GSAS in Batch Mode

David Wang, Suresh Babu and Andrew Payzant, March 2001

Metals & Ceramics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37830
USA

GSAS (General Structure Analysis System) is a powerful program, developed by Allen C. Larson and Robert B. Von Dreele at LANL, which allows for various scientific computations. It works well for small amount of data as one can change the refinable parameter individually. However, it would be very tedious and time-consuming if the amount of data is huge (thousands spectra), which now is typical for modern neutron and X-ray instruments.

A small program has been developed at ORNL, by which GSAS can be run in batch mode. The program is in ANSI C, and can be run on PCs with Windows/NT operating system. The knowledge on DOS operation system and GSAS subroutines is pre-required. The flow diagram is shown in Fig. 1. The first step is to create a .bat file with the seed and the first spectrum in the list to be refined. Execute the .bat file then create another .bat with the new seed file and the second spectrum in the list to be refined, so and so, till refine all the spectra in the list.

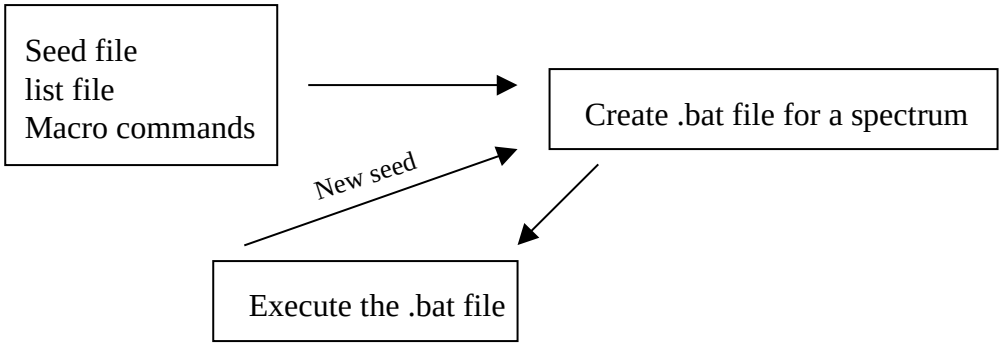

Fig. 1 The flow diagram for the GSAS batch mode program.

The following is a simple instruction and the source code. After being familiar with the code, you can easily modify it to meet your needs.

1. Input files

There are two major files, seed file and list file. The former is the name of the experiment file used as a starter for the sequential spectra refinements, and the later is the list of file name that will be refined.

The second level files are text files that act as a sort of macro commands to feed GSAS subroutines. One has to figure out what input for each GSAS subroutine and create a text file for it.

2. Output files

Except the standard output files from GSAS, the program has two output files, result.lst and the files with extension, int. The refined file name, lattice parameters for two phases and errors, as well as square root of chi are listed in the result.lst file. The files with extension int list all the peak intensities.

3. Source code

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

void main (void)
{
        FILE *file_name_to_refine,*old_exp,*re_exp,
                *gss,*temp_com,*re_lattice,*intensity,*list_file;
        char *seed;
        char *input_file;
        char *refinename;
        char *refinegss;
        char *refineexp;
        char linelength[100];
        char mark[5];
        char descr[6];
        char hname[5];
        char newrecord[100];
        char gsstitle[67];
        char lat1_mark[15],lat2_mark[15];
        char lat1[9],lat2[9],lat1err[9],lat2err[9];
//      char junk1[7],junk2[8],junk3[6],junk4[2];
        char chi_mark[9],chi2[8];
        char inten_mark[10];
        char inten_name[13];
        char list_name[13];
        char com_file_name[38];

//              LaunchExecutable("tempcom.bat");
                seed=(char*)malloc(13*sizeof(char));
                input_file=(char*)malloc(10*sizeof(char));
//Read in the file name as an initial seed experiment file.
```

```
                printf("Type the seed file name without extension:\n");
                scanf("%s", &seed);
//Read in the file name as an initial seed experiment file.
                printf("Type the file name with file lists for refinement:\n");
                scanf("%s", &input_file);
                file_name_to_refine=fopen(input_file,"r");
                re_lattice=fopen("result.lst","w");
                strcpy(mark,"HFIL");
                strcpy(descr,"DESCR");
                strcpy(hname,"HNAM");
                strcpy(lat1_mark,"CRS1  ABC     ");
                strcpy(lat2_mark,"CRS2  ABC     ");
                strcpy(inten_mark,"REFLIST");
                strcpy(chi_mark,"Reduced ");

                strcpy(com_file_name,"tempcom.dat");
                refinename=(char*)malloc(9*sizeof(char));
                refinegss=(char*)malloc(13*sizeof(char));
                refineexp=(char*)malloc(13*sizeof(char));
//Read in the file name to refine
                while(fscanf(file_name_to_refine,"%s", refinename)!=EOF)
                {
                        strcat(seed,".exp");
                        old_exp=fopen(seed,"r");
                        strcpy (refineexp,refinename);
                        strcat (refineexp,".exp");

//get the raw data file name
                        strcpy (refinegss,refinename);
                        strcat (refinegss,".gss");
//Change the gss file location and name in the experiment file
                        re_exp=fopen(refineexp,"w");
                        gss=fopen(refinegss,"r");
                        fgets(gsstitle,sizeof gsstitle, gss);
                        fclose(gss);

                        while (fgets(linelength, 100, old_exp) != NULL)
                        {
//change the description
                                if (strstr(linelength,descr)!=NULL)
                           {
                                        strncat(newrecord,linelength,14);
                                strcat(newrecord,refinename);
                                strcat(newrecord,"\n");
                                   fputs(newrecord,re_exp);
                                strcpy (newrecord, "");
```

```
//change the gss file name
                                }
                                else if (strstr(linelength,mark)!=NULL)
                                {
                                        strncat(newrecord,linelength,14);
                                  strcat(newrecord,refinegss);
                                        strcat(newrecord,"\n");
                                  fputs(newrecord,re_exp);
                                        strcpy (newrecord, "");
                                }

                                else if (strstr(linelength,hname)!=NULL)
                           {
                                        strncat(newrecord,linelength,14);
                                strcat(newrecord,gsstitle);
                                strcat(newrecord,"\n");
                                fputs(newrecord,re_exp);
                                strcpy (newrecord, "");
                           }
                           else
                           {
                                        fputs(linelength,re_exp);
                                }
                                strcpy (linelength, "");
                         }
                        fclose(re_exp);
                        fclose(old_exp);
                        strcpy (seed,refinename);

//do the refinement and only keep the experiment file
                        temp_com=fopen("tempcom.bat","w");
                        fprintf(temp_com,"@echo off\n");
//Change the cycle number to 30, turn on the background, turn off the fraction for phase 2
and h1
                        fprintf(temp_com,"expedt %s < temp1.txt >tempout1.txt\n",
refinename);
                        fprintf(temp_com,"powpref %s < temp.txt >
tempout10\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout2.txt\n",refinename);
//Switch off all background, Change cycle number to 10, switch on the texture for phase
1
                        fprintf(temp_com,"expedt %s < temp2.txt >
tempout3.txt\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout4.txt\n",refinename);
```

```
//Swich off the texture refinement for phase 1 and open for phase 2
                        fprintf(temp_com,"expedt %s < temp3.txt >
tempout5.txt\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout6.txt\n",refinename);
//Switch off the texture refinement for phase 2, switch on the phase fraction for phase 2
and h1
                        fprintf(temp_com,"expedt %s < temp4.txt >
tempout5.txt\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout6.txt\n",refinename);

/*//Switch off the texture refinement for phase 2 and back ground,switch on the profile
refinement for phase 1
                        fprintf(temp_com,"expedt %s < temp4.txt >
tempout7.txt\n",refinename);
                        fprintf(temp_com,"powpref %s < temp.txt >
tempout10\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout8.txt\n",refinename);
//Switch off the profile refinement for phase 1 and on for phase 2
                        fprintf(temp_com,"expedt %s < temp6.txt >
tempout11.txt\n",refinename);
                        fprintf(temp_com,"powpref %s < temp.txt >
tempout10\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout12.txt\n",refinename);
//Switch off the profile refinement for phase 2 and switch on phase fraction for phase 2
                        fprintf(temp_com,"expedt %s < temp7.txt >
tempout13.txt\n",refinename);
                        fprintf(temp_com,"genles %s < temp.txt >
tempout14.txt\n",refinename);*/
                        fprintf(temp_com,"reflist %s < temp5.txt >
tempout9.txt\n",refinename);
//Delete unnecessary files
                        fprintf(temp_com,"del %s.r*\n",refinename);
                        fprintf(temp_com,"del %s.p0*\n",refinename);
                        fprintf(temp_com,"del %s.old\n",refinename);
                        fprintf(temp_com,"del %s.c*\n",refinename);
                        fclose(temp_com);
                        system("tempcom.bat");
//Read the lattice parameter and error from the experiment file
                        re_exp=fopen(refineexp,"r");
                        while (fgets(linelength, 100, re_exp) != NULL)
                        {
                                if (strstr(linelength,chi_mark)!=NULL)
```

```
                                        strncpy(chi2,linelength+30,8);
                                    else if (strstr(linelength,lat1_mark)!=NULL)
                                {
                                        strncpy(lat1,linelength+14,9);
                                        fgets(linelength, 100, re_exp);
                                        strncpy(lat1err,linelength+14,9);
                                    }
                                    else if(strstr(linelength,lat2_mark)!=NULL)
                                    {
                                        strncpy(lat2,linelength+14,9);
                                        fgets(linelength, 100, re_exp);
                                        strncpy(lat2err,linelength+14,9);
                                      break;
                                    }
                                    strcpy(linelength,"");
                                }
                                fprintf(re_lattice,"%s\t%lf\t%lf\t%lf\t%lf\t%lf\n",

        refinename,atof(lat1),atof(lat1err),atof(lat2),atof(lat2err),atof(chi2));
                                fflush(re_lattice);
//put the intensity in a file, inten.txt
                                strcpy(list_name,refinename);
                                strcpy(inten_name,refinename);
                                strcat(inten_name,".int");
                                strcat(list_name,".lst");
                                intensity=fopen(inten_name,"w");
                                list_file=fopen(list_name,"r");
                                while (fgets(linelength, 100, list_file) != NULL)
                                {

                                        if (strstr(linelength,inten_mark)!=NULL)
                                            break;
                                }
                                fputs(linelength, intensity);
                                while (fgets(linelength, 100, list_file)!=NULL)
                                        fputs(linelength, intensity);
                                fclose(list_file);
                                fclose(intensity);
                                temp_com=fopen("tempcom.bat","w");
                                fprintf(temp_com,"del %s\n",list_name);
                                fclose(temp_com);
                                system("tempcom.bat");
                        }
                        free(seed);
                        free(input_file);
                        free(refinename);
```

```
        free(refinegss);
        free(refineexp);
}
```