Corrigé partiel de la séance d'exercices n°7

Voici un programme qui traite l'exercice 3. Il est directement inspiré du programme de l'exercice 2. L'écart entre la fonction à approcher et son approximation par splines cubiques est représenté graphiquement. Pour les erreurs commises dans les autres cas on peut constater (par exemple en zoomant) que les interpolations par morceaux sont très supérieures aux interpolations polynomiales. Les valeurs considérées comme exactes sont celles que fournit Matlab quand on évalue `x.*log10(2+sin(x))`.

```
clc; clear;
x=linspace(0,4*pi,17); y=x.*log10(2+sin(x));
t=0:0.05*pi:4*pi; f=t.*log10(2+sin(t));
xentier=1:12; ref=xentier.*log10(2+sin(xentier));
plot(t,f,'r-','LineWidth',2);
axis([0 15 -1 6]); grid on; hold on; pause
% Interpolation linéaire par morceaux
plot(x,y,'gO:','LineWidth',2); pause
% Interpolation parabolique par morceaux
% Il y a 17 points, donc 8 paraboles
for i=1:8
    ideb=2*i-1; imil=ideb+1; ifin=ideb+2;
    xp=x(1,ideb:ifin); yp=y(1,ideb:ifin);
    tp=linspace(xp(1),xp(3),100);
    coeff_p2=polyfit(xp,yp,2); val_p=polyval(coeff_p2,tp);
    plot(tp,val_p,'k','LineWidth',2); pause
    plot(xp,yp,'bO','LineWidth',2); pause
end
% Interpolations polynomiales de degrés 4, 5, 6, 7, et 16
couleur=['k' 'b' 'g' 'c' 'm']
cas=0;
for degre=[4 5 6 7 16]
    cas=cas+1;
    coeff_pdegre=polyfit(x,y,degre);
    fdegre=polyval(coeff_pdegre,t);
    plot(t,fdegre,couleur(cas),'LineWidth',1); pause
end
% Interpolation par splines cubiques
yspl=spline(x,y,t); plot(t,yspl,'b','LineWidth',2); pause
% On refait quelques dessins pour que ce soit plus clair
figure(2)
plot(t,fdegre,'m-','LineWidth',2); hold on, grid on; pause
plot(x,y,'O','LineWidth',2); pause
plot(t,f,'r-','LineWidth',2); pause
plot(t,yspl,'b','LineWidth',2)
% Visualisons l'écart entre l'approximation par splines et les
% valeurs « exactes »
figure(3)
plot(t,f-yspl); hold on; grid on; pause; plot(t,f,'r')
```