C Code For Allan Variance

```c
/*  allanvar.c */
/*******************************************************************************
 *  AllanVariance on an array of data x, size N
 *  Return N/2 points into y
 *
 *       The Allan Variance gives us a measure of the optimal number of averages
 *       for a set of data. If only white noise is present, then the noise will be
 *       reduced by the square root of the number of points used for the average. But
 *       if there is 1/f or low frequency drift in the data, then after a certain
 *       number of averages the value is shifted because of the drift of the data.
 *
 *       Display allan variance using a log-log plot. Variance on the Y axis and
 *       number of averages on the X axis.
 *
 *       Reference: "The Limits of Signal Averaging in Atmospheric Trace-Gas
 *       Monitoring by Tunable Diode-Laser Absorption Spectroscopy (TDLAS)"
 *       P. Werle, R. Mucke, F. Slemr, Applied Physics B 57, 131-139 (1993)
 ******************************************************************************/

/*
 * k = size of subgroup, from 1 to N
 * m = number of subgroups = N/k
 */

#include    <stdio.h>
#include    <stdlib.h>
#include    <malloc.h>
#include <string.h>


void allanvar(float *y, float *x, int N) {
        float *A,*B;
        double temp,acc;
        int k,m,s,l;

        A=(float*)malloc(N*4);                              /* space for N floats */
        if (A==NULL) {
                printf ("\nCould not allocate A...");
                exit (1);
                }
        B=(float*)malloc(N*4);                              /* space for N floats */
        if (B==NULL) {
                printf ("\nCould not allocate B...");
                exit (1);
                }

        printf("\n Computing Allan variances");
        for(k=1;k<=N/2;k++) {                       /* k=N/2 means minimum of 2 subgroups */
                m=N/k;                             /* number of subgroups */
                for(s=0; s<m; s++) {                /* average subgroups */
                        acc=0.0;
                        for(l=s*k; l<s*k+k; l++) {
                                acc+=x[l];
                                }
                        A[s]= (float) acc/k;
                        }
                acc=0;
                m--;
                for(s=0; s<m; s++){  /* short term variance, use adjacent groups */
                        temp=(A[s+1]-A[s]);
                        acc+=temp*temp;
                        }
                B[k-1]= (float) acc/(2*m);
                if (k%100 == 0) {
                        printf(".");
                        }
                }
        for(k=0;k<N/2;k++)
                y[k]=B[k];
        free(A);
        free(B);
        }


/*******************************************************************
 *  Main program
 *
 *  This program asks for the name of the source data file. The output file
 *      name is the same as the source file but with an extension of av,
 *      i.e. input: source.dat, output: source.av
 *      The input data file and the output data file are ASCII files
 *
 *      The fread/fwrite commands which are commented out could be used for
 *      reading/writing binary 4 byte floating point data
 *
******************************************************************/
void main(void) {
        FILE *infile,*outfile;
        char in[15], out[15];
        static float x[256000], invalue;
        double sqr,sum;
        int N;
        long int i;


        printf("input file >");
        gets(in);
        i = 0;
        while ((out[i] = in[i]) != '.')
                i++;
        out[++i] = 'a';
        out[++i] = 'v';
        out[++i] = '\0';

        infile=fopen(in,"r");
        fseek(infile,0,0);
        printf("\n Reading data....");
        invalue = 0.0;
        for (i=0; (fscanf(infile, "%f",&invalue) != EOF) && i<256000; i++)
                                                /* Read up to 256,000 points unless EOF */
        {
                printf("%ld) = %f\n",i,invalue);
                x[i] = invalue;
        }
        N=i++;

        printf("\n Data points read = %i",N);
        fclose(infile);
        printf("\n first point: %f",x[0]);
        printf("\n last  point: %f",x[N-1]);
        sqr=0; sum=0;
        for (i = 0; i < N; i ++) {
                if (x[i]==-99)
                        printf ("\n wrong data read.  i = %i  x[i] = %f", i, x[i]);
                sqr+=x[i]*x[i];
                sum+=x[i];
                }
        printf("\nvariance = %f",sqr/N - (sum/N)*(sum/N));

        allanvar(x,x,N);                        /* Compute Allan Variance for data */

        printf("\nfirst allan variance %g",x[0]);
        outfile=fopen(out,"wb");

        /*******************************************
         * Output Allan Variance in two columns.
         * First column the number of averages,
         * second column the allan variance
         ******************************************/
        for (i=0; i<N/2; i++) {
                fprintf(outfile,"%d\t%g\r\n",i+1,x[i]);
        }
        fclose(outfile);
        printf("\n Finished.\n");

        }
```