

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
INSTITUTO DE ECONOMIA

Oficina de Publicaciones
Casilla 76, Correo 17, Santiago
www.economia.puc.cl

El Control Óptimo y Maple

Alvaro Parra P.

Santiago, Enero 2005

Índice

1. Introducción	3
2. Operaciones básicas en Maple	3
3. Herramientas de Cálculo	7
4. Ecuaciones Diferenciales Lineales	8
5. Sistemas de Ecuaciones Diferenciales Lineales	10
6. Optimización Estática	11
7. Control Óptimo	15
8. Gráficos en 2 y 3 dimensiones	16

1. Introducción

En este manual se presentan todas las herramientas básicas para poder resolver problemas de control óptimo en Maple, es decir, comenzado de operaciones algebraicas simples pasando por optimización estática y por como resolver sistemas de ecuaciones diferenciales.

Para el estudio de este manual se recomienda seguir paso a paso cada uno de los ejercicios realizados en él, para que así, se puedan solucionar prontamente los problemas que surjan en el camino y entender mejor las potencialidades y limitaciones de Maple.

2. Operaciones básicas en Maple

Para comenzar se debe ir al menú de inicio después a todos los programas y por último a Maple. Una vez ahí nos encontraremos con una página en blanco llamada *Maple worksheet* con un signo mayor igual(>), el cual nos indica que se está listo para introducir los comandos necesarios para las operaciones que se quieren realizar, estos comandos se escriben en color rojo mientras que los resultados son representados en color azul y centrados (sí no tiene la suerte de tener el apunte en colores, de aquí en adelante, todo lo que se escriba después de un > es un comando introducido por nosotros en Maple y todo lo centrado es el resultado), ejemplo: escribe $1+1$; y presiona **enter**. (el “;” es indispensable para realizar cualquier operación en Maple e indica que ahí termina la operación que se quiere ejecutar)

```
> 1+1;
2
```

veamos más operaciones

```
> 10*5;
50
```

```
> 3^4;
81
```

```

> x*y/2;
                                1/2 xy
> log(1);
                                0
> pi*2;
                                2 pi
> exp(3*x);
                                e^(3x)

```

Creo ya tienes la idea. En Maple las operaciones sobre funciones son, por lo general, de la siguiente forma:

operación(función, dominio);

aunque esto puede parecer un poco vago se nos aclarará enseguida con algunos ejemplos, calculemos las siguientes operaciones:

```

- sum_{k=1}^{\infty} 1/k^2,
  > sum(1/k^2, k=1..infinity );
                                1/6 pi^2

```

```

- prod_{i=1}^{10} i
  > product(i, i=1..10);
                                3628800

```

```

- d/dx (2x^3)
  > diff(2*x^3, x);
                                6 x^2

```

Algo que nos puede ser de gran utilidad es darnos cuenta que Maple nos puede servir como procesador de texto ya que en la barra de herramientas existe un icono con una **T** lo cual nos permite ingresar texto entre los resultados y además si escribimos las operaciones con mayúsculas Maple no nos entrega los resultados sino que la función misma, por ejemplo:

```

> Product(i, i=1..10);

```

$$\prod_{i=1}^{10} i$$

Luego podemos escribir la operación y el resultado juntos:

```
> Product(n,n=1..10)=product(n,n=1..10);
```

$$\prod_{n=1}^{10} n = 3628800$$

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

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Después basta con ir a la barra de herramientas y en *File* se encuentra la opción *Export as* y están las opciones de guardado en Word (RTF), L^AT_EX (o Scientific Word) o HTML (para páginas Web). Gran parte de éste apunte esta escrito en Maple.

Volviendo a lo nuestro, nos será útil a futuro saber que podemos etiquetar las funciones y así no tener que escribirlas cada vez que necesitamos ejecutarlas, esto se hace así:

nombre de la función:= función; (los signos := son fundamentales)

por ejemplo, etiquetemos sin(x):

```
> a:= sin(x);
```

$$a := \sin(x)$$

ahora derivemos a *a* respecto a *x*

```
> diff(a,x);
```

$$\cos(x)$$

si lo que se necesita es obtener valores numéricos de la función la tendrá que guardar de la siguiente forma:

nombre de la función:= variables del dominio de la función -> función; por ejemplo:

```
> b:= (x,y)->x^2-6*y^3;
```

$$b := (x, y) \mapsto x^2 - 6y^3$$

¿cuanto valdrá nuestra función el punto (1,2)?, es aquí donde introducimos el comando **evalf**

```
> evalf(b(1,2));
```

$$-47,0$$

donde *b* es la función y lo que esta entre paréntesis es el punto que queremos evaluar. Ahora ya estamos en condiciones de resolver sistemas de ecuaciones, resolvamos el siguiente problema:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 5 & 4 & 3 & 2 \\ 0 & 3 & 4 & -8 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} h \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 41 \\ 20 \\ 125 \\ 9 \end{pmatrix}$$

aquí hay que hacer notar que Maple no trabaja muy bien con matrices así que para introducir el sistema habrá que introducirlo en forma de ecuaciones, esto se hace así:

```
> eqn1 := h+2*b+3*c+4*d+5*e=41;
      eqn1 := h + 2b + 3c + 4d + 5e = 41
> eqn2 := 5*h+5*b+4*c+3*d+2*e=20;
      eqn2 := 5h + 5b + 4c + 3d + 2e = 20
> eqn3 := 3*b+4*c-8*d+2*e=125;
      eqn3 := 3b + 4c - 8d + 2e = 125
> eqn4 := h+b+c+d+e=9;
      eqn4 := h + b + c + d + e = 9
```

Para resolver el sistema ocupamos el comando **solve**, podemos apreciar que es un sistema de cuatro ecuaciones y cinco incógnitas, luego tendremos que dejar los términos en función de otro, esto se hace de la siguiente manera:

```
> solve( {eqn1, eqn2, eqn3, eqn4} );
      {h = 2, c = 337/4 + 31/4 d, b = -115/2 - 11/2 d, e = -79/4 - 13/4 d, d = d}
```

Como podemos apreciar Maple dejó todo en función de d , puede ser que necesitemos dejar todo en función de otra variable, por ejemplo e esto se haría de la siguiente forma:

```
> solve( {eqn1, eqn2, eqn3, eqn4}, {h, b, c, d} );
      {h = 2, d = -79/13 - 4/13 e, c = 483/13 - 31/13 e, b = -313/13 + 22/13 e}
```

lo que se hizo aquí fue pedirle a Maple que resolviera las ecuaciones 1, 2, 3, 4 y las incógnitas h, b, c, d ; luego todo se deja en función de la variable omitida, en este caso e , agreguemos una quinta ecuación para obtener una solución numérica y no algebraica:

```
> eqn5:= e=10;
      eqn5 := e = 10
```

ahora resolvamos el sistema:

```
> solve( {eqn1, eqn2, eqn3, eqn4, eqn5});
      {h = 2, e = 10, b = -93/13, c = 173/13, d = -119/13}
```

3. Herramientas de Cálculo

En el apartado vimos que Maple no sólo resuelve operaciones matemáticas de álgebra sino que también vimos que podía derivar. A continuación se presenta como calcular integrales indefinidas y definidas, expandir por Taylor y sacar límites

```
–  $\int \sin(x)dx$   
> int(sin(x),x);  
– cos(x)
```

```
–  $\int_1^{3\pi} \sin(x)dx$   
> int(sin(x), x=1..3*Pi);  
cos(1) + 1
```

```
– expandir por taylor cos(x) respecto a x  
> taylor(cos(x),x);  
series(1 - 1/2 x^2 + 1/24 x^4 + O(x^6), x, 6)  
–  $\lim_{n \rightarrow \infty} (1 + 1/n)^n$ 
```

```
> limit((1+1/n)^n,n=infinity);  
e
```

Además Maple trae un package de cálculo vectorial llamado **VectorCalculus** que nos permite, entre otras cosas, encontrar el vector gradiente, el Hessiano y el Jacobiano de una función o un conjunto de funciones. Primero hay que activar el package

```
> with(VectorCalculus);
```

```
Warning, the assigned names <, > and <|> now have a global binding  
<br>
```

```
Warning, these protected names have been redefined and  
unprotected: *, +, ., D, Vector, diff, int, limit, series <br>
```

El segundo paso es siempre definir en que plano nos vamos a manejar (ej: Cartesiano, Polar), el número de dimensiones y cuales son las variables

```
> SetCoordinates(cartesian[x,y]);  
cartesianx,y
```

Acá lo que hicimos fue decirle a Maple que trabajaremos en un plano Cartesiano, en \mathbb{R}^2 y que nuestras variables serán x e y . Para ilustrar como funcionan estos comandos trabajaremos con la siguiente función $x^2 \sin(y)$ a la cual llamaremos c .

```

> c:=x^2*sin(y);
                                     c := x^2 sin(y)
> Gradient(c);
                                     [ 2x sin(y) ]
                                     [ x^2 cos(y) ]
> Hessian(c,[x,y]);
                                     [ 2 sin(y)  2 x cos(y) ]
                                     [ 2 x cos(y) -x^2 sin(y) ]
> Jacobian([c],[x,y]);
                                     [ 2 x sin(y)  x^2 cos(y) ]

```

Por último con un package llamado **LinearAlgebra** podemos calcular determinantes de matrices cuadradas, primero guardemos el Hessiano

```

> h:=Hessian(c,[x,y]);
                                     h := [ 2 sin(y)  2 x cos(y) ]
                                     [ 2 x cos(y) -x^2 sin(y) ]

```

y ahora ejecutar el package

```

> with(LinearAlgebra);
Warning, the names
&x, CrossProduct and DotProduct have been rebound
y el determinante se calcula de la siguiente manera
> Determinant(h);

```

$$-2 \sin(y)^2 x^2 - 4 x^2 \cos(y)^2$$

Todo esto nos será muy útil en el ítem de optimización estática.

4. Ecuaciones Diferenciales Lineales

En los siguientes apartados se introducirá la forma de como resolver ecuaciones diferenciales y sistemas de ecuaciones diferenciales, el comando para resolver éste tipo de problemas es **dsolve** (este comando es análogo a *solve* y *evalf* vistos ya con anterioridad), por otro lado, la forma que usaremos para representar las derivadas de la función será con el comando **D** y las derivadas parciales de una variable respecto a otra con **diff**, en este apartado usaremos principalmente las derivadas de la función. Partamos con el problema más simple, crecimiento exponencial ($y'_t = ay_t$), primero etiquetemos la ecuación:

```

> ecd1:= D(y)(t)=a*y(t);
                                     ecd1 := D(y)(t) = ay(t)

```

Ahora obtengamos la solución general

```
> dsolve(ecd1);
```

$$y(t) = -C1 e^{at}$$

donde $C1$ representa la *constante 1*, ¿cuál será la solución particular si $y(0) = 100$?, introduzcamos la condición inicial:

```
> CI1:= y(0)=100;
```

$$CI1 := y(0) = 100$$

y ahora resolvamos

```
> dsolve( {ecd1,CI1},{y(t)});
```

$$y(t) = 100 e^{at}$$

lo que hicimos fue decirle al Maple que resolviera una ecuación diferencial (dsolve), que la tenía que resolver para la ecuación diferencial 1 y la condición inicial 1 (ecd1, CI1) y que resolviera respecto a $y(t)$ ($\{y(t)\}$). Veamos ahora un sistema con entradas de la forma $y'_t = ay_t + b$ y con la condición inicial $y(0) = y_0$:

```
> ecd2:= D(y)(t)=a*y(t) + b;
```

$$ecd2 := D(y)(t) = ay(t) + b$$

```
> CI2:= y(0)= y0;
```

$$CI2 := y(0) = y0$$

```
> dsolve({ecd2,CI2},{y(t)});
```

$$y(t) = -\frac{b}{a} + \frac{(b+y_0 a)e^{at}}{a}$$

que es la misma solución obtenida en la página 38 del libro del curso¹.

Para explicar un poco la notación en sistemas de orden mayor consideremos el siguiente ejemplo: $y_t''' + 3y_t'' - 3y_t' + 5 = 0$. Existen al menos dos formas de escribirlo, la primera es la siguiente:

```
> ecd3:= D(D(D(y)))(t)+3*D(D(y))(t)-3*D(y)(t)+5=0;
```

$$ecd3 := (D^{(3)}(y)(t) + 3(D^{(2)}(y)(t) - 3D(y)(t) + 5) = 0$$

acá lo que se le está pidiendo a Maple es “*calcula la derivada de la derivada de la derivada de y respecto a t*” y una forma alternativa de hacerlo es así:

```
> ec4:=(D@@3)(y)(t)+3*(D@@2)(y)(t)-3*D(y)(t)+5=0;
```

$$ec4 := (D^{(3)}(y)(t) + 3(D^{(2)}(y)(t) - 3D(y)(t) + 5) = 0$$

y acá lo que se le está pidiendo es “*calcula la tercera derivada de y respecto a t*”, claramente entre mayor sea el orden del sistema más conveniente será el segundo método.

¹Introducción al análisis de sistemas dinámicos de Gonzalo Edwards, de ahora en adelante todas las referencias que se hagan son a éste libro

Un caso que quizás nos podría llegar a dar problemas sería si la solución al sistema es compleja, para ver lo que sucede resolvamos la siguiente ecuación diferencial (y de paso introduciremos otra forma de resolver problemas): $3y'' + y' + 2y = 0$; $y(0) = 30$; $y'(0) = 10$

```
> dsolve({3*(D@@2)(y)(t)+D(y)(t)+2*y(t)=0,y(0)=30, D(y)(0)=10}, {y(t)});
```

$$y(t) = \frac{90}{23} \sqrt{23} e^{-1/6t} \sin(1/6 \sqrt{23}t) + 30 e^{-1/6t} \cos(1/6 \sqrt{23}t)$$

Dos cosas son interesantes acá, la primera es que Maple nos entrega las soluciones complejas a través de senos y cosenos tal como estamos acostumbrados y, la segunda, que no es necesario etiquetar la ecuación y las condiciones iniciales por separado para después resolver, sino que se puede hacer todo junto. Mi recomendación particular es que las introduzcas por separado para así poder detectar más fácilmente los posibles errores de tipeo.

5. Sistemas de Ecuaciones Diferenciales Lineales

A continuación empezaremos a estudiar sistemas de ODE's, los principios para resolver estos sistemas en si son los mismos que en el apartado anterior sólo que en vez de pedir que nos resuelva una ODE pedimos que nos resuelva n . El método es análogo al visto para resolver un sistema de ecuaciones, lo único que cambia es el comando que ejecuta la acción que, como ya vimos, esta vez es **dsolve**. Resolvamos el siguiente sistema:

$$\begin{pmatrix} x'_t \\ y'_t \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_t \\ y_t \end{pmatrix}$$

Primero hay que introducir las ecuaciones, esto lo podemos hacer de la forma que ya conocemos (etiquetando cada una por separado) o bien etiquetando ambas de una sola vez, lo cual se hace de la siguiente forma:

```
> sys1:=D(x)(t)=2*x(t)+y(t),D(y)(t)=2*x(t)+3*y(t);
```

$$sys1 := D(x)(t) = 2x(t) + y(t), D(y)(t) = 2x(t) + 3y(t)$$

ahora resolvamos:

```
> dsolve({sys1},{y(t),x(t)});
```

$$\{x(t) = 1/2 _C1 e^{4t} - _C2 e^t, y(t) = _C1 e^{4t} + _C2 e^t\}$$

Esta es la misma solución al ejemplo 10.4 del libro, sólo que en el libro está escrita matricialmente.

Aquí, a diferencia de cuando queríamos resolver una ecuación diferencial, siempre tenemos que introducir respecto a qué estamos resolviendo. Obtengamos ahora una solución específica introduciendo las siguientes condiciones iniciales, $x(0)=100$ y $y(0)=50$

```
> CI3:= x(0)=100, y(0)=50;
```

$$CI3 := x(0) = 100, y(0) = 50$$

resolviendo obtenemos

```
> dsolve({sys1, CI3}, {y(t), x(t)});
```

$$\{x(t) = 50 e^{4t} + 50 e^t, y(t) = 100 e^{4t} - 50 e^t\}$$

Intentemos ahora resolver la siguiente ecuación (ejemplo 10.7):

$$\begin{pmatrix} x'_t \\ y'_t \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} 1 \\ 4 \end{pmatrix} e^{-0,5t}$$

Como podemos apreciar la parte homogénea no ha cambiado respecto al ejemplo anterior, sólo que ahora le hemos introducido una entrada, resolvamos el sistema:

```
> sys2:=D(x)(t)=2*x(t)+y(t)+exp(-0.5*t),
```

```
> D(y)(t)=2*x(t)+3*y(t)+4*exp(-0.5*t);
```

```
sys2 := D(x)(t) = 2 x(t) + y(t) + e^{-0,5t}, D(y)(t) = 2 x(t) + 3 y(t) + 4 e^{-0,5t}
```

```
> dsolve({sys2}, {y(t), x(t)});
```

$$\{y(t) = e^{4t} C2 + e^t C1 - \frac{32}{27} e^{-1/2t}, x(t) = 1/2 e^{4t} C2 - e^t C1 + \frac{2}{27} e^{-1/2t}\}$$

La única diferencia que existe entre la solución del ejemplo 10.7 y ésta, es que como existen términos en común en la solución homogénea y la particular, Maple incorporó estos valores a las constantes del sistema homogéneo.

6. Optimización Estática

6.1. Optimización sin Restricciones

El problema es $\max_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ o bien $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$, pero como $\max -f(\mathbf{x}) = \min f(\mathbf{x})$ sólo daremos condiciones para un máximo, estas son

- *CPO*: El gradiente igual a cero.

$$\vec{\nabla} f = 0$$

- *CSO*: El Hessiano es negativo semidefinido

Maximicemos $g(x, y) = -x^2 - 0,5y^2 - 5x - 3y$, primero introduzcamos la función

```
> g:=-x^2-0.5*y^2-5*x-3*y;
```

$$g := -x^2 - 0,5y^2 - 5x - 3y$$

ahora encontremos las condiciones de primer orden

```

> cpo1:= diff(g,x)=0;
                                cpo1 := -2x - 5 = 0
> cpo2:= diff(g,y)=0;
                                cpo2 := -1,0y - 3 = 0
y resolviendo el sistema de ecuaciones obtenemos
> solve({cpo1,cpo2});
                                {y = -3., x = -2,5}

```

¿Pero será un máximo? veamos las condiciones de segundo orden, pero primero hay que activar el package **VectorCalculus**

```

> with(VectorCalculus):
Warning, the names &x, CrossProduct and DotProduct have been
rebound
y el Hessiano es
> r:= Hessian(g,[x,y]);

```

$$r := \begin{bmatrix} -2 & 0 \\ 0 & -1,0 \end{bmatrix}$$

Podemos ver que el primer menor es igual a -2 y que el determinante es 2 y por tanto es un máximo pero la idea es usar las herramientas de Maple así que necesitamos el package **LinearAlgebra**

```

> with(LinearAlgebra):
y calculamos el determinante
> Determinant(r);
                                2,0

```

es un mínimo.

6.2. Optimización con Restricciones de Igualdad

En esta sección queremos ilustrar un método para resolver el siguiente problema

$$\begin{aligned}
& \underset{\mathbf{x} \in \mathbb{R}^n}{\text{máx}} f(x_1, x_2, \dots, x_n) && s.a. \\
& b_1 = g_1(x_1, x_2, \dots, x_n) \\
& \quad \vdots \\
& b_m = g_m(x_1, x_2, \dots, x_n)
\end{aligned}$$

Sabemos que, si usamos el método de Lagrange, la solución es aumentar el número de dimensiones de n a $n + m$ usando m variables auxiliares (tantas como restricciones tengamos), así el problema nos queda

$$\underset{\mathbf{x} \in \mathbb{R}^n, \lambda \in \mathbb{R}^m}{\text{máx}} \mathcal{L} = f(x_1, x_2, \dots, x_n) + \sum_{i=1}^m \lambda_i (b_i - g_i(x_1, x_2, \dots, x_n))$$

y las condiciones para encontrar un máximo son:

- (CPO): El gradiente del lagrangiano igual a cero

$$\vec{\nabla} \mathcal{L} = 0 \Rightarrow \vec{\nabla} f - \lambda \vec{\nabla} g = 0 \Rightarrow \vec{\nabla} f = \lambda \vec{\nabla} g$$

- (CSO): La secuencia de $n - m$ menores orlados del Hessiano, partiendo del menor $1+m$, alternen de signo de la forma $(-1)^{1+m} (h)$ o que la función objetivo sea cóncava y que el set de restricciones sea convexo $(c)^2$.

Todas estas conclusiones son validas en un problema de maximización pero como $\min f(x) = \max -f(x)$ todos los problemas se pueden transformar en uno de maximización.

Para explicar la optimización con restricciones lo más fácil, nuevamente, es partir por un ejemplo. Minimicemos la distancia de un punto en \mathbb{R}^3 respecto a otro y con una restricción (para que tenga un sentido el problema, sino la solución sería el mismo punto). Para esto tenemos que usar el package llamado **VectorCalculus**, partamos ejecutándolo:

```
> restart;with(VectorCalculus):interface(showassumed=0);
```

```
Warning, the assigned names <,> and <|> now have a global binding  
<br>
```

```
Warning, these protected names have been redefined and  
unprotected: *, +, ., D, Vector, diff, int, limit, series <br>
```

Definamos el número de dimensiones del problema

```
> SetCoordinates(cartesian[x,y,z]);
```

cartesian_{x,y,z}

Supongamos que el punto en cuestión es **(3, 2, 3)**, entonces la función que quieres minimizar es la siguiente:

```
> f:=(x-3)^2+(y-2)^2+(z-3)^2;
```

$$f := (x - 3)^2 + (y - 2)^2 + (z - 3)^2$$

Y agreguemos la siguiente restricción $x+2y+4z=10$

```
> cond := 10-(1*x + 2*y+4*z);
```

$$cond := 10 - x - 2y - 4z$$

Ahora estamos en condiciones de formar el lagrangiano (*lag*), pero como este es un problema de minimización hay que transformarlo en uno de maximización

²Se puede demostrar que $c \Rightarrow h$, pero que $h \not\Rightarrow c$, luego c es una condición más fuerte que h .

```
> lag:= -(f+lambd*cond);
```

$$lag := -(x-3)^2 - (y-2)^2 - (z-3)^2 - \lambda(10-x-2y-4z)$$

La condición necesaria de primer orden nos dice que la derivada del lagrangiano respecto a cada variable es igual a cero

```
> cpo1:= diff(lag,x)=0;
```

$$cpo1 := -2x + 6 + \lambda = 0$$

```
> cpo2:= diff(lag,y)=0;
```

$$cpo2 := -2y + 4 + 2\lambda = 0$$

```
> cpo3:= diff(lag,z)=0;
```

$$cpo3 := -2z + 6 + 4\lambda = 0$$

```
> cpo4:= diff(lag,lambd)=0;
```

$$cpo4 := -10 + x + 2y + 4z = 0$$

luego basta con resolver el sistema de ecuaciones

```
> solve({cpo1,cpo2,cpo3,cpo4});
```

$$\left\{x = \frac{18}{7}, \lambda = \frac{-6}{7}, z = \frac{9}{7}, y = \frac{8}{7}\right\}$$

Para la condición de segundo orden hay que analizar el Hessiano

```
> h:=Hessian(lag,[x,y,z,lambd]);
```

$$h := \begin{bmatrix} -2 & 0 & 0 & 1 \\ 0 & -2 & 0 & 2 \\ 0 & 0 & -2 & 4 \\ 1 & 2 & 4 & 0 \end{bmatrix}$$

para esto usamos el package **LinearAlgebra**

```
> with(LinearAlgebra):
```

```
Warning, the names &x, CrossProduct and DotProduct have been rebound
```

El primer menor tiene que ser positivo: $-2(-2)-0(0)=4$ y lo es, el segundo tiene que ser negativo y se calcula de la siguiente forma

```
> Minor(h,4,4);
```

-8

Lo hizo Maple fue calcular el menor de h correspondiente a borrar la cuarta fila y la cuarta columna, Por último el determinante debe ser positivo

```
> Determinant(h);
```

84

6.3. Optimización con Restricciones de Desigualdad

Maple tiene un package llamado **Simplex** para resolver este tipo de problemas pero tiene la gran limitación que sólo puede ser usado con funciones de grado uno. Afortunadamente con el método de **Kuhn-Tucker** uno puede replicar los pasos del apartado anterior y analizar todos los casos posibles (Esto se deja al lector). A modo de ejemplificar como funciona el package **Simplex** acá se presenta como maximizar $x - y$ sujeto a $3x - 4y \leq 5, 4x + 3y \leq 4$ y después el mismo problema pero se le agregan condiciones de no negatividad.

```
> with(simplex):
```

```
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

```
> maximize(x-y, {3*x-4*y<=5, 4*x+3*y<=4}
> );
```

$$\left\{x = \frac{31}{25}, y = \frac{-8}{25}\right\}$$

```
> maximize(x-y, {3*x-4*y<=5, 4*x+3*y<=4}, NONNEGATIVE);
{y = 0, x = 1}
```

Si lo que queremos es minimizar se tiene que ocupar el comando **minimize** en vez de maximize, además este package sirve para resolver problemas con restricciones de igualdad pero con las mismas limitaciones ya mencionadas.

7. Control Óptimo

En esta parte se desarrollara un método para resolver problemas de la forma

$$\text{Max } J = \Psi(x(T)) + \int_0^T l(x(t), u(t)) dt$$

$$\text{s.a } x'(t) = f(x(t), u(t)), \quad x(t) = x_t$$

Para resolver un problema de control óptimo debemos ocupar las herramientas desarrolladas con anterioridad. Como siempre la mejor forma de ilustrar el problema es con un ejemplo, resolvamos el siguiente problema

$$\text{Max } J = 4x_1(4) + x_2(4) - 0,5 \int_0^4 u(t)^2 dt \quad \text{s.a}$$

$$\begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t) \quad (*)$$

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad (**)$$

Así las condiciones para un máximo son:

1. (*)
2. (**)
3. $[-\lambda_1'(t) \quad -\lambda_2'(t)] = [\lambda_1(t) \quad \lambda_2(t)] \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} + [0 \quad 0]$
4. $[\lambda_1(4) \quad \lambda_2(4)] = [4 \quad 1]$
5. $\max_{u(t)} H(\lambda(t), x(t), u(t)) = \lambda'(t)f(x, u) + l(x, u)$

Resolvamos la condición c) y d), por notación definimos $\lambda_1 = z$ y $\lambda_2 = w$

- > $C3 := -D(z)(t) = 2 * z(t), \quad -D(w)(t) = 3 * w(t);$
 $C3 := -D(z)(t) = 2 z(t), \quad -D(w)(t) = 3 w(t)$
- > $C4 := z(4) = 4, \quad w(4) = 1;$
 $C4 := z(4) = 4, \quad w(4) = 1$
- > $\text{dsolve}(\{C3, C4\}, \{z(t), w(t)\});$
 $\{z(t) = 4 e^{8-2t}, w(t) = e^{12-3t}\}$

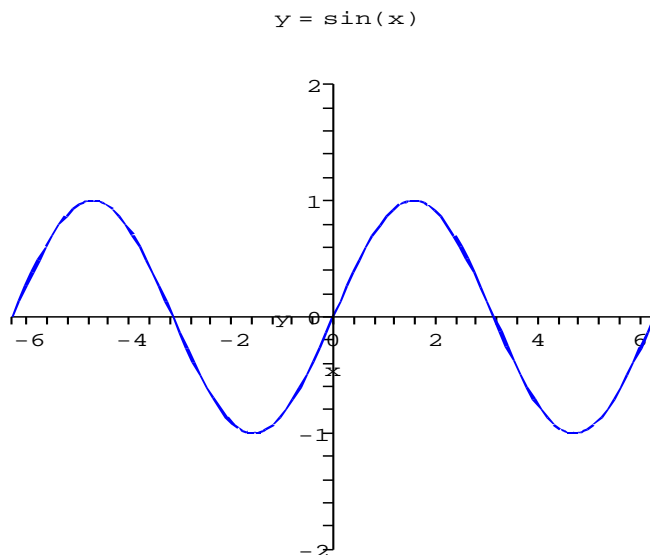
Con esto ya podemos construir el Hamiltoniano

- > $H := 8 * e^{(8-2*t)} * x + 3 * e^{(12-3*t)} * y + (4 * e^{(8-12*t)} + 2 * e^{(12-3*t)}) * u - 0.5 * u^2;$
 $H := 8 e^{8-2t} x + 3 e^{12-3t} y + (4 e^{8-2t} + 2 e^{12-3t}) u - 0,5 u^2$
- > $C5 := \text{diff}(H, u);$
 $C5 := 4 e^{8-2t} + 2 e^{12-3t} - 1,0 u$
- > $\text{solve}(C5=0, u);$
 $\{u = 4,0 e^{8,0-2,0t} + 2,0 e^{12,0-3,0t}\}$

8. Gráficos en 2 y 3 dimensiones

Maple es muy amigable para hacer gráficos, esto se hace con el comando **plot** cuando se trata de hacer gráficos en dos dimensiones y con **plot3d** cuando se quiere hacer en tres, hagamos un gráfico simple: $\sin(x)$, para el intervalo $[-2\pi, 2\pi]$, que muestre sólo valores del recorrido en el intervalo $[-2, 2]$ y que lo haga en azul (el color por default es rojo) :

```
> plot( sin(x),x=-2*Pi..2*Pi, y=-2..2,title="y = sin(x)", color=[blue]
);
```



simple ¿o no?, intentemos algo más complicado. Imaginemos una función de utilidad Cobb-Duglas de la siguiente forma... grafiquemos el mapa de curvas de indiferencia para los niveles de utilidad 2, 5, 7.5, 10 y agreguemos además la solución al problema del consumidor si el ingreso del individuo es 40 y el precio de ambos bienes es 2. dado que queremos un gráfico en dos dimensiones tendremos que dejar todo expresado en una variable, por ejemplo dejemos y en función de x:

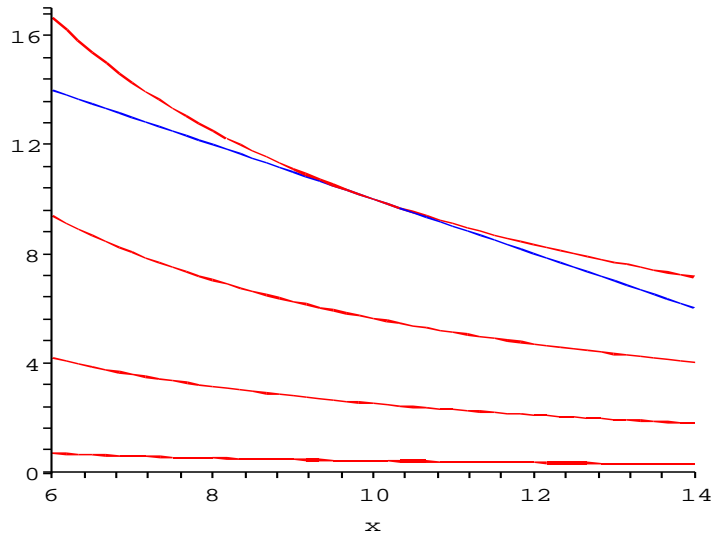
$$\bar{u} = x^{0,5}y^{0,5} \quad \Rightarrow \quad y = \left(\frac{\bar{u}}{x^{0,5}}\right)^2$$

$$I = P_x x + P_y y \quad \Rightarrow \quad y = \frac{m - P_x x}{P_y} \quad \Rightarrow \quad y = 20 - x$$

Ahora grafiquemos las curvas de indiferencia en rojo y la restricción presupuestaria en azul (ojo se acotó el dominio):

```
> plot([(10/x^(0.5))^2,(5/x^(0.5))^2,(7.5/x^(0.5))^2,(2/x^(0.5))^2,20-x],
> x=6..14, color=[red,red,red,red,blue],title="Curvas de
> indiferencia " );
```

Curvas de indiferencia



Por último hagamos un gráfico en 3d de $xe^{-x^2-y^2}$ restringiendo el dominio y el recorrido al intervalo $-2,2$ y que lo entregue en un cuadrado (esto se recomienda para no perder la noción de altura y profundidad):

```
> plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,axes=BOXED,  
title="el gráfico de un plano" );
```

Un plano

