

▼ 10. Elsőrendű differenciálegyenletek numerikus megoldása

Mérnöki, természettudományos és gazdasági folyamatok leírása és modellezése során számos esetben differenciálegyenletekhez jutunk. A differenciálegyenleteket csak kevés esetben oldhatjuk meg egzakt módon, a legtöbb esetben numerikus megoldásra van szükség.

A numerikus megoldásoknak két nagy csoportja van:

- **Analitikus módszernek** nevezzük azt a megoldás típust, amikor a közelítő megoldást függvény formájában adjuk meg.
- **Diszkrét módszer** esetén alappontrendszert veszünk fel, és a közelítő megoldást ezekben az alappontokban határozzuk meg.

Ebben a fejezetben közönséges, elsőrendű differenciálegyenletek adott kezdeti feltételt teljesítő partikuláris megoldásának keresésével foglalkozunk, vagyis az

$$y'(x)=f(x, y(x)), \quad y(x_0)=y_0$$

kezdetiérték-feladat megoldását keressük.

Ahhoz, hogy a közelítő függvények vagy pontsorozatok határértéke valóban a megoldást adja-e, szükséges azt tudnunk, létezik-e és egyértelmű-e a differenciálegyenlet megoldása. A megoldás létezését biztosítja, ha az $f(x, y)$ Lipschitz tulajdonságú.

▼ 10. 1. Definíció

Ha az $y'=f(x, y)$ egyenletben szereplő $f(x, y)$ függvény valamely $|x - x_0| < a \leq \infty$, $|y - y_0| < b \leq \infty$ tartományban korlátos ($|f(x, y)| \leq K$), folytonos és eleget tesz az $|f(x, y_2) - f(x, y_1)| \leq M|y_2 - y_1|$ feltételnek, ahol M pozitív állandó, akkor $f(x, y)$ **Lipschitz - tulajdonságú**.

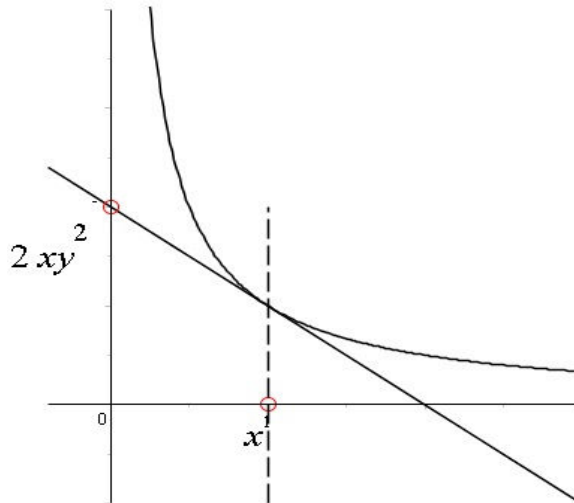
▼ 10. 2. Tétel

Picard-Lindelöf tétel: Ha $f(x, y)$ folytonos és y -ban Lipschitz tulajdonságú valamely $|x - x_0| < a \leq \infty$, $|y - y_0| < b \leq \infty$ tartományban, akkor az $y'=f(x, y)$, $y(x_0)=y_0$ kezdetiérték feladat az $[x_0 - \alpha, x_0 + \alpha]$ intervallumon egyértelműen megoldható, ahol $\alpha = \min(a, \frac{b}{K})$. ($K=0$ esetén $\frac{b}{K} = \infty$)

▼ 10. 3. Egzakt megoldás Maple-lel

Használt Maple parancsok: [DEtools csomag](#), [odeadvisor](#), [dfieldplot](#), [DEplot](#)

1. Példa Írjuk fel azoknak a görbéknek az egyenletét, amelyekre a következő állítás érvényes: a görbe bármely (x, y) pontjában húzható érintője az y tengelyből $2x \cdot y^2$ hosszúságú szakaszt metsz le.



Megoldás

A differenciálegyenlet felírásához használjuk fel, hogy egy görbe bármely pontjában az érintő iránytangense megegyezik az adott pontbeli derivált értékével. Az érintő tetszőleges pontja legyen (X, Y) , az érintési pont pedig $(x, y(x))$. Az érintő egyenes egyenlete:

> *restart; with(DEtools) : with(plots) :*

>
$$erinto := Y - y(x) = \left(\frac{d}{dx} y(x) \right) (X - x) :$$

A feladat differenciálegyenletét úgy kapjuk, hogy $Y=2 \cdot x \cdot y^2$, $X=0$ kifejezéseket helyettesítünk az érintő egyenletébe.

> *de := subs({X=0, Y=2 x y(x)^2}, erinto)*

$$de := 2 x y(x)^2 - y(x) = - \left(\frac{d}{dx} y(x) \right) x$$

Lehetőség van a differenciálegyenlet típusának meghatározására a Maple *odeadvisor* parancsa segítségével.

> *odeadvisor(de)*

[[_homogeneous, class D], _rational, _Bernoulli]

Tehát az egyenletünk homogén, Bernoulli típusú differenciálegyenlet. Az egzakt, általános megoldást megkapjuk a *dsolve* paranccsal.

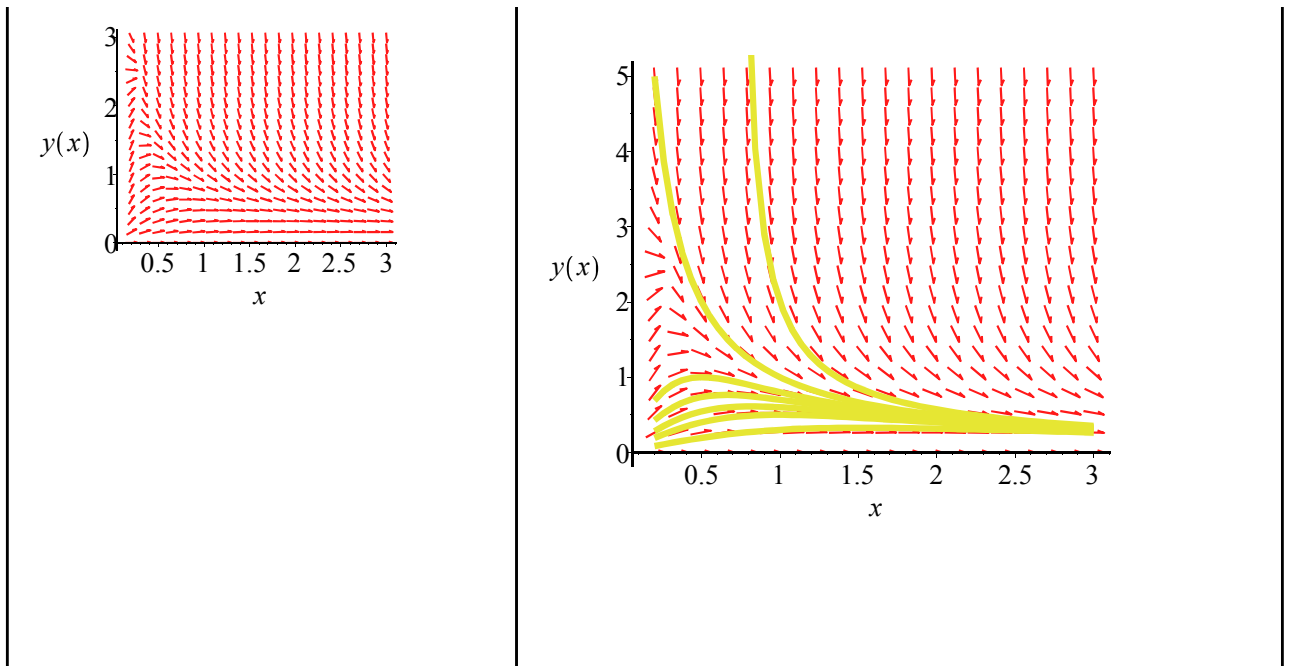
> *alt_mo := dsolve(de, y(x))*

$$alt_mo := y(x) = \frac{x}{x^2 + _C1}$$

Természetesen a megoldás egy integrációs konstanst tartalmaz. A *dfieldplot* parancs az iránymezőt, a *DEplot* különböző kezdeti feltételeknek megfelelő partikuláris megoldást és az iránymezőt mutatja.

> *dfieldplot(de, y(x), x=0.2..3, y=0..3)*

> *DEplot(de, y(x), x=0.2..3, y=0..5, {[1, 2], [1, 1], [1, 0.8], [1, 0.7], [1, 0.6], [1, 0.5], [1, 0.3]})*



Kezdeti feltételként követeljük meg még azt is, hogy a görbe haladjon át az (x_0, y_0) koordinátájú ponton, írjuk fel az ehhez a ponthoz tartozó érintőegyenest.

> $part_mo := dsolve(\{de, y(x_0) = y_0\}, y(x))$

$$part_mo := y(x) = \frac{x y_0}{x^2 y_0 + x_0 - y_0 x_0^2}$$

> $erinto := Y = subs\left(x = x_0, \frac{\partial}{\partial x} rhs(part_mo)\right) (X - x_0) + y_0$

$$erinto := Y = \left(-2 y_0^2 + \frac{y_0}{x_0}\right) (X - x_0) + y_0$$

Ellenőrizzük, igaz-e, hogy ez az érintő az y tengelyből $2x_0 y_0^2$ hosszúságú szakaszt metsz ki.

> $ellenorzes := simplify(subs(X = 0, erinto))$

$$ellenorzes := Y = 2 y_0^2 x_0$$

Az $(1, 2)$ kezdeti feltételnek megfelelő partikuláris megoldás

> $part_mol := subs(\{x_0 = 1, y_0 = 2\}, part_mo)$

$$part_mol := y(x) = \frac{2x}{2x^2 - 1}$$

Ennek tetszőleges x_1 abszcisszájú pontjában az érintő:

> $meredekseg := simplify\left(subs\left(x = x_1, \frac{\partial}{\partial x} rhs(part_mol)\right)\right) : y_1 := subs(x = x_1, rhs(part_mol)) :$

> $erinto1 := y = meredekseg (x - x_1) + y_1$

$$erinto1 := y = -\frac{2(2x_1^2 + 1)(x - x_1)}{(2x_1^2 - 1)^2} + \frac{2x_1}{2x_1^2 - 1}$$

Írjunk animációt, amely az $P(1, 2)$ kezdeti feltételnek megfelelő partikuláris megoldás különböző pontjaiba húzott érintőt és annak y tengellyel való metszéspontja koordinátáit rajzolja ki.

> **for** i **to** 5 **do**

$x1 := i - 3$;

$y1 := \text{subs}(x = x1, \text{rhs}(\text{part_mol}))$;

$\text{meredekseg} := \text{simplify}\left(\text{subs}\left(x = x1, \frac{\partial}{\partial x} \text{rhs}(\text{part_mol})\right)\right)$;

$\text{erinto}[i] := \text{meredekseg}(x - x1) + y1$;

$\text{abra} := \text{plot}(\text{erinto}[i], x = -5..7, \text{thickness} = 2, \text{color} = \text{blue})$;

$\text{metsz} := 2 \cdot x1 \cdot y1^2$;

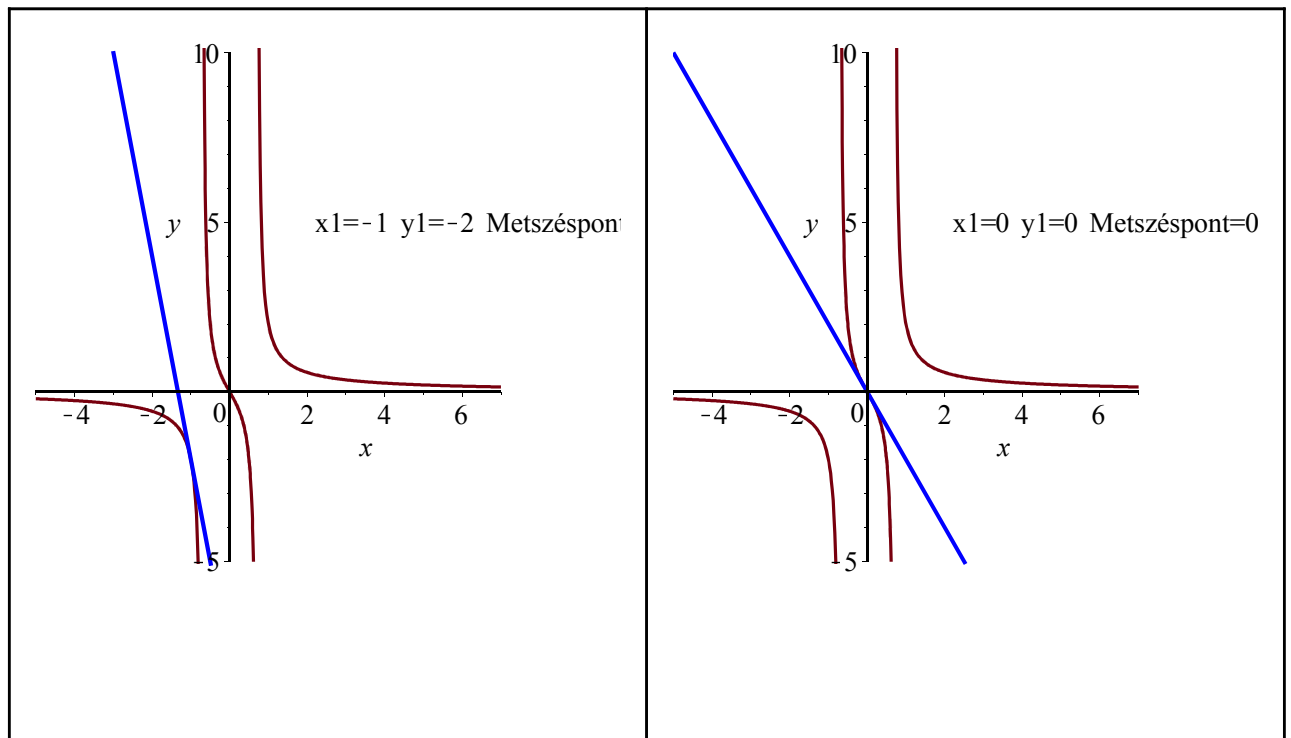
$H := \text{textplot}([2, 5, \text{typeset}("x1=", x1, " ", "y1=", y1, " ", \text{"Metszéspont="}, \text{metsz})], \text{align} = \text{RIGHT})$;

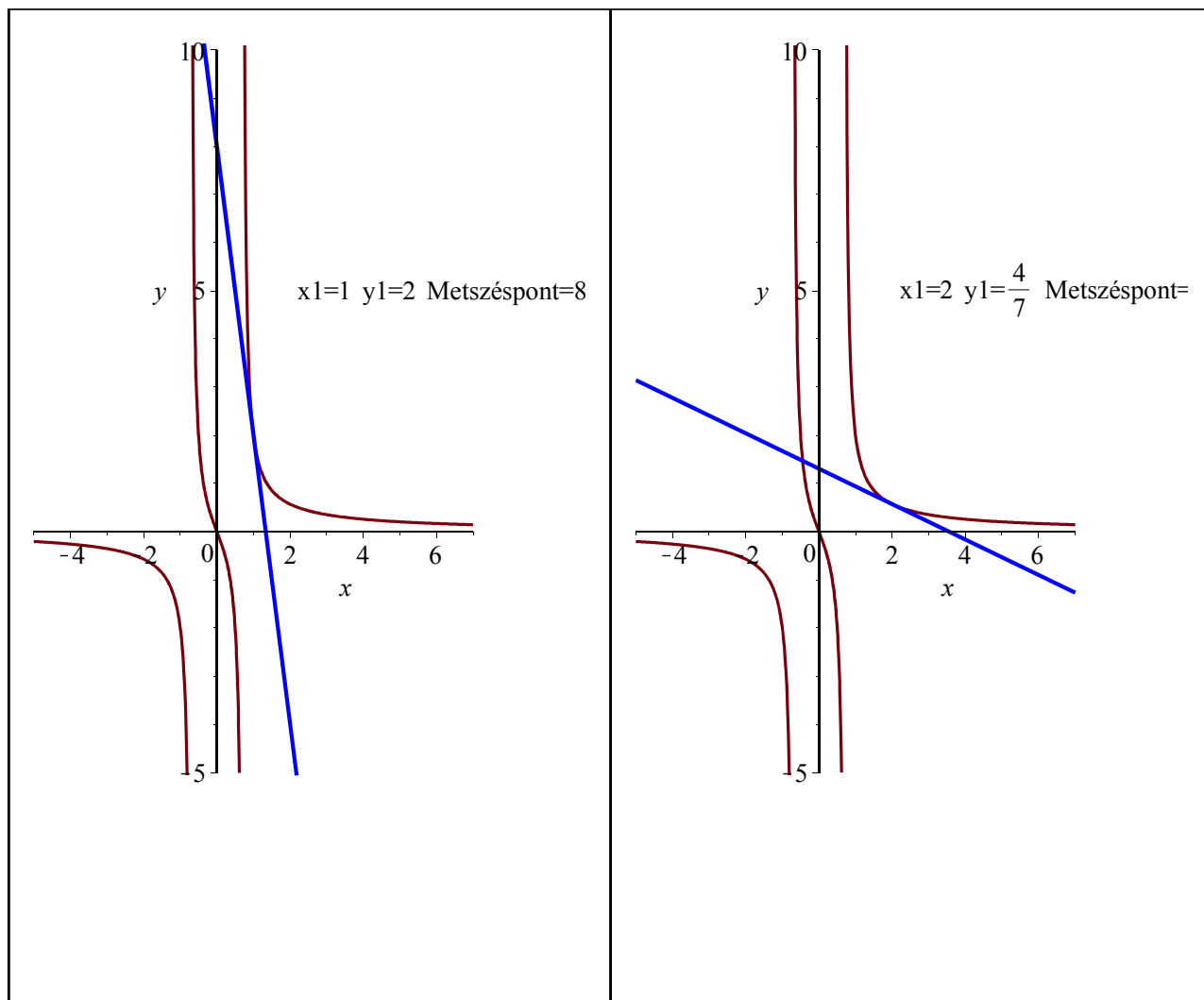
$b := \text{plot}(\text{rhs}(\text{part_mol}), x = -5..7, y = -5..10, \text{discont} = \text{true})$;

$\text{kep}[i] := \text{display}([b, H, \text{abra}])$

end do;

> $\text{abra} := \text{Matrix}(2, 2, [\text{seq}(\text{kep}[i], i = 2..5)]) : \text{plots}[\text{display}](\text{abra})$





▼ 10. 4. Analitikus megoldás

▼ 10. 4. 1. Picard iterációs módszere

A módszer lényege, hogy a differenciálegyenletet integrálegyenletté alakítjuk, és az integrálba y_0 értéket helyettesítve kapjuk y_1 -et, majd y helyébe y_1 -et beírva kapjuk y_2 -t, és így tovább. (Jelöljük a független változót t -vel, mert a legtöbb feladatban a független változó az idő.)

> *restart; with(DEtools) : with(plots) :*

>
$$de := \frac{d}{dt} y(t) = f(t, y(t)) :$$

Mindkét oldalt integrálva, felhasználva hogy $y(t_0)=y_0$

>
$$y_1(t) = y_0 + \int_{t_0}^t f(t, y_0) dt :$$

>
$$y_2(t) = y_0 + \int_{t_0}^t f(t, y_1(t)) dt :$$

>
$$y_3(t) = y_0 + \int_{t_0}^t f(t, y_2(t)) dt :$$

így egy függvénysorozatot kapunk, melynek határértéke (ha létezik) adja a differenciálegyenlet partikuláris megoldását.

2. Példa Tekintsük a következő differenciálegyenletet és oldjuk meg Picard módszerével.

$$> \quad de := \frac{d}{dt} y(t) = y(t) + t : \quad t0 := 0 : \quad y0 := 1 :$$

Megoldás

Helyettesítsünk az egyenlet jobb oldalába $y(t)=y0$ értéket., majd a kapott kifejezést integráljuk a $[t0, t]$ intervallumon és az eredményt adjuk hozzá $y0$ -hoz..

$$> \quad f1 := subs(y(t) = y0, rhs(de))$$

$$f1 := 1 + t$$

$$> \quad elso_kozelites := y1(t) = y0 + \int_{t0}^t f1 \, dt$$

$$elso_kozelites := y1(t) = 1 + t + \frac{1}{2} t^2$$

Folytassuk az eljárást:

$$> \quad f2 := subs(y(t) = rhs(elso_kozelites), rhs(de)); masodik_kozelites := y2(t) = y0 + \int_{t0}^t f2 \, dt$$

$$f2 := 1 + 2t + \frac{1}{2} t^2$$

$$masodik_kozelites := y2(t) = 1 + t + t^2 + \frac{1}{6} t^3$$

Az eljárást *for* ciklusba rendezve adjuk meg a közelítő függvénysorozat 8. tagját.

$$> \quad n := 8; f[1] := subs(y(t) = y0, rhs(de)) : k[1] := y_1(t) = y0 + \int_{t0}^t f[1] \, dt :$$

for i from 2 to n do

$$f[i-1] := subs(y(t) = rhs(k[i-1]), rhs(de)); k[i] := y_i(t) = y0 + \int_{t0}^t f[i-1] \, dt$$

end do:

$$n := 8$$

$$> \quad print(k[i-1])$$

$$y_8(t) = 1 + t + t^2 + \frac{1}{3} t^3 + \frac{1}{12} t^4 + \frac{1}{60} t^5 + \frac{1}{360} t^6 + \frac{1}{2520} t^7 + \frac{1}{20160} t^8 + \frac{1}{362880} t^9$$

A differenciálegyenlet direkt módon is megoldható:

$$> \quad mo := dsolve(\{de, y(0) = 1\}, y(t))$$

$$mo := y(t) = -1 - t + 2 e^t$$

Azt sejtjük, hogy a közelítő megoldás hasonlít az egzakt megoldás Taylor polinomjához. A bizonyításhoz képezzük az egzakt megoldás 9. Taylor polinomját.

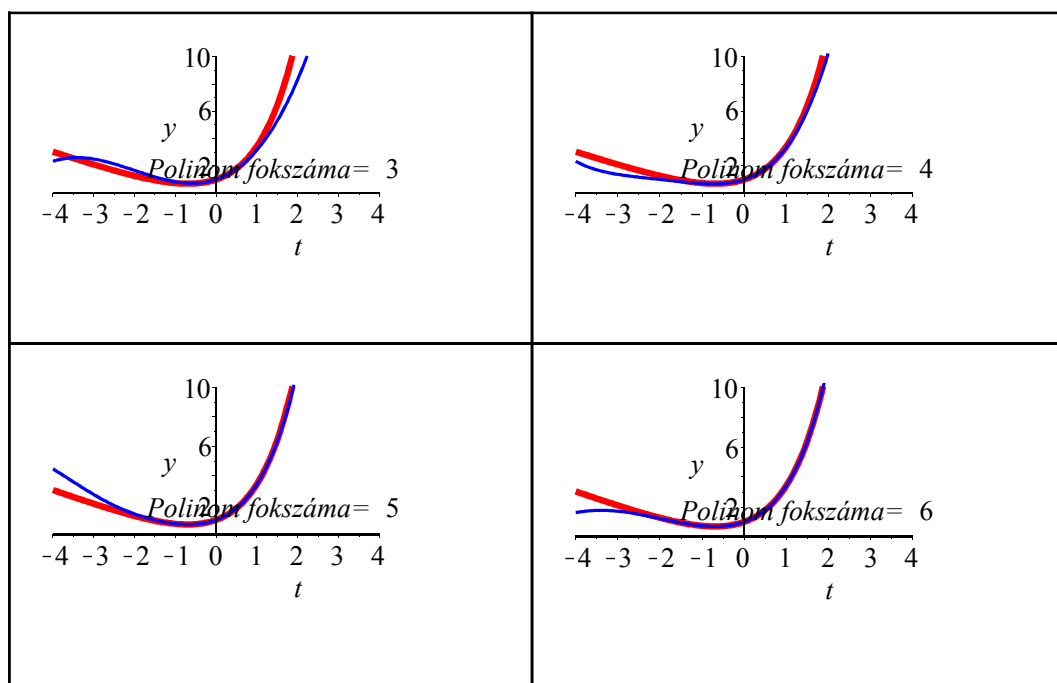
$$> \quad Taylor_polinom := convert(series(rhs(mo), t=0, 10), polynom)$$

$$Taylor_polinom := 1 + t + t^2 + \frac{1}{3} t^3 + \frac{1}{12} t^4 + \frac{1}{60} t^5 + \frac{1}{360} t^6 + \frac{1}{2520} t^7 + \frac{1}{20160} t^8 + \frac{1}{181440} t^9$$

Csak az első néhány tagig látható egyezés.

Készítsünk animációt, ami az egzakt megoldást és az iteráció elemeit ábrázolja a közelítő függvények ismeretében.

```
> Picard := proc(mo, a, b, c, d, N, Sx, Sy)
    local H, i, szam, S, P, F;
    global kep;
    H := textplot([Sx, Sy, Polinom fokszáma=], align = LEFT);
    F := plot(mo, t = a..b, y = c..d, color = red, thickness = 3);
    for i to N do
        szam := convert(i + 1, string);
        S := textplot([Sx, Sy, szam], align = RIGHT);
        P := plot(rhs(k[i]), t = a..b, color = BLUE);
        kep[i] := display([F, P, H, S]);
    end do
end proc;
> Picard(rhs(mo), -4, 4, 0, 10, 8, 4, 1.7) :
> abra := Matrix(2, 2, [seq(kep[i], i = 2..5)]) : plots[display](abra)
```



▼ 10. 4. 2. Taylor-sor módszer

A módszer lényege, hogy a differenciálegyenlet ismeretlen függvényét a Taylor sorával helyettesítjük.

```
> restart; with(DEtools) : with(plots) :
```

```
> de :=  $\frac{d}{dt} y(t) = f(t, y(t)) :$ 
```

A kezdeti feltétel

```
> kezdeti :=  $y(t_0) = y_0 :$ 
```

Képezzük az $y(t)$ t_0 körüli Taylor sorát.

```
> sor :=  $y(t) = \text{taylor}(y(t), t = t_0, 5)$ 
```

```
sor :=  $y(t) = y(t_0) + D(y)(t_0) (t - t_0) + \frac{1}{2} D^{(2)}(y)(t_0) (t - t_0)^2 + \frac{1}{6} D^{(3)}(y)(t_0) (t - t_0)^3 + \dots$ 
```

$$-t0)^3 + \frac{1}{24} D^{(4)}(y)(t0) (t-t0)^4 + O((t-t0)^5)$$

A differenciálegyenletbe $t=t0$ helyettesítéssel a $D(y)(t0)$ (a függvény első deriváltja a $t0$ helyen) kifejezhető. Ezt helyettesítsük be a *sor*-ba.

$$\begin{aligned} &> \text{convert}(de, D) : der1 := \text{subs}(t=t0, kezdeti, \%); sor := \text{subs}(der1, sor); \\ &\quad der1 := D(y)(t0) = f(t0, y0) \end{aligned}$$

$$\begin{aligned} sor := y(t) = &y(t0) + f(t0, y0) (t-t0) + \frac{1}{2} D^{(2)}(y)(t0) (t-t0)^2 + \frac{1}{6} D^{(3)}(y)(t0) (t-t0)^3 \\ &+ \frac{1}{24} D^{(4)}(y)(t0) (t-t0)^4 + O((t-t0)^5) \end{aligned}$$

A második derivált értékét a differenciálegyenlet deriválása révén kapjuk meg. A deriválás után helyettesítsük be $t=t0$ -t, a kezdeti feltételt és a derivált $t=t0$ helyen vett értékét. A harmadik deriváltat az eredeti egyenlet második deriváltjának $t=t0$ helyen vett értékéből kapjuk és így tovább.

3. Példa Oldjuk meg a következő differenciálegyenletet a Taylor sor módszerrel

$$\frac{d}{dt} y(t) = (2-t) y(t), \quad y(0) = 1$$

Megoldás:

$$> de := \frac{d}{dt} y(t) = (2-t) y(t) : kezdeti := y(0) = 1 :$$

$$> sor := y(t) = \text{taylor}(y(t), t=0, 4) : sor := \text{subs}(kezdeti, sor)$$

$$sor := y(t) = 1 + D(y)(0) t + \frac{1}{2} D^{(2)}(y)(0) t^2 + \frac{1}{6} D^{(3)}(y)(0) t^3 + O(t^4)$$

$$\begin{aligned} > \text{convert}(de, D) : derivalt1 := \text{subs}(t=0, kezdeti, \%); sor := \text{subs}(derivalt1, sor) \\ &\quad derivalt1 := D(y)(0) = 2 \end{aligned}$$

$$sor := y(t) = 1 + 2 t + \frac{1}{2} D^{(2)}(y)(0) t^2 + \frac{1}{6} D^{(3)}(y)(0) t^3 + O(t^4)$$

A második derivált:

$$\begin{aligned} > de2 := \frac{d}{dt} (de); derivalt2 := \text{subs}(t=0, kezdeti, derivalt1, \text{convert}(de2, D)); \\ &\quad sor := \text{subs}(derivalt2, sor) \end{aligned}$$

$$de2 := \frac{d^2}{dt^2} y(t) = -y(t) + (2-t) \left(\frac{d}{dt} y(t) \right)$$

$$derivalt2 := D^{(2)}(y)(0) = 3$$

$$sor := y(t) = 1 + 2 t + \frac{3}{2} t^2 + \frac{1}{6} D^{(3)}(y)(0) t^3 + O(t^4)$$

A harmadik derivált

$$> de3 := \frac{d}{dt} (de2);$$

$$\begin{aligned} derivalt3 := &\text{subs}(t=0, kezdeti, derivalt1, derivalt2, \text{convert}(de3, D)); sor := \\ &\text{subs}(derivalt3, sor) \end{aligned}$$

$$de3 := \frac{d^3}{dt^3} y(t) = -2 \left(\frac{d}{dt} y(t) \right) + (2-t) \left(\frac{d^2}{dt^2} y(t) \right)$$

$$\text{derivalt3} := D^{(3)}(y)(0) = 2$$

$$\text{sor} := y(t) = 1 + 2t + \frac{3}{2}t^2 + \frac{1}{3}t^3 + O(t^4)$$

Konvertáljuk a kapott eredményt polinommá

> `kozelito_megoldas := convert(sor, polynom)`

$$\text{kozelito_megoldas} := y(t) = 1 + 2t + \frac{3}{2}t^2 + \frac{1}{3}t^3$$

A Taylor soros közelítő megoldás a Maple `dsolve` parancsának `type=series` opciójával is előállítható.

> `Order := 4 : convert(dsolve( {de, kezdeti}, y(t), type=series), polynom)`

$$y(t) = 1 + 2t + \frac{3}{2}t^2 + \frac{1}{3}t^3$$

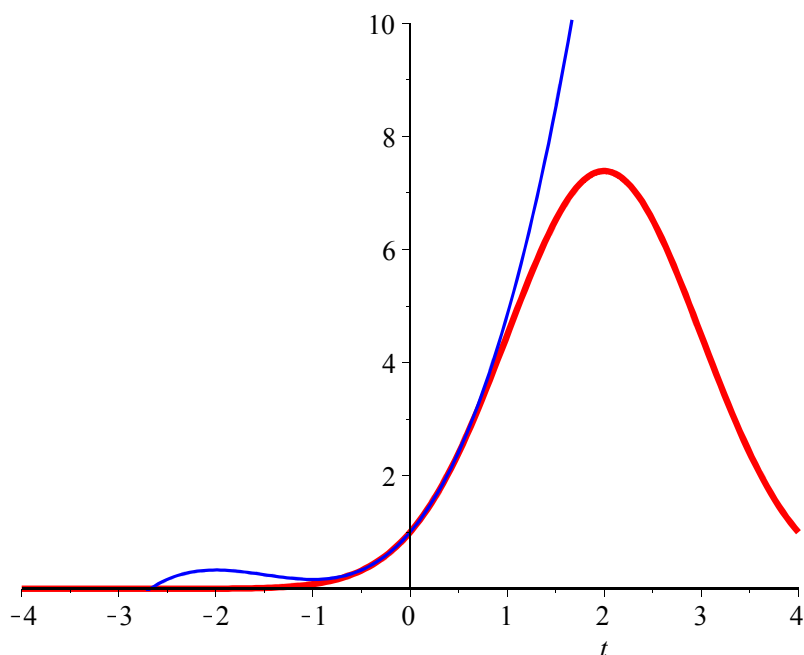
A példában szereplő differenciálegyenletnek létezik egzakt megoldása is. Ábrázoljuk az egzakt és a közelítő megoldást közös kooordináta rendszerben

> `mo := dsolve( {de, kezdeti}, y(t) )`

$$\text{mo} := y(t) = e^{-\frac{1}{2}t(-4+t)}$$

> `a := plot(rhs(mo), t=-4..4, color=red, thickness=3) :`

`b := plot(rhs(kozelito_megoldas), t=-4..4, y=0..10, color=blue) : display([a, b])`



Az ábrából látható, hogy a közelítő polinom csak a kezdeti feltétel "elég kis környezetében" közelíti meg az egzakt megoldást.

▼ 10. 5. Megoldás diszkrét módszerekkel

▼ 10. 5. 1. Euler módszere

A megoldandó differenciálegyenletet **differenciaegyenlettel** helyettesítjük. Adott $t_0, y(t_0)$ értékek ismeretében h lépésközönként az $[a, b]$ intervallumban adjuk meg a keresett

függvényértékeket. Az $[a, b]$ intervallumot n egyenlő részre osztjuk, a részintervallumok hosszát jelöljük h -val.

$$\begin{aligned} \frac{d}{dt} y(t) &= f(t, y(t)), \quad y(t_0) = y_0, \quad t_0 = a \\ \frac{y(t_0 + h) - y(t_0)}{h} &\approx f(t_0, y(t_0)) \\ y(t_0 + h) &\approx y(t_0) + h f(t_0, y(t_0)) \\ y(t_0 + 2h) &\approx y(t_0 + h) + h f(t_0 + h, y(t_0 + h)) \\ &\vdots \\ y(t_0 + nh) &\approx y(t_0 + (n-1)h) + h f(t_0 + (n-1)h, y(t_0 + (n-1)h)), \quad t_0 + nh = b \end{aligned}$$

4. Példa Adjuk meg a $\frac{d}{dt} y(t) = 2 \cdot t \cdot y(t)$, $y(0) = 1$ differenciálegyenlet közelítő megoldását a $[0, 1]$ intervallumon Euler módszerrel.

Megoldás

Osszuk fel az intervallumot először 5, majd 20 egyenlő részre, ábrázoljuk az egzakt és a numerikus megoldásokat.

> *restart*; *de* := $\frac{d}{dt} y(t) = 2 \cdot t \cdot y(t)$: *kezdeti* := $y(0) = 1$: *a* := 0 : *b* := 1 :

5 részre osztás esetén

> $n := 5$: $h := \frac{b-a}{n}$:

A kapott független változó és függvényértékek sorozatát rendezzük egy-egy halmazba.

> $t[0] := 0$: $y[0] := 1$:
for i **to** n **do**
 $t[i] := t[i-1] + h$;
 $y[i] := \text{evalf}(y[i-1] + 2 \cdot h \cdot t[i-1] \cdot y[i-1])$
end do:

> $T5 := [\text{seq}(t[i], i=0..n)]$; $Y5 := [\text{seq}(y[i], i=0..n)]$;

$$T5 := \left[0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1 \right]$$

$$Y5 := [1, 1., 1.080000000, 1.252800000, 1.553472000, 2.050583040]$$

20 részre osztás esetén

> $n := 20$: $h := \frac{b-a}{n}$:

> $t[0] := 0$: $y[0] := 1$:
for i **to** n **do**
 $t[i] := t[i-1] + h$;
 $y[i] := \text{evalf}(y[i-1] + 2 \cdot h \cdot t[i-1] \cdot y[i-1])$
end do:

> $T20 := [\text{seq}(t[i], i=0..n)]$; $Y20 := [\text{seq}(y[i], i=0..n)]$;

$$T20 := \left[0, \frac{1}{20}, \frac{1}{10}, \frac{3}{20}, \frac{1}{5}, \frac{1}{4}, \frac{3}{10}, \frac{7}{20}, \frac{2}{5}, \frac{9}{20}, \frac{1}{2}, \frac{11}{20}, \frac{3}{5}, \frac{13}{20}, \frac{7}{10}, \frac{3}{4}, \frac{4}{5}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, 1 \right]$$

```
Y20 := [1, 1., 1.005000000, 1.015050000, 1.030275750, 1.050881265, 1.077153297,
1.109467896, 1.148299272, 1.194231243, 1.247971649, 1.310370231, 1.382440594,
1.465387030, 1.560637187, 1.669881790, 1.795122924, 1.938732758, 2.103525042,
2.292842296, 2.510662314]
```

A differenciálegyenlet egzakt megoldása

```
> t := 't': y := 'y': mo := dsolve( {de, kezdeti}, y(t) )
```

$$mo := y(t) = e^{t^2}$$

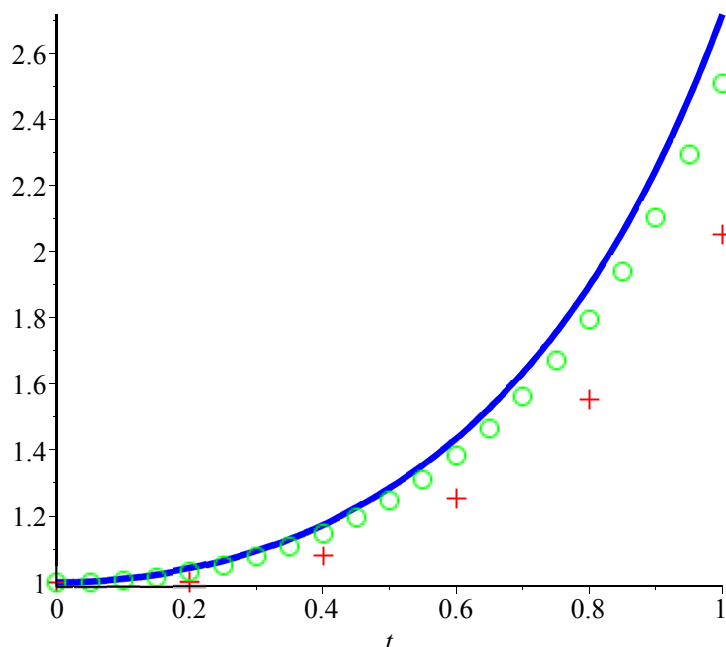
Ábrázoljuk az egzakt megoldást és a kétféle módon kapott numerikus megoldásokat egy közös koordináta rendszerben.

```
> pontok5 := plot( [ seq( [ T5p, Y5i ], i = 1 .. 6 ) ], style = point, symbolsize = 20, symbol
= cross, color = red ) :
```

```
> pontok20 := plot( [ seq( [ T20p, Y20i ], i = 1 .. 21 ) ], style = point, symbolsize = 20, symbol
= circle, color = green ) :
```

```
> abra := plot( rhs(mo), t = 0 .. 1, color = blue, thickness = 3 ) :
```

```
> plots[display]( [ abra, pontok5, pontok20 ] )
```



A körök a kisebb, a keresztek a nagyobb lépésközöket jelentik.

▼ 10. 5. 2. Heun módszer (prediktor - korrektor módszer)

A differenciálegyenlet $t_l = t_0 + h$ helyen vett megoldásának megadásához most induljunk ki az

$$\int_{t_0}^{t_l} y'(t) dt = \int_{t_0}^{t_l} f(x, y) dt$$

egyenlősből. A jobb oldalon álló integrál értékét trapéz módszerrel közelítjük, a bal oldali integrál $y(t1) - y(t0)$ értékű.

$$y(t1) - y(t0) \approx \frac{h}{2} (f(t0, y(t0)) - f(t1, y(t1)))$$

$y(t1)$ -re egy egyismeretlenes egyenletet kapunk, amit iterációval oldunk meg. Felhasználjuk az Euler módszernél mondottakat, vagyis hogy $y(t1) \approx y(t0) + hf(t0, y(t0))$. (Előzetes közelítés: **prediktor**). Ezt a jobboldalba helyettesítve:

$$y(t1) \approx y(t0) + \frac{h}{2} (f(t0, y(t0)) - f(t1, y(t0) + hf(t0, y(t0))))$$

Ezt a formát javított közelítésnek, **korrektor**nak nevezzük.

5. Példa Adjuk meg a 4. Példa $\frac{d}{dt} y(t) = 2 \cdot t \cdot y(t)$, $y(0) = 1$ differenciálegyenletének közelítő megoldását a $[0,1]$ intervallumon Heun módszerével, az intervallum 5 egyenlő részre osztásával.

Megoldás

A prediktor értékeit az előző feladat $Y5$ halmazának elemei tartalmazzák. Csak a korrektorokat kell meghatározni.

```
> T := [0, 1/5, 2/5, 3/5, 4/5, 1]; Y := [1, 1., 1.080000000, 1.252800000, 1.553472000,
      2.050583040];
```

```
> n := 5 : h := 0.2 : t[0] := 0 : y[0] := 1 : ym[0] := 1 :
  for i to n do
```

```
    t[i] := t[i-1] + h;
```

```
    y[i] := evalf(y[i-1] + 2 h t[i-1] y[i-1]);
```

```
    ym[i] := evalf(y[i-1] + (h (2 t[i-1] y[i-1] + 2 t[i] y[i])) / 2)
```

```
  end do;
```

```
> YM := [seq(ym[i], i = 0 .. n)]
```

$$T := \left[0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\right]$$

$$Y := [1, 1., 1.080000000, 1.252800000, 1.553472000, 2.050583040]$$

$$YM := [1, 1.040000000, 1.126400000, 1.316736000, 1.651691520, 2.212144128]$$

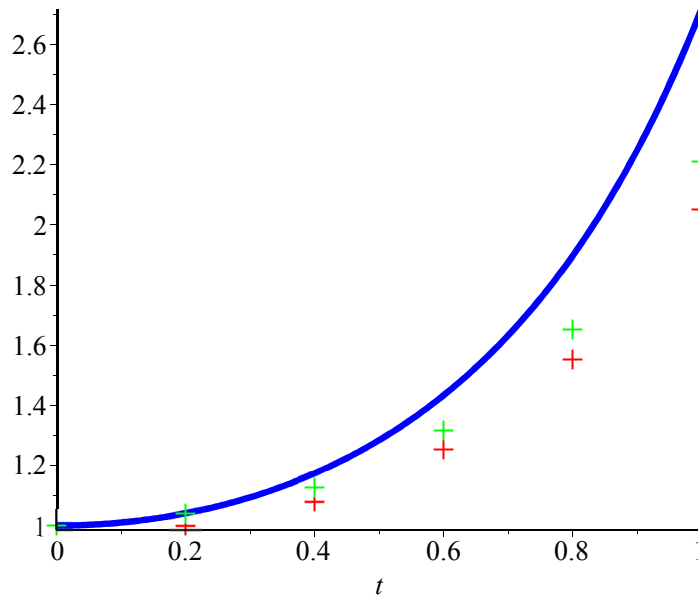
Ábrázoljuk az így kapott pontokat, az előző megoldás pontjait és az egzakt megoldást közös koordináta rendszerben.

```
> pontokE := plot([seq([T_i, Y_i], i = 1 .. 6)], style = point, symbolsize = 20, symbol = cross,
  color = red) :
```

```
> pontokH := plot([seq([T_i, YM_i], i = 1 .. 6)], style = point, symbolsize = 20, symbol
  = cross, color = green) :
```

```
> abra := plot(rhs(mo), t = 0 .. 1, color = blue, thickness = 3) :
```

```
> plots[display]([abra, pontokE, pontokH])
```



Láthatóan jobb lett a közelítés.

▼ 10. 5. 3. A Runge-Kutta módszer

A módszer – ellentétben például a Taylor soros módszerrel – a kezdeti értéktől távoli pontokban is elfogadható közelítéseket ad. Többféle Runge-Kutta módszer ismeretes, a közös kiindulópont az, hogy az y megoldásfüggvény x_{n+1} és x_n pontbeli értékének különbségét

$$y_{n+1} - y_n = \sum_{i=1}^m \omega_i k_i$$

alakban fejezi ki, ahol ω_i mennyiségek állandók, k_i -k pedig a h lépésközből és a derivált különböző helyeken vett értékeiből meghatározhatók. A leggyakrabban az alábbi negyedrendű Runge-Kutta módszert használják. A

$$\frac{d}{dt} y(t) = f(t, y(t))$$

differenciálegyenlet megoldásaihoz tartozó függvényértékeket h lépésközönként

$$y(t+h) = y(t) + \frac{k1 + 2 k2 + 2 k3 + k4}{6}$$

formában keressük, ahol

$$\begin{aligned} k1 &= h f(t, y(t)) \\ k2 &= h f\left(t + \frac{h}{2}, y(t) + \frac{k1}{2}\right) \\ k3 &= h f\left(t + \frac{h}{2}, y(t) + \frac{k2}{2}\right) \\ k4 &= h f(t+h, y(t) + k3) \end{aligned}$$

A MAPLE tartalmazza a Runge Kutta eljárás algoritmusát, csak meg kell adni, melyik helyeken, illetve milyen tartományban keressük a megoldást. A `dsolve` parancs `numeric, method=rkf45` opciójával kaphatjuk meg a megoldást.

5. Példa Oldjuk meg a $\frac{d}{dt} y(t) = y(t) - t^2, y(0) = -1$ differenciálegyenletet negyedrendű Runge-Kutta módszerrel. Adjuk meg a megoldást $h=0.2$ lépésközzel a $[0, 2]$ intervallumon.

Megoldás

> *restart; with(DETools) : with(plots) :*

> $de := \frac{d}{dt} y(t) = y(t) - t^2 :$

> $mo := dsolve(\{de, y(0) = -1\}, y(t))$

$$mo := y(t) = 2 + 2t + t^2 - 3e^t$$

A *dsolve* parancs *output=array* opciója a megadott független változó értékekhez számítja ki a függvényértékeket.

> $mo_rkf45 := dsolve(\{de, y(0) = -1\}, y(t), numeric, method = rkf45, output = array([0, 0.4, 0.8, 1.2, 1.6, 2]))$

$$mo_rkf45 := \left[\begin{array}{c} \left[\begin{array}{cc} t & y(t) \end{array} \right] \\ \left[\begin{array}{cc} 0. & -1. \\ 0.4000000000000000 & -1.51547390400416 \\ 0.8000000000000000 & -2.43662243022899 \\ 1.2000000000000000 & -4.12034992221095 \\ 1.6000000000000000 & -7.09909529644821 \\ 2. & -12.1671653053448 \end{array} \right] \end{array} \right]$$

Ha egy teljes tartományra akarjuk az értékeket kiszámítani, az *output=piecewise, range=t0..tv* opciókat kell megadni. Ebből könnyen kiolvashatjuk a megfelelő helyekhez tartozó függvényértékeket.

> $t0 := 0 : tv := 2 : n := 10 : h := \frac{(tv - t0)}{n} :$

> $mo_rkf45 := dsolve(\{de, y(0) = -1\}, y(t), numeric, method = rkf45, output = piecewise, range = t0..tv) :$

> $t[0] := t0 : \text{for } i \text{ from } 0 \text{ to } n \text{ do}$
 $y[i] := rhs(mo_rkf45_2 \Big|_{t=t[i]}) ;$
 $t[i+1] := t[i] + h$
end do;

> $T := [seq(t[i], i = 0..n)] ; Y := [seq(y[i], i = 0..n)]$

$$T := \left[0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, \frac{6}{5}, \frac{7}{5}, \frac{8}{5}, \frac{9}{5}, 2 \right]$$

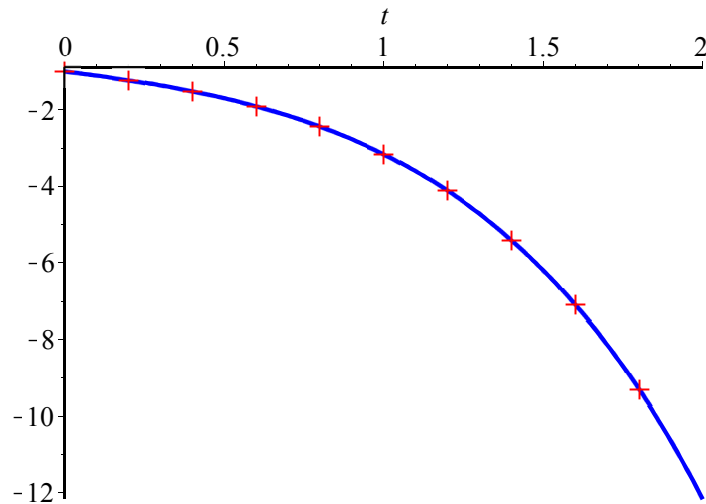
$$Y := [-1.0000000000000000, -1.22420815391229, -1.51547390400416, -1.90635612501762, -2.43662243022899, -3.15484500712383, -4.12034992221095, -5.40559849824449, -7.09909529644821, -9.30894002947776, -12.1671653053449]$$

Együttes ábrázolás:

> $pontok := plot([seq([T[i], Y[i]], i = 1..n)], style = point, symbolsize = 20, symbol = cross, color = red) :$

> $abra := plot(rhs(mo), t = 0..2, color = blue, thickness = 2) :$

> $display([abra, pontok])$



A numerikus és az egzakt megoldás gyakorlatilag nem tér el egymástól.

▼ 10. 6. Egy mérnöki feladat

6. Példa Tervezzük meg annak a patakmedernek az alakját (a szelvényét), amelynél adott a patakban folyó víz középsebessége, a vízfelszín esése, a patak medrének érdességi tényezője. A feladatot a következő differenciálegyenlet modellezi:

$$\frac{d}{dx} y(t) = \sqrt{0.66 y(t)^2 - 1}, y(0) = -1.5$$

A feladatot oldjuk meg a sorfejtéssel, Euler módszerrel és a Runge-Kutta módszerrel is.

Megoldás

> *restart; with(DETools) : with(plots) : de := $\frac{d}{dt} y(t) = \sqrt{0.66 y(t)^2 - 1}$:*

Próbáljuk megadni az egzakt megoldást.

> *mo := dsolve({de, y(0) = -1.5}, y(t)) : allvalues(%);*

$$y(t) = \frac{1}{66} \frac{\sqrt{33} \left(-100 - 197 \left(e^{\frac{1}{10} \sqrt{33} \sqrt{2} t} \right)^2 + 3 \left(e^{\frac{1}{10} \sqrt{33} \sqrt{2} t} \right)^2 \sqrt{33} \sqrt{97} \right)}{e^{\frac{1}{10} \sqrt{33} \sqrt{2} t} (3 \sqrt{33} - \sqrt{97})}$$

A kapott egzakt megoldás annyira bonyolult, hogy számításokhoz mindenképpen numerikus megoldást kell keresnünk.

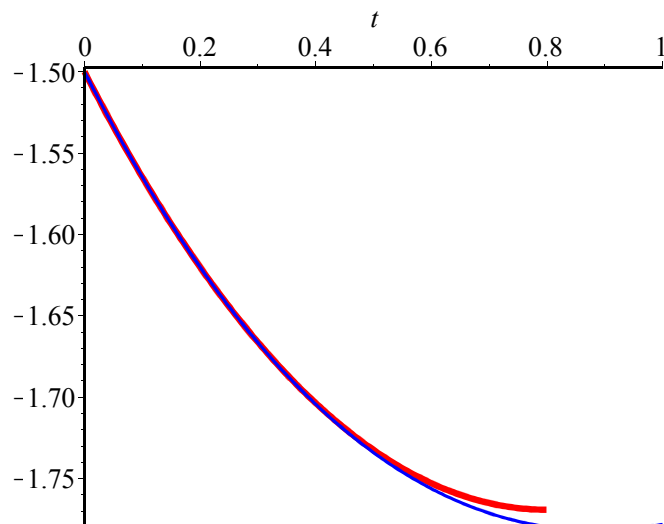
A közelítő megoldás sorokkal:

> *Order := 4 : mo_sor := convert(dsolve({de, y(0) = -1.5}, y(t), type='series'), polynom)*

$$mo_sor := y(t) = -\frac{3}{2} + \frac{1}{20} \sqrt{194} t - \frac{99}{200} t^2 + \left(\frac{3267}{1940000} \sqrt{97} \sqrt{200} - \frac{3267}{194000} \sqrt{97} \sqrt{2} + \frac{11}{2000} \sqrt{194} \right) t^3$$

> *abra := plot(rhs(-mo) - 3, t=0..1, color=red, thickness=3) :*

```
> abra_sor := plot(rhs(-mo_sor) - 3, t = 0..1, color = blue, thickness = 1) :
> plots[display]([abra, abra_sor])
```



Megoldás Euler- módszerrel, a feltételek között elő kell írni a négyzetgyökös kifejezés nemnegatív voltát:

```
> y(t+h) = y(t) + h * sqrt(0.66*y(t)^2 - 1) : h := 0.1 : y[0] := -1.5 : t[0] := 0 :
```

```
> for i to 10 while 0 ≤ 0.66 (y[i-1])^2 - 1 do
```

```
    t[i] := t[i-1] + h;
```

```
    y[i] := y[i-1] + h * sqrt(0.66 (y[i-1])^2 - 1)
```

```
end do:
```

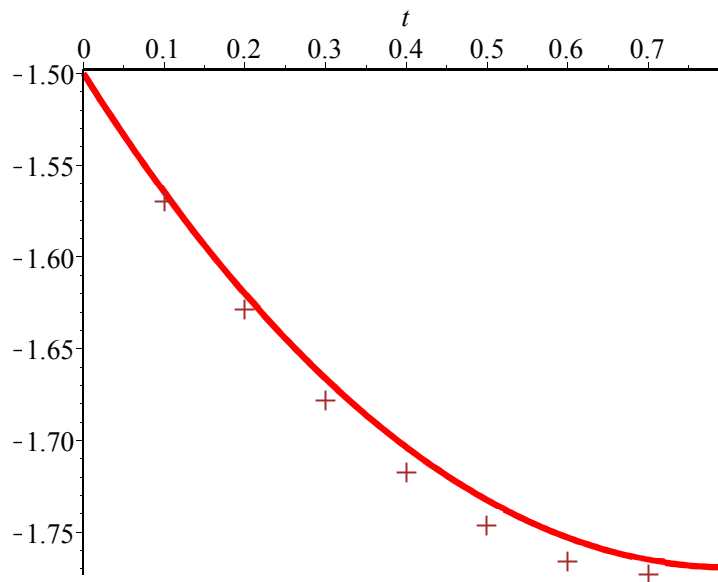
```
> T := [seq(t[i], i = 1..10)]; Y := [seq(y[i], i = 1..10)]
```

```
T := [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, t8, t9, t10]
```

```
Y := [-1.430358059, -1.371171071, -1.322092290, -1.282896309, -1.253529127,
      -1.234272627, -1.226881358, y8, y9, y10]
```

Tehát csak 7 értékes pontunk van.

```
> pontok := plot([seq([Ti - Yi - 3], i = 1..7)], style = point, symbolsize = 20, symbol = cross,
    color = brown) : plots[display]([abra, pontok]);
```



Megoldás Runge-Kutta módszerével:

> $y := 'y'; t := 't'; de := \frac{d}{dt} y(t) = \sqrt{0.66 y(t)^2 - 1}$

$$de := \frac{d}{dt} y(t) = \sqrt{0.66 y(t)^2 - 1}$$

> $mo_rkf45 := dsolve(\{de, y(0) = -1.5\}, y(t), numeric, method = rkf45, output = piecewise, range = 0..0.7) :$

> $h := 0.1 : y[0] := -1.5 : t[0] := 0 :$
for i **from** 0 **to** 7 **do** $y[i] := rhs\left(mo_rkf45_2 \Big|_{t=t[i]}\right) ; t[i+1] := t[i] + h$ **end do**;

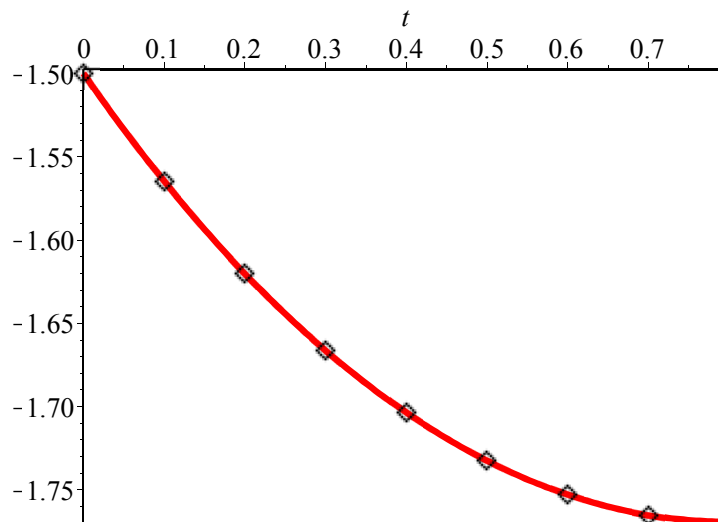
> $Tl := [seq(t[i], i = 0..7)] ; Yl := [seq(y[i], i = 0..7)]$

$Tl := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$

$Yl := [-1.500000000000000, -1.43523409435298, -1.37994590339054, -1.33377026336623, -1.29640227186366, -1.26759526798236, -1.24715897202043, -1.23495870800900]$

> $pontokl := plot([seq([Tl_i, -Yl_i - 3], i = 1..8)], style = point, symbolsize = 20, symbol = diamond, color = black) :$

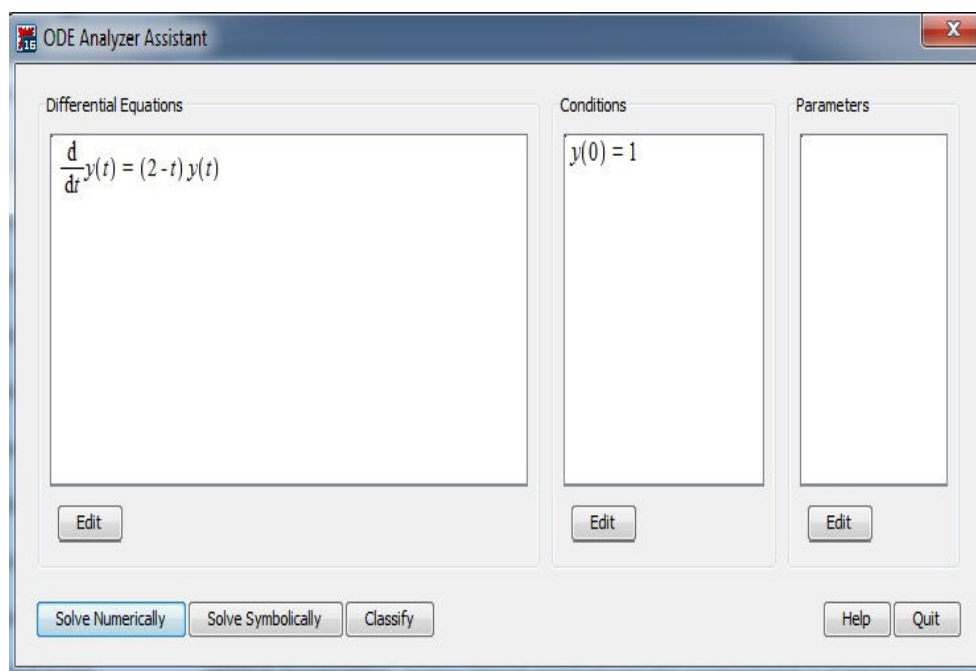
> $display([abra, pontokl])$

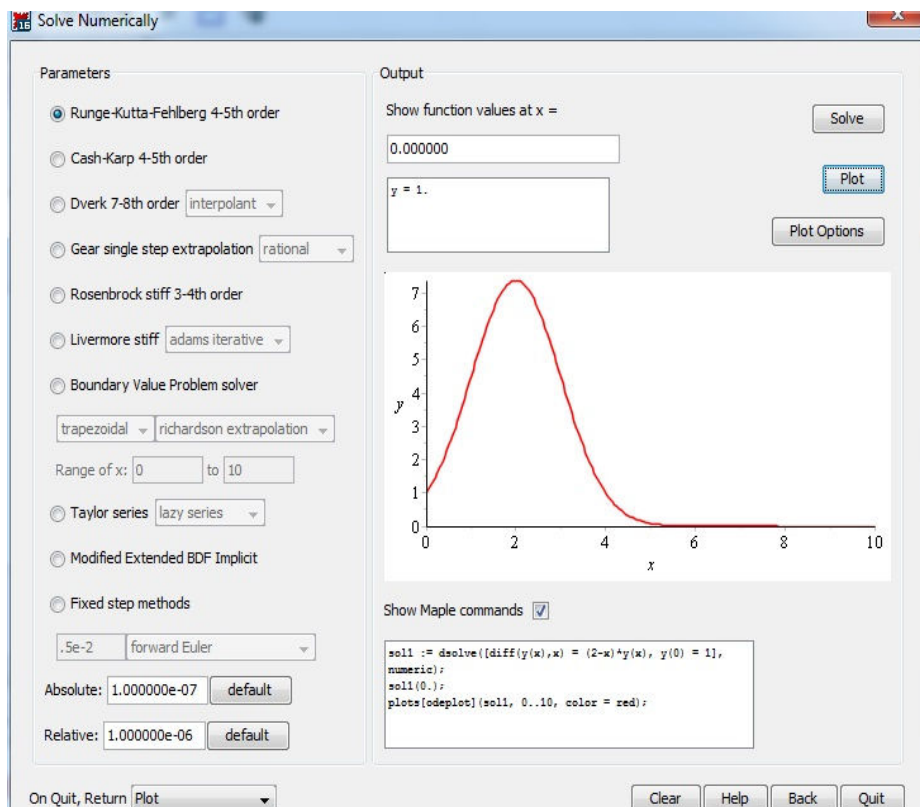
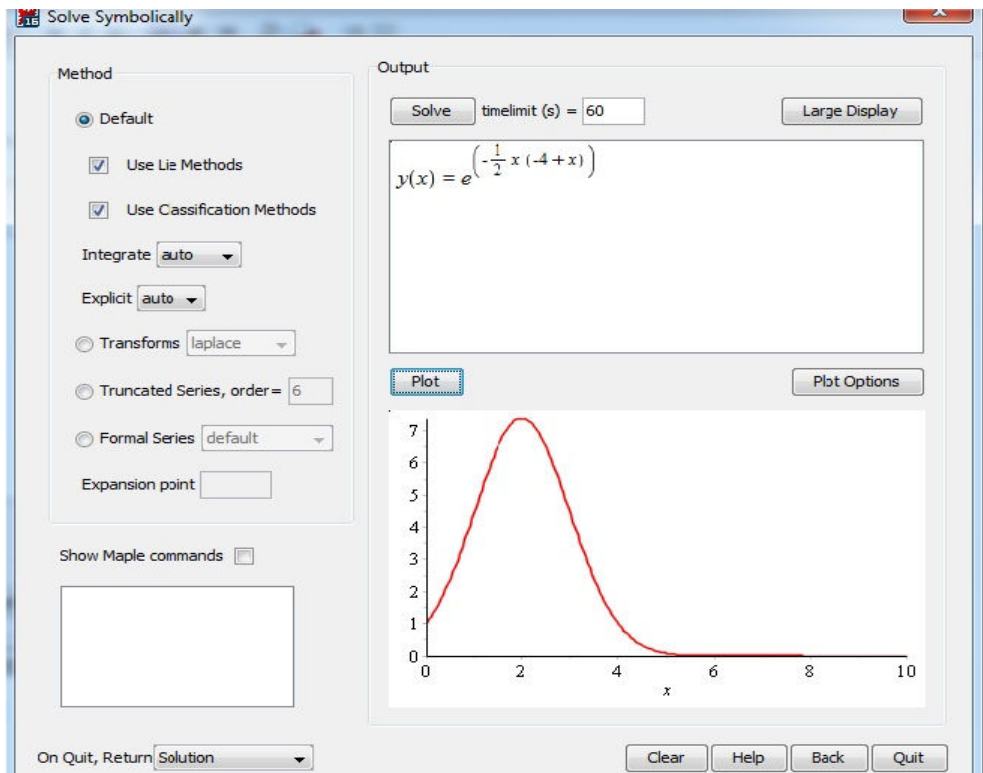


A számított függvényértékek a valódi görbére illeszkednek.

▼ 10.7. Beépített interaktív Maple lehetőségek

Eszközök $\Rightarrow$ Asszisztens $\Rightarrow$ ODE elemző Itt akár szimbolikusan akár numerikusan is megoldhatunk elsőrendű differenciálegyenleteket.





> dsolve[interactive]();

▼ 10. 9. Feladatok

1. Feladat Oldjuk meg a differenciálegyenleteket Picard módszerével a $[0, 1]$ intervallumon. Számítsuk ki a közelítő függvényértékeket, ha $x=0.5$ és $x=1.0$.

$$a; y' = y \quad y(0) = 1, \quad b; y' = x^2 + y^2 \quad y(0) = 2, \quad c; y' = \sin y \quad y(0) = 0$$

2. Feladat Oldja meg Euler módszerrel:

$$a; y' = 2y \quad y(0) = 1 \quad a = 0, \quad b = 0.5, \quad h = 0.1 \\ b; y' = 1 - xy \quad y(0) = 0 \quad a = 0, \quad b = 0.5, \quad h = 0.1$$

3. Feladat Runge-Kutta módszerrel adja meg a differenciálegyenletek megoldását.

$$a; y' = x + \frac{1}{y} \quad y(2) = 1 \quad a = 2, \quad b = 3, \quad n = 5 \\ b; y' = x - y \quad y(0) = 1 \quad a = 0, \quad b = 1, \quad n = 4 \\ c; y' = 2y + e^x \quad y(0) = 0 \quad a = 0, \quad b = 1.2, \quad n = 4 \\ d; y' + \frac{y}{x} = \frac{1}{x^2} \quad y(1) = 1 \quad a = 1, \quad b = 1.3, \quad n = 3$$

4. Feladat Számítsa ki a leeső test sebességét a $t=1, 2, 3$ s időpontokban, ha a test 1200 m magasságból $v=60 \frac{m}{s}$ kezdősebességgel kezd esni, és a levegő ellenállása a sebesség négyzetével arányos. A mozgás egyenlete:

$$v' = -k \cdot v^2 + g, \quad \text{ahol } k=0.0004, g=9.81 \frac{m}{s^2}$$

Az eredményt adja meg az ismert módszerekkel, majd hasonlítsa össze a kapott eredményeket.

5. Feladat Egy gazdának három fia és egy 80 cm átmérőjű hordója van, amelyben 1 m magasságig bor van. A hordó kilyukadt és az alján egy 3 cm sugarú, kör alakú nyílás keletkezett, amin át a bor elkezdett kifolyni. A kifolyás sebessége függ a hordóban lévő folyadék h magasságától

$$-A \cdot \frac{dh}{dt} = 0.6 \cdot \sigma \cdot \sqrt{2gh}, \quad \text{ahol } A \text{ a hordó alapterülete, } \sigma \text{ a nyílás alapterülete, } g=9.81 \frac{m}{s^2}.$$

A gazda leküldi a legidősebb fiát, hogy foltozza be a nyílást. A fiút 10 perc perc próbálkozás után felváltja a középső testvér, majd mivel ő is sikertelenül próbálkozik, 10 perc múlva a legkisebb következik. Neki 10 perc elteltével sikerül betömnie a nyílást. Runge-Kutta módszerével állapítsuk meg, hogy a próbálkozások alatt hány liter bor folyt el.

▼ 10. 8. Ellenőrző kérdések

1. Milyen elvet használ a Picard iterációs módszer közönséges elsőrendű differenciálegyenletek megoldására?
2. Hogyan közelítjük a differenciálegyenlet megoldását Taylor sorral?
3. Milyen elvet használ Euler módszere a közönséges elsőrendű differenciálegyenletek megoldására?
4. Mi a Heun módszer?
5. Fogalmazza meg a Runge-Kutta módszerek elvét.