Rezolvarea ecuatiilor si sistemelor de ecuatii

Ecuatii si sisteme de ecuatii algebrice

Specificarea ecuatiilor si sistemelor de ecuatii

- **Ecuatii**

ecuatie = expresie relationala ce contine operatorul de egalitate

Exemplu: <membru stang> = = <membru drept>

```
ec1 = a x^2 + b x + c == 0
```

$c + b x + a x^2 == 0$

```
ec2 = x^2 - x + 1 == 0
```

$1 - x + x^2 == 0$

- **Sisteme de ecuatii**

Sistem de ecuatii = ansamblu (lista) de ecuatii

Exemple:

1. lista de ecuatii: {ec1, ec2, ..., ecn}
2. egalitate a doua liste: {ms1,ms2,...,msn}=={md1,md2,...,mdn}
3. expresie relationala compusa: ec1 && ec2 && ... && ecn

Fie sistemul $\begin{cases} 3x^2 + y == 2 \\ 4x + 5xy == 1 \end{cases}$

```
sistem1 = {3 x^2 + y == 2, 4 x + 5 x y == 1}
```

$\{3 x^2 + y == 2,\ 4 x + 5 x y == 1\}$

```
sistem2 = {3 x^2 + y, 4 x + 5 x y} == {2, 1}
```

$\{3 x^2 + y,\ 4 x + 5 x y\} == \{2, 1\}$

```
sistem3 = 3 x^2 + y ==  2 && 4 x + 5 x y == 1
```

$3 x^2 + y == 2\ \&\&\ 4 x + 5 x y == 1$

Observatie: sistemele liniare pot fi specificate prin: A.X==B

```
A = {{a, b}, {c, d}}; B = {e, f}; X = {x, y};
```

```
sistem4 = A.X == B
```

{a x + b y, c x + d y} == {e, f}

Rezolvare prin metode exacte

Mathematica poate rezolva exact orice ecuatie polinomiala de grad mai mic decat 5, orice sistem liniar, precum si ecuatii transcendente in care intervin functii inversabile.

Functii principale: Solve (furnizeaza valori exacte ale solutiilor)

NSolve (furnizeaza valori aproximative ale solutiilor - pot fi aplicate doar ecuatiilor/sistemelor cu coeficienti numerici)

Mod de apel: Solve[ecuatie, necunoscuta] sau Solve[sistem, lista necunoscute]

similar pentru NSolve

Solutii: Solve si NSolve returneaza solutiile sub forma unor reguli de transformare

- **Exemple de rezolvare a ecuatiilor**

```
sol1 = Solve[ec1, x]
```

$$\left\{\left\{x \to \frac{-b - \sqrt{b^2 - 4\,a\,c}}{2\,a}\right\}, \left\{x \to \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a}\right\}\right\}$$

```
(* extragerea valorilor radacinilor din regulile de transformare
 returnate de catre Solve *)
```

```
x /. sol1
```

$$\left\{\frac{-b - \sqrt{b^2 - 4\,a\,c}}{2\,a}, \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a}\right\}$$

```
(* verificarea
 radacinilor: se aplica regulile de transformare asupra ecuatiei *)
```

```
ec1 /. sol1
```

$$\left\{c+\frac{b\left(-b-\sqrt{b^2-4\,a\,c}\right)}{2\,a}+\frac{\left(-b-\sqrt{b^2-4\,a\,c}\right)^2}{4\,a}==0,\ c+\frac{b\left(-b+\sqrt{b^2-4\,a\,c}\right)}{2\,a}+\frac{\left(-b+\sqrt{b^2-4\,a\,c}\right)^2}{4\,a}==0\right\}$$

```
Simplify[ec1 /. sol1]
```

```
{True, True}
```

```
(*  Rezolvarea ecuatiei x^2- x +1==0 *)
```

```
sol2 = Solve[ec2, x]
```

$$\left\{\left\{x\to(-1)^{1/3}\right\},\ \left\{x\to-(-1)^{2/3}\right\}\right\}$$

$$\left\{\left\{x\to(-1)^{1/3}\right\},\ \left\{x\to-(-1)^{2/3}\right\}\right\}$$

```
N[%]  (* determinarea valorilor aproximative *)
```

$\{\{x\to 0.5+0.866025\,i\},\ \{x\to 0.5-0.866025\,i\}\}$

```
sol2b = NSolve[ec2, x]
 (* determinarea directa a valorilor aproximative *)
```

$\{\{x\to 0.5-0.866025\,i\},\ \{x\to 0.5+0.866025\,i\}\}$

■ Ecuatii transcendente

```
Solve[Log[x^2] == 4, x]
```

$\{\{x\to e^2\}\}$

```
Solve[Sin[x^2] == 0.5, x]
```

```
Solve::ifun :
 Inverse functions are being used by Solve, so some solutions may not
   be found; use Reduce for complete solution information. >>
```

```
{{x → -0.723601}, {x → 0.723601}}
```

■ Ecuatii polinomiale de grad mare

```
Solve[x^10 - 3 x^5 + 4 x^3 - x^2 + 1 == 0, x]
```

```
{{x → -1}, {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 1]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 2]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 3]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 4]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 5]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 6]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 7]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 8]},
 {x → Root[1 - #1 + 4 #1^3 - 4 #1^4 + #1^5 - #1^6 + #1^7 - #1^8 + #1^9 &, 9]}}
```

```
N[%]
```

```
{{x → -1.}, {x → -0.604405},
 {x → -0.948402 - 0.927994 i}, {x → -0.948402 + 0.927994 i},
 {x → 0.333717 - 0.547602 i}, {x → 0.333717 + 0.547602 i},
 {x → 0.339644 - 1.31575 i}, {x → 0.339644 + 1.31575 i},
 {x → 1.07724 - 0.277579 i}, {x → 1.07724 + 0.277579 i}}
```

Observatie. Root permite identificarea succesiva a radacinilor a unei ecuatii polinomiale folosind metode numerice de izolare a radacinilor (functioneaza pentru polinoame de orice grad)

```
N[Root[#^8 + #^5 + 1 &, 3]]
```

```
-0.314537 - 0.844821 i
```

```
N[Roots[x^8 + x^5 + 1 == 0, x]]
```

$x == -1.00377 - 0.270225\, i \,||\, x == -1.00377 + 0.270225\, i \,||$
$x == -0.314537 - 0.844821\, i \,||\, x == -0.314537 + 0.844821\, i \,||$
$x == 0.490558 - 1.02215\, i \,||\, x == 0.490558 + 1.02215\, i \,||$
$x == 0.827745 - 0.448046\, i \,||\, x == 0.827745 + 0.448046\, i$

Reduce permite identificarea tuturor seturilor de parametri care conduc la solutii:

```
Reduce[a x^2 + b x + c == 0, x]
```

$$\left(a \neq 0 \,\&\&\, \left(x == \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \,||\, x == \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}\right)\right) \,||$$
$$\left(a == 0 \,\&\&\, b \neq 0 \,\&\&\, x == -\frac{c}{b}\right) \,||\, (c == 0 \,\&\&\, b == 0 \,\&\&\, a == 0)$$

■ Sisteme de ecuatii

$$\begin{cases} 3 x^2 + y == 2 \\ 4 x + 5 x y == 1 \end{cases}$$

```
Solve[sistem1, {x, y}]
```

$$\left\{\{y \to -1, x \to -1\}, \left\{y \to \frac{7}{10} + \frac{\sqrt{\frac{33}{5}}}{2}, x \to \frac{1}{30}\left(15 - \sqrt{165}\right)\right\},\right.$$
$$\left.\left\{y \to \frac{1}{10}\left(7 - \sqrt{165}\right), x \to \frac{1}{30}\left(15 + \sqrt{165}\right)\right\}\right\}$$

```
Solve[sistem2, {x, y}]
```

$$\left\{\{y \to -1, x \to -1\}, \left\{y \to \frac{7}{10} + \frac{\sqrt{\frac{33}{5}}}{2}, x \to \frac{1}{30}\left(15 - \sqrt{165}\right)\right\},\right.$$
$$\left.\left\{y \to \frac{1}{10}\left(7 - \sqrt{165}\right), x \to \frac{1}{30}\left(15 + \sqrt{165}\right)\right\}\right\}$$

```
Solve[sistem3, {x, y}]
```

$$\left\{\{y\to -1,\ x\to -1\},\ \left\{y\to \frac{7}{10}+\frac{\sqrt{\frac{33}{5}}}{2},\ x\to \frac{1}{30}\left(15-\sqrt{165}\right)\right\},\ \left\{y\to \frac{1}{10}\left(7-\sqrt{165}\right),\ x\to \frac{1}{30}\left(15+\sqrt{165}\right)\right\}\right\}$$

```
s = NSolve[sistem1, {x, y}]
```

```
{{x → 0.0718256, y → 1.98452},
 {x → 0.928174, y → -0.584523}, {x → -1., y → -1.}}
```

Construirea listei cu valori ale solutiilor:

```
{x, y} /. s
```

```
{{0.0718256, 1.98452}, {0.928174, -0.584523}, {-1., -1.}}
```

Extragerea valorilor uneia dintre necunoscute (de exemplu, x):

```
x /. s
```

```
{0.0718256, 0.928174, -1.}
```

■ Sisteme de ecuatii liniare (A.X==B)

```
A = {{a, b}, {c, d}}; B = {e, f}; X = {x, y}; sistem4 = A.X == B
```

```
{a x + b y, c x + d y} == {e, f}
```

```
Solve[sistem4, X]
```

$$\left\{\left\{x\to -\frac{-d\,e+b\,f}{-b\,c+a\,d},\ y\to -\frac{-c\,e+a\,f}{b\,c-a\,d}\right\}\right\}$$

Functie dedicata sistemelor de ecuatii liniare: LinearSolve

```
LinearSolve[A, B]
```

$$\left\{\frac{d e - b f}{-b c + a d}, \frac{c e - a f}{b c - a d}\right\}$$

Alta varianta aplicabila sistemelor liniare

```
Simplify[Inverse[A].B]
```

$$\left\{\frac{d e - b f}{-b c + a d}, \frac{c e - a f}{b c - a d}\right\}$$

Alte functii:

Eliminate: permite eliminarea prin substitutie a unei necunoscute din toate ecuatiile sistemului

```
Eliminate[sistem1, x]
```

$$-64 y - 10 y^2 + 25 y^3 == 29$$

Rezolvare prin metode iterative

Anumite ecuatii/sisteme nu pot fi rezolvate exact. Ideea rezolvarii prin metode iterative consta in a porni de la o aproximatie initiala a solutiei si a o imbunatati pana cand se obtine o aproximare suficient de buna a solutiei.

Functia: FindRoot

Mod de apel: FindRoot[ecuatie, {x,x0}] (x0 reprezinta aproximatia initiala)

FindRoot[sistem, {x,x0},{y,y0}, ...] ((x0,y0,...) reprezinta aproximatia initiala)

Observatii:

1. Numarul de iteratii poate fi controlat prin optiunea MaxIterations->val

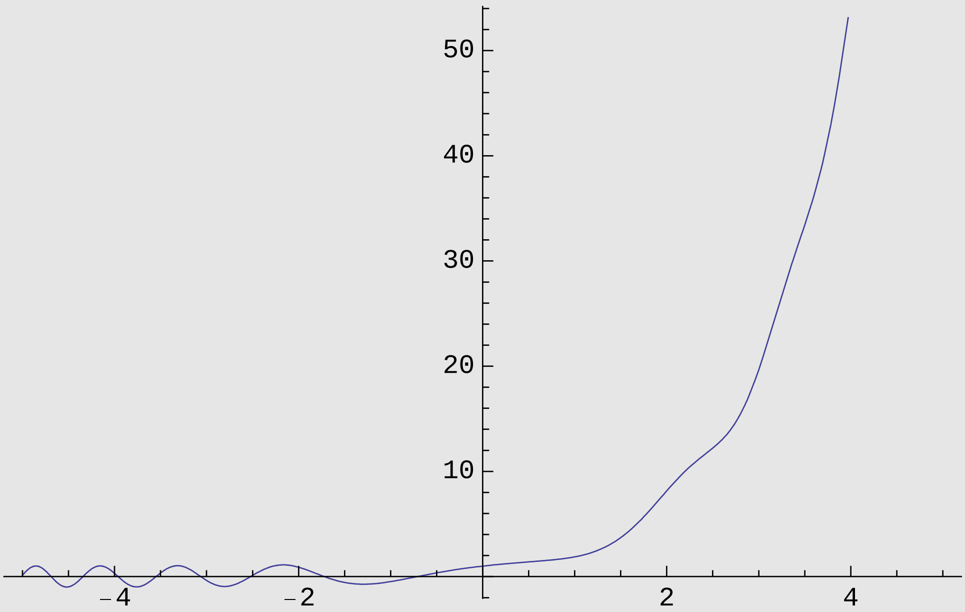
2. Stabilirea valorii initiale se bazeaza pe informatii despre solutie extrase, eventual, din reprezentarea grafica a functiei

3. Functia se bazeaza pe utilizarea metodelor de tip Newton

Exemplul 1

```
f[x_] := Exp[x] - Sin[x^2]
```

```
Plot[f[x], {x, -5, 5}]
```




```
NSolve[f[x] == 0, x]
```

Solve::tdep :
 The equations appear to involve the variables to be solved
 for in an essentially non-algebraic way. >>

```
NSolve[e^x - Sin[x^2] == 0, x]
```

```
FindRoot[f[x] == 0, {x, 0, 1}]
```

```
{x → -0.714969}
```

```
FindRoot[f[x] == 0, {x, -1}]
```

```
{x → -0.714969}
```

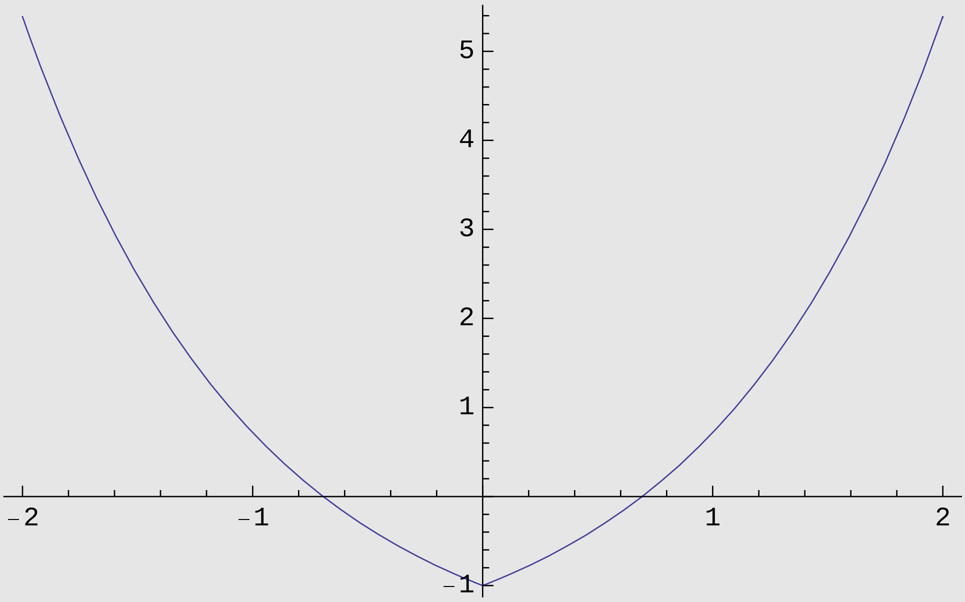
```
FindRoot[f[x] == 0, {x, -2}]
```

```
{x → -1.72097}
```

- **Exemplul 2:**

```
g[x_] := Exp[Abs[x]] - 2
```

```
Plot[g[x], {x, -2, 2}]
```




```
FindRoot[g[x] == 0, {x, 1}]
```

```
{x → 0.693147}
```

Observatie: metoda lui Newton poate fi aplicata si pentru sisteme de ecuatii, insa apar dificultati daca jacobianul nu poate fi calculat simbolic. In acest caz se poate folosi metoda secantei (trebuie specificate doua valori pentru aproximatia initiala).

```
FindRoot[g[x] == 0, {x, 0, 1}]
```

```
{x → 0.693147}
```

```
FindRoot[g[x] == 0, {x, -5, -3}]
```

```
{x → -0.693147}
```

Ecuatii si sisteme de ecuatii diferentiale

Specificarea ecuatiilor diferentiale

Fara conditii initiale

Se specifica similar ecuatiilor algebrice: ca expresii relationale ce contin operatorul de egalitate insa in care intervin derivate ale functiilor necunoscute

(i) ecuatie: y'(x)=ay(x) ==> y'[x]==a y[x]

(ii) sistem: x'(t)=-2 x(t)+y(t) ==> {x'[t]==-2 x[t]+y[t], y'[t]==-y[t]+1}

y'(t)=-y(t)+1

```
ecdif1 = y'[x] == a y[x]
```

```
y′[x] == a y[x]
```

```
sisdif1 = {x'[t] == -2 x[t] + y[t], y'[t] == -y[t] + 1}
```

```
{x′[t] == -2 x[t] + y[t], y′[t] == 1 - y[t]}
```

Cu conditii initiale (probleme Cauchy)

Ecuatiile diferentiale cu conditii initiale se specifica prin liste de expresii relationale (conditia initiala se intrepreteaza ca o ecuatie suplimentara)

y'(x)=ay(x) ==> {y'[x]==a y[x], y[0]==1}

y(0)=1

```
pbCauchy1 = {y'[x] == a y[x], y[0] == 1}
```

```
{y′[x] == a y[x], y[0] == 1}
```

```
pbCauchy2 = {x'[t] == -2 x[t] + y[t], y'[t] == -y[t] + 1, x[0] == 0, y[0] == 1}
```

```
{x′[t] == -2 x[t] + y[t], y′[t] == 1 - y[t], x[0] == 0, y[0] == 1}
```

Rezolvarea analitica a ecuatiilor diferentiale

Exista functii pentru rezolvarea ecuatiilor cu/fara conditii initiale ce contin parametri simbolici sau numerici. Se bazeaza pe metode analitice de rezolvare a ecuatiilor diferentiale.

Functia: DSolve

Variante de apel:

(i) DSolve[ecuatie in y[x], y[x], x] (returneaza reguli de transformare pentru expresia y[x])

(ii) DSolve[ecuatie in y[x], y, x] (returneaza reguli de transformare pentru functia y)

■ Exemple

```
sd1 = DSolve[ecdif1, y[x], x]
```

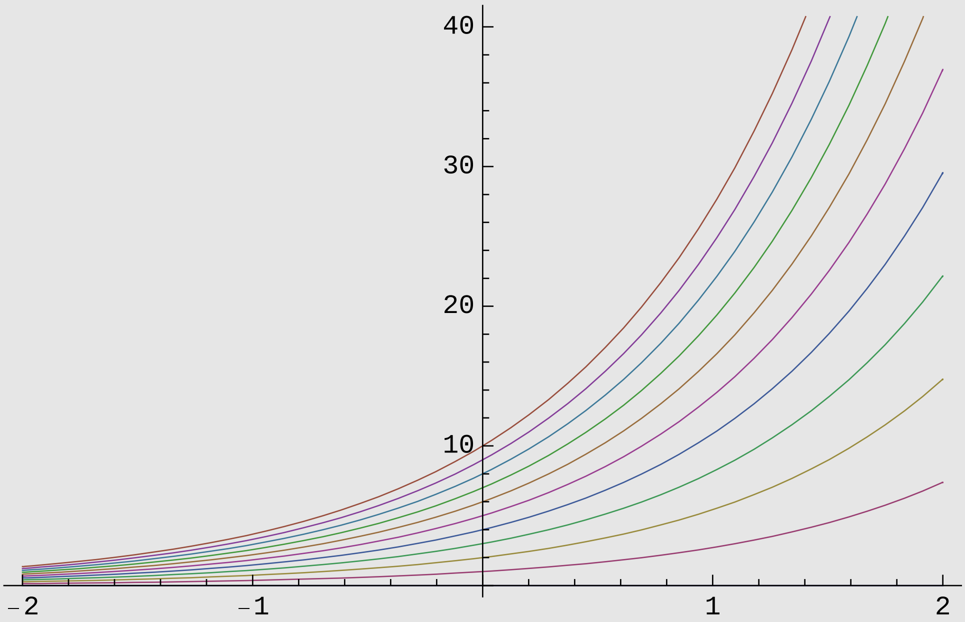
$\{\{y[x] \to e^{a x} C[1]\}\}$

```
sd1 = DSolve[ecdif1, y, x]
```

$\{\{y \to \text{Function}[\{x\}, e^{a x} C[1]]\}\}$

```
(* Reprezentarea grafica a familiei de solutii pentru valori
 ale constantei de integrare cuprinse intre 0 si 10 *)
```

```
Plot[Evaluate[Table[y[x] /. sd1 /. {a → 1, C[1] → i}, {i, 0, 10}],
  {x, -2, 2}]]
```



Rezolvarea sistemului x'[t] == -2 x[t] + y[t], y'[t] == -y[t] + 1

```
sd2 = DSolve[sisdif1, {x, y}, t]
```

$$\left\{\left\{x \to \text{Function}\left[\{t\}, -\frac{1}{2} e^{-t} \left(-2 + e^{t}\right) + e^{-t} \left(-1 + e^{t}\right) + e^{-2t} C[1] + e^{-2t} \left(-1 + e^{t}\right) C[2]\right], y \to \text{Function}\left[\{t\}, 1 + e^{-t} C[2]\right]\right\}\right\}$$

```
Simplify[x[t] /. sd2 /. {C[1] → -1, C[2] → 1}]
```

$$\left\{\frac{1}{2} - 2 e^{-2t} + e^{-t}\right\}$$

Varianta cu functiile necunoscute sub forma de expresii:

```
sd3 = DSolve[pbCauchy2, {x[t], y[t]}, t]
```

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

```
x[t] /. sd3 /. t → 1
```

$$\left\{\frac{-1+e^2}{2\, e^2}\right\}$$

Ecuatii diferentiale cu conditii initiale

Varianta cu functiile necunoscute sub forma de functii pure:

```
sd4 = DSolve[pbCauchy2, {x, y}, t]
```

$$\left\{\left\{x \to \text{Function}\left[\{t\},\ \frac{1}{2}\, e^{-2t}\left(-1+e^{2t}\right)\right],\ y \to \text{Function}[\{t\},\ 1]\right\}\right\}$$

Calculul valorii solutiei pentru un argument dat

```
x[1] /. sd4
```

$$\left\{\frac{-1+e^2}{2\, e^2}\right\}$$

DSolve permite si rezolvarea ecuatiilor de ordin superior (exemplu: ecuatia oscilatorului armonic liniar):

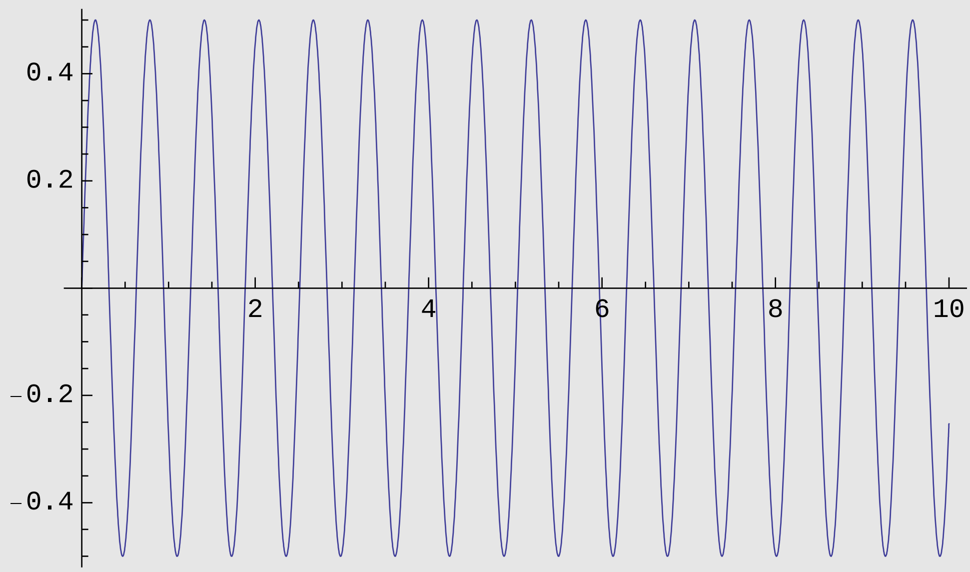
```
ecdif2 = {x''[t] + ω^2 x[t] == 0, x[0] == 0, x'[0] == v0}
```

$$\left\{\omega^2\, x[t] + x''[t] == 0,\ x[0] == 0,\ x'[0] == v0\right\}$$

```
sd5 = DSolve[ecdif2, x, t]
```

$$\left\{\left\{x \to \text{Function}\left[\{t\},\ \frac{v0\,\text{Sin}[t\,\omega]}{\omega}\right]\right\}\right\}$$

```
Plot[x[t] /. sd5 /. {ω → 10, v0 → 5}, {t, 0, 10}]
```



Rezolvarea numerica a ecuatiilor diferentiale

Se aplica ecuatiilor diferentiale cu coeficienti numerici si conditii initiale care nu pot fi rezolvate analitic.

Functia: NDSolve

Variante de apel:

(i) NDSolve[ecuatie in y[x], y[x], {x,x0,x1}] (returneaza reguli de transformare pentru expresia y[x])

(ii) NDSolve[ecuatie in y[x], y, {x,x0,x1}] (returneaza reguli de transformare pentru functia y)

Metode folosite: metode numerice de rezolvare a ecuatiilor diferentiale (Runge Kutta, Adams etc.)

Rezultat: obiect de tip InterpolatingFunction, care poate fi evaluat pentru orice argument in domeniul specificat in NDSolve

```
DSolve[{x'[t] == x[t] * Exp[x[t]], x[0] == 1}, x, t]
```

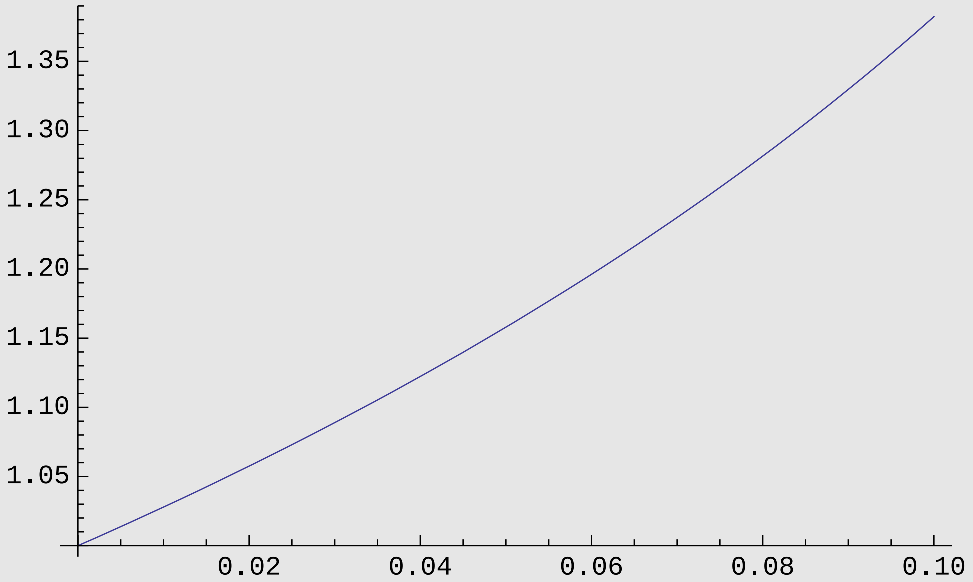
```
InverseFunction::ifun :
 Inverse functions are being used. Values may be
   lost for multivalued inverses. >>

InverseFunction::ifun :
 Inverse functions are being used. Values may be
   lost for multivalued inverses. >>
```

```
{{x → Function[{t}, -ExpIntegralEi^(-1)[
      t + InverseFunction[ExpIntegralEi^(-1), 1, 1][-1]]]}}
```

```
sol = NDSolve[{x'[t] == x[t] * Exp[x[t]], x[0] == 1}, x, {t, 0, 0.1}]
```

```
{{x → InterpolatingFunction[{{0., 0.1}}, <>]}}
```

```
Plot[x[t] /. sol, {t, 0, 0.1}]
```




```
x[0.2] /. sol
```

```
InterpolatingFunction::dmval :
 Input value {0.2} lies outside the range of data in the
   interpolating function. Extrapolation will be used. >>
```

```
{2.33439}
```