

1. Gaia

Mathematica

1.1 Sarrera

Mathematica kalkulu matematikoak egiteko aplikazio informatikoa da. Batiketako, derivatuak, integralak, ekuaioen ebazenak, determinanteak, ekuazio diferentzialen ebazpenak, grafikoak, eta abar luze bat egiteko gai da.

Mathematica aplikazioa bai Linux sisteman bai Windows sisteman erabiltzeko dago prestatuta.

Idatziera berezia erabiltzen du, oso formal eta zorrotza. Beraz, erabiltzen dugunean, kontu handiz idatzi beharko ditugu adicrazpen matematikoak.

Abiarazi

Abiarazteko nahiak da saguaren ezkerreko botoia bi aldiz sakatzea, kurtsorea *Mathematica*-ren ikono gainean dagoelarik. Kernela eta interfazea; Kernela ez da kargatzeko osagai dauzka, *Mathematica*-k bi osagai gara idatzten gure aginduak. Lehenengo eragileta egin arte.

Kurtsorea

I Testuan edo sarreran (Input) kurtsoea bertsikala izango da.

Idatzi bezain prorio, eskuin aldean kortexe bat agertuko da:

] Formateatu gabe dauden gelaxkak kortexe bat adieraziko ditu aplikazioak. Gelaxka hauetan edo agindu bat edo testu bat idatziko dugu.

Idatzitako agindua ebaluatzeara nahi badugu, "shift" eta "return" teklak batera saltatu behar ditugu. Orduan jarriko da martxan aplikazioaren Kernela. Hurrengo ebaluazioetarako prest egingo da.

Mathematica aplikazioak sarrerak In[] eta irteerak Out[] izendatu eta zenbakitu egiten ditu; beraz, guk ez dugu hori idatzi behar. Kontuz ibili behar dugu; ez aplikazioak sarrera bati zenbaki bat esleituko dio guk ebaluatzeko erabiliko ditugu. Parentesiak eragileken artean dagoen lehentasuna adierazteko erabiliko ditugu.

Bi gelaxken artean kurtsorea horizontala izango da; eta pantaila osoa zeharkatzen duen zuzen horizontal baten gainean agertzen da.

| ←
Gelaxkaren kortexetea aukeratzean, kurtsorea marra bertikal baten kontra dagoen gezi bi-hurtuko da.

⊕
Gelaxka formateatuetan kurtsoreak ⊕ itxura izango du.

+
Grafiko baten barnean, gurutze geziduna izango da kurtsoea. Grafiko mugitzeko balio du.

↔
Grafiko baten ertzean geziak agertuko dira, grafikoaren tamaina aldatzeko.

]
Aktibatu gabeko gelaxkaren kortexete barruan marratxo bat agertuko da goiko aldean. Gelaxka hauetan dagoena ezin da aktibatu.

]]
Gelaxkaren kortexete bikoitzak sarrera eta irteera gelaxkak batera hartzen ditu.

Sarrerak eta irteerak zenbakitzenten dituen bezala, beste agindu batzuk ere etiketazten ditu (*Graphics*, *Graphics3D*, ...). Horiek ere sarrerak ebaluatutakoan agertuko dira. Horiek dira formateatutako gelaxka batzuk.

1.2 Idazkera

1.2.1 Orokorrak

Aplikazioaren agindu guztiak letra larriz hasten dira, baita agindu koposatueta agertzen diren hitzak ere letra larriz hasiko dira. Hori dela eta, norberak sortzen dituen aldagaiaik, aginduak, programak.. letra xchez idaztea gomendatzzen dugu.

(*, *)
Aginduren bat ez ebaluatzeko edo aginduen artean ohar bat idatzeko, agindua edota oharra (* eta *) karaktereen artean idatziko ditugu.

() Parentesiak eragileken artean dagoen lehentasuna adierazteko erabiliko ditugu.

Kortxeteak aplikazioaren aginduen argumentuak idazteko erabiliko ditugu.

{ }

Giltzak zerrendak idatzeko erabiliko ditugu.

==

Ekuazioen bi atalak elkartzeleko berdintza bikoitza erabiliko dugu.

<=, =>

$\leq \text{eta} \geq \text{ikurrak}$ idatzeko, $<= \text{eta} \Rightarrow \text{ikurrak}$ erabiliko ditugu, hurrenez hurren.

;

Irteera ez agertzen, aginduen bukaeran ; idatziko dugu.

Lero berean agindu bat baino gehiago idatz dezakegu. Aginduen artean ; tartekatuz, tza soilik erakutsiko dugu. Tarteko emaitzak ikusi nahi baditugu, agindua leho desberdinetan idatzi beharko ditugu, "return" tekla sakatuz eta, ikurrik gabe; azken agindu berean.

1.2.2 Laguntzak

Aplikazioak aukera desberdinak ematen dizkigu laguntzak eskatzeko. Hauek aginduei edo funtzioei buruzko informazioak lortzeko balio dute.

?Agindua

Aginduaren argumentuak nola idatzi behar diren eta zer emaitza lortuko den esango digu.

In[1]:= ?Plot

Plot[f,{x,x_{min},x_{max}}] generates a plot of f as a function of x from x_{\min} to x_{\max} .

?Plot[f_i,{x,x_{min},x_{max}}] plots several functions f_i . >>

?Agindua

Aginduari buruzko informazio gehigarria emango du.

In[2]:= ??Plot

Plot[f,{x,x_{min},x_{max}}] generates a plot of f as a function of x from x_{\min} to x_{\max} .

Attributes[Plot]={HoldAll,Protected}

AxesPosition->Automatic,AxesOrigin->Center,AspectRatio->1/GoldenRatio,Axes->True,

ColorFunctionScaling->True,ColorOutput->None,CoordinatesToolOptions->Automatic,Epilog->{},Exclusions->Automatic,ExclusionsStyle->None,Filling->None,FrameLabel->None,FrameStyle->Automatic,

FormatType:>TraditionalForm,Frame->False,FrameLabel->None,FrameStyle->{},

```
FrameTicks->Automatic,FrameTicksStyle->{},GridLines->None,GridLinesStyle->{},ImageMargins->0.,ImagePadding->All,ImageSize->{},LabelStyle->{},MaxRecursion->All,ImageSizeRaw->Automatic,MeshShading->None,MeshStyle->Automatic,MeshFunctions->{#1&},PerformanceGoal->PerformanceGoal,PlotLabel->Automatic,PlotRange->{Full,Automatic},PlotPoints->Automatic,PlotRegion->Automatic,PlotStyle->Automatic,PreserveImageOptions->Automatic,RegionFunction->(True&),RotateLabel->True,Ticks->Automatic,TicksStyle->{},WorkingPrecision->MachinePrecision}
```

Options[agin]

Aginduaren hautazko argumentuak emango dizkigu.

In[3]:= Options[Solve]

Out[3]= { InverseFunctions -> Automatic, MakeRules -> False, Method -> 3, -> ∞ } Mode -> Generic, Sort -> True, VerifySolutions -> Automatic, WorkingPrecision -> 3,

?W*

W letraz hasten diren aginduen zerrada aterako du.

In[4]:= ?Z*

System'	ZernikeR	ZeroWidthTimes	ZetaZero	ZTransform
ZeroTest	Zeta			
Names["W*"]				

W letraz hasten diren aginduen zerrada emango digu. Irteera erabil daitelke.

In[5]:= Names["Z*"]

Out[5]= {"ZernikeR", "ZeroTest", "ZeroWidthTimes", "Zeta", "ZetaZero", "ZipfDistribu-

tion", "ZTransform"}

?Hitz*

Hitz horretaz hasten diren aginduen zerrada emango digu.

In[6]:= ?Plot*

System'	Plot	PlotDivision	PlotMarkers	PlotRangeClipping	PlotStyle
Plot	Plot3D	PlotJoined	PlotPoints	PlotRangePadding	
Plot3Matrix	PlotLabel	PlotRange	PlotRegion		

?*hitz*

Hitz horri duten aginduen zerrada emango du.

In[7]:= ?*Map*

System'	ColorMap	Map	MapAll	MapAt	MapIndexed
		Map	MapAll	MapAt	MapThread
				ParallelMap	

?*Q

Predikatu-funtzioen zerrenda emango du. Negative, NonNegative eta Positive predikatu-funtzioak ez dira agertzen.

Out[10]= $6 - 2x + 3x^2 + x^3$

?@* Iker berezien eta \$ edo letra x-hez hasten diren aginduen zerrenda emango du.

?In Saio horretako sarrera guztien zerrenda emango du.

?Out Izen batetik duen balioa ezabatuko du.
=:

In[11]:= a =.

In[12]:= a

Out[12]= a

?Out Saio horretako irteera guztien zerrenda emango du.

Head[expr] Adierazpenaren mota itzuliko digu.

In[8]:= Head[3 - 2 I]

Out[8]= Complex

SetOptions[\$Output, PageWidth -> 45]
Orrialdearen zabalera aldatuko du. Jatorriz, 78 karaktereko zabalera du.

1.2.3 Gelditzreak

Agindu hauiek kalkuluak eten egingo ditugu:
Alt . PGetan kalkuluak eteteko.

<komandoa>-

Macintoshetan kalkuluak eteteko.

<kontrola>-c

UNIXean kalkuluak eteteko.

<kontrola>-pausa

MS-DOSean kalkuluak eteteko.

1.2.4 Esleipenak Aldagai edo adierazpenei behin behin betiko balioak eramateko erabil daitez-keen aginduak dira.

= Izen bat bali oinko bat esleituko digu. Sarrera batean izena agertzen den bakoitzean bali hori esleituko dio. Balio hori gordeko da kontrako agindua eman arte.

In[9]:= a = $x^3 + 3x^2 - 2x - 1$
Out[9]= $-1 - 2x + 3x^2 + x^3$
In[10]:= a + 7

?@* Izen batetik duen balioa ezabatuko du.
=: Izen bat bali aldakor bat esleituko digu. Ez du irteerarik sortzen. Izen hori deitzen denean, eskuineko balioa ebaluatuko da berriro.

In[13]:= a := RandomReal[]

Out[13]= 0.779942

In[14]:= a

Out[14]= 0.779942

?In[15]:= b := RandomReal[]

In[16]:= b

Out[16]= 0.956603

In[17]:= b

Out[17]= 0.470458

-> Aldagai bat erregela bat aplikatuko diogu adierazpen batean ordezkatzeko. Eskuineko erregela ez zaio atxekitzten aldagaiariurrengoginduetan.

In[18]:= Table[x, {4}] / . x -> RandomReal[]

Out[18]= { 0.503516, 0.503516, 0.503516, 0.503516 }

> Eskuineko erregela berriro ebaluatuko da, ezkerreko adierazpenea deitzen den bakoitzean. Ezkerreko adierazpenean aldagai bat eskunean agertzen den erregela aplikatuko dio.

In[19]:= Table[x, {4}] / . x -> RandomReal[]

Out[19]= { 0.718093, 0.406067, 0.607718, 0.632935 }

//. Eskuineko erregela behin eta berriro aplikatuko du.

In[20]:= c = $x^2 - 2x + 1$

Out[20]= $1 - 2x + x^2$

In[21]:= c / . { { x -> 0 }, { x -> 1 } }

Out[21]= { 1, 0 }

In[22]:= rules = { Log[x - y] :> Log[x] + Log[y], Log[x - k] :> k Log[x] };

In[23]:= Log[Sqrt[ab^c]] /. rules

```
Out[23]= 1/2 (Log[a] + c Log[b])
```

```
++i
```

i-ren balioa 1 handituko du eta balio berria emango du.

```
In[24]:= d = Pi
```

```
Out[24]=  $\pi$ 
```

```
In[25]:= ++d
```

```
Out[25]= 1 +  $\pi$ 
```

```
In[26]:= d
```

```
Out[26]= 1 +  $\pi$ 
```

```
i++
```

i-ren balioa emango du eta 1 gehituko dio.

```
In[27]:= d++
```

```
Out[27]= 1 +  $\pi$ 
```

```
In[28]:= d
```

```
Out[28]= 2 +  $\pi$ 
```

```
x += dx
```

x-ri dx gehituko dio eta *x*-ren balio berria itzuliko du.

```
In[29]:= x = 3
```

```
Out[29]= 3
```

```
In[30]:= x += 2
```

```
Out[30]= 5
```

```
x -= dx
```

x-ri dx kenduko dio eta *x*-ren balio berria iztuliko du.

```
In[31]:= x -= 1
```

```
Out[31]= 4
```

```
x *= c
```

x bider *c* egingo du eta *x*-ren balio berria itzulik du.

```
In[32]:= x *= -2
```

```
Out[32]= -8
```

```
x /= c
```

x zati *c* egingo du eta *x*-ren balio berria itzulik du.

```
In[33]:= x /= -2
```

```
Out[33]= 4
```

```
ToRules[ ]
```

== berdinaz -> esleipena bilurtuko du.

```
In[34]:= Roots[y^2 - 2y == 1, y]
```

```
Out[34]=  $y == 1 - \sqrt{2} \quad || \quad y == 1 + \sqrt{2}$ 
```

```
In[35]:= {ToRules[%]}
```

```
Out[35]= \{ \{ y -> 1 - \sqrt{2} \}, \{ y -> 1 + \sqrt{2} \} \}
```

```
$PrePrint = Short
```

Esleipena eginez emaitza guztia laburutuko dira, kontrako agindua eman arte.

```
$PrePrint = .
```

Aurreko agindua ezabatzeko esleipena da.

1.2.5 Testuinguruak

Testuingurua aldatzeko erabiliko ditugun aginduak hauek dira.

```
Begin[ "izena`" ]
```

Testuingurua aldatuko du eta *izena`* izeneko testuingurura joango da.

```
End[ ]
```

Oraingo testuingurua aurreko *Begin[]* baino lehenagokoa izango da.

```
BeginPackage[ "izena`" ]
```

Testuingurua aldatuko du eta *izena`* izeneko testuingurura joango da. Horrez gain, testuingurmak bilatzeko bidea (*\$ContextPath*) aldatuko du, berria eta *System`* testuinguruak bakarrik utzik ditu.

```
EndPackage[ ]
```

Azken *BeginPackage[]* baino lehenago testuingurura itzuliko da. Sortutako testuinguru berria hasierakoak baino lehen zerrrendatuko du.

Mathematica aplikazioak baditu agindu batzuk paketea kargatu berezietan gordeta. Agindu horietako bat erabil ahal izateko, aurretik paketea kargatu behar da.

```
Needs[ "Testuingurua`Paketearen izena`" ]
```

Testuinguruan dagoen paketea kargatu egingo du.

```
<<Testuingurua`Paketearen izena`
```

Testuinguruan dagoen paketea kargatu egingo du.

1.3 Aginduak

1.3.1 Programazioa

Agindu hauek zer ikusi dute kalkuluak etearekin edota jauzia egitearekin.

%

Aginduen argumentu gisa erabiltzen denean, azkeneko emaitza ordezkatuko du.

1.3. Aginduak

```
In[36]:=  $y^2 - 2y / . \%$ 
```

```
Out[36]=  $\{ -2(1 - \sqrt{2}) + (1 - \sqrt{2})^2, -2(1 + \sqrt{2}) + (1 + \sqrt{2})^2 \}$ 
```

%n
n zenbakiko irteera ordezkatuko du agindu baten argumentuan.

```
In[37]:= D[%9, x]
```

```
Out[37]=  $-2 + 6x + 3x^2$ 
```

```
Abort[]
```

Kalkulu bat bertan behera utzeko, eten bat sortuko du.
Aldagai globalak erabiliko ditu, baina aginduetan agertzen diren balio lokalekin.

```
Block[ { alda1, alda2, ... }, agin ]
```

Bukle batetik atteratzeko agindua da.
Aldagai globalak erabiliko ditu, baina aginduetan agertzen diren balio lokalekin.

```
Break[]
```

Aplikazioak prozesatzen segi dezala balio du.
Adierazpenaren n. posizioan dagoen adierazpena emango digu; "espr" lekuau adierazpenari nari eman diogun izena. Jari behar dugu.

```
In[38]:= c[[2]]
```

```
Out[38]=  $-2x$ 
```

espr[[n]]
Adierazpenaren n. posizioan dagoen adierazpena emango digu; "espr" lekuau adierazpenari nari eman diogun izena. Jari behar dugu.

```
In[39]:= c[[2,1]]
```

```
Out[39]=  $-2$ 
```

Goto[etik]

Label[etik] etiketaren ondoko agindura igaroko da aplikazioa.

```
In[ n ]
```

n zenbakiko sarrera berriro ebaluatuko du.

```
In[40]:= In[32]
```

```
Out[40]= -8
```

Range[m, n]

Zenbakia arrunten zerrenda bat emango du, 1etik n-ra.

```
In[41]:= InString[ {1, 2, 3} ]
```

1, 2 eta 3 zenbakietako sarrerak zerrendan aterako ditu.

```
In[42]:= Range[5, 13]
```

```
Out[41]= { 4, -8, 4 }
```

Interrupt[]

Eten bat sortuko du kalkuluaren. Programetan erabiltzen da.

```
Module[ { alda1, alda2, ... }, agin ]
```

Aldagaiak lokalak dira; aginduak bete eta gero, aurretik zeukanen balioa berreskurtatu daite.

Position[espr, elem]

Elementuak adierazpenean betetzan dituen posizioen zerrenda emango digu.

```
In[43]:= Position[c, x]
```

```
Out[43]= { { 2, 2 }, { 3, 1 } }
```

1.3.2 Zerrendak

Zerrenden erabilera zuzentzen duten aginduak ikusiko ditugu hemen, atalka sailkatuta.

1.3.2.1 Zerrendak eraikitzea

Zerrendak sortzeko aukera desberdinak ditugu.

{ , , , }

Zerrendak idazteko, giltzak eta komak erabiliko ditugu.

```
Array[ espr, n ]
```

n luzerako zerrenda osatuko du, espr[1] osagaietik.

```
In[44]:= Array[a, 7]
```

```
Out[44]= { a[1], a[2], a[3], a[4], a[5], a[6], a[7] }
```

```
Array[ espr, {m, n} ]
```

m × n dimentsioiko matrizea sortuko du.

```
In[45]:= Array[x^2, {3, 2}]
```

```
Out[45]= { { (x^2)[1, 1], (x^2)[1, 2] }, { (x^2)[2, 1], (x^2)[2, 2] }, { (x^2)[3, 1], (x^2)[3, 2] } }
```

Range[n]

Zenbakia arrunten zerrenda bat emango du, 1etik n-ra.

```
In[46]:= z = Range[13]
```

```
Out[46]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 }
```

Range[m, n]

Zenbakia arrunten zerrenda bat emango du, m-ri zenbakia arruntak batuz eta n zenbakia gainditu gabe.

```
In[47]:= Range[5, 13]
```

Out[47]= { 5, 6, 7, 8, 9, 10, 11, 12, 13 }

Range[m, n, d]

Zenbaki errealen zerrrenda bat emango du m -ri d zenbakia batuz, m -tik eta n zenbakia gainditu gabe.

In[48]:= Range[5, 13, 3]

Out[48]= { 5, 8, 11 }

Table[expr, {n}]

Adierazpena n aldiur errepikaturik duen zerrrenda sortuko du.

In[49]:= Table[m, {5}]

Out[49]= { m, m, m, m, m }

Table[expr, {i, imin,imax}]

Adierazpenean dagoen i indizea letik $imax$ -era ordeztatuko du zerrrenda bat sortzeko.

In[50]:= Table[i^3, {i, 5}]

Out[50]= { 1, 8, 27, 64, 125 }

Table[expr, {i, imin,imax}]

Adierazpenean dagoen i indizea $imin$ -etik $imax$ -era ordeztatuko du zerrrenda bat sortzeko.

In[51]:= Table[i^2/3, {i, 5, 11}]

Out[51]= { $\frac{25}{3}$, 12, $\frac{49}{3}$, $\frac{64}{3}$, 27, $\frac{100}{3}$, $\frac{121}{3}$ }

$\rightarrow i$ ere adierazten da baina

Table[expr, {i, imin,imax, iurr}]

Adierazpenean dagoen i indizea ordezkatuko du zerrrenda bat sortzeko, $imin$ -etik hasiz eta $iurr$ urratsak emanenez, baina $imax$ gainditu gabe.

In[52]:= Table[i^2/3, {i, 5, 11, 2}]

Out[52]= { $\frac{25}{3}$, $\frac{49}{3}$, 27, $\frac{121}{3}$ }

Table[{ expr1, expr2 }, {i, ...}]

Zerrrenden zerrrenda sortuko du. Bigarren argumentuak aurreko lau ankerak onartzen ditu.

Table[expr, {i, ...}, {j, ...}]

Indize bat baino gehiago erabiltzen dituen zerrrenden zerrrenda bat emango du, lehen indizea i da. Zerrrenda hori matrizeen txurran idatz daiteteke, i errenkada eta j zutabe.

In[53]:= Table[10 i + j, {i, 3}, {j, 4}]

Out[53]= { { 11, 12, 13, 14 }, { 21, 22, 23, 24 }, { 31, 32, 33, 34 } }

Table[expr, {m}, {n}]

$m \times n$ dimentsioko matrizea sortuko du.

In[55]:= Table[j^3, {5}, {3}]

Out[55]= { {j^3, j^3, j^3}, {j^3, j^3, j^3}, {j^3, j^3, j^3}, {j^3, j^3, j^3} }

In[56]:= MatrixForm[%]

$$\begin{pmatrix} j^3 & j^3 & j^3 \\ j^3 & j^3 & j^3 \\ j^3 & j^3 & j^3 \\ j^3 & j^3 & j^3 \end{pmatrix}$$

Out[56]//MatrixForm=

Zerrrenden elementuen ordena aldatzeko aukerak dira.

Permutations[zerr]

Zerrrendaren elementuen permutazio guztiak emango ditu.

In[57]:= Permutations[%48]

Out[57]= { {5, 8, 11}, {5, 11, 8}, {8, 5, 11}, {8, 11, 5}, {11, 5, 8}, {11, 8, 5} }

Reverse[zerr]

Zerrrendaren elementuen ordena alderantzikatuko du.

In[58]:= Reverse[%48]

Out[58]= { 11, 8, 5 }

RotateLeft[zerr, n]

Zerrrendaren lehenengo n elementuak atzean idatziko ditu.

In[59]:= RotateLeft[%47, 3]

Out[59]= { 8, 9, 10, 11, 12, 13, 5, 6, 7 }

RotateRight[zerr, n]

Zerrrendaren azkeneko n elementuak aurrean idatziko ditu.

In[59]:= RotateRight[%47, 3]

Out[59]= { 11, 12, 13, 5, 6, 7, 8, 9, 10 }

Sort[zerr]

Zerrrendaren elementuak ordena gorakorreean idatziko ditu.

In[60]:= Sort[%19]

Out[60]= { 0.406067, 0.607718, 0.632935, 0.718093 }

1.3.2.3 Zerrrendak aldatzea

Zerrrendetan elementuak sartu eta kendu ahal izateko aginduak dira.

Append[zerr, elem]

Zerrrendari elementua erantsiko dio amaieran. Ez du hasierako zerrenda aldatuko.

In[61]:= Append[z, 0]

Out[61]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }

In[62]:= z

1.3. Aginduak

```
Out[62]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 }
```

AppendTo[zerr, elem]
Zerrendari elementua erantsiko dio amaitera eta hasierako zerrenda aldatuko.

```
In[63]:= AppendTo[z, 0]
```

```
Out[63]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

```
In[64]:= z
```

```
Out[64]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

Delete[zerr, n]
Zerrendari n . elementua kenduko dio. Ez du hasierako zerrenda aldatuko.

```
In[65]:= Delete[z, 14]
```

```
Out[65]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 }
```

Drop[zerr, n]
Zerrendari lehenengo n elementuak kenduko dizkio. Ez du hasierako zerrenda aldatuko.

```
In[66]:= Drop[z, 5]
```

```
Out[66]= { 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

Drop[zerr, -n]
Zerrendari azkeneko n elementuak kenduko dizkio. Ez du hasierako zerrenda aldatuko.

```
In[67]:= Drop[z, -5]
```

```
Out[67]= { 1, 2, 3, 4, 5, 6, 7, 8, 9 }
```

Insert[zerr, elem, n]
Zerrendari elementua erantsiko dio n . posizioan. Ez du hasierako zerrenda aldatuko.

```
In[69]:= Insert[z, 7, 8]
```

```
Out[69]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

Prepend[zerr, elem]
Zerrendari elementua erantsiko dio hasieran. Ez du hasierako zerrenda aldatuko.

```
In[70]:= Prepend[z, 0]
```

```
Out[70]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

```
Out[72]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

Rest[zerr]
Zerrendari lehenengo elementua kenduko dio. Ez du hasierako zerrenda aldatuko.

```
In[73]:= Rest[z]
```

```
Out[73]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

Take[zerr, n]
Zerrendatik lehenengo n elementuak hartuko ditu. Ez du hasierako zerrenda aldatuko.

```
In[74]:= Take[z, 4]
```

```
Out[74]= { 0, 1, 2, 3 }
```

Take[zerr, -n]
Zerrendatik azkeneko n elementuak hartuko ditu. Ez du hasierako zerrenda aldatuko.

```
In[75]:= Take[z, -4]
```

```
Out[75]= { 11, 12, 13, 0 }
```

Drop[zerr, {m, n}]
Zerrendari m . eta n . elementuen artekoak kenduko ditu. m . eta n . elementuak barne.

```
In[76]:= Drop[z, {5, 11}]
```

```
Out[76]= { 4, 5, 6, 7, 8, 9, 10 }
```

Intersection[zerr1, zerr2, zerr3, ...]
Zerrenda desberdinen elementuak konbinatzeko aukera desberdinak dira.

```
In[77]:= Intersection[z, x, y]
```

```
Out[77]= { 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39 }
```

```
In[78]:= x = 2*Range[1,3]
```

```
Out[78]= { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26 }
```

```
In[79]:= Complement[z, x, y]
```

```
Out[79]= { 0, 1, 5, 7, 11, 13 }
```

Complement[zerr1, zerr2, zerr3, ...]
Zerrenda guztietai batetara dauden elementuak emango ditu, ordenaturik.

```
In[80]:= Complement[z, x, y]
```

```
Out[80]= { 6, 12 }
```

Join[zerr1, zerr2, zerr3, ...]
Zerrenda guztien elementuak emango ditu, zerrendak bata bestearen ondoren idatzi, elementuen ordena aldatu gabe eta errepikapenekin.

```
In[81]:= Join[z, x, y]
```

PrependTo[zerr, elem]
Zerrendari elementua erantsiko dio hasieran eta hasierako zerrenda aldatuko du.

```
In[71]:= PrependTo[z, 0]
```

```
Out[71]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

AppendTo[zerr, elem]
Zerrendari elementua erantsiko dio hasieran eta hasierako zerrenda aldatuko du.

```
In[72]:= AppendTo[z, 0]
```

```
Out[72]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0 }
```

```
Out[81]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,
```

```
Union[ zerr1, zerr2, zerr3, ... ]
```

Zerrendaren guztien elementu guztia emango ditu, errepikatu gabe eta ordenaturik.

```
In[82]:= Union[z, x, y]
```

```
Out[82]= { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 21, 22, 24, 26,
```

```
27, 30, 33, 36, 39 }
```

1.3.2.5 Zerrenden elementuak

Zerrendatik elementuak lortzeko aukera emango digute.

```
First[ zerr ]
```

Zerrendaren lehenengo elementua emango du.

```
In[83]:= First[z]
```

```
Out[83]= 0
```

```
Last[ zerr ]
```

Zerrendaren azkeneko elementua emango du.

```
In[84]:= Last[z]
```

```
Out[84]= 0
```

```
zerr[[ i ]]
```

Zerrendaren *i*. elementua emango du.

```
In[85]:= z[[6]]
```

```
Out[85]= 5
```

1.3.2.6 Kalkuluak zerrendekin

Agindu batelk Listable ezagarrria duenean, zerrenda bati aplikatzeko zaio, eta zerrenda berri bat sortu.

```
AppendTo[ Attributes[Agin], Listable ]
```

Zerrendaren attributuei Listable erantsiko dio.

```
Apply[ Plus, zerr ]
```

Zerrendaren elementuen batura emango du.

```
In[86]:= Apply[ Plus, z ]
```

```
Out[86]= 91
```

```
Map[ agin, zerr ]
```

Agindua zerrendaren elementu bakoitzari aplikatuko dio.

```
In[87]:= Map[Sqrt, z]
```

```
Out[87]= { 0, 1,  $\sqrt{2}$ ,  $\sqrt{3}$ , 2,  $\sqrt{5}$ ,  $\sqrt{6}$ ,  $\sqrt{7}$ ,  $2\sqrt{2}$ , 3,  $\sqrt{10}$ ,  $\sqrt{11}$ ,  $3\sqrt{3}$ ,  $\sqrt{13}$ , 0 }
```

```
Map[ agin, espr ]
```

Agindua batugai bakoitzari aplikatuz sortzen den adierazpena emango du.

```
In[88]:= Map[Sqrt, a + b + c + d]
```

```
Out[88]=  $\sqrt{a} + \sqrt{b} + \sqrt{c} + \sqrt{d}$ 
```

```
Sum[ zerr[[ i ]], f, Length[zerr] ]
```

Zerrendaren elementuen batura emango du.

```
In[89]:= Sum[x[[i]], i, Length[x]]
```

```
Out[89]= 182
```

1.3.2.7 Zerrendak formateatzea

```
ColumnForm[ zerr ]
```

Zerrendaren elementuak zutabean idatziko ditu.

```
In[90]:= ColumnForm[%48]
```

```
Out[90]= 

|   |   |    |
|---|---|----|
| 5 | 8 | 11 |
|---|---|----|


```

```
ColumnForm[ zerr, Center ]
```

Zerrendaren elementuak zutabean eta zentraturik idatziko ditu.
Hemendik aurrera *x*, *y* eta *z* zerrendak ezabatuko ditugu.

1.3.3 Baldintzak

```
If[ bald, egor, gor[], ebeb ]
```

Baldintza betetzen bada, egiazko gorputza itzuliko du; baldintza ez bada betetzen, gorputz

faltzen itzuliko du; eta baldintza ezin bada egiazta, "ezbataezbestea" itzuliko du.

```
In[91]:= f[x_]:= If[x > 0, 1, -1]
```

```
In[92]:= f[-3]
```

```
Out[92]=-1
```

```
In[93]:= f[]
```

Greater::nord: Invalid comparison with I attempted. >>

```
Out[93]= If[ > 0, 1, -1]
```

Adierazpena ebahatutakoan, haritzen duen forma forma horietako batekin bat datorrean, formari dagokion balioa itzuliko du.

```
In[94]:= g[x_]:= Switch[Mod[x, 3], 0, a, 1, b, 2, c]
```

```
In[95]:= g[538]
```

```
Out[95]= b
```

Which[bald1, balio1, bald2, balio2, ...]

Egiazaten den lehenengo baldintzari dagokion balioa itzuliko du.

```
In[96]:= h[x_]:= Which[x > 1, x - 1, x < -1, x + 1, True, 0]
```

```
In[97]:= h[-2]
```

1.3. Aginduak

```
Out[97]:= -1
In[98]:= h[0]
Out[98]:= 0
```

1.3.4 Iterazioak

Clear[alda1, alda2, ...]

Aldagaien balioak memoriatik ezabatuko ditu.

Do[expr, {max}]

Adierazpena max aldiz ebaluatuko du, baina ez du ezer itzuliko. Agindu bat denean, bete egindo du.

```
In[99]:= Do[a^3, {3}]
In[100]:= Do[Print[a^3], {3}]
```

```
a^3
a^3
a^3
In[101]:= Do[Print[a^3], {a, 3}]
```

1

8

27

In[102]:= For[i = 3, i < 7, i++, Print[i]]

3

4

5

6

For[hasi, bald, gehi, gorp]
Gorputza itzuliko du, indizearen hasieratik, indizearen balioak gehikuntza handituz eta baldintza egiaztatzen den bitartean.

In[103]:= Product[i + 1, {i, 4}]

4

5

6

Product[expr, {i, min, max}]
i indizea letik max-era aldatzean, adierazpenak hartzen dituen balioak batuko ditu.

```
In[103]:= Product[1/(j^2(i + 1)^2), {i, 1, Infinity}, {j, 1, j}]
In[106]:= Sum[1/(j^2(i + 1)^2), {i, 1, Infinity}, {j, 1, j}]
Out[106]:= Sum[π^4/120, {i, 1, 3}, {j, 1, j}]
In[107]:= Sum[1/((j + a)^2(i + a)^2), {i, 1, 3}, {j, 1, j}]
Out[107]:= 1/(1+a)^4 + 1/(2+a)^4 + 1/(3+a)^4 + 1/(4+a)^4 + 1/(2+a)^2 + 1/(3+a)^2 + 1/(4+a)^2/(3+a)^2 + 1/(2+a)^2/(3+a)^2 + 1/(3+a)^2/(4+a)^2
```

Sum[expr, {i, imin,imax}, {j, jmin, jmax}]
i eta j indizeak min-etaik max-era aldatzean, adierazpenak hartzen dituen balioak batuko ditu.

Out[97]:= -1

In[98]:= h[0]

Out[98]:= 0

```
In[106]:= Sum[π^4/120, {i, 1, 3}, {j, 1, j}]
In[107]:= Sum[1/((j + a)^2(i + a)^2), {i, 1, 3}, {j, 1, j}]
Out[107]:= 1/(1+a)^4 + 1/(2+a)^4 + 1/(3+a)^4 + 1/(4+a)^4 + 1/(2+a)^2 + 1/(3+a)^2 + 1/(4+a)^2/(3+a)^2 + 1/(2+a)^2/(3+a)^2 + 1/(3+a)^2/(4+a)^2
```

1.3.4 Iterazioak

While[bald, gorp]
Gorputza itzuliko du baldintza betzetzen den bitartean. Gorputzak (..., +i) itxura du.

While[bald, gorp]
Gorputza itzuliko du baldintza betzetzen den bitartean. Gorputzak (..., +i) itxura du.

In[108]:= n = 21

Out[108]:= 21

In[109]:= While[(n = Floor[n/2]) ≠ 0, Print[n - 1]]

9
4
1
0

1.4 Aljebra

1.4.1 Aritmetika

+	Batzuketa	-	Kenketak	*	edo hutsunea	Biderketa
/	Zatiketa	^	Berreketak	!		Faktoriala
&&	Eta		Edo	!		Eza

```
In[110]:= 12345 + 67890
Out[110]:= 80235
In[111]:= -55545
Out[111]:= -55545
In[112]:= 888102050
Out[112]:= 888102050
In[113]:= 12345 67890
Out[113]:= 838102050
In[114]:= 12345/67890
Out[114]:= 823/4526
In[115]:= 0.181838
Out[115]:= 0.181838
In[116]:= 27^4
Out[116]:= 479001600
In[117]:= 479001600
Out[117]:= 12!
In[118]:= 12! > 2^12 && 12! < 6^12
Out[118]:= 12! > 2^12 && 12! < 6^12
In[119]:= 12! > 2^12
Out[119]:= 3 > 2^12
In[120]:= 3 != 2
Out[120]:= True
Out[119]:= False
Out[118]:= True
```

“Txikiago ... baino”, emaitza egiazkoak (True) edo faltsoa (False) izango da.

```
In[121]:= 3 < 4
Out[121]:= True
```

“Txikiago edo berdin”, emaitza egiazkoak (True) edo faltsoa (False) izango da.
<=

```
In[105]:= Sum[a^2i/(2i), {i, 5}]
Out[105]:= a^2 + a^4 + a^6 + a^8 + a^10
```

In[122]:= $3 - 1 \leq 1 + 1$
 Out[122]= True

19

> "Handiago ... baino", emaitza egiazkoa (True) edo falsua (False) izango da.

In[123]:= $10! > 2^{2^5}$
 Out[123]= False

>=

"Handiago edo berdin", emaitza egiazkoa (True) edo falsua (False) izango da.

In[124]:= $2 >= \sqrt{4}$
 Out[124]= True

==

Berdintza baten bi atalak berdinak diren edo ez adieraziko du, berdinak direnean True emango du, bestela False. Erabaki ezin dueanean, berdintza bera emango du.

In[125]:= $3! == 7$
 Out[125]= False

Abs[zenb]

Zenbakien balio absolutua edo moduluua emango du.

In[126]:= Abs[$2x^2 - 5x + 1$ /. $x \rightarrow 1$]
 Out[126]= 2

In[127]:= Abs[$1 - 3$ I]
 Out[127]= $\sqrt{10}$

Arg[zenb]

Zenbakien argumentu nagusia emango du.

In[128]:= Arg[$1 + \text{Sqrt}[3]$ I]
 Out[128]= $\frac{\pi}{3}$
 In[129]:= N[%]
 Out[129]= 1.0472

BaseForm[zenb, oina]

Zenbakia onarri horretan idatziko du.
 $b^{\wedge} zenb$

Zenbakia baino handiagoa edo berdina den zenbaki oso txikiena emango du.

Ceiling[zenb]
 Zenbakia baino handiagoa edo berdina den zenbaki oso txikiena emango du.
 In[130]:= Ceiling[N[1 + Sqrt[98]]]
 Out[130]= 11

20

Chop[zenb]
 Otoki oso hurbil dagoen zenbakia 0 bjhurtuko du.

Conjugate[zenb]
 Zenbaki kompleku baten konjugatua emango du.

In[131]:= Conjugate[$1 - 3$ I]
 Out[131]= $1 + 3$ I

$\frac{\pi}{180} = 0,017453292$ zenbakia erabiltzen du graduetatik radianetara pasatzeko.

Divisors[n]
 n zenbakaren zatitzaleen zerrenda emango du.

In[132]:= Divisors[12345]
 Out[132]= { 1, 3, 5, 15, 823, 2469, 4115, 12345 }

E
 $e = 2,718281828$.

Factorial[n]
 n zenbakien faktoriala ($n!$) emango du.

In[133]:= Factorial[12]
 Out[133]= 479001600

FactorInteger[n]
 n zenbakien zatitzale lehenak emango ditu beren berretzaileekin.

In[134]:= FactorInteger[18]
 Out[134]= { { 2, 1 }, { 3, 2 } }

Floor[zenb]

Zenbakien zati osoa emango du.
 $In[135]:= Floor[N[1 + Sqrt[98]]]$
 Out[135]= 10

GCD[n1, n2, ...]
 n_1, n_2, \dots zenbakien zatitzale komun handiena (z_{kh}) emango du.

In[136]:= GCD[128, 144, 312]
 Out[136]= 8

GoldenRatio
 $\frac{1 + \sqrt{5}}{2} = 1,618033989$ urrezko arrazoia da. Alderantzikoa, 0,618033989 da.

1.4. Aljebra

21

I
 i unitate irudikaria da.

Im[zenb]
 Zenbakiko kompletxuaren zati irudikaria emango du.

In[137]:= **Im[x / . Solve[x^2 + x + 1 == 0, x]]**
 Out[137]= $\left\{ -\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2} \right\}$

IntegerDigits[n, b]
 n zenbakiek b oinarriz duen adierazpenaren digituen zerrenda emango du.

LCM[n1, n2, ...]
 n_1, n_2, \dots zenbakien multiplio komun txikiena (mkt) emango du.

In[138]:= **LCM[14, 25, 60]**
 Out[138]= 2100

Max[x1, x2, ..., xn]
 x_1, x_2, \dots, x_n Zerrendako zenbaki handiena emango du.

Min[x1, x2, ..., xn]
 x_1, x_2, \dots, x_n Zerrendako zenbaki txikiena emango du.

Mod[m, n]
 $\frac{m}{n}$ zatiketaren hondarra emango du.

In[139]:= **Mod[12386, 23]**
 Out[139]= 12

N[expr]
 Adierazpenaren zenbakizko balio hurbildua kalkulatuko du.

In[140]:= **N[Sqrt[3]/2]**
 Out[140]= 0.866025403784439

N[expr, n]
 Adierazpenaren zenbakizko balio hurbildua kalkulatuko du, n zifra zehatzekin.

In[141]:= **N[Sqrt[3]/2, 15]**
 Out[141]= 0.866025403784439

Or[expr1, expr2]
 Adierazpenen artean "edo" eragile logikoa jarriko du.

Pi
 $\pi = 3,141592654.$

Precision[expr]
 Adierazpenaren zehatzasuna, zifra-kopuru zehatzza, emango du.

Prime[n]
 n . zenbaki lehena emango du.
 In[142]:= **Prime[121]**
 Out[142]= 661

PrimePi[n]
 n baino txikiagoak edo berdinak diren zenbaki lehenen $\pi(n)$ kopurua emango du.

Quotient[n, m]
 $\frac{n}{m}$ zatiketaren zatidura emango du.
 In[143]:= **Quotient[12386, 23]**
 Out[143]= 538

Rationalize[zenb]
 Koma mugikorreko zenbakia zatiki bihurtuko du.
 In[144]:= **Rationalize[1.0472] (= Rationalize[%129])**
 Out[144]= $\frac{1309}{1250}$

Ref[zenb]
 Zenbaki kompletxuaren zati erreala emango du.
 In[145]:= **Ref[x / . Solve[x^2 + x + 1 == 0, x]]**
 Out[145]= $\left\{ -\frac{1}{2}, -\frac{1}{2} \right\}$

Round[zenb]
 Zenbakitik gertuen dagoen zenbaki osoa emango du. Zenbaki konplexuetan, zati erreala eta irudikaria bakoitzera aldetik biribilduko ditu.
 In[146]:= **Round[3/(1 + 2 I)]**
 Out[146]= 1 - i

Sign[expr]
 Adierazpenaren zeinua emango du: 1 positibotarako, -1 negatibotarako eta 0 zero zenbakirako.
 In[147]:= **Sqrt[zenb]**
 Zenbakaren erro karratu positibo emango du.
 In[147]:= **Sqrt[(1 + 3)^3 - 4]**
 Out[147]= $2\sqrt{15}$

Xor[expr1, expr2]
 Adierazpenen artean "ala" eragile logikoa jarriko du.

1.4.2 Matrizeak

Matrizeekin gehien erabiliko ditugun aginduak dira.

```
{ { , , , } , { , , , } , ... , { , , , } }
```

Matrizea zerrrenda batzuen zerrrenda gisa idatziko du.

```
m[[ i, j ]]
```

Matrizearen i . errenkada eta j . zutabeko elementua emango digu (Indexazioa).

```
Det[ matr ]
```

Matrizearen determinantea kalkulatuko du.

```
DiagonalMatrix[ al, ... , an ]
```

Osagai horiekin matrize diagonala eraikiko du.

```
Dot[ matr1, matr2 ] edo matr1 . matr2
```

matr1 eta matr2 matrizeen arteko biderketa egingo du.

```
Eigenvalues[ matr ]
```

Matrizearen autobaloioak kalkulatuko ditu.

```
Eigenvectors[ matr ]
```

Matrizearen autobektoreak kalkulatuko ditu.

```
IdentityMatrix[ n ]
```

$n \times n$ ordenako identitate (unitate) matriza emango du.

```
Inverse[ matr ]
```

Matrizearen alderantzikro matriza kalkulatuko du.

```
MatrixForm[ expr ]
```

Adierazpena matrizea gisa idatziko du.

```
Minors[ matr, k ]
```

Matrizearen k ordenako minoren matriza emango du.

```
NullSpace[ matr ]
```

Matrizearen espazio nuluaren oinarriko bektoreak emango ditu.

```
RowReduce[ matr ]
```

Matrizearen forma laburtua emango du errenkadekiko.

SingularValues[matr]

matr = $U^t M V$ deskomposaketaren U eta V matrize ortogonalak eta M matrize diagonala emango ditu.

Transpose[matr]

Matrizearen matrize iraulia emango du.

1.4.3 Ekuazioak

Ekuazio mota desberdinak ebazteko aginduak dira.

```
FindRoot[ espr1 == espr2, {x, x0} ]
```

Bi adierazpenen arteko berdintza (ekuazioa) betetzen duten soluzioetako bat bilatuko du, $x = x_0$ balioitik hasita.

```
In]:= FindRoot[Sin[x] == Exp[x], {x, 1}]
```

```
Out]= {x -> -6.28131}
```

```
FindRoot[ espr, {x, x0} ]
```

Adierazpena = 0 ekuazioa betetzen duten soluzioetako bat bilatuko du, $x = x_0$ balioitik hasita.

```
In]:= FindRoot[x - Exp[x], {x, 1}]
```

```
Out]= {x -> -6.28131}
```

```
LinearSolve[ matr, bek ]
```

matr . $\mathbf{x} = \mathbf{bek}$ ekuazioaren soluzioa den \mathbf{x} bektorea emango du.

```
In]:= LinearSolve[ { { 1, 0 }, { 2, 3 }, { 1, 2 } }
```

```
Out]= { 1, 0 }
```

```
NRoots[ espr1 == espr2, x ]
```

Berdintza (ekuazio polinomikoa) betetzen duten soluzionen balio hurbilduak emango ditu.

```
In]:= NRoots[2x^4 - 2x^3 + 3 == 0, x]
```

```
Out]= x == -0.176605 - 1.20282i || x == -0.176605 + 1.20282i || x == 1.35321
```

```
NSolve[ ekua, x ]
```

Ekuazio polinomikoa x aldagaiarekiko ebatzeko du.

```
In]:= NSolve[2x^4 - 2x^3 + 3 == 0, x]
```

```
Out]= { x -> -0.589053 - 0.744147i, {x -> -0.589053 + 0.744147i}, {x -> 1.08905 - 0.69229i}, {x -> 1.08905 + 0.69229i} }
```

```
NSolve[ { ekua1, ekua2, ... }, {x1, x2, ... } ]
```

Ekuazio-sistemanen aldagaien zenbakizko ebazpena emango du.

```
In]:= NSolve[x^2 - y^2 == 1, x - y == 4, {x, y}]
```

```
Out]= { {x -> 2.125, y -> -1.875} }
```

1.4. Aljebra

Reduce[expr, alda]
 Reduktua erabiltzen da aldagaiaren kalkuluaren ebatzitzat. Emandako adierazpena laburuko du aldagaiarekiko ekuaazioak edo inekuaazioak ebatzitzat. Eta zenbatzaileak ezabatuz.

```
In]:= Reduce[x^2 - y^2 == 1, {x, y}]
Out]:= y == -Sqrt[-1 + x^2] || y == Sqrt[-1 + x^2]
```

Reduce[{ expr1, expr2, ... }, {x1, x2, ...}]
 Reduktua adierazpenak laburuko du aldagaiarekiko ekuaazioak edo inekuaazioak ebatzitzat. Emandako adierazpenak laburuko du aldagaiarekiko ekuaazioak eta zenbatzaileak ezabatuz.

```
In]:= Reduce[x^2 - y^2 == 1 && x > 0 && y > 0, {x, y}, Reals]
Out]:= x > 1 && y == Sqrt[-1 + x^2]
```

Roots[ekua, alda]
 Polinomioaren aldagaiarekiko erroak emango ditu.

Ekuazio polinomikoa laburuko du, polinomioaren maila 4 edo txikiagoa izanik.

```
In]:= Roots[2x^3 - x^2 - 3x - 1 == 0, x]
Out]:= x == 1/(1 - Sqrt[5]) || x == 1/2(1 + Sqrt[5]) || x == -1/2
```

Solve[poli == 0, x]
 Polinomioaren erroak bilatuko ditu, polinomioaren maila 4 edo txikiagoa izanik.

```
In]:= Solve[x^2 + ax + b == 0, x]
Out]:= {x -> 1/2(-a - Sqrt[a^2 - 4b])}, {x -> 1/2(-a + Sqrt[a^2 - 4b])}
```

Solve[{ eku1, eku2, ..., ekuN }]
 Ekuazio-sistema ebatziko du ekuazioetan agertzen diren aldagai guztiakik.

```
In]:= Solve[{x^2 - y^2 == 1, x - y == 4, {x, y}}
Out]:= {x -> 17/8, y -> -15/8}
```

Zenbakitzalearen eta izendatzalearen faktore komunak kenduko ditu.

```
Cancel[ expr ]
In]:= Cancel[(x^3 + 1)/(x^2 - 1)]
Out]:= 1/(x - 1) - 1/(x + 1)
```

1.4.4 Polinomioak

Apart[expr]
 Adierazpen frakzionarioa frakzio baktunen batuketan deskomposatuko du.

```
In]:= Apart[1/(2 - 3x + x^2)]
Out]:= 1/(2 + x) - 1/(-1 + x)
```

Cancel[expr]
 Zenbakitzalearen eta izendatzalearen faktore komunak kenduko ditu.

```
In]:= Cancel[(x^3 + 1)/(x^2 - 1)]
Out]:= 1/(x - 1) + x^2/(x + 1)
```

Collect[expr, {x1, ..., xn}]
 Berrekizun eta berretzaile berak dituzten gaiak bilduko ditu.

```
In]:= Collect[(1 + a + b)^3, b]
```

Cyclotomic[n, x]
 x aldagaiaren n ordenako polinomio zirkulotomikoa emango du.

```
In]:= Cyclotomic[8, x]
Out]:= 1 + x^4
```

Expand[expr]
 Adierazpen aljebraikoak zabalduko du. Trigonometrian "Trig -> True" aukera eginez, formula trigonometrikoak zabalduko ditu.

```
In]:= Expand[(1 + 2x + y)(x + y)^3]
Out]:= x^3 + 2x^4 + 3x^2y + 7x^3y + 3xy^2 + 9x^2y^2 + y^3 + 5xy^3 + y^4
```

Factor[expr]
 Adierazpen aljebraikoak faktorizatuko du. Trigonometrian "Trig -> True" aukera eginez, formula trigonometrikoak faktorizatuko ditu.

```
In]:= Factor[x^8 - 1]
Out]:= (-1 + x)(1 + x)(1 + x^2)(1 + x^4)
```

InterpolatingPolynomial[{ {x1, f1}, {xn, fn}, ... }, x]
 Puntu horietatik igaroko den polinomioa emango du.

```
In]:= InterpolatingPolynomial[{{-3, 3}, {-1, 2}, {0, -1}, {1, 6}}, x]
Out]:= 3 + (3 + x) \left(\frac{1}{2} + (1 + x) \left(\frac{5}{6} + \frac{35 x}{24}\right)\right)
```

In]:= Expand[%]
Out]:= -1 + 13x/24 + 5x^2 + 35x^3/24

PolynomialQuotient[poli1, poli2, x]
 Polinomioen arteko $\frac{poli1}{poli2}$ zatiketaren zatidura emango du, x aldagalko polinomiotzat hartuta.

```
In]:= PolynomialQuotient[x^3 + xy^2 + x^2y + y^3, x + y, x]
Out]:= x^2 + y^2
```

PolynomialRemainder[poli1, poli2]

Polinomioen arteko $\frac{poli1}{poli2}$ zatiketaren hondarra emango du.

```
In]:= PolynomialRemainder[x^3 + xy^2 + x^2y + y^3, x + y, x]
Out]:= 0
```

PolynomialGCD[poli1, poli2, ...]
 Polinomioen zatiztale komun handiena (zkh) emango du.

Polinomioen zatiztale komun handiena (zkh) emango du.

In]:= PolynomialGCD[3 + 4x - 14x^2 + 4x^3 + 3x^4, 6 - x - 4x^2 - x^3]

```
Out]:= -3 + 2x + x2
In]:= Factor[%]
Out]:= (-1 + x)(3 + x)
```

Polinomioen multipo komun txikiena (mkt) emango du.

```
In]:= PolynomialLCM[3 + 4x - 14x2 + 4x3 + 3x4, 6 - x - 4x2 - x3]
```

```
In]:= Factor[%]
Out]:= -(-1 + x)2(2 + x)(3 + x)(1 + 3x)
```

```
Simplify[ expr ]
```

Aadierazpen aljebraikoa laburtuko du. Trigonometrian “Trig -> False” aukera eginez, formula ez du laburtuko.

```
In]:= D[(Log[-1 + Sqrt[2] - x] - Log[1 + Sqrt[2] + x])/2 Sqrt[2], x]
```

```
In]:= Simplify[%]
Out]:=  $\frac{\sqrt{2}}{2}$ 
```

```
Out]:=  $\frac{-1 + 2x + x^2}{-1 + 2x + x^2}$ 
```

```
Together[ expr ]
```

Frakzioen batuketa izendatzaile komunera laburtuko du.

```
In]:= Together[x2/(x2 - 4) - x/(x - 1) - (x + 1)/(x + 2) - 1]
Out]:=  $\frac{-3 + 6x - 2x^2}{(-2 + x)(-1 + x)}$ 
```

1.5 Kalkulua

Aplikazioak aurredefinirita ditu funtzioko asko; horiek guztia lehenengo letra lariz idatziko ditugu ($\text{Sin}[x]$), baita hitz kompositua denean ere ($\text{ArcSin}[x]$). Bestalde, guk funtziotako definitu nahi dugunean, honela definitutako dugu: $f[x, y, \dots] := \text{expr}$; askotan komeniko gaiak letra xehekin idatziko ditugu, kortexeten artean eta koma batez banaturik; guk definitutako funtziotan, aldagaiaren eskuin eta beheko aldean marratxa idatziko dugu, hutsunerik utzi gabe; aurredefinirako funtziotan, marratxoa ez dugu idatziko. Ebaluatzien denean, ez du irteerarik sortuko. Funtzioko ebaliatuak da agindu baten bidez deitzten den bakoitzean.

1.5.1 Funtzio elementala

```
Exp[x]
f(x) = ex funtzio esponentziala da.
```

```
Out]:= -3 + 2x + x2
In]:= Factor[%]
Out]:= (-1 + x)(3 + x)
```

Log[x], Log[b,x]

Funtzio logaritmikoa dira, $f(x) = \ln x$ logaritmo nepertarra eta $f(x) = \log_b x$ b oinarriko logaritmoa.

```
Sinh[x], Cosh[x], Tanh[x], Cot[x], Sec[x], Csc[x]
```

Funtzio trigonometrikoak dira: $\sin x$, $\cos x$, $\tan x$, $\cot x$, $\sec x$, $\csc x$.

```
ArcSin[x], ArcCos[x], ArcTan[x], ArcCot[x], ArcSec[x], ArcCsc[x]
```

Funtzio trigonometrikoen alderantzikro funtziok dira (funtzio zikloometrikoak): $\arcsin x$, $\arccos x$, $\arctan x$, $\arccot x$, $\sec x$, $\csc x$.

```
Sinh[x], Cosh[x], Tanh[x], Coth[x], Sech[x], CsCh[x]
```

Funtzio hiperbolikoak dira: $\sinh x$, $\cosh x$, $\tanh x$, $\coth x$, $\sech x$, $\csch x$.

```
ArcSinh[x], ArcCosh[x], ArcTanh[x], ArcCoth[x], ArcSech[x], ArcCsch[x]
```

Funtzio hiperbolikoen alderantzikro funtziok dira: $\text{arsinh } x$, $\text{arcosh } x$, $\text{artanh } x$, $\text{arccoth } x$.

```
Composition[f,g,h][x]
```

$(f \circ g \circ h)(x) = f(g(h(x)))$ funtzioko konposatura emango du.

```
In]:= f[x_]:= x2
In]:= g[x_]:= x + 1
In]:= h[x_]:= 2x - 1
```

```
In]:= Composition[f, g, h][x]
```

```
In]:= Composition[h, g, f][x]
Out]:= -1 + 21+x2
```

1.5.2 Diferentziala

```
D[ funt, x ]
```

Funtzioaren x adagaiarekiko deribatu partziala emango du.

```
In]:= D[x4 - 2x2y3 + y2, x]
```

```
Out]:= 4x3 - 4xy3
```

```
D[ funt, {x, n} ]
```

Funtzioaren x aldagaiaren n ordenako deribatu partziala emango du.

```
In]:= D[x4 - 2x2y3 + y2, {x, 2}]
Out]:= 12x2 - 4y3
```

```
D[ funt, x1, x2, ... ]
```

Funtzioaren x_1, x_2, \dots adagaiarekiko deribatu partzial gurutzatua emango du.

```
In]:= D[x4 - 2x2y3 + y2, x, y]
Out]:= -12xy2
```

15 Kalkuluua

D[funt, {x1, n1}, {x2, n2}, ...] ordenako deribatu partzial gurutzatua
 Funtzioaren x_1, x_2, \dots aldagaiarekiko $n_1 \cdot n_2 \dots$ emango du.

$f'(x)$ funtziaren deribatua emango du.

$\text{m}[] := \text{f}'[x]$

Funtzioaren minimoa bilatuko du (x_0, y_0, \dots) puntuaren inguruan.

卷之三

Puntu horietatik igaroko den eta deribatu horiek dituen funtziorik errazena emango du.

Limit [$\text{espr}, x \rightarrow x_0$] Adierazpenaren limitea kalkulatuko du x_0 puntuari. Aukeran “Direction $- > 1$ ” edo “Direction $- > -1$ ” sartuz gero, eskuin- eta ezker-limiteak kalkulatuko ditu, hurrenean hurren.

Normal [seri]
Seriezko garapenaren gai osagarria kenduko du.

Residue[funt, {x, a}]
bendorrakallinkatuoko du $x = a$ puntuau.

Series[expr, {x, x0, n}]
dierazporen Taylor-en n ordenako gaitapena kalkulatuko du x_0 puntuaren inguruan,
ai osagarria $0[x]^{n+1}$ moduan emango duelarik.

152 Integrala

Integrate[espr, x]
Adierazpenaren x -rekiko integral mugagabea kalkulatuko du.

```
Integrate[expr, {x, xmin, xmax}]
```

Adierazpenaren X-FERKU muzeoan
artean.

Integrate[expr, {x, xmin, xmax}, {y, ymin, ymax}]
 Adierazpenaren integral mugatu biloitza kalkulatuko du, integratuz y-rekiko lehenago eta x-rekiko gerogoa, y-ren *ymin* eta *ymax* balioen artean eta *x-ren* *xmin* eta *xmax* balioen artean.

NIntegrate[expr , { x , x_{\min} , x_{\max} }]
 Adierazpenaren x -rekiko integral mugatuaren balio hurbildua kalkulatuko du, x -ren x_{\min} eta x_{\max} balioen artean. Aldagai anitzekin ere balio du.

NIntegrate[expr, {x, xmin, x1, ..., xn, xmax}]
 Adierazpenaren integral mugatuaren balio hurbildua kalkulatuko du, integrazio-tartean
 $\int_{\text{min}}^{\text{max}}$ sinbeliarrauk daudela kontuan hartuz.

1.5.4 Ekuazio differentzialak

Dsolve[{ ekual1, ekua2, ... }, { y1[x1,...], y2[x1,...], ... } { x1, x2, ... }]

Ekuazio differentzialen $y_i[x_1, \dots]$ soluzioak emango ditu. $y_i[x_1, \dots]$ ezin izango dira funtzizkira erabili. Ekuazioen artean hastapen-baldintzak ere izan daitezke.

DSolve[{ ekual, ekua2, ... }, {y1, y2, ... } {x1, x2, ... }]
 Ekuazio diferentzialen y_i soluzioak emango ditu. y_i funtzio gisa erabili ahal izango dira.
 Ekuazioen artean hastapen-balintzaak ere izan daitezke.

NDsolve[{ekual1, ekua2, ... }, { y1[x], y2[x], ... } , {x, xmin, xmax}]

Ekuazio diferentzialen $y_i[x]$ soluzioak emango ditu [x_{min}, x_{max}] tartean egindako interplazioaren bidez. $y_i[x]$ ezin izango dira funtzio gisa erabili. Hastapen-baldintzak ekuazio artean jarri behar dira.

NDSolve[{ ekual1, ekua2, ... }, {y1, y2, ...}, {x, xmin, xmax}]
 Elkuazio differentzialen y_i soluzioak emango ditu $[xmin, xmax]$ tartean egindako interlazioaren bidez. y_i funtzio gisa erabili abal izango dira. Hastapen-baldintzak ekuziazartean jarri behar dira.

1 Crafkow

Environ Biol Fish (2007) 79:29–36

Irudiak planoan agertuko dira.

Mathematica aplikazioak aukera desberdinak ematen dizkigu irudiak egiteko. Bi eta hiru dimentsioakoak berciziko ditugu; ondoren, Graphics mota emango dugu; azkenik, irudiak eriteko erabil daitzkeen aukeraak zerrrendatuko ditugu.

Fit[datu, { funt }, { alda }]
 Datuen hurbilena emango du, aldagai horien funtzioen kombinazio linealen bidez (funtzioko ez dira soilik polinomioak izango).

ListPlot[{ y1, y2, ... }]

y_1, y_2, \dots balioak irudikatuko ditu x aldagaiaaren 1, 2, ... balioen gainean (segidak).

ListPlot[{ {x1, y1}, {x2, y2}, ... }]

(x_i, y_i) puntuak irudikatuko ditu.

ParametricPlot[{ fx, fy }, {t, min, max}]

t parametroaren fx eta fy funtziok osatzen duten kurba irudikatuko du.

ParametricPlot[{ { fx, fy }, { gx, gy }, ... }, {t, min, max}]

t parametroaren f, g, \dots funtzioen kurbak irudikatuko ditu.

Plot[expr, {x, xmin, xmax}, {t, min, max}]

Adierazpenaren irudia egingo du $[xmin, xmax]$ tartean, eta *aude* aukerak kontuan hartuz.

Plot[{ f1, ..., fn }, {x, xmin, xmax}, {t, min, max}]

f_1, \dots, f_n funtziok irudikatuko ditu $[xmin, xmax]$ tartean, eta *aude* aukerak kontuan hartuz.

1.6.2 Hiru dimentsioakoak

Irudiak espazioan agertuko dira, kutxa moduko batean. Irudiaren zentroa kuttaren zenitarekin bat dator; x aldagai orrikit gure aldera, y aldagai, gutxi gorabehera, ezkerretik eskuinera eta z aldagai bertikalean irudikatzen ditu. Aukeretik irudiaren posizioa alda daiteke.

ListPlot3D[zerr]

Zerrendaren balioekin hiru dimentsioko irudi bat egingo du.

ListPlot3D[zerr, itza]

Irudia itzal desberdinak irudikatuko du.

ParametricPlot3D[{ fx, fy, fz }, {t, min, max}]

t parametroaren fx, fy eta fz funtziok osatzen duten kurba irudikatuko du.

ParametricPlot3D[{ fx, fy, fz }, {t, tmin, tmax}, {u, umin, umax}]

t eta u parametroen fx, fy eta fz funtziok osatzen duten gainazala irudikatuko du.

ParametricPlot3D[{ { fx, fy, fz }, { gx, gy, gz }, ... }, {t, tmin, tmax}, {u, umin, umax}]

t eta u parametroen f, g, \dots funtziok osatzan duten gainazala irudikatuko du.

Plot3D[expr, {x, xmin, xmax}, {y, ymin, ymax}, {aude}]
 Adierazpenaren irudia egingo du $[xmin, xmax] \times [ymin, ymax]$ tartean, eta *aude* aukerak kontuan hartuz.

1.6.3 Graphics

Ohiko irudi geometriko batzuk egiteko aukera ematen digute.

Animate[graft], {t, tmin, tmax}]

GraphicsAnimation paketea satu eta gero, grafikoen animazioa egingo du, t -ren balioetarako.

Graphics[jato, aude]

Ondoko jatorrizkoak irudikatzeko balio du:

Circle[{x, y}, r]

(2D) (x, y) zentroko eta r erradioko zirkunferentzia irudikatuko du.

Disk[{x, y}, r]

(2D) (x, y) zentroko eta r erradioko zirkulua irudikatuko du.

Line[{ { x1, y1, z1 } }, ...]

(x_i, y_i, z_i) puntuetatik igarotzen den lerro poligonala irudikatuko du.

Point[{ x, y, z1 }], ...]

(x_i, y_i, z_i) koordenatuak puntuua irudikatuko du.

Polygon[{ { x1, y1, z1 } }, ...]

(x_i, y_i, z_i) puntuetatik igarotzen den poligono itxia irudikatuko du.

Rectangle[{xmin, ymin}, {xmax, ymax}]

(2D) Erpin horietako laukizuzena irudikatuko du.

Cuboid[{xmin, ymin, zmin}, {xmax, ymax, zmax}]

Prisma bat irudikatuko du.

ContourPlot[expr, {x, xmin, xmax}, {y, ymin, ymax}]

Hiru dimentsioko irudiaren maila-grafikoa emango du.

DensityPlot[f[x,y], {x, xmin, xmax}, {y, ymin, ymax}]

$f(x, y)$ funtziaren dentsitate-grafikoa emango du.

GraphicsArray[{ { graf1, ..., grafn }, ... , { grafm1, ..., grafmn } }]

Grafikoen matrize bat erakutsiko du.

ListContourPlot[zerr]

1.6. Grafiokoak

33

Zerrendari dagokion hiru dimentsioko irudiaren maila-grafikoa egingo du, maiaren arabera tono desberdinak erabiliz.

`Show[graf, auke]`

Bi edo grafiko gehiago batera erakutsiko ditu.

`Show[g1, g2, ...]`

Zenbait plot konbinaturik erakutsiko ditu.

1.6.4 Aukerak

Ez badugu ezer esaten, aukerak bi eta hiru dimentsioko irudietarako balioko du.

`AspectRatio`: Ardatzen arteko eskala emateko balio du.

-> $(\text{GoldenRatio})^{-1}$: 0,618033989... proportzioa erabiliko du.

-> `Automatic`: Altuera eta zabalera eskala berean irudikatuko ditu.

-> `r`: Ardatzen arteko eskala *r* izango da.

-> `False`: Ez ditu ardatzak irudikatuko.

`Axes`: Ardatzak irudikatzeko balio du.

-> `Automatic`

-> `True`: Ardatzak irudikatuko ditu.

-> `{False, True}` edo `{True, False}`: OY edo OX ardatza irudikatuko du,

-> `{x, y}`: OY edo OX ardatza irudikatuko ditu.

-> `hurrenez hurren`.

-> `False`: Ez ditu ardatzak irudikatuko.

`AxesEdge`: (3D) Ardatzen norabideak erabakitzeko balio du.

-> `Automatic`

-> `{x, y}`: Ardatzak *x* eta *y* puntuak erabiltzen.

-> `xy`: Ardatzak *x* eta *y* puntuak erabiltzen.

-> `yz`: Ardatzak *y* eta *z* puntuak erabiltzen.

-> `zx`: Ardatzak *x* eta *z* puntuak erabiltzen.

`AxesLabel`: Ardatzetaen izenak idazteko balio du.

-> `None`: Ez du izenik erabiliko.

-> `OY`: Ardatzetaen idaztako du aipatu izena.

-> `izen`: Ardatzetaen izenak idaztuko ditu.

-> `{x, y}`: Ardatzetaen izenak idaztuko ditu.

-> `{x, y, z}`: Ardatzetaen izenak idaztuko ditu.

`AxesOrigin`: (2D) Ardatzen arteko ebaki-puntua finkatzeko balio du.

-> `Automatic`

-> `{x, y}`: Ardatzak (*x*, *y*) puntutik igarotako dira.

-> `True`: Ardatzak aipatu estiloen agertuko dira.

`AxesStyle`: Ardatzetaen itxura definitzeko balio du.

-> `Automatic`

-> `estilox`: Ardatzak aipatu estiloen agertuko dira.

-> `{xesti, yesti}`: Ardatz bakoipta bere estiloen irudikatuko ditu.

`Boxed`: (3D) Irudia kutxa batean sartutako du.

-> `True`: Irudia kutxa batean sartutako du.

-> `False`: Ez du kuxxa irudikatuko.

`BoxRatios`: (3D) Ardatzen arteko eskala emateko balio du.

-> `{1, 1, 0.4}`: Eskala hori erabiliko du.

-> `Automatic`: Eskala koordenatuengen balioen arabera aukeratuko du.

-> `{rx, ry, rz}`: Aukeratu eskanan irudikatuko ditu ardatzak.

`BoxStyle`: (3D) Kutxaren itxura definitzeko balio du. Ikuhi `PlotStyle`.

`DefaultColor`: Letro eta puntuengen kolore osagarria erabiliko du.

-> `Automatic`: Hondoaren kolorearen kolore osagarria erabiliko du.

`DefaultFont`: Letra mota eta tamaina definitzeko balio du.

-> `$DefaultFont`

-> `{"mota", tama}`: Testua mota eta tamaina horietan idatziko du.

`Frame`: (2D) Irudiaren inguruan marko bat irudikatzeko balio du.

-> `False`: Ez du markoa irudikatuko.

-> `{True, False, True, False}`: Markoaren goiko eta beheko aldeak irudikatuko ditu.

`FrameLabel`: (2D) Markoaren ertzetan izenak idazteko balio du.

-> `None`: Ez du izenik erabiliko.

-> `{x, y}`: Izenak behoko eta ezkerreko ertzetan idatziko ditu.

-> `{x, y, z}`: Izan hori irudiaren gainean idatziko du, komatxoengen artean.

`FrameStyle`: (2D) Markoaren itxura definitzeko balio du.

-> `Automatic`

-> `estilox`: Irudiaren horretan irudikatuko ditu.

-> `{x, y}`: Izan hori irudiaren gainean idatziko du, komatxoengen artean.

`PlotLabel`: Irudiaren gainean izena idazteko balio du.

-> `None`: Ez du izenik erabiliko.

-> `"izen"`: Izan hori irudiaren gainean idatziko du, komatxoengen artean.

`PlotPoints`: Irudia egiteko erabiliko den puntu-kopuruaren finkatzeko balio du.

-> `25`: Aplikazioak 25 puntu erabiliko ditu irudiak egiteko.

-> `n`: Irudiak egiteko *n* puntu erabiliko ditu.

-> `15`: Irudiak egiteko $15 \cdot 15 = 225$ puntu erabiliko ditu.

-> `{nx, ny}`: Irudiak egiteko *nx* · *ny* puntu erabiliko ditu.

`PlotRange`: Irudian agertuko den OY ardatzaren zatiaren luzera aukeratzeko balio du.

-> `Automatic`: Puntu urrunak ez dira agertuko.

-> `All`: Irudiaren puntu guztialak agertuko dira.

-> `{min, max}`: OY ardatzeko [*min*, *max*] tartean egindako du irudia.

$\rightarrow \{ \{x_{\min}, x_{\max}\}, \{y_{\min}, y_{\max}\} \}$: Ardatz bakoitzaren tartearen egingo du irudia.

PlotRegion: Irudiak bete behar duen eskualdea definitzeko balio du.

\rightarrow Automatic

$\rightarrow \{ \{sx_{\min}, sx_{\max}\}, \{sy_{\min}, sy_{\max}\} \}$: Irudia eskualde horretan egingo du. Eskualdea mugatzeko balioak 0 eta 1 bitartekoak dira.

PlotStyle: (2D) Irudiko lerroen itxura definitzeko balio du.

\rightarrow Automatic

\rightarrow esti: Irudiaren lerroak estilo horretan egingo ditu.

$\rightarrow \{ \{\text{estil1}, \text{estil2}\}, \dots \}$: Irudiko lerroak irudikatu ahalak estilo horiek erabiliko ditu.

RotateLabel: (2D) Ardatz bertikaletan izenak jartzeko balio du.

\rightarrow True: Markoaren ardatz bertikalen izenak ibratuko ditu.

Ticks: Ardatzetako marratxoak irudikatzeko balio du.

\rightarrow Automatic

\rightarrow None: Ez ditu marratxoak irudikatuko.

Binomial[m, n]
 $\binom{m}{n}$ zenbaki binomiala emango du.

1.8 Bibliografia

Malaina, J. Luis eta Martin, Ana I.; Matematika aurreratuak *Mathematica*-rekin; EHU, 2000

Mota horretako zorizko zenbaki bat emango du, *min* eta *max* balioen artean. Mota hauek dira: Integer, Real, Complex.

1.7.1 Konbinatoria

Binomial[m, n]
 $\binom{m}{n}$ zenbaki binomiala emango du.

ViewCenter: (3D) Irudi osoaren zentroa aukeratzeko balio du.
 \rightarrow Automatic: Kutxa azkeneko irudiaren arabera zentratuko du.
 $\rightarrow \{x, y, z\}$: Azkeneko irudiaren zentroa (x, y, z) puntua izango da.

ViewPoint: (3D) Irudiaren ikuspuntua finkatzeko balio du.
 $\rightarrow \{1'3, -2'4, 2\}$: Ikuspuntu horretatik ikusten dira irudiak.
 $\rightarrow \{x, y, z\}$: Ikuspunturen kokapena da, kuxaren zentroarekiko.

1.7 Estatistika

Bi grafiko, barra-diagrama eta sektore-diagrama, eta bi agindu, randoma eta zenbaki binomialak, baino ez dugu hemen azalduko.

BarChart[zerr]

Zerrendari dagokion barra-grafikoa eraikiko du.

PieChart[zerr]

Zerrendari dagokion sektore-diagrama egingo du.

Random[]

Zorizko zenbaki bat emango du, 0 eta 1 balioen artean.

Random[mota, {min, max}]

B ikasleak 100 puntu lortu ditu; berez, azterketan 50 atera beharko luke ikasgia ginditzeko.
 C ikasleak 144 puntu lortu ditu; beraz, azterketan 6 atera beharko luke ikasgia ginditzako? Ez, minimoa 30 delako.
 Hiru kasuetan ikasgai osoaren ideia bat hartu dute ikasleek eta azterketa prestatzeko egora hobean daude.
 Orain, alderantziz gingo dugu. Azterketan 30 puntuko minimoa ateraz gero, zenbat puntu lortu beharko ditu ikasle batek lanetan? Erantzuna 120 puntu da. Hau da, lanetan 120 puntu eta azterketan 30 puntu lortuz gero, ikasgaiaren nota 5 izango da.

Tritoretsa-orduak erabil ditzakezue sortzen zaizkizuen zaldartzak argitzeko.

1. Tritoretsa-orduak erabil ditzakezue sortzen zaizkizuen zaldartzak argitzeko.

PRAKTIKAK

Laborategiko lehenengo saioetan *Latex* eta *Mathematica* aplikazioak erabiltzen ikasiko duzu. *Mathematica* aplikazioa kalkulu simbolika egiteko prestatuta dago. Aplikazio honen bidez kontzeptu batzuk landuko dituzu, alde teorikoa ulertzen saiatzeko. Hurrengo laborategietan hiru praktikak egin beharko dituzue. Praktikak egiteko, enuntziatu irakurriko ditzue eta egiten diren galderak erantzun beharko dituzue. Testuan proposatzen diren egingintzak egin beharko dituzue. Emaitzak gorde egin beharko dituzue eta, nahi duzuean, baina epeareen barruan, Moodle plataforma eta igo.

Praktiken zerrenda

1. a) Zenbaki konplexuen eragiketak. Adierazpen grafikoa
- b) Ekuazioen ebazpena zenbaki konplexuekin eta adierazpen grafikoa

2. Bezier-en kurbak
3. Serieen batura hurbildua eta errorearen bornapena

Idatzizko lana apunteak idaztean datza. Horrek esan nahi du arbeletik hartzen dituzuen apunteek Latex-en txukun idatzi behar dituzuela. Arbeletik hartutako apunteak osa daitezke, esaterako, teoremen edo propietateen frogak, adibide berrikin, ariketa-orriko arketekin, eta bururatzuen zaizkizuen kontuekin. Beti ere, kontuan hartu behar duzue ez ditzuela gehiegi lutzatu behar.
 Lana zoriz egokituko zaizue.
 Lanaren ldoa ikasgelaen ikusitakoa izango da.
 Lanaren luzera lauzpabost orrialdekoa izango da.
 Hauek dira gaien atalak:

1. gaia: Zenbaki multzoak
 - (a) Zenbaki arrazionalak. Zenbaki errealsak
 - (b) Zenbaki konplexuak
2. gaia: Topologia
 Espazio metrikoak. Espazio normadunak
3. gaia: Segidak
 - (a) Segidak. Segida konbergenteak
 - (b) Eragiketak. Indeterminazioak. Ebazpen-metodoak
4. gaia: Serieak
 - (a) Serieak. Serieen izaerak
 - (b) Gai positiboko serieak
 - (c) Serie baten batura hurbildua. Serie alternatuak

Lanaren balorazioan gaiaren zorroztasunaz gain, adierazpen matematikoen erabilera zuzena, euskara zuzena, aurkezpen txukuna eta abar ere hartuko dira kontuan. Lana egiten laguntzeko lehenengo asto berezioko eguna erabiliko ditzue. Lana ondo idatzeko, liburu hau gomendatzetan ditzuegu: *Zientzia eta teknikako euskara arantzeko gomentazioak*, Jose Ramon Etxebarria. Izenburua idatzi bilatzalean eta PDF formatuan agertuko zaizue. Bestalde, Eustalbar instalatzen badizue, hiztegiak, zuzentzailea, corpusak eta abar ere izango dituzne eskura.

Eta, edozein zalantza izanez gero, tutoretza-orduak dituzue kontu guztiak argitzeko.

Analisi Matematikoa

IKASGAI: ANALISI MATEMATIKOA
IRAKASLEA: Patxi Angulo (339 bulogoa)

IKASKETA

Bi bide izango duzue ikasleek ikasgai hau gainditzeko, ikasketa librea eta ikasketa bideratua. Ikasketa mota aukeratzeko azkeneko data bigarren aste berezia izango da. Dena dela, ikasketa bideratuan derrigorrez egin behar dira lan batzuk; beraz, horietako bat ez egitea nahiakoa da ikasketa bideratua baztertzeko. Hasteke, ikastle guztiek zudee ikasketa bideratuan. Norbaitek ikasketa librea egin nahi badu, nahiakoa da posta elektronikoaren bidez jakinaraztea niri. Ikasketa bideratua uztelko azkeneko data bat atuan mantentzen zareteela uztutuko dut. Ikasketa libreena eta bideratuarren artean erabakitzeko, oso garrantzitsua da lortutako puntuak kontuan hartzea; beraz, zenbat eta lehenago aurkeztu idazlana, gero eta lehenago jakingo duzue dagokizun nota.

Bi ikasuetan eskoletara etortzea komuni zaizue, bai ikasketa librea aukeratzzen duzuenoi, bai ikasketa bideratua aukeratzzen duzuenoi. Astean 6 ordu izango dituzue tutoretzan erabilitzeko.

1. Ikasketa librea

Zuek antolatuiko duzue noiz ikasi, nola ikasi, zer ikasi eta ahar une bakotzeen. Ikasketa librea aukeratzzen dituzen ikasleek ikasgai osoen ebaluazio globala lauhilekoaren bukaeran egingo duzue azterketa baten bidez. Ikasleek bi detaldi izango dituzue ikasgai gainditzeko, bata urtarrilean (urtarriak 22) eta bestea ekainean (ekainak 17).

Tutoretza-orduak erabil ditzakezue sortzen zaizkizuen zalduntzak argitzeko.

2. Ikasketa bideratua

Ikasketa bideratua aukeratzzen duzuen ikasleek lauhilekoan zehar lan batzuk eta bukaeran azterketa egingo dituzue derrigorrez. Hauek dira lauhilekoan zehar egin beharko dituzuen lanak:

(a) Lau ariketa eskatuko zaizkizue lauhilekoan zehar zoriz aukeratutako egunetan;

paperetan eta eskuz idatzita eman beharko dituzue. Ariketa bakoitzeko gehienez 5 puntu lortuko dituzue. Ariketa zuen izena aipatzen den momentuan eman beharko duzue, bestola aukera galduzatz emango zaizue. Ariketak momentuan azaltzen ari den gaiaren atalekoa izan behar du.

(b) Bi ariketa eskatuko zaizkizue zoriz aukeratutako egunetan; arbelean azaldiko dituzue eta momentuan zuzenduko dira. Ariketa bakoitzeko gehienez 20 puntu lortuko dituzue. Zuen izena aipatzen den momentuan atera beharko duzue arbelera, bestela aukera galduzatz emango zaizue. Komuni zaizue ariketa bat baino gehiago prestatzea, aurreko batek egindako ariketa ez errepikatzeko.

Bestela, arbelera deitua izan baino lehen ere atera zaizkete boluntario; horrek momentuan azaltzen ari den gaiaren atalekoa izan behar du beti. Papereko edo arbeleko ariketetan lortutako nota hobetzeko aukera izango duzue, arbelera boluntario ateraz gero; baina, gehienez 10 puntu hobetuko dituzue. Lehenastuna izango dute momentuan desituak direnek. Kasu batean ezin duzue aurreko nota hobetu: deitzen zaizuneean ez baduzue ariketa ematen edo arbelan egiten. Horrengain, papereko bi ariketa emateari edo arbeleko bat egiteari uko egiten badiozue, ebaluazio globalera pasatuko zarete zuzenean. Aureko bi lanetan, (a) eta (b), ariketaren zaitasuna kontuan hartuko da nota jartzean.

(c) Behoko zerrrendako hiru praktikak derrigorrez egin beharko dituzue laborategian, baita laborategiko saioetara joan ere. Praktika bakoitzeko gehienez 20 puntu lortuko dituzue. Oso kontuan hartuko da praktiketan egiten diren ariketen originaltasuna. Praktikak emateko epeak Moodle-n zehaztuko dira.

(d) Idatzizko lan bat egin beharko duzue zuek arkeletik kopiatzen dituzten apunteak txukundu eta osatzu. Idatzizko lanagatik gehienez 80 puntu lortuko dira. Jarraitu beharko diozue. Idatzizko lanagatik gehienez 80 puntu lortuko dituzue. Lehenengo astek berezko eguna idazten praktikatzeko erabiliko dugu. Lanen emateko azken eguna diagokion gaia bukatu eta bi astera izango da.

Aurreko lau lan horiekin 200 puntu lor ditzakezue.

Azterketa urtarrilean (urtarriak 22) izango da. Azterketan gutxienez 30 puntu lortu beharko dituzue. Lehen deialdiko azterketetan ez badituzue 30 puntu lortzen, aktan gutxiegia izango duzue eta bigarren deialdiko azterketa egin beharko duzue ekainean (ekainak 17); lanaren puntuazioa gordetxo zaizue, zuek eskatitz gero. Bigarren aukera horretarako, beraz, zuek erabaki beharko duzue lanaren nota gordetzea nahi duzuen, edo ez.

Ikasgaiari diagokion nota: l bada lauhilekoan zehar egindako lanen nota eta a azterketaren, ikasgaiaren nota formula honekin kalkulatuko da:

$$n = \frac{l + a}{30}.$$

Adibidez, pentzatu hiru ikaslek puntuazio hauek lortu dituztela lau ataletan:

	a	b	c	d	guztira
A ikaslea	4	10	30	30	74
B ikaslea	10	20	30	40	100
C ikaslea	14	30	40	60	144

A ikasleak 74 puntu lortu ditu; beraz, azterketetan 76 atera beharko luke ikasgia gainditzeko.

IRAKASLEA: Patxi Angulo Martín (339 bulegoa)

Analisi Matematikoa

2014-15eko programa-2

1. Gaia: Zerbakı-multzoak
 - 1.1. Zenbakı arruntak eta osoak.
 - 1.2. Zenbakı arrazionalak.
 - 1.3. Zenbakı errealkak.
 - 1.4. Zenbakı komplexuak.
 2. Gaia: Topologia
 - 2.1. Espazio metrikoak.
 - 2.2. Espazio normadunak.
 - 2.3. Multzo bornatuak.
 3. Gaia: Segidak \mathbb{R} multzoan
 - 3.1. Segidak. Segiden limiteak.
 - 3.2. Segida konvergentekak.
 - 3.3. Segiden arteko ergakietak eta limiteak. Indeterminazioak.
 - 3.4. Indeterminazioak ebazteko metodoak.
 - 3.5. Cauchy-ren segidak.
 4. Gaia: Serieak
 - 4.1. Serieak. Serieen izaerak.
 - 4.2. Gai positiboko serieak.
 - 4.2.1. Definizio eta propietateak.
 - 4.2.2. Konparaziozko irizpide orokorra.
 - 4.2.3. Konparaziozko irizpide orokorren aplikazioak.
 - 4.2.4. Serie baten batura hurbidua.
 - 4.3. Serie alternatuak.
 5. Gaia: Aldagai errealeko funtzioka. Jarraitutasuna
 - 5.1. Aldagai errealeko funtzioka.
 - 5.2. Funtzioen limiteak. Limiteen propietateak.
 - 5.3. Funtzioen arteko eragiketak eta limiteak.
 - 5.4. Infinitesimalk eta infinituak.
 - 5.5. Funtzio jarraituak.
 - 5.6. Funtzio jarraittuen propietateak.
- BIBLIOGRAFIA**
- Teoria**
- J. I. Barraqués estab.: *Analisi Matematikoa*, Pearson, Madrid, 2012
 - N. Piskunov: *Kalkulu Diferenziala eta Integrala*. 2. arg., UEU Bilbao, 2009
 - M. J. Zarate: *Matematika Orokorra I, 1. partea*. UEU, Bilbo, 1980
 - L. Abellanas, A. Galindo: *Métodos de Cálculo*. Mc Graw-Hill, Madrid, 1989
 - T. M. Apostol: *Análisis Matemático*. Ed. Reverté, Barcelona, 1977
 - F. Galindo etab.: *Guía práctica de cálculo infinitesimal en una variable*. Thomson, Madrid, 2003
 - F. García, A. Gutiérrez, *Cálculo Infinitesimal I, 1 eta 2*. Pirámide, Madrid, 1987
 - F. Granero, *Cálculo*. Mc Graw-Hill, Madrid, 1990
 - J. Martínez Salas: *Elementos de Matemáticas*. Valladolid, 1979
- Ariketak**
- J. Aizpuru, P. Angulo, *Bektoreak eta zenbakı komplexuak*. Elhuyar, Usurbil, 1994
 - P. Angulo, *Derribatuk eta integrak*. Elhuyar, Usurbil, 1994
 - P. Angulo, *Funtzien atzerazpen grafikoa*. Elhuyar, Usurbil, 1994
 - L. Abellanas, A. Galindo, *Métodos de Cálculo*. Mc Graw-Hill, Madrid, 1989
 - F. Ayres Jr., *Cálculo Diferencial e Integral*. Mc Graw-Hill, Mexiko, 1987
 - F. Granero, *Cálculo*. Mc Graw-Hill, Madrid, 1990
- Web-orriak**
- <http://zthiztegia.elhuyar.org/>
 - hiru.com matematika: <http://www.hiru.com/matematica>
 - WolframMathWorld: <http://mathworld.wolfram.com/>
 - Wikipedia
 - DivulgamAT: <http://divulganat.ehu.es/>
 - Maths online Gallery: <http://www.univie.ac.at/future.media/moe/galerie.html>

$$\sqrt{a+b} = \sqrt{\frac{1}{2}(a+b) + \frac{1}{2}(a-b)}$$

$\log_{10} 2 \approx 0.3010$

$\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{2}$

$a = 2$ $b = 0.3010$ $c = 0.3010$ $d = 0.3010$

