فهرست مطالب

مقدمه

Sage یک نرم افزار رایگان است که از شاخه های جبر، هندسه، نظریه اعداد، رمز نگاری، محاسبات عددی و شاخه های مرتبط پشتیبانی می کند. هدف نهایی سیج، ایجاد یک نرم افزار رایگان و متن باز با قابلیت نرم افزار هایی چون ... matlab, maple, mathematica, maxima, magma است.

اولین نسخه سیج در سال ۲۰۰۵ تولید شد. مدیریت این پروژه بر عهده ی William Stein یک ریاضیدان از دانشگاه واشنگتن بود. او دریافته بودکه نرم افزارهای ریاضی زیادی وجود دارندکه در زبان های برنامه نویسی مختلف نوشته شده اند و زمانی که نیاز است تاکار های متفاوتی را انجام دهیم بایستی با تک تک این زبان ها آشنا باشیم اما در نرم افزار سیج که بر اساس زبان برنامه نویسی python نوشته شده است، حتی نیازی به مسلط بودن بر زبان python نیست، تنها کافیست کمی به زبان انگلیسی تسلط داشت.

اهدافی که Sage دنبال می کند عبارتند از :

۱) کاربردی. کاربران اصلی سیج دانشجویان، مدرسان و محققان ریاضیات می باشند. هدف، تولید نرم افزاری در ساختارهای ریاضیات مانند جبر، هندسه، نظریه اعداد و ... و شاخه های مرتبط است.

۲) کارایی. سیج از نرم افزارهای بسیار بهینه شده مانند GMP, PARI, NTL استفاده می‌کند و در عملیات اصلی بسیار سریع است.

۳) رایگان و متن باز بودن. کد منبع به طور کاملاً مناسبی در دسترس و خوانا است. کاربران می‌توانند اینکه سیستم در هنگام اجرا واقعا چه کاری انجام می‌دهد را درک کنند.

۴) مشارکت. این نرم افزار از نظر ظاهر اجرا شباهت های زیادی با اکثر نرم افزارهای ریاضیات موجود دارد.

۵) محیط کاربری مناسب. می‌توان با مشاهده متن، کد عملکرد را تحلیل کرد.

بخش اول

مروری بر پیش نیاز ها

۲.۱. مروری بر پیش نیاز ها

تعریف ۱.۲.۱ . یک رابطه ترتیب $<$ روی یک مجموعه یک جمله ای های حلقه $k[x_1, \dots, x_n]$ را یک ترتیب یک جمله ای می نامیم هرگاه

i. $<$، یک رابطه ترتیب کلی (خطی) باشد.

ii. $<$، با ضرب در $K[x]$ سازگار باشد. یعنی اگر X^α، X^β و X^γ یک جمله ای های دلخواه در $k[x_1, \dots, x_n]$ باشند، در اینصورت

$$X^\alpha > X^\beta \ \rightarrow \ X^\gamma . X^\alpha > X^\gamma . X^\beta$$

iii. $<$، خوشترتیب است. یعنی هر مجموعه ناتهی از یکجمله ایهای $k[x_1, \dots, x_n]$ نسبت به $<$، دارای کوچکترین عضو باشد.

تعریف ۲.۲.۱. ترتیب الفبایی (lexicographic order)

گوییم $\alpha >_{lex} \beta$ هرگاه در بردار تفاضل $\alpha-\beta \in Z^n$ چپ ترین درایه غیر صفر مثبت باشد. می نویسیم $X^\alpha >_{lex} X^\beta$ هرگاه $\alpha >_{lex} \beta$

تعریف ۳.۲.۱ . ترتیب الفبایی مدرج (graded lex order)

گوییم $\alpha >_{grlex} \beta$ هرگاه $|\alpha| > |\beta|$ یا اگر $|\alpha| = |\beta|$ در اینصورت $\alpha >_{lex} \beta$

تعریف ٤.۲.۱. ترتیب الفبایی معکوس مدرج (graded reverse lex order)

گوییم که $\alpha >_{grevlex} \beta$ هرگاه $|\alpha| > |\beta|$ یا هرگاه $|\alpha| = |\beta|$ در اینصورت در بردار تفاضل

$\alpha\text{-}\beta \in Z^n$ راست ترین درایه غیر صفر منفی باشد.

تعریف ۵.۲.۱. فرض میکنیم $f = \sum a_\alpha x^\alpha$ یک چند جمله ای غیر صفر در $K[x]= k[x_1, \dots , x_n]$ و $>$

، یک ترتیب یک جمله ای روی یک جمله ای های $K[x]$ باشد. در اینصورت درجه ی کلی، درجه مرکب،

ضریب پیشرو، یک جمله ای پیشرو و جمله ی پیشروی f به صورت زیر تعریف می شوند

Total degree = f درجه کلی $= \deg(f) := \mathrm{Max}\,\{|\alpha| \mid a_\alpha \neq 0\}$

Multi degree = f درجه مرکب $= \mathrm{mdeg}(f) := \mathrm{Max}\{ \alpha \mid a_\alpha \neq 0\}$

Leading degree = f ضریب پیشرو $= \mathrm{LC}(f) := a_{\mathrm{mdeg} f} \in K$

Leading monomial = f یکجمله ای پیشرو $= \mathrm{LM}(f) := X^{\mathrm{mdeg} f}$

Leading term = f جمله پیشرو $= \mathrm{LT}(f) := \mathrm{LC}(f)\,\mathrm{LM}(f)$

تعریف ۶.۲.۱. یک ترتیب یکجمله ای را روی N^n ثابت میگیریم. یک چند جمله ای $r \in k[x_1, \dots, x_n]$ را نسبت به یک مجموعه از چند جمله ای های غیر صفر $F=\{f_1,\dots f_s\}$ تحویل یافته گوییم هرگاه $r=0$ یا r یک ترکیب k- خطی از یک جمله ای هایی باشد که هیچ یک از آنها بر هیچ یک از $LT(f_1),\dots, LT(f_s)$ بخشپذیر نباشد.

تعریف ۷.۲.۱. یک چند جمله ای $f \in k[x_1, \dots, x_n]$ را روی میدان K تحویل ناپذیر گوییم هرگاه f ثابت نباشد و $h \epsilon K$ یا

$$\forall\ g, h \in K[x_1, \dots, x_n] \quad f = gh \rightarrow\ g \in K$$

قضیه ۱.۲.۱. یک ترتیب یکجمله ای را روی N^n ثابت میگیریم. فرض کنیم $f = (f_1, \dots, f_s)$ یک s- تایی مرتب از چند جمله ای های ناصفر در $K[x]$ باشد در اینصورت هر $f \epsilon K[X]$ را میتوان به صورت $f = a_1 f_1 + \dots + a_s f_s + r$ نوشت که در آن $r \epsilon K[x]$ ، a_i و r نسبت به $\{f_1, \dots, f_s\}$ تحویل یافته است.

r را باقی مانده تقسیم f بر F می نامند و به علاوه اگر $a_i f_i \neq 0$ ، در اینصورت

$$mdeg\ (f) \geq mdeg(a_i\ f_i)$$

تعریف ۸.۲.۱ . فرض کنیم $<$ یک ترتیب یکجمله ای ثابت باشد. مجموعه متناهی $G=\{g_1,\dots,g_t\}$ از ایده آل I را یک پایه گروبنر I نسبت به $<$ گوییم هرگاه $<LT(I)> = <LT(g_1), \dots, LT(g_t)>$

تعریف ۹.۲.۱. فرض کنیم f , g چند جمله ای های ناصفردر $K[x]$ با $LM(f)=x^{\alpha}$ و $LM(g)=x^{\beta}$ باشند.

فرض کنیم x^{γ} کوچکترین مضرب مشترک x^{α} و x^{β} باشد، یعنی به ازای هر هر $i = 1, \dots, n$،

$\gamma_i =: max\{\alpha_i,\beta_i\}$ در اینصورت چندجمله ای

$$S(f,g):= \frac{x^{\gamma}}{LT(f)} f - \frac{x^{\gamma}}{LT(g)} g$$

را $S-$چند جمله ای f و g می نامیم.

قضیه ۲.۲.۱. فرض I یک ایده آل در $K[x]$ باشد در اینصورت یک مجموعه از مولد $G = \{g_1,\dots,g_t\}$ از ایده-آل I یک پایه گروبنر برای I است اگر و فقط اگر برای هر زوج i و j با $i\neq j$، باقی مانده $S(g_i,g_j)$ بر G (مرتب شده نسبت به یک ترتیب) صفر باشد.

قضیه ۳.۲.۱. برای یک ایده آل مفروض ناصفر I از حلقه $K[x]$ می توانیم یک پایه گروبنر بیابیم. فرض کنیم $I=<f_1,\dots, f_j>$ یک ایده آل ناصفر باشد در اینصورت الگوریتم زیر در تعدادی متناهی مرحله یک پایه گروبنر برای ایده آل I محاسبه می کند.

ورودی (input): $F= (f_1,\dots, f_s)$

خروجی (output): یک پایه گروبنر $G=\{g_1,\dots,g_t\}$ برای $f \in G$

مقدار دهی اولیه (initialization) :

G:= F

$\mathcal{G}$:={(f_i , f_j)$|f_i$,$f_j \epsilon G, f_i \neq f_j$}

h:= 0

WHILE $\mathcal{G}\neq 0$ DO

Choose any {f,g} $\epsilon\mathcal{G}$

$\mathcal{G}$:=$\mathcal{G}$ \{ { f,g}}

h:= $\overline{S(f,g)}^G$

IF h≠0 THEN

$\mathcal{G}$:= $\mathcal{G} \cup$ { $\{u, h\}|u\epsilon\ G\}$

G:= G ∪ {h}

تعریف ۱۰.۲.۱ فرض کنیم $G=\{g_1,\ldots,g_t\}$ یک پایه گروبنر ایده آل $I\epsilon K[x]$ باشد. G را پایه گروبنر تحویل یافته گوییم هرگاه

i. $\forall i \quad LC(g_i) = 1$

ii. هر g_iنسبت به $G\setminus\{g_i\}$ تحویل یافته باشد.

تعریف ۱ .۲ .۱۱. برای یک ایده آل مفروض $I=<f_1,...,f_s>$ و $f \ \epsilon K[x]$ به طریق زیر میتوانیم تشخیص دهیم که آیا $f \in I$ یا خیر

i. ابتدا یک پایه گروبنر G را توسط الگوریتم بوخبرگر برای ایده آل I محاسبه می کنیم.

ii. این حقیقت را به کار می بریم که $f \in I \leftrightarrow \bar{f}^G = 0$

قضیه ۱ .۲ .٤.فرض کنیم K یک میدان دلخواه باشد و$I = < f_1,\dots,f_s>$

$$f \in \sqrt{I} \quad \leftrightarrow \quad 1 \in \tilde{I} := < f_1 , \dots , f_s, 1\text{-}yf > \subseteq K[x_1,\dots,x_n,y]$$

$$\leftrightarrow \tilde{I} = k[x_1, \dots , x_n, y]$$

تعریف ۱ .۲ .۱۲. برای یک میدان K و یک عدد صحیح مثبت n مجموعه ی

$k^n = \{(a_1,...,a_n)|a_1,..., a_n \in K\}$ را فضای آفین n– بعدی می گوییم.

تعریف ۱ .۲ .۱٦. یک مجموعه $X \subseteq K^n$ را واریه آفین نامیم هرگاه : $X = V(S) \qquad \exists S \subseteq K[X]$

قضیه ۱ .۲ .۵. فرض می کنیم $I \trianglerighteq k[x_1, \dots , x_n]$ و G یک پایه گروبنر از I نسبت به ترتیب الفبایی با

$x_1 > x_2 > \cdots > x_n$ باشد، در اینصورت برای هر $0 \leq l \leq n$، مجموعه ی

$G_l := G \cap k[x_1, \dots , x_n]$ یک پایه گروبنر برای ایده آل حذفی l ام I_l است.

فصل دوم

نصب و راه اندازی

روش های اجرای Sage

i. نصب آن به صورت نرم افزار

ii. اجرای مستقیم آن در سایت www.sagemath.org البته برخی امکانات را نخواهیم داشت.

iii. استفاده در محیط پایتون

که در اینجا به بررسی مورد اول می پردازیم.

۱.۲. روش نصب Sage در Windowse

ابتدا نرم افزار Sage را از آدرس زیر دانلود می کنیم.

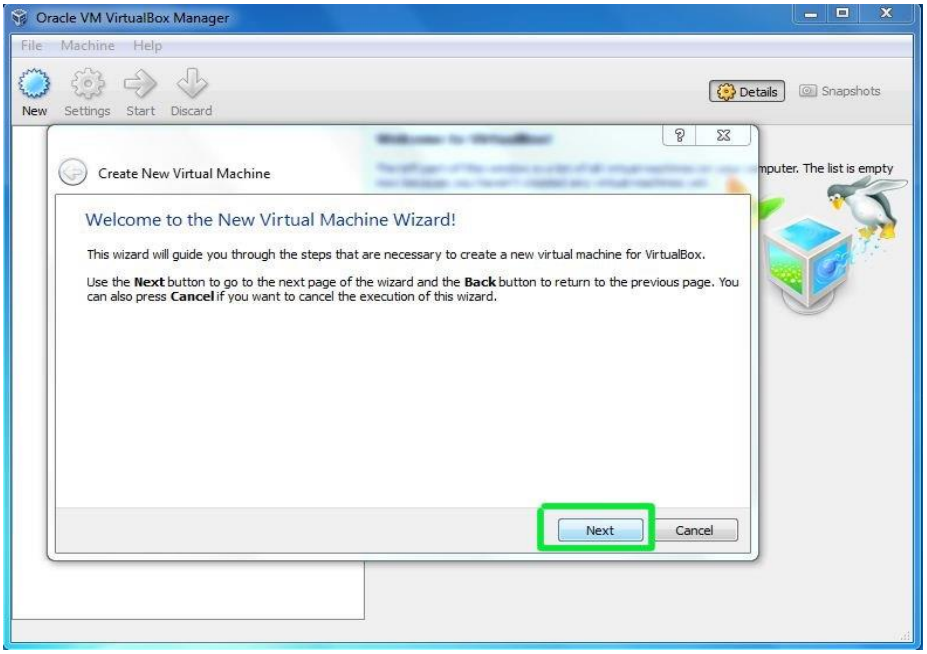
http://www.sagemath.org/download-source.html

برای نصب سیج روی ویندوز احتیاج به نصب نرم افزار $Virtualbox$ داریم. $Virtualbox$ به شما این امکان را می دهد تا یک سیستم عامل را در یک سیستم عامل دیگر اجرا کنید. برای نصب این نرم افزار مراحل زیر را انجام می دهیم.

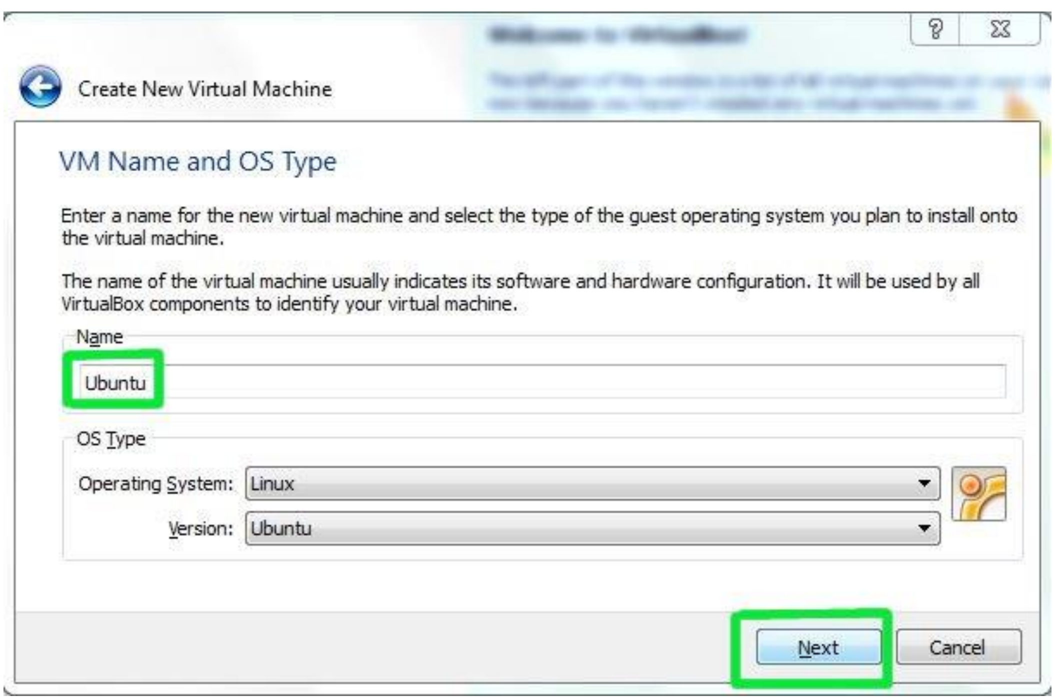
❖ ابتدا نرم افزار $Virtualbox$ را از سایت $www.oracle.com$ دانلود کنید.

این نرم افزار دانلود شده را مانند تمامی نرم افزارهای دیگر $install$ کنید تا آیکونی با نام $Virtualbox$ بروی دسکتاپ شما ظاهر شود. سپس مانند تصاویر زیر عملیات نصب را انجام دهید.

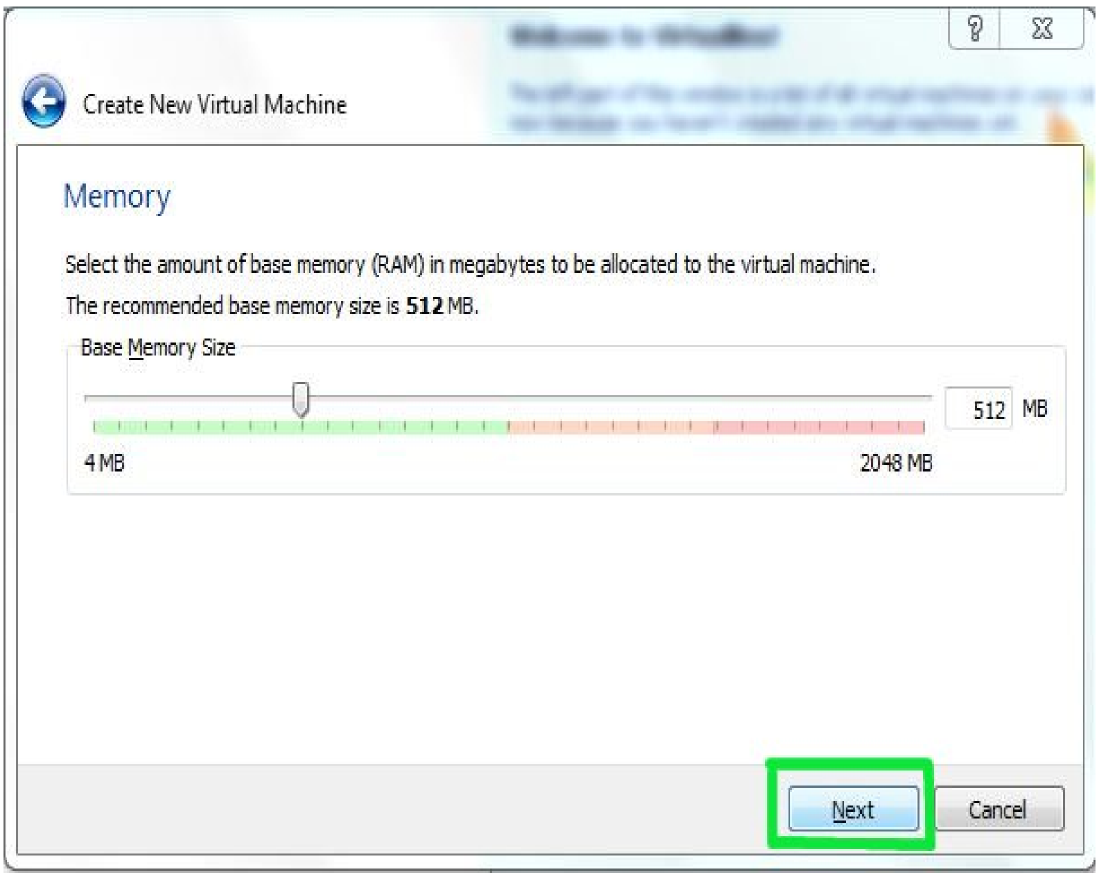
❖ بروی آیکون $Virtualbox$ کلیک کرده تا صفحه ای مطابق شکل زیر باز شود. روی آیکون New کلیک کنید.



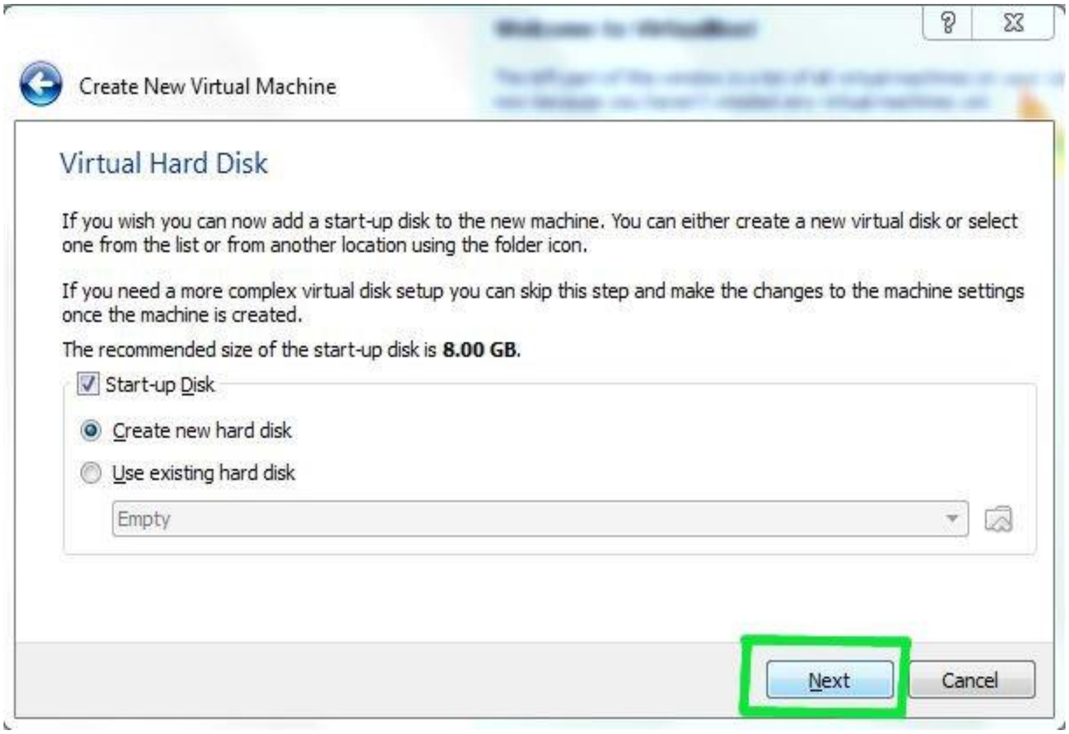
❖ شما میتوانید این نرم افزار را بطور دلخواه نام گذاری کنید. از آنجایی که سیتم عامل Ubuntu را نصب میکنید همین نام را بروی نرم افزار نام گذاری میکنیم و مطابق شکل عمل می کنیم.



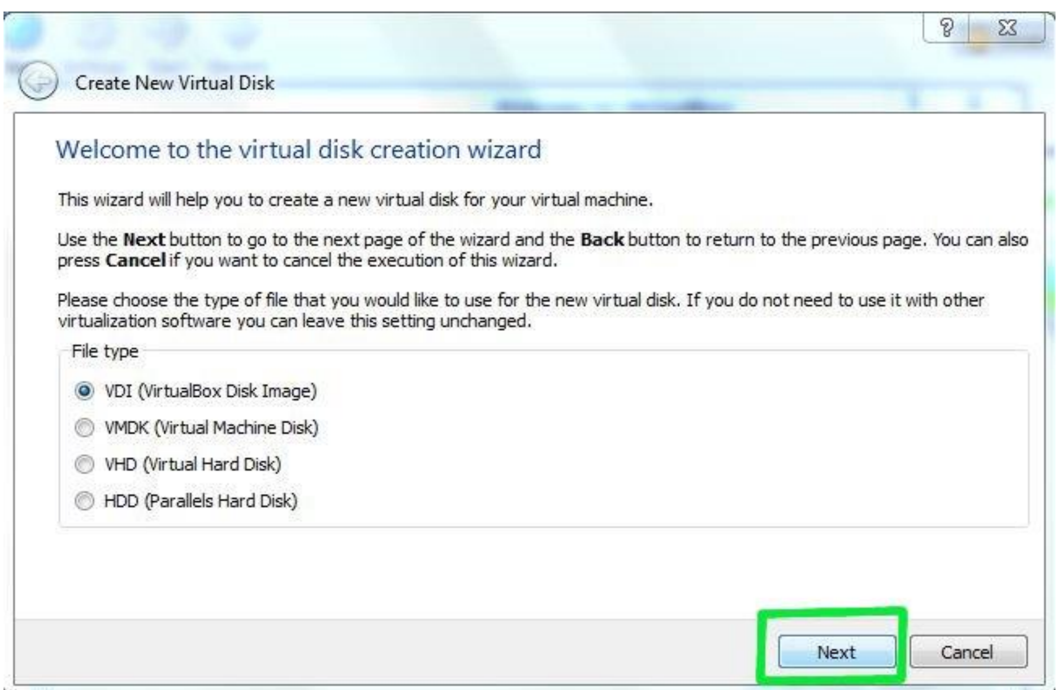
❖ اگر RAM شما ۴ گیگابایت باشد این نرم افزار $1GB$ را به خود اختصاص میدهد. اگر RAM شما $2GB$ باشد، آنگاه MB ۵۱۲ برای اختصاص دادن به این نرم افزار خوب است. اگر شما هیچ ایده ای راجب RAM دستگاه مورد استفاده ی خود ندارید مطابق تصویر زیر عملیات نصب را انجام دهید.



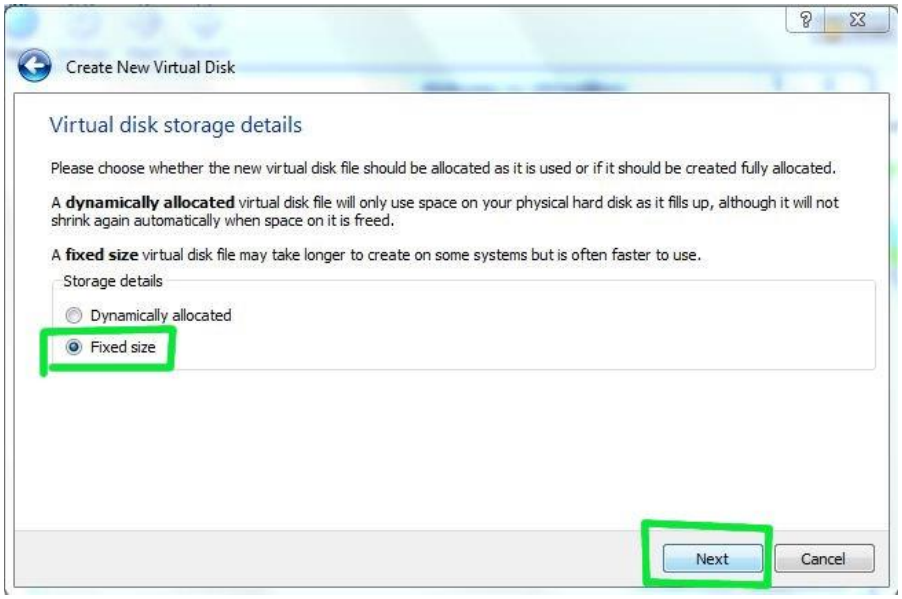
❖ اگر برای بار اول از نرم افزار $Virtualbox$ استفاده می کنید باید یک هارد دیسک جدید مطابق شکل زیر ایجاد کنید.



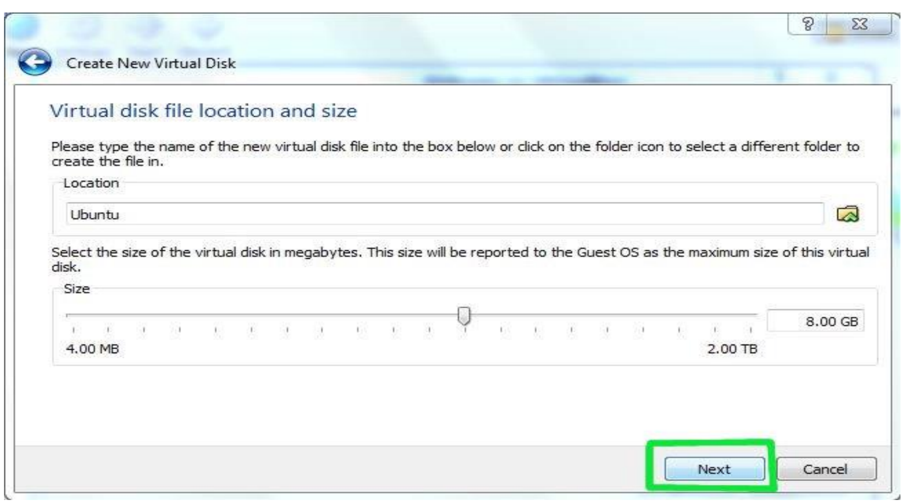
❖ مطابق شکل زیر دکمه ی $next$ را بزنید.

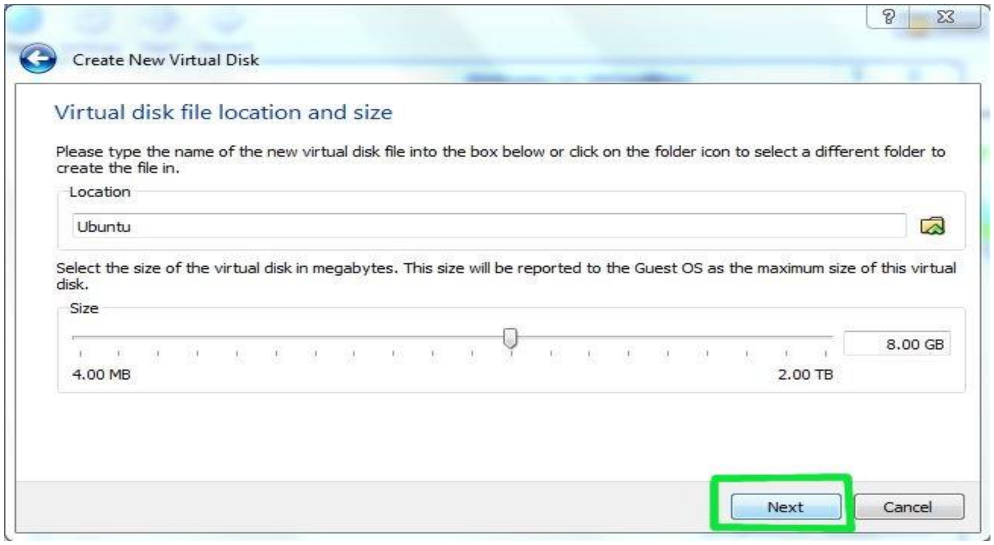


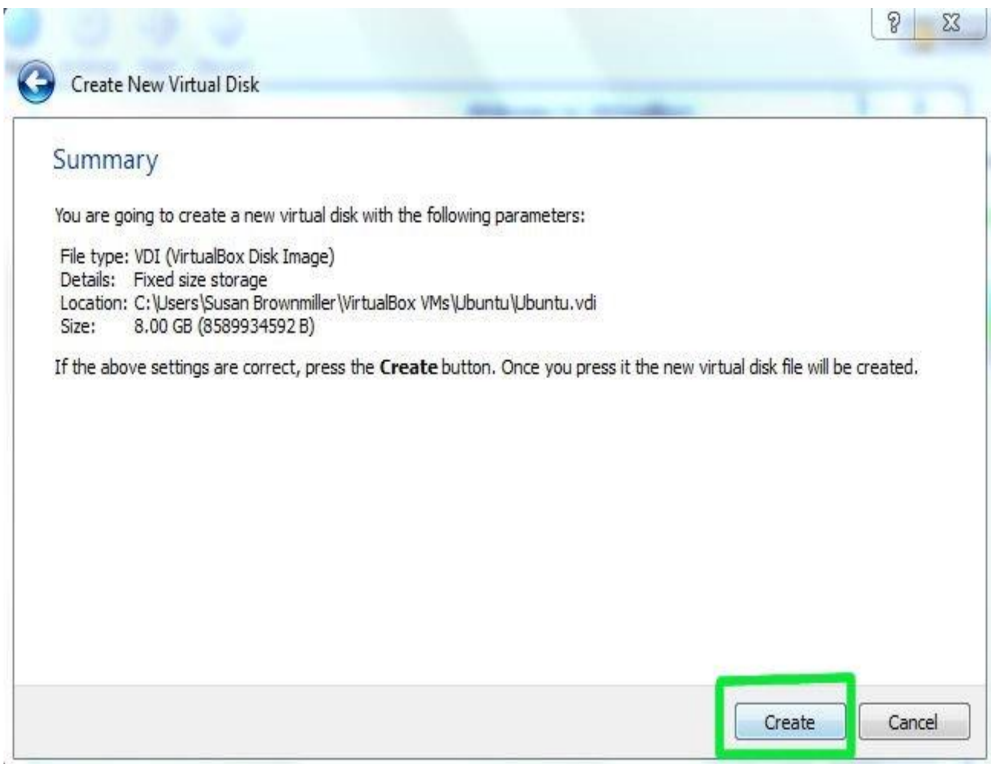
❖ گزینه ی $fixed\ size$ را انتخاب کرده وگرینه ی $next$ را میزنید.

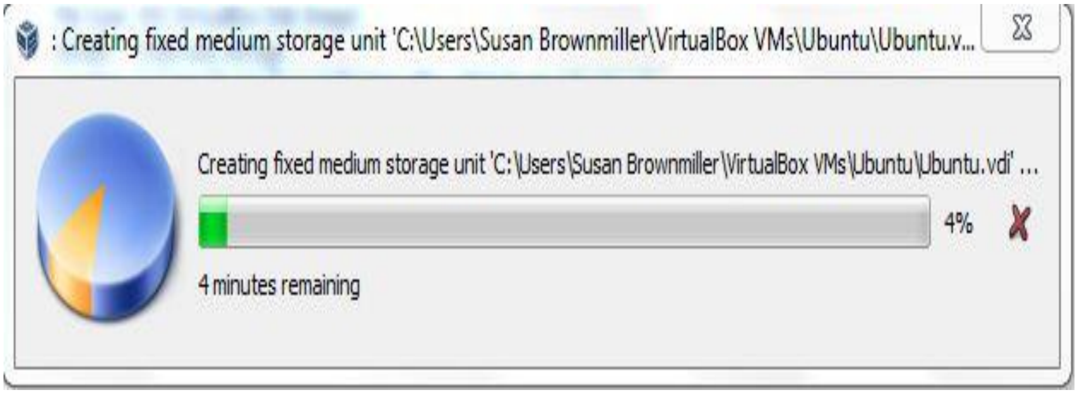


❖ عملیات نصب را مطابق شکل زیر انجام دهید...

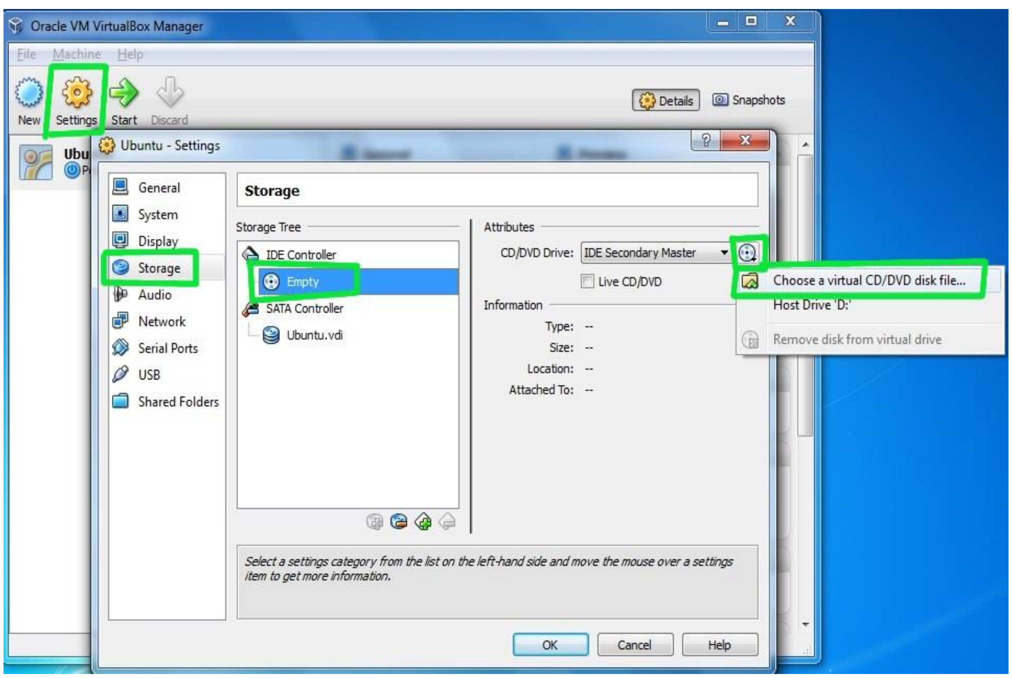



Create New Virtual Disk
Virtual disk file location and size
Please type the name of the new virtual disk file into the box below or click on the folder icon to select a different folder to create the file in.
Location
Ubuntu
Select the size of the virtual disk in megabytes. This size will be reported to the Guest OS as the maximum size of this virtual disk.
Size
8.00 GB
4.00 MB
2.00 TB
Next
Cancel

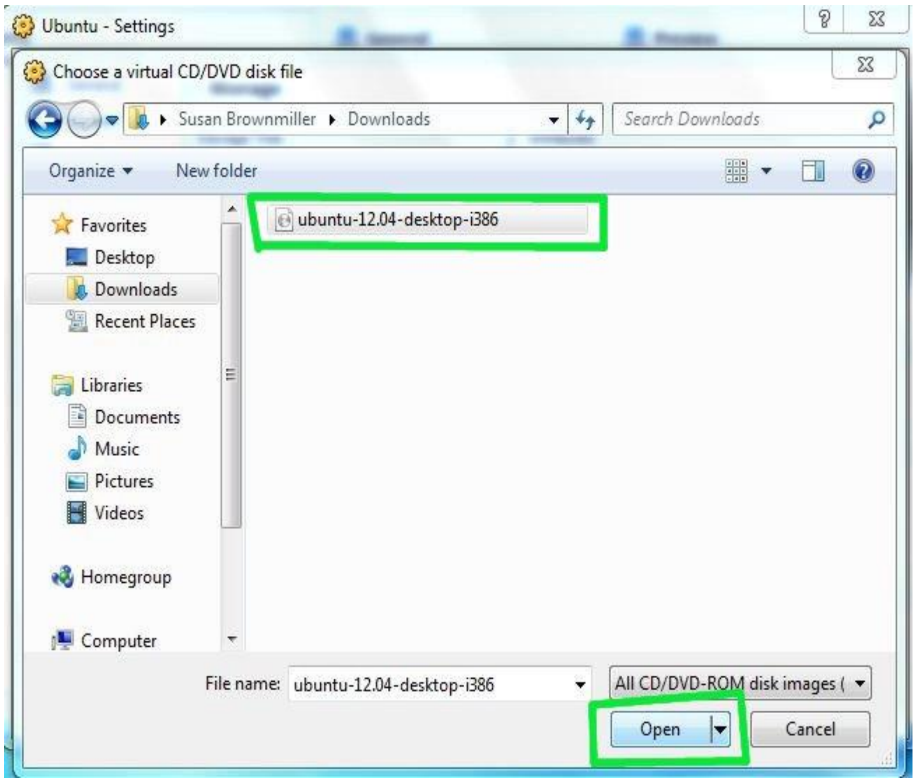

Create New Virtual Disk
Summary
You are going to create a new virtual disk with the following parameters:
File type: VDI (VirtualBox Disk Image)
Details: Fixed size storage
Location: C:\Users\Susan Brownmiller\VirtualBox VMs\Ubuntu\Ubuntu.vdi
Size: 8.00 GB (8589934592 B)
If the above settings are correct, press the Create button. Once you press it the new virtual disk file will be created.
Create
Cancel

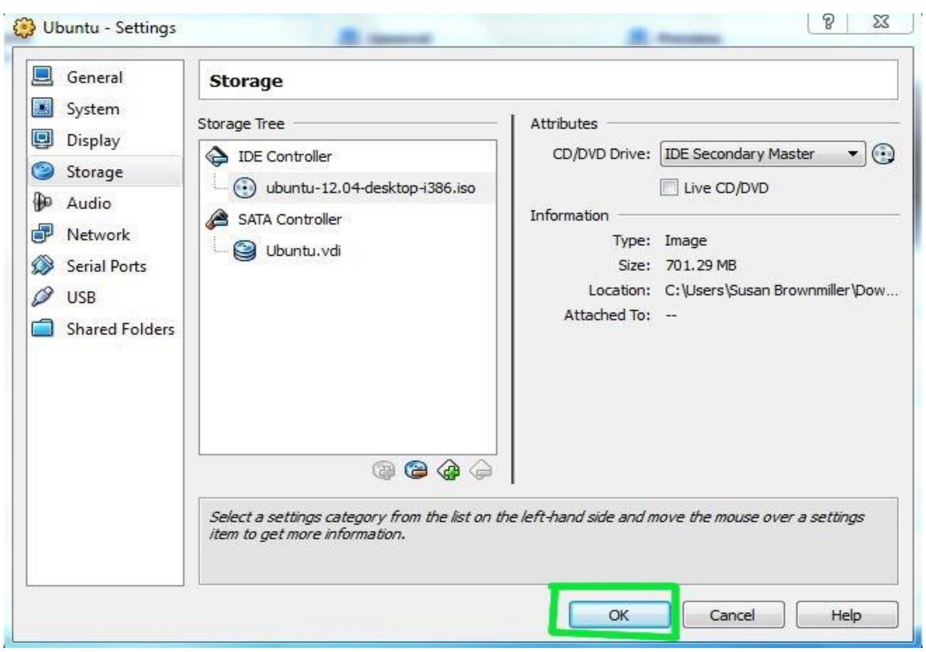


❖ پس از اینکه این مراحل را به اتمام رسید در کنار گزینه ی new گزینه ی setting را انتخاب کرده و مراحل زیر را انجام دهید.



❖ از قسمتی که نرم افزار vitrualbox را روی دستگاه خود نصب کردید تصویر یک فولدرکوچکی را می بینید، روی آن کلیک کرده و گزینه ی open را انتخاب کنید.

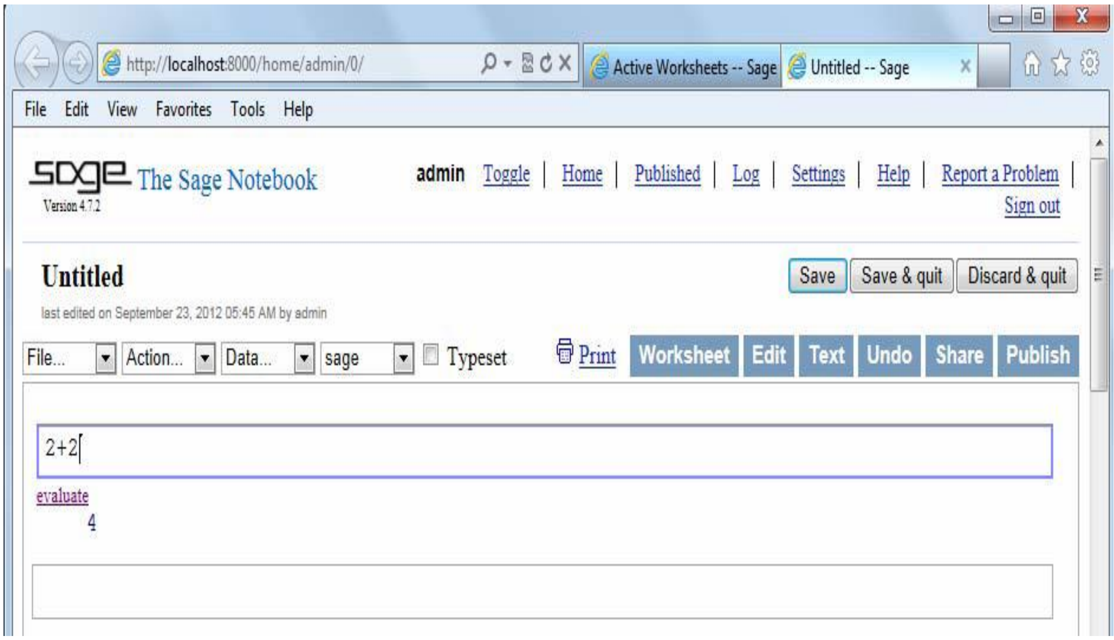




❖ گزینه ی Ok را بزنید.

حال نرم افزار vitrualbox بروی دستگاه شما نصب شده و آیکون نرم افزار Sage شما که قبل از نصب vitrualbox به رنگ سفید بود به شکل مکعبی نارنجی در میاید. سپس بروی آن کلیک کنید و پس از چند دقیقه نرم افزار Sage از طریق vitrualbox اجرا خواهد شد.

پس از اینکه تمامی مراحل نصب به درستی انجام شد پس از اجرای نرم افزار صفحه ای مطابق شکل قابل مشاهده است که برای نوشتن برنامه با کلیک کردن روی گزینه New worksheet صفحه ی جدیدی باز خواهد شد که شما پس از نام گذاری آن قادر به نوشتن برنامه در این نرم افزار خواهید بود.

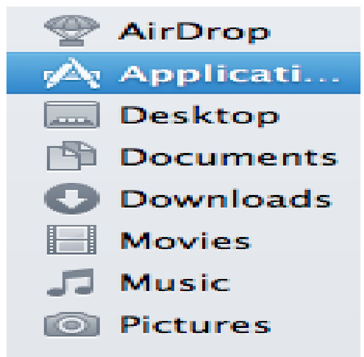


۲.۲. طریقه نصب Sage در سیستم عامل Mac

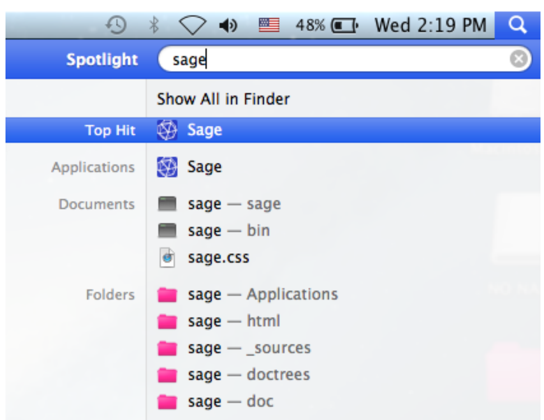
❖ ابتدا نرم افزار Sage را از سایت www.sagemath.org دانلود کنید. فایلی مطابق شکل زیر خواهیم داشت.

❖ بروی این فایل کلیک کرده تا پوشه ای مطابق شکل زیر بدست آید.

❖ این پوشه را داخل Application می اندازیم.



❖ در قسمت $search$ دسکتاپ نام $sage$ را تایپ می کنیم.



بروی آیکون $Sage$ کلیک کرده تا نرم افزار اجرا شود. در صورت اجرا نشدن نرم افزار گزینه ی Sage-Sage را انتخاب کرده و در ترمینال باز شده دستور "$notebook()$"را تایپ می کنیم. نرم افزار اجرا خواهد شد. آیکون آبی رنگ ظاهر شده در این قسمت نرم افزار $sage$ می باشد.

۱ .۳. طریقه نصب Sage در سیستم عامل لینوکس (Linux)

- ابتدا نرم افزار را از سایت زیر دانلود می کنیم.

http://www.Sagemath.org/download-source.html

آخرین نسخه را برای لینوکس دانلود کنید این فایل حدوداً ۴۰۰ مگابایت حجم دارد.

- فایل دانلود شده را extract کنید.
- فایل Sage.sh را اجرا کنید.

بایستی بسته gfortran نیز نصب شود. از دستور sudo apt-get install build-essential gfortran استفاده کنید.

فصل سوم

معرفی نماد ها

در نرم افزار Sage از نماد های منطقی == ، $\leq$، $\geq$ ، $>$ و $<$ استفاده میشود.

به عنوان مثال

Sage: a

5

Sage: 2==2

$True$

Sage: 2=3

$False$

Sage: 2<3

$True$

نماد های ریاضی به صورت زیر تعریف میشوند.

a^b = a**b or a^b

a mod b = a%b

a÷b = a/ b

از // برای نشان دادن خارج قسمت یک تقسیم استفاده می شود.

مثال.

Sage: 4*(10//4) +10%4= =10

True

برای به دست آوردن نوع داده ی وارد شده از دستور type( ) استفاده می کنیم.

Sage: a = 5   # a is an integer

Sage: type(a)

< type ’Sage. rings. integer. Integer’ >

Sage: a = 5/3   # now a is a rational number

Sage: type(a)

< type ’Sage. rings. rational. Rational’ >

Sage: a = hello

Sage: type(a)

<’type str’>

جهت استفاده از help نرم افزار از دستور ?command استفاده میکنیم. به عنوان مثال برای یافتن اطلاعات درمورد جدول Sudoku به این ترتیب عمل می کنیم.

Sudoku?

Definition: sudoku(m)

Docstring:

Solves Sudoku puzzles described by matrices.

INPUT:

m - a square Sage matrix over Z, where zeros are blank entries

OUTPUT:

A Sage matrix over Z containing the first solution found, otherwise None.

EXAMPLE:

An example that was used in previous doctests.

Sage: A = matrix(ZZ,9,[5,0,0, 0,8,0, 0,4,9, 0,0,0, 5,0,0, 0,3,0, 0,6,7, 3,0,0, 0,0,1, 1,5,0, 0,0,0, 0,0,0, 0,0,0, 2,0,8, 0,0,0, 0,0,0, 0,0,0, 0,1,8, 7,0,0, 0,0,4, 1,5,0, 0,3,0, 0,0,2, 0,0,0, 4,9,0, 0,5,0, 0,0,3])

Sage: A

[5 0 0 0 8 0 0 4 9]

[0 0 0 5 0 0 0 3 0]

[0 6 7 3 0 0 0 0 1]

[1 5 0 0 0 0 0 0 0]

[0 0 0 2 0 8 0 0 0]

[0 0 0 0 0 0 0 1 8]

[7 0 0 0 0 4 1 5 0]

[0 3 0 0 0 2 0 0 0]

[4 9 0 0 5 0 0 0 3]

Sage: sudoku(A)

[5 1 3 6 8 7 2 4 9]

[8 4 9 5 2 1 6 3 7]

[2 6 7 3 4 9 5 8 1]

[1 5 8 4 6 3 9 7 2]

[9 7 4 2 1 8 3 6 5]

[3 2 6 7 9 5 4 1 8]

[7 8 2 9 3 4 1 5 6]

[6 3 5 1 7 2 8 9 4]

[3 2 6 7 5 8 1 9 4]

لیست ها

گاهی نماد ها به صورت لیست می باشند در اینصورت برای معرفی لیست ها به صورت زیر عمل می کنیم.

Sage: a = [1, 7, 2]; b = [4, 5]

Sage: c = a + b; c

[1, 7, 2, 4, 5]

Sage: c.sort( ); c

[1, 2, 4, 5, 7]

Sage: c.<tab> با فشار دادن کلید Tab می توان عملیات های گوناگونی را محاسبه کرد.

c.append c.extend c.insert c.remove c.sort

c.count c.index c.pop c.reverse

مثلا

عنصری را به لیست اضافه می کنیم

Sage: c.append ("foo"); c

[1, 2, 4, 5, 7, 'foo']

Sage: c; c[0]

['foo', 7, 5, 4, 2, 1]; 'foo'

عناصر یک لیست را می توان نامگذاری کرد.

Sage: c[0] = 11

Sage: c

[11, 7, 5, 4, 2, 1]

می توان عناصر دلخواه یک لیست را معرفی نمود.

Sage: c[0:2]

[11, 7]

می توان یک لیست دلخواه را ساخت.

Sage: [n^2 for n in range(2,10)]

[4, 9, 16, 25, 36, 49, 64, 81]

تعریف تابع در Sage

هر چند که در نرم افزار Sage تمامی توابع گنجانده شده اند اما Sage به ما این امکان را می دهد که توابعی را در صورت نیاز تعریف کنیم.

برای این منظور دستور def را نوشته و در آخر دستور از علامت " : " استفاده می کنیم.

مثال. تابعی بنویسید که عددی را دریافت کرده و زوج یا فرد بودن آن را مشخص کند.

```
Sage: def is_even(n):
... return n%2 == 0
...
Sage: is_even(2)
True
Sage: is_even(3)
False
```

مثال.

```
Sage: def  is_divisible_by(number, divisor=2):
... return number%divisor == 0
Sage: is_divisible_by(6,2)
True
Sage: is_divisible_by(6)
True
```

```
Sage: is_divisible_by(6, 5)
False
```

فصل چهارم

کاربرد Sage در حساب دیفرانسیل

و

رسم نمودار

۱.٤ . اعمال مختلف ریاضی

$\sqrt{x}$= sqrt(x)

$\sqrt[n]{x}$=$x^n(\frac{1}{n})$

$|x|$=abs(x)

Lag_b(x)=Log (x,b)

$\sum_{i=k}^{n} f(i)$ = sum (f(i) for i in (k..n))

$\lim_{x \to a} f(x) = limit(f(x), x = a)$

$\frac{d}{dx}$ (f(x))=diff (f(x) , x)

$\frac{\partial}{\partial x}$ (f(x,y))=diff (f(x,y),x)

diff=differentiate

$\int f(x)dx = integral(f(x), x)$

$\int_{a}^{b}$ f(x)dx=integral(f(x) , x , a , b)

Taylorpolynomial , deg n around (a) = taylor (f(x) , x , a , n)

$$\prod_{i=k}^{n} f(i) = prod\ (f(i) for\ i\ in\ (k..n)$$

$expand\_log$ باز کردن لگاریتم

$simplify\_log$ ساده سازی عبارات لگاریتم

$simplify\_rational$ ساده سازی عبارات کسر دار

$radical\_simplify$ ساده سازی رادیکال

$simplify\_exp$ ساده سازی عبارات توان دار

$simplify\_factorial$ ساده سازی عبارات فاکتوریل

$simplify\_full$ ساده سازی تمام عبارات بالا

۲.٤ . برخی نمادها

$\pi = pi$

E = e

$\infty = 00$

∅= golden_ratio

Integer Z = ZZ

Rational Q = QQ

Real R = RR

Complex C= CC

Finite field F_q = GF

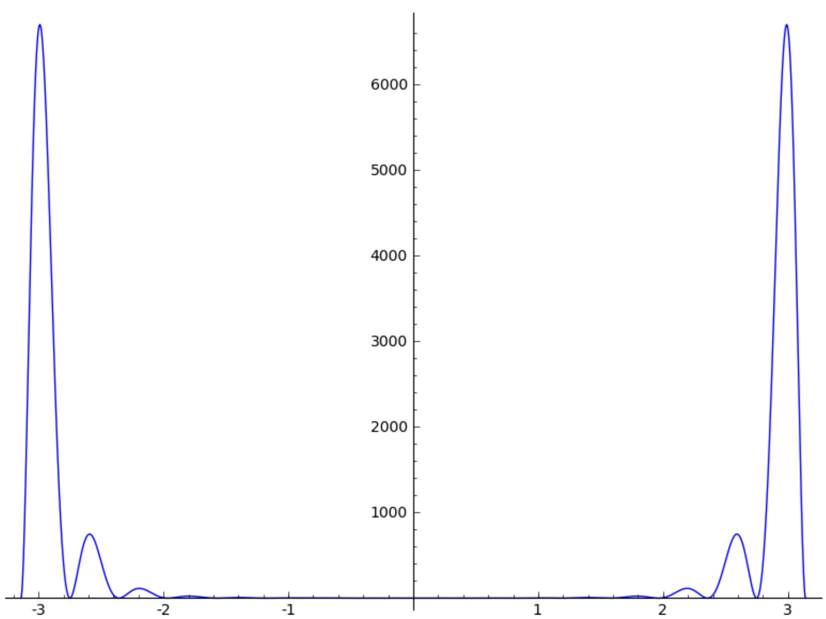
۳.٤ . رسم نمودار

برای رسم نمودار در ساده ترین حالت از دستور plot استفاده می کنیم.

به عنوان مثال

$f(x)=\sin(8x)^2 e^{x2}$

Sage: plot (sin(8*x)^2 * e^(x^2),x, -pi , pi)



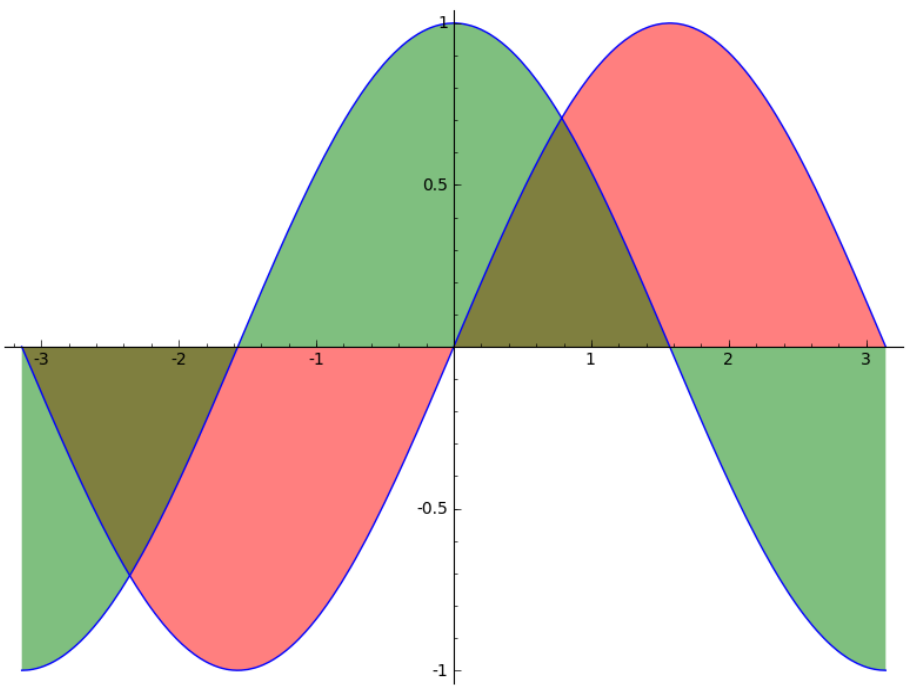
٤. ۳ . ۱. اختیارات رسم نمودار

داخل منحنی رنگ می شود.	Fill=true
انتخاب رنگ داخل منحنی	fillcolor= ' green '
رنگ خط	rgbcolor = ' color '
alpha میزان مرئی بودن پر کردن نواحی	(0=opaque, 1=transparent)
fillalpha میزان مرئی بودن پر کردن نواحی	(0=opaque, 1=transparent)
حداکثر عمق زمانیکه تابع به شدت تغییر می کند	adaptive_recursion
تغییراتی که باعث توقف بازگشت می شود	adaptive_toleranc
تشخیص جاهایی که تابع بی نهایت می شود	detect_poles
لیست نقاطی که از نمودار جا افتاده اند	exclude
تعداد نقاطی که در نمودار به کار گرفته شده اند	plot_points

میتوان چند نمودار را در یک شکل کشید.

به عنوان مثال

```
f(x)= sin(x)

g(x)=cos(x)

P1=plot(sin(x),x,-pi,pi,fill=true,fillcolor='red')

P2=plot(cos(x),x,-pi,pi,fill=true, fillcolor= 'green')

plt=(P1+P2)

show(plt)
```

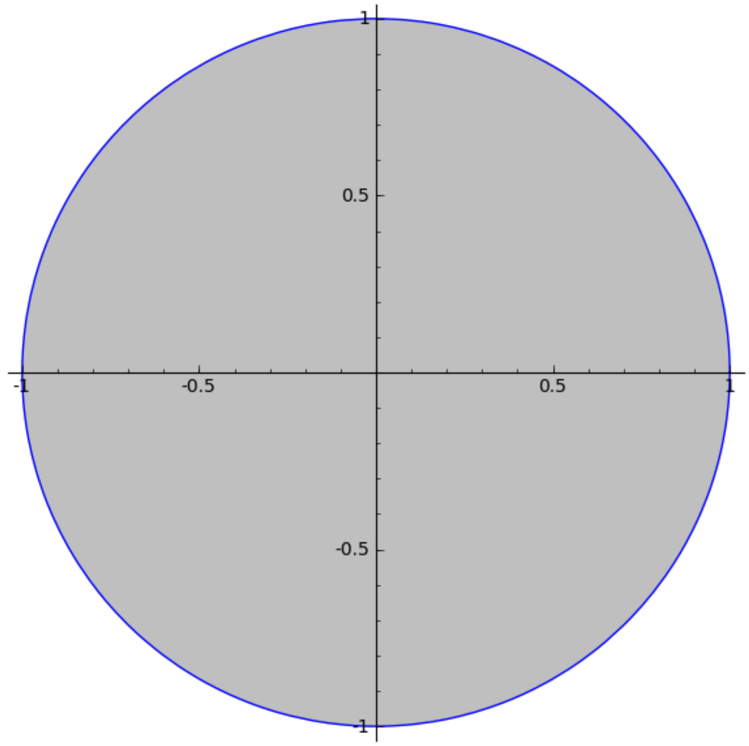


٤.٤ . رسم توابع پارامتری

برای رسم اینگونه توابع از دستور parametric_plot استفاده می کنیم.

دایره (cos(t),sin(t),t) را به صورت پارامتری رسم میکنیم. ابتدا باید متغیر t را معرفی کنیم.

```
t= var('t')

p=parametric_plot3((cos(t),sin(t)),(t,0,2*pi),fill=true)

show(p)
```

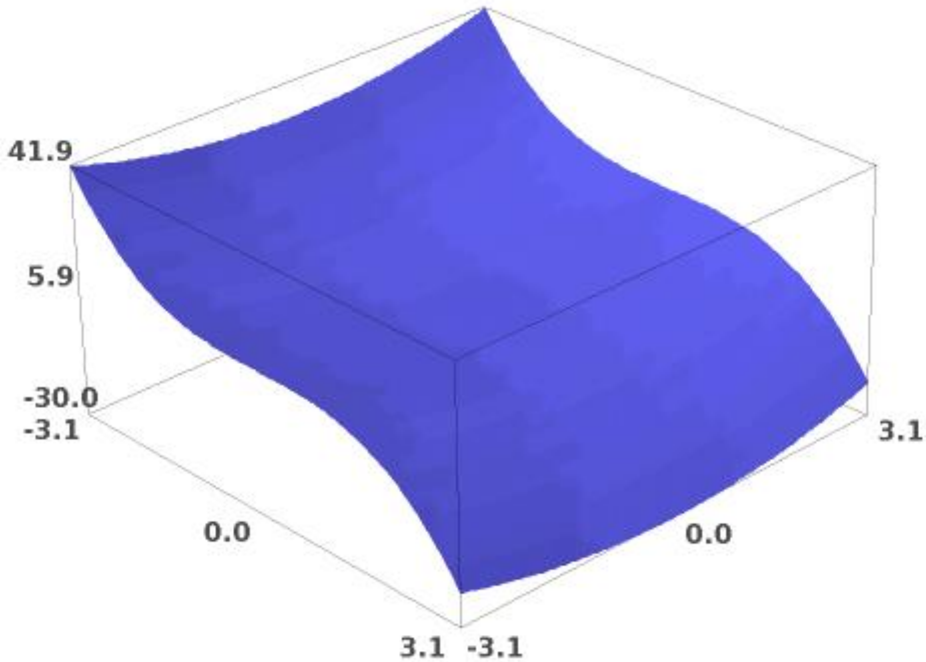


٥ . ٤ . رسم نمودار سه بعدی

برای رسم نمودار های سه بعدی پس از معرفی متغیر ها از دستور plot3d استفاده می کنیم.

تابع y^2+1-x^3-x را رسم می کنیم.

```
x , y = var('x,y')

p=plot3d(y^2+1-x^3-x,(x,-pi,pi),(y,pi,pi))

p.show()     or   show(p)
```



میتوانیم به صورت دلخواه محورها را هم رسم کنیم. کافیست پس از تعیین بازه از عبارت axes=true استفاده کرد.

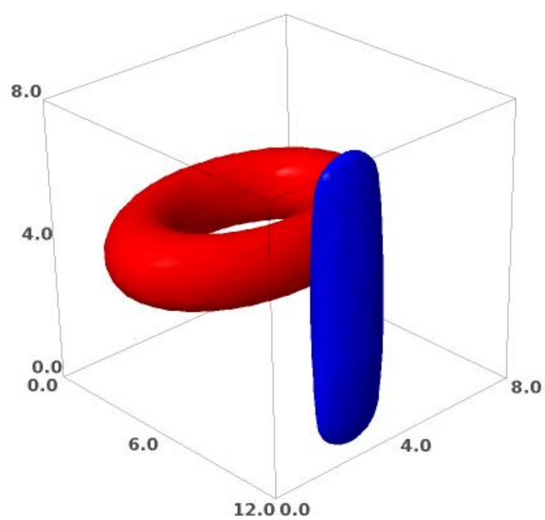
به علاوه میتوان خطوط اطراف منحنی را نیز حذف کرد. برای این منظور باید عبارت frame=false را نوشت.

٦.٤ . توابع سه بعدی پارامتری

دستوری که برای توابع سه بعدی پارامتری استفاده میشود به صورت parametric_plot3d می باشد.

مثال.

```
x,y = var('u,v')

f1=(4+(3+cos(v))*sin(u),4+(3+cos(v))*cos(u),4+sin(v))

f2=(8+(3+cos(v)*cos(u),3+sin(v)),4+(3+cos(v))*sin(u))

p1=parametric_plot3d(f1,(u,0,2*pi),(v,0,2*pi),texture='red')

p2=parametric_plot3d(f2, (u,0,2*pi),(v,0,2*pi), texture='blue')

combination= p1+p2

combination.show()
```



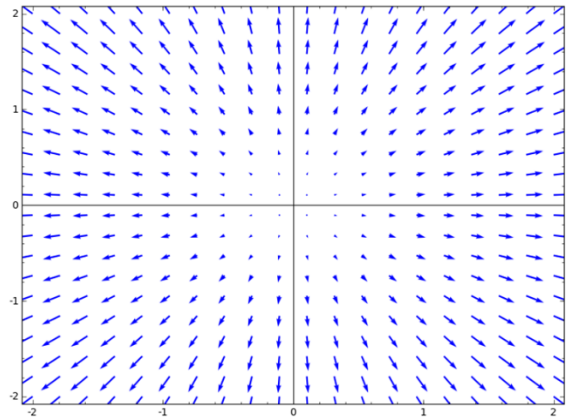
۴ . ۷ . میدان برداری

میدان های برداری را میتوان با استفاده از دستور plot_vector_field رسم کرد.

مثال.

$x, y = \ var('x, y')$

$$A = plot\_vector\_field((x, y), (x, -2,2), (y, -2,2), color = 'blue')$$



٤ . ٨ . برخی از توابع رسم نمودار

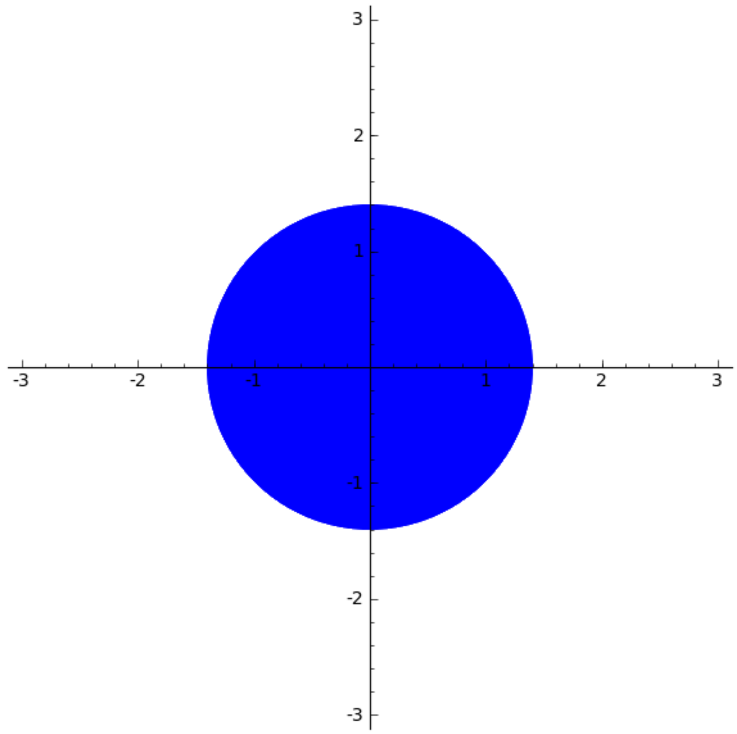
❖ ( )Implicit_plot : یک تابع دو متغیره را میگیرد و منحنی $f(x, y) = 0$ را رسم می کند.

مثال.

$x, y = var('x, y')$

$f(x, y) = x^{\wedge} + y^{\wedge}2 - 2$

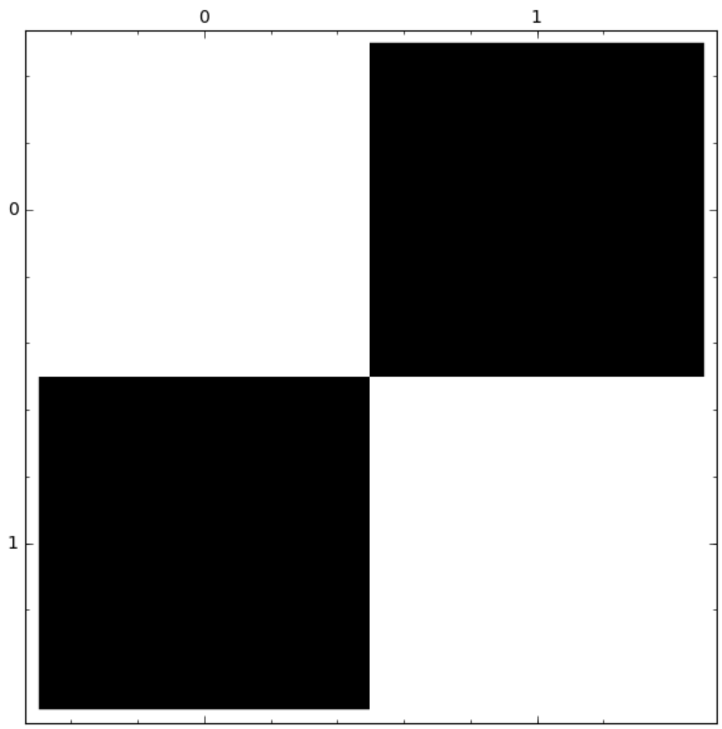
$implict - plot(f, (-3,3), (-3,3), fill = Truel)$



❖ $metrix - plot(\ )$: نقاط را می گیرد و برای یک سری بردار به صورت پیکسلی ماتریس را نمایش می دهد.

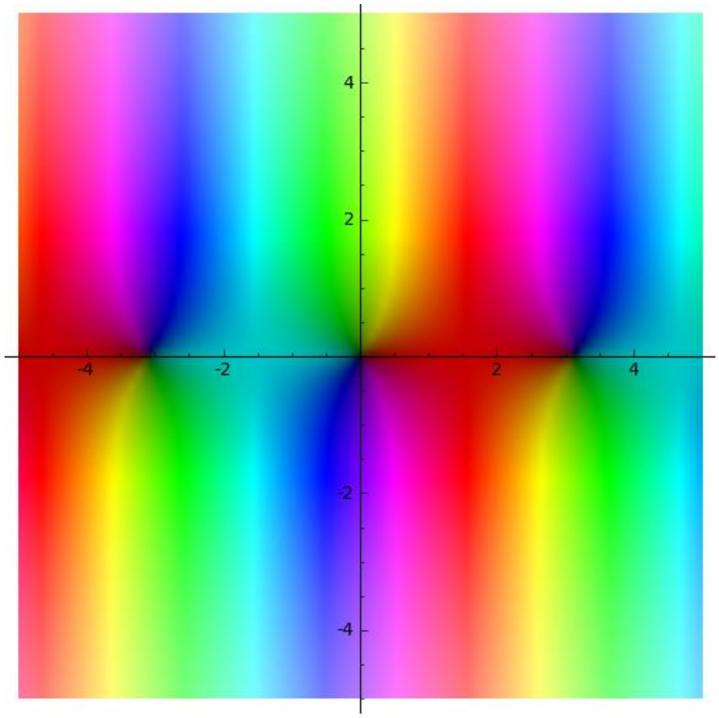
$Sage: x, y = var('x, y')$

$Sage: matrix - plot([1,0], [0,1]), fontsize = 10)$



❖ $complex - plot(\,)$: رسم نمودار یک تابع یک متغیره با ورودی اعداد مختلف f(z)

$Sage: complex - plot(f(x), (-5,5), (-5,5))$



- $circle(\ )$ رسم دایره با شعاع دلخواه.
- $ellipse(\ )$ رسم بیضی با شعاع و زاویه دلخواه.
- $arc(\ )$ یک کمان از یک دایره یا بیضی
- $Line(\ )$ یک خط با نقاط مشخص شده
- $polygon(\ )$ رسم یک چند ضلعی

فصل پنجم

میدان ها و حلقه ها

۵ . ۱ . معرفی میدان ها

در نرم افزار Sage انواع میدان ها به راحتی قابل تعریف می باشند.

در این قسمت به معرفی برخی میدان های مورد نیاز برای تعریف حلقه ها می پردازیم.

۵ . ۱ . ۱ . میدان اعداد گویا

این میدان را با نماد QQ و یا دستور ()RationalField نشان می دهیم.

مثال.

```
Sage: RationalField()

Sage:QQ

Sage:1/2 in QQ

True

Sage:sqrt(2) in QQ

False
```

۵ . ۱ . ۲ . میدان اعداد مختلط

برای نشان دادن میدان اعداد مختلط از نماد CC استفاده می کنیم.

مثال.

Sage: CC

Sage:CC.0   # 0^{th} generator of CC #

10000000000 ∗ I

Sage: a, b = 4/3, 2/3

Sage: z = a + b*i

Sage: z

1.33333333333333 + 0.666666666666667 ∗ I

Sage: z.imag()   # imaginary part

0.666666666666667

Sage: z.real() == a   # automatic coercion before comparison

True

Sage: a + b

2

۳.۱.۵ . میدان های متناهی

میدان های متناهی را به صورت GF(...) نشان می دهند.

Sage: GF(3)

Finite Field of size 3

Sage: GF(27, 'a') # need to name the generator if not a prime field

Finite Field in a of size 3^3

٤.١.٥ . میدان به طور جبری بسته

برای نشان دادن میدان به طورجبری بسته به صورت زیر عمل می کنیم.

```
Sage: QQbar
```

```
Sage:sqrt(3) in QQbar
```

True

۲.۵ . حلقه ها

۱.۲.۵ . حلقه چند جمله ای ها

جهت معرفی حلقه ی چند جمله ای ها، به طور کلی از دستور

R=PolynomialRing(Field, number of variables, variables, order)

استفاده می شود و در مرحله بعد مولد ها را تعریف میکنیم.

اما دستورهای دیگر برای معرفی حلقه ی چند جمله ای به شرح زیر می باشد.

- R=PolynomialRing(QQ,3, 'x,y,z', 'lex')
- R=PolynomialRing(QQ, 't')
- R.<t>= PolynomialRing(QQ)
- R.<t>= QQ []
- R=PolynomialRing(RationalField(),'x')
- R=PolynomialRing(GF(97), 'x').gen()
- R=GF(5)['x,y,z']; x,y,z=R.gens()
- R.<x>=PolynomialRing(QQ)
- Realpoly.<z>=PolynomialRing(RR)
- Ratpoly.<t>=PolynomialRing(QQ)

۱.۱.۲.۵ . معرفی مولد ها

معرفی مولد های یک حلقه به صورت زیر می باشد.

- R=PolynomialRing(QQ,3, 'x,y,z', 'lex')
- x,y,z = R.gens()
- R=PolynomialRing(RationalField(),'x').gen()

اگر بیش از یک مولد داشته باشیم از عبارت gens() استفاده می کنیم.

۲.۲.۵. حلقه خارج قسمتی

برای تعریف یک حلقه خارج قسمتی از R.quo () استفاده می کنیم.

مثال.

```
Sage: R.<x> = PolynomialRing(ZZ)

S.<xbar> = R.quo((4 + 3*x + x^2, 1 + x^2)); S

 Quotient of Univariate Polynomial Ring in x over Integer Ring by the

ideal (x^2 + 3*x + 4, x^2 + 1)

 Sage: R.<x,y> = QQ[]; S.<a,b> = R.quo(1 - x*y); type(a)

 <class'Sage.rings.quotient_ring_element.QuotientRing_generic_with_category.ele
 ment_class'>

 Sage: a*b
```

1

Sage: S(1).is_unit()

True

۳.۲.۵. حلقه ی Z_n

اگر بخواهیم حلقه Z_n نشان دهیم از دستور Integers(n) استفاده می کنیم.

Sage: Integers(7)

Ring of integers modulo 7

می توان محاسبات معمول را روی این حلقه انجام داد.

Sage: R=Integers(13)

Sage: a=R(6)

Sage: b=R(5)

Sage: a+b

11

Sage: a*b

4

Sage: a.additive_order()

13

Sage:a.multiplicative_order()

12

Sage: a.is_unit()

True

معکوس جمعی عنصر a در این حلقه به صورت a– و معکوس ضربی به صورت

a^(-1) و یا 1/a می باشد.

Sage: (-a)

7

Sage: (a^(-1))

11

همچنین می توان برخی ویژگی های حلقه را نیز به وسیله ی دستور های زیر مشاهده کرد.

Sage: R=Integers(24)

Sage: R

Ring of Integers modulo 24

Sage: R.order()

24

Sage: R.is_Ring()

True

Sage: R.is_integral_domain()

False

R.is_field()

False

چون این حلقه متناهی است پس میتوان تمامی عناصر آن را پیدا کرد.

Sage:R=Integers(13)

Sage: R.list()

[0,1,2,3,4,5,6,7,8,9,10,11,12]

می دانیم Z_{13} میدان است. اگر حلقه ی ما میدان نباشد میدانیم یکه های Z_n تحت ضرب یک گروه تشکیل می دهند. Sage می تواند لیست مولد های گروه یکه ها را با دستور unit_gens() محاسبه کند.

Sage: R=Integers(12)

Sage; R.uni

R.unit_gens R.unit_group_order

R.unit_group_exponent R.unit_group_order

Sage: R.unit_gens()

[7,5]

می توانیم مرتبه ی این زیرگروه را محاسبه کنیم.

Sage: R.unit_group_order()

4

متاسفانه Sage دستوری که مستقیما یکه های Z_n به عنوان گروه را مشخص کند ندارد.

میتوان از راه های گوناگون این عناصر را یافت به عنوان مثال

```
Sage: [x for x in R if  x.is_unit() ]
[1,5,7,11]
```

۵.۲.۳.۱. حل معادلات در Z_n

میخواهیم معادله $9x=6$ را در Z_{21} حل کنیم .برای این منظور طبق دستورات زیر عمل می کنیم.

```
Sage: R=Integers(21)
Sage: a=R(9)
Sage: [x  for  x  in  R  if   R(9)*x == R(6) ]
[ 3,10,17 ]
```

راه دوم

```
Sage: solve_mod(9*x== 6, 21)
[ 3, 10 , 17 ]
```

همچنین معادلات چند متغیره را نیز می توان به همین نحو حل کرد.

۴.۲.۵. حلقه ی $\frac{Z}{Z_n}$

حلقه $\frac{Z}{Z_n}$ با استفاده از دستور Z mod (n) تعریف می شود.

مثال. حلقه $\frac{Z}{Z_{17}}$ و حلقه خارج قسمتی S روی حلقه $\frac{Z}{Z_{17}}$ را معرفی کرده و $(a+b)^{17}$ را محاسبه می کنیم.

Sage: R.<x,y>=Zmod(17)[ ]

Sage: R

Multivariate Polynomial Ring in x, y over Ring of integers modulo 17

Sage: S.<a,b>=R.quotient((x^2+y^2))

Sage: S

Quotient of Multivariate Polynomial Ring in x, y over Ring of integers

modulo 17 by the ideal (x^2 + y^2)

Sage: (a+b)^17

a*b^16 + b^17

فصل ششم

معرفی چند جمله ای ها

۱.٦. برای معرفی اعمالی که می توان روی چند جمله ای ها انجام داد ابتدا حلقه ی مورد نظر را تعریف کرده و سپس از دستورهای زیر استفاده می کنیم.

Sage: x,y,z = PolynomialRing(RationalField( ),3,[$'x','y','z'$], 'lex').gens()

Sage: $f = 9*y^6 - 9*x^2*y^5 - 18*x^3*y^4 - 9*x^5*y^4 - 9*x^6*y^2 + 9*x^7*y^3 + 18*x^8*y$^2 $- 9*x$^11

Sage: f.factor()

می توان چند جمله ای را تجزیه کرد.

(9) ∗ (−x^5 + y^2) ∗ (x^6 − 2 ∗ x^3 ∗ y^2 – x^2 ∗ y^3 + y^4)

Sage: f = 3 ∗ x^3 + x

Sage: g = 9 ∗ x ∗ (x + 1)

Sage: f.gcd(g)

می توان ب. م. م دو چند جمله ای را بدست آورد.

x

Sage: k= x^3 − 1

Sage: k.roots()

ریشه ی جند جمله ای با این دستور محاسبه می شود.

[(1,1)]

Sage: f= (x+3*y+x^2*y)^3

چند جمله ای را می توان گسترش داد.

Sage: f.expand()

$$X^6 * y^3 + 3 * x^5 * y^2 + 9 * x^4 * y + 18 * x^3 * y^2 + 27 * x^2 * y^3 + x^3 + 9 * x^2 * y + 27 * x * y^2 + 27 * y^3$$

جمله ی پیشرو با استفاده از عبارت رو به رو به دست می آید. f.lt( )

می توان ضریب پیشرو یک چند جمله ای را نیز به دست آورد. f.lc( )

یک جمله ای پیشرو نیز با نوشتن عبارت رو به رو بدست خواهد آمد. f.lm( )

مثال. چند جمله ای $f = x_0 + x_1 - 2x_1 x_2$ را روی میدان Q تعریف کنید و $f(1,2,0)$ را بدست آورید.

Sage: x= PolynomialRing(RationalField(),3, 'x').gens()

Sage: f= x[0]+x[1]-2*x[1]*x[2]

Sage: f

-2*x1*x2+x0+x1

Sage:f(1,2,0)

3

۲.٦. ایجاد آرایه ای از متغیر ها به صورت $a_i x^i$

برای نوشتن چند جمله ای ها به صورت $a_0x^0+ a_1x^1+ \ldots + a_nx^n$ با استفاده از نرم افزار maxima کافیست دستور زیر را بنویسیم.

```
Sage: P= maxima('sum(a[i]*x^i,i,0,n) ')
```

به عنوان مثال برای نوشتن $a_0+a_1x^1+a_2x^2+a_3x^3+a_4x^4$ در sage کافیست به جای n عدد 4 را جایگذاری کنیم.

```
Sage: p=maxima('sum(a[i]*x^i,i,0,4) ')

Sage: p

a[4]*x^4+a[3]*x^3+a[2]*x^2+a[1]*x+a[0]
```

مثال. حلقه ای تعریف کنید که ۴ عنصر اول آن ترتیب degree reverse lexicographical و دو متغیر آخر آن ترتیب negative lexicographical داشته باشد.

```
Sage: P.<a,b,c,d,e,f> = PolynomialRing(QQ,6, order='degrevlex(4),neglex(2)')

Sage: a> c^4

True

Sage: e > f^2
```

False

مثال. ب.م.م زیر را بدست آورید.

GCD(x^4+x^2+1, x^4-x^2-2x-1,x^3-1)

```
Sage: R.<x> = PolynomialRing(QQ,'x')

Sage: f= x^4+x^2+1

Sage: g= x^4-x^2-2x-1

Sage: h= x^3-1

Sage: gcd([f,g,h)]

1
```

٦. ٣. الگوریتم تقسیم (اعداد و چند جمله ای ها)

در حالت کلی برای نوشتن الگوریتم تقسیم از الگوریتم زیر را می نویسیم.

```
def euclide(a,b):
   r=a%b
   print (a,b,r)
   while r != 0:
      a=b; b=r
      r=a%b
      print (a,b,r)
```

در نتیجه دستوری به نام euclide (a,b) ساختیم و میتوانیم از این پس از این دستور برای الگوریتم تقسیم استفاده کنیم.

```
Sage: euclide(12,5)
(12, 5, 2)
(5, 2, 1)
(2, 1, 0)
```

٤.٦. الگوریتم تقسیم چند جمله ای ها

❖ روش اول.

برای تقسیم چند جمله ای ها الگوریتم زیر را می نویسیم و آن را به عنوان فایلی ذخیره می کنیم تا در صورت نیاز بتوان به راحتی از آن استفاده کرد.

```
def division(dividend, divisor) :
   print 'quotient: ', (dividend._maxima_().divide(divisor).Sage())[0]
   print 'remainder: ', (dividend._maxima_().divide(divisor).Sage())[1]
```

مثال.

در واقع ما این الگوریتم را بر اساس دستور برنامه "maxima" نوشتیم و از این پس میتوانیم از دستور division(dividend, divisor) استفاده کنیم.

```
division(x^4 + 2*x^3-x^2+5*x - 2,x^2+1)
quotient:  x^2 + 2*x - 2
remainder:  3*x
```

❖ روش دوم.

تابع را می توانیم به شکل زیر نیز بنویسیم.

```
def division(dividend, divisor) :
   q,r = dividend.maxima_methods().divide(divisor)
   print 'quotient: ', q
   print 'remainder: ', r
```

مثال.

```
Sage: f(x)=x^3+5*x^2-3*x+1
Sage: g(x)=x+1
Sage: f.maxima_methods().divide(g)
[x^2 + 4*x - 7, 8]
```

❖ روش سوم.

ممکن است بخواهیم دو چند جمله ای را با ترتیب خاصی مانند ...,lex,grlex بر هم تقسیم کنیم در این صورت کافی است نرم افزار "maxima" را به شکل زیر فراخوانی کنیم.

مثال. دو چند جمله ای $f= x^7y^2+x^3y^2-y+1$ و $F=(xy^2,x-y^3)$ و ترتیب grlex را در نظر میگیریم.

```
var('x,y')
maxima('load(grobner)')
maxima('poly_monomial_order:grlex')
F=[x*y^2-x , x-y^3]
ans=maxima('poly_pseudo_divide(x^2*y^2+x^3*y^2-y+1,[x*y^2-x,x-y^3],[x,y])')
print ans
(quo,rem,n,m)=ans
p=quo[0]*F[0]+quo[1]*F[1]+rem
print p
```

$y^4 + y^3 + (x y + y + x^2 + x + 1) (x - y^3) + (y^2 + x y + y + x^2 + x + 1) (x y^2 - x)$
$- y + 1$

٦. ٥. چند جمله ای های تحویل پذبر و تحویل ناپذیر

برای مشخص کردن تحویل پذیری یک چند جمله ای از فرآیند زیر بهره می بریم.

```
Sage:R.<x> = QQ [ ]
F=(x^3-x^y+y^2-x)*(x^5-3/2*x-y)
Sage: f.factor( )
Sage: len(f.factor( ))
2
```

بدست می آید ۱ باشد آنگاه چند جمله ای تحویل پذیر است در غیر (( )len(f.factor اگر جوابی که از دستور اینصورت تحوبل ناپذیر است.

فصل هفتم

ایده آل ها

و

پایه گروبنر

۷. ۱. ایده آل ها

برای معرفی ایده آل ابتدا حلقه ی مورد نظر را تعریف میکنیم. سپس میتوان از دو راه ایده آل یک حلقه را تعریف کرد.

```
R=PolynomialRing(QQ,3, 'x,y,z', 'lex')
```

- ❖ I= ideal(generators)/ I= Ideal(generators)
- ❖ I= f*R   #f is a generator of Ideal I#

مثال.

```
Sage: P.<x,y,z> = PolynomialRing(ZZ,order='lex')
```

```
Sage: I = ideal(-y^2 - 3*y + z^2 + 3, -2*y*z + z^2 + 2*z + 1, \
            x*z + y*z + z^2, -3*x*y + 2*y*z + 6*z^2)
Sage: I
Ideal (-y^2 - 3*y + z^2 + 3, -2*y*z + z^2 + 2*z + 1, x*z + y*z + z^2,
-3*x*y + 2*y*z + 6*z^2) of Multivariate Polynomial Ring in x, y, z over
Integer Ring
```

۲.۷. عضویت در ایده آل

عضویت یک چند جمله ای در یک ایده آل با استفاده از دستور in می باشد.

مثال.

```
R.<x>=PolynomialRing(QQ, 'x')
Sage: I=ideal(x^4-6*x^2+12*x-8,2*x^3-10*x^2+16*x-8)
Sage: f = x^2-4*x+4
Sage: f in I
True
```

۳.۷. تشخیص نوع ایده آل

برای تشخیص اینکه ایده آل اصلی، ماکسیمال و یا اول است از دستور is_prime/principal/maximal () استفاده می کنیم.

(به طور کلی با نوشتن ... _is و سپس فشردن کلید Tab میتوان دستورهای زیادی از این قبیل را یافت.)

مثال. با توجه به ایده آل $I = (x^4 - 6x^2, 2x^3 - 10x^2)$

```
Sage:I.is_prime()
false
Sage:I.is_principal()
True
```

۴.۷. ایده آل رادیکال

در مواردی که بخواهیم رادیکال یک ایده آل را بدست آوریم از عبارت I.radical( ) استفاده می کنیم.

```
Sage: I=ideal(x+y+z-3,x^2+z^2+y^2-5,x^3+y^3+z^3-7)
Sage: I.radical()
Ideal(x+y+z-3,y^2+y^z+z^2-3*y-3*z+2,3*z^3-9*z^2+6*z+2) of Multivariate
Polynomial Ring in x,y,z over Rational Field
```

۵.۷. اشتراک دو ایده آل

برای بدست آوردن اشتراک دو ایده آل I و J پس از معرفی حلقه و ایده آل ها از دستور I.intersection(J) استفاده می کنیم.

```
R.<x> = PolynomialRing(QQ, 1)
Multivariate Polynomial Ring in x over Rational Field
Sage: I = R.ideal(x^2-1) ; I
Ideal (x^2 - 1) of Multivariate Polynomial Ring in x over Rational Field
```

```
Sage: J = R.ideal(x^2-2) ; J
Ideal (x^2 - 2) of Multivariate Polynomial Ring in x over Rational Field
Sage: I.intersection(J)
Ideal (x^4 - 3*x^2 + 2) of Multivariate Polynomial Ring in x over Rational Field
```

٦.٧. پایه گروبنر

دستوری که برای محاسبه پایه گروبنر محاسبه می شود عبارت

```
B=I.groebner_basis( )
```

می باشد.

مثال.

```
Sage: P.<x,y,z> = PolynomialRing (ZZ,order='lex')
Sage: I = ideal(−y^2 − 3 ∗ y + z^2 + 3, −2 ∗ y ∗ z + z^2 + 2 ∗ z + 1,\
        x ∗ z + y ∗ z + z^2, −3 ∗ x ∗ y + 2 ∗ y ∗ z + 6 ∗ z^2)
Sage:I,groebner_basis()
[x + 130433 ∗ y + 59079 ∗ z, y^2 + 3 ∗ y + 17220, y ∗ z + 5 ∗ y
        + 14504, 2 ∗ y
 +158864, z^2 + 17223, 2 ∗ z + 41856, 164878]
```

٧.٧ . الگوریتم بوخبرگر

برای محاسبه الگوریتم بوخبرگر راه های زیادی وجود دارد اما ساده ترین حالت آن استفاده از دستوری از نرم افزار "maxima" می باشد. کافیست ابتدا برنامه Maxima را فراخوانی کرده و دستور زیر را وارد کنیم.

```
Sage: maxima('load(grobner) ')
```

```
Sage: ans=maxima('poly_buchberger ([−y^2 − 3 ∗ y + z^2 + 3, −2 ∗ y ∗ z +
 z^2 + 2 ∗ z + x ∗ z + y ∗ z + z^2, −3 ∗ x ∗ y + 2 ∗ y ∗ z + 6 ∗
z^2], [x, y, z])′)
```

Ans

```
[z^2 − y^2 − 3 ∗ y + 3,2 ∗ z^2 − y ∗ z + x ∗ z + 2 ∗ z, 6 ∗ z^2 + 2 ∗ y ∗ z − 3 ∗ x
    ∗ y, 8 ∗ z^3 + 3 ∗ y ∗ z^2 + 24 ∗ z^2 +\6 ∗ z − 9 ∗ x, 3 ∗ z^3 + 8 ∗ y
    ∗ z^2 + 15 ∗ y ∗ z − 9 ∗ z, −440 ∗ z^4 − 1671 ∗ z^3 − 1752 ∗ z^2
    − 99 ∗ y ∗ z − 747 ∗ z\, −55 ∗ z^5 − 312 ∗ z^4 − 606 ∗ z^3 − 504
    ∗ z^2 − 189 ∗ z]
```

۸.۷ . حذف یک متغیر از ایده آل

اگر بخواهیم یک متغیر را از ایده آل I حذف کنیم می توانیم از دستور

I.reduce(variable) استفاده کنیم.

به عنوان مثال اگر بخواهیم متغیر y را از ایده آل $I = (x^3 + y, y)$ حذف کنیم با استفاده از این دستور خواهیم داشت.

```
Sage: R.<x,y>=PolynomialRing(QQ,2)
Sage: I=ideal(x^3+y,y)
```

```
Sage:I
Ideal (x^3 + y, y) of Multivariate Polynomial Ring in x, y over Rational
Field
Sage: I.reduce(y)
0
```

اگر بخواهیم طبق قضیه حذف ایده آل حذفی ایده آل I را بدست آوریم کافیست از دستور

I.elimination_ideal([…,…]) و دستور پایه گروبنر استفاده کنیم.

مثال.

```
R.<x,y,t,s,z>=PolynomialRing(QQ,5)
I=ideal(x-t, y-t^2, z-t^3, s-x+y^3)
I.elimination_ideal([t,s])
Ideal(y^2-x*z, x*y-z,x^2-y) of multivariate PolynomialRing in x,y,t,s,z over
Rational Field
```

۷ . ۹. محاسبه S – چند جمله ای

برای محاسبه S–چند جمله ای ها ابتدا لازم است دستور spol را از sage.ring.polynomial خارج کنیم.

برای این منظور به روش زیر عمل می کنیم.

```
Sage: from Sage.rings.polynomial.toy_buchberger import spol

Sage: R.<x,y,z> = PolynomialRing(QQ,3,' lex' )
```

```
Sage: f = 4*x^2*z -7*y^2

Sage: g=x*y*z^2+3*x*z^4

Sage: spol(f,g)

-1/3*x^2*y*z^2 - 7/4*y^2*z^3
```

فصل هشتم

واریّه های آفین

و

صفحه آفینی

۱.۸. فضای آفینی و واریه های آفین

برای معرفی واریته آفین مانند $V(x^2 - y^2z^2 + z^3)$ ابتدا فضای آفینی را معرفی می کنیم و سپس به معرفی واریه آفین می پردازیم.

مثال.

Sage: A3.<x,y,z>=AffineSpace(QQ,3)

$Affine\ Space\ of\ dimension\ 3\ over\ Rational\ Field$

Sage: V= A3.subscheme ([x^2-y^2*z^2+z^3])

$-y\hat{}z\hat{}2 + z\hat{}3 + x\hat{}2$

Sage: V

$Closed\ subscheme\ of\ Affine\ Space\ of\ dimention\ 3\ over\ Rational\ Field$

$defined\ by$:

$-y\hat{}2 * z\hat{}2\ +\ z\hat{}3 +\ x\hat{}2$

Sage:V.rational_points (bound = 3)

[(-3, -2, 3), (-3, 2, 3), (-1, 0, -1), (0, -3, 0), (0, -2, 0), (0, -3/2, 0), (0, -1, 0), (0, -1, 1), (0, -2/3, 0), (0, -1/2, 0), (0, -1/3, 0), (0, 0, 0), (0, 1/3, 0), (0, 1/2, 0), (0, 2/3, 0), (0, 1, 0), (0, 1, 1), (0, 3/2, 0), (0, 2, 0), (0, 3, 0), (1, 0, -1), (3, -2, 3), (3, 2, 3)]

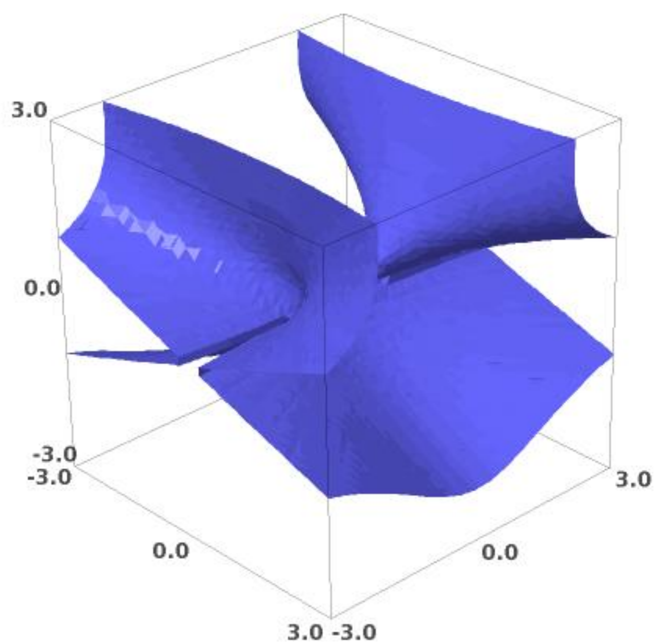
۲.۸. رسم واریه های آفین

جهت رسم واریته های آفین از روش های زیر استفاده می کنیم.

مثال. واریته آفین($V(x^2 - y^2z^2 + z^3)$ را رسم کنید.

مطابق قواعد ذکر شده در مورد رسم نمودار ها ابتدا متغیر ها را معرفی کرده سپس ازدستور implicit_plot3d() استفاده می کنیم.

```
Sage:Var(x,y,z)
Sage: implicit_plot3d(x^2-y^2*z^2+z^3,[-3,3],[-3,3],[-3,3])
```



مثال.

$V((x-2)(x^2-y),y(x^2-y),(z+1)(x^{\wedge 2}-y))$ را رسم می کنیم.

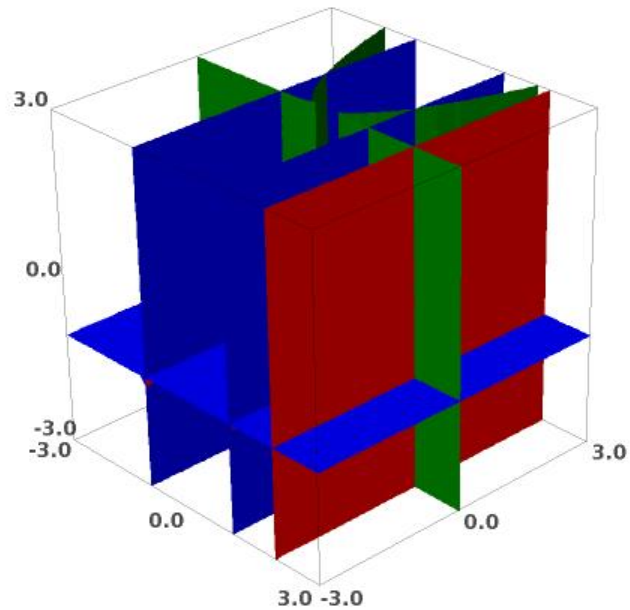
اینگونه واریته ها را با دو روش میتوان رسم کرد.

❖ روش اول.

```
var('x y z')
p=implicit_plot3d((x-2)*(x^2-1),[x,-3,3],[y,-3,3],[z,-3,3],color='red')
p+=implicit_plot3d(y*(x^2-y),[x,-3,3],[y,-3,3],[z,-3,3],color='green')
p+=implicit_plot3d((z+1)*(x^2-1),[x,-3,3],[y,-3,3],[z,-3,3],color='blue')
show(p)
```

❖ روش دوم.

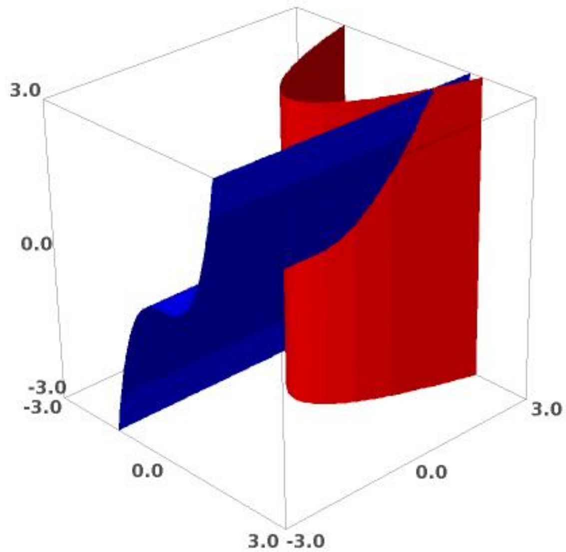
```
var('x y z')
V=[(x-2)*(x^2-1) , y*(x^2-y),(z+1)*(x^2-1)]
c=['red','green','blue']
p=add([implicit_plot3d(V[i],[x,-3,3],[y,-3,3],[z,-3,3],color=c[i]) for i in [0..2]])
show(p)
```



مثال. $V((y-x^2),(z-x^3)]$ را رسم کنید.

توجه شود که تنها تفاوت این مثال با مثال قبل ، در تعریف بازه ی مناسب برای i می باشد.

```
Var('x,y,z')
V=[(y-x^2),(z-x^3)]
C=['red', 'green']
P= add([implicit_plot3d(V(i),[x,-3,3],[y,-3,3],[z,-3,-3],color = C[i] ) for i in [0..1]])
```



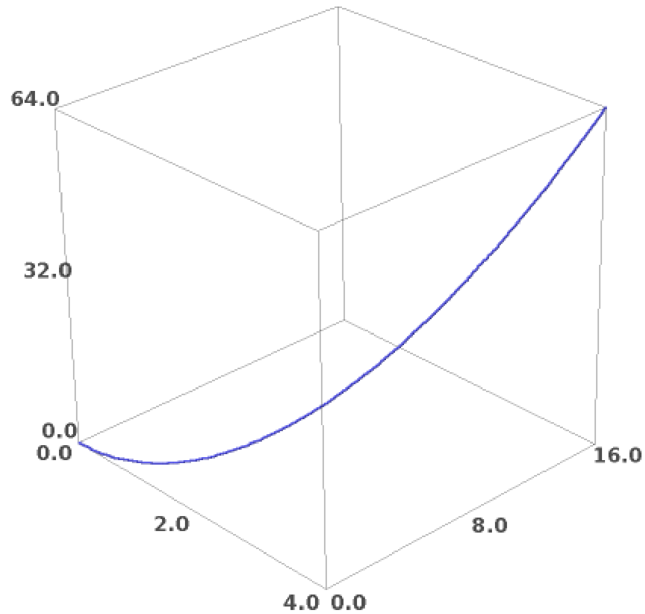
۸۹

مثال. $V(x^2 + y^2 - 1)$ که همان منحنی درجه ۳ تابدار می باشد را رسم می کنیم.

Var (' t ')

Parametric_plot3d((t,t^2,t^3),(t,0,4),thickness=3)

Thickness ضخامت منحنی را تغییر می دهد.


64.0
32.0
0.0
0.0
2.0
4.0
0.0
8.0
16.0

فصل نهم

ماتریس ها

و

حل معادلات

۹.۱. ماتریس ها

برای معرفی یک بردار از دستور vector([----]) استفاده می‌شود.

V=Vector([1,2,3,4])

V[0] = 1

اعمال ریاضی روی بردارها براحتی قابل محاسبه است.

Sage : 7*V

(7,14,21,28)

Sage: V+Vector([1,4,6,8])

(2.6,9,12)

Sage: V*V

30

برای ساختن یک ماتریس از دستور matrix ( ) استفاده می‌کنیم، سپس در پرانتز سطر های ماتریس را به صورت مجزا معرفی می‌کنیم.

Sage : matrix ([[1,2],[3,4]])

و یا می توان سطر وستون ماتریس را معرفی کرد ، سپس تمامی عناصر را به ترتیب نوشت.

Sage : matrix (4,2,[1,2,3,…,8])

- برای یک ماتریس مربعی تعداد ستون ها را می توان ننوشت.

Matrix (2,[1,2,3,4])

Sage به صورت قرارداد ماتریس را روی کوچکترین مجموعه ای تعریف می‌کند که عناصر از آن مجموعه انتخاب شده اند.

Sage : parent (matrix (2,[1,2,2/1,4])

$Full\ MatrixSpace\ of\ 2\ by\ 2\ dense\ matrices\ over\ Rational\ Field$

Sage : parent (matrix (2,[x,x^2,x-1,x^3])

$Full\ MatrixSpace\ of\ 2\ bye\ 2\ dense\ matrices\ over\ symbolic\ Ring$

می توان ماتریس را روی مجموعه دلخواه تعریف کرد.

Sage: matrix (QQ,2,[1,1,3,4])

[1 1]

[3 4]

ماتریس همانی را به صورت رو به رو تعریف می کنیم.

Sage : identity_matrix(3)

[1 0 0]

[0 1 0]

[0 0 1]

اگر نیاز به معکوس یک ماتریس داریم ابتدا ماتریس را نام گذاری کرده سپس از دستور 1-^ استفاده می‌کنیم.

Sage :A=matrix(2,[1,1,0,1])

Sage :A^-1

[1 -1]

[0 1]

A.det( )

1 دستوری که برای یافتن دترمینان به کار می رود به صورت مقابل است.

Sage:A=matrix ([[$-1/2$,0,-1], [0,-2,2], [1,0,$-1/2$]])

Sage: A.det( )

-5/2

۹. ۱. ۱ عملیات سطری مقدماتی

می‌توان ضریبی از یک سطر یا ستون را سطر یا ستون دیگر افزود برای انجام این عمل از دستور

add_multipl_of_row( )

استفاده می‌شود.

Sage: M=matrix(QQ,[[1,2,3],[4,5,6],[7,8,9]])

Sage: M.add_multiple_of_row(1,0,-4);M

[1 2 3]

[0 -3 -6]

[7 8 9]

می‌توان سطر یا ستونی را در عددی ضرب کرد. دستور

Sage: M.rescale_row(1,$-1/4$);M

[1   2   3]

[0 ¾ 3/2]

[7 8   9]

این کار را انجام می دهد.

اگر فرم echelon یک ماتریس را بخواهیم از عبارت ( )ehelon-form یا ( )ehelonsize استفاده می-

کنیم.

حال با اطلاعات بدست آمده قادر خواهیم بود با استفاده از ماتریس افزوده که با دستور :

M.augment ( )

بدست می‌آید و فرم ehelon یک ماتریس۴ × ۳ را به صورت Mx=b حل کنیم.

ابتدا M و b را تعریف می‌کنیم.

```
Sage: M=matrix(QQ,[[2,4,6,2,4],[1,2,3,1,1],[2,4,8,0,0],[3,6,7,5,9]]); M
[2 4 6 2 4]
[1 2 3 1 1]
[2 4 8 0 0]
[3 6 7 5 9]
```

```
Sage:b=vector ([56,23,34,101]);b
(56, 23, 34, 101)
```

حال ماتریس به فرم (M | b) را می سازیم و سپس فرم echelon را به دست می‌آوریم.

```
Sage: M_aug=M.augment(b); M_aug

[ 2  4  6  2  4  56]
[ 1  2  3  1  1  23]
[ 2  4  8  0  0  34]
[ 3  6  7  5  9 101]
```

```
Sage:M_aug.echelon_form( )
```

[1 2 0 4 0 21]

[0 0 1 -1 0 -1]

[0 0 0 0 1 5]

[0 0 0 0 0 0]

این نشان می‌دهد ما یک فضای جواب یک بعدی از بردار هایی به فرم V=c(-2,1,0,0,0)+(21,0,1,0,5) هستند. برای بدست آوردن جواب ، از دستور

solve_right( )

استفاده می شود. یعنی

```
Sage: M.solve_right(b)
(21, 0, -1, 0, 5)
```

۲ .۹ . پارامتری سازی و حل معادلات

به طور کلی برای حل معادلات از دستور solve([ ]) استفاده می شود و همانطور که در مبحث چند جمله ای ها ذکر شد، برای بدست آوردن ریشه های یک چند جمله ای f از دستور f.roots( ) استفاده می کنیم.

مثال.اگر $f = x^2 + y^2 - 1$ دایره ای به مرکز مبدا و شعاع ۱ باشد و

$g = (x - c)^2 + y^2 - r^2$ معادله ی دایره ای دیگر باشد؛ جواب های این دو معادله را بدست آورید.

```
sage: c,r = var('c,r')
```

Sage:f= x^2+y^2-1

Sage:g= (x-c)^2+y^2-r^2

Sage: solve([f= =0, g= =0],x,y)

[[x = = ½*(c^2 – r^2 +1)/c , y = = - 1/2*sqrt(-c^4 + 2*c^2*r^2 + 2*c^2 – r^4 +2 * r^2 -1)/c] ,[x = = 1/2*(c^2-r^2 +1)/c , y = = 1/2*sqrt(-c^4 + 2*c^2*r^2 + 2*c^2 – r^4 +2*r^2 -1)/c]]

مثال.

اگر $x=cos\ t$ و $y=cos\ 2t$ قستی از یک سهمی را پارامتری کند، yرا بر حسب x بدست می آوریم.

Sage: t = var('t')

برای اینکه بتوان y را تابعی از x نشان داد باید از دستور simplify_trig استفاده کنیم.

Sage: cos(2*t).simplify_trig()

2*cos(t)^2 – *1*

مثال.

$$x + 2y - 2z + w = -1$$

$$x + y + z - w = 2$$

Sage: var('x,y,z,w')

Sage: solve([x+2*y-2*z+w==-1,x+y+z-w==2],[x,y])

[[x = = 3*w - 4*z + 5, y = = -2*w + 3*z - 3]]

ضمیمه

مقایسه Sage با چند نرم افزار ریاضی (کلیات)

System	Creator	Development started	First public release	Cost (USD)	Open source	License
Algebrator	Neven Jurkovic	1986	1999	$58.99	No	Proprietary
GAP	GAP Group	1986	1986	Free	Yes	GPL
Java Algebra System	Heinz Kredel	2000	2005	Free	Yes	GPL or LGPL
Maple	Symbolic Computation Group, University of Waterloo	1980	1984	$2,275 (Commercial), $2,155 (Government), $1245(Academic), $239 (Personal Edition), $99 (Student), $79 (Student, 12-Month term)[2]	No	Proprietary
Mathematica	Wolfram Research	1986	1988	$2,495 (Professional), $1095 (Education), $140 (Student), $69.95 (Student annual license)[4] $295 (Personal)[5]	No	Proprietary
Maxima	MIT Project MAC and Bill Schelter et al.	1967	1998	Free	Yes	GPL
Microsoft Mathematics	Microsoft	?	2005	Free	No	Proprietary
Sage	William A. Stein	2005	2005	Free	Yes	GPL
SINGULAR	University of Kaiserslautern	1984	1997	Free	Yes	GPL
MATLAB	MathWorks	1989	2008	$2900 including required MATLAB	No	Proprietary
Wolfram Alpha	Wolfram Research		2009	Pro version: $4.99 / month, Pro version for students: $2.99 / month, Regular version free.	No	Proprietary

مقایسه Sage با چند نرم افزار ریاضی (توانایی)

	Formula editor	Arbitrary precision	Integration	Integral transforms	Equations	Inequalities	Diophantine equations	Differential equations	Recurrence relations	Graph theory	Number theory	Quantifier elimination	Boolean algebra	Tensors	Probability	Control Theory
Algebrator	Yes	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	?	?
Magma	No	Yes	No	No	Yes	No	Yes	No	No	Yes	Yes	No	No	No	?	?
Maple	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Mathematica	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
MATLAB	No	Yes	Yes	Yes	Yes	No	No	Yes	?	No	?	No	No	?	?	?
Maxima	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	Yes	?	?
Microsoft Mathematics	Yes	No	Yes	No	Yes	Yes	No	No	No	No	No	No	Yes	No	?	?
Sage	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Wolfram Alpha	Pro version only	Yes	Yes	Yes	Yes	Yes	?	Yes	Yes	Yes	Yes	Yes	Yes	No	?	?