Subject
Date

امیرحسین کریم زادگان

- evalf ( ) → اگر بخواهیم جواب عملیات ریاضی اعشاری داده شود
- evalf (عملیات و تعداد ارقام اعشار که می‌خواهیم)

یا ↓

- evalf [تعداد ارقام اعشار] (عملیات)
- Digits → تعداد ارقام اعشار default که نمایش داده می‌شود
- x := 5 → برای تعریف یک متغیر باید از علامت تعریف := استفاده کرد
- Digits := 20 → تعداد ارقام اعشار default به 20 تغییر می‌کند
- about (متغیر) → اطلاعات مربوط به متغیر را نمایش می‌دهد (مثلا integer بودن یا ...)
- x := 'x' → این کار مقدار تخصیص داده شده به متغیر x را پس می‌گیرد
- Pi → برای کار کردن با عدد π باید Pi با حرف اول بزرگ استفاده کرد
- بدون زدن evalf() در ابتدا، حاصل به صورت کسری نمایش داده می‌شود
- convert (35 degrees , radians) این دستور 35 درجه را به رادیان تبدیل می‌کند
- sin (%) → استفاده از % یعنی استفاده از آخرین دستور. استفاده از % % یعنی استفاده از یکی مانده به آخرین دستور (که می‌توان از % بیشتر برای دسترسی به خطوط قبل تر استفاده کرد)
- ? convert → علامت سوال، بعد هر چیزی، help مربوط به آن چیز را می‌دهد (در این مثال help مربوط به convert می‌آید)

Subject
Date

- sqrt() → ریشه یک عدد را می دهد

- root(7, 3) → ریشه سوم عدد 7 داده می شود

- برای یافتن ریشه می توان از عدد به توان کسری استفاده کرد

- proot() → اگر ریشه صحیح باشد آن را می دهد. اگر صحیح نبود می گوید ریشه ندارد

- ifactor() → عوامل تشکیل دهنده عدد را نشان می دهد

- abs(x) → یعنی قدر مطلق x

- $P2 = x^{\frac{1}{5}} + 3\sin x - 4|x|$
  در برنامه: P2:= root(x, 5) + 3 sin(x) - 4abs

- P2(3) → این دستور به دستی جای x عدد 3 را نمی گذارد

- subs(x = 3, P2) → این دستور به دستی جای x عدد 3 را می گذارد در تابع P2

- P3:= factor(P2) → تابع P2 را به عوامل اولیه اش تجزیه می کند

- expand(P3) → P3 که تابع تجزیه شده به عوامل اولیه است را به حالت تابع عادی برمی گرداند

- P4:= x+1 • P5:= $x^2 - 3x - 4$

- simplify($\frac{P5}{P4}$) → دو تابع را تقسیم برهم می کند (اینجا x-4 می دهد)

- solve(P5, x) → این ریشه های x در P5 را می یابد

P1 := 4x + 5y = 3 , P2 = 5x+y=15 LINOMAX

- solve({P1, P2}, {x, y}) → این ریشه های دو معادله دو مجهول را می دهد

Subject
Date

امیرحسین کریم زادگان

- $quo(h, p, x)$ — این دستور خارج قسمت تقسیم دو چند جمله ای $\left(\frac{h}{p}\right)$ را می دهد بر مبنای x

چند جمله ای — مبنا

- $iquo(10, 2) \rightarrow \left(\frac{10}{2}\right)$ این دستور خارج قسمت تقسیم دو عدد بر هم را می دهد و اگر حاصل اعشاری شود، به پایین روند می کند

- $op(h)$ — عدد ثابت را در چند جمله ای نمایش می دهد. اگر ما عدد ثابت نداشته باشد 0 نمایش داده می شود

چند جمله ای

- $nops(h) \rightarrow h := x^2+2x+2$ این تعداد اعضای چند جمله ای را نمایش می دهد که 3 نشان می دهد

چند جمله ای

- $randpoly(x, degree = 3)$ یک چند جمله ای random با متغیر x و ماکسیمم در جه 3 ایجاد می کند. چون ضرایب random هستند می توانند 0 هم باشند و به همین دلیل برخی در جه ها ممکن است ظاهر نشوند

بیشتر از ۱ هم ممکن: $\{x, y\}$

* درجه=3 یعنی جمع درجات x و y بشود 3

- $randpoly(x, degree = 3, terms = 3)$ terms تعداد اعضای چند جمله ای را مشخص می کند

- $sort(h)$ چند جمله ای را مرتب می کند از توان بزرگ به کوچک

- $h := x^3y^2 + 2x^4y^3 - 4x$ چند جمله ای h را برحسب x از توان های بزرگش به کوچک مرتب می کند
$sort(h, [x])$

- $sort(h, [x, y], ascending)$

چند جمله ای — ترتیب مرتب سازی x و y — از کوچک به بزرگ

LINOMAX

Subject

Date

- coeff (h , x) — چیزهایی که در x ضرب شده اند را نمایش می دهد ، برای h ، -4 را نمایش می دهد چون ضریب x است

- f := x → $x^4 - 4x$ — تابع ساخته می شود (نه چند جمله ای)
  برای ساخت فلش از خط فاصله (-) به علاوه علامت بزرگتر است

- $f(x) = \dfrac{\sin x}{\arccos(\frac{x}{4})}$ — تابع ساخته می شود

- P := randpoly (x)
  g := unapply (P , x)
  این دو خط یک چند جمله ای را به یک تابع تبدیل می کند

- h := x → if x < 4 then $x^2 - 4$
  else $e^x \sin(x)$ end if

- exp(x) — این دستور e^x را نتیجه می دهد

- r := x → if x < 6 then x + 2
  elif x = 6 then x
  else x^2 end if
  اگر ضابطه های بیشتری بخواهیم اضافه کنیم از elif استفاده می کنیم

- floor(1.6) — عدد داخل را به نزدیک ترین عدد صحیح کوچکتر تبدیل می کند
  1.6 را به 1 تبدیل می کند

- ceil (1.6) — عدد داخل را به نزدیک ترین عدد صحیح بزرگتر تبدیل می کند
  1.6 را به 2 تبدیل می کند

- Piecewise (شرط۱ , ضابطه۱ , شرط۲ , ضابطه۲ , ...) — تابع چند ضابطه ای ایجاد می کند (بدون دردسر نوشتن if , elif , endif)

LINOMAX

- Piecwise (شرط۱ , ضابطه۱ , شرط۲ , ضابطه۲ , ضابطه۳) — می توان برای ضابطه ۳ شرط تعریف نکرد . عدد های باقی مانده اتومات شرط ضابطه ۳ می شوند

Subject
Date

امیرحسین کریم‌زادگان

- piecewise(x<4 , x+3 , x=4 , x-10 , x+5)

در اینجا متغیرهایی که x+5 می‌شوند متمم شروط قبلی است (یعنی x>4)

- P1
P2
divide(P1 , P2)

این دستور دو چند جمله ای را بر هم تقسیم می کند و اگر قابل تقسیم باشند True می دهد و در غیر این صورت False می دهد $\left(\frac{P_1}{P_2}\right)$

- int(x^3 , x)

x^3: از این انتگرال گرفته می‌شود — x: مبنا

انتگرال نامعین را می دهد بر حسب x

- int($x^2 - \frac{1}{x^3}$, x = 0 .. 10)

انتگرال معین را می دهد برحسب x در بازه 0 تا 10 (بین دو عدد مورد نظر باید دو تا نقطه بگذاریم)

- r = piecewise(x>2 , x^2 , x=2 , 4x , x<2 , x-2)
int(r , x)

انتگرال تابع چند ضابطه‌ای را می دهد

- with(plots)

plot ها را import می کند و آنها را نمایش می دهد

- with(plots):

plot ها را import می کند ولی آنها را نمایش نمی دهد

- plot( )

نمودار چیزی که داخل () است را می دهد

- plot(x^2 , x = -10 .. 10 , color = blue , thickness = 3)

نمودار x^2 را در بازه -10 تا 10 رسم می کند. (در تعریف رنگ ، ضخامت ، ترتیب مهم نیست)

- plot(, scaling = constrained)

ابعاد محور x ها و y ها را برابر می کند

LINOMAX

- plot([x^2 , x^3 , arcsin(x)])

چند نمودار را رسم می کند

Subject
Date

- Circle ((0,0) , 6)   برای رسم دایره
  مختصات مرکز   شعاع

- Circle ((0,0) , 6 , 0.. Pi)   برای رسم منحنی
  بازه زاویه‌ها ، (خلاف جهت عقربه‌های ساعت حرکت می‌کند)

- P := Point (2 , 2)   نقطه P را می‌توان مرکز دایره در نظر گرفت
  Circle (P, 6, 0 .. Pi)

- plot ([x^2 , x^3] , color = [red, blue] , thickness = [10, 3])
  وقتی نمودار چند چیز را می‌خواهیم و به آن‌ها رنگ و ضخامت مختلف می‌خواهیم دهیم

- P1 := plot (x , x = 0 .. 2 , color = red ,...) :   وقتی نمودار چند چیز
  P2 := plot (x^2 , x = -2 .. 2) :   را هر کدام در بازه‌های متمایز
  display (P1 , P2)   بخواهیم از display استفاده می‌کنیم
  که (در پایان خط ۱ و ۲ ، دو نقطه می‌گذاریم تا P1 و P2 را نمایش ندهد)

- P := plot (x^2 , x = 0 .. 10 , y = -5 .. 5)   می‌توان هم برای x ، هم y بازه تعیین کرد

- P := ([$(9-x^2)^{\frac{1}{2}}$, $-(9-x^2)^{\frac{1}{2}}$] , x = -3 .. 3 , y = -3 .. 3)   برای رسم دایره

- plot (floor(x) , discont = True)   اگر یک تابع گسسته مانند [x] داشته باشیم ، maple سعی می‌کند آن را پیوسته رسم کند . برای اینکه شکل گسسته رسم شود ، از discont = true استفاده می‌کنیم

- S1 := 1, a, 4, 9, 1, ab, c, 2, 12   این دستور یک دنباله ایجاد می‌کند
  در این حالت خودمان برای جملات مقدار تعیین می‌کنیم

LINOMAX

CamScanner

Subject
Date

- $S2 := seq(\frac{1}{i^2}, i = 1..100)$ یک دنباله با ضابطه $\frac{1}{i^2}$ می نویسد و 100 جمله اول آن را نمایش می دهد

- $S3 := seq(\sqrt[3]{i}, i \ in \ S2)$ یک دنباله با ضابطه $\sqrt[3]{i}$ می سازد ولی i های خویش را از جملات دنباله S2 بدست می آورد: (مثال: $i=1 \rightarrow S2: \frac{1}{1^2} \rightarrow S3: \sqrt[3]{\frac{1}{1^2}} = 1$)

- $r4 := solve(x^2 + y^2 = 9)$ این به ما یک دنباله با ۲ عضو می دهد

- r4 [1] عضو اول r4 را نمایش می دهد

- L1 = [S1] یک لیست تشکیل می دهد که اعضای دنباله S1 با حفظ ترتیب در آن هستند

- A = {S1} یک مجموعه ایجاد می کند و اعضای S1 را در آن قرار می دهد
که I ترتیب را از کوچک به بزرگ می کند
که II اعضای تکراری را فقط یک بار نشان می دهد

- op (L1) دستور op ، اعضای یک لیست یا مجموعه را خارج از آن نشان می دهد

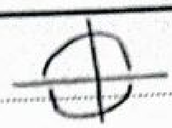
- nops (L1) تعداد اعضای یک لیست یا مجموعه را نمایش می دهد

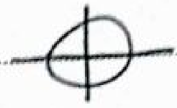
- L2 := [1, 2, 8, 1, [1, 3], 9, 5] با دستور member می توانیم چک کنیم آیا یک عضو خاص در لیست وجود دارد.
member (2, L2)
اگر وجود داشت true می دهد، لیست مد نظر L که آیا 2 در آن وجود دارد
اگر وجود نداشت False می دهد

LINOMAX

CamScanner

Subject
Date

- plot([r4[1], r4[2]]) را  این رسم می کند

- d:=[solve($x^2+y^2=9$, y)] این دستور، یک جا هر دو بخش را رسم می کند 

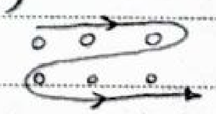
یک فاصله — یک فاصله

- A union B — اجتماع
  A intersect B — اشتراک
  A minus B — تفاضل

- f(x):= x^2 — با این دستور، مقادیری از تابع f نمایش داده می شود که مضرب ۳ باشند
  s52:= seq(3i, i=1..5) — f(3)، f(6)، f(9)، ....
  map(f,{s52})
  می شد به این شکل هم نوشت: map(f,[s52])

  {s52} → مقادیر را به شکل مجموعه نشان می دهد
  [s52] → مقادیر را به شکل لیست نشان می دهد

- with(linalg) / with(linear algebra) — برای استفاده از ماتریس باید ابتدا این را فراخوانی کنیم

- A1:= matrix(2, 3, [1, 4, 5, 3, 2, 6])
  سطر، ستون، مقادیر از چپ بالا به راست پایین 

- A2:= Matrix(2, 3, 1) — ماتریس تشکیل می دهد که مولفه ی آن یکسان است
  سطر، ستون، مقدار همه مولفه ها (اگر چیزی ننویسیم صفر در نظر گرفته می شود)

- r:= seq($\frac{1}{i}$, i=1..10)

UNOMAX

- rb:= [r]

Subject
Date

- a := Matrix (2, 5, rb) این ماتریس مقادیرش با استفاده از دنباله ۲
که با استفاده از rb به یک لیست تبدیل شده بدست می آید.
← * تعداد جملات دنباله باید با تعداد مولفه های ماتریس برابر باشد

- evalm (A1 + B1) برای جمع دو ماتریس به کار می رود
A1: یک ماتریس m×n ، B1: یک ماتریس m×n

- evalm (5 × A1) برای ضرب عدد ثابت در مولفه های یک ماتریس

- evalm (A1 &* A2) برای ضرب دو ماتریس در هم
A1: ماتریس x×m ، &: and ، *: ستاره ، A2: ماتریس m×y
(تعداد ستون های چپی = سطر های راستی)

- S1 := seq(i², i = 1..3)
  S2 := seq(2i, i = 1..3)
  S3 := seq(i, i = 1..3)

- A5 := <<S1>|<S2>|<S3>> ایجاد یک ماتریس که ستون های
مولفه هایش بر اساس ۳ دنباله S1 و S2 و S3 به وجود می آید

- transpose (A5) این دستور ترانهاده A5 را می دهد، (جای سطر ها و ستون ها
را عوض می کند

- det (A7) دترمینان یک ماتریس را می دهد (ماتریس A7 باید مربعی باشد)

- trace (A7) trace یک ماتریس را می دهد (ماتریس A7 باید مربعی باشد)
← trace به معنی جمع مقادیر قطر اصلی است [╲]
← (trace = اثر ماتریس)

Subject
Date

- S:= inverse(A7)   وارون یک ماتریس را می دهد

- s:= inverse(A7)

  r:= det(A7)

  evalm(s &* r) = ماتریس قطری [1 ... 1] به اندازه A7

- simplify(evalm(s &* r)) = ماتریس قطری [1 ... 1] به اندازه A7

- مقادیر ویژه و بردار های ویژه:

  λ مقدار ویژه ماتریس A است $\lambda a = A a$

  a بردار ویژه A متناظر با مقدار ویژه λ است

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$\lambda \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$\lambda a_1 = 1 a_1 \longrightarrow \lambda = 1$$
$$\lambda a_2 = 4 a_2 \longrightarrow \lambda_2 = 4$$
$$\lambda a_3 = 3 a_3 \longrightarrow \lambda_3 = 3$$

- eigenvalues(A)

- eigenvectors(A)

- a:= Array([0, 0, 0])
  b:= [0, 0, 0]
  } برای تعریف بردار

- 3a   برای ضرب یک عدد ثابت در بردار

- matadd(a, b, 3, 4)   برای جمع دو بردار که هرکدام ضریب هم دارند

  $2 = 3a + 4b$

Subject
Date

* اگر Limit را limit بنویسیم ، مقدار حد را بدون نیاز به Value به ما می دهد

- $f(x) := \frac{\sin x}{x}$

A := Limit (f(x) , x = 0)

A' := Value (A)

حد یک تابع را در یک نقطه نشان می دهد.
Value گاهی اوقات عبارت را ساده نمی کند
مثلا e^4 نمایش می دهد. برای دیدن مقدار عددی باید از Value ، evalf بگیریم

- $g(x) := \begin{cases} e^{x^2} & x < -2 \\ x^4 & o.w \end{cases}$

g(x) := Piecewise(x<-2 , exp(x^2) , x^4)

B := Limit (g(x) , x = -2)

B := Value(B)

این دستور حد یک تابع چند ضابطه ای را بررسی می کند . در این مورد ، نقطه x = -2 را بررسی می کنیم که حد ندارد . حاصل undefined را نمایش می دهد

- C := Limit(g(x) , x = -2 , left)

C := value (c)

D := Limit (g(x) , x = -2 , right)

D := value (D)

این دستور حد چپ و راست را نمایش می دهد ، برای اثبات حد نداشتن یک نقطه حد چپ و راست را بدست می آوریم و نشان می دهیم برابر نیستند

- evalb (a = b)

با True و False نشان می دهد a و b برابر هستند یا نه

- Limit (sin(1/x) , x = infinity)

می توان تابع مورد نظر را در خود دستور حد تعریف کرد

- Limit (cos(x) , x = infinity) ⟶ -1 .. 1

نتیجه می دهد : که تابع در این نقطه حد مشخصی ندارد بخاطر همین ، برنامه نتیجه گنگی می دهد

Subject
Date

- $f(x) := x^5 - 3x$ — نحوه یافتن مشتق با استفاده از حد

$Z := \text{limit}\left(\frac{f(x+h) - f(x)}{h}, h=0\right)$

- $\text{limit}\left(\frac{f(2+h)-f(2)}{h}, h=0\right)$ — برای یافتن مشتق در $x=2$

- $A := \text{seq}\left(\frac{1}{x}, x=1..20\right)$ — برای یافتن مشتق در چندین نقطه (مانند یک دنباله (ex: $\frac{1}{1}$)

$B := \text{Seq}(z, x \text{ in } A)$

B مشتق را به‌ازای x مساوی با ۱ و $\frac{1}{2}$ و $\frac{1}{3}$ و ... می دهد

- $\text{subs}(x=2, z)$ — این دستور مشتق را در نقطه $x=2$ می دهد

- $A := \text{seq}\left(\frac{1}{x}, x=1..20\right)$ — برای یافتن مشتق در چند ین نقطه از این روش هم می توان استفاده کرد

$F := \text{unapply}(z, x)$

$\text{map}(F, [A])$

*) دستور unapply چند جمله ای z را به تابعی با متغیر x تبدیل می کند)

- $f(x) := x^2+1$ — دستور مستقیم برای گرفتن مشتق

~~diff~~

$\text{diff}(f(x), x)$

- $\text{diff}(\sin(x), x)$ — دستور مستقیم برای گرفتن مشتق (در اینجا $\cos x$ می دهد)

- $\text{diff}(\sin(x), x, x)$ — برای گرفتن مشتق دوم

- $\text{dif}(\cos(xy), x, y)$ — اول نسبت به x مشتق می گیرد و سپس از آن مشتق نسبت به y مشتق می گیرد

LINOMAX

CamScanner

امیرحسین کریم‌زادگان

- diff (sin(x) , x $10) — برای اینکه چندین بار نسبت به متغیری مشتق بگیریم از $ استفاده می‌کنیم (در اینجا از عبارت 10 بار مشتق گرفتیم)

- diff (z.x^3.y^4 - y^z , z) — مشتق برحسب z عبارت را می‌دهد
  * برای مشتق گرفتن برحسب z از y^z باید y^z را این طوری ببینیم: $(y^z = e^{z \ln y})$

- f(x) := x^3 - 3x^2 + 2 — این دستور یک تابع رسم می‌کند و همین طور خط مماس بر تابع در نقطه $x=6$
  A := diff (f(x) , x)
  B := unapply (A , x)
  m := B(6)
  y := m.(x-6) + f(6)
  with (plots):
  plot ([f(x), y])

- P:= taylor(f(x) , x=a , k) — این دستور بسط تیلور تابع f(x) را می‌دهد
  f(x): تابع
  k: جملات بسط که می‌خواهیم نمایش داده شوند

- taylor (sin(x) , x=a , 3) — بسط تیلور همراه با تقریب در انتها

- mtaylor (sin(x), x=a , 3) — بسط تیلور بدون تقریب در انتها

- limit $\left(\dfrac{f(x) - P}{(x-a)^k} \right)$ — اگر این حد نزدیک صفر شود یعنی بسط تیلور ما دقیق است
  * (متغیر ها ی [P= خود بسط]
  ([k= تعداد جملات] [a = چیزی که x مساوی آن است])

LNOMAX

Subject
Date

k=2 ?

- $f(x) := (x^3 - 3x^2 + 2)$ — برای رسم مماس بر منحنی در هر نقطه دلخواه $x=a$

A := mtaylor(f(x), x=a, 2) — می توانیم از بسط تیلور در مرتبه

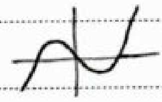
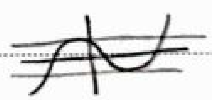
plot([f(x), A]) — اول یا همان k=2 استفاده کنیم

(* مماس در هر نقطه ای را که می خواهیم
باید به جای a آن عدد را قرار دهیم)

- $r(x) := x^3 - 50x$

C := solve(diff(r(x), x) = 0, x)

plot({ r(x), seq(mtaylor(r(x), x=a, 1), a in C) })

این دستور در تابعی مانند  مماس هایی موازی با محور x ها، رسم شوند
به شکل 

- with(Student[Calculus1]): — برای یافتن انتگرال تقریبی:

ApproximateInt(x^2+10, x=1..5) — انتگرال تقریبی می دهد

ApproximateInt(x^2+10, x=1..5, output = sum) — حاصل را به شکل Σ می دهد

ApproximateInt(x^2+10, x=1..5, output = plot) — با استفاده از نمودار، مقدار انتگرال را تقریب می زند

ApproximateInt(x^2+10, x=1..5, method = midpoint) — روش تقریب زنی را می توان انتخاب کرد
newton cotes[5]
boole
simpson
⋮