I sen päiden välisen jännitteen V funktiona. Jos oletetaan, että komponentti noudattaan Ohmin lakia, $V = RI$, tehtävänä on määrittää sellainen resistanssin R arvo, että funktio $V(I)$ vastaa havaintoaiheistoa (V_i, I_i) *mahdollisimman hyvin.*
- Paljon käytetty sovitusmenetelmä on *pienimmän neliösumman menetelmä.* Olk. havaintoaineisto (x_i, y_i), $i = 1, \ldots, n$. Pienimmän neliösumman menetelmässä etsitään funktio f, jolle

  $$\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

  saa pienimmän arvonsa. Huom. *ei vaadita, että funktio f kulkisi yhdenkään datapisteen kautta.*
- Esim. $f(x) = ax + b$. Merk.

  $$S = \sum_{i=1}^{n}(ax_i + b - y_i)^2.$$

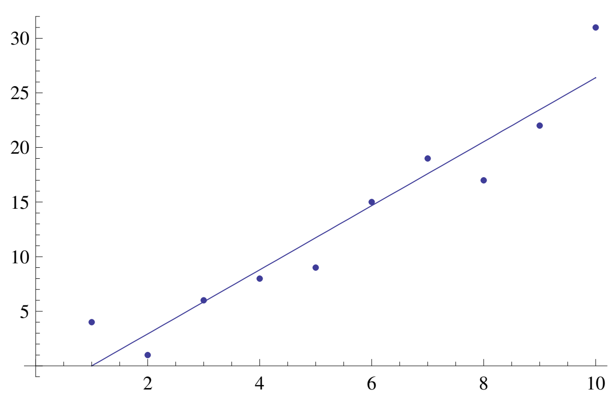
  Derivoidaan tämä a:n ja b:n suhteen, asetetaan derivaatat nolliksi ja ratkaistaan a ja b:

  $$a = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2}$$
  $$b = \frac{S_y S_{xx} - S_x S_{xy}}{nS_{xx} - S_x^2},$$

  missä $S_x = \sum x_i$, $S_y = \sum y_i$, $S_{xx} = \sum x_i^2$ ja $S_{xy} = \sum x_i y_i$. *Tehtävä*: Laske yo. laskun välivaiheet!

In[629]:=
```
a = Table[{n, Prime[n] + Random[Integer, {-2, 2}]}, {n, 1, 10}];
y = Fit[a, {1, x}, x];
Show[ListPlot[a], Plot[y, {x, 1, 10}]]
```

Out[631]=



- Jos halutaan origon kautta kulkeva suora, merkitään vaan $f(x) = ax$ ja lasketaan yo. lasku uudestaan. *Tee tämä!*
- Sovitettava funktio voi periaattessa olla minkä muotoinen tahansa.
- Sovituksen hyvyyttä voidaan (ja pitäisikin) arvioida tilastollisesti. Tässä kurssissa siihen ei puututa.

Yhtälöiden ratkaiseminen

- Mathematicalla onnistuu analyyttinen yhtälöiden ratkaisu kätevästi.
  - Muotoa
    $$a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = 0$$
    olevilla yhtälöillä on aina n kappaletta (kompleksista) juurta. Jos $n \leq 4$, yhtälölle on olemassa *ratkaisukaava*.

```
In[638]:= Solve[{x^2 + 2 y + 1 == 0, x + y == 2}, {x, y}]

Out[638]= {{y → 1 - 2 i, x → 1 + 2 i}, {y → 1 + 2 i, x → 1 - 2 i}}
```

  - Lineaarisia yhtälöryhmiä voidaan ratkaista lineaarialgebran keinoin.

```
Solve[{x + 2 y - 2 z == 1, -x + 2 y + 3 z == 2,
  -2 y + 3 z == -1}, {x, y, z}]
```

$$m = \begin{pmatrix} 1 & 2 & -2 \\ -1 & 2 & 3 \\ 0 & -2 & 3 \end{pmatrix}; \; b = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix};$$

```
LinearSolve[m, b] // MatrixForm
```

Out[649]= $\left\{\left\{x \to -\frac{1}{7},\ y \to \frac{5}{7},\ z \to \frac{1}{7}\right\}\right\}$

Out[651]//MatrixForm=
$$\begin{pmatrix} -\frac{1}{7} \\ \frac{5}{7} \\ \frac{1}{7} \end{pmatrix}$$

- "Hankalan" mallisille yhtälöille, esim. $x = \tan x$, voidaan etsiä likimääräinen ratkaisu. Kirjoitetaan yhtälö muotoon $f(x) = 0$. Kumpikin seuraavista menetelmistä nojaa johonkin alkuarvaukseen yhtälön ratkaisulle. Tämä arvaus voidaan etsiä vaikka graafisesti, lukemalla funktion kuvaajasta, tai ehkä fysikaalisilla argumenteilla.
- Newtonin menetelmä. Olkoon x_0 alkuarvaus, ts. $f(x_0)$ on itseisarvoltaan (mielellään) pieni.
  - Piirretään funktiolle tangentti pisteeseen $(x_0, f(x_0))$:
    $$y = f'(x_0)(x - x_0) + f(x_0)$$

– Lasketaan tangentin ja x-akselin leikkauspiste x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

– Piirretään funktiolle tangentti pisteeseen $(x_1, f(x_1))$, lasketaan tangentin ja x-akselin leikkauspiste x_2, ...

– Näin saadaan jono parempia arvioita funktion nollakohdalle:

$$x_0, x_1, x_2, \ldots$$

– Jono katkaistaan pisteessä x_k, kun $f(x_k)$ on itseisarvoltaan riittävän pieni.

– Jos pisteeseen $(x_i, f(x_i))$ piirretty tangentti on liian loiva, seuraava piste $(x_{i+1}, f(x_{i+1}))$ voi olla kaukana tarkasta ratkaisusta.

– Newtonin menetelmä ei suppene epäjatkuvuuskohtaan.

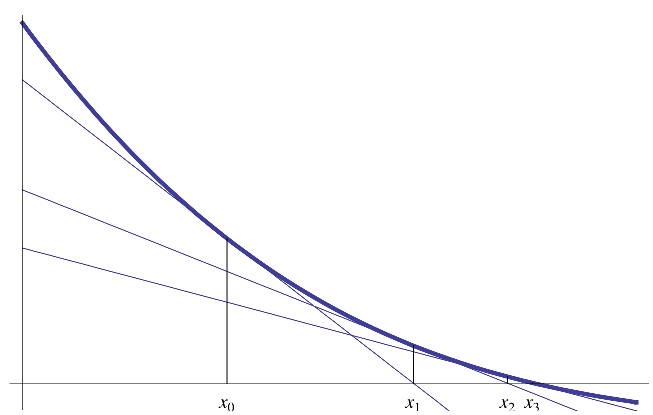
– Jos funktiolla on paikallinen minimi ennen nollakohtaa, algoritmi voi jäädä pyörimään tämän minimin ympärille.

– Jos funktio on sopivalla tavalla symmetrinen nollakohtansa suhteen, algoritmi voi jäädä pyörimään nollakohdan ympärille.

– Huom. Newtonin menetelmä vaatii funktion derivaatan laskemisen

```
c = {{0, 1}, {1, 0}, {0.5, 0.3}, {1.5, -0.1}};
t = Interpolation [c];
y = 0.4; (*alkuarvaus *)
g[0] = Graphics [{Line [{{y, 0}, {y, t[y]}}],
    Text [x₀, {y, -0.05}]}];
Do[
 q[i] = t '[y] (z - y) + t[y]; (*z on muuttuja*)
 y = z /. Flatten [Solve [q[i] == 0, z]];
 g[i] = Graphics [{Line [{{y, 0}, {y, t[y]}}],
    Text [xᵢ, {y, -0.05}]}];,
 {i, 1, 3}]
Show [Plot [t[z], {z, 0, 1.2}, PlotStyle → Thick],
 Table [Plot [q[i], {z, 0, 1.2}], {i, 1, 3}],
 Table [g[i], {i, 0, 3}], Ticks → None]
```

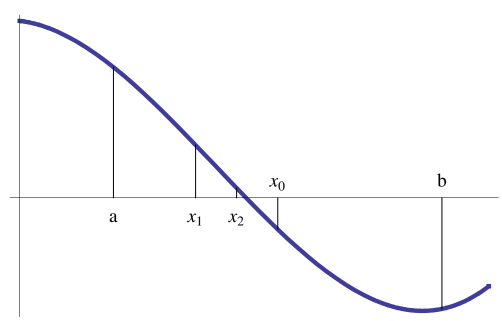
Out[699]=



- Välinpuolitusmenetelmä. Olkoon $[a, b]$ väli, jonka päätepisteissä funktio f saa erimerkkisen arvon: $f(a)f(b) < 0$. Oletetaan, että f on jatkuva, jolloin on olemassa $x \in [a, b]$: $f(x) = 0$.
  - Lasketaan funktion arvo välin keskipisteessä, merk. $x_1 = \frac{a+b}{2}$.
  - Jos $f(a)$ ja $f(x_1)$ ovat erimerkkiset, funktiolla on nollakohta välillä $[a, x_1]$. Jos taas $f(b)$ ja $f(x_1)$ ovat erimerkkiset, nollakohta on välillä $[x_1, b]$. Oletetaan tässä, että $f(a)f(x_1) < 0$.
  - Merk. $x_2 = \frac{a+x_1}{2}$, lasketaan $f(x_2)$ ja valitaan se osaväli ($[a, x_2]$ tai $[x_2, x_1]$), jonka päätepisteissä funktiolla on erimerkkiset arvot.
  - Puolittamalla tarkasteltava väli joka askeleella, saadaan jono arvioita funktion nollakohdalle:
$$x_0, x_1, x_2, \ldots$$
Jokainen x_i on sellaisen välin keskipiste, jolla f vaihtaa merkkiään, ja jonka pituus on $(b-a)/2^{i-1}$
  - Jono katkaistaan (algoritmi päättyy), kun tarkasteltavan välin pituus on tarpeeksi pieni.
  - Välinpuolitusmenetelmä löytää aina joko nollakohdan tai epäjatkuvuuskohdan, ts. se suppenee aina, mutta on hitaampi kuin Newtonin menetelmä.
  - Ei vaadi derivaatan laskemista.

```
c = {{0, 1}, {1, -0.5}, {0.3, 0.5}, {0.7, -0.5}};
t = Interpolation [c];
a = 0.2; b = 0.9; (*alkuarvaukset  a ja b*)
z = (a + b) / 2;
g[0] = Graphics [{Text ["a", {a, -0.1}],
    Line [{{a, 0}, {a, t[a]}}]}];
g[1] = Graphics [{Text ["b", {b, 0.1}],
    Line [{{b, 0}, {b, t[b]}}]}];
g[2] = Graphics [{Text [x0, {z, 0.1}],
    Line [{{z, 0}, {z, t[z]}}]}];
Do[
 If[t[z] > 0, a = z, b = z];
 z = (a + b) / 2;
 g[i] = Graphics [{Text [xi-2, {z, -0.1}],
    Line [{{z, 0}, {z, t[z]}}]}],
 {i, 3, 4}]
Show[Plot[t[x], {x, 0, 1}, PlotStyle -> Thick],
 Table[g[i], {i, 0, 4}], Axes → True, Ticks → None]
```

Out[620]=



- Välinpuolitusmenetelmää käytetään myöhemmin reuna-arvoproblemojen ratkaisemisessa. Se sopii hyvin moneen ongelmaan.

Differentiaaliyhtälöt

- Monia luonnonilmiöitä kuvaa jonkinlainen differentiaaliyhtälö
  - Radioaktiivinen hajoaminen:
  $$\frac{dN}{dt} = -\lambda N$$
  Ratkaisu on eksponentiaalisesti vähenevä funktio. Hajoamisnopeuden määrää *hajoamisvakio* λ.
  - Newtonin liikeyhtälöt, esim. harmoninen oskillaattori:
  $$m\frac{d^2x}{dt^2} = -kx - m\gamma\frac{dx}{dt}$$
  Tämän ratkaisu on vaimenevaa värähtelyä. Toinen esimerkki: Maan vetovoiman alaisuudessa liikkuva kappale, joka kokee väliaineen vastuksen:
  $$m\mathbf{r}''(t) = -G\frac{mM}{|\mathbf{r}(t)|^3}\mathbf{r}(t) - \gamma|\mathbf{r}'(t)|\mathbf{r}'(t)$$
  Tämä vastaa kolmea kytkettyä differentiaaliyhtälöä ($x(t)$:lle, $y(t)$:lle ja $z(t)$:lle).
  - Kirchhoffin lait virtapiireissä voidaan muotoilla differentiaaliyhtälöiden avulla. Esim. LR-piiri:
  $$L\frac{dI}{dt} + RI = V(t)$$
  Jos piirissä on käämi ja kondensaattori, sille kirjoitettu Kirchhoffin laki muistuttaa hyvin paljon harmonisen oskillaattorin liikeyhtälöä.
  - Eliöpopulaatioiden kilpailu
  $$\begin{cases} \frac{dN_1}{dt} = N_1(\alpha - \beta N_2) \\ \frac{dN_2}{dt} = N_2(\gamma - \epsilon N_1) \end{cases}$$
  Tässä kaksi populaatiota (suuruudet $N_1(t)$ ja $N_2(t)$) kilpailevat saman ympäristön resursseista. Toisen populaation kasvu varjostaa toisen menestymistä.
  - Kvanttimekaaniset todennäköisyysjakaumat:
  $$\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2} + V(x)\psi = E\psi$$
  Tämä on esimerkki differentiaaliyhtälöstä, jolla on *ominaisarvo*. Yhtälön ratkaisuna saadaan ominaisarvo E ja sitä vastaava funktio $\psi(x)$.

Analyyttinen ratkaisu

- Yksinkertaisten (yleensä *lineaaristen*) differentiaaliyhtälöiden analyyttinen ratkaisu hoituu Mathematicalla kätevästi.

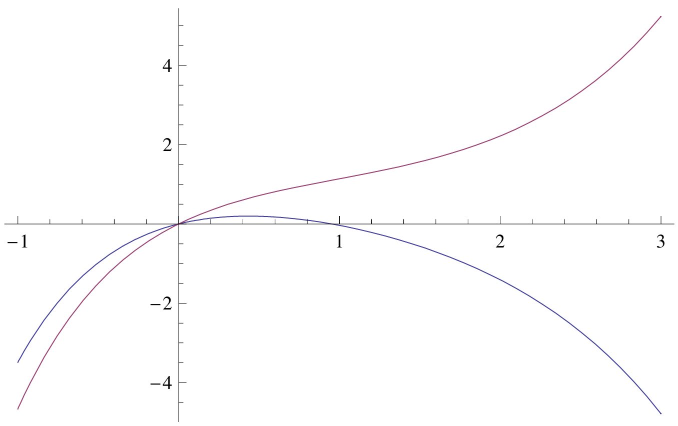
```
In[86]:= p = DSolve[y''[x] + y'[x] == -2 x^2, y[x], x]
         q = DSolve[{x'[t] + y[t] == a Exp[-t],
             y'[t] + x[t] == b Exp[-t], x[0] == 0, y[0] == 0},
            {x[t], y[t]}, t]
         r = {a → 1, b → 2};
         Plot[Evaluate[{x[t], y[t]} /. q /. r], {t, -1, 3}]
```

Out[86]= $\left\{\left\{y[x] \to -4\,x + 2\,x^2 - \frac{2\,x^3}{3} - e^{-x}\,C[1] + C[2]\right\}\right\}$

Out[87]= $\left\{\left\{x[t] \to \frac{1}{4}\,e^{-t}\left(-a + b + a\,e^{2\,t} - b\,e^{2\,t} + 2\,a\,t + 2\,b\,t\right),\right.\right.$

$\left.\left. y[t] \to -\frac{1}{4}\,e^{-t}\left(-a + b + a\,e^{2\,t} - b\,e^{2\,t} - 2\,a\,t - 2\,b\,t\right)\right\}\right\}$

Out[89]=



- Jos differentiaaliyhtälössä esiintyy n:nnen kertaluvun derivaattoja, sen yleisessä ratkaisussa on n vakiota. Nämä vakiot voidaan määrätä *reunaehdoilla*.
- Harjoituksissa esimerkkejä.

Numeerinen ratkaisu

- Differentiaaliyhtälöiden numeeriseen ratkaisuun on monia menetelmiä. Tässä kurssissa ei käydä mitään niistä kovin tarkasti läpi. Esitellään niistä eräs helppotajuinen ja yleisesti käytetty.
- *Differenssimenetelmässä* differentiaaliyhtälössä esiintyvät derivaatat korvataan likimääräisillä erotusosamäärillä (ks. kappale derivoinnista). Tarkastellaan esimerkiksi harmonisen oskillaattorin liikeyhtälöä:

$$\frac{d^2x(t)}{dt^2} + \gamma\frac{dx(t)}{dt} + \omega^2 x(t) = 0, \tag{5}$$

missä $\omega^2 = k/m$.

Alkuarvoprobleema

- Tarkastellaan yhtälön (5) ratkaisemista *alkuarvoprobleemana*, ts. oletetaan, että systeemin tilanne (paikka ja nopeus) alussa on tunnettu, ts. $x(t_0)$ ja $x'(t_0)$ on tunnettu.

- Muodostetaan yhtälö (5) pisteissä t_i, $i = 1, \ldots$:

$$\left.\frac{d^2x(t)}{dt^2}\right|_{t=t_i} + \gamma \left.\frac{dx(t)}{dt}\right|_{t=t_i} + \omega^2 x(t_i) = 0.$$

  Sijoitetaan tähän yhtälöön derivaattojen approksimaatiot yhtälöistä (2) ja (3):

$$\frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} + \gamma \frac{x_{i+1} - x_{i-1}}{2h} + \omega^2 x_i = 0, \qquad (6)$$

  missä $i = 1, \ldots$.

- Yhtälö (6) on yhtälön (5) *diskretöity* versio, ts. differentiaaliyhtälöä (5) vastaava *differenssiyhtälö*.

- Yhtälöstä (6) voidaan ratkaista x_{i+1} x_i:n ja x_{i-1}:n funktiona:

$$(1 - \frac{\gamma h}{2})x_{i-1} + (\omega^2 h^2 - 2)x_i + (1 + \frac{\gamma h}{2})x_{i+1} = 0,$$

  josta

$$x_{i+1} = \frac{1}{1 + \frac{\gamma h}{2}} \left((2 - \omega^2 h^2)x_i + (\frac{\gamma h}{2} - 1)x_{i-1} \right)$$

  Oskillaattorin paikka hetkellä t_{i+1} määräytyy siis sen paikoista kahdella aikaisemmalla hetkellä t_i ja t_{i-1}. Jotta yhtälö voidaan ratkaista, oskillaattorin paikka on tunnettava *kahdella* ensimmäisellä ajanhetkellä.

- Yleisemmin: Jos yhtälössä esiintyy n:nnen kertaluvun derivaattoja, se tarvitsee reunaehtoina x_i:n arvot n pisteessä.

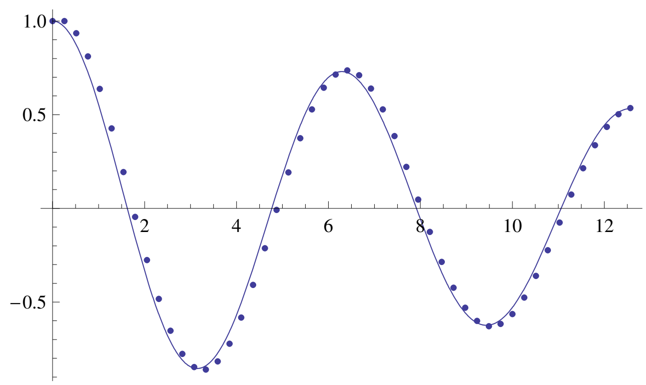
- Usein alkuehtona annetaan kappaleen paikka ja nopeus alkuhetkellä, ts. x_1 ja $x'(t_1)$. Tällöin voidaan esimerkiksi approksimoida

$$x'(t_1) \approx \frac{x_2 - x_1}{h},$$

  josta saadaan $x_2 \approx x_1 + x'(t_1)h$.

In[62]:=
```
n = 50; t[1] = 0; t[n] = 4 π; h = (t[n] - t[1])/(n - 1);
Do[t[i] = t[1] + (i - 1) h, {i, 2, n - 1}]
γ = 0.1; ω = 1; x0 = 1; dx0 = 0;
x[1] = x0; x[2] = x[1] + dx0 h;
Do[x[i] = 1/(1 + γh/2) ((2 - ω^2 h^2) x[i - 1] + (γh/2 - 1) x[i - 2]),
  {i, 3, n}]
ϕ[r_] = y[r] /. Flatten[DSolve[
      {y''[r] + γ y'[r] + ω^2 y[r] == 0, y[0] == 1, y'[0] == 0},
      y[r], r]];
Show[ListPlot[Table[{t[i], x[i]}, {i, n}]],
  Plot[ϕ[r], {r, 0, t[n]}]]
```

Out[68]=



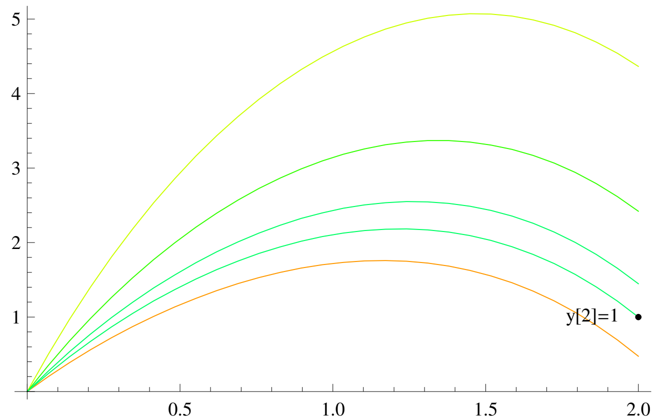
Kahden pisteen reuna-arvoprobleema

- Jos probleeman reunaehdot koskevat funktion käyttäytymistä kahdessa (voi olla useampikin piste, mutta rajoitutaan tässä kahteen) eri pisteessä, edellä esitetty ratkaisumenetelmä ei toimi, sillä alkuehdoista ei saada suoraan ratkaistua funktion arvoa missään pisteessä.
- Eräs tapa on arvata ratkaisu ensimmäisessä pisteessä, ja katsoa kuinka paljon arvattu ratkaisu poikkeaa halutusta reuna-arvosta toisessa pisteessä. Tämän poikkeaman mukaan hiotaan arvausta. (ns. *shooting method*)

```
eq = y''[x] + y'[x] == -2 x^2;
n = 30; t[1] = 0; t[n] = 2; h = (t[n] - t[1])/(n - 1);
Do[t[i] = t[1] + (i - 1) h, {i, 2, n - 1}]
(*reunaehdot olkoot y[0]==0 ja y[2]==1*)
w[1] = 0; (*merk. funktion arvoja w[i]*)
(*arvataan funktion arvo toisessa pisteessä*)
a = 0.2; w[2] = a;
Do[w[i] = 1/(1/h^2 + 1/(2 h)) (2/h^2 w[i - 1] + (1/(2 h) - 1/h^2) w[i - 2] - 2 t[i]^2);,
 {i, 3, n}]
Print["w[2]=", a, " => w[n]=", w[n]];
j = 1; g[j] = ListPlot[Table[{t[i], w[i]}, {i, n}],
  Joined -> True, PlotStyle -> Hue[j / 10]];
(*toinen alkuarvaus välinpuolitusmenetelmälle*)
b = 0.5; w[2] = b;
Do[w[i] = 1/(1/h^2 + 1/(2 h)) (2/h^2 w[i - 1] + (1/(2 h) - 1/h^2) w[i - 2] - 2 t[i]^2);,
 {i, 3, n}]
Print["w[2]=", b, " => w[n]=", w[n]];
j++; g[j] = ListPlot[Table[{t[i], w[i]}, {i, n}],
  Joined -> True, PlotStyle -> Hue[j / 10]];
(*alkuarvauksilla a ja b w[n] heittää eri suuntiin
  halutusta arvosta. sovelletaan välinpuolitusmenetelmää:*)
While[Abs[a - b] > 10^-10,
 z = (a + b) / 2;
 w[2] = z;
 Do[w[i] = 1/(1/h^2 + 1/(2 h)) (2/h^2 w[i - 1] + (1/(2 h) - 1/h^2) w[i - 2] - 2 t[i]^2);,
  {i, 3, n}];
 j++; g[j] = ListPlot[Table[{t[i], w[i]}, {i, n}],
   Joined -> True, PlotStyle -> Hue[j / 10]];
 If[w[n] > 1, b = z, a = z];]
jmax = j; Print["w[2]=", z, " => w[n]=", w[n]]
```

```
w[2]=0.2 => w[n]=0.47351

w[2]=0.5 => w[n]=4.36498

w[2]=0.240588 => w[n]=1.
```

```
Show[Table[g[j], {j, 1, 4}], g[jmax], Graphics[Point[{2, 1}]],
 Graphics[Text["y[2]=1", {1.85, 1}]], PlotRange → All]
```



- Jos reunapisteet ovat “hankalia”, esim. singulaarisia pisteitä (nimittäjän nollakohta tms.), tai reunaehtoina annetaan funktion *asymptoottinen käyttäytyminen* esim. äärettömyydessä tai origossa, ei edellä kuvattu menetelmä välttämättä ollenkaan toimi. Parempi tapa on luoda yritefunktiot kumpaankin reunapisteeseen erikseen ja sovittaa nämä arvaukset yhteen jossain välipisteessä.

- Tällaisia yhtälöitä kannattaa tutkia analyyttisesti, esim. sarjakehitelmien avulla.

- Olkoon yhtälö muotoa

$$y''(r) + 2\frac{y(r)}{r} - y(r) = 0 \tag{7}$$

ja reunaehtoina annetaan, että $y(0) = 0$ ja $y(10) = 10^{-4}$. Muodostetaan yhtälöä (7) vastaava differenssiyhtälö:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \left(\frac{2}{r_i} - 1\right) y_i = 0, \tag{8}$$

ts.

$$y_{i+1} = \left(h^2 + 2 - \frac{2h^2}{r_i}\right) y_i - y_{i-1}, \tag{9}$$

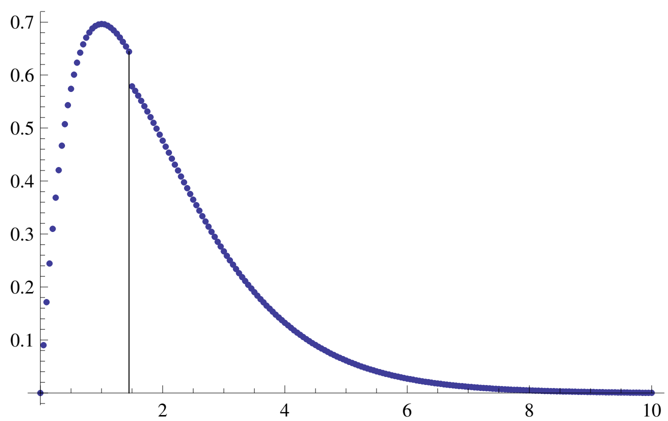
missä $y_i = y(r_i)$ ja $0 \leq r_i \leq 10$. Reunaehdoista saadaan siis $y_1 = 0$ ja $y_n = 10^{-4}$.

- Arvataan funktion derivaatta kummassakin reunapisteessä, ts. arvataan funktion arvot y_2 ja y_{n-1}. Lähdetään ratkaisemaan yhtälöä (9) lähtien kummaltakin reunalta ja sovitetaan ratkaisut yhteen jossakin välipisteessä, merk. r_m. Kohti origoa kuljettaessa yhtälöstä (9) täytyy ratkaista y_{i-1} y_i:n ja y_{i+1}:n funktiona:

$$y_{i-1} = \left(h^2 + 2 - \frac{2h^2}{r_i}\right) y_i - y_{i+1}, \tag{10}$$

```
(* merkitään r[m]:llä sovituspistettä *)
m = 30; y[1] = 0.0; a1 = 0.09; y[2] = a1;
Do[y[i] = (h^2 + 2 - 2 h^2/r[i - 1]) y[i - 1] - y[i - 2],
  {i, 3, m}];
y[n] = 10.0^-4; b1 = 1.7 * 10^-4; y[n - 1] = b1;
Do[y[i] = (h^2 + 2 - 2 h^2/r[i - 1]) y[i + 1] - y[i + 2],
  {i, n - 2, m + 1, -1}];
Show[ListPlot[Table[{r[i], y[i]}, {i, 1, n}]],
 Graphics[Line[{{r[m], 0}, {r[m], y[m]}}]],
 PlotRange → All]
(* a1:n ja b1:n arvoja muuttelemalla funktiosta
    saadaan (annetun resoluution rajoissa) jatkuva
    ja derivoituva *)
(* huom. funktion arvojen absoluuttista suuruutta
 ei tiedetä -- sitä varten täytyisi olla lisäehtoja,
 esim. funktion integraalille yli tarkasteltavan
 välin*)
```

Out[481]=



- Toinen tapa ratkaista useamman pisteen reuna-arvoprobleemoja on lineaarialgebran avulla (lineaarisen yhtälöryhmän ratkaisu). Siirretään differenssiyhtälöissä kaikki muu paitsi funktion arvot oikealle puolelle ja saadaan matriisiyhtälö
$$MY = B \tag{11}$$
missä $Y = (y_1, \ldots, y_n)$, M on yhtälöryhmän kerroinmatriisi, ja B on vektori, joka sisältää "vakiot".

- Esimerkkinä yhtälö
$$y''(x) + y'(x) = x \tag{12}$$
Ratkaistaan tämä pisteissä x_i, $i = 1, \ldots, n$, reunaehdoilla $y(x_1 - h) = 0$ ja $y(x_n + h) = 1$. Tätä vastaava differenssiyhtälö on (osoita!)
$$\left(\frac{1}{h^2} + \frac{1}{2h}\right) y_{i+1} - 2y_i + \left(\frac{1}{h^2} - \frac{1}{2h}\right) y_{i-1} = x_i \tag{13}$$

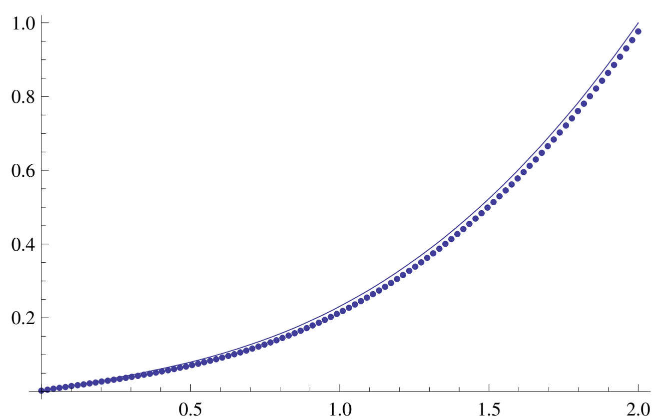
kun $i = 2, \ldots, n-1$. Kun $i = 1$, saadaan

$$\left(\frac{1}{h^2} + \frac{1}{2h}\right) y_2 - 2y_1 = x_1 \tag{14}$$

ja kun $i = n$:

$$-2y_n + \left(\frac{1}{h^2} - \frac{1}{2h}\right) y_{n-1} = x_n - \frac{1}{h^2} - \frac{1}{2h} \tag{15}$$

```
nn = 100; x[1] = 0.0; x[nn] = 2.0; h = (x[nn] - x[1])/(nn - 1);
Do[x[i] = x[1] + (i - 1) h, {i, 2, nn - 1}];
A = Table[Switch[i - k, -1, 1/h^2 + 1/(2 h), 0, -2/h^2, 1, 1/h^2 - 1/(2 h), _, 0],
   {i, nn}, {k, nn}];
B = Table[If[i == nn, x[i] - (1/h^2 + 1/(2 h)), x[i]], {i, nn}];
xx = LinearSolve[A, B];
q[p_] = r[p] /. Flatten[DSolve[{r''[p] + r'[p] == p,
      r[0] == 0, r[2] == 1}, r[p], p]];
Show[Plot[q[p], {p, 0, 2}],
 ListPlot[Table[{x[i], xx[[i]]}, {i, nn}]]]
```



- Lineaarialgebrallinen ratkaisumenetelmä toimii vain tasavaliselle pisteistölle. Toistorakenteeseen perustuvassa menetelmässä askelpituutta voidaan muuttaa funktion ominaisuuksien mukaan (tästä lisää ATK4-kurssissa).
- Harjoituksissa käytetään differenssimenetelmän lisäksi Mathematican valmista funktiota `NDSolve`, jonka avulla resoluutio-ongelmilta vältytään.

Ominaisarvoprobleema

- Esimerkkinä differentiaaliyhtälöstä, jolla on ominaisarvo, tarkastellaan kvanttimekaanista Schrödingerin yhtälöä,

$$-\frac{\hbar^2}{2m}\nabla\psi(\mathbf{r}) + V(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r})$$

Tämän yhtälön ratkaisuna saadaan *aaltofunktio* ψ ja energia E. Vetyatomin elektronia kuvaa Schrödingerin yhtälö (erikoistapaus yo. yhtälöstä)

$$-y''(r) - \frac{2}{r}y(r) = Ey(r), \tag{16}$$

missä r on pallokoordinaatiston radiaalimuuttuja, $r \geq 0$. Tässä potentiaali on siis elektronin ja protonin sähköisestä vuorovaikutuksesta johtuva Coulombin potentiaali, joka on kääntäen verrannollinen varattujen kappaleiden etäisyyteen.

- Elektronia kuvaava aaltofunktio on $\psi(r) = y(r)/r$ ja siitä voidaan laskea elektronin todennäköisyysjakauma ytimen ympärillä.
- Huom. yhtälö (7) on erikoistapaus yhtälöstä (16)
- Aaltofunktion asymptoottiselle käyttäytymiselle voidaan johtaa seuraavat ehdot (käydään tarkemmin läpi kvanttimekaniikan kursseissa):
  a) lähellä origoa y käyttäytyy kuten $c_1 r$
  b) kaukana origosta (äärettömyydessä) y käyttäytyy kuten $c_2 e^{-\sqrt{-E}r}$ (huom. ominaisarvo E on negatiivinen – tämä johtuu Coulombin potentiaalin nollatason valinnasta)
  - Huom. numeerisissa ratkaisuissa ei voida mennä äärettömyyteen, ja origo on tässä tapauksessa singulaarinen piste, joten yo. reunaehdot voidaan tulkita siten, että $y(r_0) = c_1 r_0$, missä r_0 on jokin "pieni" muuttujan arvo, ja $y(r_\infty) = c_2 e^{-\sqrt{-E}r_\infty}$, missä r_∞ on "suuri".
- Vakiot c_1 ja c_2 määräytyvät *normituksesta* (ks. myöhemmin)
- Nämä reunaehdot määräävät käytännössä funktion arvon *ja sen derivaatan arvon* kahdessa pisteessä. Yleisesti tällaisella probleemalla ei ole ratkaisua (neljä ehtoa 2. asteen differentiaaliyhtälölle).
- Ratkaisuja on olemassa *vain tietyillä (diskreeteillä) parametrin E arvoilla.* Niitä E:n arvoja, joille ratkaisuja on olemassa, kutsutaan probleeman *ominaisarvoiksi.*
- Probleeman ratkaisu etenee seuraavasti (vrt. kahden pisteen reuna-arvoprobleeman ratkaisu):
- Jaetaan tarkasteltava väli kahtia: $I_0 = [r_0, r_i]$, $I_\infty = [r_i, r_\infty]$, missä r_i on "sopivasti valittu" *sovituspiste*
- *Arvataan E:lle jokin arvo*
- Ratkaistaan yhtälö (16) alueessa I_0 lähtien reunaehdosta a), ja alueessa I_∞ lähtien reunaehdoista b). Merkitään ratkaisuja y_0 ja y_∞.
- Jos ratkaisufunktiot y_0 ja y_∞ sopivat yhteen pisteessä r_i, kyseinen E:n arvo todellakin on yhtälön ominaisarvo.
  - Funktiot "sopivat yhteen", jos funktioiden arvot *ja niiden derivaattojen arvot* täsmäävät pisteessä r_1.

– Tämä ehto voidaan muotoilla seuraavasti:

$$\frac{y_0'(r_i)}{y_0(r_i)} = \frac{y_\infty'(r_i)}{y_\infty(r_i)},$$

mikä mahdollistaa sen, että em. vakiot c_1 ja c_2 voidaan valita mielivaltaisesti ratkaisun alussa. Toinen niistä määräytyy funktion y jatkuvuusehdosta, ja toinen lopuksi normituksesta.

- Jos y_0 ja y_∞ *eivät sovi yhteen*, arvataan uusi E:n arvo ja ratkaistaan yhtälö uudelleen. Ominaisarvon etsiminen voidaan toteuttaa esim. välinpuolitusmenetelmällä.

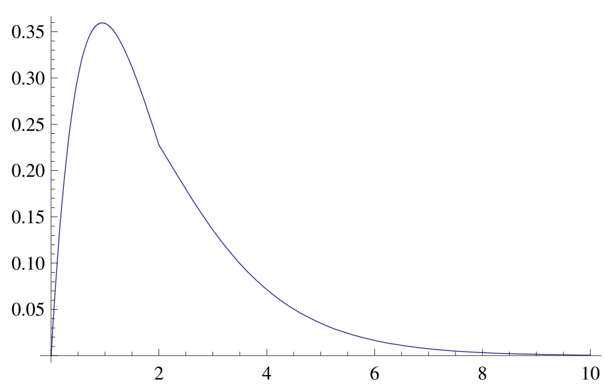
  – Ominaisarvon etsiminen voidaan tulkita funktion

  $$\Delta(E) = \frac{y_0'(r_i)}{y_0(r_i)} - \frac{y_\infty'(r_i)}{y_\infty(r_i)}$$

  nollakohdan etsimiseksi. Mathematicalla on kätevä niputtaa ratkaisuprosessi funktioksi.

```
In[735]:= ϵ1 = -0.9;
q0 = Flatten[NDSolve[{eq /. ϵ → ϵ1,
     y[r0] == f0[r0], y'[r0] == f0'[r0]},
    y[r], {r, r0, ri}]];
y0[r_] = y[r] /. q0;
q∞ = Flatten[NDSolve[{eq /. ϵ → ϵ1,
     y[r∞] == f∞[r∞] /. ϵ → ϵ1, y'[r∞] == f∞'[r∞] /. ϵ → ϵ1},
    y[r], {r, ri, r∞}]];
y∞[r_] = y[r] /. q∞;
Show[Plot[y0[r], {r, r0, ri}],
 (*skaalataan y∞[r], jotta kuvaaja olisi
  havainnollisempi *)
 Plot[y0[ri]/y∞[ri] y∞[r], {r, ri, r∞}],
 PlotRange → All, AxesOrigin → {0, 0}]
Print["Δ(ϵ1) = ", y0'[ri]/y0[ri] - y∞'[ri]/y∞[ri]]
```

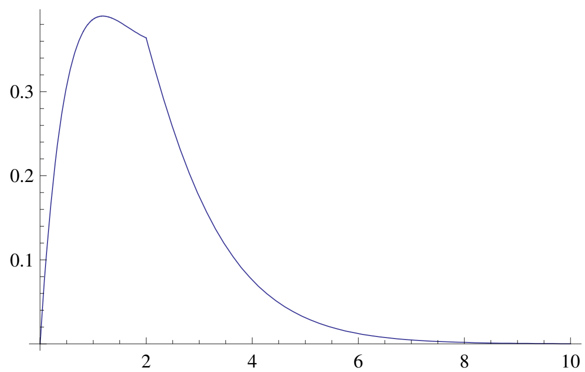
Out[740]=




```
Δ(ϵ1) = -0.381443
```

```
In[756]:= ϵ2 = -1.2;
          y0[r_] = y[r] /. Flatten[NDSolve[{eq /. ϵ → ϵ2,
                 y[r0] == f0[r0], y'[r0] == f0'[r0]},
               y[r], {r, r0, ri}]];
          y∞[r_] = y[r] /. Flatten[NDSolve[{eq /. ϵ → ϵ2,
                 y[r∞] == f∞[r∞] /. ϵ → ϵ2, y'[r∞] == f∞'[r∞] /. ϵ → ϵ2},
               y[r], {r, ri, r∞}]];
          Show[Plot[y0[r], {r, r0, ri}],
           Plot[y0[ri]/y∞[ri] y∞[r], {r, ri, r∞}],
           PlotRange → All, AxesOrigin → {0, 0}]
          Print["Δ(ϵ2) = ", y0'[ri]/y0[ri] - y∞'[ri]/y∞[ri]]
```

Out[759]=




```
Δ(ϵ2) = 0.57066
```

```
In[742]:= Δ[e_] := Module[{},
            q0 = Flatten[NDSolve[{eq /. ϵ → e,
                 y[r0] == f0[r0], y'[r0] == f0'[r0]},
               y[r], {r, r0, ri}]];
            y0[r_] = y[r] /. q0;
            q∞ = Flatten[NDSolve[{eq /. ϵ → e,
                 y[r∞] == f∞[r∞] /. ϵ → e, y'[r∞] == f∞'[r∞] /. ϵ → e},
               y[r], {r, ri, r∞}]];
            y∞[r_] = y[r] /. q∞;
            y0'[ri]/y0[ri] - y∞'[ri]/y∞[ri]]

In[761]:= Δ[-0.9]
          Δ[-1.2]

Out[761]= -0.381443

Out[762]= 0.57066
```

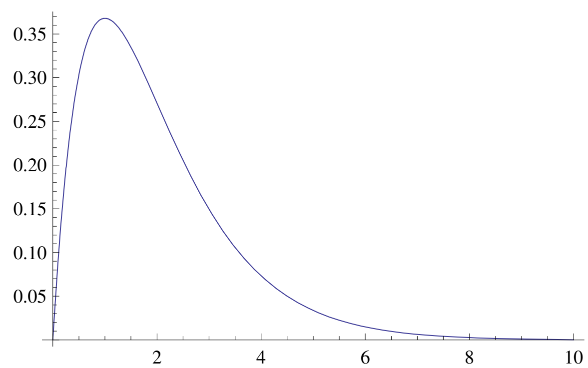
- Kun on löydetty kaksi alkuarvausta, E_1 ja E_2 joille $\Delta(E_1)$ ja $\Delta(E_2)$ ovat erimerkkiset, voidaan etsiä tarkempi ratkaisu välinpuolitusmenetelmällä.

```
In[763]:= e1 = -0.9; e2 = -1.2;
          i = 1; e[i] = (e1 + e2) / 2;

In[765]:= While[Abs[e1 - e2] > 10^-6,
           If[Δ[e[i]] Δ[e2] < 0, e1 = e[i], e2 = e[i]];
           i++;
           e[i] = (e1 + e2) / 2;]

In[767]:= e[i]

Out[767]= -1.
```

- Ratkaisufunktio y saadaan nyt ratkaisemalla yhtälö (16) E:n arvolla -1.

```
ϵ0 = -1.0;
y0[r_] = y[r] /. Flatten[NDSolve[{eq /. ϵ → ϵ0,
       y[r0] == f0[r0], y'[r0] == f0'[r0]},
      y[r], {r, r0, ri}]];
y∞[r_] = y[r] /. Flatten[NDSolve[{eq /. ϵ → ϵ0,
       y[r∞] == f∞[r∞] /. ϵ → ϵ2, y'[r∞] == f∞'[r∞] /. ϵ → ϵ0},
      y[r], {r, ri, r∞}]];
(* /; mahdollistaa paloittain määritellyn funktion
  määrittelyn, esim. f[x_]:=x^2/;x>0 *)
(* määrittely f[x_]:=x^2 pätee, kun /;-merkin jälkeen
  kirjoitettu ehto on voimassa *)
η[r_] := y0[r] /; (0 ≤ r && r < ri)
η[r_] := y0[ri]/y∞[ri] y∞[r] /; (ri ≤ r && r < r∞)
Show[Plot[η[r], {r, r0, r∞}],
 PlotRange → All, AxesOrigin → {0, 0}]
```

Out[779]=



- Normitus: (oletetaan, että aaltofunktion arvo ei riipu kulmamuuttujista)

$$\int_V |\psi(\mathbf{r})|^2 dV = \int_\Omega d\Omega \int_0^\infty r^2 |\psi(r)|^2 dr = 4\pi \int_0^\infty y(r)^2 dr = 1 \qquad (17)$$

Merkitään

$$y(r) = \begin{cases} c_1 y_0(r) & 0 \le r < r_i \\ c_2 y_\infty(r) & r_i \le r \le r_\infty \end{cases}$$

jolloin

$$\frac{1}{4\pi} = \int_{r_0}^{r_\infty} y(r)^2 dr = c_1^2 \int_{r_0}^{r_i} y_0(r)^2 dr + c_2^2 \int_{r_i}^{r_\infty} y_\infty(r)^2 dr$$
$$= c_1^2 \left(\int_{r_0}^{r_i} y_0(r)^2 dr + \frac{c_2^2}{c_1^2} \int_{r_i}^{r_\infty} y_\infty(r)^2 dr \right)$$
$$\equiv c_1^2 N$$

Vakioiden c_1 ja c_2 suhde määräytyy jatkuvuusehdosta: $c_1 y_0(r_i) = c_2 y_\infty(r_i)$, josta

$$\frac{c_2}{c_1} = \frac{y_0(r_i)}{y_\infty(r_i)}$$

Integraali N voidaan nyt laskea yo. yhtälöstä, jolloin

$$c_1 = \frac{1}{\sqrt{4\pi N}}$$

- Varsinainen aaltofunktio on

$$\psi(r) = \frac{y(r)}{r}$$

```
n = NIntegrate[(η[r])^2, {r, r0, r∞}];
g1 = Plot[η[r]/(Sqrt[4 π n] r), {r, r0, r∞},
   PlotRange → All, Filling → Axis];
g2 = Plot3D[η[Sqrt[x^2 + y^2]]/(Sqrt[4 π n] Sqrt[x^2 + y^2]), {x, -r∞, r∞},
   {y, -r∞, r∞}, RegionFunction →
    Function[{x, y}, x^2 + y^2 < r∞^2 && x^2 + y^2 > r0^2],
   PlotRange → {0, 0.5}];
GraphicsArray [{g1, g2}]
```

Out[812]=