```matlab
function [Y,X,iter,incr]=Difnolin1582(f,fy,fz,a,b,alfa,beta,n,maxiter,tol)

%
[Y,X,iter,incr]=difnolin('f(x,y,z)','p(x,y,z)','q(x,y,z)',a,b,alfa,beta,n,max
iter,tol)
%
%   Resuelve el problema de contorno de orden 2
%                y" = f(x,y,y')     a<=x<=b
%                y(a) = alfa,  y(b) = beta
%   por el método de diferencias finitas con n subintervalos, aplicando el
%   método de Newton para resolver el sistema no lineal resultante.
%
%   Llamamos z a la derivada de y respecto a x, y'
%   f = f(x,y,z) es la ecuación diferencial
%   p = p(x,y,z) es la parcial de f respecto a y
%   q = p(x,y,z) es la parcial de f respecto a y'= z
%
%   maxiter: máximo de iteraciones de Newton
%   tol: criterio de parada de Newton
%
%   X: vector de nodos
%   Y: valores de la solución
%
%   Ejemplo: para resolver el problema de contorno
%               y" = (32 + 2x^3 -yy')/8                 1<=x<=3
%               y(1)=17,   y(3)=43/3
%   hacemos
```

```
%   difnolin('(32 + 2*x.^3 - y.*z)/8','(-1)*z/8','(-
1)*y/8',1,3,17,43/3,20,20,1e-8)


h=(b-a)/(n+1);  k=(beta-alfa)/(n+1);
X=a:h:b;    Y=alfa:k:beta;
x=X(2:n+1);   y=Y(2:n+1);

incr=tol+1;                  % Para que se haga una iteración al menos
iter=0;

while incr>tol && iter<maxiter
    z=(Y(3:n+2)-Y(1:n))/(2*h);        % Estimaci¢n de la derivada por
                                        % diferencias centrales
    fe=feval(f,x,y,z);
    fye=feval(fy,x,y,z);
    fze=feval(fz,x,y,z);

    a=2+h^2*fye;                          % redefino a, b
    b=-1+h/2*fze(1:n-1);
    c=-1-h/2*fze(2:n);
    d=diff(Y,2)-h^2*fe;
    v=Crout(a,b,c,d);
    y=y+v';
    Y=[alfa y beta];
    incr=max(abs(v));
    iter=iter+1;
end
```

```
end


% Funciones que aparecen como parámetros de entrada
function t=f(x,y,z)
t=(1/8)*(32+2*x.^3-y.*z);
end
function t=fy(x,y,z)
t=-(1/8)*z;
end
function t=fz(x,y,z)
t=-(1/8)*y;
end
```