

Weighting Function Modification in XCIST

Botao Zhao

October 10, 2025

1 Introduction

This document explains how to create a new reconstruction C file in XCIST. In this example, we created a new reconstruction weighting file, `Parallel_Axial_with_partialWeighting.c`. This file was modified from `Parallel_FDK_Helical_3DWeighting.c` and allows users to select the helical scan mode while setting the table speed to zero. In the future, you can also use this process as a reference to implement your own custom weighting functions.

2 Creating New Recon File in C

As mentioned, we changed the weighting function in the `Parallel_FDK_Helical_3DWeighting.c` to allow users to set the table speed to zero, and the modified file was saved as `Parallel_Axial_with_partialWeighting.c`. The modification and file path are shown below:

```
162
163     for(yi=0;yi<YI;yi++)
164     {
165         y = -(yi-YIC)*dy-YOffSet;
166         #pragma omp parallel for private(xi,x, UU, Yr, Zr, UI, U,V1,V, Dey,Dez,touying,weight1,weight2,Gama,Gama_C,m1,m2,weight)
167         for (xi=0;xi<XI;xi++)
168         {
169             x = -(xi-XIC)*dx-XOffSet;
170             UU = -x*cos(View)-y*sin(View);
171             Yr = -x*sin(View)+y*cos(View);
172             Zr = (z-h*(View+sin(Yr/ScanR)))/(2.0*pi)*(DistD)/(sqrt(ScanR*ScanR-Yr*Yr)+UU); //03/05/23 Yu
173             UI = Yr/dY+YIC;
174             U = cell(UI);
175             V1 = Zr/dZ+ZIC;
176             V = cell(V1);
177             Dey = U*UI;
178             Dez = V*V1;
179             //Linear interpolate
180             if ((U>0)&&(U<VL)&&(V>0)&&(V<ZL))
181             {
182                 touying = Dey*Dez*t->GF[U-1][V-1][ProjInd]
183                     +Dey*(1-Dez)*t->GF[U-1][V][ProjInd]
184                     +(1-Dey)*Dez*t->GF[U][V-1][ProjInd]
185                     +(1-Dey)*(1-Dez)*t->GF[U][V][ProjInd];
186             }
187             weight1 = w[d1];
188             weight2 = w[d2];
189             Gama = fabs((z-h*View)/(2.0*pi))/(sqrt(ScanR*ScanR-Yr*Yr)+UU);
190             if (ProjInd < s0)
191             {
192                 Gama_C = fabs((z-h*(View+pi)/(2.0*pi))/(sqrt(ScanR*ScanR-Yr*Yr)-UU));
193             }
194             else
195             {
196                 Gama_C = fabs((z-h*(View-pi)/(2.0*pi))/(sqrt(ScanR*ScanR-Yr*Yr)-UU));
197             }
198             m1 = pow(Gama, k1); //m1 = std::real(std::pow(Gama,k1*h));
199             m2 = pow(Gama_C, k1); //m2 = std::real(std::pow(Gama_C,k1*h));
200             weight = (weight1*m2)/(weight2*m1+weight1*m2);
201             t->RecIm[yi][xi][z]-t->RecIm[yi][xi][z]+weight*touying*DeltaFa;
202             t->RecIm[yi][xi][z+1]-t->RecIm[yi][xi][z+1]+weight*weight2*DeltaFa;
203         }
204     }
205 }
```

Lib > site-packages > gecatsim > reconstruction > src					Search src
Name	Date modified	Type	Size		
Parallel_Axial_with_partialWeighting	10/9/2025 1:55 PM	C Source File	8 KB		
Parallel_FDK_Helical_3DWeighting	Type: C Source File Size: 7.04 KB Date modified: 10/9/2025 1:55 PM	C Source File	7 KB		
Parallel_FDK_Axial_3DWeighting		C Source File	7 KB		
interface_fdk_angle	9/22/2025 8:59 AM	C Source File	6 KB		

3 How to Compile Recon lib on Windows

Download mingw64 at: <https://winlibs.com/#download-release> and unzip it to your desired location. For example: C:\ProgramFiles\mingw64

Download

The following downloads are available (for Windows only).
*You will need a decompressor like [7-Zip](#) (free) to unzip .7z archives, but they are a lot smaller than .zip files.

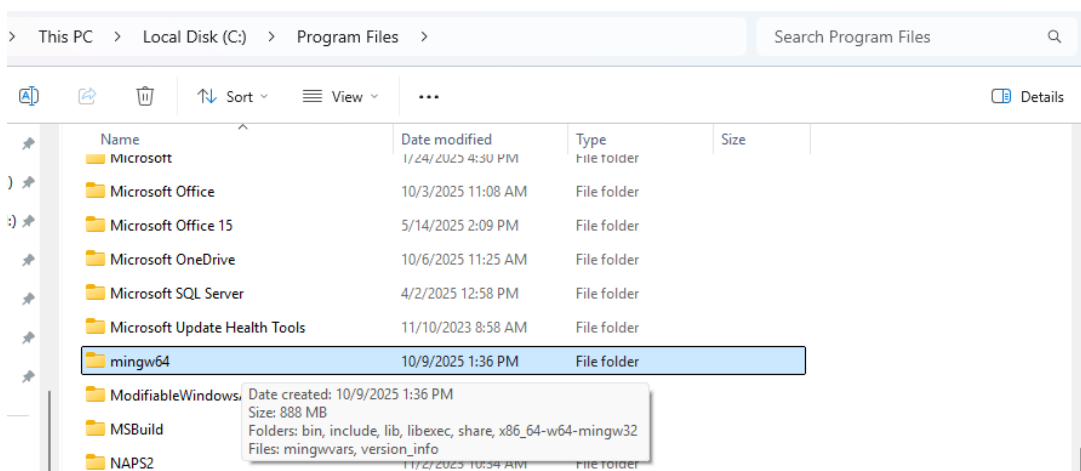
Each version comes in 2 flavors:

- Win32 - i686 - Windows 32-bit version, runs natively on and compiles for Windows 32-bit (also runs on Windows 64-bit, but in 32-bit mode)
- Win64 - x86_64 - Windows 64-bit version, runs natively on and compiles for Windows 64-bit (will not run on Windows 32-bit)

Help! I don't know which download to choose!

Don't worry. For most purposes the latest Windows 64-bit release version with MSVCRT runtime and POSIX threads is a good choice.

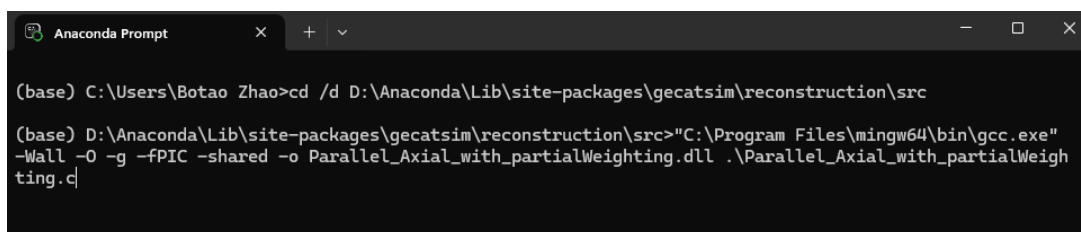
Download it [here](#)



Open the Anaconda Prompt and navigate to the folder containing your C file, for example: D:\Anaconda\Lib\site-packages\gecatsim\reconstruction\src

Then, run the following command in the Anaconda Prompt:

```
"C:\ProgramFiles\mingw64\bin\gcc.exe" -Wall -O -g -fPIC -shared -o Parallel_Axial_with_partialWeighting.dll .\Parallel_Axial_with_partialWeighting.c
```



where "C:\ProgramFiles\mingw64\bin\gcc.exe" is the path of your gcc compiler and "Parallel_Axial_with_partialWeighting" is your new C file name.

After compilation, you should see output messages in the prompt window. Ensure that no errors appear, warnings are generally acceptable.

```

Anaconda Prompt
106 | float x,y,z,Dey,Dez,touying,UU,U1,V1,Beta0,Yr,Zr,View,weight,weight1,weight2,Gama,Gama_C,m1,m2;
.\Parallel_Axial_with_partialWeighting.c:106:60: warning: unused variable 'weight' [-Wunused-variable]
106 | float x,y,z,Dey,Dez,touying,UU,U1,V1,Beta0,Yr,Zr,View,weight,weight1,weight2,Gama,Gama_C,m1,m2;
.\Parallel_Axial_with_partialWeighting.c:50:50: warning: variable 'k1' set but not used [-Wunused-but-set-variable]
50 | float h,h1, BetaE, BetaS, delta, HSCoef, k1;
.\Parallel_Axial_with_partialWeighting.c:50:21: warning: unused variable 'BetaE' [-Wunused-variable]
50 | float h,h1, BetaE, BetaS, delta, HSCoef, k1;
.\Parallel_Axial_with_partialWeighting.c:49:29: warning: unused variable 'startangle' [-Wunused-variable]
49 | float XMC, YNC, ZNC, startangle;
.\Parallel_Axial_with_partialWeighting.c:48:49: warning: variable 'RadiusSquare' set but not used [-Wunused-but-set-variable]
48 | float dx,dy,dz,dYL,dZL,DeltaFai,YLC,ZLC,RadiusSquare,XOffset,YOffset,ZOffset;
.\Parallel_Axial_with_partialWeighting.c:47:57: warning: variable 'dectorYoffset' set but not used [-Wunused-but-set-variable]
47 | float ScanR, DistD,Decl,DecHeight,DecWidth,ObjR,dectorYoffset,dectorZoffset;
.\Parallel_Axial_with_partialWeighting.c:92:29: warning: 'dectorZoffset' is used uninitialized [-Wuninitialized]
92 | ZLC = (ZL-1)*0.5 + dectorZoffset;
.\Parallel_Axial_with_partialWeighting.c:47:71: note: 'dectorZoffset' was declared here
47 | float ScanR, DistD,Decl,DecHeight,DecWidth,ObjR,dectorYoffset,dectorZoffset;
(base) D:\Anaconda\Lib\site-packages\gecatsim\reconstruction\src>

```

Then, you should see a new .dll file was generated in the folder:

Lib > site-packages > gecatsim > reconstruction > src				Search src
<div> <div>🔍</div> <div>🗑️</div> <div>↕️ Sort ▾</div> <div>≡ View ▾</div> <div>⋮</div> </div>				
Name	Date modified	Type	Size	
Parallel_Axial_with_partialWeighting.dll	10/9/2025 4:48 PM	Application extens...	99 KB	
Parallel_Axial_with_partialWeig	Date created: 10/9/2025 4:48 PM Size: 98.9 KB	C Source File	8 KB	
Parallel_FDK_Helical_3DWeighting		C Source File	7 KB	
Parallel_FDK_Axial_3DWeighting	10/9/2025 1:06 PM	C Source File	7 KB	
interface_fdk_angle	9/22/2025 8:59 AM	C Source File	6 KB	

Next, move that .dll file into your lib folder. The folder path should be:
D:\Anaconda\Lib\site-packages\gecatsim\reconstruction\lib

Lib > site-packages > gecatsim > reconstruction > lib				Search lib
<div> <div>🔍</div> <div>🗑️</div> <div>↕️ Sort ▾</div> <div>≡ View ▾</div> <div>⋮</div> <div>Details</div> </div>				
Name	Date modified	Type	Size	
fdk_equiAngle.dll	9/22/2025 8:59 AM	Application extension	64 KB	
fdk_equiAngle.so	9/22/2025 8:59 AM	SO File	13 KB	
fdk_equiAngle_macos.so	9/22/2025 8:59 AM	SO File	33 KB	
helicalrecon.dll	9/22/2025 8:59 AM	Application extension	73 KB	
helicalrecon.so	9/22/2025 8:59 AM	SO File	17 KB	
helicalreconaxial.dll	10/9/2025 1:42 PM	Application extension	99 KB	
Parallel_Axial_with_partialWeighting.dll	10/9/2025 1:55 PM	Application extension	99 KB	
readme	10/9/2025 1:51 PM	MD File	2 KB	

Finally, make sure to update the file name in your recon file. For example the helical mode:
D:\Anaconda\Lib\site-packages\gecatsim\reconstruction\pyfiles\helical_equiAngle.py.

Now you should be able to set table speed to 0:

```

80
81
82 def load_C_recon_lib():
83     # add recon lib path to environment value "PATH" for depending DLLs
84     # # # recon_lib = my_path.find_dir("top", os.path.join("reconstruction", "lib"))
85     # # # my_path.add_dir_to_path(recon_lib)
86
87     # my_path.find_dir doesn't have the key "reconstruction", use the temp solution below:
88     recon_lib = os.path.join(os.path.dirname(os.path.abspath(__file__)), "../Lib")
89
90     # load C/C++ lib
91     ll = ct.cdll.LoadLibrary
92     if os.name == "nt":
93         #lib_file = "helicalrecon.dll"
94         lib_file = "Parallel_Axial_with_partialWeighting.dll"
95     else:
96         lib_file = "helicalrecon.so"
97     clib = ll(os.path.join(recon_lib, lib_file))
98
99     return clib
100

```

