

> restart

ESTA ES LA PRIMERA CLASE DE MAPLE

> EcuacionAlgebraica := $x \cdot 2 + 5 \cdot x + 6 = 0$:

> LadoIzquierdoEcuacion := lhs(EcuacionAlgebraica); LadoDerechoEcuacion
:= rhs(EcuacionAlgebraica);

$LadoIzquierdoEcuacion := x^2 + 5x + 6$

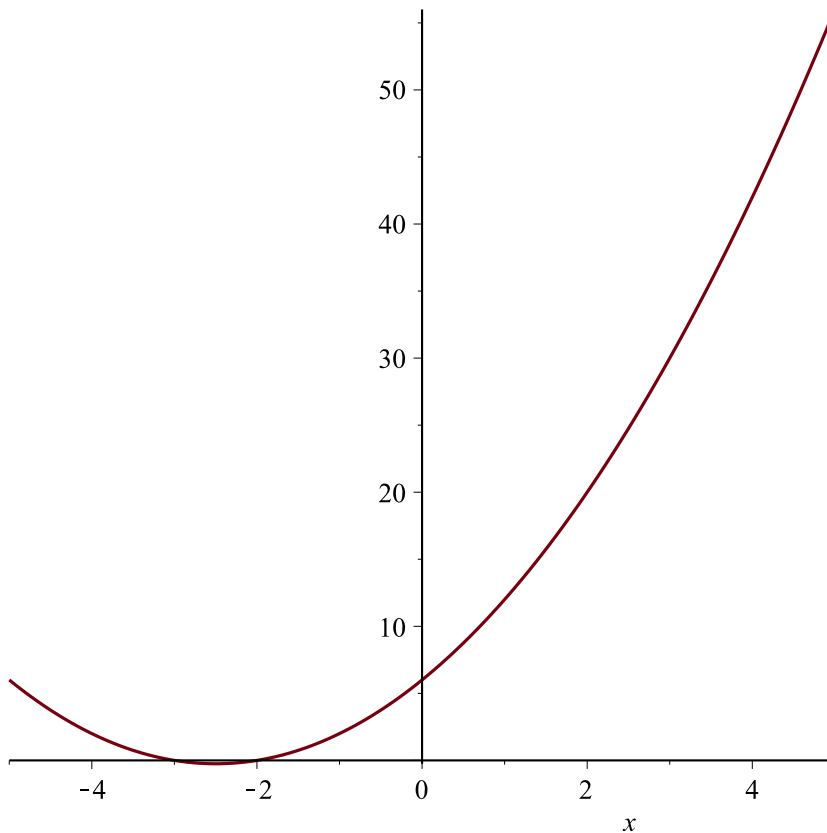
$LadoDerechoEcuacion := 0$

(1)

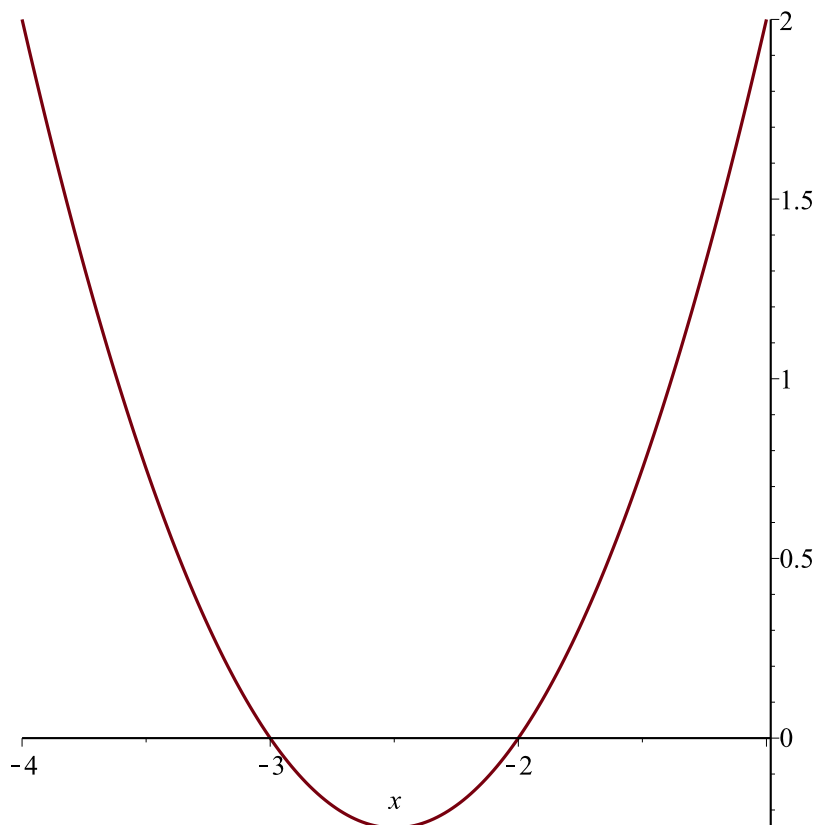
> plot(EcuacionAlgebraica, x=-5..5)

Error, invalid input: plot expects its 1st argument, p, to be of type {set, array, list, rtable, algebraic, procedure, And (`module`, applicable)}, but received $x^2 + 5x + 6 = 0$

> plot(LadoIzquierdoEcuacion, x=-5..5)



> plot(LadoIzquierdoEcuacion, x=-4..-1)



```
> Raiz := solve(EcuacionAlgebraica)
```

$Raiz := -2, -3$

(2)

```
> Raiz[1]
```

-2

(3)

```
> Raiz[2]
```

-3

(4)

```
> EcuacionOriginal := expand((x - Raiz[1]) * (x - Raiz[2])) = 0
```

$EcuacionOriginal := x^2 + 5x + 6 = 0$

(5)

```
> restart
```

```
> evalf(pi)
```

π

(6)

```
> evalf(PI)
```

Π

(7)

```
> evalf(Pi)
```

3.141592654

(8)

```
> evalf(Pi, 25)
```

3.141592653589793238462643

(9)

> evalf(Pi, 1000)

3.1415926535897932384626433832795028841971693993751058209749445923078164062862\ (10)
08998628034825342117067982148086513282306647093844609550582231725359408128\
48111745028410270193852110555964462294895493038196442881097566593344612847\
56482337867831652712019091456485669234603486104543266482133936072602491412\
73724587006606315588174881520920962829254091715364367892590360011330530548\
82046652138414695194151160943305727036575959195309218611738193261179310511\
85480744623799627495673518857527248912279381830119491298336733624406566430\
86021394946395224737190702179860943702770539217176293176752384674818467669\
40513200056812714526356082778577134275778960917363717872146844090122495343\
01465495853710507922796892589235420199561121290219608640344181598136297747\
71309960518707211349999998372978049951059731732816096318595024459455346908\
30264252230825334468503526193118817101000313783875288658753320838142061717\
76691473035982534904287554687311595628638823537875937519577818577805321712\
26806613001927876611195909216420199

> evalf(Pi, 5000)

3.1415926535897932384626433832795028841971693993751058209749445923078164062862\ (11)
08998628034825342117067982148086513282306647093844609550582231725359408128\
48111745028410270193852110555964462294895493038196442881097566593344612847\
56482337867831652712019091456485669234603486104543266482133936072602491412\
73724587006606315588174881520920962829254091715364367892590360011330530548\
82046652138414695194151160943305727036575959195309218611738193261179310511\
85480744623799627495673518857527248912279381830119491298336733624406566430\
86021394946395224737190702179860943702770539217176293176752384674818467669\
40513200056812714526356082778577134275778960917363717872146844090122495343\
01465495853710507922796892589235420199561121290219608640344181598136297747\
71309960518707211349999998372978049951059731732816096318595024459455346908\
30264252230825334468503526193118817101000313783875288658753320838142061717\
76691473035982534904287554687311595628638823537875937519577818577805321712\
26806613001927876611195909216420198938095257201065485863278865936153381827\
96823030195203530185296899577362259941389124972177528347913151557485724245\
41506959508295331168617278558890750983817546374649393192550604009277016711\
39009848824012858361603563707660104710181942955596198946767837449448255379\
77472684710404753464620804668425906949129331367702898915210475216205696602\
40580381501935112533824300355876402474964732639141992726042699227967823547\
81636009341721641219924586315030286182974555706749838505494588586926995690\
92721079750930295532116534498720275596023648066549911988183479775356636980\
74265425278625518184175746728909777727938000816470600161452491921732172147\
72350141441973568548161361157352552133475741849468438523323907394143334547\
76241686251898356948556209921922218427255025425688767179049460165346680498\
86272327917860857843838279679766814541009538837863609506800642251252051173\
92984896084128488626945604241965285022210661186306744278622039194945047123\
71378696095636437191728746776465757396241389086583264599581339047802759009

94657640789512694683983525957098258226205224894077267194782684826014769909\
 02640136394437455305068203496252451749399651431429809190659250937221696461\
 51570985838741059788595977297549893016175392846813826868386894277415599185\
 59252459539594310499725246808459872736446958486538367362226260991246080512\
 43884390451244136549762780797715691435997700129616089441694868555848406353\
 42207222582848864815845602850601684273945226746767889525213852254995466672\
 78239864565961163548862305774564980355936345681743241125150760694794510965\
 96094025228879710893145669136867228748940560101503308617928680920874760917\
 82493858900971490967598526136554978189312978482168299894872265880485756401\
 42704775551323796414515237462343645428584447952658678210511413547357395231\
 13427166102135969536231442952484937187110145765403590279934403742007310578\
 53906219838744780847848968332144571386875194350643021845319104848100537061\
 46806749192781911979399520614196634287544406437451237181921799983910159195\
 61814675142691239748940907186494231961567945208095146550225231603881930142\
 09376213785595663893778708303906979207734672218256259966150142150306803844\
 77345492026054146659252014974428507325186660021324340881907104863317346496\
 51453905796268561005508106658796998163574736384052571459102897064140110971\
 20628043903975951567715770042033786993600723055876317635942187312514712053\
 29281918261861258673215791984148488291644706095752706957220917567116722910\
 98169091528017350671274858322287183520935396572512108357915136988209144421\
 00675103346711031412671113699086585163983150197016515116851714376576183515\
 56508849099898599823873455283316355076479185358932261854896321329330898570\
 64204675259070915481416549859461637180270981994309924488957571282890592323\
 32609729971208443357326548938239119325974636673058360414281388303203824903\
 75898524374417029132765618093773444030707469211201913020330380197621101100\
 44929321516084244485963766983895228684783123552658213144957685726243344189\
 30396864262434107732269780280731891544110104468232527162010526522721116603\
 96665573092547110557853763466820653109896526918620564769312570586356620185\
 58100729360659876486117910453348850346113657686753249441668039626579787718\
 55608455296541266540853061434443185867697514566140680070023787765913440171\
 27494704205622305389945613140711270004078547332699390814546646458807972708\
 26683063432858785698305235808933065757406795457163775254202114955761581400\
 25012622859413021647155097925923099079654737612551765675135751782966645477\
 91745011299614890304639947132962107340437518957359614589019389713111790429\
 78285647503203198691514028708085990480109412147221317947647772622414254854\
 54033215718530614228813758504306332175182979866223717215916077166925474873\
 89866549494501146540628433663937900397692656721463853067360965712091807638\
 32716641627488880078692560290228472104031721186082041900042296617119637792\
 13375751149595015660496318629472654736425230817703675159067350235072835405\
 67040386743513622224771589150495309844489333096340878076932599397805419341\
 447377441842631298608099888687413260472

> Digits := 20

Digits := 20

(12)

```

> evalf(Pi)
3.1415926535897932385 (13)
=
> evalf(sqrt(2))
1.4142135623730950488 (14)
=
> Digits := 50
Digits := 50 (15)
=
> evalf(Pi)
3.1415926535897932384626433832795028841971693993751 (16)
=
> sqrt(3)
 $\sqrt{3}$  (17)
=
> evalf(sqrt(3))
1.7320508075688772935274463415058723669428052538104 (18)
=
> evalf(sqrt(3), 3)
1.73 (19)
=
> alpha
 $\alpha$  (20)
=
> beta
 $\beta$  (21)
=
> Beta
B (22)
=
> Alpha
A (23)
=
> restart
> Complejo := 5 + 3 I
Complejo := 5 + 3 I (24)
=
> ComplejoSimetrico := 5 - 3 I
ComplejoSimetrico := 5 - 3 I (25)
=
> Ecuacion := expand((x - Complejo) · (x - ComplejoSimetrico)) = 0
Ecuacion :=  $x^2 - 10x + 34 = 0$  (26)
=
> evalf(I)
1. I (27)
=
> Euler := exp(Pi · I)
Euler := -1 (28)
=
> EcuacionDos :=  $x^2 - 10x + 34 = 0$ 
EcuacionDos :=  $x^2 - 10x + 34 = 0$  (29)
=
> RaizDos := solve(EcuacionDos)
RaizDos := 5 + 3 I, 5 - 3 I (30)
=
> RealUno := Re(RaizDos[1]); ImaginarioDos := Im(RaizDos[2])
RealUno := 5
ImaginarioDos := -3 (31)
=
> restart
> F :=  $x \cdot 2 + 8 \cdot x + 4$ 

```

```

=>
F := x^2 + 8 x + 4 (32)
> DerF := diff(F, x)
DerF := 2 x + 8 (33)
=>
> AntiDerF := int(F, x)
AntiDerF := 1/3 x^3 + 4 x^2 + 4 x (34)
=>
> IntegralF := AntiDerF + C[1]
IntegralF := 1/3 x^3 + 4 x^2 + 4 x + C1 (35)
=>
> restart
> EcuaDif := diff(y(x), x$3) = 0
EcuaDif := d^3/dx^3 y(x) = 0 (36)
=>
> EcuaDifDos := int(lhs(EcuaDif), x) = C[1]
EcuaDifDos := d^2/dx^2 y(x) = C1 (37)
=>
> EcuaDifTres := int(lhs(EcuaDifDos), x) = int(rhs(EcuaDifDos), x) + C[2]
EcuaDifTres := d/dx y(x) = x C1 + C2 (38)
=>
> SolucionUno := int(lhs(EcuaDifTres), x) = int(rhs(EcuaDifTres), x) + C[3]
SolucionUno := y(x) = 1/2 C1 x^2 + C2 x + C3 (39)
=>
> Solucion := dsolve(EcuaDif)
Solucion := y(x) = 1/2 _C1 x^2 + _C2 x + _C3 (40)
=>
> restart
> Semana := lunes, martes, miércoles, jueves, viernes, sábado, domingo
Semana := lunes, martes, miércoles, jueves, viernes, sábado, domingo (41)
=>
> DiaIntermedio := Semana[3]
DiaIntermedio := miércoles (42)
=>
> DiaHabil := Semana[1..5]
DiaHabil := lunes, martes, miércoles, jueves, viernes (43)
=>
> FinDeSemana := Semana[6..7]
FinDeSemana := sábado, domingo (44)
=>
> Dia := [Semana]
Dia := [lunes, martes, miércoles, jueves, viernes, sábado, domingo] (45)
=>
> Dia[4]
jueves (46)
=>
> Conjunto := {Semana}
Conjunto := {domingo, jueves, lunes, martes, sábado, viernes, miércoles} (47)
=>
> Conjunto[1]
domingo (48)
=>
> VectorA := array([1, 2, 3])

```

$$\text{VectorA} := \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \quad (49)$$

> *VectorB* := array([4,-5,6])

$$\text{VectorB} := \begin{bmatrix} 4 & -5 & 6 \end{bmatrix} \quad (50)$$

> *VectorC* := array([7,8,9])

$$\text{VectorC} := \begin{bmatrix} 7 & 8 & 9 \end{bmatrix} \quad (51)$$

> *MatrizA* := array([[1,2,3],[4,-5,6],[7,8,9]])

$$\text{MatrizA} := \begin{bmatrix} 1 & 2 & 3 \\ 4 & -5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (52)$$

> with(linalg)

[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylveste, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian] (53)

> *ValorA* := det(*MatrizA*)

$$\text{ValorA} := 120 \quad (54)$$

> *InversaA* := inverse(*MatrizA*)

$$\text{InversaA} := \begin{bmatrix} -\frac{31}{40} & \frac{1}{20} & \frac{9}{40} \\ \frac{1}{20} & -\frac{1}{10} & \frac{1}{20} \\ \frac{67}{120} & \frac{1}{20} & -\frac{13}{120} \end{bmatrix} \quad (55)$$

> *Identidad* := evalm(*MatrizA* &* *InversaA*)

$$\text{Identidad} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (56)$$

>