

SỬ DỤNG MAPLE

TRONG HỌC TẬP, NGHIÊN CỨU VÀ GIẢNG DẠY TOÁN

Nguyễn Chánh Tú

Khoa Toán, ĐHSP Huế

Chương 1. Giới thiệu về phần mềm Maple

Maple là một hệ thống tính toán trên các biểu thức đại số và minh họa toán học mạnh mẽ của công ty Waterloo Maple Inc. (<http://www.maplesoft.com>), ra đời khoảng năm 1991, đến nay đã phát triển đến phiên bản 10. Maple có cách cài đặt đơn giản, chạy trên tất cả các hệ điều hành, có cấu trúc linh hoạt để sử dụng tối ưu cấu hình máy và đặc biệt có trình trợ giúp (**Help**) rất dễ sử dụng. Từ phiên bản 7, Maple cung cấp ngày càng nhiều các công cụ trực quan, các gói lệnh tự học gắn liền với toán phổ thông và đại học. Ưu điểm đó làm cho nhiều nước trên thế giới lựa chọn sử dụng Maple cùng các phần mềm toán học khác trong dạy học toán trước đòi hỏi của thực tiễn và sự phát triển của giáo dục.

1.1. Các tính năng cơ bản của Maple.

Có thể nêu vắn tắt các chức năng cơ bản của Maple như sau:

- là một hệ thống tính toán trên các biểu thức đại số;
- có thể thực hiện được hầu hết các phép toán cơ bản trong chương trình toán đại học và phổ thông;
- cung cấp các công cụ minh họa hình học thuận tiện gồm: vẽ đồ thị tĩnh và động của các đường và mặt được cho bởi các hàm tùy ý trong nhiều hệ tọa độ khác nhau;
- một ngôn ngữ lập trình đơn giản và mạnh mẽ có khả năng tương tác với các ngôn ngữ lập trình khác;
- cho phép trích xuất ra các định dạng khác nhau như LaTeX, Word, HTML,...
- Một công cụ biên soạn giáo án và bài giảng điện tử, thích hợp với các lớp học tương tác trực tiếp;
- một trợ giáo hữu ích cho học sinh và sinh viên trong việc tự học;

1.2. Cấu trúc và giao diện.

Cấu trúc tài nguyên của Maple

- Khi khởi động Maple, chương trình chỉ tự động kích hoạt nhân của Maple bao gồm các phép toán và chức năng cơ bản nhất. Phần nhân chiếm khoảng 10% dung lượng của toàn chương trình.
- Các dữ liệu và chương trình còn lại của Maple được lưu giữ trong thư viện Maple và được chia ra 2 nhóm: nhóm các lệnh cơ bản và nhóm các gói lệnh. Maple 9.0 có khoảng 85 gói lệnh. Gói lệnh có thể nạp vào bằng:

> **with(plots):**

Lệnh của Maple

- Lệnh được gõ vào trang làm việc (worksheet) tại dấu nhắc lệnh ">" và theo ngầm định được hiển thị bằng font Courier màu đỏ. Một lệnh được kết thúc bởi dấu ":" hoặc dấu ";" và được ra lệnh thực hiện bằng việc nhấn Enter khi con trỏ đang ở trên dòng lệnh.

> **factor(2*x^102+x^100-2*x^3-x+60*x^2+30):**

>

- Kết quả của lệnh được hiển thị ngay bên dưới dòng lệnh nếu dùng dấu ";" . Có thể dễ dàng dùng chuột và bàn phím để thực hiện các chức năng bôi đen, copy, paste, cut, delete...đối với dữ liệu trên dòng lệnh hay kết quả thực hiện.

Sử dụng dịch vụ trợ giúp (Help) trong Maple

Maple có dịch vụ trợ giúp khá đầy đủ và thuận lợi bao gồm **cú pháp**, giải thích **cách dùng** và các **ví dụ** đi kèm. Để nhận được trợ giúp, có thể:

- Nếu đã biết tên lệnh thì từ dấu nhắc gõ vào

> **?factor**

- Nếu dùng một gói lệnh thì khi nạp gói lệnh, Maple sẽ hiển thị toàn bộ lệnh trong gói đó.
- Một cách thông dụng nữa là dùng trình **Help|Topic Search** rồi gõ vào từ khóa cần tìm.

1.3. Lưu giữ và trích xuất dữ liệu.

- Trang làm việc của Maple sẽ được lưu giữ bằng file có đuôi ".mws". File được lưu giữ bằng trình **File|Save**. Một file đã có được mở bằng **File|Open**.
- Ngoài việc lưu giữ bằng định dạng của Maple như trên, dữ liệu có thể được trích xuất thành các định dạng khác như Word, LaTeX hay HTML. Trích xuất bằng **File|Export**.

1.4. Các môi trường làm việc trong Maple

Maple có 2 môi trường làm việc là **toán** và **văn bản**. Sau khi khởi động, Maple tự động bật môi trường toán. Muốn chuyển sang môi trường văn bản, kích chuột vào biểu tượng **T** trên thanh công cụ hay vào trình **Insert->Text**. Ngược lại, từ môi trường văn bản, kích chuột vào dấu "[>" trên thanh công cụ hay vào **Insert** để chuyển sang môi trường toán.

> **ifactor(58600);**

(2)³ (5)² (293)

>

Chương 2. Sơ lược về các tính toán số học, đại số và giải tích trong Maple

2.1 Các dấu phép toán, hàm và hằng số cơ bản

Các phép toán và dấu phép toán

<u>Cú pháp</u>	<u>Giải thích</u>	<u>Ví dụ</u>
!	giai thừa	100!
^	lũy thừa	a^5
+	cộng	a+b
-	trừ hoặc số âm	x-y
*	nhân	2*x
/	chia	120/5
<	nhỏ hơn	a<100
>	lớn hơn	b>100
>=	lớn hơn hoặc bằng	x>=1/2
<=	nhỏ hơn hoặc bằng	x<=1/2

=	bằng	a=b
:=	phép gán	x:=2/3

Các hàm thông dụng

<u>Cú pháp</u>	<u>Giải thích</u>	<u>Ví dụ</u>
sin, cos, tan	các hàm lượng giác	sin(x)
arcsin, arccos, arctan	các hàm LG ngược	arcsin(x)
abs	hàm trị tuyệt đối	abs(x)
exp	hàm mũ cơ số e	exp(x) hay E^x
log hay ln	hàm logarit cơ số e	log(x) hay ln(x)
log[10]	hàm logarit cơ số 10	log[10](x)
sqrt	khai căn bậc 2	sqrt(3)

Các hằng số thông dụng

<u>Cú pháp</u>	<u>Hằng số</u>
Pi	π
exp(1)	e
infinity	∞

>

2.1. Các tính toán số học

a) Maple có thể làm việc như một máy tính bỏ túi hiện đại

> 5*3;

> 120/7+2^100;

Khả năng tính toán số học của Maple là rất lớn, có thể làm việc với những số có đến $2^{28} = 268435456$ chữ số.

> 300!;

```
30605751221644063603537046129726862938858880417357699941677674125947
6533176716867465515291422477573349939147888701726368864263907759
0315422684292790697455984122547693027195460400801221577625217685
2559653569035067887252643218962642993652045764488303889097539434
9625436053225980776521270822437639449120128678675368305712293681
4364995646049816645022771650018517654646934011222603472972406633
2585835068701501697941688503537521375549102891264071571548302822
4937952636580145235233156936482233436799254594095276820608062232
123873838808170496000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
```

> length(%);

615

Ta thấy số 300.000! có 1.512.852 chữ số, khoảng 20 ngàn dòng trên màn hình.

> ifactor(1512852);

$(2)^2$ (3) (11) (73) (157)

> **FermatPrime:=2^(2^n)+1;**

FermatPrime := $2^{(2^n)} + 1$

> **[seq(FermatPrime,n=1..6)];**

[5, 17, 257, 65537, 4294967297, 18446744073709551617]

> **map(ifact, %);**

[(5), (17), (257), (65537), (641) (6700417), (67280421310721) (274177)]

b) Các hàm trên số nguyên

> **isprime(1388990297):**

> **nextprime(123456789):**

> **prevprime(123456789):**

> **ilcm(786,120):**

> **igcd(786,120):**

> **irem(786,120):**

> **iquo(786,120):**

>

Ngoài ra còn có các lệnh sau:

max, min	Tìm số lớn nhất và nhỏ nhất trong tập các số cho trước.
iroot	Tìm nghiệm nguyên xấp xỉ căn bậc n của 1 số nguyên.
isqrt	Tìm nghiệm nguyên xấp xỉ căn bậc 2 của 1 số nguyên.
mod	Các phép toán trên hệ thặng dư modulo.
rsolve	Giải phương trình hàm nhờ các công thức truy hồi.
convert	Chuyển đổi số nguyên sang các hệ cơ số khác nhau.

c) Tính toán chính xác và gần đúng

- Khi làm việc với số hữu tỷ hoặc căn thức, Maple có khả năng tính toán với kết quả chính xác. Điều này hết sức quan trọng khi cần các tính toán nhiều bước.

> **A:=7/3+6^10/7;**

> **3^(2/5);**

> **Pi;**

- Tuy nhiên, khi cần Maple cũng có thể tính gần đúng với độ chính xác tùy ý.

> **evalf(10/3);**

> **B:=7.0/3+6^10/7;**

> **evalf(3^(2/5),20);**

> **evalf(Pi,100);**

- Maple làm việc thuận lợi trên các số phức:

> **(2+2*I)/(1-3*I);**

> **sqrt(1+I);**

> **evalf(%);**

d) Tính tổng, tích hữu hạn và vô hạn

3.1. Tính tổng hữu hạn.

Ví dụ: Tính tổng $\sum_{i=1}^{100} \frac{1+i}{1+i^4}$. Ta có thể hoặc là dùng 2 lệnh'

```
> Sum((1+i)/(1+i^4),i=1..20);  
> value(%);
```

Hoặc tính trực tiếp:

```
> sum((1+i)/(1+i^4),i=1..20);
```

3.2. Tính tổng vô hạn.

Ví dụ: Tính tổng $\sum_{k=1}^{\infty} \frac{1}{k^2}$. Tương tự như trên, ta có thể dùng

```
> Sum(1/k^2,k=1..infinity);  
> value(%);
```

Hoặc

```
> sum(1/k^2,k=1..infinity);  
>
```

Hoàn toàn tương tự với tính tổng, ta có thể tính các tích hữu hạn và vô hạn với Maple. Cách làm như trên với việc thay lệnh Sum hay sum bởi Product hay product.

2.3. Các tính toán đại số

2.3.1. Các tính toán trên biểu thức đại số

Gán tên cho biểu thức và trị cho biến

Dùng phép ":" để gán tên và lệnh "subs" để gán trị cho biến.

```
> A:=a*x^2+b*x+c;  
> A1:=subs(a=1,b=2,c=1,A);
```

Biến đổi biểu thức đại số

Lệnh khai triển với expand

```
> B:=(x+1)*(x-2)+3*x+2;  
> expand(A);  
> expand(sin(x+y));  
> L:=exp(a+ln(b));  
> expand(%);
```

Lệnh đơn giản biểu thức với simplify

```
> C:=cos(x)^5+sin(x)^4+2*cos(x)^2-2*sin(x)^2-cos(2*x);  
> simplify(C);  
> simplify(sin(x)^2+ln(3*x)+cos(x)^2);  
> simplify(sin(x)^2+ln(3*x)+cos(x)^2,'trig');  
> simplify(sin(x)^2+ln(3*x)+cos(x)^2,'ln');
```

Lưu ý: Lệnh simplify là một lệnh rất "mơ hồ" do không có một tiêu chuẩn rõ ràng cho sự đơn giản hóa. Nhiều khi ưu tiên của Maple trong việc đơn giản một biểu thức không giống như kỳ vọng của người dùng. Hơn thế nó cần rất nhiều bộ nhớ để simplify. Trong đa số trường hợp, lệnh expand là một lệnh đơn giản tốt hơn.

```
> ?seq
```

Ngược lại với expand là lệnh factor và combine

```
> expand((x-2)*(x+3));
> factor(%);
> expand(e^(2*x+y)+sin(2*x));
> combine(%);
```

Lệnh chuẩn hóa với normal, đặc biệt dùng để đơn giản các phân thức về dạng chuẩn tắc

```
> PT:=(x^2-y^2)/(x^3+y^3+3*x^2*y+3*x*y^2);
> normal(PT);
> factor(PT);
```

Lệnh convert cho phép chuyển các biểu thức về các dạng biểu diễn khác nhau

```
> convert(sin(x),exp);
> M:=Matrix([[a,b],[x,y]]);
> convert(M,'listlist');
> convert(M,set);
> convert(%,list);
```

Lệnh map cho phép gán lệnh đồng thời cho nhiều biến trong bảng hay tập.

```
> bang:=[Pi/3,0,-Pi/2];
> map(sin,bang);
> f:=x->x^2+x-1;
> map(f,{-1,0,1});
> map(f,[-1,0,1]);
```

chú ý sự khác nhau của hai kết quả trên.

Tính toán trên đa thức

Các lệnh thông dụng

Lệnh sắp xếp đa thức với sort và collect

```
> f:=x^2+1+3*x+4*x^4-3*x^5;
> sort(f,x);
> g:=y^3+x^2*y^2+a*x^3;
> sort(g);
> sort(g,[x,y]);
> sort(g,[y,x]);
> sort(g,[y,x],plex);
> h:=x*y+x*y*z-3*y*z^2+x+z*x+y^3+x^4;
> collect(h,x);
> collect(h,y);
> collect(h,z);
```

Xác định hệ số và bậc với lệnh coeff và degree

```
> h:=t^4-4*x^3*y*z+y^3-3*y*z^2+x*y+x*z+x;
> sort(h);
> degree(h);
```

```
> degree(h,y);
> ldegree(h);
> ldegree(h,x);
> coeff(h,z);
> coeff(h,z,2);
> coeffs(h);
> lcoeff(h,t);
```

Các lệnh cho phép chia đa thức với rem, quo và divide

```
> r:=rem(x^4+2*x^2+x-1,x^2+2*x+3,x);
> r:=quo(x^4+2*x^2+x-1,x^2+2*x+3,x);
> divide(x^4+2*x^2+x-1,x^2+2*x+3);
> divide(x^2-y^2,x-y);
> ?gcdex
```

Phân tích đa thức thành nhân tử với lệnh factor

```
>
```

Lệnh factor phân tích đa thức thành nhân tử trên trường sinh ra bởi trường số hữu tỷ và các hệ số của đa thức.

```
> factor(x^3+3);
> factor(x^3+3,3^(1/3));
> factor(x^3+3,((-3)^(1/2),3^(1/3)));
```

Tham số của lệnh factor có thể là real hay complex nếu như muốn phân tích thành nhân tử trên trường số thực hay phức. Chú ý rằng khi đó kết quả cho hệ số là các số gần đúng.

```
> factor(x^3+3.0);
> factor(x^3+3,real);
> factor(x^3+3,complex);
```

Giải phương trình, bất phương trình và hệ phương trình

Với lệnh solve và fsolve

```
> restart;
> equs:={x+2*y-8,x^2-2*x-3};
> equ:=x^3+x^2-x;
> sol1:=solve(equ);
> sol:=solve(equ,{x});
Có thể kiểm tra lại kết quả bằng lệnh eval:
> sol[1];sol[2];sol[3];
> expand(subs(x=sol1[2],equ));
> expand(eval(equ,x=sol1[2]));
> seq(expand(eval(equ,sol[i])),i=1..3);
```

Hãy thực hiện việc thử lại với sol1 trên.

KQ:

Muốn thử lại với sol1, cần dùng

```
> eval(equ,x=sol1[1]);
```

Có thể trích xuất các nghiệm thành bộ theo thứ tự định trước với các nghiệm nhiều thành phần.

```
> equs;
```

```
> sols:=solve(equs,{x,y});
> x1:=eval(x,sols[2]);
> seq(eval(x,sols[i]),i=1..2);
```

Cũng có thể trích xuất các nghiệm thành bộ theo thứ tự định trước.

```
> eval([x,y],sols[2]);
> eval([y,x],sols[2]);
```

Cách khác để kiểm tra nghiệm là dùng lệnh `map`:

```
> map(subs,[sols[1],sols[2]],equ);
```

Chú ý rằng lệnh `map` không thể tác động trên dữ liệu có định dạng tập hợp:

```
> map(subs,sol,equ);
> map(subs,[sol],equ);
```

Hãy thử dùng lệnh `map` với nghiệm của `equ` cho trên.

Lệnh `solve` có thể cho biết tất cả các nghiệm chính xác của đa thức 1 biến có bậc không quá 4, tức là nghiệm biểu diễn bằng căn thức.

```
> solve(x^3+2*x+8);
```

Tuy nhiên, với phương trình bậc 4, do nghiệm bằng căn thức quá phức tạp và không có tính ứng dụng cao, Maple ngầm định đưa ra nghiệm dưới dạng `RootOf`. Tuy nhiên ta có thể nhận được nghiệm chính xác bằng cách chọn `_EnvExplicit:=true`.

```
> solve(x^4+x^3-9);
> evalf(%);
> _EnvExplicit:=true;
> solve(x^4+x^3-9);
```

Với các đa thức 1 biến có bậc cao hơn 5, lệnh `solve` cho nghiệm `RootOf` và có thể nhận được nghiệm xấp xỉ bằng `evalf`.

```
> solve(x^7+x^5+x^2+x+1);
> evalf(%,20);
```

Lệnh `solve` để giải các phương trình siêu việt hoặc chứa căn thức.

```
> solve(2*cos(x)^10-x+1,{x});
> evalf(%);
> solve(ln(x)+x+1,x);
> evalf(%);
```

Khi giải các phương trình phức tạp, Maple thường chỉ cho ta 1 nghiệm. Cần dùng các đánh giá khác để có thể tìm ra các nghiệm khác, hoặc có thể dùng lệnh `fsolve`. Đây là lệnh tìm nghiệm xấp xỉ, và có thể tìm nghiệm với các điều kiện hạn chế.

```
> f:=3*2^x+2*3^x-5^x-1;
> sol:=solve(f,x);
> fsolve(f);
> evalf(sol);
> fsolve({f=0},{x},-4..0);
```

Lệnh `solve` có thể dùng để giải hệ phương trình và bất phương trình.

```
> pts:={x+y+2*t-1,3*x+2*y+3-t,x+y-t};
> solve(pts,{x,y,t});
> solve(x^2+3*x+1>0,x);
> pts1:={x+y+2*t-1,3*x+2*y+3-t,x+y-t>0};
> solve(pts1,{x,y,t});
```

Hạn chế của lệnh solve

- Không thể tìm nghiệm 1 cách triệt để, nhất là với các hàm siêu việt hoặc phức tạp. Cần sử dụng kết hợp với đồ thị và các phương pháp đánh giá khác để tìm hết tất cả các nghiệm.
- Cần luôn luôn kiểm tra lại nghiệm bằng cách dùng lệnh eval.

```
> equ:=(x-2)^2/(x^2-4);
> sol:=solve(equ,{x});
> eval(equ,sol);
```

Các lệnh tìm nghiệm khác

Lệnh isolve để tìm nghiệm nguyên

```
> isolve(2*x+3*y);
> isolve(2*x-3);
```

Lệnh msolve để tìm nghiệm nguyên modulo p

```
> msolve({2*x-7*y+1,3*x-7*y-5},9);
```

So sánh với:

```
> solve({2*x-7*y+1,3*x-7*y-5});
> msolve(2^n=3,11);
> solve(2^n=3);
```

Lệnh rsolve để giải các bài toán truy hồi

```
> restart;
> rsolve({f(n)=f(n-1)+f(n-2),f(0)=1,f(1)=1},{f(n)});
```

Đọc giả quan tâm có thể tìm hiểu thêm các lệnh nâng cao về vấn đề này trong gói lệnh LREtools.

Lệnh dsolve để giải các phương trình vi phân (xem phần tính toán của Maple với giải tích)

```
> ?polytools
```

2.3.2. Đại số tuyến tính với các gói lệnh linalg và LinearAlgebra

Từ version 7 trở đi, Maple cung cấp thêm 1 gói lệnh LinearAlgebra bên cạnh gói lệnh linalg đã có trước đây. Hầu như tất cả các chức năng trong linalg đều được tích hợp lại trong LinearAlgebra với cú pháp tương tự, cộng thêm một số chức năng mới.

Ma trận và các phép toán cơ bản

Tạo véc tơ, ma trận với các lệnh vector/Vector, matrix/Matrix và array/Array

```
> restart;
> u:=vector([a,b]);
> U1:=Vector([a,b]);
> U2:=Vector[row]([a,b]);
> A0:=matrix([[a[11],a[12],a[13]], [a[21],a[22],a[23]]]) ;
> B0:=Matrix([[b[11],b[12],b[13]], [b[21],b[22],b[23]]]) ;
> A0[1,2];B0[1,2];
> a[11]:=0;b[11]:=0;
> A0;B0;
> evalm(A0);
> A0[1,1]:=0;
```

```

> A:=matrix(3,2,[2,1,2,-2,a,b]);
> A1:=matrix([[2,1],[2,-2],[a,b]]);
> B:=Matrix([[2,1],[2,-2],[a,b]]);
> B1:=<<2,2,a>|<1,-2,b>>;
> whattype(A);whattype(B);
> LinearAlgebra[Equal](B,B1);
> linalg[equal](A,B1);

```

Ma trận cũng có thể tạo nên bởi lệnh array/Array. Thực ra lệnh matrix mà một dạng dữ liệu đặc biệt của array. Tuy nhiên điều đó không đúng với Matrix.

```

> T:=array([[2,1],[2,-2],[a,b]]);
> type(T,matrix);
> T1:=Array([[2,1],[2,-2],[a,b]]);
> type(T1,Matrix);
> B+T1;

```

Có thể dùng lệnh convert để chuyển các dạng về ma trận

```

> A;B;
> B2:=convert(A,Matrix);
> whattype(B2);
> LL:=[[2,1],[2,-2],[a,b]];
> whattype(LL);
> convert(LL,matrix);whattype(%);
> convert(LL,Matrix);whattype(%);

```

Các ma trận được xây dựng đặc biệt

```

> matrix(1,2,4);Matrix(1,2,3);
> matrix(1,2);Matrix(1,2);Matrix(3);
> M:=Matrix(3,3,shape=identity);

```

Rất nhiều tùy chọn mới cho lệnh Matrix. Dùng lệnh ?Matrix nếu muốn tìm hiểu thêm.

```

> f:=(i,j)->x^(i+j+1);
> matrix(3,2,f);

```

Truy xuất các phần tử của ma trận

```

> B:=Matrix([[2,1],[2,-2],[a,b]]);
> B[3,1];

```

```

> convert(B,Vector[row]);
> convert(B,set);
> C:=Matrix(5,(i,j)->2*i+j);

```

Để trích xuất một phần hay toàn bộ một dòng hay cột, ta dùng:

```

> C[5,1..3], C[3..5,2];C[2,1..-1];C[1..-1,4];

```

Để trích xuất một ma trận con, thay thế các tọa độ bằng các khoảng:

```

> C[3..5,1..3];

```

Các phép toán trên ma trận

```
> restart;
> with(LinearAlgebra):
> A:=Matrix([[a,1,c],[2,-b,1]]);
> B:=Matrix([[x,y,1],[c,2,z]]);
```

- **Phép cộng ma trận**

```
> A+B;
> Add(A,B);
> A-B;
```

- **Phép cộng ma trận với 1 số** được xem như cộng với một ma trận hằng:

```
> 2+A;
```

- **Phép nhân vô hướng**

```
> 3*A;
> 3 . A;
```

Nếu muốn dùng phép nhân bằng "." thì phải có khoảng trống sau phép toán.

- **Phép nhân hai ma trận**

```
> C:=RandomMatrix(3,2);
> C.A;
```

Chú ý không cần khoảng trống sau phép toán "." .

Các phép biến đổi sơ cấp và phép chuyển vị

```
> with(LinearAlgebra):
A := <<1,2,3>|<4,5,6>|<7,8,9>>;
> RowOperation(A, 3, 3);
> ColumnOperation(A, [1,3]);
> RowOperation(A, [1,3], -2);
Dòng thứ 1 được thay bởi ( $d1-2*d3$ ).
> A:=RandomMatrix(3,2);
> Transpose(A);
```

Định thức và ma trận nghịch đảo

```
> restart;with(LinearAlgebra):
> A:= Matrix([[b[11],b[12],b[13]],[b[21],b[22],b[23]],[b[31],b[32],b[33]]]);
> Determinant(A);
>
> C:=RandomMatrix(3,3);
> MatrixInverse(C);
Đa thức đặc trưng. Giá trị riêng và véc tơ riêng
> with(LinearAlgebra):
M := <<-2,0,0>|<1,0,0>|<0,3,2>>;
```

```
> CharacteristicPolynomial(M,x);
> solve(%,x);
> Eigenvalues(M,output='list');
> Eigenvectors(M);
```

Các cột của ma trận bên phải là các véc tơ riêng ứng với các giá trị riêng cho trong cột bên trái.

Các dạng chuẩn tắc

Dạng rút gọn Gauss và Gauss-Jordan

Đây là dạng chuẩn tắc ứng dụng trong khi giải hệ phương trình tuyến

```
> restart;with(LinearAlgebra):
> A:=RandomMatrix(4,6);
> b:=<1,0,-2,3>;
> GaussianElimination(A);
> GaussianElimination(A,'method'='FractionFree');
> ReducedRowEchelonForm(<A|b>);
```

Dạng chuẩn tắc Jordan

```
> with(LinearAlgebra):
A := <<0,-2,-2,-2>|<-3,1,1,-3>|<1,-1,-1,1>|<2,2,2,4>>;
J := JordanForm(A);
> Eigenvalues(A, output='list');
> Q := JordanForm(A, output='Q');
> Q^(-1) . A . Q;
>
```

Dạng chuẩn tắc hữu tỷ

```
> restart;
> with(LinearAlgebra):
A := <<0,1,1,1,1>|<2,-2,0,-2,-4>|<0,0,1,1,3>|<-6,0,-3,-1,-3>|<2,2,2,2,4>>;
> FrobeniusForm(A);
> factor( CharacteristicPolynomial(A, x) );
> (F,Q):=RationalCanonicalForm(A,output=['F','Q']);
> FrobeniusForm(A,output='Q');
> Q^(-1) . A . Q;
```

Chiều và cơ sở của không gian véc tơ

Bài tập

Cho hệ pttt:

$$\begin{aligned} [-x_1 + 2x_2 + 6x_3 - 8x_4 - 14x_5 + 3x_6 = 0, & 2x_1 + 4x_2 + x_3 - 8x_4 + 5x_5 - x_6 = 0, \\ -3x_1 + x_2 + 4x_3 - 9x_4 - 10x_5 = 0, & 3x_1 - 2x_2 - x_3 + 12x_4 - x_5 + 4x_6 = 0, \\ 5x_1 + 7x_2 + 11x_3 - 11x_4 - 19x_5 + 9x_6 = 0] \end{aligned}$$

- 1) Hãy lập ma trận hệ số của hệ.
- 2) Chứng tỏ rằng ma trận trên có thể nhận các cột của x_1, x_2, x_3 làm cơ sở củ không gian nghiệm
- 3) Biểu thị tuyến tính x_4, x_5, x_6 qua x_1, x_2, x_3 .
- 4) Giải hệ phương trình trên.

```

> restart;
> with(LinearAlgebra):A := <<1,2,4>|<0,2,2>|<1,0,2>|<0,0,0>>;
> RowSpace(A);
> GramSchmidt(%);
> ColumnSpace(A);

> B :=
Matrix(5,6,[[ -1,2,6,-8,-14,3],[2,4,1,-8,5,-1],[-3,1,4,-9,-10,0],[3,-2,-1,12,-1,4],[5,7,11,-11,-19,9]]);
> GenerateEquations(B,[x1,x2,x3,x4,x5,x6]);
>
> Rank(B);
> ColumnSpace(B);
> RedB:=ReducedRowEchelonForm(B);
> B1:=DeleteColumn(B,4..6);Rank(B1);
> B2:=ColumnSpace(B);
> v1:=B1[1..-1,1];v2:=B1[1..-1,2];v3:=B1[1..-1,3];v4:=B[1..-1,4];v5:=B[1..-1,5];v6:=B[1..-1,6];
> <<LinearSolve(B1,v4)>|<LinearSolve(B1,v5)>|<LinearSolve(B1,v6)>>;
> NullSpace(B);

```

Hệ phương trình tuyến tính

```

> restart;
> with(LinearAlgebra):
> sys := [ 3*x[1]+2*x[2]+3*x[3]-2*x[4] = 1,
           x[1]+ x[2]+ x[3]      = 3,
           x[1]+2*x[2]+ x[3]- x[4] = 2 ];var:=[x[1],x[2],x[3],x[4]];
>
(A, b) := GenerateMatrix(sys,var);
> A1:=GenerateMatrix(sys,var,augmented=true);
> LinearSolve(A1);
> LinearSolve(A,b);
> GenerateEquations(A,[x1,x2,x3,x4],<1, -1,0>);
> GenerateEquations(A1,[x1,x2,x3,x4]);

```

2.4. Các tính toán giải tích

a) Xác định hàm và lệnh map

Hàm 1 biến:

```

> f:=x->a*x^2+b*x+c;
> f(2);

```

Với hàm nhiều biến:

```

> g:=(x,y,z)->x^2+y^2-z^2;
> g(3,4,5);

```

Hàm từng đoạn có thể định nghĩa bởi lệnh `piecewise()`, cấu trúc được minh họa qua ví dụ sau:

```

> f:=piecewise(x<0,1,x<1,x,x<2,3-x,sin(x)+2);

```

Lệnh map thường dùng để xác định giá trị của 1 hàm tại một dãy các giá trị của biến

```
> f:=x->x^2-x+1;  
> lst:=[-2,-1,-1/2,0,1,2,3/2,4];  
> map(f,lst);
```

b) Các phép tính toán cơ bản

b1) Tìm giới hạn

```
> limit(exp(x),x=-infinity);  
> limit(exp(x),x=infinity);  
> Limit(exp(x),x=-infinity)=limit(exp(x),x=-infinity);;  
> limit(1/x,x=0);
```

b2) Tính đạo hàm

Lệnh diff hoặc D cho phép tính đạo hàm của hàm 1 biến hoặc đạo hàm riêng. Cú pháp được minh họa qua các ví dụ sau:

```
> diff(7*x^2 + 4*x^3, x); diff(5*x^2 - Pi*x^3, x);  
> g:=x->x^2-exp(x)+sin(x);  
> D(g);
```

b3) Tích phân

- Tính tích phân bất định bằng lệnh int(hàm, biến).

```
> int(x^2-x+2,x);  
> Int(x^2-x+2,x);  
> value(%);
```

- Tính tích phân xác định int(hàm, miền).

```
> restart;  
> f:=x->x-2*sin(x)+1;  
> int(f,-5..8);
```

Chương 3. Công cụ vẽ hình và minh họa trong Maple

3.1. Vẽ hình trong hệ tọa độ Descartes

a) Lệnh plot và plot3d để vẽ đồ thị hàm hiện và tham số

- Lệnh vẽ hình đơn giản và thông dụng nhất là plot (trong mặt phẳng) và plot3d (trong không gian 3 chiều). Các lệnh này nằm trong phần nhân của Maple. Cú pháp:

plot(f(x),x=a..b,options) và plot3d(f(x,y),x=a..b,y=c..d,options).

```
> ?plot
```

```
> plot(x*sin(3*x),x=0..2*Pi);
```

Chú ý rằng lệnh trên vẽ đồ thị hàm $y = x \sin(3x)$ với x từ 0 đến π . Tương tự lệnh sau vẽ đồ thị hàm $z = f(x, y)$ trong miền chỉ ra:

```
> plot3d(x*sin(3*y),x=-1..1,y=0..Pi);
```

Kích chuột trên hình vẽ, ta có thể quay hình vẽ để xem bằng các góc độ tùy ý. Trên thanh công cụ mới, có các tùy chọn để xem. Hãy sử dụng!

- Có thể vẽ nhiều đồ thị trong cùng một hình:

```
> plot({x*sin(3*x),x^2+2*x-4},x=-2*Pi..2*Pi);
```

Khi kích chuột trên hình vẽ, có các tùy chọn trên thanh công cụ mới. Hãy sử dụng!

```
> plot3d([x*sin(3*y),x-y],x=-1..1,y=-Pi..Pi,color=[red,blue]);
```

Với gói lệnh plots, có thể dùng lệnh display.

b) Save hình vẽ bằng các định dạng khác nhau

Hãy trở lại đồ thị:

```
> plot3d({x*sin(3*y),x-y},x=-1..1,y=-Pi..Pi);
```

Khi kích nút phải của chuột, phía dưới các công cụ điều chỉnh hình vẽ là nút Export As, cho phép ta lưu giữ hình vẽ ra các định dạng khác nhau. Xem lệnh ?plotsetup để biết thêm cách điều chỉnh hình vẽ khi save.

c) Đồ thị của hàm tham số

Có 3 dạng hàm tham số ứng với đường cong trong mặt phẳng, mặt trong không gian và đường cong trong không gian.

Trong mặt phẳng, cú pháp là:

```
plot([ f(t), g(t), t=a..b], options).
```

Ví dụ:

```
> plot([3*cos(t),sin(t),t=0..2*Pi]):
```

Đồ thị mặt phẳng trong không gian, cú pháp là:

```
plot3d([ f(s,t), g(s,t), h(s,t)], s=a..b, t=c..d, options).
```

```
> plot3d([ 2*t-3*s^2*sin(t), s*t, 2*s-3*cos(t)], s=-2..2, t=-2..2);
```

Với đường cong tham số trong không gian, dùng lệnh spacecurve trong gói lệnh plots. Cú pháp:

```
spacecurve([f(t),g(t),h(t)],t=a..b,options).
```

```
> with(plots):
```

```
> spacecurve([sin(3*t)*cos(t),sin(3*t)*sin(t),t],t=0..Pi,shading=z);
```

d) Gói lệnh plots

Với gói lệnh plots, Maple cung cấp rất nhiều công cụ cho việc vẽ các dạng đồ thị khác nhau

Lệnh pointplot và pointplot3d để vẽ từng điểm

Trong mặt phẳng:

```
> with(plots):
```

```
> pointplot([1,3],[2,4],[3,4]):
```

Hoặc:

```
> pointplot([1,3,2,4,3,4]):
```

```
> f:=n->n/(n+1);
```

```
> pointplot({seq([n,f(n)],n=1..50)},symbol=box):
```

Hãy sử dụng các tùy chọn "circle,diamond,cross" của symbol. Cũng có thể điều chỉnh cỡ của điểm bằng options symbolsize, cỡ ngầm định là 10pt.

Tương tự, trong không gian, lệnh vẽ từng điểm được dùng như sau:

```
> pointplot3d([3,2,-1],[2,3,4],[5,6,0]):
```

Ta phải thay cỡ hoặc hình dạng của điểm thì mới dễ nhìn thấy trong không gian

```
> pointplot3d([0,1,1,1,-1,2,3,0,5],symbol=box,axes=BOXED):
```

```
> pointplot3d([0,1,1,1,-1,2,3,0,5],symbol=box,symbolsize=18,axes=BOXED):
```

Chú ý Việc vẽ từng điểm cũng có thể thực hiện với lệnh plot và plot3d thông thường mà không cần phải kích hoạt gói lệnh plots. Khi đó trong options của plot và plot3d, chọn style=point.

Lệnh display để biểu diễn nhiều đồ thị trên cùng một hình

```
> with(plots):
```

```
> S:=plot3d(4-x^2-2*y^2,x=-4..4,y=-3..3):
```

```
> P:=plot3d(6-4*y,x=-4..4,y=-3..3):
> display(S):
> display(S,P):
```

Rõ nét và mịn hóa đồ thị

a) Làm rõ nét của đồ thị bằng tùy chọn `thickness`

```
> f:=x->exp(x/2);
> plot({f(x),f(-x)},x=-3..3,thickness=4):
```

Giá trị ngầm định của `thickness` là **0**. Giá trị **cao nhất** là **15**.

b) Mịn hóa bằng cách tăng giá trị của `numpoints` hay với `grid`.

```
> plot3d((x^2+y^2)/sin(x*y),x=-1..1,y=-1..1,axes=normal):
so sánh với:
> plot3d((x^2+y^2)/sin(x*y),x=-1..1,y=-1..1,axes=normal,numpoints=1000):
```

Maple dùng giá trị ngầm định là $645(\text{pt})=25 \times 25$. Khi đưa vào giá trị `n` cho `numpoints` thì

Maple sẽ tự động gán $\left[n^{\left(\frac{1}{2}\right)} \right]$ điểm cho miền xác định của từng biến.

Một cách khác để mịn hóa đồ thị là xác định giá trị cho options `grid`.

```
> plot3d(sin(x)/y^2,x=-Pi..Pi,y=-1..1,grid=[30,40]):
```

Điều chỉnh màu cho đồ thị

Màu cho đồ thị được điều chỉnh bằng options `color`

```
> with(plots):
> f:=x->x^4-4*x^3+10;
> C:=plot(f(x),x=-1..4,color=red):
> t1:=plot(f(0),x=-1..1,color=blue):
> t2:=plot(f(3),x=2..4,color=blue):
> display(C,t1,t2):
```

Dùng trình trợ giúp trong Maple để xem các màu có sẵn. Người dùng có thể định nghĩa màu mới như cách hướng dẫn trong:

```
> ?plot[color]
```

Một cách khác để điều chỉnh độ đậm của màu sắc là dùng options `shading` như ví dụ sau:

```
> plot3d(x*y,x=-2..2,y=-2..2,shading=z):
> plot3d(x*y,x=-2..2,y=-2..2):
```

Điều chỉnh tỷ lệ giữa các trục tọa độ

Khi vẽ, Maple tự động điều chỉnh tỷ lệ trên các trục tọa độ sao cho hình vẽ vừa với kích cỡ hiển thị của trang làm việc. Nghĩa là, độ dài đơn vị trên mỗi trục tọa độ không nhất thiết bằng nhau. Điều đó đôi khi làm cho hình vẽ không thật. Tùy chọn `scaling=constrained` bắt buộc Maple vẽ hình theo tỷ lệ 1:1 đối với các trục tọa độ'

```
> plot3d([cos(x)/(y^2-1)],x=0..2*Pi,y=0..2*Pi,scaling=constrained):
>
```

Các options khác: `discont`, `view`, `label`...

a) Sử dụng `discont=true` để bỏ đi các đường nối tại các điểm không liên tục.

```
> plot(tan(x),x=-Pi..Pi);
```

```
> plot(tan(x),x=-Pi..Pi,y=-4..4);
> plot(tan(x),x=-Pi..Pi,y=-4..4,discont=true);
```

b) Sử dụng view để hạn chế hình vẽ tại một phần nào đó.

```
> plot3d(2*x^2+y^2,x=-2..2,y=-3..3,style=patchcontour,axes=normal);
> plot3d(2*x^2+y^2,x=-2..2,y=-3..3,style=patchcontour,axes=normal,view=0..8);
```

c) Options label dùng để đặt tên cho các trục tọa độ.

```
> plot(-4*t^2+2*t+40.1,t=0..3,labels=["Thời gian","Nhiệt
do"],labeldirections=[horizontal,vertical],labelfont=[TIMES,BOLD,14],axesfont=[HELVETI
CA,14]);
>
>
```

e) Đồ thị hàm ẩn

Đồ thị hàm ẩn trong mặt phẳng

```
implicitplot(f(x,y),a..b,c..d,options)
```

```
> with(plots):
> implicitplot(x^2+y^2=1, x=-1..1, y=-1..1):
> p:= proc(x,y) if x<y then x^2+y^2-2 else 2*x-y-1 end if end proc:
implicitplot(p, -2..2, -1..3);
```

Có thể vẽ nhiều đồ thị với lệnh implicitplot. Ví dụ

```
> implicitplot([x^2-y^2=1, y=exp(x)], x=-Pi..Pi, y=-Pi..Pi,
color=[blue, green], legend=[plot1,plot2]);
```

Đồ thị hàm ẩn trong không gian

Cú pháp:

```
implicitplot3d(f(x,y),x=a..b,y=c..d,z=e..f,options)
```

hay

```
implicitplot3d(f(x,y),a..b,c..d,e..f,options)
```

```
> EP:=1/4*x^2+1/9*y^2-z = 0;
> HP:= 1/4*x^2-1/9*y^2-z = 0;
> EC:=1/4*x^2+1/9*y^2-z^2 = 0;
> ES:=1/4*x^2+1/9*y^2+z^2 = 1;
> H1:= 1/4*x^2+1/9*y^2-z^2 = 1;
> H2:= 1/4*x^2+1/9*y^2-z^2 = -1;
> with(plots):
> implicitplot3d(EP,x=-5..5,y=-7..7,z=-2..2, axes=boxed);
> implicitplot3d(HP,x=-5..5,y=-7..7,z=-2..2, axes=boxed);
> implicitplot3d(EC,x=-5..5,y=-7..7,z=-2..2, axes=boxed);
```

Maple vẽ các mặt này với grid ngầm định là [10,10,10]. Do đó đỉnh của Elliptic Cone (0,0,0) không được Maple gán điểm vẽ. Muốn làm cho hình vẽ chính xác hơn, ta cần điều chỉnh grid hoặc tăng numpoints, ví dụ:

```
> implicitplot3d(EC,x=-5..5,y=-7..7,z=-2..2,grid=[9,9,9], axes=boxed);
> implicitplot3d(EC,x=-5..5,y=-7..7,z=-2..2,numpoints=2000, axes=boxed);
```

Để có được các hình vẽ chất lượng cao hơn, cần vẽ Elliptic cone trong hệ tọa độ trụ.

3.2. Vẽ hình trong các hệ tọa độ khác

Maple cho phép vẽ đồ thị trong các hệ tọa độ khác. So với đồ thị hàm ẩn, đồ thị trong hệ tọa độ cực, trụ hay cầu thường cho chất lượng cao hơn.

a) Trong hệ tọa độ cực

Với options coords=polar trong lệnh plot, Maple sẽ vẽ đồ thị trong hệ tọa độ cực. Tọa độ của mỗi điểm trong hệ tọa độ cực là (r, θ) , trong đó r là khoảng cách từ điểm đến gốc tọa độ và θ là góc định hướng giữa nửa đường thẳng chọn trước và véc tơ tạo bởi điểm đó.

Maple đòi hỏi r là một hàm của θ . **Cú pháp là:**

$\text{plot}(r(\theta), \theta=a..b, \text{coords}=\text{polar}, \text{options}).$

> $\text{plot}(\sin(4*t), t=0..2*\text{Pi}, \text{coords}=\text{polar}, \text{scaling}=\text{constrained});$

Dạng tham số trong hệ tọa độ cực có cú pháp lệnh như sau :

$\text{plot}([r(\theta), \theta(t), t=a..b], \text{coords}=\text{polar}, \text{options}).$

> $\text{plot}([\cos(t), 3*t, t=0..Pi], \text{coords}=\text{polar});$

Với gói lệnh plots, lệnh vẽ trong tọa độ cực là polarplot với cú pháp hoàn toàn tương tự như trên và không cần phải có tùy chọn coords=polar.

b) Trong hệ tọa độ trụ

Trong hệ tọa độ trụ, tọa độ của mỗi điểm được cho bởi (r, θ, z) , trong đó (r, θ) là tọa độ cực của hình chiếu của điểm trên mặt phẳng (Oxy) và z là khoảng cách từ điểm đó đến một trục Oz . Maple đòi hỏi r là một hàm của θ và z .

- Dùng lệnh plot3d với tùy chọn coords=cylindrical.

Cú pháp là:

$\text{plot3d}(r(\theta, z), \theta=a..b, z=c..d, \text{coords}=\text{cylindrical}, \text{options}).$

> $\text{plot3d}(\theta*\sqrt{1-z}, \theta=0..2*\text{Pi}, z=0..1, \text{coords}=\text{cylindrical}, \text{axes}=\text{normal});$

Với hàm tham số, cú pháp là

$\text{plot3d}([r(s, t), \theta(s, t), z(s, t)], s=a..b, t=c..d, \text{coords}=\text{cylindrical}, \text{options}).$

> $\text{plot3d}([s, t, s^2+t^2], s=-1..1, t=0..2*\text{Pi}, \text{coords}=\text{cylindrical}, \text{axes}=\text{normal});$

b) Dùng lệnh cylinderplot trong gói lệnh plots.

Cú pháp hoàn toàn tương tự như lệnh plot3d với tùy chọn coords=cylindrical được bỏ đi. Hãy thực tập vẽ lại các hình trên.

Thực hành vẽ mặt Elliptic cone trong tọa độ trụ. So sánh với kết quả trong hàm ẩn.

- áp dụng vẽ mặt Elliptic Cone trong tọa độ trụ

Ta viết tọa độ của Elliptic cone trong hệ tọa độ trụ là

$$\frac{1}{4} r \cos(t) + \frac{1}{9} r \sin(t) - z^2 = 0 \quad \forall$$

Maple muốn r là một hàm của t và z , chúng ta có ngay: $r = \frac{6z}{(9 - 5 \sin(t)^2)^{\left(\frac{1}{2}\right)}}$.

> with(plots):

> $\text{cylinderplot}(6/((9-5*\sin(t)^2)^(1/2))*z, t=0..2*\text{Pi}, z=-2..2, \text{axes}=\text{boxed});$

Các mặt bậc 2 trong hệ tọa độ trụ

Ta viết tọa độ của Elliptic cone trong hệ tọa độ trụ là $\frac{r \cos(t)}{4} + \frac{r \sin(t)}{9} - z^2 = 0$. Vì Maple

muốn r là một hàm của t và z , chúng ta có ngay: $r = \frac{6z}{\sqrt{9 - 5 \sin(t)^2}}$.

> **with(plots):**

> **cylinderplot(6/((9-5*sin(t)^2)^(1/2))*z,t=0..2*Pi,z=-2..2,axes=boxed);**

Với mặt Ellipsoids:

>

cylinderplot([6*sqrt(1-z^2)/sqrt(9*cos(theta)^2+4*sin(theta)^2)],theta=-Pi..Pi,z=-1..1,axes=normal);

Với mặt Hyperboloid 1 và 2 tầng

>

cylinderplot([6*sqrt(1+z^2)/sqrt(9*cos(theta)^2+4*sin(theta)^2)],theta=-Pi..Pi,z=-4..4,axes=normal);

>

cylinderplot([6*sqrt(-1+z^2)/sqrt(9*cos(theta)^2+4*sin(theta)^2)],theta=-Pi..Pi,z=-3..3,axes=normal);

Mặt Paraboloids Elliptic and Hyperbolic

> **cylinderplot([6*sqrt(z)/sqrt(9*cos(theta)^2+4*sin(theta)^2)],theta=-Pi..Pi,z=0..2);**

>

cylinderplot([6*sqrt(z)/sqrt(9*cos(theta)^2-4*sin(theta)^2)],theta=-3..3,z=-1..1,axes=normal);

>

c) Trong hệ tọa độ cầu

Tọa độ của một điểm M trong hệ tọa độ cầu là (r, θ, ϕ) , trong đó r là khoảng cách đến gốc tọa độ, góc θ là góc cực của hình chiếu của M trên (Oxy) và ϕ là góc giữa (Oxy) với vec(OM). Maple yêu cầu r là một hàm của θ và ϕ .

a) Dùng tùy chọn coords=spherical với lệnh plot3d.

Cú pháp là:

plot3d(r(θ , ϕ), θ =a..b, ϕ =c..d, coords=spherical,options)

> **plot3d(1+sin(phi)-cos(theta),theta=0..2*Pi,phi=0..Pi,coords=spherical,numpoints=2000);**

Với hàm tham số, cú pháp là:

plot3d([r(s,t), θ (s,t), ϕ (s,t)],s=a..b,t=c..d, coords=spherical,options).

> **plot3d([s^3+t^3,s+t,t],s=0..2*Pi,t=0..Pi, coords=spherical);**

b) Với gói lệnh plots, vẽ hình trong tọa độ cầu bằng lệnh **sphereplot**, với cú pháp tương tự như với **plot3d** nhưng tùy chọn coords=spherical được bỏ đi. Hãy thực hành vẽ lại các mặt trên.

3.3. Vẽ hình động với Maple

Lệnh animate tạo ra một chuỗi các hình, theo ngầm định là 25 hoặc 16, và cho hiển thị liên tiếp để tạo ra hình động. Độ dao động của vị trí ban đầu và vị trí kết thúc được xác định bởi một biến tham số t , gọi là *biến hình (frame variable)*.

a) **Hình động trên mặt phẳng với lệnh animate**

a1) Với hàm hiện

Cú pháp của lệnh với hàm hiện như sau:

```
animate(plot,[f(x,t),x=a..b],t=c..d,options)
animate(f(x,t),x=a..b,t=c..d,options)
```

trong đó toàn bộ các options giống như options của plot. Sau khi hình vẽ được thực hiện, bằng cách kích chuột trên hình vẽ, một menu mới sẽ xuất hiện trên thanh công cụ chính của Maple cho phép điều khiển sự cử động của hình. Hãy thực hành với ví dụ sau:

```
> restart;
> with(plots):setoptions(thickness=2);
> animate(sin(t*x),x=0..2*Pi,t=1..4,color=blue,scaling=constrained);
```

Lệnh trên có số khung hình là 16. Tham biến hình t không hiển thị. So sánh với lệnh sau:

```
> animate(plot,[sin(t*x),x=0..2*Pi,color=blue],t=1..4,scaling=constrained);
```

Trong animate còn có một options mới so với plot, đó là option frames, cho phép điều số lượng khung hình. Số lượng càng lớn, sự cử động càng "nhịp nhàng" do sự khác giữa 2 khung hình liên tiếp ít hơn. Để tăng độ mịn của hình vẽ, cần tăng giá trị của numpoints. Giá trị ngầm định là 50. Tuy nhiên càng tăng số lượng, Maple càng cần nhiều thời gian và bộ nhớ.

Bài tập:

Hãy thực hiện lệnh vẽ trên với numpoints=200. Hãy thay đổi hàm trên bởi các hàm $t*\sin(x)$, $\sin(x+t)$, $\sin(x)+t$ rồi quan sát hình động của nó. Nêu lên ý nghĩa của biến hình t .

a2) Với hàm tham số

Đối với hàm tham số, lệnh animate được sử dụng như trong các ví dụ sau:

```
> with(plots):
> animate(plot,[[r^2*cos(t),r*sin(t),t=0..2*Pi]],r=1..4,scaling=constrained);
> animate([r^2*cos(t),r*sin(t),t=0..2*Pi],r=1..4,scaling=constrained);
> animate(plot, [[cos(t),sin(t),t=0..A]],
    A=0..2*Pi, scaling=constrained, frames=50 );

> animate(plot, [(1-t^2)/(1+t^2),2*t/(1+t^2),t=-10..A]],
    A=-10..10, scaling=constrained, frames=50 );
```

a3) Trong hệ tọa độ cực

Cú pháp như trong ví dụ sau:

```
> restart;with(plots):
> animate(polarplot,[1+a*cos(t),t=0..2*Pi],a=-2..2,scaling=constrained);
> animate(1+a*cos(t),t=0..2*Pi,a=-2..2,coords=polar,scaling=constrained);
```

Bài tập: Hãy vẽ hình động trong tọa độ cực các hàm sau và quan sát biến hình:

1) $2a \cos(\theta)$ với θ từ $0..2\pi$, a từ $-2..2$.

2) $a + \cos(\theta)$ với θ từ $0..2\pi$, a từ $-2..2$.

3) $\frac{1}{\sigma\pi} \left(\frac{1}{2}\right)^{\left(-\frac{x^2}{2\sigma^2}\right)}$ với $\sigma=1..2$, $x=-5..5$.

a4) Vẽ liên tục với lệnh animatecurve

Maple cung cấp thêm một công cụ vẽ liên tục một đường cong với lệnh animatecurve trong

gói lệnh plots.

> restart;with(plots):

> animatecurve($\frac{1}{3}x^5 - \frac{1}{2}x - \frac{1}{10}$, $x=-2..3$, view=-5..5, numpoints=200, frames=50);

> animate(plot, $\frac{1}{3}x^5 - \frac{1}{2}x - \frac{1}{10}$, $x=-2..t$], $t=-2..3$, color=red, view=-5..5);

Lệnh animatecurve có tất cả các tùy chọn (options) giống như animate.

Nó cũng làm việc được với hàm tham số.

> animatecurve([$3\cos(\theta)$, $\sin(\theta)$], $\theta=0..2\pi$], scaling=constrained);

b) Hình động trong không gian với lệnh animate và animate3d

Giống như trong mặt phẳng, cú pháp của lệnh vẽ hình động trong không gian là:

animate(plot3d, $[f(x,y,t), x=a..b, y=c..d]$, $t=c..d$, options)

hay

animate3d($f(x,y,t)$, $x=a..b, t=c..d$, options)

> restart;with(plots):

> animate(plot3d, $[1-t^2x^2-t^2y^2, x=-1..1, y=-1..1]$, $t=-2..2$);

> animate3d($1-t^2x^2-t^2y^2$, $x=-1..1, y=-1..1, t=-2..2$);

Ngoài ra, vẽ hình với hàm tham số hay hàm ẩn, cú pháp là như sau:

animate(plot3d, $[x(s,t,a), y(s,t,a), z(s,t,a)]$, $s=m..n, t=c..d$], $a=a1..a2$, options)

> animate(plot3d, $[2t-3as^2\sin(t), as^2t, 2s-3\cos(t)]$, $s=-2..2, t=-2..a$], $a=-2..10$);

- Đối với hàm ẩn, cú pháp là:

animate(implicitplot, $[f(x,y,z,t), x=a..b, y=c..d, z=e..f]$, $t=t1..t2$, options)

> HP:= $\frac{1}{4}x^2 - \frac{1}{9}y^2 - z$;

> EC:= $\frac{1}{4}x^2 + \frac{1}{9}y^2 - z^2$;

> animate(implicitplot3d, $[\frac{1}{4}t^2x^2 - \frac{1}{9}y^2 - t^2z, x=-2..2, y=-2..2, z=-2..2]$, $t=-2..2$);

> B := plot3d($1-x^2-y^2$, $x=-1..1, y=-1..1$, style=patchcontour):

opts := thickness=3, color=red;

animate(spacecurve, $[[t, 1-2t^2], t=-1..A]$, opts],

A=-1..1, frames=11, background=B);

Chương 4. Maple trong nghiên cứu và giảng dạy toán

2.4.1. Gói lệnh Student hỗ trợ cho việc dạy và học toán.

- Từ Maple 8, gói lệnh Student được phát triển từ gói lệnh student trước đó nhằm hỗ trợ cho việc dạy và học toán ở đại học và phổ thông. Khai thác khả năng của gói lệnh này sẽ đem đến cho giáo viên rất nhiều công cụ hỗ trợ mới trong phương pháp dạy học. Có thể nói rằng gói lệnh này đã đề cập đến tất cả các nội dung toán học của đại học và phổ thông, cung cấp nhiều lệnh và thủ tục cho các phép toán và algorithm xuất hiện trong chương trình giảng dạy, cung cấp nhiều công cụ tương tác dưới dạng Maplet và hỗ trợ việc làm từng bước các phép toán cơ bản của vi tích phân.
- Gói lệnh Student có 3 gói lệnh con là Calculus1, LinearAlgebra và Precalculus. Để nạp từng gói lệnh, làm như sau:

> with(Student[Precalculus]):

Gói lệnh này gồm nhiều Tutor. Ví dụ sau về Tutor của các hàm sơ cấp:

> StandardFunctionsTutor():

Sau khi nhấn Enter, một cửa sổ độc lập. Ta đưa vào hàm $f(x)$ và chương trình sẽ vẽ đồ thị của hàm $f(x)$ và hàm $a f(x+b) + d$ trên cùng hệ tọa độ.

- Gói lệnh con Calculus1 là gói lệnh quan trọng nhất của Student. Nó chứa các công cụ hỗ trợ từ hướng dẫn thực hiện các phép tính vi tích phân cho đến khảo sát và vẽ đồ thị hàm; từ việc minh họa vẽ tiếp tuyến đường cong cho đến việc tính diện tích, thể tích mặt tròn xoay, v.v...

Ví dụ: Khảo sát hình học và thể tích của vật thể tròn xoay.

> with(Student[Calculus1]):

> VolumeOfRevolution(cos(x) + 3, sin(x) + 2, x=0..4*Pi);

> VolumeOfRevolution(cos(x) + 3, sin(x) + 2, x=0..4*Pi, output=integral);

> VolumeOfRevolution(cos(x) + 3, sin(x) + 2, x=0..4*Pi, output=plot);

2.4.2. Sử dụng Maple như một môi trường dạy học tương tác.

Việc dạy học tương tác là một xu hướng tất yếu của nền giáo dục hiện đại, nhờ có các ưu điểm nổi bật của nó. Việc dạy học tương tác có nhiều hình thức và mức độ phụ thuộc vào nội dung và phương tiện dạy học. Có thể nêu ra 3 hình thức và mức độ sau đây:

- 1) Dạy trên lớp học High Class cả một môn học, một chương hoặc một nội dung cụ thể. Hình thức dạy học này đòi hỏi phải có hệ thống high class hiện đại.
- 2) Chỉ dùng lớp học High Class trong giờ thực hành hoặc ứng với nội dung thích hợp.
- 3) Giáo viên dùng LCD kết nối với máy tính để thực hiện một số khâu trong bài giảng.

Khả năng chuyển đổi giữa hai môi trường toán và văn bản trong Maple

- Maple có 2 môi trường làm việc là toán và văn bản. Người dùng có thể chuyển đổi 1 cách dễ dàng giữa 2 môi trường này. Khi kích hoạt Maple, trang làm việc mở ra với môi trường toán. Con trỏ nằm ở dấu nhắc > màu đỏ. Muốn chuyển sang môi trường văn bản, chỉ cần kích chuột vào biểu tượng T trên thanh công cụ hoặc vào Insert | Text, hay dùng tổ hợp phím Ctrl+T.
- Trong môi trường văn bản, Maple cho phép biên soạn tài liệu theo cấu trúc, cho phép hiển thị theo nhiều tầng lớp, rất phù hợp với việc giới thiệu tổng quan hoặc tổng kết ôn tập. Giống như một hệ soạn thảo văn bản lý tưởng, Maple cho phép thay đổi các font chữ, màu sắc và đặc biệt có thể tạo ra các bookmark để truy xuất nhanh chóng đến các vị trí tùy ý trong trang làm việc hiện hành hay các trang làm việc khác; tạo ra các siêu liên kết để kích hoạt trang làm việc khác, để nối với trang web hay phần trợ giúp của Maple.

Khả năng tạo ra giáo án điện tử hoặc các định dạng văn bản khác.

Maple hỗ trợ tiếng Việt (font ABC) và có thể trích xuất hoặc cho phép nhúng các văn bản của Microsoft Office. Ngoài định dạng Word, Maple cho phép trích xuất ra file LaTeX của văn bản toán hay file HTML để dùng như một giáo án điện tử trên trang web.

2.4.3. Sử dụng Maple như một phương tiện minh họa các khái niệm toán học và đối tượng hình học.

Ví dụ: Minh họa hình ảnh tự nhiên của các đường conic như giao tuyến của một mặt nón và mặt phẳng cắt nó.

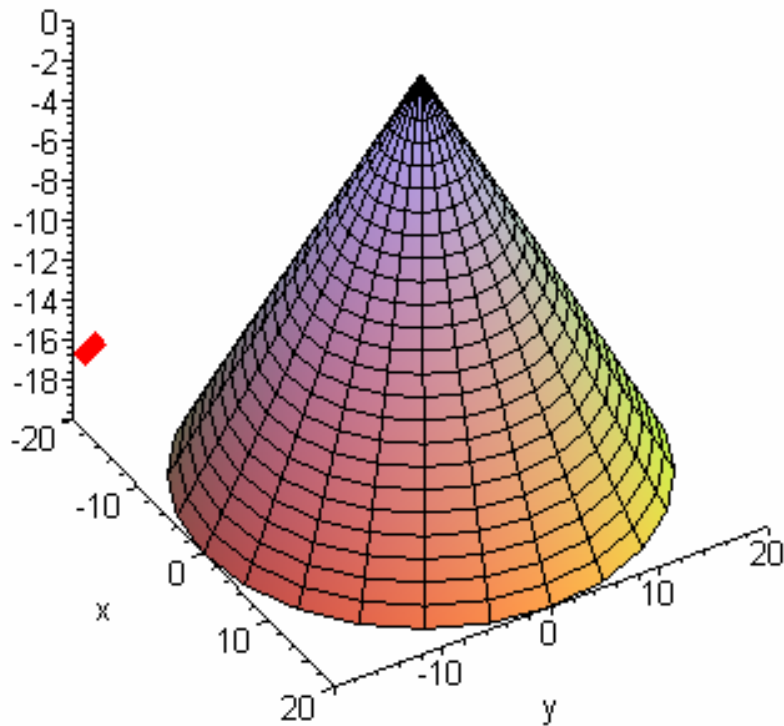
> with(plots):

>

```
animate(plot3d,[y/3-10,x=-20..t,y=-20..t,color=red,style=PATCHNOGRID],t=-18..17,axes=framed,background=plot3d([z*cos(t),z*sin(t),z],z=-20..0,t=-Pi..Pi));
```

Warning, the name changecoords has been redefined

$t = -18.$



Bằng cách thay đổi phương trình thích hợp của mặt phẳng ta có thiết diện là đường hyperbol hay parabol.

2.4.4. Sử dụng Maple để hình thành các khái niệm toán học.

Ví dụ: Khái niệm tích phân xác định và ý nghĩa hình học của nó.

```
> restart;
```

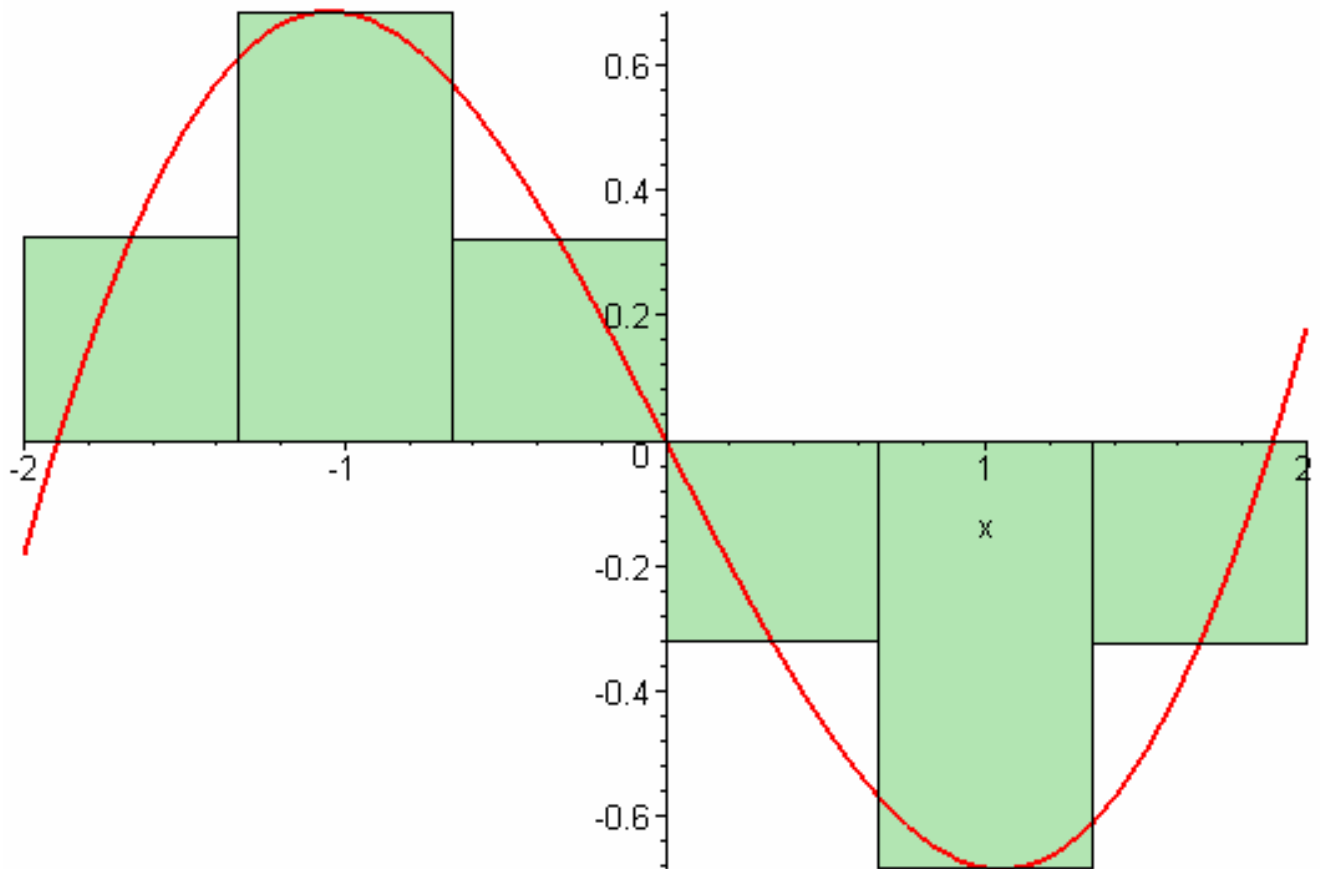
```
> with(plots):with(student):
```

```
> f:=x->x-2*sin(x);
```

Warning, the name changecoords has been redefined

$$f := x \rightarrow x - 2 \sin(x)$$

```
> display(seq(middlebox(f(x),x=-2..2,SoHinh),SoHinh=6..80),insequence=true);
```



Khi ta kích chuột trên hình vẽ, trên thanh công cụ sẽ xuất hiện thanh điều khiển hình vẽ. Kích chuột trên thanh điều khiển, số hình chữ nhật của tổng Riemann sẽ tăng từ 6 lên 80 và dần dần phủ kín phần mặt giới hạn bởi đường cong.

Ghi chú: Với Maple, tất cả các tích phân xác định đều tính được nếu hàm số khả tích. Với những phương pháp mẹo mực như ở phổ thông, các lớp hàm có thể tính được tích phân xác định vô cùng nhỏ so với số các lớp hàm khả tích.

Ví dụ:

```
> g:=3*sin(x)/(exp(x)-x);
> int(g,x=-5..5);
> evalf(%);
> plot(g,x=-5..5);
> display(seq(middlebox(g,x=-5..5,SoHinh),SoHinh=6..80),insequence=true);
```

2.4.5. Sử dụng Maple để dự đoán các kết quả toán học.

Ví dụ: dãy hội tụ và không hội tụ

```
> pointplot([seq([n,sin(n)/(n+1)],n=1..150)],color=blue);
> pointplot([seq([n,abs(sin(n)+1/n)^(sqrt(n))],n=1..1000)],color=blue);
```

2.4.6. Maple hỗ trợ học sinh trong hoạt động tự học và thúc đẩy tìm tòi sáng tạo.

Sử dụng các Tutor trong các gói của Student và các hỗ trợ tính toán từng bước. Ví dụ sau cho phép ta làm với Tutor về tính tích phân.

Ví dụ 1.

```
> with(Student[Calculus1]):
IntTutor();
> int(3*sin(x)/(x^2-cos(x)),x=1..2);
```

```
> evalf(%);
```

```
>
```

```
>
```

Ví dụ 2. Giải phương trình $x^3 + 1 - e^x = 0$:

```
> f:=x^3+1-exp(x);
```

```
> solve(f,x);fsolve(f);
```

Muốn tìm các nghiệm còn lại ta thử vẽ đồ thị hàm f .

```
> plot(f,x=-3..5,y=-5..15);
```

Ta thấy phương còn có 2 nghiệm nữa trong các khoảng $(-1,0)$ và $(1,2)$.

```
> fsolve(f,x=-1..-1/2);fsolve(f,x=1..2);
```

Bằng cách suy luận logic kết hợp với đồ thị ta có thể chỉ ra rằng phương trình có đúng 3 nghiệm. Tương tự, xét phương trình $g = 0$ với:

```
> g:=3*sin(x)-ln(x)-x;
```

```
>
```

2.4.7. Maple hỗ trợ giáo viên trong các hoạt động giảng dạy khác.

Có thể nêu vài ý tưởng về việc sử dụng Maple cho các hoạt động giảng dạy khác của giáo viên toán như sau:

- 1) Dùng Maple để tìm và soạn hệ thống bài tập, đề thi theo ý muốn.
- 2) Kiểm tra các kết quả của các bài toán tính toán để dự đoán các chứng minh (ví dụ về các bài toán giải phương trình, phân tích hoặc rút gọn đa thức, phân thức...)
- 3) Soạn giáo án, vẽ các đồ thị chính xác phục vụ giảng dạy hoặc sinh hoạt chuyên môn; viết các báo cáo khoa học.
- 4) Công cụ hỗ trợ trong bồi dưỡng học sinh giỏi hoặc hoạt động tập dượt nghiên cứu khoa học.
- 5) Là nguồn dữ liệu phong phú để lựa chọn các kịch bản lên lớp.
- 6) Maple là một nguồn mở, cho phép người dùng dễ dàng tạo ra các lệnh và chương trình cho riêng mình bằng các modul lệnh có sẵn và ráp nối bằng các lệnh đơn giản.

Huế ngày 1 tháng 11 năm 2006

Tài liệu tham khảo

1. Corless R. M., *Essential Maple 7, An Introduction for Scientific Programmers*, Springer, 2002.
 2. Phạm Huy Điển, *Tính toán, lập trình và giảng dạy toán học trên Maple*, NXB KH và KT, 2002.
 3. Putz J., *Maple Animation*, Charman & Hall/CRC, 2003.
 4. Nguyễn Chánh Tú, *ứng dụng Maple trong đổi mới phương pháp học tập và giảng dạy toán học*, Kỷ yếu Hội thảo KH, ĐHSP Huế, 4/2004.
 5. Waterloo Maple, *Maple 9, Learning Guide*, 2004.
 6. Waterloo Maple, *Maple 7, Programming Guide*, 2004.
- <http://nctu2006.googlepages.com/student>