fonctions.c

```c
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "fonctions.h"
#include <stdlib.h>
#include <time.h>

double frand_a_b(double a, double b){
    return ( rand()/(double)RAND_MAX ) * (b-a) + a;
}

double * Y_extraction(double * X, int taille){
    int k0 = 0, k1 = 1000; int l = k0;
    double g, somme = 0;
    int i;
    while(l<k1){
        somme += pow(X[l],1/3);
        l += 1;                              //On prend arbitrairement un pas de 1Hz
    }
    somme = somme/(k1-k0+1);
    g = pow(somme,3);
    printf("g = %lf\n",g);
    double * Y = NULL;
    Y = calloc(taille, sizeof(Y));
    for(i=k0;i<k1;i++){
        *(Y+i-k0) = log(1 + 1/g*(*(X+i-k0)));
        printf("%lf\n", *(Y+i-k0));
    }
    return Y;
    }

double * moving_average(double * Y, double largeur, int taille){
    double * movingA = calloc(taille, sizeof(*movingA));
    int i,j;
    for(i=0;i<=taille;i++){
        if(i-largeur/2 < 0 || i+largeur/2 > taille){
            movingA[i] = Y[i];
        }
        else{
            double localA = 0;
            for(j=i-largeur/2;j<=i+largeur/2;j++){
                localA += Y[j];
            }
            movingA[i] = localA/largeur;
        }
    }
    return movingA;

}

double * Z_calc(double * Y, double * N, int taille){
    int i;
    double * Z = calloc(taille,sizeof(*Z));
    for(i=0;i<taille;i++){
        Z[i] = ((Y[i]-N[i])> 0) ? Y[i]-N[i] : 0;
    }
    for(i=0;i<=80;i++){
        *(Z+i) = 0;
    }
    for(i=930;i<=1000;i++){
        *(Z+i) = 0;
    }
    return Z;
}
```