به نام خدا.

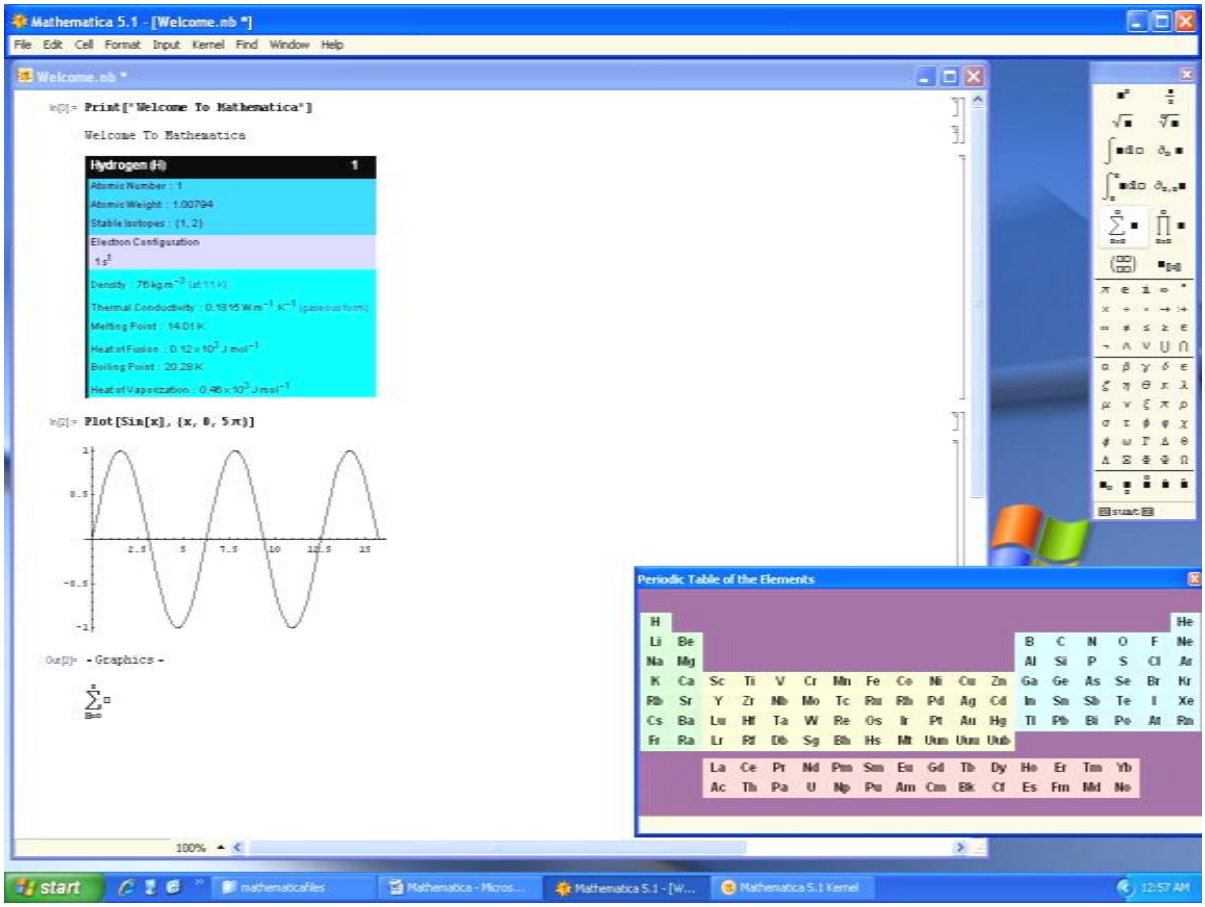
مقدمه ای بر

Mathematica

فهرست مطالب

مقدمه

اکثر عملیات ریاضی به کمک نرم افزار Mathematica با دستورهایی ساده براحتی قایل اجرا می باشد. این نرم افزار بعنوان یک زبان برنامه نویسی سطح بالا علاوه بر توانایی و امکانات زبان هایی چون C و Pascal دارای مزیتهای زیادی مثل دربرداشتن بسیاری از توابع، قبول ورودی (حتی توابع) بروشی ساده، رسم اشکال گرافیکی، متحرک سازی و ... می باشد. (شکل زیر)



برای انجام عملیات در این نرم افزار باید دستورات را وارد و سپس اجرا کنیم. هدف این جزوه بیان نحوه نوشتن دستورات است. اما برای اجرای یک دستور بعداز نوشتن آن باید کلید Enter سمت راست صفحه کلید و یا کلید Shift و Enter سمت چپ را همزمان استفاده کرد. با اجرای یک دستور خود دستور در یک بلاک زیر علامت In[n]:= و پاسخ آن نیز در یک بلاک زیر علامت Out[n]:= قرار می گیرد و در نهایت این دو بلاک در یک بلاک کلی قرار میگیرند. در ادامه چند نکته مهم برای کار کردن با Mathematica را توضیح می دهیم:

* اگر بخواهیم در متن برنامه در بین دستورات دستوری را اضافه کنیم علامت ماوس را بین دو بلاک مورد نظر می بریم وقتی نشانگر ماوس تغییر کرد کلید Enter را می زنیم.

* اگر بخواهیم در متن برنامه توضیحاتی اضافه کنیم که اجرا نشوند آنها را به صورت یک خطی و بین دو علامت (* و *) قرار می دهیم.

* اگر بخواهیم در متن برنامه یک ورودی یا خروجی را پاک کنیم روی بلاک مورد نظر کلیک کرده و کلید Delete را می زنیم.

* اگر در انتهای دستورات از نقطه ویرگول (;) استفاده کنیم دستور اجرا می شود ولی خروجی آن نمایش داده نمی شود.

* برای تایپ چند دستور در یک بلاک از کلید Enter سمت چپ استفاده می کنیم.

* برای تغییر Style صفحه نمایش میتوان از گزینه Stylesheet درون منوی Format استفاده کرد.

* برای تغییر اندازه نوشته ها از گزینه Magnification درون منوی Format استفاده می کنیم. مقدار پیش فرض 100% است.

* برای انصراف از ادامه اجرا در حین اجرای یک دستور میتوان از گزینه AbortEvaluation یا Quit Kernel در منوی Kernel استفاده کرد.

* متغیرهایی که در برنامه تعریف میشوند همواره شناخته شده هستند برای پاک کردن حافظه از گزینه Quit Kernel در منوی Kernel استفاده می شود.

* برای اجرای کلیه دستورهای پنجره از گزینه Evaluation Notebook در گزینه Evaluation از منوی Kernel استفاده می کنیم.

* برای اجرای قسمتی از دستور آن را با ماوس یا Shift انتخاب می کنیم و کلید Ctrl و Shift و Enter را با هم می زنیم.

* بسیاری از کاراکترها، برخی عملگرهای ریاضی و ... به صورت آماده در Mathematica وجود دارد که باعث سهولت در نوشتن می شود برای استفاده از این امکانات میتوان از گزینه های Basic Input و Complete Character و ... در گزینه Palette از منوی File استفاده کرد. برخی از این نمادها را میتوان از طریق صفحه کلید نیز وارد کرد؛ به عنوان مثال برای نوشتن حروف یونانی کافیست کلید Esc را فشار داده سپس چند حرف اول نام آن را تایپ و دوباره کلید Esc را بزنیم.شما هنگامی که روی نماد مورد نظر در پنجره مربوطه می روید اگر آن نماد روش نوشتن با صفحه کلید داشته باشد در انتهای پنجره نمایش داده خواهد شد.

* برای اضافه کردن Palette های جدید مثل جدول تناوبی ، ثابت های فیزیکی و ... بعد از نصب Mathematica به آدرس زیر رفته

\Wolfram Research\Mathematica\5.1\Documentation\English\Demos\Palettes مسیر نصب

و Palette های موردنظر را به پوشه زیر کپی کنید. با اینکار این Palette ها به گزینه Palettes منوی فایل اضافه خواهند شد:

\Wolfram Research\Mathematica\5.1\SystemFiles\FrontEnd\Palettes مسیر نصب

همچنین لازم می دانم چند نکته را که باعث کاهش خطا در هنگام نوشتن دستورات می شود را ذکر کنم:

* در Mathematica حروف کوچک و بزرگ با هم متفاوت هستند و حرف اول کلیه دستورات و توابع باید حرف بزرگ باشد مثل Sin و Plot3D و ArcTan و ... .

* در تمام دستورات عبارتهای مربوط به دستور داخل براکت « []» قرار میگیرد؛ همچنین آرگومانهای توابع نیز داخل براکت قرار می گیرد.

* از پرانتز برای تغییر اولویت عملگرها می توان استفاده کرد.

* Mathematica همه توابعش را در هنگام اجرا لود نمی کند از اینرو برای اجرای برخی دستورات لازم است ابتدا بسته ای را لود کنیم شکل کلی لود یک Package به صورت زیر است :

<< نام نوع بسته` نام زیربسته`

در دستور فوق علامت “ ` ” همان کلید سمت چپ عدد 1 صفحه کلید است نه کلید سمت چپ Enter .

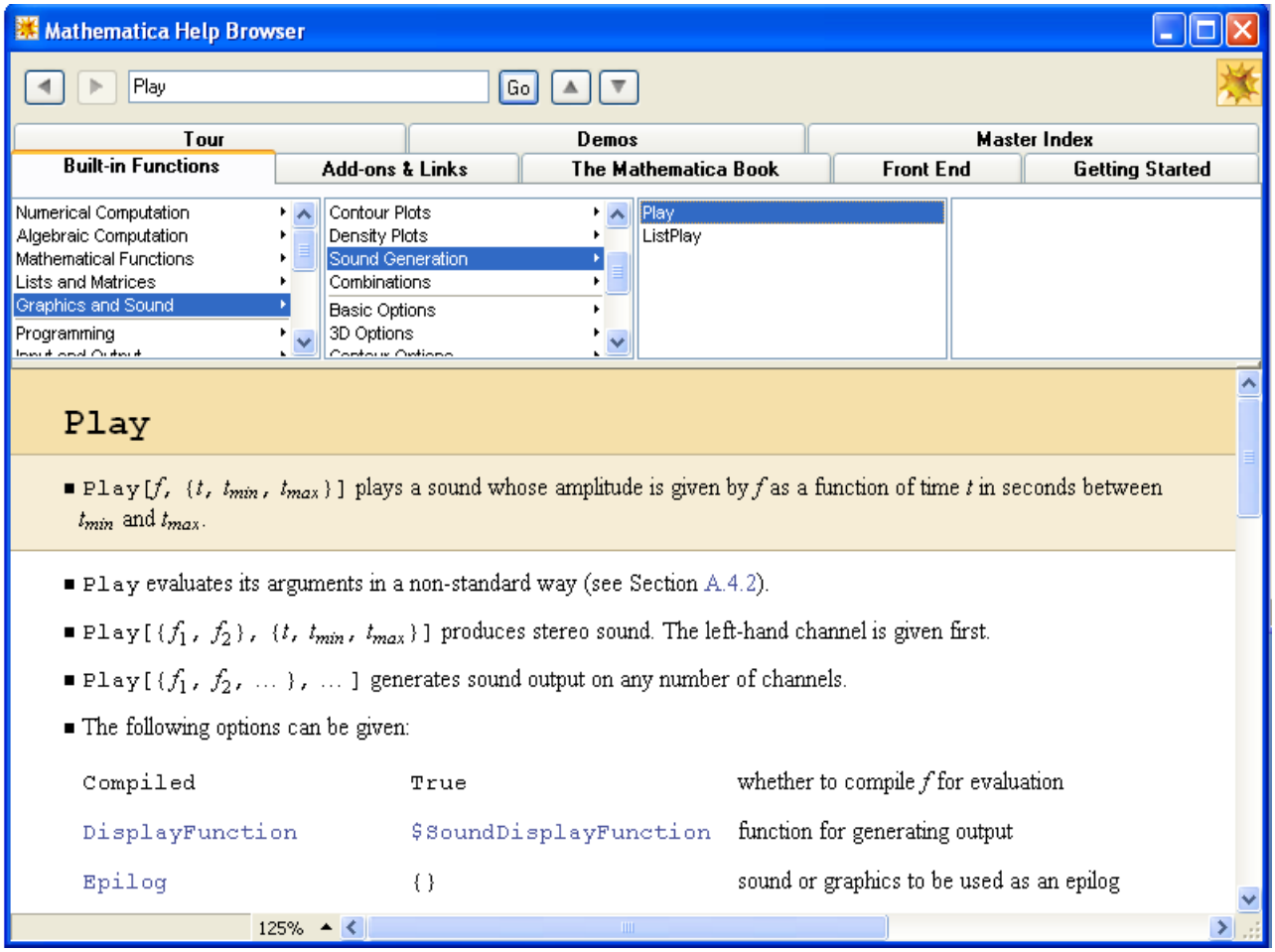
* از آنجا که متغیرها و توابعی که در این محیط تعریف شده اند تا انتهای برنامه شناخته می شوند در استفاده مجدد از آنها برای اهداف دیگر باید آنها را پاک کرد، این کار را می توان توسط دستور Clear انجام داد یا دستور Quit Kernel را اجرا کرد و یا دستور زیر را به کار برد که کلیه متغیرها را پاک می کند:

```
Clear[“Global`*”]
```

همچنین اجرای دستور []Quit و یا []Exit معادل گزینه Quit Kernel است.

استفاده از Help

- از آنجا که Mathematica دارای دستورات زیادی است در بسیاری از موارد استفاده از Help لازم میشود.برای این کار روش های متفاوتی وجود دارد. یکی از ساده ترین روش ها استفاده از منوی Help میباشد که میتوان با تایپ دستور یا حروف اولیه آن جزئیات مربوط به آن را با مثال هایی مشاهده کرد.



در برخی موارد میتوان با دستور زیر اطلاعاتی راجع به دستورها پیدا کرد:

نام دستور ?

نام دستور ??

در مورد دوم اطلاعات بیشتری نمایش داده می شود. در قسمت نام دستور میتوان از علامت * نیز استفاده کرد .

بسیاری از دستورها دارای Option های متفاوتی هستند که برای مشاهده آنها می توان دستور زیر را به کار برد:

Option[نام دستور]

روش دیگر استفاده از Help انتخاب دستور یا Option های آن و سپس زدن کلید F1 است.

بعد از تایپ چند کاراکتر اولیه یک دستور با زدن همزمان کلیدهای Ctrl+K میتوان کلیه دستورهای که با آن کاراکترها آغاز می شوند را مشاهده کرد. به عنوان مثال با نوشتن کلمه Arc و سپس زدن همزمان دو کلید Ctrl+K کلیه توابع آرک مشاهده می شوند.

مرورگر Help (Mathematica Help Browser) ، شکل فوق ، دارای Tab های مختلفی می باشد که شرح مختصر آنها به صورت زیر است :

Built-in Functions : دارای بخش های کاربردی مختلفی است که هر بخش خود شامل توابع زیادی در زمینه مربوط به خود می باشد ، از جمله Numerical Computation ، Algebric Computation ، Mathematical Functions ، Programming ، System Interface و ... می باشد.

Add-ons & Links : شامل بسته های استاندارد (که باید لود شوند) ، رابط گرافیکی کاربر GUI ، بسته های Web ، لینک ها و ... می باشد.

The Mathematica Book : کتاب آموزشی Mathematica و نحوه کار با آن.

Front End : شامل گزینه های کار با منوها ، تغییر استیل صفحات ، کلیدهای میانبر و ... می باشد.

Getting Started : نحوه شروع کار با Mathematica را شرح می دهد.

Tour : شامل یک مرور سریع ومفید جهت آشنایی اولیه با امکانات Mathematica است.

Demos : شامل گالری فرمول ها ، گرافیک ها ، صوت ها ، Notebooks و ... میباشد.

Master Index : فهرست الفبایی کلیه دستورات ، توابع و ... .

مثال :

In[1]:= **? Plot**

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin
   to xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. More...
```

In[2]:= **?? Plot**

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin
   to xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. More...

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None, AxesOrigin → Automatic,
  AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic, Compiled → True,
  DefaultColor → Automatic, DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
  Epilog → {}, FormatType :→ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
  FrameTicks → Automatic, GridLines → None, ImageSize → Automatic, MaxBend → 10., PlotDivision → 30.,
  PlotLabel → None, PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic,
  PlotStyle → Automatic, Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[3]:= **? *p**

```
System`
CellGroup     EllipticExp              Map         Top
Chop          Exp                      MatrixExp   TrigToExp
Clip          ExponentStep             Reap        UnitStep
DragAndDrop   HelpBrowserLookup        Skip        Up
Drop          MacintoshSystemPageSetup StringDrop
```

In[4]:=

? Arc*

```
System`
ArcCos  ArcCosh  ArcCot  ArcCoth  ArcCsc  ArcCsch  ArcSec  ArcSech  ArcSin  ArcSinh  ArcTan  ArcTanh
```

In[5]:= **Options[Solve]**

Out[5]= {InverseFunctions → Automatic, MakeRules → False, Method → 3,
Mode → Generic, Sort → True, VerifySolutions → Automatic, WorkingPrecision → ∞}

دستورهای مقدماتی

اعمال اصلی

برای محاسبه اعمال اصلی از نمادهای + ، – ، * ، / و ^ استفاده میکنیم. در این عملگرها اولویت به ترتیب با توان ، ضرب و تقسیم و نهایتا جمع و تفریق است. برای تغییر اولویت از پرانتز استفاده میشود. به عنوان مثال دستورهای 3^2*5+1 و 3^2*(5+1) به ترتیب مقادیر 46 و 54 را می دهند. نماد ضرب را میتوان به صورت « × » نیز وارد کرد که برای این کار * را بین دو کلید Esc می زنیم، یعنی : « Esc+*+Esc » . استفاده از جای خالی (Space Bar) بین اعداد یا متغیرها به منزله ضرب آنها تلقی می شود. برای نوشتن نماد تقسیم به صورت « ÷ » و یا خط کسری به ترتیب از ترکیبهای « Esc+/+Esc » و « Ctrl+/ » استفاده می شود. نماد توان را نیز می توان با استفاده از کلیدهای « Ctrl+^ » به صورت ملموس تری نوشت. همچنین لازم به ذکر است برخی ثابتها به صورت آماده وجود دارند که به تعدادی از آنها اشاره می شود:

– عدد π : که میتوان آن را به صورت Pi یا Esc+P+Esc وارد کرد.

– عدد نپر : که به صورت E یا Esc+E+Esc قابل نوشتن است.

– عدد موهومی : که به صورت I یا Esc+ii+Esc قابل نوشتن است.

– عدد طلایی : که به صورت "GoldenRatio" قابل نوشتن است.

– نماد بینهایت : که به صورت Infinity یا Esc+inf+Esc قابل نوشتن است.

جایگزینی و متغیرها

دو روش جایگزینی و مقدار دادن به متغیرها وجود دارد که عبارتند از:

مقدار یا عبارت = نام متغیر

مقدار یا عبارت := نام متغیر

در روش اول مقدار یا عبارت بلافاصله ارزیابی شده و سپس در متغیر قرار می گیرد ولی در روش دوم مقدار یا عبارت زمانی ارزیابی می شود که متغیر در یک محاسبه وارد شود.

متغیرها دارای انواع مختلفی هستند که برخی از آنها عبارتند از:

- Integer :اعداد صحیح .

- Rational : اعداد گویا.

- Real : اعداد حقیقی.

- Complex : اعداد مختلط.

- String : رشته ها .

که با دستور زیر می توان نوع آنها را تعیین کرد:

Head[نام متغیر]

همچنین برای پاک کردن یک متغیر می توان از دستورات زیر استفاده کرد :

Clear[نام متغیر]

Remove[نام متغیر]

همچنین از دستور زیر می توان برای پاک کردن کلیه متغیرها استفاده کرد :

Clear[“Global`*”]

برای نمایش محتوای یک متغیر از دستور زیر استفاده کرد :

نام متغیر ?

محاسبات دقیق و تقریبی

برای محاسبه عددی یک عبارت می توان از دستورات زیر استفاده کرد :

N[عبارت] یا N// عبارت

مقدار عددی عبارت را تا پنج رقم اعشار می دهد.

N[عبارت , n]

مقدار عددی عبارت را تا n رقم اعشار می دهد.

نکته : برای دستیابی به اطلاعات قبلی از کاراکتر % استفاده می شود :

% : آخرین نتیجه قبلی

% ... %% : اگر تعداد % ها n باشد n امین نتیجه قبلی را می دهد.

%n : خروجی n ام.

In[n] : ورودی n ام.

مثال :

```
In[1]:= 12 + (* these words will be ignored by the kernel *)17
Out[1]= 29

In[2]:= (12 + 3 i) (3 - 6 i^(1/2)) / (5 - 6 i)
        N[%]
        N[%, 20]
        N[%%%, 20]
Out[2]= (42/61 + 87 i/61) (3 - 6 (-1)^(1/4))
Out[3]= 5.19539 - 4.69345 i
Out[4]= 5.19539 - 4.69345 i
Out[5]= 5.1953906708257021572 - 4.6934532563670128506 i

In[6]:= a = 1;
        ? a;
        Clear[a];
        ? a

        Global`a

        a = 1

        Global`a

In[10]:= x = 2; y = 4.5; z = 2 + i; name = "Mojtaba Golshani";
         Print[Head[x], " , ", Head[y], " , ", Head[z], " , ", Head[name]]

         Integer , Real , Complex , String

In[12]:= x = y^2; x
         y = 3; x
Out[12]= 20.25
Out[13]= 20.25

In[14]:=
         x = 1; y = 2; a = 3; b = 4;
         {x, y, a, b}
         Clear["Global`*"]
         {x, y, a, b}
Out[15]= {1, 2, 3, 4}
Out[17]= {x, y, a, b}
```

برخی توابع موجود در Mathematica

برنامه Mathematica دارای توابع ریاضی و فیزیکی زیادی می باشد که ما در زیر به برخی از آنها اشاره می کنیم ، برای بدست آوردن اطلاعات بیشتر در مورد توابع از Help کمک بگیرید (در توابع زیر x در حالت کلی مختلط است) :

تابع جذر :

Sqrt[x] یا "Ctrl+2"

تابع نمایی :

Exp[x] یا "E^x"

فاکتوریل :

Factorial[x] یا "x!"

لگاریتم طبیعی :

Log[x]

لگاریتم در مبنای b :

Log[b,x]

توابع مثلثاتی ، هیپربولیک و معکوس آنها (آرگومان توابع مثلثاتی بر حسب رادیان است) :

Sin[x] , Cos[x] , Tan[x] , Cot[x] , Sec[x] , Csc[x]

ArcSin[x] , Sinh[x] , Csch[x] , ArcTanh[x] , …

نکته : در توابع مثلثاتی اگر بخواهیم آرگومان درجه باشد بعد از آن Degree (یا Esc + deg + Esc) میگذاریم.

تابع جز صحیح پایین (بزرگترین عدد صحیح نابیشتر از x) :

Floor[x]

تابع جز صحیح بالا (کوچکترین عدد صحیح بزرگتر از x) :

Celling[x]

تابع قدر مطلق :

Abs[x]

تابع علامت :

Sign[x]

تابع < x > (نزدیکترین عدد صحیح به x) :

Round[x]

تابع تولید عدد تصادفی در بازه صفر و یک :

Random[]

تابع تولید عدد تصادفی از نوع خاص در بازه a و b :

Random[نوع متغیر , {a,b}]

n امین عدد اول :

Prime[n]

عدد n را به صورت ضرب عوامل اول می نویسد :

FactorInteger[n]

اگر n اول باشد True وگرنه False برمی گرداند :

PrimeQ[n]

کوچکترین مضرب مشترک m و n :

LCM[m,n]

بزرگترین مقسوم علیه مشترک m و n :

GCD[m,n]

اگر n فرد باشد True وگرنه False برمی گرداند :

OddQ[n]

باقیمانده m بر n :

Mod[m,n]

خارج قسمت m بر n :

Quotient[m,n]

عبارت را محاسبه و زمان محاسبه را می دهد :

Timing[عبارت] یا عبارت //Timing

ترکیب r از n :

Binomial[n,r]

عبارتهای یک ، دو و ... را در خروجی چاپ می کند :

Print[عبارت 1 , عبارت 2 , ...]

خواندن یک عبارت به عنوان ورودی :

Input[]

جاری شدن یک prompt و سپس خواندن یک عبارت به عنوان ورودی :

Input[“prompt”]

خواندن یک رشته به عنوان ورودی :

InputString[]

جاری شدن یک prompt و سپس خواندن یک رشته به عنوان ورودی :

InputString[“prompt”]

تابع بسل $J_n(x)$:

BesselJ[n,x]

تابع بسل $Y_n(x)$:

BesselY[n,x]

تابع بسل تعدیل یافته $I_n(x)$:

BesselI[n,x]

تابع بسل تعدیل یافته $K_n(x)$:

BesselK[n,x]

تابع خطا :

Erf[x]

چندجمله ای لژاندر مرتبه n ($P_n(x)$) :

LegendreP[n,x]

چندجمله ای لژاندر وابسته ($P_n^m(x)$) :

LegendreP[n,m,x]

تابع گاما ($\Gamma(x)$) :

Gamma[x]

چند جمله ای هرمیت مرتبه n ($H_n(x)$) :

HermiteH[n,x]

چند جمله ای لاگر مرتبه n ($L_n(x)$) :

LaguerreL[n,x]

چند جمله ای لاگر وابسته ($L_n^m(x)$) :

LaguerreL[n,m,x]

توابع هماهنگ کروی ($Y_l^m(\theta,\phi)$) :

SphericalHarmonicY[l,m,θ,ϕ]

تابع دلتای دیراک ($\delta(x)$) :

DiracDelta[x]

تابع دلتای کرونکر (δ_{nm}) :

DiscreteDelta[n,m]

اگر عبارت درست باشد True وگرنه False می دهد :

TrueQ[عبارت]

اتصال دو یا چند رشته :

StringJoin[رشته1 , رشته 2 , ...]

رشته 1 <> رشته 2 <> ...

طول یک رشته :

StringLength[رشته]

معکوس کردن یک رشته :

StringReverse[رشته]

تبدیل عبارت به رشته :

ToString[عبارت]

قسمت حقیقی و موهومی و آرگومان یک عدد مختلط :

Re[عدد مختلط] , Im[عدد مختلط] , Arg[عدد مختلط]

همیوغ مختلط (مزدوج) یک عدد ($i \rightarrow -i$) :

Conjugate[عدد]

نوشتن فرم دستور زبان Fortran و C یک عبارت :

FortranForm[عبارت] , CForm[عبارت]

مثال :

In[1]:=
```
x = Input[" Enter first number? "];
y = Input[" Enter second number? "];
Maxnum = (x + y + Abs[x - y])/2
Minnum = (x + y - Abs[x - y])/2
```

Out[3]= 15

Out[4]= 12

In[5]:=
```
name = InputString["Enter your name?"];
Print["Hello ", name]
```

```
Hello Ezat
```

In[7]:=
```
Clear[x]
BesselJ[3/2, x] + BesselK[1/2, x]
```

Out[8]= $\frac{e^{-x}\sqrt{\frac{\pi}{2}}}{\sqrt{x}} + \frac{\sqrt{\frac{2}{\pi}}\left(-\text{Cos}[x] + \frac{\text{Sin}[x]}{x}\right)}{\sqrt{x}}$

In[9]:=
```
SphericalHarmonicY[2, 1, θ, ϕ]
```

Out[9]= $-\frac{1}{2}e^{i\phi}\sqrt{\frac{15}{2\pi}}\,\text{Cos}[\theta]\,\text{Sin}[\theta]$

In[10]:=
```
grade = Random[Real, {0, 20}]
```

Out[10]= 14.1307

In[11]:=
```
PrimeQ[252097800623]
```

Out[11]= True

In[12]:=
```
FactorInteger[99]
```
$3^2 \times 11^1$

Out[12]= {{3, 2}, {11, 1}}

Out[13]= 99

In[14]:=
```
LegendreP[n, 1]
```

Out[14]= 1

In[15]:=
```
TrueQ[1 + 1/(1 + 1/(1 + 1/2)) == 8/5]
```

Out[15]= True

محاسبات جبری و مثلثاتی

یکی از ویژگیهای مهم Mathematica توانایی آن در محاسبات Symbolic که شامل متغیرها می باشند ، است. برخی از دستورات که در ساده سازی عبارات جبری و مثلثاتی کاربرد دارند عبارتند از :

بسط یک ضرب یا توان در چندجمله ایها :

Expand[چندجمله ای]

تجزیه چندجمله ای به عوامل اول (فاکتورگیری) :

Factor[چندجمله ای]

ساده کردن عبارت کسری :

Cancel[عبارت کسری]

ترکیب چند عبارت و ساده کردن آنها :

Together[عبارت ها]

تجزیه یک کسر به کسرهای جزئی :

Apart[عبارت کسری]

ساده کردن یک کسر مثلثاتی :

Cancel[عبارت کسری مثلثاتی , Trig → True]

فاکتورگیری عبارات مثلثاتی :

TrigFactor[عبارت مثلثاتی]

ترکیب چند عبارت مثلثاتی :

Together[عبارتهای مثلثاتی , Trig → True]

بسط عبارت مثلثاتی :

TrigExpand[عبارت مثلثاتی]

عبارت مثلثاتی را برحسب عبارات خطی مثلثاتی می نویسد :

TrigReduce[عبارت مثلثاتی]

تبدیل عبارت مثلثاتی یا هیپربولیک به نمایی :

TrigToExp[عبارت مثلثاتی یا هیپربولیک]

تبدیل عبارت نمایی به مثلثاتی یا هیپربولیک :

ExpToTrig[عبارت نمایی]

ساده کردن یک عبارت :

Simplify[عبارت]

FullSimplify[عبارت]

برای محاسبه یک عبارت شامل متغیرها به اضای مقادیرخاص از متغیرها از دستور زیر استفاده می شود :

عبارت /.{ $x \to x_0 , y \to y_0 , \dots$ }

نکته : برای نوشتن " → " به صئرت پیوسته از کلیدهای " Esc + > + - +Esc " کنید.

مثال :

In[1]:=
```
Expand[(3 x - 2 Sqrt[x])^3]
Factor[%]
```

Out[1]= $-8\,x^{3/2} + 36\,x^2 - 54\,x^{5/2} + 27\,x^3$

Out[2]= $\left(-2 + 3\sqrt{x}\right)^3 x^{3/2}$

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

Out[3]= $-1 + x$

In[4]:=
```
Together[1/(x + 1) + 2/(x^2 - 1)]
```

Out[4]= $\frac{1}{-1 + x}$

In[5]:=
```
Apart[(x^2 + 5 x)/(x^4 + x^3 - x - 1)]
```

Out[5]= $\frac{1}{-1 + x} + \frac{2}{1 + x} + \frac{-1 - 3\,x}{1 + x + x^2}$

In[6]:=
```
Cancel[Sin[x]/(1 - Cos[x]^2)]
Cancel[Sin[x]/(1 - Cos[x]^2), Trig -> True]
```

Out[6]= $-\frac{\text{Sin}[x]}{-1 + \text{Cos}[x]^2}$

Out[7]= $\text{Csc}[x]$

In[8]:=
```
TrigExpand[(Sin[x] + Cos[2 x])^2]
```

Out[8]= $1 - \frac{\text{Cos}[x]^2}{2} + \frac{\text{Cos}[x]^4}{2} - \text{Sin}[x] + 3\,\text{Cos}[x]^2\,\text{Sin}[x] + \frac{\text{Sin}[x]^2}{2} - 3\,\text{Cos}[x]^2\,\text{Sin}[x]^2 - \text{Sin}[x]^3 + \frac{\text{Sin}[x]^4}{2}$

In[9]:=
```
TrigReduce[Sin[2 x]^2 + Sin[x] Cos[3 x]^3]
```

Out[9]= $\frac{1}{8}\,(4 - 4\,\text{Cos}[4\,x] - 3\,\text{Sin}[2\,x] + 3\,\text{Sin}[4\,x] - \text{Sin}[8\,x] + \text{Sin}[10\,x])$

In[10]:=
```
Simplify[Expand[(1/(x + 1) + 1/(x + 2) + 1/(x + 3))^3]]
```

Out[10]= $\frac{(11 + 12\,x + 3\,x^2)^3}{(6 + 11\,x + 6\,x^2 + x^3)^3}$

In[11]:=
```
(Tan[x]^2 + Sin[x]^2)^3 // TrigExpand // FullSimplify
```

Out[11]= $\frac{1}{8}\,(3 + \text{Cos}[2\,x])^3\,\text{Tan}[x]^6$

In[12]:=
```
TrigToExp[Sinh[x]]
```

Out[12]= $-\frac{e^{-x}}{2} + \frac{e^{x}}{2}$

In[13]:=
```
2 x + 3 y ArcTan[Cos[x^5]] /. {x -> 0, y -> 5}
% // N
```

Out[13]= $\frac{15\,\pi}{4}$

Out[14]= 11.781

سریها ، حاصلضربها ، توابع

سریها و حاصلضربها

برای محاسبه مجموع و حاصلضرب ها میتوان از نمادهای مربوط موجود در BasicInput واقع در گزینه Palettes منوی File استفاده کرد (برای جابجایی بین خانه ها از Tab استفاده کنید.) . روش دیگر برای محاسبه مجوع و حاصلضرب ها استفاده از دستورات زیر است :

محاسبه $\sum_{i=m}^{n} f(i)$ با طول گام واحد :

```
Sum[ f[i] , { i , m , n } ]
```

محاسبه $\sum_{i=m}^{n} f(i)$ با طول گام di :

```
Sum[ f[i] , { i , m , n , di } ]
```

محاسبه $\prod_{i=m}^{n} f(i)$:

```
Product[ f[i] , { i , m , n } ]
```

در دستورات فوق n می تواند بینهایت نیز باشد. برای محاسبه تقریبی میتوان ار دستور N که قبلا توضیح داده شد استفاده کرد و یا ابتدای دستورات فوق از حرف N استفاده کرد (NSum و NProduct) .

مثال :

In[1]:=
$$\sum_{i=1}^{\infty} \frac{1}{i^2}$$

```
% // N
```

Out[1]=
$$\frac{\pi^2}{6}$$

Out[2]= 1.64493

In[3]:=
```
Sum[Prime[k], {k, 1, 100000}] // Timing
```

Out[3]= {0.218 Second, 62260698721}

In[4]:=
$$\sum_{i=0}^{n} \text{Binomial}[n, i]$$

Out[4]=
$$2^n$$

In[5]:=
$$\prod_{i=2}^{10} \sum_{j=1}^{i} \frac{1}{j}$$

Out[5]=
$$\frac{7301752355616983}{3901685760000}$$

In[6]:=
$$\sum_{i=1}^{10} \text{Sin}[i\,x] \text{ // TrigFactor}$$

Out[6]=
$$2\,\text{Cos}\left[\frac{x}{2}\right]\,(1 - 2\,\text{Cos}[x] + 2\,\text{Cos}[2\,x])\,(1 + 2\,\text{Cos}[x] + 2\,\text{Cos}[2\,x])$$
$$(1 + 2\,\text{Cos}[x] + 2\,\text{Cos}[2\,x] + 2\,\text{Cos}[3\,x] + 2\,\text{Cos}[4\,x] + 2\,\text{Cos}[5\,x])\,\text{Sin}\left[\frac{x}{2}\right]$$

In[7]:=
$$\prod_{i=1}^{n} i$$

Out[7]=
$$n!$$

تعریف توابع در Mathematica

دستورات معمول برای تعریف توابع یک یا چند متغیره و یا محاسبه مقدار تابع در نقاط به صورت زیر می باشد :

تعریف تابع تک متغیره :

f [x_] = x عبارتی بر حسب

تعریف تابع دو متغیره :

f [x_ , y_] = y و x عبارتی بر حسب

پاک کردن تابع f :

Clear[f]

محاسبه f(x) در x=a :

f[a]

محاسبه f(x,y) در (a,b) :

f[a , b]

همچنین برای تعریف توابع چندضابطه ای از دستورات زیر می توان استفاده کرد :

تعریف ضابطه f(x) برای x های خاص :

f [x_ /; x شرطی روی] = x عبارتی بر حسب

f [x_] = x عبارتی بر حسب /; x شرطی روی

f[x_] = If[x شرطی روی , x عبارتی برحسب]

تعریف ضابطه f(x,y) برای x و y های خاص :

f [x_ /; x شرطی روی , y_ /; y شرطی روی] = y و x عبارتی بر حسب

مثال :

```
In[1]:= f[x_] = x^2 - BesselJ[3, x]; g[x_] = x Erf[2 x];

        f[5] + g[x] /. x → 3 // N
        Clear[f, g]

Out[2]= 27.6352

In[4]:= f[x_ /; x ≥ 0] = x^2
        f[x_ /; x < 0] = "undefined"
        {f[2], f[-4]}

Out[4]= x^2

Out[5]= undefined

Out[6]= {4, undefined}

In[7]:= h[x_] = Input["Enter a function ?"]
        h[3]

Out[7]= x^2 - Sin[x]

Out[8]= 9 - Sin[3]

In[9]:= Clear[f]
        f[0] := 1;
        f[n_] := n f[n - 1];
        f[5]

Out[12]= 120
```

محاسبه حد ، مشتق و انتگرال

برای محاسبه حد از دستورات زیر استفاده می شود :

حد f(x) در x = a :

Limit[f[x] , x → a]

حد راست f(x) در x = a :

Limit[f[x] , x → a , Direction → -1]

حد چپ f(x) در x = a :

Limit[f[x] , x → a , Direction → +1]

حد f(x , y) در (a,b) :

Limit[f[x , y] , { x → a , y → b }]

برای محاسبه مشتق از دستورات زیر استفاده می کنیم :

محاسبه f'(x) ، f''(x) و ... :

f '[x] , f ''[x] , …

محاسبه مشتق n ام :

D[f , {x,n}]

Derivative[n][f][x]

$\partial_{\{x,n\}} f[x]$

محاسبه $\frac{\partial f}{\partial x}$:

D[f , x]

محاسبه $\frac{\partial^n f}{\partial x^n}$:

D[f , {x,n}]

محاسبه $\frac{\partial^k f}{\partial x_1 \partial x_2 \partial x_3 \dots \partial x_k}$:

D[f , x_1 , x_2 , x_3 , … , x_k]

$\partial_{\{x_1,x_2,\dots,x_k\}} f[x_1, x_2, \dots, x_k]$

محاسبه $\frac{\partial^n f}{\partial {x_1}^{n_1} \partial {x_2}^{n_2} \partial {x_3}^{n_3} \dots \partial {x_k}^{n_k}}$:

D[f , {x_1 , n_1} , {x_2 , n_2} , {x_3 , n_3} , … , {x_k ,n_k}]

Derivative[n_1 , n_2 , n_3 , … , n_k][f][x_1 , x_2 , x_3 , … , x_k]

محاسبه مشتق n ام f(x) در x = a :

Derivative[n][f][a]

محاسبه $\frac{\partial^n f}{\partial {x_1}^{n_1} \partial {x_2}^{n_2} \partial {x_3}^{n_3} \dots \partial {x_k}^{n_k}}$ در $(a_1, a_2, a_3, \dots, a_k)$:

Derivative[n_1 , n_2 , n_3 , … , n_k][f][a_1 , a_2 , a_3 , … , a_k]

برای محاسبه انتگرالهای معین و نامعین میتوان از نمادهای مربوط موجود در BasicInput استفاده کرد . همچنین میتوان از دستورات زیر برای محاسبه انتگرالهای معین یگانه یا چندگانه استفاده کرد :

محاسبه انتگرال نامعین f(x) :

Integrate[f[x] , x]

محاسبه انتگرال نامعین دوگانه :

Integrate[f[x , y] , x , y]

محاسبه انتگرال $\int_a^b f(x)dx$:

Integrate[f[x] , { x , a , b }]

یکی از Option های این دستور Assunptions است که به کمک آن میتوان یک فرض در حین انتگرالگیری وارد کرد .

محاسبه انتگرال $\int_a^b \int_c^d f(x,y)dxdy$:

```
Integrate[ f[x , y] , { x , a , b } , { y , c , d }]
```

برای محاسبه انتگرالها به صورت عددی و یا محاسبه عددی انتگرال های معین غیرقابل حل از دستور N استفاده می شود .

مثال :

In[1]:=
```
Limit[(Tan[x] - x)/x^3, x → 2] // N
Limit[(1 + Sin[x])^Cot[2 x], x → 0]
```

Out[1]= -0.52313

Out[2]= $\sqrt{e}$

In[3]:=
```
Limit[Sin[x]/Abs[x], x → 0, Direction → -1]
Limit[Sin[x]/Abs[x], x → 0, Direction → 1]
```

Out[3]= 1

Out[4]= -1

In[5]:=
```
f[x_] = (x^3 - 2 x + 3)^3/((x - 1) (x + 3));
```

In[6]:=
```
f'''''[2]
```

Out[6]= $\frac{22090008}{3125}$

In[7]:=
```
∂{x,2} f[x] // Simplify
```

Out[7]=
$$\frac{1}{(-3+2x+x^2)^3}(2\,(189+189x-1881x^2+2815x^3+27x^4-2682x^5+1317x^6+756x^7-612x^8-107x^9+96x^{10}+21x^{11}))$$

In[8]:=
```
∂{x,2},{y,5} (x y^3 - 3 x Cot[y^2]/Exp[x]) /. {x → 3, y → 6} // N
```

Out[8]= 670416.

In[9]:=
```
Clear[x]
∫(Csc[x]/(1 - Sin[x]))^2 dx
```

Out[10]=
$$\frac{1}{6\,(-1+\sin[x])^2}\Big((\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])\,(2\,(\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])-3\cot[\tfrac{x}{2}]\,(\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])^3-12\log[\cos[\tfrac{x}{2}]]\,(\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])^3+12\log[\sin[\tfrac{x}{2}]]\,(\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])^3+4\sin[\tfrac{x}{2}]-28\sin[\tfrac{x}{2}]\,(-1+\sin[x])+3\,(\cos[\tfrac{x}{2}]-\sin[\tfrac{x}{2}])^3\tan[\tfrac{x}{2}]))$$

In[11]:=
$$\int_0^\infty \text{Exp}[-t^2]\,dt$$

Out[11]=
$$\frac{\sqrt{\pi}}{2}$$

In[12]:=
$$\int_0^1 \frac{\text{Sin}[x]}{x}\,dx$$
```
N[%, 40]
```

Out[12]=
```
SinIntegral[1]
```

Out[13]=
```
0.9460830703671830149413533138231796578123
```

In[14]:=
```
Clear[x]
```
$$f[x\_] := \int_0^x \frac{t\,\text{Cos}[t]}{x^2 + t^2}\,dt;\ f[5]\ //\ N$$

Out[15]=
```
-0.149478 + 0. i
```

In[16]:=
$$\int_0^{2\pi}\int_0^{\pi}\int_0^{R} \rho\, r^2\, \text{Sin}[\theta]\,dr\,d\theta\,d\phi$$

Out[16]=
$$\frac{4}{3}\pi R^3 \rho$$

In[17]:=
$$\text{Integrate}[\text{Exp}[a\,x], \{x, 0, \infty\}, \text{Assumptions} \to a < 0]$$

Out[17]=
$$-\frac{1}{a}$$

In[18]:=
$$\text{Assuming}\left[x \in \text{Reals}, \int \frac{\text{Sign}[x]}{x^{1/3}}\,dx\right]$$

Out[18]=
$$\begin{cases} -\frac{3\,x^{2/3}}{2} & x \le 0 \\ \frac{3\,x^{2/3}}{2} & \text{True} \end{cases}$$

In[19]:=
```
NIntegrate[Sin[Sin[x]], {x, 0, 1}]
```

Out[19]=
```
0.430606
```

رسم توابع در Mathematica

در این قسمت چند دستور مقدماتی برای رسم توابع یک و دو متغیره و پارامتری بیان می گردد :

رسم تابع f در بازه [a,b] :

```
Plot[ f[x] , { x , a , b }]
```

رسم توابع f_1 و f_2 و ... در بازه [a,b] :

```
Plot[{ f1[x] , f2[x] , …} , { x , a , b }]
```

رسم رویه z=f(x,y) در مستطیل a<x<b و c<y<d :

```
Plot3D[ f[x , y] , { x , a , b } , { y , c , d }]
```

رسم تابع پارامتری x=x(t) و y=y(t) در a<t<b :

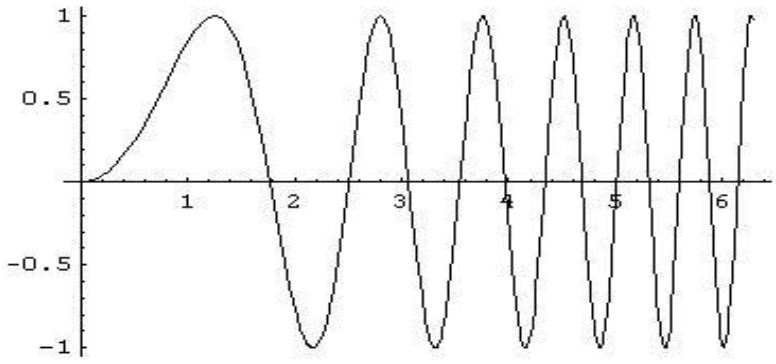
```
ParametricPlot[{x[t] , y[t] } , { t , a , b }]
```

رسم رویه پارامتری x=x(s,t) و y=y(s,t) و z=z(s,t) در بازه a<t<b و c<s<d :

```
ParametricPlot3D[{x[s,t ] , y[s,t] , z[s,t]} , {s , c , d } , { t , a , b }]
```

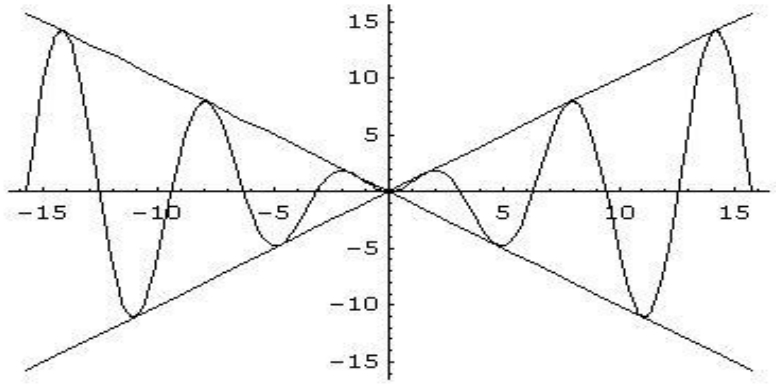
مثال :

In[1]:= `Plot[Sin[x^2], {x, 0, 2 π}]`



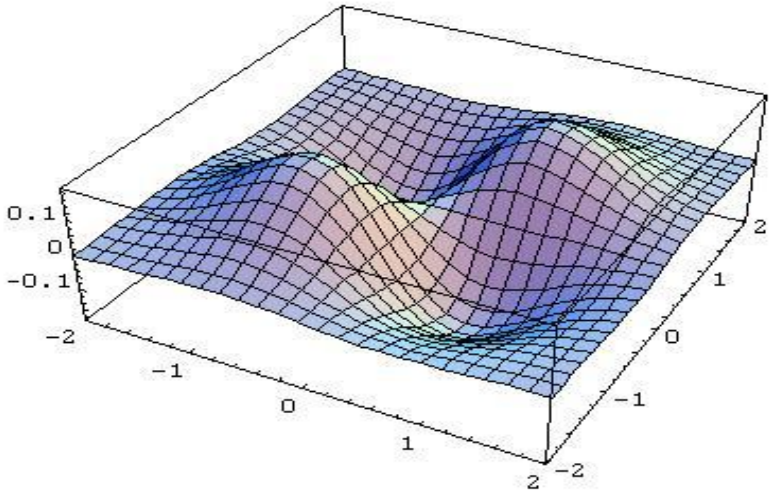
Out[1]= - Graphics -

In[2]:= `Plot[{x, -x, x Sin[x]}, {x, -5 π, 5 π}]`



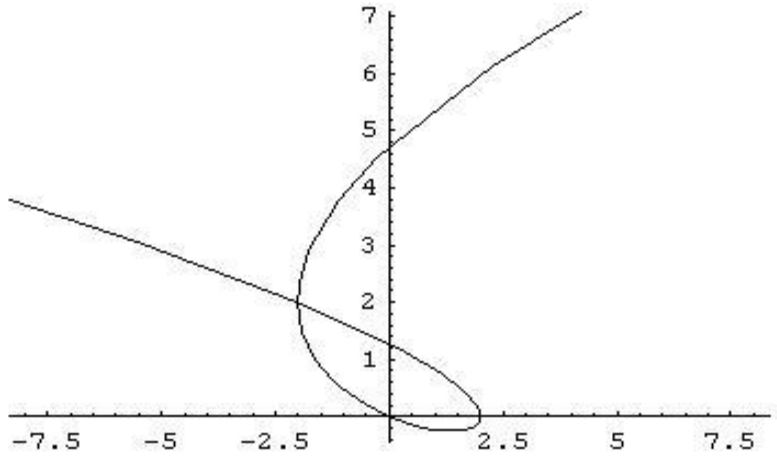
Out[2]= - Graphics -

In[3]:= `Plot3D[x y Exp[-(x^2 + y^2)], {x, -2, 2}, {y, -2, 2}]`



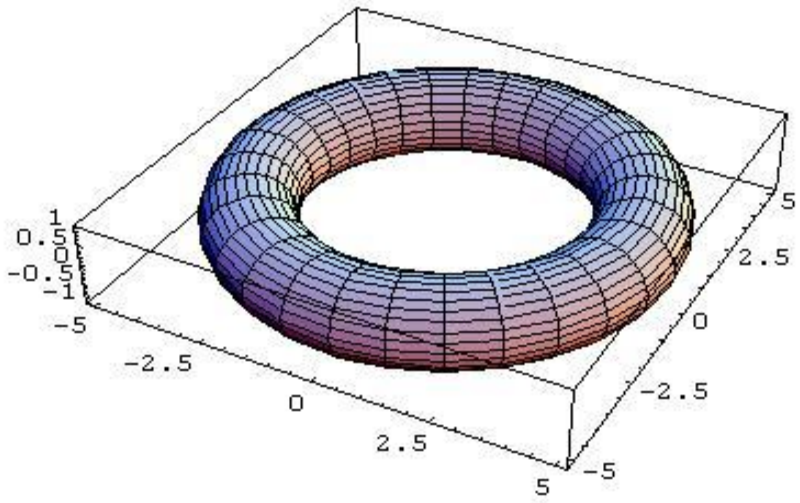
Out[3]= - SurfaceGraphics -

In[4]:=
```
ParametricPlot[{t^3 - 3 t, t^2 + t}, {t, -3, 5}]
```



Out[4]= - Graphics -

In[5]:=
```
x[s_, t_] := (4 + Sin[s]) Cos[t]
y[s_, t_] := (4 + Sin[s]) Sin[t]
z[s_, t_] := Cos[s]
ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, 0, 2π},
 {t, 0, 2π}]
```



Out[8]= - Graphics3D -

لیستها و جداول و ...

لیست یک شی عمومی است که شامل اشیاء دیگری است . برای ساختن یک لیست می توان از دستورهای زیر استفاده کرد :

یک لیست با عناصر $a_1 , a_2 , \dots , a_n$:

List[$a_1 , a_2 , \dots , a_n$]

{$a_1 , a_2 , \dots , a_n$}

روی لیست میتوان یک سری اعمال انجام داد مثلا اگر x یک لیست شامل اعداد باشد دستورات x^2 ، $\sqrt{x}$ و x! لیست های جدیدی ایجاد میکند که به ترتیب دارای عناصر x به توان 2 ، جذر عناصر x و فاکتوریل عناصر x می باشد . همچنین دو لیست با تعداد عناصر یکسان را می توان با هم جمع ، کم ، ضرب ، تقسیم و ... کرد که این کار بر روی عناصر نظیر به نظیر صورت می گیرد و جواب نهایی یک لیست جدید است . روشهای دیگر ایجاد لیست عبارتند از :

ساختن لیست {a[1] , … , a[n]} :

Table[a[i] , { i , n }]

ساختن لیست {a[m] , … , a[n]} با گام واحد :

Table[a[i] , { i , m , n }]

ساختن لیست {a[m] , a[m + d] , … , a[n]} با گام d :

Table[a[i] , { i , m , n , d }]

ساختن یک لیست دو بعدی به صورت زیر :

{{$a[m_1,n_1] , a[m_1,n_1+1] , \dots , a[m_1,n_2]$} , … , {$a[m_2,n_1] , a[m_2,n_1+1] , \dots , a[m_2,n_2]$}}

Table[a[i , j] , { i , m_1 , n_1 } , { j, m_2 , n_2 }]

ساختن آرایه دوبعدی m×n که اولین عنصر به صورت a[r,s] است :

Array[a , { m , n } , { r , s }]

ایجاد لیستی شامل کاراکترهای یک رشته :

Characters[" نام رشته "]

لیستی شامل ارقام عدد صحیح میدهد :

IntegerDigits[عدد صحیح]

در ادامه دستورات کار با لیست ها را ذکر می کنیم :

طول یک لیست (تعداد عناصر لیست) :

Length[نام لیست]

مولفه k ام یک لیست از ابتدا :

Part[نام لیست , k]

نام لیست [[k]]

مولفه k ام یک لیست از انتها :

Part[نام لیست , -k]

نام لیست [[-k]]

در لیستهای دوبعدی که هر عنصر خود یک لیست است می توان برای بدست آوردن عنصر j ام موجود در عنصر (لیست) i ام لیست اصلی از دستور زیر استفاده کرد :

نام لیست [[i]][[j]]

نام لیست [[i , j]]

دستورات زیر برای تغییر لیست ها مورد استفاده قرار می گیرند :

لیست جدیدی شامل n درایه اول لیست اصلی را بر می گرداند :

Take[نام لیست , n]

لیست جدیدی شامل n درایه آخر لیست اصلی را بر می گرداند :

Take[نام لیست , -n]

لیست جدیدی شامل درایه های m ام تا n ام لیست اصلی را بر می گرداند :

Take[نام لیست , { m , n }]

درآیه n ام لیست اصلی را حذف می کند :

Delete[نام لیست , n]

درآیه n ام لیست اصلی را از آخر حذف می کند :

Delete[نام لیست ,- n]

لیست جدیدی که در آن درآیه m ام تا n ام حذف شده اند را بر می گرداند :

Delete[نام لیست , { m , n }]

عنصر x را در مکان n ام لیست قرار می دهد :

Insert[نام لیست , x , n]

عنصر x را در مکان n ام لیست از آخر قرار می دهد :

Insert[نام لیست , x , -n]

عنصر n ام لیست را با x عوض می کند :

ReplacePart[نام لیست , x , n]

عنصر n ام لیست را از آخر با x عوض می کند :

ReplacePart[نام لیست , x , -n]

لیست را بر اساس مرتبه بزرگی مرتب می کند :

Sort[نام لیست]

ترتیب عناصر لیست را برعکس می کند :

Reverse[نام لیست]

اتصال دو لیست و تبدیل آنها به یک لیست :

Join[لیست1 , لیست2]

اجتماع دو لیست :

Union[لیست1 , لیست2]

اشتراک دو لیست :

Intersection[لیست1 , لیست2]

متمم لیست 2 نسبت به لیست 1 :

Complement[لیست1 , لیست2]

برای نمایش لیست دوبعدی به صورت آرایه مستطیلی میتوان از دستور TableForm به صورت زیر استفاده کرد :

TableForm[لیست , Options]

TableForm // لیست

برخی از Option های مهم این دستور عبارتند از :

- TableAlignments : که میتواند در سه حالت Right ، Left و Center باشد و برای مرتب کردن ستونها از راست ، چپ یا وسط به کار میرود (مقدار پیشفرض Left است) .

- TableHeadings : که میتواند در حالت None ، Automatic و یا به صورت { rowlist , columnlist } باشد .

مثال :

```
In[1]:= a = List[1, 2, 3, 4, 5, 6, 7, 8, 9]
        b = Sin[a] // N
        a^3 + b! // N

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

Out[2]= {0.841471, 0.909297, 0.14112, -0.756802, -0.958924, -0.279415, 0.656987, 0.989358, 0.412118}

Out[3]= {1.94305, 8.96498, 27.936, 67.7329, 148.807, 217.267, 343.901, 512.996, 729.887}

In[4]:= RankA = Input["Enter Rank Of Matrix"]
        A = Table[Input["Enter A(" <> ToString[i] <> "," <> ToString[j] <> ") :"], {i, 1, RankA},
          {j, 1, RankA}]

Out[4]= 2

Out[5]= {{1, 2}, {3, 4}}

In[6]:= Clear[A]
        A = Table[LaguerreL[n, x], {n, 5}];
        % // TableForm
        L1[x_] = A[[1]]
```

Out[8]//TableForm=

$$1 - x$$

$$\frac{1}{2}\,(2 - 4\,x + x^2)$$

$$\frac{1}{6}\,(6 - 18\,x + 9\,x^2 - x^3)$$

$$\frac{1}{24}\,(24 - 96\,x + 72\,x^2 - 16\,x^3 + x^4)$$

$$\frac{1}{120}\,(120 - 600\,x + 600\,x^2 - 200\,x^3 + 25\,x^4 - x^5)$$

```
Out[9]= 1 - x

In[10]:= Primset = Table[Prime[k], {k, 1, 25}]

Out[10]= {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}

In[11]:= Table[ 5 (20 i + j), {i, 0, 3}, {j, 1, 3}];
         TableForm[%, TableAlignments → Center,
          TableHeadings → { {"Row1", "Row2", "Row3", "Row4"}, {"Col1", "Col2", "Col3"}}]
```

Out[12]//TableForm=

	Col1	Col2	Col3
Row1	5	10	15
Row2	105	110	115
Row3	205	210	215
Row4	305	310	315

معادلات و نامعادلات جبری

برای حل معادلات جبری در حالت عمومی از دستور Solve یا Nsolve استفاده میشود. در صورت وجود جواب جواب ها درون یک لیست قرار می گیرند :

$\{ \{ x \rightarrow x_1 \} , \{ x \rightarrow x_2 \} , \dots \}$

برای نوشتن یک معادله باید توجه داشت که از علامت تساوی منطقی (==) استفاده شود . به عنوان مثال معادله $x^2+3x=0$ در Mathematica به صورت x^2+3 x==0 نوشته میشود. شکل دستور Solve به صورت زیر است :

Solve[لیست معادلات جبری , لیست مجهولات]

در لیست های فوق اگر یک معادله و یک مجهول داشته باشیم نیازی به استفاده از آکولاد ({}) نمیباشد. به عنوان مثال خروجی دستورات :

Solve[7x + 3==0 , x]

Solve[{ x^2+ y==5 , x + y ==3 } , { x , y }]

به ترتیب به صورت زیر است :

$\{\{ x \rightarrow -\frac{3}{7} \}\}$

{{y → 1, x → 2}, {y → 4, x → -1}}

اگر بخواهیم جواب معادله را در یک عبارت جایگزین کنیم و مقدار عبارتی را به ازای آنها حساب کنیم از روش زیر استفاده می کنیم :

Solution = Solve[لیست معادلات جبری , لیست مجهولات]

عبارت /.Solution

در دستور فوق نام Solution اختیاری است و می توان از هر نام دیگری استفاده کرد. گاهی اوقات معادله با دستور Solve قابل حل نیست در این حالت درصورتی که معادله دارای پارامتر غیرعددی نباشد می توان از دستور Nsolve آن را به صورت عددی حل کرد. دستور زیر جواب را تا n رقم می دهد :

NSolve[لیست معادلات جبری , لیست مجهولات , n]

برای حل معادلات غیر خطی از دستورات زیر استفاده می کنیم :

حل معادله با روش نیوتن و نقطه شروع x_0 :

FindRoot[معادله جبری , { x , x_0 }]

حل معادله با روش سکانت و نقاط شروع x_0 و x_1 :

FindRoot[معادله جبری , { x , x_0 , x_1 }]

حل معادله در بازه [a,b] و با نقطه شروع x_0 :

FindRoot[معادله جبری , { x , x_0 , a , b }]

حل دستگاه معادلات با نقطه شروع (x_0 , y_0 , ...) :

FindRoot[لیست معادلات جبری , { x , x_0 } , { y , y_0 } , ...]

نکته : برای پیداکردن نقطه شروع می توان دو طرف معاذله را با هم در یک دستگاه مختصات با دستور Plot رسم کرد و از روی نمودار نقطه تقاطع حدس زده شده را نقطه شروع قرار دهیم.

نکته : اگر در دستور FindRoot ریشه به صورت زبان ماشین بیان شدند از دستور N استفاده کنید .

یکی از Option های مهم این دستور WorkingPrecision میباشد که مقدار دقت لازم در محاسبات را وارد می کند و مقدار پیشفرض 16 است .

یکی دیگر از دستورات که در هنگام نیاز به دقت بالا به کار می رود دستور زیر است که معادله را با مقادیر اولیه a و b حل می کند ، توجه کنید برای استفاده از این دستور باید اول Package مربوط به InterpolateRoot را توسط دستور زیر لود کرد :

<<NumericalMath`InterpolateRoot`

اما دستور :

InterpolateRoot[معادله جبری , { x , a , b }]

این دستور نیز دارای Option ، WorkingPrecision می باشد. به عنوان مثال دستور زیر ریشه تابع بسل مرتبه صفر که نزدیک 2 و 3 است را می دهد :

InterpolateRoot[BesselJ[0, x], {x, 2, 3}, WorkingPrecision → 1000]

توجه کنید در دستور فوق یا FindRoot اگر فقط یک طرف معادله را بنویسیم طرف دوم صفر در نظر گرفته می شود.

برای حل نامعادلات از دستور Reduce استفاده میشود ، این دستور علاوه بر توانایی حل نامعادلات توانایی حل سیستمی مرکب از معادلات به همراه نامعادلات را نیز دارد و به صورت زیر به کار می رود :

Reduce[معادلات و نامعادلات لیست , مجهولات لیست , مجهولات نوع]

در قسمت نوع مجهولات میتوان Reals و یا Complexes را به کار برد.

مثال :

In[1]:=
```
eq := a x^2 + b x + c == 0
Solve[eq, x]
```

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

In[3]:=
```
sol = Solve[{x + y == 5, x - y == 1}, {x, y}]
Sqrt[x^2 + Tan[y]^2] /. sol // N
```

Out[3]= {{x → 3, y → 2}}

Out[4]= {3.71139}

In[5]:=
```
Solve[x^5 - 3 x^3 + 1 == 0, x]
```

Out[5]=
```
{{x → Root[1 - 3 #1^3 + #1^5 &, 1]},
 {x → Root[1 - 3 #1^3 + #1^5 &, 2]}, {x → Root[1 - 3 #1^3 + #1^5 &, 3]},
 {x → Root[1 - 3 #1^3 + #1^5 &, 4]}, {x → Root[1 - 3 #1^3 + #1^5 &, 5]}}
```

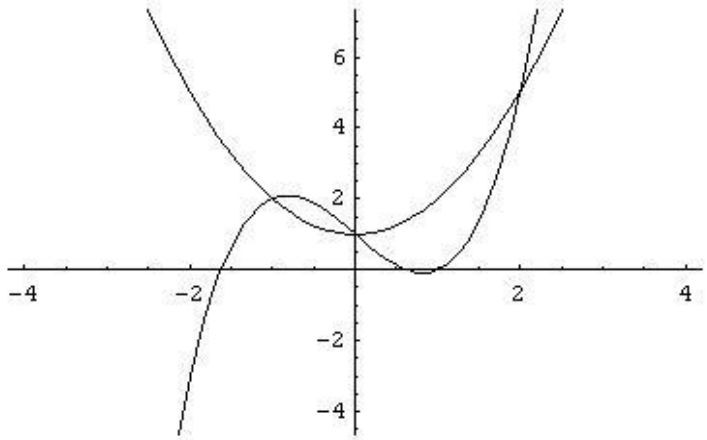
In[6]:=
```
N[%]
```

Out[6]=
```
{{x → -1.78231}, {x → 0.741814}, {x → 1.66878},
 {x → -0.314141 - 0.595441 i}, {x → -0.314141 + 0.595441 i}}
```

In[7]:=
```
NSolve[x + Sqrt[x] == 3, x, 20]
```

Out[7]= {{x → 1.69722436226800535 34}}

In[8]:=
```
f[x_] := x^3 - 2 x + 1
g[x_] := x^2 + 1
Plot[{f[x], g[x]}, {x, -4, 4}]
```



Out[10]= - Graphics -

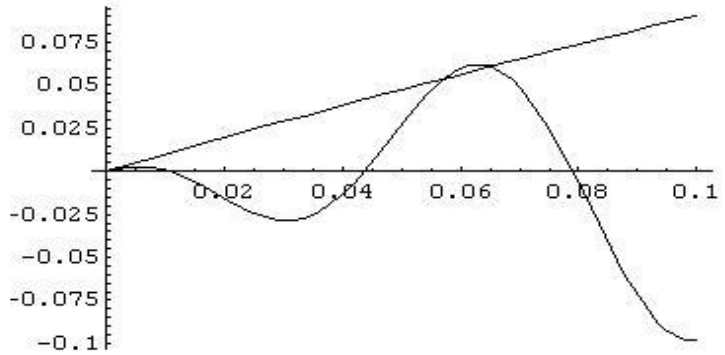
In[11]:= `solutions = Solve[f[x] == g[x], x]`

Out[11]= `{{x → -1}, {x → 0}, {x → 2}}`

In[12]:= $\sum_{k=1}^{3}$ `solutions[[k, 1, 2]]`

Out[12]= 1

In[13]:= $\text{Plot}\left[\left\{x \cos\left[\frac{100}{x+1}\right], \frac{x}{x+1}\right\}, \{x, 0, 0.1\}\right]$



Out[13]= - Graphics -

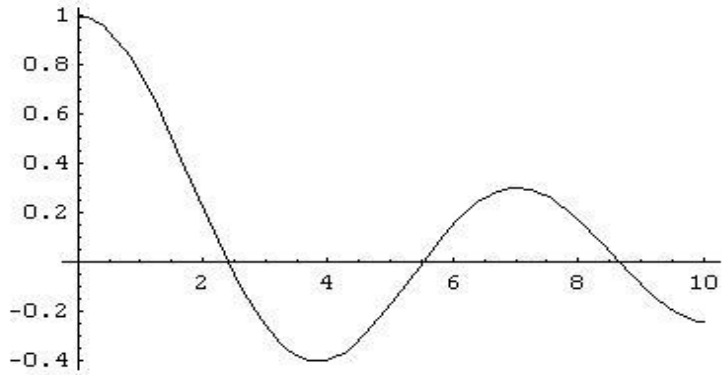
In[14]:= $\text{FindRoot}\left[x \cos\left[\frac{100}{x+1}\right] == \frac{x}{x+1}, \{x, 0.05\}\right]$

$\text{FindRoot}\left[x \cos\left[\frac{100}{x+1}\right] == \frac{x}{x+1}, \{x, 0.062\}\right]$

Out[14]= `{x → 0.057322}`

Out[15]= `{x → 0.0650013}`

In[16]:= `Plot[BesselJ[0, x], {x, 0, 10}]`



Out[16]= - Graphics -

In[17]:= `FindRoot[BesselJ[0, x], {x, 2}]`

Out[17]= `{x → 2.40483}`

In[18]:=
```
<< NumericalMath`InterpolateRoot`
```

In[19]:=
```
InterpolateRoot[BesselJ[0, x], {x, 2, 3}, WorkingPrecision → 1000]
```

Out[19]=
```
{x →
  2.404825557695772768621631879326454643124244909145967135706999090596765838677194029204436343760145254786892450444769865326938788049028412365949012688455332524230714323602601146641559413251838173780254759398849431603273379257463532524324426550939340991722884724461713970218789692538913562214263839257333735392628465340592792325908503379822010496628817558607258757526283230537733145142413070073300535813176103253323691602967384964204806909490632983052690121074404613711626025254902922183306480728314710184168405079944091145645317087032027180994560781999857373204665260253212114069680344305090177684667709789182659746369396336049404972833072134780030536375014857409599698511627793088466802413134266097897978658232226582208245474876447000588006724603541046480680741047817557893113894649129377001527791267358717445905751266065280683336052923706656070446728149795008510472509456111124510158927250699438334859718055893709330649382689692889902482504979886297467549647406753655581678360152152415148364700936520426970896141641 42}
```

In[20]:=
```
Reduce[{2 x + y ≤ 5 , 3 x - y > -2}, {x, y}]
```

Out[20]=
$$\left(x \leq \frac{3}{5} \;\&\&\; y < 2 + 3\,x\right) \;||\; \left(x > \frac{3}{5} \;\&\&\; y \leq 5 - 2\,x\right)$$

In[21]:=
```
Reduce[{2 x + y ≤ 5 , 3 x - y == -2}, {x, y}]
```

Out[21]=
$$x \leq \frac{3}{5} \;\&\&\; y == 2 + 3\,x$$

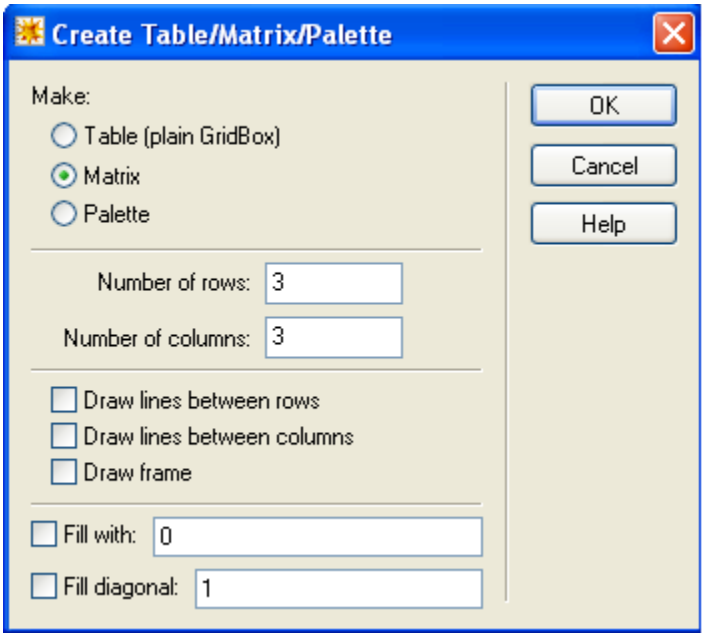
بردارها و ماتریسها

بردارها و ماتریسها درواقع همان لیست ها هستند، بردار لیست یک بعدی و ماتریس لیستی از بردارهاست. برای نوشتن یک بردار یا ماتریس به صورت مولفه مولفه از روش زیر استفاده می شود :

$$V = \{ x_1 , x_2 , \dots , x_n \}$$

$$A = \{ \{ x_{11} , x_{12} , \dots , x_{1n} \} , \dots , \{ x_{m1} , x_{m2} , \dots , x_{mn} \} \}$$

برای ورود ساده تر یک بردار و یا ماتریس میتوان از پنجره Create Table/Matrix/Palette واقع در منوی Input (با کلید میانبر Shift + Ctrl + C) استفاده کرد :



در این پنجره با انتخاب دکمه رادیویی Matrix و نوشتن تعداد سطرها و ستونها و زدن دکمه OK میتوان ماتریس را وارد کرد.

بعد از ورود ماتریس برای جابجایی بین مکان درآیه ها از کلید Tab و یا ماوس استفاده می شود .

اگر درآیه های بردار یا ماتریس قانونمند باشند می توان از دستورهای ساخت لیست قانونمند مثل Table و Array استفاده کرد.

به عنوان مثال دستورات زیر به ترتیب یک بردار n بعدی و یک ماتریس m×n ایجاد می کند :

Table[a[i] , { i , n }]

Table[a[i , j] , { i , m } , { j , n }]

برای نمایش یک ماتریس به صورت آرایه مستطیلی از دستور :

MatrixForm[نام ماتریس]

MatrixForm // نام ماتریس

استفاده می شود . دستورات مربوط به محاسبات ماتریسی و برداری به صورت زیر است (در مواردی که خروجی ماتریس است همواره دستور را در بین دو پرانتز و در انتها دستور MatrixForm را به کار برید چون ممکن است کار نکند!!!) :

اندازه (نرم) بردار V :

Norm[V]

جمع دو بردار V_1 و V_2 هم بعد :

$V_1 + V_2$

تفریق V_2 از V_1 هم بعد :

$V_1 - V_2$

ضرب اسکالر λ در بردار V :

λV

ضرب داخلی دو بردار هم بعد V_1 و V_2 :

$V_1 . V_2$

نکته : اگر بجای نقطه از علامت ضرب اعداد (جای خالی ، * و یا ×)استفاده شود بردارها مولفه مولفه در هم ضرب می شوند و جواب بجای عدد بردار است.

ضرب خارجی دو بردار سه بعدی V_1 و V_2 :

Cross[V_1 , V_2]

نکته :ضرب خارجی را میتوان توسط دستور $V_1 \times V_2$ نیزانجام داد که برای وارد کردن نماد "×" ازکلیدهای Esc+cross+Esc استفاده می شود.

مولفه i ام بردار V :

V[[i]]

جمع و تفریق دو ماتریس هم مرتبه :

($m_1 \pm m_2$) // MatrixForm

ضرب دو ماتریس در صورت امکان :

($m_1 . m_2$) // MatrixForm

نکته :اگر بجای نقطه از علامت ضرب اعداد (جای خالی ، * و یا ×)استفاده شود ماتریسها مولفه مولفه در هم ضرب می شوند.

محاسبه ترانهاده ماتریس :

(Transpose[ماتریس]) // MatrixForm

تعیین مرتبه (تعداد سطرها و ستونهای) ماتریس :

Dimensions[ماتریس]

محاسبه ماتریس همیوغ مختلط (مزدوج) ماتریس ($i \rightarrow -i$) :

(Conjugate[ماتریس]) // MatrixForm

تعیین دترمینان ماتریس مربعی :

Det[ماتریس]

تعیین رد ماتریس مربعی (جمع عناصر قطر اصلی) :

Tr[ماتریس]

ماتریس معکوس ماتریس مربعی در صورت وجود :

(Inverse[ماتریس]) // MatrixForm

توان n ام ماتریس مربعی :

(MatrixPower[ماتریس , n]) // MatrixForm

ماتریس e^A :

(MatrixExp[ماتریس]) // MatrixForm

ماتریس همانی مرتبه n :

(IdentityMatrix[n]) // MatrixForm

ماتریس قطری مرتبه n با عناصر قطری a_1 ، a_2 و ... و a_n :

(DiagonalMatrix[{ a_1 , a_2 , … , a_n }]) // MatrixForm

تعیین درآیه سطر i ام و ستون j ام ماتریس :

نام متریس [[i , j]]

خل دستگاه خطی AX=B :

LinearSolve[A , B]

حل دستگاه همگن AX=0 :

NullSpace[A]

چندجمله ای مشخصه ماتریس با متغیر x :

CharacteristicPolynomial[ماتریس , x]

ویژه مقادیر ماتریس :

Eigenvalues[ماتریس]

ویژه مقادیر ماتریس به صورت عددی تا n رقم اعشار :

Eigenvalues[N[ماتریس , n]]

ویژه بردارهای ماتریس :

Eigenvectors[ماتریس , x]

مثال :

In[1]:=
```
{m1 = {{4, 3 - I, 8}, {3 + I, 0, -2}, {8, -2, 3}}} // MatrixForm

{m2 = {{7, 4 - I, 8 I}, {3 - I, 0, 5}, {8, 1, 3}}} // MatrixForm

(m1 + m2.m1) // MatrixForm

{Det[m1], Det[m2]}

Inverse[m1].m1 // MatrixForm

{λ1, λ2, λ3} = Eigenvalues[m1] // N

Eigenvectors[m1] // N // MatrixForm
```

Out[1]//MatrixForm=
$$\begin{pmatrix} 4 & 3-i & 8 \\ 3+i & 0 & -2 \\ 8 & -2 & 3 \end{pmatrix}$$

Out[2]//MatrixForm=
$$\begin{pmatrix} 7 & 4-i & 8i \\ 3-i & 0 & 5 \\ 8 & 1 & 3 \end{pmatrix}$$

Out[3]//MatrixForm=
$$\begin{pmatrix} 45+65i & 24-24i & 56+26i \\ 55-3i & -2-6i & 37-8i \\ 67+i & 16-8i & 74 \end{pmatrix}$$

Out[4]= {-142, 100 + 5 i}

Out[5]//MatrixForm=
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Out[6]= {11.6263, -6.50412, 1.87784}

Out[7]//MatrixForm=
$$\begin{pmatrix} 1.10608+0.0254242i & 0.111199+0.101697i & 1. \\ -0.995078+0.0342936i & 0.771746+0.137174i & 1. \\ -0.64517-0.14301i & -2.0196-0.572038i & 1. \end{pmatrix}$$

In[8]:=
```
If[TrueQ[Conjugate[Transpose[m1]] == m1] , Print["m1 is Hermition Matrix "],
 Print["m1 is not Hermition Matrix "]]
If[TrueQ[Conjugate[Transpose[m2]] == m2] , Print["m2 is Hermition Matrix "],
 Print["m2 is not Hermition Matrix "]]
```

```
m1 is Hermition Matrix
m2 is not Hermition Matrix
```

In[10]:=
```
v1 = {1, i, 3 - i}; v2 = {√2 , 3 i, 7};
Norm[v1]
(Cross[v1, v2] + 2 v2).(3 v1 + v2) // N
∑_{j=1}^{3} v1[[j]]
```

Out[11]= $2\sqrt{3}$

Out[12]= $200.485 - 42.\,i$

Out[13]= 4

In[14]:=
```
A = {{3, 4, 5}, {-2, 8, 2}, {4, -3, 5}}; B = {{5}, {-6}, {7}};
LinearSolve[A, B] && Clear[A, B]
```

Out[15]= $\left\{\left\{\frac{27}{10}\right\}, \left\{\frac{1}{10}\right\}, \left\{-\frac{7}{10}\right\}\right\}$ && Null

گرافیک دوبعدی ، انیمیشن و گرافیک سه بعدی

گرافیک دو بعدی

یکی از ویژگیهای مهم Mathematica توانایی بالا در کشیدن اشکال دوبعدی ، سه بعدی و ایجاد اشکال متحرک (انیمیشن) است. دستور Plot دستوری پایه برای کشیدن توابع یک متغیره در یک بازه می باشد :

رسم تابع f در بازه [a,b] :

```
Plot[ f[x] , { x , a , b } ]
```

رسم توابع f و g و ... در بازه [a,b] :

```
Plot[ {f[x] , g[x] , … } , { x , a , b } ]
```

برای نمایش دو یا چند نمودار با هم (بخصوص در مواردی که دو تابع دامنه رسم متفاوتی داشته باشند) از دستور Show استفاده می شود. اگر g_1 ، g_2 و ... گرافیک باشند (Graphics) برای نمایش همزمان آنها می نویسیم :

```
Show[ { g1 , g2 , … } ]
```

در حالت کلی آرگومان Show میتواند یک آرایه از عناصر گرافیکی باشد . اگر این آرایه با دستور GraphicsArray ایجاد شود عناصر گرافیکی به صورت مجزا ولی در آرایه مستطیلی (وابسته به نوع آرایه تعریف شده توسط این دستور) به نمایش در می آیند :

آرایه یک بعدی از گراف ها :

```
g=GraphicsArray[ { g1 , g2 , … } ]
```

آرایه دو بعدی از گراف ها :

```
g=GraphicsArray[ { { g11 , g12 , … } , { g21 , g22 , … } , … } ]
```

دستور Plot دارای Option های زیادی است که با دستور Options[Plot] قابل مشاهده است ، دستور Plot با Option به صورت زیر به کار می رود :

```
Plot[ {f[x] , g[x] , … } , { x , a , b } , Option → Value ]
```

درواقع Mathematica میتواند امکاناتی را جهت کنترل خروجی گرافیکی فراهم آورد ، در زیر به برخی از این Option ها اشاره می کنیم :

محورهای مختصات را با "x" و "y" نامگذاری می کند :

AxesLabel → { "x" , "y" }

محورها را نشان نمی دهد :

Axes → None

حدود محورهای افقی و عمودی را به ترتیب [a,b] و [c,d] قرار می دهد :

PlotRange → {{ a,b} , { c,d }}

نمایش نمودار در چهارچوب :

Frame → True

تعیین برچسب "x" و "y" برای چهارچوب :

FrameLabel → { "x" , "y" }

رسم نمودار با ضخامت n (عددی بین صفر و یک که ضخامت را نسبت به پهنای گراف می سنجد) :

PlotStyle → Thickness[n]

رسم نمودار با رنگی که از ترکیب سه رنگ اصلی قرمز، سبز و آبی به ترتیب با وزن r ، g و b (اعداد حقیقی بین صفر و یک) ایجاد می شود :

PlotStyle → RGBColor[r , g , b]

رسم نمودار با رنگی با طیف h ، اشباع s و درخشندگی b (اعداد حقیقی بین صفر و یک) :

PlotStyle → Hue[h , s , b]

نکته : نوشتن دستور فوق با Hue[h] معادل Hue[h,1,1] است.

نسبت مقیاس عمودی به افقی را r میگذارد (مقدار پیش فرض یک به روی عدد طلایی است) :

AspectRatio → r

رسم نمودار با خط چین به طول n (عددی بین صفر و یک نسبت به پهنای گراف) :

PlotStyle → Dashing[{n}]

برای مشخص کردن نمودار با عنوان text از Option زیر استفاده می شود :

PlotLegend → { “text” }

ولی توجه شود قبل از استفاده Package زیر را لود کنید :

<<Graphics`Legend`

توجه شود اگر دستور Plot برای چند تابع به کار می رود برای کل آنها فقط یک عنوان Option به کار می رود ولی مقادیر وابسته درون یک لیست قرار می گیرد. به عنوان مثال دستور زیر نمودارهای سینوس و کسینوس را در بازه صفر تا پی و با رنگهای قرمز و سبز رسم می کند :

Plot[{Sin[x],Cos[x]} , { x,0,π } , PlotStyle → { Hue[0] , Hue[0.4] }]

اما برخی دیگر از دستورات رسم در Mathematica که برای رسم اشکال دوبعدی خاص بکار می روند به صورت زیر است :

رسم دایرهای رنگی به مرکز (x,y) و شعاع r با رنگ زمینه دلخواه :

Show[Graphics[Circle[{x,y} ,r] , DefaultColor → Hue[n] , BackColor → Hue[m]]]

رسم دیسکی رنگی به مرکز (x,y) و شعاع r با رنگ زمینه دلخواه :

Show[Graphics[Disk[{x,y} ,r] , DefaultColor → Hue[n] , BackColor → Hue[m]]]

رسم نقطه (x,y) رنگی با رنگ زمینه دلخواه :

Show[Graphics[Point[{x,y}] , DefaultColor → Hue[n] , BackColor → Hue[m]]]

رسم خط رنگی شکسته ای که نقاط را به هم وصل می کند با رنگ زمینه :

Show[Graphics[Line[{{ x_1, y_1},{ x_2 , y_2 }, … }] , DefaultColor → Hue[n] , BackColor → Hue[m]]]

رسم مستطیل با رنگ دلخواه :

Show[Graphics[{ Hue[n] ,Rectangle[{{ x_1, y_1},{ x_2 , y_2 } }] }]]

نوشتن رشته رنگی text در مکان با مرکز (x,y) با رنگ زمینه دلخواه :

Show[Graphics[Text[“text” ,{ x, y}] , DefaultColor → Hue[n] , BackColor → Hue[m]]]

دستور Graphics دارای Option های زیاد دیگری است که از جمله می توان به Axes → True اشاره کرد که باعث نمایش محورها می شود.

برای رسم توابع پارامتری از دستورات زیر می توان استفاده کرد :

رسم منحنی پارامتری x=x(t) و y=y(t) در (a,b) :

ParametricPlot[{x[t],y[t]} ,{ t,a,b}]

رسم منحنیهای پارامتری $x_1(t)$ و $y_1(t)$ ، $x_2(t)$ و $y_2(t)$ و ... در بازه (a,b) :

ParametricPlot[{{x_1[t],y_1[t]} , {x_2[t],y_2[t]} , … } ,{ t,a,b}]

برای رسم توابع ضمنی باید ابتدا بسته زیر را باز کرد :

<<Graphics`ImplicitPlot`

و سپس یکی از دستورهای زیر را به کار برد :

رسم منحنی f(x,y)=0 در بازه (a,b) :

ImplicitPlot[f[x,y] == 0 , { x , a , b }]

رسم منحنی f(x,y)=0 در بازه (a,b) با صرفنظر از نقاط x1 ، x2 و ...

ImplicitPlot[f[x,y] == 0 , { x , a , x_1, x_2 , … , b }]

یرای رسم منحنی در مختصات قطبی دوبعدی نیز ابتدا باید بسته زیر را باز کرد :

<<Graphics`Graphics`

و سپس یکی از دستورات زیر را به کار برد :

رسم منحنی $r=f(\theta)$ در بازه $a< \theta < b$:

PolarPlot[f[θ] , { θ , a , b }]

رسم منحنیهای $r=f_1(\theta)$ ، $r=f_2(\theta)$ و ... در بازه $a< \theta < b$:

PolarPlot[{ f_1[θ] , f_2[θ] , … } , { θ , a , b }]

در Mathematica دستوراتی برای رسم لیستی از نقاط نیز وجود دارد .

اگر { … , x3 , x2, x1}=x لیستی یک بعدی باشد دستور زیر مجموعه نقاط { … , (x2,f(x2)) , (x1,f(x1)) } را رسم می کند :

ListPlot[f(x) , Options]

برای رسم مجموعه نقاط { … , {x_2,y_2} , {x_1,y_1}} از دستور زیر استفاده می شود :

ListPlot[{ {x_1,y_1} , {x_2,y_2} , … }]

برخی از Option های دستور ListPlot شبیه Option های دستور Plot است. از مهمترین Option های این دستور می-توان به DefaultColor → RGBColor[r,g,b] و PlotStyle → PointSize[n] (n عدد حقیقی بین صفر و یک) اشاره کرد.

برای رسم مجموعه نقاط { … , {x_2,y_2} , {x_1,y_1}} از دستور زیر نیز می توان استفاده می شود (خوبی این دستور در رسم چند گروه از اینطور مجموعه ها با هم و ویژگیهای (مثل رنگ و ...) متفاوت است) .:

g = Graphics[{RGBColor[r,g,b] , Point/@ لیست نقاط } , Options]

لیست نقاط می تواند به هر صورت مجاز دیگری مثلا [{ a[i] , b[i] } , {i , m , n }]Table نیز باشد.

از Option های مهم در اینجا میتوان به Axes → True و "text" PlotLabel → اشاره کرد.

انیمیشن و متحرکسازی

عمل متحرک سازی یک بحث جالب است که هم درمورد اشکال دوبعدی و هم درمورد اشکال سه بعدی قابل استفاده است. ایده اصلی در این بحث ایجاد دنباله ای از Frame هاست ، با نمایش پشت سرهم این Frame ها تصور وجود شکل متحرک ایجاد می شود . بعد از ایجاد دنباله Frame ها در یک بلاک کافیست روی یکی از آنها دوبار کلیک کنیم (یا همه را انتخاب و کلیدهای Ctrl + y را بزنیم) تا انیمیشن ایجاد شود. هرچه تعداد Frame ها بیشتر و تفاوت Frame های متوالی کمتر باشد انیمیشن زیباتر خواهد بود.

برای ایجاد Frame ها روشهای متعددی وجود دارد ، مثلا میتوان تعدادی عکس را با دستور Import (این دستور بعدا توضیح داده میشود) وارد کرد و از آنها شی متحرک ایجاد کرد .

یکی از ساده ترین روش ها استفاده از دستورات تکرار و گرافیکی با هم است مثلا دستورات زیر :

```
Do [ Plot [ f[x,t] , { x,a ,b }] , {t , t1 ,t2,dt} ]

Do [ Plot3D [ f[x,y,t] , { x,a ,b } , { y, c ,d }] , {t , t1 ,t2,dt} ]
```

در دستورات فوق به اندازه جزء صحیح t2-t1 تقسیم بر dt عدد Frame ایجاد خواهد شد که با پشت سرهم نمایش دادن آنها انیمیشن ایجاد خواهد شد.

برای ذخیره این انیمیشن به صورت فایل gif که در خارج از Mathematica نیز قابل مشاهده باشد (با برنامه Internet Explorer میتوان فایلهای gif ایجاد شده در Mathematica را باز کرد.) می توان از روش زیر استفاده نمود :

- ابتدا کلیه Frame ها را در آرایهای ذخیره می کنیم (هر عضو آرایه یک Frame) .

- مسیر دلخواه خود برای ذخیره فایل را مشخص می کنیم که این کار توسط دستور زیر صورت می گیرد :

```
SetDirectory[" مسیر مورد نظر "]
```

اگر این دستور را به کار نبریم فایل در مسیر پیش فرض که به صورت زیر است ذخیره می شود :

مسیر نصب \ Wolfram Research \ Mathematica \ 5.1

نکته : برای انکه بدانیم مسیر کنونی Mathematica چیست از دستور Directory[] استفاده می کنیم.

- دستور زیر را برای ایجاد فایل gif با نام و اندازه دلخواه به کار می بریم :

Export[" نام فایل . gif " , نام آرایه , ImageSize → { عرض تصویر , طول تصویر }]

به عنوان مثال دستورات زیر را مشاهده کنید :

```
i = 0 ;
Do [ i = i +1 ; g[ i ]=Plot[ Sin[ x – 2 t ] , { x , 0 , 10 Pi }] , { t , 0 , 3 , 0.1 } ] ;
n = i ;
gg = Table[ g[ i ] , { i , 1 , n } ] ;
SetDirectory [ " C : \ Documents and Settings \ All Users \ Application Data \ Mathematica " ]
Export[ "wave.gif " ,  gg , ImageSize → { 100 , 350 } ]
```

گرافیک سه بعدی

- یکی از تواناییهای دیگر Mathematica رسم نمودارهای سه بعدی است ، در ادامه چند دستور مهم در این زمینه را ذکر می کنیم. برای اطلاعات بیشتر به Help مراجعه کنید .

یرای رسم رویه $z = f(x,y)$ در ناحیه مستطیلی $a < x < b$ و $c < y < d$ از دستور زیر استفاده می شود :

```
Plot3D[ f[x,y] , { x , a , b } , { y , c , d } ]
```

این دستور نیز مانند دستور Plot دارای Option های زیادی است که ما در اینجا به برخی از آنها اشاره می کنیم :

نامگذاری محورها با "x" ، "y" و "z" :

```
AxesLable → { "x" , "y" , "z" }
```

رسم رویه بدون نمایش محورهای مختصات :

```
Axes → None
```

رسم رویه بدون نمایش چهارچوب :

```
Boxes → None
```

رسم رویه بدون مش بندی :

Mesh → False

قرار دادن زاویه دید از نقطه (α,β,γ) :

ViewPoint → (α , β , γ)

دستور رسم رویه پارامتری در سه بعد به صورت زیر است :

ParametricPlot3D[{ x[s,t] , y[s,t] , z[s,t] } , { s , s_1 , s_2 } , { t , t_1 , t_2 }]

برای رسم خم پارامتری از دستور زیر استفاده می شود :

ParametricPlot3D[{ x[,t] , y[t] , z[t] } , { t , t_1 , t_2 }]

برای رسم منحنی در مختصات استوانهای و قطبی کروی ابتدا باید بسته زیر را باز کرد :

<<Graphics`ParametricPlot3D`

و سپس دستورات زیر را بکار برد :

CylindricalPlot3D[z[r , θ] , { r , r_1 , r_2 } , { θ , θ1 , θ_2 }]

SphericalPlot3D[r[θ , p] , { θ , θ_1 ,θ_2 } , { ϕ , ϕ_1 , ϕ_2 }]

دو دستور مهم دیگر که به ترتیب برای رسم نمودار تراز و چگالی رویه $z = f(x,y)$ به کار می رود عبارتند از :

ContourPlot[f[x,y] , { x , a , b } , { y , c , d }]

DensityPlot[f[x,y] , { x , a , b } , { y , c , d }]

Option های مربوط به دستورات فوق را می توان از Help پیدا کرد .

برای رسم مجموعهای از نقاط در سه بعد میتوان بسته زیر را لود و دستور مربوطه را به کار برد :

<<Graphics`Graphics3D`

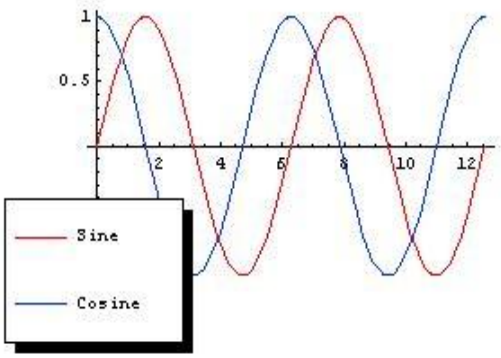
ScatterPlot3D [لیست نقاط]

یکی از مهمترین Option های این دستور PlotStyle → PoinSize [n] می باشد.

علوه بر دستورات ترسیم فوق دستورات دیگری مثل ListPlot3D (برای رسم آرایه ارتفاعدار در سه بعد)، ContourPlot3D (رسم نمودار تراز تابع سه متغیره) ، LogPlot (رسم نمودار لگاریتمی) ، ImplicitPlot (رسم نمودار توابع ضمنی) ، InequalityPlot (رسم نامعادلات) ، Show[Graphics3D [?]]] (برای رسم اشکالی مثل نوار موبیوس و ...) و ... نیز وجود دارند که برای اطلاعات بیشتر به Help سری بزنید.

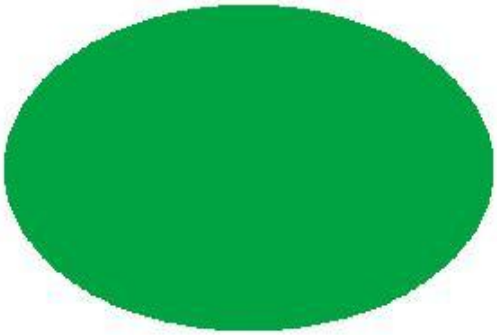
مثال :

```
In[1]:= <<Graphics`Legend`
        Plot[{Sin[x], Cos[x]}, {x, 0, 4 π}, PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0.2, 1]},
         PlotLegend → {"Sine", "Cosine"}]
```



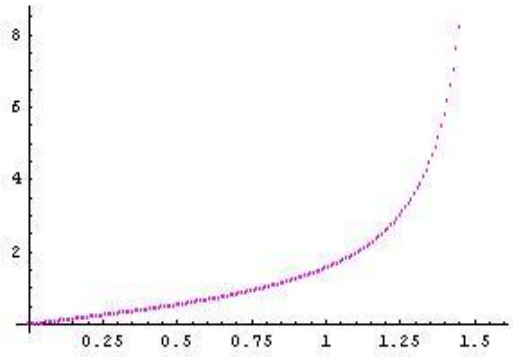

```
Out[2]= - Graphics -
```

```
In[3]:= g = Graphics[Disk[{0, 0}, 2], DefaultColor → Hue[0.4, 1, 0.8 .8]];
        Show[g]
```



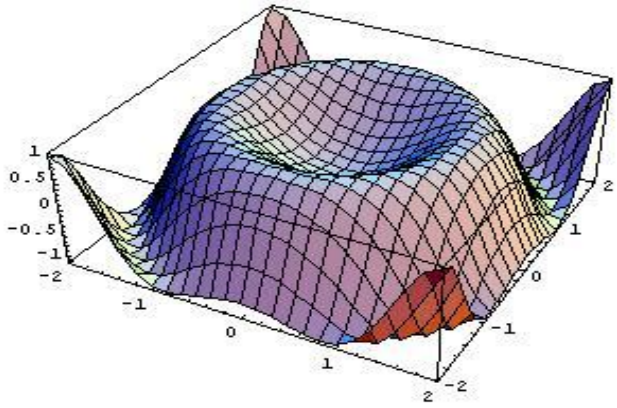
```
Out[4]= - Graphics -
```

```
In[5]:= Show[Graphics[{RGBColor[1, 0, 1], Point /@ Table[{i, Tan[i]}, {i, 0, π/2, 0.01}]}], Axes → True]]
```

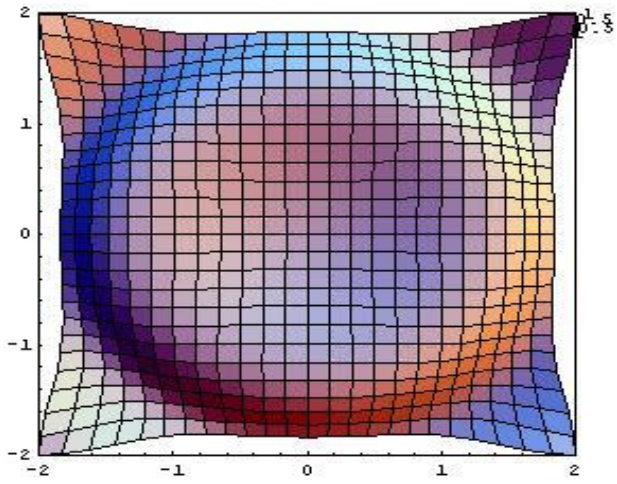



```
Out[5]= - Graphics -
```

In[6]:=
```
Plot3D[Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}]
Plot3D[Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, ViewPoint → {0, 0, 4}]
```

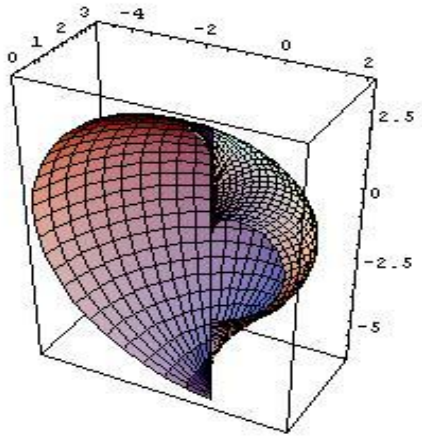


Out[6]= - SurfaceGraphics -



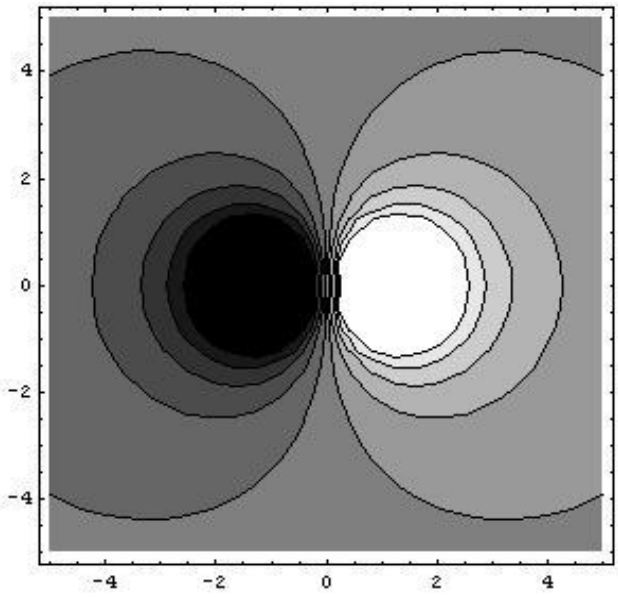
Out[7]= - SurfaceGraphics -

In[8]:=
```
<< Graphics`ParametricPlot3D`
SphericalPlot3D[θ + ϕ, {θ, 0, π}, {ϕ, 0, π}]
```



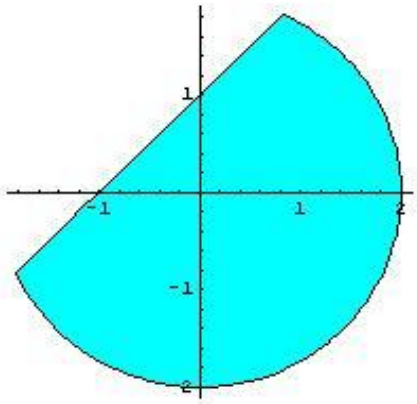
Out[9]= - Graphics3D -

In[10]:=
```
(* Draw Electric Potential Of Electric Dipole *)
q = 1; a = 1; k = 1;
ContourPlot[kq/Sqrt[(x - a)^2 + y^2] - kq/Sqrt[(x + a)^2 + y^2], {x, -5, 5}, {y, -5, 5}]
```



Out[11]= - ContourGraphics -

In[12]:=
```
<< Graphics`InequalityGraphics`
InequalityPlot[{x^2 + y^2 ≤ 4, x - y > -1}, {x, -3, 3}, {y, -3, 3}]
```



Out[13]= - Graphics -

معادلات دیفرانسیل

معادله دیفرانسیل معمولی رابطه ایست میان متغیر وابسته y ، متغیر مستقل x و مشتقات متغیر y نسبت به x که بزرگترین مرتبه مشتق گیری درون معادله را مرتبه آن معادله گویند. Mathematica توانایی حل صریح و یا عددی معادلات دیفرانسیل را دارد. نوشتن یک معادله دیفرانسیل در Mathematica بسیار ساده است مثلا معادله xy''+2y=cos(x) را در Mathematica می توان به صورت :

```
x y''[x] + 2 y[x] == Cos[x]
```

یا :

$$x\partial_{x,x}y[x] + 2y[x] == Cos[x]$$

نوشت. گاهی اوقات ما دنبال جوابی از معادله ایم شرایط اولیه و یا مرزی خاصی را برآورده می کند. نوشتن این شرایط نیز ساده است مثلا شرایط y(0)=1 و یا y''(2)=5 به ترتیب به صورت y[0] == 1 و y''[2] == 5 نوشته می شوند.

دستورات زیر برای حل صریح معادلات دیفرانسیل معمولی به کار می رود :

حل یک معادله دیفرانسیل معمولی :

```
Dsolve[ معادله دیفرانسیل , y[x] , x]
```

حل معادله دیفرانسیل معمولی با شرایط اولیه یا مرزی :

```
Dsolve[ { معادله دیفرانسیل , شرایط اولیه یا مرزی } , y[x] , x]
```

حل دستگاه معادلات دیفرانسیل با شرایط اولیه یا مرزی :

```
Dsolve[ { معادله 1 , معادله 2 , ... , شرط 1 , شرط 2 , ... } , {y1[x] ,y2[x] , ... } , x]
```

خروجی دستورات فوق لیستی شامل تابع جواب معادله دیفرانسیل است که میتوان با دستور [[ ]] به آن دسترسی پیدا کرد.

اکثر معادلات دیفرانسیل جواب صریح ندارند ، برای بدست آوردن جواب عددی یک معادله دیفرانسیل بدون پارامتر با شرایط مشخص از دستور NDSolve استفاده می شود. شکل کلی این دستور به صورت زیر است :

```
NDsolve[ { معادله 1 , معادله 2 , ... , شرط 1 , شرط 2 , ... } , {y1[x] ,y2[x] , ... } , {x , a, b } ]
```

که دستگاه معادلات دیفرانسیل معمولی را به صورت عددی در بازه (a,b) حل میکند. خروجی این دستور یک Interpolating function است . برای کار با این تابع کافی است ابتدا آن را در متغیری ذخیره کنیم و سپس از طریق نماد " ./" از آن استفاده کنیم . به عنوان مثال دستورات زیر را ببینید :

```
Solution = NDSolve [ { y''[x] + 2y'[x] +y[x] == -y[x]2 , y[0]==1 , y'[0]==0 } ,y,{x,0,10} ];
var=Input[ "Enter  Real Number between 0 and 10 : "];
Print[ "y(", var , ") = " , y[var] / . Solution ]
Plot[ y[x] / . Solution , { x , 0, 10 } ]
```

این دستور دارای Option های زیادی می باشد . از جمله میتوان به MaxSteps (در جاهایی که طول بازه زیاد است باید مقدار این Option را زیاد کرد) ، WorkingPrecision ، Method و ... اشاره کرد.

اگر تابع متغیر وابسته شامل چند متغیر مستقل باشد و مشتقات آن نسبت به متغیرهایش در معادله ظاهر شود معادله دیفرانسیل را جزئی گویند. مهمترین معادلات دیفرانسیل جزئی در فیزیک معادله موج ، معادله لاپلاس ، معادله شرویدینگر و معادله گرما می باشد. به عنوان مثال معادله موج در یک بعد به صورت $\frac{\partial^2\psi}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2\psi}{\partial t^2}$ میباشد که در Mathematica می توان آنرا به صورت $\partial_{x,x}\psi[x,t] == \frac{1}{c^2}\,\partial_{t,t}\psi[x,t]$ نوشت . معادلات دیفرانسیل جزئی نیز میتوانند دارای شرط مرزی باشند مثلا شرط ψ[x,0] == Cos[x] یک شرط اولیه برای معادله موج یک بعدی است.

برای حل معادلات دیفرانسیل جزئی از همان دستورات DSolve و NDSolve به صورت زیر استفاده می شود :

Dsolve[{ معادله دیفرانسیل , شرایط اولیه یا مرزی } , y , { x_1 , x_2 , ... }]

NDsolve[{ معادله دیفرانسیل , شرایط اولیه یا مرزی } , y , { x_1 , a_1 , b_1 } , { x_2 , a_2 , b_2 } , ...]

دستور دوم معادله را در ابر مکعب مستطیل $a_1 < x_1 < b_1$ ، $a_2 < x_2 < b_2$ و .... به صورت عددی حل میکند.

اگر تعداد متغیرهای مستقل دو باشد تابع جواب را میتوان توسط دستور زیر رسم کرد (اگر خروجی در متغیر Solution ذخیره شده باشد) :

Plot3D[Evaluate [y[x_1,x_2] / . Solution] , { x_1 , a_1 ,b_1 } , { x_2 , a_2 , b_2 }]

برای حل معادلات دیفرانسیل میتوان از روش سری توانی ، تبدیل لاپلاس ، تبدیل فوریه و ... نیز استفاده کرد ، برای اطلاعات بیشتر به Help مراجعه کنید .

مثال :

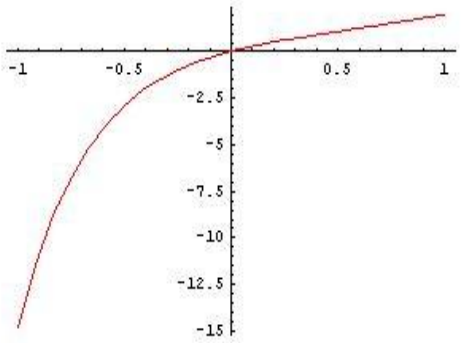
```
In[1]:= DSolve[y''[x] == -y[x], y, x]
```

Out[1]= {{y → Function[{x}, C[1] Cos[x] + C[2] Sin[x]]}}

```
In[2]:= eq = y''[x] + 2 y'[x] - 3 y[x] == 0
        sol1 = DSolve[{eq, y[0] == 0, y[1] == 2}, y, x]
        Plot[y[x] /. sol1, {x, -1, 1}, PlotStyle → Hue[10]]
        y[-1] /. sol1 // N
```

Out[2]= $-3\,y[x] + 2\,y'[x] + y''[x] == 0$

Out[3]= $\left\{\left\{y \to \text{Function}\left[\{x\}, \frac{2\,e^{3-3x}\,(-1+e^{4x})}{-1+e^{4}}\right]\right\}\right\}$



Out[4]= - Graphics -

Out[5]= {-14.7781}

```
In[6]:= sol2 = DSolve[{x''[t] == -y[t], y''[t] == x[t], x[0] == 0, x'[0] == 1, y[0] == 2, y'[0] == 3}, {x[t], y[t]}, t]
```

Out[6]=
$$\left\{\left\{x[t] \to -\frac{1}{2}e^{-\frac{t}{\sqrt{2}}}\left(2\sqrt{2}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - 2\sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - 2\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + 2e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right]\right),\right.$$
$$\left. y[t] \to \frac{1}{2}e^{-\frac{t}{\sqrt{2}}}\left(2\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - \sqrt{2}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + 2e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + 2\sqrt{2}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + 2\sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right]\right)\right\}\right\}$$

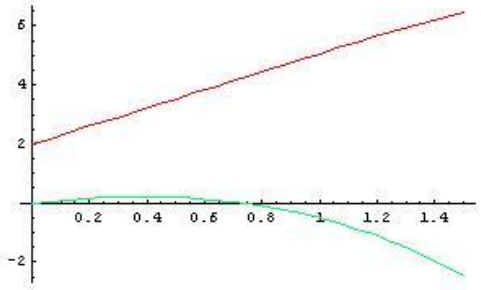
```
In[7]:= x[t_] = sol2[[1, 1, 2]]
```

Out[7]=
$$-\frac{1}{2}e^{-\frac{t}{\sqrt{2}}}\left(2\sqrt{2}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - 2\sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - 2\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + 2e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right]\right)$$

```
In[8]:= y[t_] = sol2[[1, 2, 2]]
```

Out[8]=
$$\frac{1}{2}e^{-\frac{t}{\sqrt{2}}}\left(2\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] - \sqrt{2}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + 2e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + \sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Cos}\left[\frac{t}{\sqrt{2}}\right] + 2\sqrt{2}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right] + 2\sqrt{2}\,e^{\sqrt{2}\,t}\,\text{Sin}\left[\frac{t}{\sqrt{2}}\right]\right)$$

In[9]:=
```
Plot[{x[t], y[t]}, {t, 0, 1.5}, PlotStyle → {Hue[0.4], Hue[3]}]
```



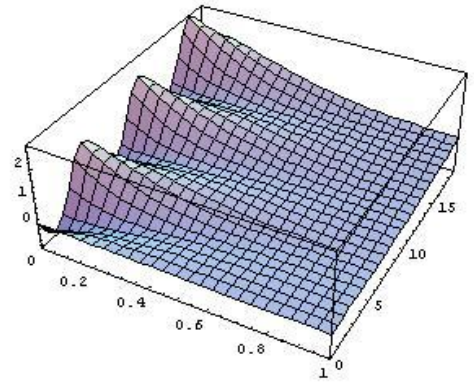
Out[9]= - Graphics -

In[10]:=
```
eq2 := ∂{x,x} ψ[x, t] == 1/v^2 ∂{t,t} ψ[x, t]
```

In[11]:=
```
DSolve[eq2, ψ[x, t], {x, t}]
```

Out[11]=
$$\left\{\left\{\psi[x, t] \to C[1]\left[t - \frac{\sqrt{v^2}\, x}{v^2}\right] + C[2]\left[t + \frac{\sqrt{v^2}\, x}{v^2}\right]\right\}\right\}$$

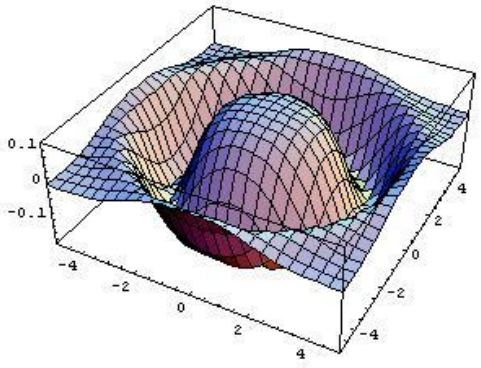
In[12]:=
```
solution =
  u /. First[NDSolve[{∂t u[x, t] == 2 ∂{x,2} u[x, t] / (9 π^2), u[x, 0] == 0, u[0, t] == t/10 - Sin[t], u^(1,0)[1, t] == 0},
    u, {x, 0, 1}, {t, 0, 6 π}]]; Plot3D[solution[x, t], {x, 0, 1}, {t, 0, 6 π}, PlotRange -> All,
  PlotPoints -> 25];
```



In[13]:=
```
sol =
  First[NDSolve[{∂t,t u[t, x, y] == ∂x,x u[t, x, y] + ∂Y,Y u[t, x, y] - Sin[u[t, x, y]], u[0, x, y] == E^-(x^2 + Y^2),
    u^(1,0,0)[0, x, y] == 0, u[t, -5, y] == u[t, 5, y], u[t, x, -5] == u[t, x, 5]}, u, {t, 0, 4}, {x, -5, 5}, {y, -5, 5}]]
```

Out[13]= {u → InterpolatingFunction[{{0., 4.}, {..., -5., 5., ...}, {..., -5., 5., ...}}, <>]}

In[14]:=
```
Plot3D[Evaluate[u[4, x, y] /. sol], {x, -5, 5 }, {y, -5, 5}];
```



آنالیز برداری

در بخش بردارها و ماتریسها درمورد جبر برداری صحبت کردیم ، در اینجا در مورد آنالیز برداری صحبت خواهیم کرد.

برای کار با توابع برداری ابتدا باید بسته زیر را لود کرد :

```
<<Calculus`VectorAnalysis`
```

اولین قدم در آنالیز برداری تعیین دستگاه مختصات موردنظر است ، دستورات مربوط به دستگاه مختصات به صورت زیر است :

نمایش سیستم مختصات کنونی :

```
CoordinateSystem
```

نمایش متغیرهای سیستم مختصات کنونی :

```
Coordinates[]
```

انتخاب سیستم مختصات با متغیرهای پیش فرض Mathematica :

```
SetCoordinates[ نام دستگاه مختصات ]
```

انتخاب سیستم مختصات با متغیرهای var_1 ، var_2 و var_3 :

```
SetCoordinates[ نام دستگاه مختصات [ var1 , var2 , var3 ] ]
```

نمایش حدود متغیرهای سیستم مختصات کنونی :

```
CoordinateRanges[]
```

نمایش حدود متغیرهای یک سیستم مختصات :

```
CoordinateRanges[ نام دستگاه مختصات ]
```

در دستورات فوق نام دستگاه مختصات میتواند یکی از 14 دستگاه مختصات تعریف شده در Mathematica باشد که در زیر نام برخی از آنها آمده است :

- Cartesian : دستگاه مختصات دکارتی که دستگاه مختصات پیش فرض است.

- Spherical : دستگاه مختصات قطبی کروی.

- Cylindrical : دستگاه مختصات استوانه ای.

- ParabolicCylindrical : دستگاه مختصات سهموی استوانه ای

- و ...

بعد از تعیین دستگاه مختصات باید توابع برداری را تعریف کنیم. توسط دستور زیر می توانیم تابع برداری F را تعریف کنیم :

$F = \{ f_1 , f_2 , f_3 \}$

$F [var_1 , var_2 , var_3] = \{ f_1 , f_2 , f_3 \}$

که var1 ، var2 و var3 متغیرهای دستگاه مختصات انتخاب شده و f1 ، f2 و f3 توابع اسکالری از این متغیرها هستند. در ادامه برخی دستورات کار با توابع برداری را ذکر می کنیم :

محاسبه ضرب داخلی دو میدان برداری F و G در سیستم مختصات کنونی :

DotProduct[F , G]

محاسبه ضرب داخلی دو میدان برداری F و G در یک سیستم مختصات :

DotProduct[F , G , نام دستگاه مختصات]

محاسبه ضرب خارجی دو میدان برداری F و G در سیستم مختصات کنونی :

CrossProduct[F , G]

محاسبه ضرب خارجی دو میدان برداری F و G در یک سیستم مختصات :

CrossProduct[F , G , نام دستگاه مختصات]

محاسبه ضرب سه گانه اسکالر سه میدان برداری F و G و H در سیستم مختصات کنونی :

ScalarTripleProduct[F , G , H]

محاسبه ضرب سه گانه اسکالر سه میدان برداری F و G و H در یک سیستم مختصات :

ScalarTripleProduct[F , G , H , نام دستگاه مختصات]

محاسبه گرادیان میدان اسکالر f در دستگاه مختصات کنونی :

Grad[f]

محاسبه گرادیان میدان اسکالر f در یک دستگاه مختصات :

Grad[f , نام دستگاه مختصات]

محاسبه لاپلاسین میدان اسکالر f در دستگاه مختصات کنونی :

Laplacian[f]

محاسبه لاپلاسین میدان اسکالر f در یک دستگاه مختصات :

Laplacian[f , نام دستگاه مختصات]

محاسبه دیورژانس میدان برداری F در دستگاه مختصات کنونی :

Div[F]

محاسبه دیورژانس میدان برداری F در یک دستگاه مختصات :

Div[F , نام دستگاه مختصات]

محاسبه کرل (تاو) میدان برداری F در دستگاه مختصات کنونی :

Curl [F]

محاسبه کرل (تاو) میدان برداری F در یک دستگاه مختصات :

Curl [F , نام دستگاه مختصات]

محاسبه لاپلاسین میدان برداری F در دستگاه مختصات کنونی :

Laplacian [F]

محاسبه لاپلاسین میدان برداری F در یک دستگاه مختصات :

Laplacian [F , نام دستگاه مختصات]

یکی از روشهای دیگر انجام عملیات برداری تعریف آنها و سپس استفاده از آنها میباشد برای آشنایی با این روش به مثال های آخر این بخش رجوع کنید.

یکی دیگر از امکانات Mathematica امکان رسم میدانهای برداری دو و سه بعدی است دستورات انجام این کار به صورت زیر است :

رسم میدان برداری دو بعدی :

```
<<Graphics`PlotField`

PlotVectorField [ F[ x , y ] , { x , x1 , x2 } , { y , y1 , y2 } ]
```

رسم میدان برداری سه بعدی :

```
<<Graphics`PlotField3D`

PlotVectorField3D [ F[ x , y , z ] , { x , x1 , x2 } , { y , y1 , y2 } , { z , z1 , z2 } ]
```

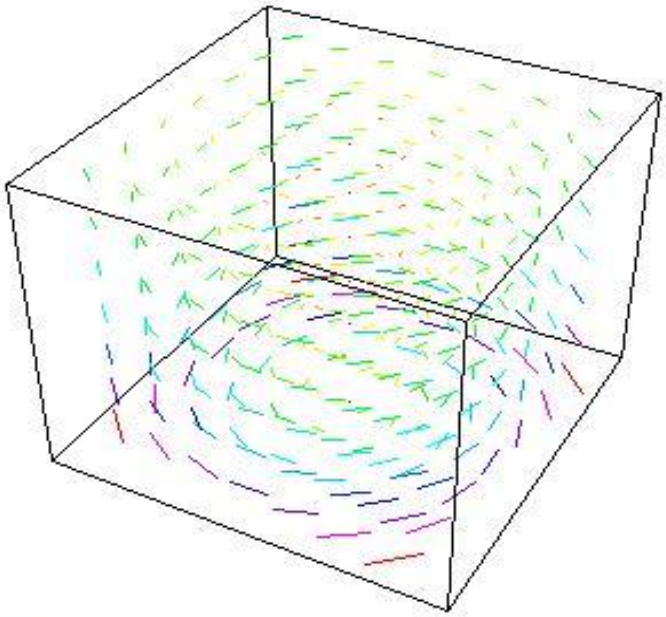
برای دیدن Option های دستورات فوق به Help رجوع کنید.

مثال :

```
In[2]:= <<Calculus`VectorAnalysis`

In[3]:= {CoordinateSystem, Coordinates[]}

Out[3]= {Cartesian, {Xx, Yy, Zz}}

In[4]:= SetCoordinates[Spherical]
        SetCoordinates[Spherical[r, θ, ϕ]]
        CoordinateRanges[]

Out[4]= Spherical[Rr, Ttheta, Pphi]

Out[5]= Spherical[r, θ, ϕ]

Out[6]= {0 ≤ r < ∞, 0 ≤ θ ≤ π, -π < ϕ ≤ π}

In[7]:= F = {r, θ^2, ϕ + 1};
        G = {-r, 0, 0};
        H = {1, 2, 3};
        DotProduct[F, G]
        DotProduct[F, G, Cartesian]
        ScalarTripleProduct[F, G, H]
        ScalarTripleProduct[F, G, H, Cartesian]

Out[10]= -r^2 Cos[θ^2]

Out[11]= -r^2

Out[12]= r^2 Cos[1 + ϕ] Sin[2] Sin[3] Sin[θ^2] - r^2 Cos[3] Sin[2] Sin[θ^2] Sin[1 + ϕ]

Out[13]= -2 r + 3 r θ^2 - 2 r ϕ

In[14]:= Grad[5 r^2 + 3 Cos[θ] Sin[ϕ], Spherical[r, θ, ϕ]]
         v = {r Sin[θ] Cos[ϕ], r Sin[θ] Sin[ϕ], r Cos[θ]}
         Div[v, Spherical[r, θ, ϕ]]
         Curl[v, Spherical[r, θ, ϕ]]
         TrueQ[Grad[Div[v]] - Curl[Curl[v]] == Laplacian[v]]

Out[14]= {10 r, -(3 Sin[θ] Sin[ϕ])/r, (3 Cos[ϕ] Cot[θ])/r}

Out[15]= {r Cos[ϕ] Sin[θ], r Sin[θ] Sin[ϕ], r Cos[θ]}

Out[16]= (Csc[θ] (3 r^2 Cos[ϕ] Sin[θ]^2 + 2 r^2 Cos[θ] Sin[θ] Sin[ϕ]))/r^2

Out[17]= {(Csc[θ] (r^2 Cos[θ]^2 - r^2 Cos[ϕ] Sin[θ] - r^2 Sin[θ]^2))/r^2,
          (Csc[θ] (-2 r Cos[θ] Sin[θ] - r Sin[θ] Sin[ϕ]))/r, (-r Cos[θ] Cos[ϕ] + 2 r Sin[θ] Sin[ϕ])/r}

Out[18]= True
```

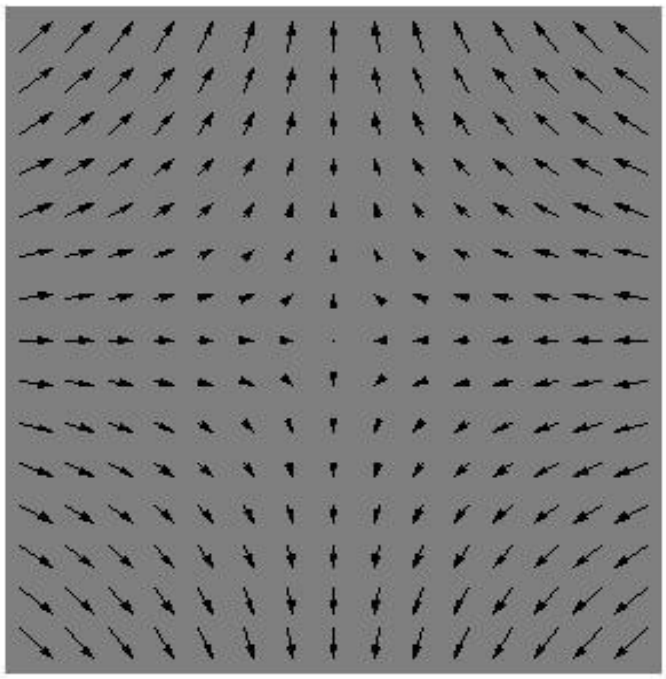
In[19]:=
```
<< Graphics`PlotField3D`
```

In[20]:=
```
PlotVectorField3D[{y, -x, 0}/z, {x, -1, 1}, {y, -1, 1}, {z, 1, 3},
 ColorFunction → Hue]
```



Out[20]= - Graphics3D -

In[21]:=
```
<< Graphics`PlotField`
PlotVectorField[{-x, y}, {x, -2, 2}, {y, -2, 2}, Background → GrayLevel[0.5]]
```



Out[22]= - Graphics -

In[23]:=
```
(* Define And Use *)
```

In[24]:=
```
Clear[i];
∇ /: ∇[i_]×{fx_, fy_, fz_} := {∂fz/∂y[i] - ∂fy/∂z[i], ∂fx/∂z[i] - ∂fz/∂x[i], ∂fy/∂x[i] - ∂fx/∂y[i]}
∇ /: ∇[i_]·{fx_, fy_, fz_} := Simplify[∂fx/∂x[i] + ∂fy/∂y[i] + ∂fz/∂z[i]]
∇ /: ∇[i_](f_) := {∂f/∂x[i], ∂f/∂y[i], ∂f/∂z[i]} // Simplify
```

∇(■) ∇·■ ∇×■

In[28]:=
```
∇[1] (x[1] + y[1] y[1])
∇[2] · {x[1] y[2], y[2] + 3 y[1] x[2], z[1]}
```

Out[28]=
```
{1, 2 y[1], 0}
```

Out[29]=
```
1
```

آمار در Mathematica

در این بخش قصد داریم تعدادی از دستورات کار با داده ها را بیان کنیم. برای دیدن اطلاعات بیشتر به Help رجوع کنید. در ابتدا دادهها را در یک لیست یک بعدی (بردار) قرار می دهیم :

نام لیست دادهها = { داده 1 , داده 2 , ... , داده n }

بعد از دستورات زیر استفاده می کنیم :

محاسبه مینیمم داد ه ها :

Min[نام لیست داده ها]

محاسبه ماکزیمم داده ها :

Max[نام لیست داده ها]

محاسبه تعداد داده ها :

Length[نام لیست داده ها]

محاسبه میانگین :

Mean[نام لیست داده ها]

محاسبه میانه :

Median[نام لیست داده ها]

محاسبه مجموع داده ها :

Total[نام لیست داده ها]

محاسبه واریانس داده ها :

Variance[نام لیست داده ها]

محاسبه انحراف معیار داده ها :

StandardDeviation[نام لیست داده ها]

محاسبه چهارک q ام :

Quantile[نام لیست داده ها , q]

مرتب کردن داده ها :

Sort[نام لیست دادهها]

رسم داده ها :

ListPlot[نام لیست داده ها , Options]

از Option های مهم این دستور زیر اشاره کرد :

PlotStyle → PointSize[n]

PlotRange → { { x_{min} , x_{max}} , { y_{min} , y_{max}} }

برای Fit کردن نمودار دلخواه روی داده ها از دستور زیر استفاده می شود :

f = FindFit[نام لیست داد ه ها , نام متغیر , لیست پارامترها , تابع مورد نظر با چند پارامتر آزاد و یک متغیر]

برای رسم تابع Fit شده نیز میتوان از دستور زیر استفاده کرد :

Plot[تابع با پارامترها /. f , { x , x_{min} , x_{max} }]

برای درونیابی از دستور زیر استفاده می شود :

f = Interpolation[نام لیست داده ها]

و با دستور f[x] میتوان مقدار برآوردی برای عدد x را بدست آورد.

مثال :

In[1]:=
```
G1 = {12.3, 14.5, 17.6, 13.4, 8, 9.75, 14.6, 18, 20}
```

Out[1]=
```
{12.3, 14.5, 17.6, 13.4, 8, 9.75, 14.6, 18, 20}
```

In[2]:=
```
Min[G1]
Max[G1]
Sort[G1]
Length[G1]
Total[G1]
Mean[G1]
% / Length[G1]
Variance[G1]
Total[(G1 - Mean[G1])^2] / (Length[G1] - 1)
StandardDeviation[G1]
Sqrt[Variance[G1]]
Median[G1]
Quantile[G1, 1/4]
```

Out[2]= 8

Out[3]= 20

Out[4]= {8, 9.75, 12.3, 13.4, 14.5, 14.6, 17.6, 18, 20}

Out[5]= 9

Out[6]= 128.15

Out[7]= 14.2389

Out[8]= 1.5821

Out[9]= 15.2961

Out[10]= 15.2961

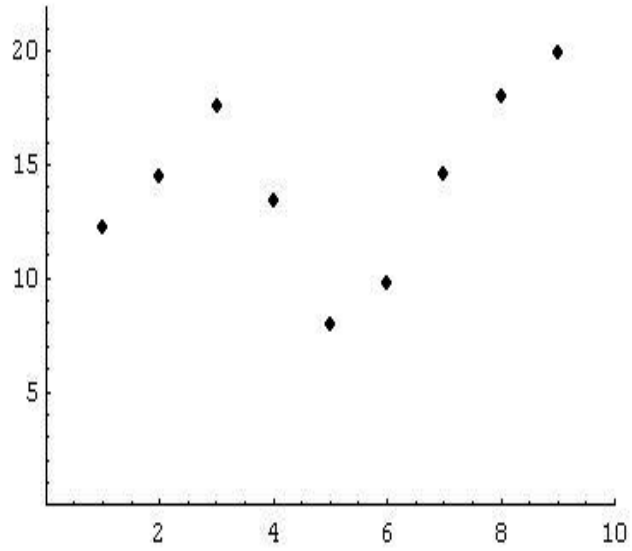
Out[11]= 3.91102

Out[12]= 3.91102

Out[13]= 14.5

Out[14]= 12.3

In[15]:= plot0 = ListPlot[G1, PlotStyle → PointSize[0.02], PlotRange → {{0, 10}, {0, 22}}]

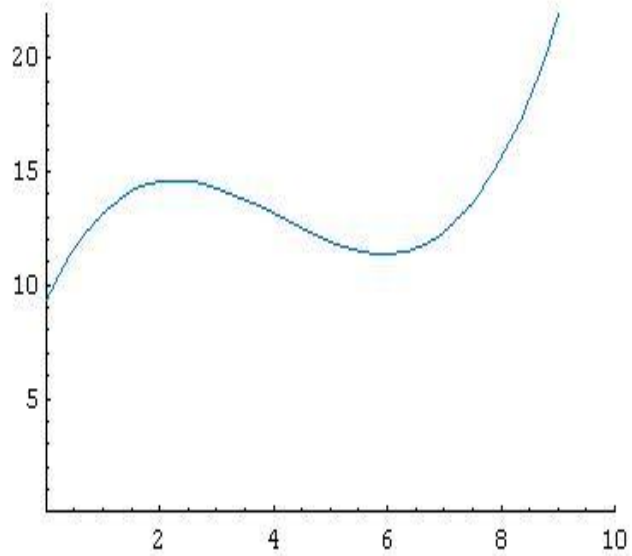


Out[15]= - Graphics -

In[16]:= f1 = FindFit[G1, c + d x + h x^2 + g x^3, {c, d, h, g}, x]

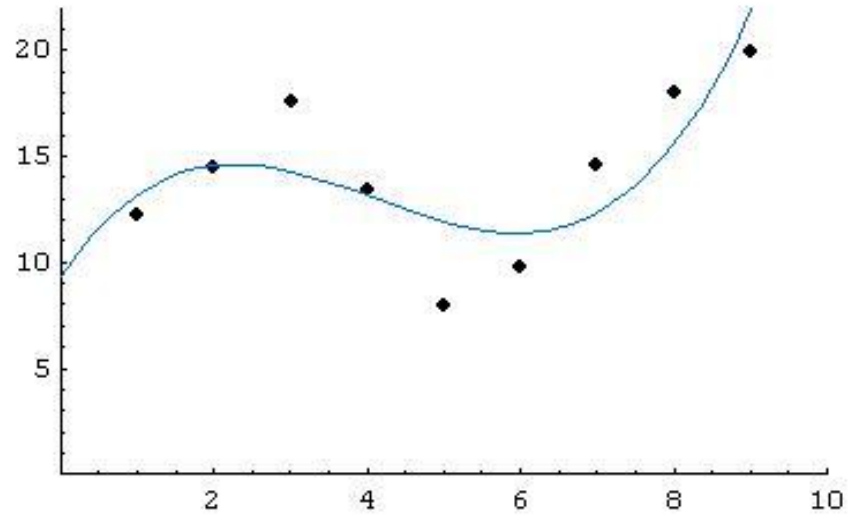
Out[16]= {c → 9.34603, d → 5.30995, h → -1.61183, g → 0.130598}

In[17]:= plot1 = Plot[c + d x + h x^2 + g x^3 /. f1, {x, 0, 10}, PlotRange → {{0, 10}, {0, 22}},
PlotStyle → Hue[0.6]]



Out[17]= - Graphics -

In[18]:=
```
Show[plot0, plot1]
```



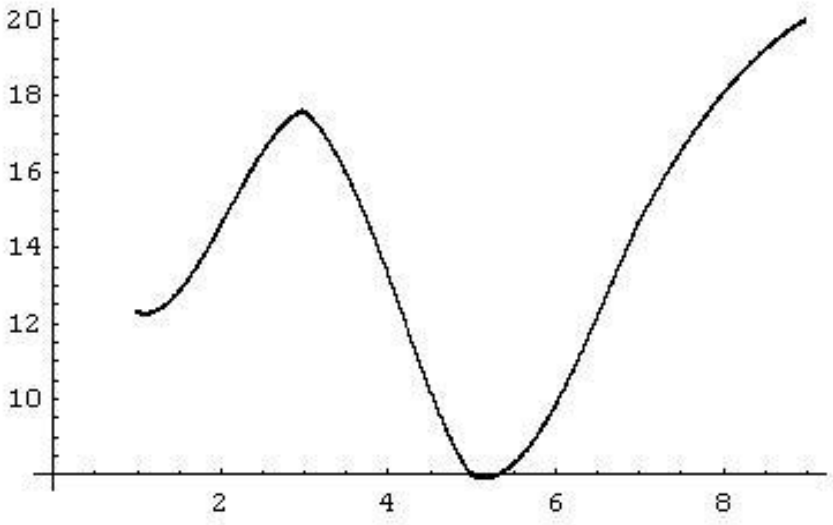
Out[18]= - Graphics -

In[19]:=
```
Int = Interpolation[G1]
{Int[3], Int[9], Int[4]}
```

Out[19]= InterpolatingFunction[{{1., 9.}}, <>]

Out[20]= {17.6, 20., 13.4}

In[21]:=
```
var = Table[{x, Int[x]}, {x, 1, 9, 0.01}];
ListPlot[var]
```



Out[22]= - Graphics -

برنامه نویسی درMathematica

برای نوشتن برنامه در Mathematica ابتدا باید متغیرها و نوع آنها را بشناسیم . متغیرها از لحاظ محتوا دارای انواع مختلفی مثل Integer ، Real ، Complex ، String ، Graphic ، List و ... میباشند. متغیرها را از لحاظ حوزه کاربرد هم می توان به دو دسته تقسیم کرد :

1 – متغیرهای سراسری : این متغیرها به طور عادی تعریف میشوند و در کل برنامه شناخته شده اند.

2 – متغبرهای محلی : متغیرهایی که فقط در یک حوزه خاص تعریف و مقداردهی می شوند و مقدار خود را در بیرون از این حوزه حفظ نمی کنند.

متغیرهای محلی معمولا در دستور Module استفاده می شوند :

Module[{ x , y , ... } , دستور یا دستورات]

Module[{ $x=x_0$, $y=y_0$, ... } , دستور یا دستورات]

در دستورات فوق x ، y و ... متغیرهای محلی اند که فقط درون دستور Module مقدار خود را حفظ می کنند و در مورد دوم مقدار دهی اولیه نیز می شوند. برای به کار بردن چنددستور در Module از علامت **نقطه ویرگول** (;) برای جداکردن دستورات استفاده می شود. به عنوان مثال خروجی برنامه فوق به ترتیب 3 ، 5 و 3 است .

```
a=3; Print[a]
Module[ {a} , a=5 ; Print[a] ]
Print[ a]
```

عملگرها را میتوان به سه دسته محاسباتی ، رابطه ای و منطقی تقسیم کرد که اولویت آنها در عبارات به همین صورت است.

عملگرهای محاسباتی به ترتیب تقدم عبارتند از :

1 – توان رسانی (^)

2 – ضرب و نقسیم (* و /)

3 – جمع وتفریق (+ و –)

عملگرهای رابطه ای دارای اولویت یکسان هستند و عبارتند از :

- تساوی (==)
- نامساوی (=!)
- بزرگتر (>)
- کوچکتر (<)
- بزرگترمساوی (=>)
- کوچکترمساوی (=<)

عملگرهای منطقی به جز عملگر نقیض که اولویت بالاتری دارد تقدم یکسانی دارند وعبارتند از :

- نقیض (!)
- And منطقی (&&)
- Or منطقی (||)
- Xor منطقی (Xor)

جدول مربوط به عملگرهای منطقی به صورت زیر است :

p	q	! p	! q	p && q	p || q	Xor[p,q]
True	True	False	False	True	True	False
True	False	False	True	False	True	True
False	True	True	False	False	True	True
False	False	True	True	False	False	False

نکته : برای تغییر اولویت عملگرها از پرانتز استفاده میشود.

نکته : برای نوشتن توضیح تک خطی در متن برنامه آن را بین (* *) بنویسید.

روشهای جایگذاری و مقداردهی به متغیرها (توسط دو عملگر = و = :) قبلا بیان شد، چند روش دیگر جایگذاری عبارتند از :

- i ++ : پس افزایش i به اندازه واحد .
- i - - : پس کاهش i به اندازه واحد .
- ++ i : پیش افزایش i به اندازه واحد .
- - - i : پیش کاهش i به اندازه واحد .
- i += a : افزایش i به اندازه a .

- i - = a : کاهش i به اندازه a .
- i *= a : تبدیل i به i*a .
- i /=a : تبدیل i به i /a .

در روشهای فوق ، در پیش افزایش (کاهش) ابتدا متغیر افزایش (کاهش) می یابد و سپس وارد محاسبه می شود ولی در پس افزایش (کاهش) ابتدا متغیر وارد محاسبه می شود بعد افزایش (کاهش) می یابد.

آرایه ها یکی از عناصر مهم در برنامه نویسی اند که در بخش مربوط به لیست ها و ماتریس ها مورد بررسی قرار گرفت.

در برخی از برنامه ها نیاز به ورود و خروج داده ها داریم (درمورد ورود و خروج با فایل بعدا صحبت میشود.) دستورات ورودی Input و InputString و ... و دستورات خروجی Print ، StylePrint ، GridBox و ... می باشند. همچنین دستوراتی تحت عنوان کلی Number Formatting برای فرمت بندی اعداد خروجی وجود دارد که در صورت نیاز میتوان به آنها رجوع کرد.

همانطور که می دانیم برنامه های ساختاریافته (که مقدمه برنامه نویسی شی گرا است) سه ساختار کنترل وجود دارد ، ساختار دنباله ، ساختار انتخاب (تصمیم گیری یا شرطی) و ساختار تکرار . اجرای خط به خط دستورات به ترتیب قرارگیری آنها همان ساختار دنباله است . ساختار دوم ساختار تصمیم گیری است ، مهمترین دستورات شرطی Mathematica عبارتند از :

اگر شرط درست باشد دستور اجرا می شود وگرنه خیر :

شرط ; / دستور

ساختار شرطی If : اگر شرط درست باشد دستور 1 وگرنه دستور 2 اجرا می شود.:

If [شرط , دستور 1 , دستور 2]

ساختار شرطی Which : اگر شرط 1 درست باشد دستور 1 ، اگر شرط 2 درست باشد دستور 2 ، ... اجرا می شود :

Which[شرط 1 , دستور 1 , شرط 2 , دستور 2 , ...]

ساختار Switch : عبارت را محاسبه میکند بعد با فرم ها مقایسه می کند و با هرکدام همخوانی داشت مقدار متناظر با آن را برمی گرداند :

Switch[عبارت , فرم 1 , مقدار 1 , فرم 2 , مقدار 2 , ...]

ساختار تکرار برای تکرار یک عمل به کار می رود. مهمترین دستورات مربوط به این ساختار عبارتند از :

ساختار Do : دستور یا دستورات را به ازای مقادیر متغیر i از a تا b با گام c اجرا میکند . مقدار پیش فرض a و c برابر 1 است

Do[دستور یا دستورات , { i , a , b , c }]

ساختار تکرار While : تا زمانی که شرط درست باشد دستور یا دستورات را انجام می دهد :

While [شرط , دستور یا دستورات]

ساختار For : در این دستور یک متغیر شمارنده (i) داریم که در start مقدار اولیه به آن می دهیم بعد در test شرط لازم برای اانجام ادامه عملیات چک می شود تا در صورت درست بودن دستورات قسمت body انجام شود ؛ سپس در expr مقدار شمارنده تغییر کرده و این روند تکرار می شود :

For [start , test , expr , body]

مثلا دستور زیر اعداد 1 تا 10 را چاپ می کند :

For [i=1 , i < 11 , i = i + 1 , Print [i]]

نکته : بین دستورات از نقطه ویرگول " ; " استفاده می شود .

در ساختار تکرار یکسری دستورات کنترلی وجود دارند که دو مورد از آنها عبارتند از :

خروج از نزدیکترین حلقه :

Break [ ]

رفتن به عنصر [نام برچسب]Label :

Goto [نام برچسب]

توصیه برنامه نویسی : تا حد امکان ار دستور Goto استفاده نشود .

چند دستورمهم دیگر که ممکن است مفید واقع شوند عبارتند از :

این دستور باعث توقف اجرای برنامه می شود :

Abort[]

این دستور باعث توقف اجرای برنامه تا حداقل n ثانیه می شود :

Pause[n]

این دستور امکان اجرای دستورات را تا n ثانیه می دهد و اگر اجرای دستور بیشتر طول بکشد آن را متوقف می کند :

TimeConstrained[دستور یا دستورات , n]

این دستور امکان اجرای دستوراتی که حافظه مورد نیاز آنها تا n بایت باشد را می دهد وگرنه اجرای دستور را متوقف می کند :

MemoryConstrained[دستور یا دستورات , n]

مثال :

In[1]:=
```
x = 10;
Print[x];
Module[{x = 3}, Print[x - 1]; x = x + 1; Print[x]]
Print[x]
```

```
10

2

4

10
```

In[5]:=
```
f[0] := 1; f[1] := 1; f[n_] := f[n - 1] + f[n - 2];
TimeConstrained[f[20], 2]
TimeConstrained[f[30], 2]
```

Out[6]= 10946

Out[7]= $Aborted

In[8]:=
```
a = Table[Input["Enter a number :"], {i, 3}, {j, 3}];
a // MatrixForm
Print["|A|= ", Det[a]]
```

Out[9]//MatrixForm=

$$\begin{pmatrix} 2.3 & 3.6 & 5.6 \\ 3 & 9 & -89.3 \\ 35 & -3.6 & 12.5 \end{pmatrix}$$

```
|A|= -13691.9
```

In[11]:=
```
Clear[x0, x, f];
f[x_] = Input["Enter a function ? "]
x0 = Input["Enter a number near root ? "];
ϵ = Input["Enter ϵ ? "]
x1 = x0 - f[x0]/f'[x0];
While[Abs[x1 - x0] > ϵ, x0 = x1; x1 = x0 - f[x0]/f'[x0]];
Print["The root is : ", x1 // N]
```

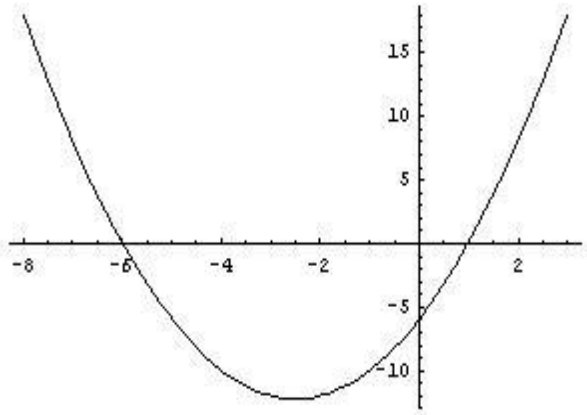
Out[12]= $x^2 - \text{Sin}[2x]$

Out[14]= 0.00001

```
The root is : -4.23559×10^-14
```

In[18]:=
```
(* "This Program Solve the eq ax^2+bx+c "*)
a = Input["Enter non zero number as a ? "];
b = Input["Enter b ? "];
c = Input["Enter c ? "];
sol = Solve[a x^2 + b x + c == 0, x];
α = sol[[1, 1, 2]] // N;
β = sol[[2, 1, 2]] // N;
Print["The Solution is : "]
Print[α]
Print[β]
Plot[a x^2 + b x + c, {x, α - 2, β + 2}]
```

```
The Solution is :
-6.
1.
```



Out[27]= - Graphics -

In[28]:=
```
For[i = 1, i ≤ 10, i++, If[PrimeQ[i], Print[i]]]
```

```
2
3
5
7
```

In[29]:=
```
fact = 1;
n = Input[];
Do[fact *= i, {i, n}];
TimeConstrained[Print[n, "!= ", fact], 1, Print["This Evaluation Take More Than 5 sec "]]
```

```
30!= 265252859812191058636308480000000
```

In[33]:=
```
x = Input["Enter a number ? "];
sgn[x_] := If[x ≠ 0, x/Abs[x], 0];
Print["sign[", x, "]= ", sgn[x]]
```

```
sign[-9]= -1
```

In[36]:=
```
(* "This Program Evaluate the ∫_a^b f[x]dx by remman series method" *)
Clear[f, a, b, x];
f[x_] = Input["Enter a function ? "];
a = Input["Enter a ?"];
b = Input["Enter b ?"];
n = 1000;
Δx = (b - a)/n;
s = 0;
For[i = 1, i ≤ n, i++, s = s + f[a + i Δx]];
integral = s Δx;
Print["a = ", a, " , b = ", b]
Print["∫_a^b(", f[x], ")dx = ", integral // N]
```

```
Null
```

```
a = 0 , b = π
```

$$\int_a^b (\text{Sin}[x])dx = 2.$$

In[48]:=
```
f[x_] = Input["Enter a function ?"];
a = Input["Enter a number ? "];
ϵ = 10^-30;
h = 0.1;
f0 = f[a + h];
h = h / 2;
f1 = f[a + h];
While[Abs[f1 - f0] ≥ ϵ, h = h/2; f0 = f1; f1 = f[a + h]];
Print["Limit_(x→a) (", f[x], ") =", f1 // N]
```

$$\text{Limit}_{x \to a}\left(\frac{\text{Sin}[x]}{x}\right) = 1.$$

In[57]:=
```
Do[
 Print[i]; If[i == 5 , Abort[]]
 , {i, 1, 10}]
```

```
1
2
3
4
5
```

Out[57]= $Aborted

ورود و خروج داده ها در Mathematica

در برخی موارد نیاز داریم روی داده هایی کار کنیم که در یک فایل متنی ذخیره شده اند (این فایل می تواند خروجی یک برنامه محاسباتی باشد) . برای ورود داده ها به Mathematica از دستور Import استفاده می شود. (در حالت کلی این دستور برای وارد کردن داده ها از انواع فایل ها به Mathematica مورد استفاده قرار می گیرد) . در دستور Import باید نام فایل را به همراه طریقه ورود آن را مشخص کرد. برخی از طرق ممکن عبارتند از :

"CSV" : ورود داده ها جدول وار با جداکننده کاما .

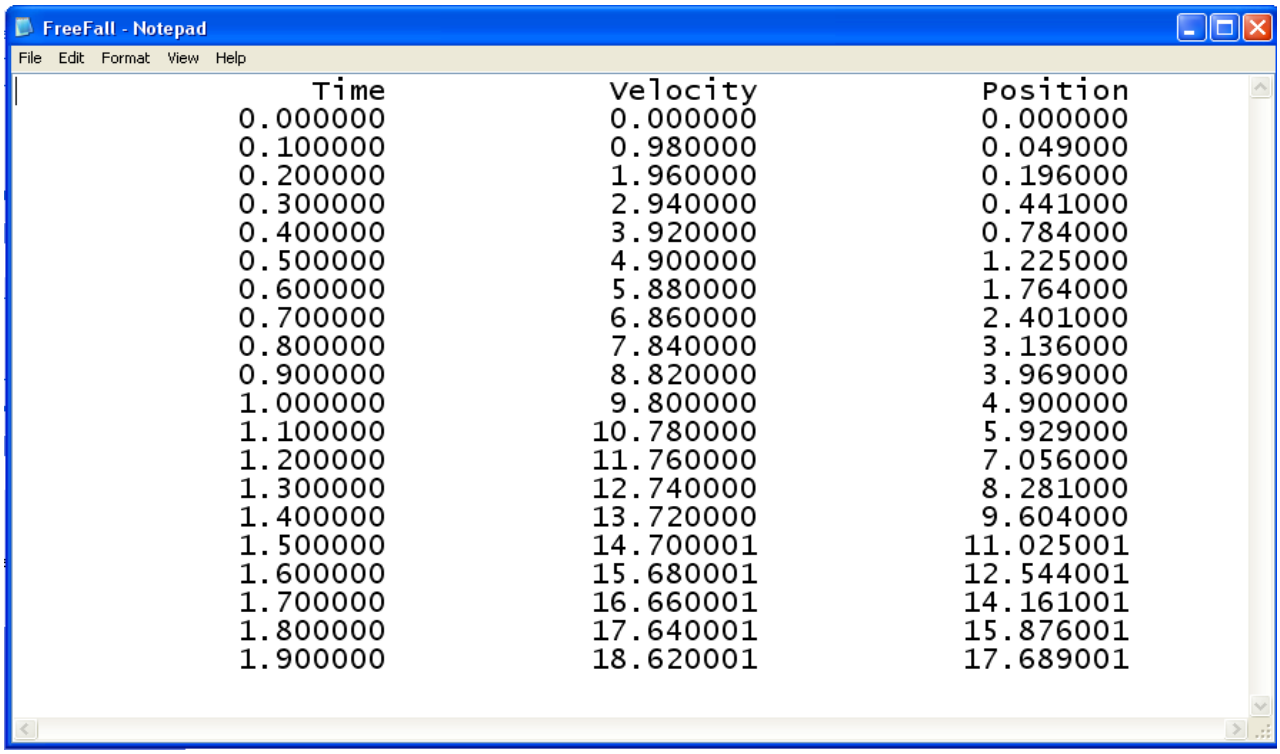
"Lines" : ورود داده ها به صورت خط به خط .

"List" : ورود داده ها به صورت لیست هایی شامل اعداد و رشته های هر خط .

"Table" : ورود داده ها به صورت آرایه دوبعدی از اعداد و رشته ها ..

"TSV" : ورود داده ها جدول وار با جداکننده Tab .

ما این قسمت را با یک مثال ساده توضیح می دهیم. فرض کنید فایل متنی (FreeFall.txt) داریم که حاوی دادههایی درمورد زمان ، سرعت و مکان یک جسمی باشد که سقوط آزاد انجام می دهد . (شکل زیر) .



برای وارد کردن داده های این فایل در محیط Mathematica از دستورات زیر استفاده می کنیم :

```
SetDirectory[ "C:\Documents and Settings\Administrator\Desktop" ]

x = Import[ "FreeFall.txt" , "Table" ]
```

نحوه ورود داده ها را "Table" قرار داده ایم تا متغیر x به صورت یک آرایه (در اینجا دوبعدی با مرتبه 3×21) باشد . عناصر x را میتوان توسط x[[i , j]] که 0 < i < 22 و 0 < j < 4 نشان داد . مثلا داریم :

```
x[[1,1]]="Time" ,  x[[3,3]]=0.049000
```

با دستور Dimensions[x] میتوان به ابعاد x (حدود i و j) پیبرد. حال میتوان هر عملیات دلخواهی روی عناصر x انجام داد. به عنوان مثال دستور

```
Max[ Table[ x[[i ,2]] , { i , 2 , 21} ]]
```

ماکزیمم سرعت شی را می دهد . یا با دستور

```
Export[ "velocity.txt" , Table[ x[[i ,2]],{ i , 2 , 21} ], "Table"]
```

میتوان سرعت ها را در فایلی دیگر بنام velocity.txt ریخت. برای رسم نمودار مکان–زمان میتوان دستور زیر را به کار برد.

```
g= Graphics[ { RGBColor[ 1,0,0] , Point/@Table[ { x[[i,1]] , x[[i,3]] } ,{ i , 2 ,21 } ] }, Axes →True , AxesLabel → { "Time" , "Position" } ];

Show[ g ]
```

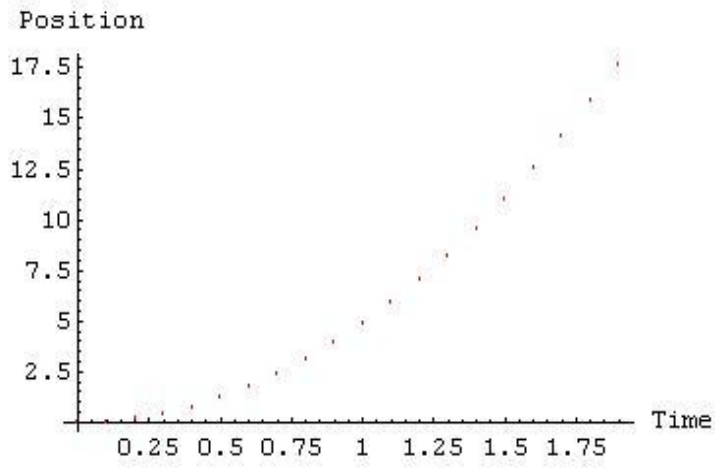
یا با دستور زیر میتوان یک سهمی را روی داده های مکان–زمان Fit کرد و نمودار سهمی و نقاط را با هم رسم کرد :

```
Pos=Table[ {x[[i , 1]] , x[[i , 3]] }, { i , 2, 21} ]

f1= FindFit[Pos , d+c r , { c , d } , r ]

Plot[ d+c r /. f1, {r ,0,1.9} , Epilog →Prepend[ Point/@ Pos , PointSize[0.02]]] ;
```

خروجی دستور فوق مقادیر پارامترهای c و d (شیب و عرض از مبدا خط) و نموداری شامل رسم نقاط (t , x) و خط Fit شده روی آنها می باشد.

مثال :

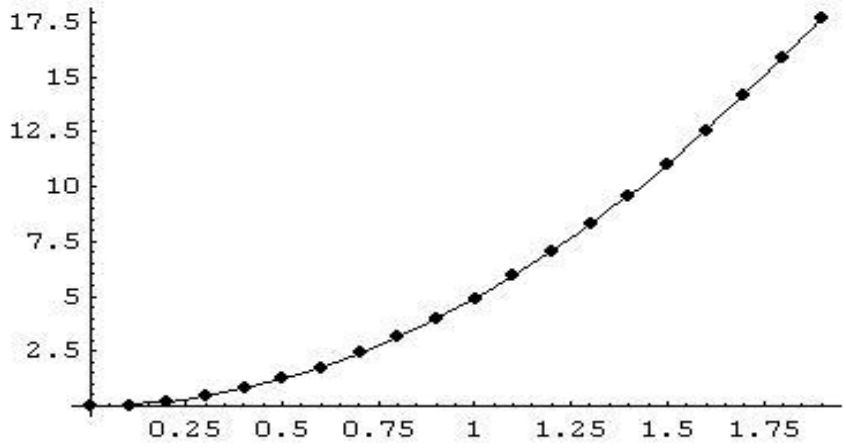
```
In[1]:= SetDirectory["C:\Documents and Settings\Administrator\Desktop"]
Out[1]= C:\Documents and Settings\Administrator\Desktop

In[2]:= x = Import["FreeFall.txt", "Table"];

In[3]:= Dimensions[x]
        {x[[1, 1]], x[[3, 3]]}
Out[3]= {21, 3}

Out[4]= {Time, 0.049}

In[5]:= Max[Table[x[[i, 2]], {i, 2, 21}]]
Out[5]= 18.62

In[6]:= Export["velocity.txt", Table[x[[i, 2]], {i, 2, 21}], "Table"]
Out[6]= velocity.txt

In[7]:= g =
          Graphics[{RGBColor[1, 0, 0],
            Point /@Table[ { x[[i, 1]], x[[i, 3]]}, {i, 2, 21}]}, Axes → True,
           AxesLabel → {"Time", "Position"}];
        Show[g]
```




```
Out[8]= - Graphics -
```

In[9]:=
```
Clear[c, d, b];
pos = Table[{x[[i, 1]], x[[i, 3]]}, {i, 2, 21}];
f1 = FindFit[pos, d + c r + b r^2, {b, c, d}, r]
Plot[d + c r + b r^2 /. f1, {r, 0, 1.9},
 Epilog → Prepend[Point /@pos, PointSize[0.02]]]
```

Out[11]= $\{b \to 4.9,\ c \to -7.88904 \times 10^{-7},\ d \to 1.2013 \times 10^{-7}\}$



Out[12]= - Graphics -

پایان