Sobre las rutinas Adapt.

Adapt

La integración por cuadratura adaptativa es un algoritmo sumamente interesante. Se basa en integrar una función a través del uso de reglas de cuadratura (para solo evaluar la función) comprobando el error en cada iteración, si éste resulta ser más grande que cierta tolerancia permitida definida (relacionada con el error absoluto y relativo) entonces se subdivide el intervalo y se recalcula la integral, hasta satisfacer el criterio de tolerancia.

A continuación, se adjuntan los códigos de la función `adapt`, misma que necesita las funciones auxiliares de `add` (auxiliar para añadir elementos a la cola de proceso) y `quad` (auxiliar para calcular la regla de cuadratura en un intervalo).

```
function [answer,errest,flag,nfe] = Adapt(f,a,b,abserr,relerr)
%  Estimate the definite integral of f(x) from a to b using an
%  adaptive quadrature scheme based on Gauss-Kronrod (3,7) formulas.
%  REQUIRES Add.m and Quad.m.
%
% Input parameters:
%    f     = name of the function defining f(x).
%   a, b   = end points of integration interval.
%   abserr = absolute error tolerance desired.
%   relerr = relative error tolerance desired.
%
% Output parameters:
%   answer = computed estimate of the integral.
%   errest = estimate of the absolute error in answer.
%   flag   = 0  for normal return;
%          = 1  insufficient storage in queue.
%          = 2  too many function evaluations.

%Número máximo de elementos en la cola y creación de la Queue
maxq = 120;
queue = zeros(maxq,4);
%Número máximo de evaluaciones de la función
maxfe = 3577;

%  Test input data.
if relerr < 10*eps
    error('relerr < 10*eps is not allowed in Adapt.')
end
if abserr < 0
    error('abserr <= 0 is not allowed in Adapt.')
end

%  Initialization.
%Inicialización de la cola y otros parametros
length = 0;
top = 1;
bottom = 1;
nfe = 0;

%  Form an initial approximation answer to the integral over [a,b].
```

```
%  If it is not sufficiently accurate, initialize the queue and
%  begin the main loop.

%Calculamos una aproximación inicial para comenzar el proceso
[q,e,nfe] = Quad(f,a,b,nfe);
%Guardamos la respuesta global y el error global
answer = q; errest = e;

%Si nuestra aproximación inicial es menos exacta de lo que pedimos
if abs(errest)>max(abserr,relerr*abs(answer))
    %Comenzamos el proceso metiendo a la cola para procesar
    [queue,length,bottom] = Add(queue,q,e,a,b,length,bottom);
end

%  Main loop; if queue is empty then return, else subdivide the top entry.
%Mientras haya elementos en la cola ...
while length > 0
    %  Remove the top entries from the queue.
    q = queue(top,1);
    e = queue(top,2);
    alpha = queue(top,3);
    beta =  queue(top,4);
    %Como sacó un elemento decrece en 1
    length = length - 1;
    %Forma de moverse condicionada para que no haya error de indice
    if top < maxq
        top = top + 1;
    else
        top = 1;
    end

    %Calcula el centro del intervalo [alpha,beta]
    h = (beta-alpha)/2;
    %Calcula la integral en los intervalos [alpha,mitad],[mitad,beta]
    [ql,el,nfe] = Quad(f,alpha,alpha+h,nfe);
    [qr,er,nfe] = Quad(f,alpha+h,beta,nfe);

    %  Update answer and the error estimate.
    %Actualizamos la respuesta general con el resultado nuevo
    answer = answer + ((ql + qr) - q);
    errest = errest + ((el + er) - e);

    %  Test for failures.
    %Se acabó el espacio, demasiados intervalos en la Queue
    if length >= maxq - 1
        flag = 1;
        return;
    end
    %Demasiadas evaluaciones de función
    if nfe >= maxfe
        flag = 2;
        return;
    end

    %  Test for convergence.
```

```
    %Vemos si nuestra aproximación ya es lo suficientemente buena
    tol = max(abserr, relerr * abs(answer));
    if abs(errest) <= tol
        %Si sí, acabamos el proceso
        flag = 0;
        return;
    end

    %Sino, aún no tenemos una respuesta lo suficientemente buena
    %  Add new subintervals to queue if errors are too big.
    tol = tol * h / (b - a);
    %Si el intervalo izquierdo tiene demasiado error, se mete a la Queue
    if abs(el) > tol
        [queue,length,bottom] =
Add(queue,ql,el,alpha,alpha+h,length,bottom);
    end
    %Si el intervalo derecho tiene demasiado error, se mete a la Queue
    if abs(er) > tol
        [queue,length,bottom] =
Add(queue,qr,er,alpha+h,beta,length,bottom);
    end
end

%Si llega aquí terminó todos los elementos de la Queue y lo logra
flag = 0;


function [queue,length,bottom] = Add(queue,q,e,alpha,beta,length,bottom)
%  Add an entry to the end of the queue.

%bottom es el indice donde se guardó el último elemento
[maxq,cols] = size(queue);
queue(bottom,1) = q;
queue(bottom,2) = e;
queue(bottom,3) = alpha;
queue(bottom,4) = beta;
length = length + 1;
%Técnica de almacenamiento eficiente, volver al inicio
if bottom < maxq
    bottom = bottom + 1;
else
    bottom = 1;
end


function [q,e,nfe] = Quad(f,alpha,beta,nfe)
%  A Gauss-Kronrod (3,7) quadrature over (alpha,beta).

%Valores de A_1,A_2,A_3,A_4 y x_1,x_2,x_3
a = [0.2684880898683334, 0.1046562260264672, ...
     0.4013974147759622, 0.4509165386584744];
x = [0.7745966692414834, 0.9604912687080202, 0.4342437493468026];
%Evaluación de la cuadratura de Kronrod (solo las funciones)
h = (beta - alpha)/2;
center = alpha + h;
```

```
f1 = feval(f,center);
f2 = feval(f,center - h * x(1));
f3 = feval(f,center + h * x(1));
f4 = feval(f,center - h * x(2));
f5 = feval(f,center + h * x(2));
f6 = feval(f,center - h * x(3));
f7 = feval(f,center + h * x(3));
%Añadimos 7 evaluaciones de funciones al conteo general
nfe = nfe + 7;
%Calculamos la integral con cuadratura Gaussiana de tres puntos
q = h * (5 * (f2 + f3) + 8 * f1) / 9;
%Calculamos la integral con cuadratura Krondor de siete puntos
qkronrod = h*(a(1)*(f2+f3)+a(4)*f1+a(2)*(f4+f5)+a(3)*(f6+f7));
%El error es aproximadamente la diferencia entre una y otra
e = qkronrod - q;
```

Diego Emilio Dominguez Tableros

5.4 To test out Adapt try the following integrands.

a) $\frac{4}{1+x^2}$ b) $x^{1/10}$ c) $\frac{1}{(3x-2)^2}$

d) $1+\sin^2 38\pi x$ e) $f(x)=\begin{cases}0 & 0\le x\le 0.1\\ 2 & 0.1\le x\le 0.6\\ -1 & 0.6\le x\le 1\end{cases}$ f) $\left|x-\frac{1}{4}\right|^{-1/2}$

Use ABSERR $= 10^{-12}$, RELERR $= 10^{-6}$, and A $= 0$, B $= 1$ for all parts. Which problems require the most function evaluations? Why? What are the exact answers? What are the actual errors? How good an estimate is ERREST? On which problems is it better tan others? Why?

En cada inciso se utilizará `Adapt`, `Integral` y `Quadgk` para una mejor comparación.

a) La integral exacta es

$$\int_0^1 \frac{4}{1+x^2}\,dx = 4\int_0^1 \frac{1}{1+x^2}\,dx = 4[\tan^{-1}x]_0^1 = 4\left(\frac{\pi}{4}-0\right)=\pi$$

Se muestran los resultados obtenidos en la Tabla 1. El resultado se obtiene en solo 21 evaluaciones de funciones ya que la función es suave.
Logra satisfacer la tolerancia del error relativo, obteniendo 5 dígitos de precisión en el resultado.
Comparando con el valor exacto de la integral, se tiene un error absoluto de

$$1.43120695916821\times 10^{-6}$$

que concuerda con `ERREST` en nueve dígitos de precisión, por lo que en este caso `ERREST` es una buena estimación.

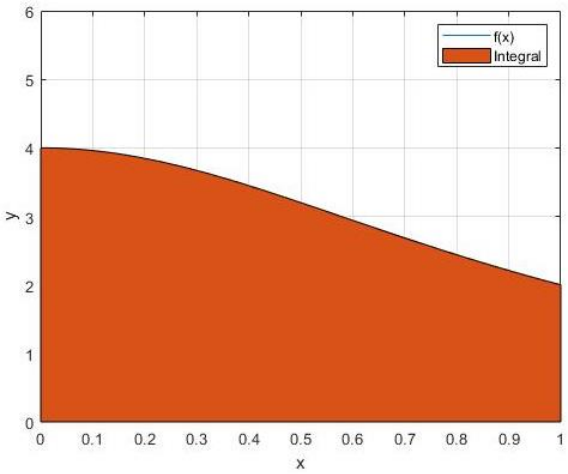


Tabla 1. Resultados para a).

adapt		
Resultado	nfe	ERREST
3.14159122238283	21	$1.43119677220582\times 10^{-6}$
quadgk		**integral**
Resultado	ERRBND	Resultado
3.14159265358979	$1.60982338570648\times 10^{-16}$	3.14159265358979

b) La integral exacta es

$$\int_0^1 x^{1/10}\,dx = \left[\frac{x^{11/10}}{11/10}\right]_0^1 = \frac{10}{11}\left[x^{11/10}\right]_0^1 = \left(\frac{10}{11}-0\right)=\frac{10}{11}$$

Se muestran los resultados obtenidos en la Tabla 2. El resultado se obtiene en 175 evaluaciones ya que la función tiene un aumento súbito de curvatura cerca de 0.
Logra satisfacer la tolerancia del error relativo, obteniendo 5 dígitos de precisión en el resultado.
Comparando con el valor exacto de la integral, se tiene un error absoluto de

$$7.91647373432625 \times 10^{-7}$$

que concuerda con `ERREST` en siete dígitos de precisión, por lo que en este caso `ERREST` es una buena estimación.

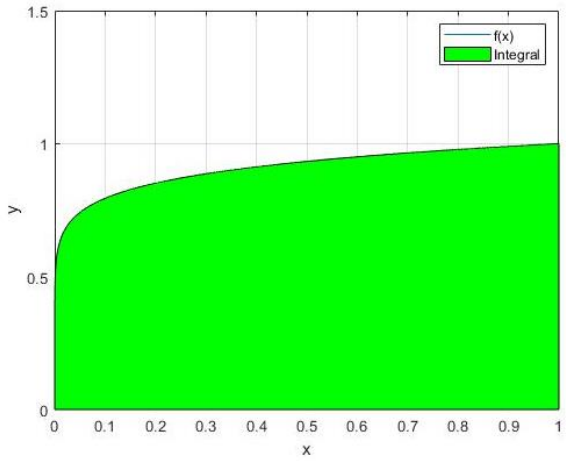


Tabla 1. Resultados para a).

`adapt`		
Resultado	`nfe`	`ERREST`
0.909091700738282	175	$-7.46357388342034 \times 10^{-7}$

`quadgk`		`integral`
Resultado	`ERRBND`	Resultado
0.909090905973262	$2.80582579970795 \times 10^{-7}$	0.909090905973262

c) La integral exacta es $\int \frac{dx}{(3x-2)^2} = -\frac{1}{3(3x-2)}$, sin embargo, en $\frac{2}{3}$ el resultado tiene una singularidad en la que no es integrable, por lo tanto $\int_0^1 \frac{dx}{(3x-2)^2}$ diverge.

Se muestran los resultados obtenidos en la Tabla 3. Como la integral diverge, solo `integral` logra dar el resultado correcto, en cambio `adapt` y `quadgk` dan un resultado muy grande (pues al evaluar cerca de la singularidad, se divide entre valores cercanos a cero), más aún, ambos arrojan errores de número de evaluaciones excedido y número máximo de intervalos excedido, además por las cotas del error podemos decir que fallan al calcular.

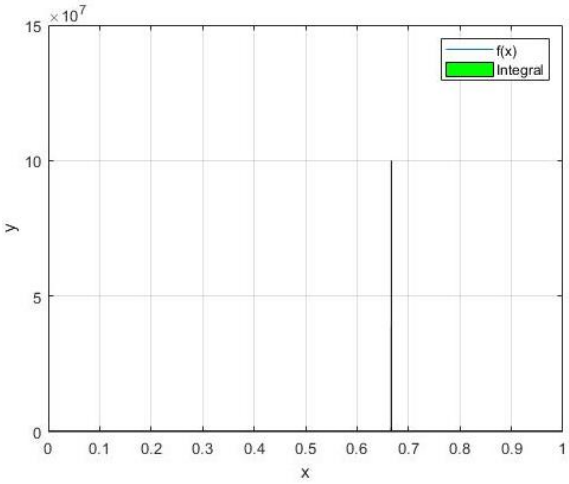


Tabla 3. Resultados para c).

`adapt`		
Resultado	`nfe`	`ERREST`
25,243,266.3418523	3577	64,858,744.0222327

`quadgk`		`integral`
Resultado	`ERRBND`	Resultado
15,511,561,908,600.3	14,695,823,982,672.5	Inf

d) La integral exacta es

$$\int_0^1 1+\sin^2(38\pi x)\,dx = 1+\left[\frac{x}{2}-\frac{\sin(76\pi x)}{152\pi}\right]_0^1 = 1+\frac{1}{2}-0-0+0=\frac{3}{2}$$

Se muestran los resultados obtenidos en la Tabla 4. El resultado se obtiene en 49 evaluaciones de funciones ya que, a pesar de tener oscilaciones, éstas son periódica, por lo que se logra evaluar la cuadratura.
Logra satisfacer la tolerancia del error absoluto. Comparando con el valor exacto de la integral, se tiene un error absoluto de

$$1.998401444325282\times10^{-15}$$

que es mas grande que el `ERREST`, por lo que en este caso el error estimado no es apropiado.

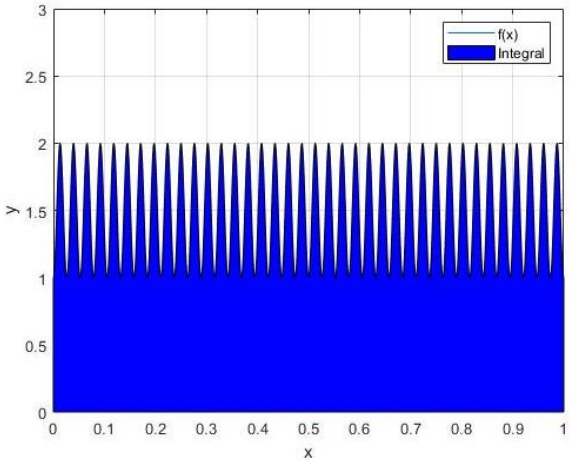


Tabla 4. Resultados para d).

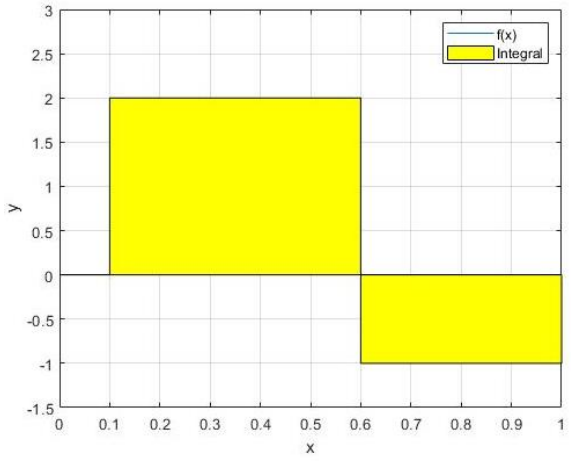
`adapt`		
Resultado	`nfe`	`ERREST`
1.499999999999998	49	$6.106226635438361\times10^{-16}$
`quadgk`		`integral`
Resultado	`ERRBND`	Resultado
1.5	$1.360954426168903\times10^{-8}$	1.5

e) La integral exacta es

$$\int_0^1 f(x)dx = \begin{cases} 0 & 0\le x\le 0.1 \\ 2x & 0.1\le x\le 0.6 \\ -x & 0.6\le x\le 1 \end{cases}, \text{ es decir } \int_0^1 f(x)dx = (0)+(1)+(-0.4)=0.6$$

Como tenemos un polinomio a trozos se utilizaron las funciones ppval y mkpp para introducir la función. Se muestran los resultados obtenidos en la Tabla 5. El resultado se obtiene en 497 evaluaciones de funciones ya que la función no es suave, al ser a trozos tiene un incremento súbito en los puntos.
Logra satisfacer la tolerancia del error relativo.
Comparando con el valor exacto de la integral, se tiene un error absoluto de

$$4.66240776986204\times10^{-7}$$



que concuerda con ERREST en siete dígitos de precisión, por lo que en este caso ERREST es una buena estimación.

Tabla 5. Resultados para e).

adapt		
Resultado	nfe	ERREST
0.599999533759223	497	$4.17682981710829 \times 10^{-7}$
quadgk		integral
Resultado	ERRBND	Resultado
0.599999750791744	$4.58034787092795 \times 10^{-7}$	0.599999750791745

f) Esta integral $\int_0^1 \left|x-\frac{1}{4}\right|^{-1/2} = \int_0^1 \frac{dx}{\sqrt{|x-1/4|}}$ tiene una asíntota vertical en $x = 1/4$, por lo que emplearemos las propiedades de la integral impropia para calcular su valor exacto:

$$\int_0^1 \frac{dx}{\sqrt{|x-1/4|}} = \int_0^{1/4} \frac{dx}{\sqrt{1/4-x}} + \int_{1/4}^1 \frac{dx}{\sqrt{x-1/4}} = \lim_{\varepsilon\to 1/4} \int_0^{\varepsilon} \frac{dx}{\sqrt{1/4-x}} + \lim_{\varepsilon\to 1/4} \int_{\varepsilon}^1 \frac{dx}{\sqrt{x-1/4}}$$
$$= \lim_{\varepsilon\to 1/4} \int_0^{\varepsilon} \left(\frac{1}{4}-x\right)^{-1/2} dx + \lim_{\varepsilon\to 1/4} \int_{\varepsilon}^1 \left(x-\frac{1}{4}\right)^{-1/2} dx$$
$$= \lim_{\varepsilon\to 1/4} \left(-2\sqrt{\frac{1}{4}-\varepsilon} + 2\sqrt{\frac{1}{4}}\right) + \lim_{\varepsilon\to 1/4} \left(2\sqrt{1-\frac{1}{4}} - 2\sqrt{\varepsilon-\frac{1}{4}}\right) = 1+\sqrt{3}$$

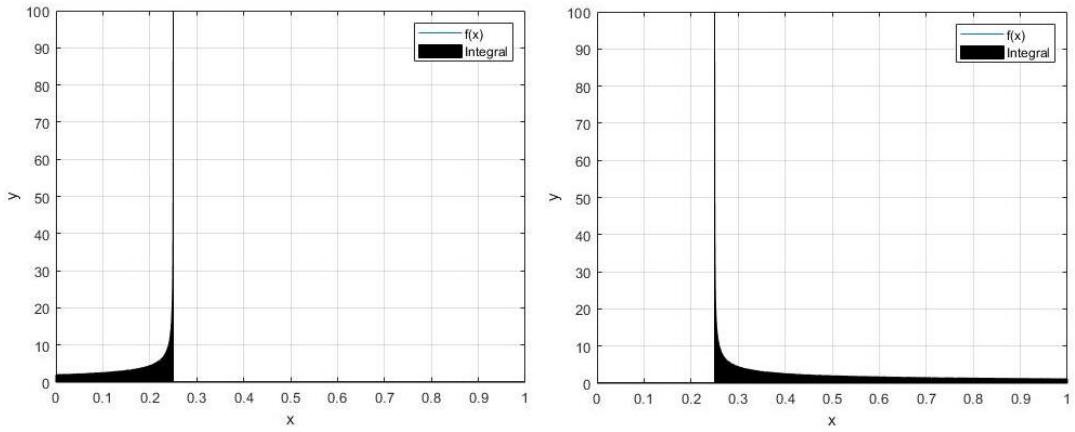
Es muy difícil para el método evaluar esta integral pues la asíntota provoca que el valor se dispare en una vecindad de $x = 1/4$, por lo que optaremos por evaluar la integral dividiendo el intervalo en dos justo en el valor singular, es decir

$$\int_0^1 \left|x-\frac{1}{4}\right|^{-1/2} = \int_0^{1/4} \frac{dx}{\sqrt{1/4-x}} + \int_{1/4}^1 \frac{dx}{\sqrt{x-1/4}}$$

Se obtuvo que $\int_0^{1/4} \frac{dx}{\sqrt{1/4-x}} \approx 0.999998479349704$ (con un ERREST= $9.85235058181257 \times 10^{-7}$) y $\int_{1/4}^1 \frac{dx}{\sqrt{x-1/4}} \approx 1.73204817363348$ (con un ERREST= $1.7065672574462 \times 10^{-6}$) combinando ambos resultados obtenemos

$$\int_0^1 \left|x-\frac{1}{4}\right|^{-1/2} \approx 0.999998479349704 + 1.73204817363348 \approx 2.73204665298318$$

Comparando con el resultado analítico, este resultado tiene un error absoluto de $4.15458569325011 \times 10^{-6}$.




```
%Definimos nuesta tolerancia absoluta y relativa
ABSERR=1e-12; RELERR=1e-6;
%Intervalo de integración [a,b]
a=0; b=1;
%Definimos la función (DESCOMENTAR PARA ELEGIR)
% f=@(x) 4./(1+x.^2);
% f=@(x) x.^(1/10);
% f=@(x) 1./((3.*x-2).^2);
% f=@(x) 1+(sin(38*pi.*x)).^2;
% f=@(x) ppval(mkpp([0,0.1,0.6,1],[0,2,-1]),x);
f=@(x) 1./sqrt(0.25-x);

%Integramos con INTEGRAL
disp('Integral provista por Integral:');
disp(integral(f,a,b,'AbsTol',ABSERR,'RelTol',RELERR));

%Integramos con QUADGK
[res,error]=quadgk(f,a,b,'AbsTol',ABSERR,'RelTol',RELERR);
disp('Integral provista por Quadgk y su cota de error:');
disp([res;error]);

%Integramos con ADAPT
[res,error,flag,no_eval]=Adapt(f,a,b,ABSERR,RELERR);
disp('Integral provista por Adapt:');
disp(res);
disp('Error de Adapt, Bandera y NoDeEvaluaciones hechas:');
disp([error;flag;no_eval]);

%Grafica bonita
equis=[0:0.0001:1]';
plot(equis,f(equis)); hold on; area(equis,f(equis),'facecolor','y');
maxy=max(f(equis)); miny=min(f(equis));
axis([0,1,min(0,miny+0.5*miny),maxy+0.5*maxy]); grid on;
xlabel('x'); ylabel('y'); legend('f(x)','Integral');
```

5.5 The integrand in Exercise 5.4d has period $1/38$. Rewrite this as

$$38\int_0^{1/38} 1+\sin^2(38\pi x)\,dx$$

and use `Adapt` on this with the same tolerances as in Exercise 5.4. Is this approach faster? More accurate?

La integral exacta sigue siendo

$$38\int_0^{1/38} 1+\sin^2(38\pi x)\,dx = \frac{38}{38}+38\left[\frac{x}{2}-\frac{\sin(76\pi x)}{152\pi}\right]_0^{1/38} = 1+\frac{38}{76}+0-0+0=\frac{3}{2}$$

Sin embargo, la integral provista por `adapt` resulta ser exactamente 1.5 con una estimación del error de 0 y calculada con 21 evaluaciones.

La diferencia es clara con respecto al ejercicio 5.4d, esta modificación de la integral nos permite mejorar la exactitud del resultado y el tiempo de ejecución. Podemos hacer esta modificación sin problemas ya que la función es periódica, entonces realmente estamos integrando en un solo periodo y de allí multiplicándolo por el número de periodos en el intervalo [0,1], que son 38. Esto se puede ver más claramente en la Fig. 1.

La conclusión es que, si integramos una función periódica, resulta mucho mas conveniente integrar en un periodo y multiplicarlo adecuadamente, sin importar que este cambio no sea muy útil para dar una respuesta analítica al problema.

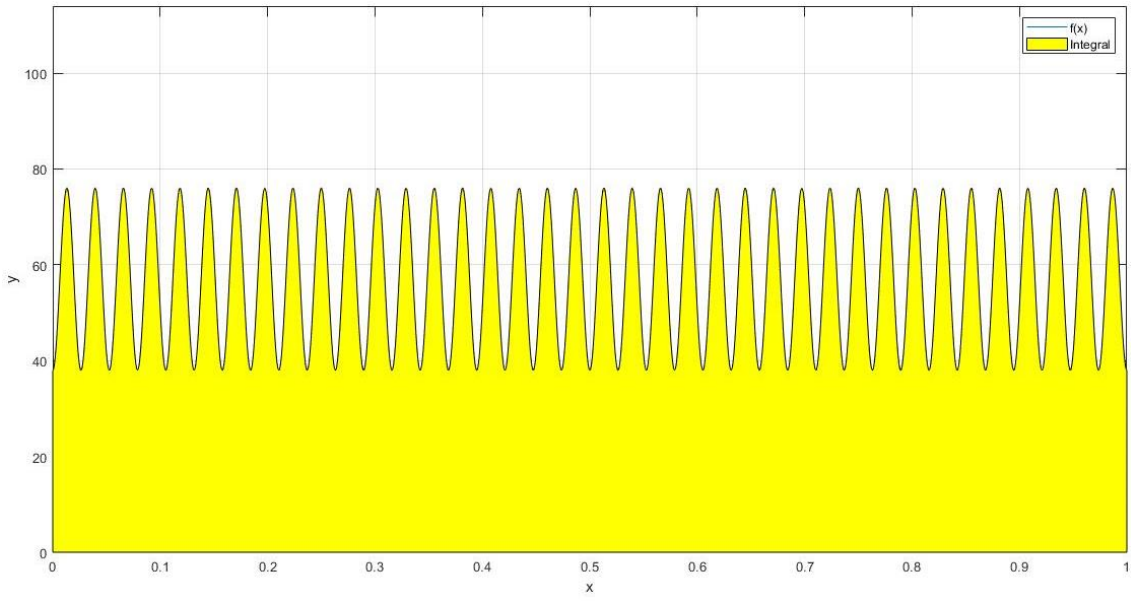


Fig. 1. Gráfica de $1+\sin^2(38\pi x)$ en el intervalo [0,1].

5.6 If a package is not available for evaluating the zeroth order Bessel function $J_0(x)$, then an alternative is based on the formula $J_0(x) = \frac{1}{\pi}\int_0^{\pi} \cos(x \sin\theta)\, d\theta$. Use the formula to evaluate $J_0(x)$ for $x = 1.0, 2.0, 3.0$. Compare with tabulated results (e.g., *Handbook of mathematical functions. Abramowitz & Stegun*).

Utilizando `adapt` se evaluó la integral para cada x dado y se obtuvo el error absoluto con respecto a los valores reportados en la referencia. Las gráficas de las $J_0(x)$ correspondientes se muestran en la Fig. 2 y los resultados se muestran en la Tabla 7.

Tabla 7. Integraciones de la función de Bessel de orden cero para distintos x.

x	Resultado	ERREST	Error con el valor tabulado
1.0	0.765197682326543	$4.23138007965562 \times 10^{-9}$	$4.23142454408776 \times 10^{-9}$
2.0	0.223890779141234	$2.01401395560907 \times 10^{-15}$	$2.41473507855972 \times 10^{-15}$
3.0	−0.260051954903168	$1.23442228661119 \times 10^{-12}$	$1.23495658144179 \times 10^{-12}$

Notemos que los errores absolutos reales corresponden en buena medida con los errores estimados, el mayor es el orden de 4×10^{-9}, por lo que en los tres casos la integración para obtener el valor de $J_0(x)$ es una buena aproximación.

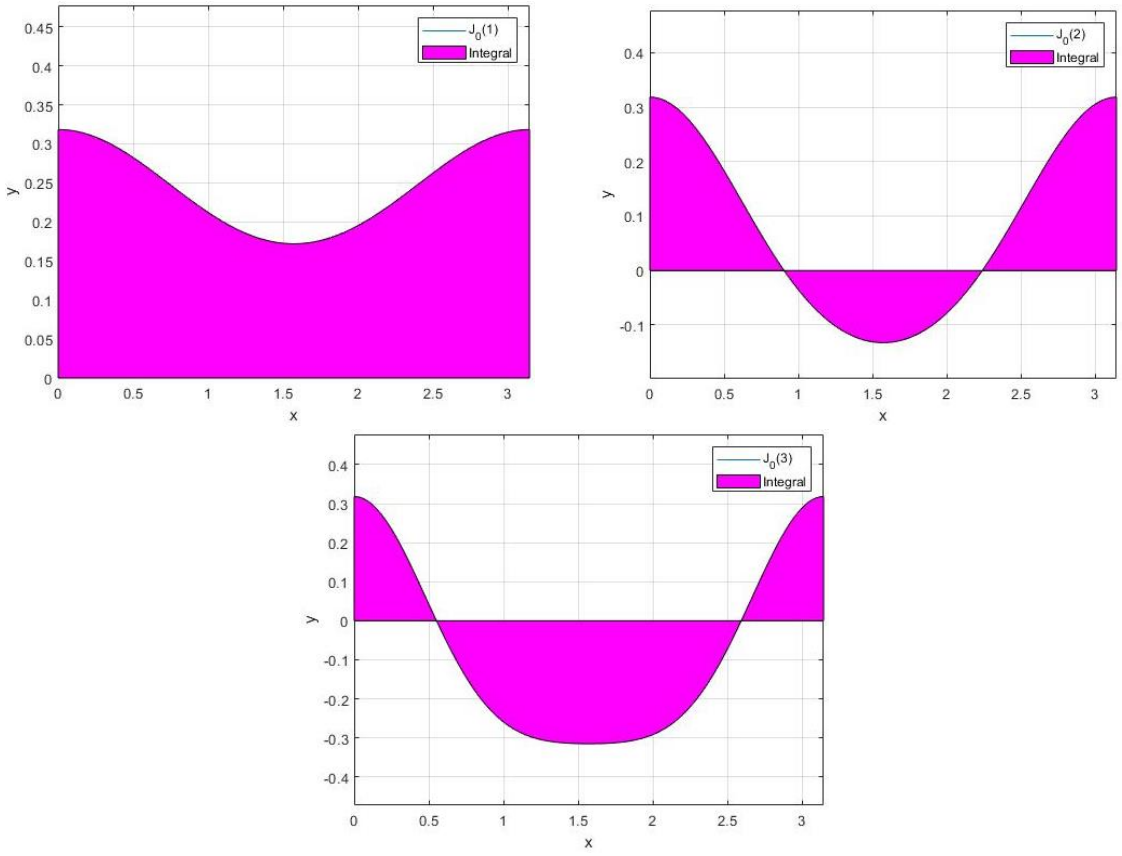


Fig. 2. Funciones de Bessel de orden cero como integrales de la forma $J_0(x) = \frac{1}{\pi}\int_0^{\pi} \cos(x \sin\theta)\, d\theta$.

Diego Emilio Dominguez Tableros

5.7 The function $y(x) = e^{-x^2}\int_0^x e^{t^2}dt$ is called Dawson's integral. Tabulate this function for $x = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5$. To avoid unnecessary function evaluations, split the integral into pieces.

Escribamos cada valor de $y(x)$ como

$$y(0) = 0$$
$$y(0.1) = e^{-(0.1)^2}\int_0^{0.1} e^{t^2}dt$$
$$y(0.2) = e^{-(0.2)^2}\int_0^{0.2} e^{t^2}dt = e^{-(0.2)^2}\left(\int_0^{0.1} e^{t^2}dt + \int_{0.1}^{0.2} e^{t^2}dt\right)$$
$$y(0.3) = e^{-(0.3)^2}\int_0^{0.3} e^{t^2}dt = e^{-(0.3)^2}\left(\int_0^{0.2} e^{t^2}dt + \int_{0.2}^{0.3} e^{t^2}dt\right)$$
$$\vdots$$
$$y(0.5) = e^{-(0.5)^2}\int_0^{0.5} e^{t^2}dt = e^{-(0.1)^2}\left(\int_0^{0.4} e^{t^2}dt + \int_{0.4}^{0.5} e^{t^2}dt\right)$$

Esto nos permite visualizar que podemos ocupar el valor de la integral anterior para solo integrar en un intervalo mas pequeño, es decir $\int_0^x e^{t^2}dt = \int_0^{x-0.1} e^{t^2}dt + \int_{x-0.1}^x e^{t^2}dt$, para $x = 0.1,0.2,\cdots,0.5$, minimizando las evaluaciones. Esto se implemento en Matlab, calcular los valores de las integrales para finalmente multiplicar los resultados por las constantes adecuadas. Los resultados obtenidos se muestran en la Tabla 8 y la gráfica de $y(x)$ en la Fig. 3

Tabla 8. Valores de $y(x) = e^{-x^2}\int_0^x e^{t^2}dt$ para distintos x

x	0	0.1	0.2	0.3	0.4	0.5
$y(x)$	0.0000	0.0993	0.1948	0.2826	0.3599	0.4244

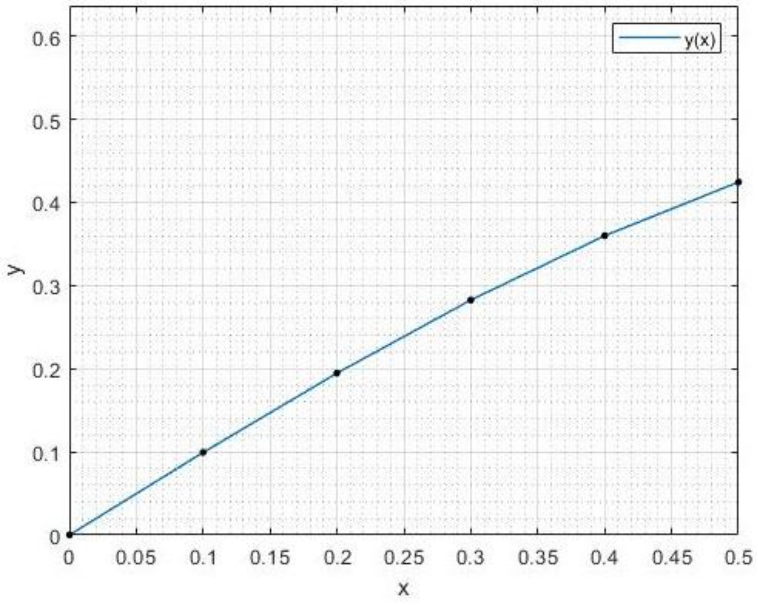


Fig. 3. Gráfica de $y(x) = e^{-x^2}\int_0^x e^{t^2}dt$ en función de x.

```
%Definimos nuesta tolerancia absoluta y relativa
ABSERR=1e-8; RELERR=1e-6;
%Definimos nuestros valores de x=0,0.1,...,0.5
x=0:0.1:0.5; y=x;
%Definimos la función del integrando
f=@(t) exp(t.^2);
%Calculamos las integrales iterativamente
for i=2:length(x)
    %Integramos en [x(i)-0.1,x(i)]
    a=x(i)-0.1; b=x(i);
    %Integramos con ADAPT
    [res,error,flag,no_eval]=Adapt(f,a,b,ABSERR,RELERR);
    %Guardamos y sumamos la integral anterior sobre [0,x(i)-0.1]
    y(i)=res+y(i-1);
end
%Finalmente multiplicamos cada resultado por su correspondiente constante
y=exp(-(x.^2)).*y;
%Mostramos los resultados
disp('Valores de (x,y(x)) calculados'); disp(x); disp(y);

%Grafica bonita
plot(x,y,'linewidth',1); hold on;
plot(x(1:6),y(1:6),'k.','markersize',10,'handlevisibility','off')
maxy=max(y); miny=min(y);
axis([0,0.5,min(0,miny+0.5*miny),maxy+0.5*maxy]); grid on; grid minor;
xlabel('x'); ylabel('y'); legend('y(x)');
```

5.10 A population is governed by a seasonally varying ability of the environment to support it. A simple model of this is the differential equation

$$\boldsymbol{P'(t) = kP(t)\left[M\left(1 - r\cos\frac{\pi}{6}t\right) - P(t)\right]}$$

where t measures time in months, $P(t)$ is the population at time t, and the remaining parameters are known constants. This equation will be solved numerically in the next chapter (Exercise 6.19); here we note that the problem can be transformed into

$$\boldsymbol{P(t) = \frac{P(0)F(t)}{1 + kP(0)\int_0^t F(s)ds}}$$

Where $F(t) = \exp\left(kM\left(t - \frac{6r}{\pi\sin\left(\frac{\pi t}{6}\right)}\right)\right)$. Assume that $k = 0.001$, $M = 1000$, $r = 0.3$, $P(0) = 250$ and use `Adapt` efficiently to table $P(t)$ for $t = 0, 3, 6, 9, \cdots, 36$.

Para obtener los valores de $P(t)$ escribamos con los parámetros dados como

$$P(t) = \frac{250}{1 + 0.25\int_0^t F(s)ds}\exp\left(\mathrm{t} - \frac{1.8}{\pi\sin\left(\frac{\pi}{6}t\right)}\right), \qquad t = 0,3,\cdots,36$$

La única dificultad para evaluar reside en integrar $\int_0^t F(s)ds$ para $t = 0,3,\cdots,36$. Aplicando la técnica del ejercicio anterior podemos hacer mas eficiente y exacta esta integración utilizando integrales calculadas anteriormente y acortando el intervalo de integración de esta manera:

$$\begin{aligned}
&\int_0^0 F(s)ds = 0\\
&\int_0^3 F(s)ds = \int_0^3 F(s)ds + \int_0^0 F(s)ds\\
&\int_0^6 F(s)ds = \int_0^3 F(s)ds + \int_3^6 F(s)ds\\
&\int_0^9 F(s)ds = \int_0^6 F(s)ds + \int_6^9 F(s)ds\\
&\vdots\\
&\int_0^{36} F(s)ds = \int_0^{33} F(s)ds + \int_{33}^{36} F(s)ds
\end{aligned}$$

En pocas palabras, integraremos en intervalos de 3 unidades de longitud y sumaremos el resultado de la integral previa. Una vez obtenidos todos los resultados simplemente multiplicamos por el resto de la función $P(t)$ y tendremos los valores pedidos. Los resultados se muestran en la Tabla 9 y se gráfica $P(t)$ en la Fig. 4.

Tabla 9. Poblaciones en función del tiempo $P(t)$ calculadas.

t	0	3	6	9	12	15	18
$P(t)$	250	715	1236	1111	778	862	1246
t	21	24	27	30	33	36	
$P(t)$	1112	778	862	1246	1112	778	

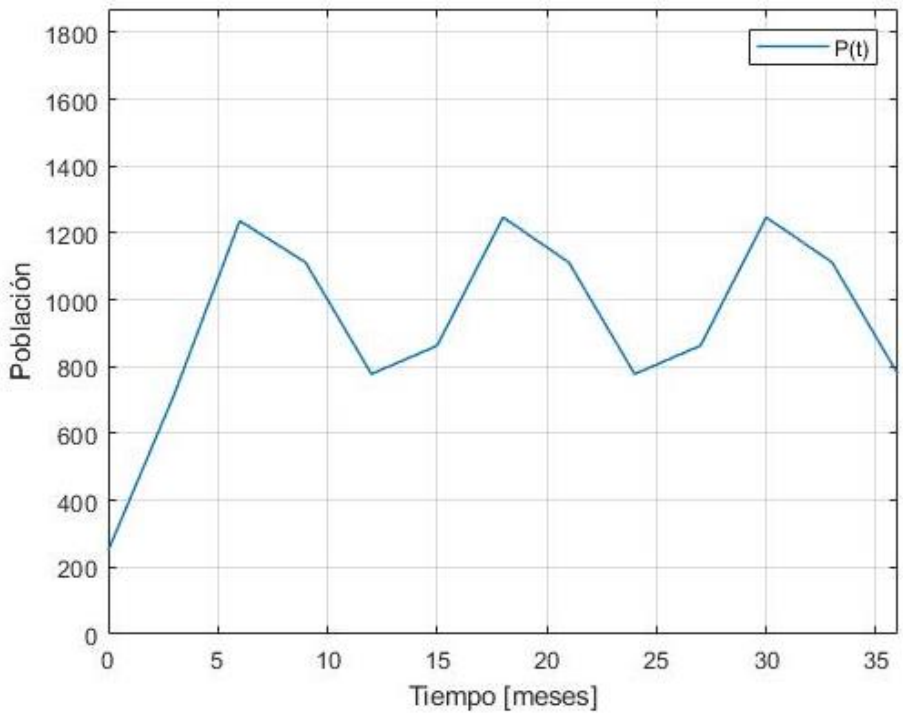


Fig. 4. Población en función del tiempo $P(t)$. Se aprecia su comportamiento cíclico, característico de un modelo poblacional.

```
%Definimos nuesta tolerancia absoluta y relativa
ABSERR=1e-8; RELERR=1e-6;
%Definimos nuestros valores de x=0,3,...,36
t=0:3:36; P=t;
%Definimos las constantes
k=0.001; M=1000; r=0.3; P0=250;
%Definimos la función del integrando
f=@(s) exp(k*M*(s-((6*r*sin(pi*s/6))/pi)));
%Calculamos las integrales iterativamente
for i=2:length(t)
    %Integramos en [x(i)-3,x(i)]
    a=t(i)-3; b=t(i);
    %Integramos con ADAPT
    [res,error,flag,no_eval]=Adapt(f,a,b,ABSERR,RELERR);
    %Guardamos y sumamos la integral anterior sobre [0,x(i)-3]
    P(i)=res+P(i-1);
end
%Finalmente multiplicamos cada resultado por su correspondiente constante
for i=1:length(t)
    P(i)=(P0*f(t(i)))/(1+k*P0*P(i));
end
```

```
%Mostramos los resultados
disp('Poblaciones calculadas para t=0,3,6,...,36'); disp(P');

%Grafica bonita
plot(t,P,'linewidth',1); hold on;
maxy=max(P); miny=min(P);
axis([0,36,min(0,miny+0.5*miny),maxy+0.5*maxy]); grid on;
xlabel('Tiempo [meses]'); ylabel('Población'); legend('P(t)');
```

Command Window

```
>> ejercicio_5_10
Poblaciones calculadas para t=0,3,6,...,36
                        250
           715.03964565608
          1236.48150560969
          1111.82354999511
          778.030072218471
          862.945234482576
          1246.85581590696
          1112.05711215332
          778.040169091272
          862.946331224555
          1246.85588018373
          1112.05711358868
          778.040169153309
```