**Párrafo 11.1 Sucesiones**

El objetivo de este párrafo es estudiar algunas propiedades de las sucesiones y sus límites.

Existen, fundamentalmente, dos maneras de definir una sucesión:

a) en forma directa, es decir, a través de una fórmula, como, por ejemplo, en  ;

b) en forma recursiva, como en la sucesión de Fibonacci siguiente: , , 

> **restart:**

Para los futuros gráficos:

> **with(plots):**

Para definirlas en forma directa se recurre al comando **seq:**

> **N:=20:**

> **a:=[seq(1/n,n=1..N)];**

Los corchetes permiten definir la sucesión como una lista. Con ello, por ejemplo:

> **a[10];**

> **a[2016];**

Conviene mejor definir al principio el número N de términos que se quiere de la sucesión , pues resulta más fácil el cambiarlo luego en todas partes. Debe notarse sí que el comando: **seq** no puede ser usado para crear sucesiones infinitas:

> **a:=[seq(1/n,n=1..infinity)];**

Pero sí podemos calcular el límite de la sucesión, mediante (previa desasignación de la variable n). El comando a usar es: **limit(****, infinity)** (con **Limit(****, infinity)** se escribe el límite):

> **n:='n':**

> **Limit(1/n,n=infinity); limit(1/n,n=infinity);**

O mejor, en una línea:

> **Limit(1/n,n=infinity)=limit(1/n,n=infinity);**

Para graficar esta sucesión, en el eje X:

> **A:=[seq([a[n],0],n=1..N)]:**

> **plot(A,0..1,style=point,symbol=cross);**

Mejor es graficarla como una función:

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,a[n]]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **display(seq(p||n,n=1..N));**

Para el caso de una definición recursiva, podemos usar el comando: **for** (nótese que el **from 1** lo subentiende):

> **b[1]:=1: b[2]:=1:**

> **for n to N-2 do b[n+2]:=b[n]+b[n+1]: end do;**

Esta sucesión de Fibonacci, por su importancia, tiene un comando propio: **fibonacci** en el paquete: **combinat**:

> **with(combinat,fibonacci):**

> **fibonacci(N);**

> **n:='n':**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,b[n]]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **display(seq(p||n,n=1..N));**

Se puede construir otras sucesiones a partir de las dadas. Por ejemplo (una manera equivalente de definir sucesiones es mediante el uso del operador **$**):

> **n:='n':**

> **c:=[1/n^2+(-1)^n\*a[n] $n=1..N];**

> **c[10];**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,c[n]]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **p0:=implicitplot(y=0,x=0..N,y=-1..1,color=black,thickness=2):**

> **display({p0,seq(p||n,n=1..N)},axes=none);**

Por su parte:

> **n:='n': limit(1/n^2+(-1)^n/n,n=infinity);**

Se puede también operar con las expresiones:

> **eval(a[5]+b[5]); eval(a[5]\*b[5]); eval(a[5]/c[5]);**

El programa puede también resolver ecuaciones recursivas, mediante el comando: **rsolve**. Supongamos que se quiere resolver, por ejemplo, para la sucesión de Fibonacci :

> **F:=rsolve({f(n+2)=f(n)+f(n+1),f(1)=1,f(2)=1},f);**

Para transformar la expresión F en una función de n se utiliza el comando: **unapply**:

> **F:=unapply(F,n);**

> **radnormal(F(N));**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,F(n)]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **display(seq(p||n,n=1..N));**

> **n:='n': limit(F(n),n=infinity);**

Estudiemos ahora la sucesión de Euler, tal vez más la importante de todas:

> **e:=[seq((1+1/n)^n,n=1..N)]:**

> **evalf(e[N]);**

> **n:='n':**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,e[n]]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **eu:=evalf(exp(1)):**

> **p0:=implicitplot(y=eu,x=0..N,y=2..3,color=black,thickness=2):**

> **display({p0,seq(p||n,n=1..N)});**

Consideremos la sucesión como una función:

> **n:='n': Eu:=unapply((1+1/n)^n,n):**

y preguntemos:

> **solve(Eu(n)<3,n);**

Obviamente, no entendemos nada de la primera parte de la respuesta, pero la segunda nos dice que para todo n positivo la función, y por lo tanto, también la sucesión está acotada por 3. Verifiquemos hasta n = 500 que la sucesión es creciente. El comando **while** (= mientras) genera un "loop" que nos puede ayudar:

> **n:=1: while n<500 do a:=evalb(Eu(n+1)>Eu(n)): if a=false then print(n) else n:=n+1: end if: end do: print(n);**

sospechando entonces que la sucesión tiene un límite menor que 3. Y, por supuesto:

> **n:='n': limit(e[n],n=infinity);**

Mejor:

> **Limit((1+1/n)^n,n=infinity)=limit((1+1/n)^n,n=infinity);**

En Maple el número e tiene un símbolo especial , que equivale a exp(1):

> **e:='e': e;**

(nótese la cursiva).

> **exp(1);**

> **eu:=evalf(%);**

De lo dicho para esta sucesión vemos que es monótona creciente hacia el valor **e**. Luego, por la definición de límite debe haber un valor  para  > 0 tal que e(n) > **e** - , para todo n > . Determinémoslo para, por ejemplo,  = .1. Sigamos trabajando la sucesión como una función para poder superar n (eventualmente) más allá de N = 20:

El comando **while** (= mientras) nuevamente genera un "loop" que nos puede ayudar:

> **d:=eu-.1;**

> **n:=1: while Eu(n)<d do n:=n+1: end do: print(n);**

Así,  = 13. En efecto:

> **N0:=13: evalf(Eu(N0-1)); evalf(Eu(N0));**

Graficando la idea de límite para este ejemplo:

> **n:='n':**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,Eu(n)]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **F1:=implicitplot(y=eu,x=0..N,y=2..3,color=blue,thickness=2):**

> **F2:=implicitplot(y=eu+.1,x=0..N,y=2..3):**

> **F3:=implicitplot(y=eu-.1,x=0..N,y=2..3):**

> **F4:=implicitplot(x=N0,x=0..N,y=2..3,color=black,linestyle=3):**

> **t1:=textplot([[13,2,`N0`]],font=[TIMES,BOLD,10],color=black):**

> **display([t1,F1,F2,F3,F4,seq(p||n,n=1..N)],view=[0..N,2..3],title="\n La convergencia de la sucesión de Euler",titlefont=[TIMES,BOLD,14]);**

Comprobando un importante límite:

> **n:='n': Limit(n\*sin(1/n),n=infinity)=limit(n\*sin(1/n),n=infinity);**

Consideremos ahora la sucesión:

> **r:=[seq((-3/4)^n,n=1..N)]:**

> **n:='n':**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,r[n]]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **display(seq(p||n,n=1..N));**

El gráfico sugiere que la sucesión converge a cero. En efecto:

> **n:='n': Limit((-3/4)^n,n=infinity)=limit((-3/4)^n,n=infinity);**

Otro importante ejemplo es el de sumar los términos de una sucesión:

> **S:=sum(k\*k!,k=0..N);**

> **s:=sum(k\*k!,k=0..n);**

¿Dónde radica la diferencia entre S y s?.

En un texto se afirma que la sucesión  tiene un límite entre  y 1. Veamos que dice el Maple:

> **s:=n->sum(1/k,k=n..(2\*n));**

> **s(n+1)-s(n);**

Obviamente esta respuesta no nos ayuda a estudiar si es una sucesión monótona. Pero:

> **limit(s(n),n=infinity);**

> **evalf(%);**

Veamos ahora un ejemplo de aplicación del criterio del sandwich:

> **j:=k->1/(n^2+k);**

> **H:=n->sum(j(k),k=1..n);**

> **simplify(j(k)-j(k+1));**

Este resultado muestra que j(k) es decreciente y luego: n·j(n) < H(n) < n·j(0), y como:

> **n\*j(n); n\*j(0);**

> **limit(n\*j(n),n=infinity); limit(n\*j(0),n=infinity);**

entonces:  = 0. En efecto:

> **limit(H(n),n=infinity);**

> **n:='n':**

> **for n to N do if n mod 2=0 then cl:=blue else cl:=red: end if: p||n:=plot([[[n,H(n)]]$3],style=point,symbol=[circle,diamond,cross],color=cl): end do:**

> **display(seq(p||n,n=1..N));**

Veamos finalmente algunos ejemplos:

1) Calcular: 

> **restart:**

> **Limit((n+1)\*(n+2)\*(n+3)/(n^3),n=infinity)=limit((n+1)\*(n+2)\*(n+3)/(n^3),n=infinity);**

2) Si 0 < , demostrar que la sucesión de término general:  converge a *a*

> **restart:**

> **assume(0<b): additionally(b<=a):**

> **Limit((a^n+b^n)^(1/n),n=infinity)=limit((a^n+b^n)^(1/n),n=infinity);**

En esta forma el programa no fue capaz de calcular el límite propuesto; pero, en la forma equivalente:

> **Limit((a^n+b^n)^(1/n),n=infinity)=limit(a\*(1+(b/a)^n)^(1/n),n=infinity);**

3) Calcular: , , , 

> **restart:**

> **Limit((n)^(1/n),n=infinity)=limit((n)^(1/n),n=infinity);**

> **Limit((-1)^n\*cos(n)/n^2,n=infinity)=limit((-1)^n\*cos(n)/n^2,n=infinity);;**

> **Limit((sin(n)/3^n,n=infinity))=limit((sin(n)/3^n,n=infinity));**

4) Calcular: 

> **restart:**

> **Limit(n\*((8+2/n)^(1/3)-2),n=infinity)=limit(n\*((8+2/n)^(1/3)-2),n=infinity);**

> **radnormal(%);**

5) Calcular: 

> **restart:**

> **Limit((1+1/(n+1))^(4\*n),n=infinity)=limit((1+1/(n+1))^(4\*n),n=infinity);**

6) Calcular: 

> **restart:**

> **Limit(((n^2+3)/(n^2+4\*n))^((n^2-1)/n),n=infinity)=limit(((n^2+3)/(n^2+4\*n))^((n^2-1)/n),n=infinity);**

7) Decidir si existe el límite para la sucesión definida por la recurrencia: 

> **restart:**

> **G:=rsolve({g(n+2)=(5\*g(n+1)-g(n))/6,g(1)=1,g(2)=6},g);**

> **G:=unapply(G,n);**

> **evalf(G(1610));**

> **n:='n': limit(G(n),n=infinity);**

8) Demuestre que:  = 1

> **restart:**

> **Limit(sum(1/sqrt(n^2+k),k=1..n),n=infinity)=limit(sum(1/sqrt(n^2+k),k=1..n),n=infinity);**