Interpolación (Kernel ridge regresión/Krigging) en procesos gaussianos bidimensionales

Presentación en vídeo: http://personales.upv.es/asala/YT/V/krig2dm.html

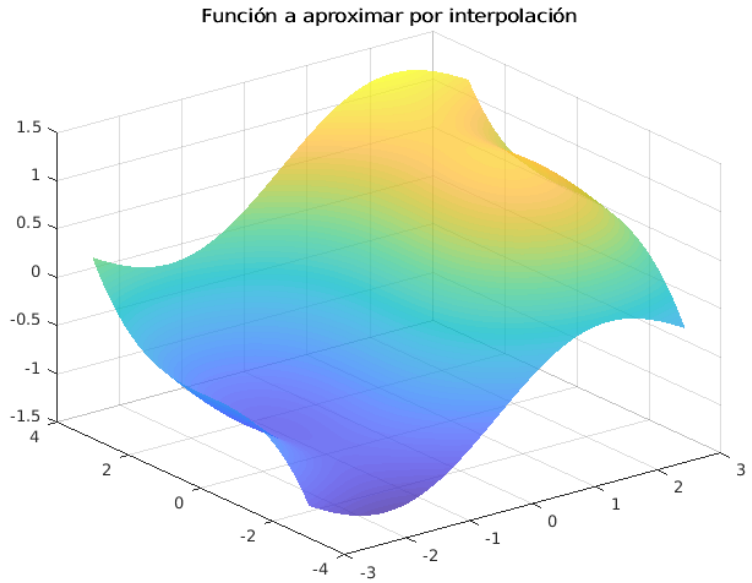
Este código ejecutó correctamente en Matlab **R2018b**

Función "de verdad", suave:

```
modelo=@(G1,G2) sin(0.95*G1).*exp(-G2.^2/15-G1.^2/90)+G1/12+G2.^3/35;
```

Mallado de puntos donde vamos a comprobar la interpolación

```
X1v=-3:0.1:3;
X2v=-3:0.1:3;
[G1v,G2v]=meshgrid(X1v,X2v);
Ymeshv=modelo(G1v,G2v);
N1v=size(X1v,2);N2v=size(X2v,2);
surf(X1v,X2v,Ymeshv)
shading interp
alpha 0.75
title('Función a aproximar por interpolación')
```

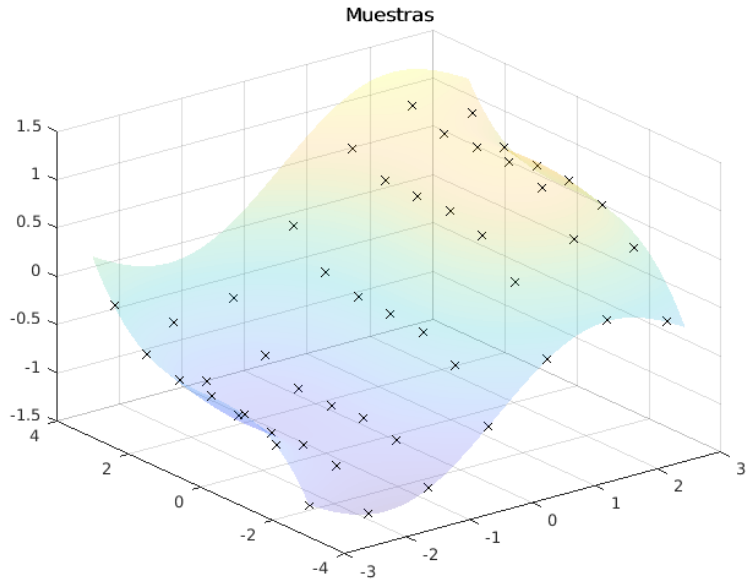


Generación de muestras en unos pocos puntos

```
X1=-3:0.95:3; X2=-3:0.9:3; n=2;N1=size(X1,2);N2=size(X2,2);
[G1,G2]=meshgrid(X1,X2);
Ymesh=modelo(G1,G2);
N=N1*N2
```

```
N =
    49
```

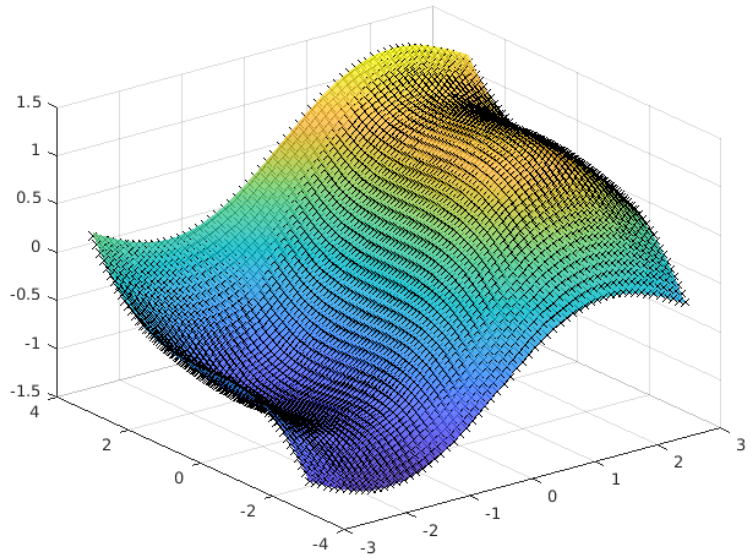
```
std_ruidomed=5e-3;
rng(1234)
Ymesh=Ymesh+randn(size(Ymesh))*std_ruidomed;
surf(X1v,X2v,Ymeshv)
shading interp
alpha 0.2
hold on
plot3(G1,G2,Ymesh,'xk'), grid on, title('Muestras')
hold off
```




```
X=[G1(:) G2(:)]';
Y=Ymesh(:)';
Xv=[G1v(:) G2v(:)]';
Yv=Ymeshv(:)';
vza_ruidomed=std_ruidomed^2;
param=3; %gaussiana 3.0, OK. Cono: 8.3 ok
K=CovMatrix(X,X,param);
Kval=CovMatrix(X,Xv,param);
Yprediccionval=Y/(K+vza_ruidomed*eye(N))*Kval;
errorval=norm(Yv-Yprediccionval)
```

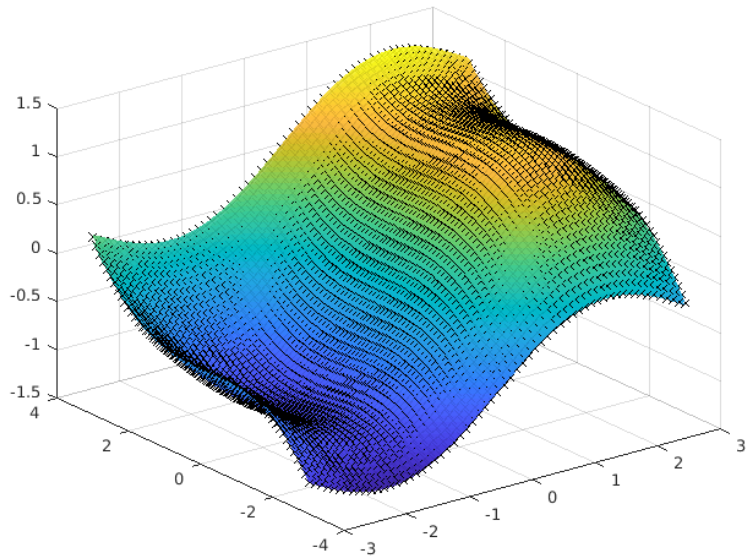
```
errorval =
   0.502113056568180
```

```
YPVM=reshape(Yprediccionval,[N1v,N2v]);
surf(X1v,X2v,Ymeshv); shading interp
alpha 0.8, hold on
plot3(G1v,G2v,YPVM,'xk')
hold off
```



Intervalos de confianza 95% (proceso gaussiano):

```
nm=size(Kval,2);
VVV=ones(nm,1)*K(1,1)-diag(Kval'*inv(K+vza_ruidomed*eye(N))*Kval);
stddd=sqrt(VVV);
DDD=reshape(stddd,[N1v,N2v]);
surf(X1v,X2v,2*DDD+YPVM); shading interp
alpha 0.9, hold on
surf(X1v,X2v,-2*DDD+YPVM); shading interp
alpha 0.9
plot3(G1v,G2v,YPVM,'xk')
hold off
```



```
function K=CovMatrix(X,Z,param)
Nx=size(X,2);
Nz=size(Z,2);
K=zeros(Nx,Nz);
    for i=1:Nx
        for j=1:Nz
            K(i,j)=exp(-norm(X(:,i)-Z(:,j))^2/2/param^2); %Gaussian covariance functio
            %K(i,j)=max( 1-norm(X(:,i)-Z(:,j))/param , 0); %cono de covarianza... venta
        end
    end
end
```