


```

disp('potencia utilizada el este paso ')
Pahora
pause

while abs(Ptot-Pahora)>=tol % Bucle while que termina cuando la potencia
introducida sea la potencia disponible

    delta=Ptot-Pahora; % Potencia restante
    nivel=nivel+delta/N; % nos acercamos al valor (nivel) optimo

    for i=1:N
        if nivel-sigma_2(i)>0
            vecP(i)=nivel-sigma_2(i);
        else
            vecP(i)=0;
        end
    end

    Pahora=sum(vecP);
    nivel

    %%%%%%%%%%%%%%%
    %%% grafica %%%
    %%%%%%%%%%%%%%%
    cla % Borra la grafica en cada paso
    colormap summer
    bar(1:N,[sigma_2' vecP'],'stack') % Muestra el nivel de ruido de cada
subcanal
    axis([0 N+1 0 max(sigma_2)+4])
    hold on
    plot(0:N+1,nivel*ones(1,N+2),'r--') % Muestra el nivel de waterfilling
introducido
    set(gca,'fontSize',15)
    set(gca,'fontWeight','Bold')
    xlabel('verde=varianza ruido, amarillo=potencia en cada subcanal')
    Title('Waterfilling (discreto)')
    %%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%

    disp('potencia utili zada el este paso ')
    Pahora
    pause

end

disp('distribucion de potencia final en cada subcanal')

```

```
vecP
disp('suma potencia utilizada')
sum(vecP)
disp('potencia total que podemos utilizar - suma potencia utilizada')
disp('(tiene que ser menor (o igual) a la tolerancia)')
abs(Ptot-sum(vecP))
```