

```

function homework2(part) switch part case 1 exo1_fit(); end end function exo1_fit() x = [-2.063
-0.471 0.882 -1.316 -0.101 1.687 -1.102 0.257 2.014]; y = [3.858 0.209 0.771 2.369 -0.214
2.361 1.091 0.139 3.857]; dy = [0.176 0.560 0.085 0.074 0.165 0.073 0.052 0.281 0.196]; [x id]
= sort(x); for j=1:length(y) y2(j) = y(id(j)); end y = y2; %----- %
%SOLUTION SANS TENIR COMPTE DE LA DEVIATION DY %
%----- %Solution
pour fit linéaire %----- %On veut fitter ici la fonction  $y = a + bx \rightarrow A^*(a,b) = y$  avec  $A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots \end{pmatrix}$  for i=1:length(x) A_lin_old(i,1) = 1; A_lin_old(i,2) = x(i); end
%Pseudo inverse de A A_inv = pseudoinverse(A_lin_old); %La solution est donnée par sol_lin
= A_inv*y'; %----- %Solution pour fit carré %----- %On veut
fitter ici la fonction  $y = a + bx + cx^2 \rightarrow A^*(a,b) = y$  avec  $A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots \end{pmatrix}$  for
i=1:length(x) A_square_old(i,1) = 1; A_square_old(i,2) = x(i); A_square_old(i,3) = x(i)^2; end
%Pseudo inverse de A A_inv = pseudoinverse(A_square_old); %La solution est donnée par
sol_square = A_inv*y'; %----- %Solution pour fit exp %-----
%On veut fitter ici la fonction  $y = a + b\exp(-x) + c\exp(x) \rightarrow A^*(a,b) = y$  avec  $A = \begin{pmatrix} 1 & \exp(-x_1) & \exp(x_1) \\ 1 & \exp(-x_2) & \exp(x_2) \\ \dots \end{pmatrix}$  for
i=1:length(x) A_exp_old(i,1) = 1; A_exp_old(i,2) = exp(-x(i)); A_exp_old(i,3) = exp(x(i)); end
%Pseudo inverse de A A_inv = pseudoinverse(A_exp_old); %La solution est donnée par
sol_exp = A_inv*y'; %----- %Plot des solutions
%----- %On veut plotter le résultat pour voir que le fit marche bien figure;
subplot(2,1,1); errorbar(x,y,dy,'ok'); hold on; plot(x,sol_lin(1)+sol_lin(2)*x,'-r');
plot(x,sol_square(1)+sol_square(2)*x+sol_square(3)*x.^2,'-b');
plot(x,sol_exp(1)+sol_exp(2)*exp(-x)+sol_exp(3)*exp(x),'-g'); legend('data','fit linéaire','fit
carré','fit exp','Location','best'); title('Sans tenir compte de la déviation standard'); xlabel('x');
ylabel('y'); %----- % %SOLUTION EN TENANT
COMPTE DE LA DEVIATION DY % %-----
%----- %Solution pour fit linéaire %----- %On veut
fitter ici la fonction  $y = a + bx \rightarrow A^*(a,b) = y$  avec  $A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots \end{pmatrix}$  for i=1:length(x) A_lin(i,1)
= 1/dy(i); A_lin(i,2) = x(i)/dy(i); end %Pseudo inverse de A A_inv = pseudoinverse(A_lin); %La
solution est donnée par sol_lin_am = A_inv*(y./dy); %----- %Solution pour fit
carré %----- %On veut fitter ici la fonction  $y = a + bx + cx^2 \rightarrow A^*(a,b) = y$ 
avec  $A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots \end{pmatrix}$  for i=1:length(x) A_square(i,1) = 1/dy(i); A_square(i,2) =
x(i)/dy(i); A_square(i,3) = x(i)^2/dy(i); end %Pseudo inverse de A A_inv =
pseudoinverse(A_square); %La solution est donnée par sol_square_am = A_inv*(y./dy);
%----- %Solution pour fit exp %----- %On veut fitter ici la
fonction  $y = a + b\exp(-x) + c\exp(x) \rightarrow A^*(a,b) = y$  avec  $A = \begin{pmatrix} 1 & \exp(-x_1) & \exp(x_1) \\ 1 & \exp(-x_2) & \exp(x_2) \\ \dots \end{pmatrix}$  for i=1:length(x) A_exp(i,1) =
1/dy(i); A_exp(i,2) = exp(-x(i))/dy(i); A_exp(i,3) = exp(x(i))/dy(i); end %Pseudo inverse de A
A_inv = pseudoinverse(A_exp); %La solution est donnée par sol_exp_am = A_inv*(y./dy);
disp('-----') disp('Coefficient interpolation sans
tenir compte de la déviation standard')
disp('-----') disp('coefficient a b pour linéaire
sans tenir compte de dev std'); disp(sol_lin); disp('coefficient a b c pour square sans tenir
compte de dev std'); disp(sol_square); disp('coefficient a b c pour exp sans tenir compte de dev
std'); disp(sol_exp); disp('-----') disp('Coefficient
interpolation amélioré par la déviation standard')
disp('-----') disp('coefficient a b pour linéaire
amélioré par dev std'); disp(sol_lin_am); disp('coefficient a b c pour square amélioré par dev
std'); disp(sol_square_am); disp('coefficient a b c pour exp amélioré par dev std');
disp(sol_exp_am); disp('-----')

```

```

%----- %Plot des solutions %----- %On veut plotter
le resultat pour voir que le fit marche bien subplot(2,1,2); errorbar(x,y,dy,'ok'); hold on;
plot(x,sol_lin_am(1)+sol_lin_am(2)*x,'-r');
plot(x,sol_square_am(1)+sol_square_am(2)*x+sol_square_am(3)*x.*x,'-b');
plot(x,sol_exp_am(1)+sol_exp_am(2)*exp(-x)+sol_exp_am(3)*exp(x),'-g'); legend('data','fit
linaire','fit carré','fit exp','Location','best'); title('En tenant compte de la déviation standard');
xlabel('x'); ylabel('y'); %----- % %CALCUL DES
MACHINS TRUCS KI KARRÉ TRUC BIDULE %
%----- %----- %linaire %-----
ki_lin = 0; eq_lin = A_lin_old*sol_lin_am;%donne ax^2+bx+c for i=1:length(x) ki_lin = ki_lin +
((y(i)-eq_lin(i))/dy(i))^2; end ki_lin_red = ki_lin/7; %covariance [U_lin S_lin V_lin] = svd(A_lin);
W_lin = diag(1./diag(S_lin)); covar_lin = V_lin*W_lin.^2*V_lin;
disp('-----') disp('Valeurs du ki carré ki réduit et
covariance') disp('-----')
disp('-----') disp('Interpolation linéaire') disp('-----') disp('ki pour
linéaire'); disp(ki_lin); disp('ki réduit pour linéaire'); disp(ki_lin_red); disp('covariance pour
linéaire'); disp(covar_lin); %----- %square %-----
ki_square = 0; eq_square
= A_square_old*sol_square_am;%donne ae^x+be^-x+c for i=1:length(x) ki_square = ki_square
+ ((y(i)-eq_square(i))/dy(i))^2; end ki_square_red = ki_square/6; %covariance [U_square
S_square V_square] = svd(A_square); W_square = diag(1./diag(S_square)); covar_square =
V_square*W_square.^2*V_square; disp('-----') disp('Interpolation carré')
disp('-----') disp('ki pour carré'); disp(ki_square); disp('ki réduit pour carré');
disp(ki_square_red); disp('covariance pour carré'); disp(covar_square); %-----
%exp %-----
ki_exp = 0; eq_exp = A_exp_old*sol_exp_am;%donne ax+b for
i=1:length(x) ki_exp = ki_exp + ((y(i)-eq_exp(i))/dy(i))^2; end ki_exp_red = ki_exp/6;
%covariance [U_exp S_exp V_exp] = svd(A_exp); W_exp = diag(1./diag(S_exp)); covar_exp =
V_exp*W_exp.^2*V_exp; disp('-----') disp('Interpolation exponentielle')
disp('-----') disp('ki pour exp'); disp(ki_exp); disp('ki réduit pour exp');
disp(ki_exp_red); disp('covariance pour exp'); disp(covar_exp); end

```