

```

clear
// cas ou "n" est impair
// Attention, ici la periode est prise egale a 2.*%pi
getf('fonct.sci')

pp=1; // essayer en augmentant pp
n=2*pp+1; // n doit etre impair
omega=cos(2.*%pi/n) + %i * sin(2.*%pi/n);

// points d interpolation
for k=0:n-1
    xk(k+1)=k*2.*%pi/n;
    yk(k+1)=fonct(xk(k+1));
end
xk=xk;

// coefficients de Fourier
for j=0:n-1
    cj(j+1)=0.;
    for k=0:n-1
        cj(j+1)=cj(j+1)+yk(k+1)*omega^(-k*j);
    end
    cj(j+1)=cj(j+1)/n;
end
cj=cj;

// erreur de la representation de Fourier
nn=1000;
for p=0:nn
    xx(p+1)=p*2.*%pi/nn;
    yy(p+1)=fonct(xx(p+1));
end
errmx=0.;
for p=0:nn
    zz(p+1)=0.;
    for j=-pp:pp
        jj=j;
        if (j<0) then
            jj=j+n;
        end
        zz(p+1)=zz(p+1)+cj(jj+1)*exp(%i*j*xx(p+1));
    end
    err(p+1)=zz(p+1)-yy(p+1);
    merr=abs(err(p+1));
    errmx=max(errmx,merr);
end
errr=real(err);
errri=imag(err);

xmin=0. ; xmax=+6.29 ;
ymin=-errmx ; ymax=+errmx ;

plot2d(xx,errr,rect=[xmin,ymin,xmax,ymax])
plot2d(xx,errri,rect=[xmin,ymin,xmax,ymax])
xtitle('erreur','x','y')

```