

## Building the Acoustics Toolbox through Windows 11 command prompt using Gfortran compiler

### I. Install MSYS2:

1) You need to get an archive utility that can handle .xz format. I downloaded PeaZip Portable for this task:

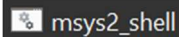
a) <https://peazip.github.io/peazip-64bit.html>

2) Download the MSYS2 base archive from here:

a) <https://www.msys2.org/>

b) I downloaded the latest base archive called msys2-x86\_64-20260322.exe

3) In your (C:\msys64) msys64 folder, double-click on the file msys2\_shell.cmd



4) Now update the package databases:

a) `pacman -Syuu`

b) Type Y when asked if you want to proceed with installation

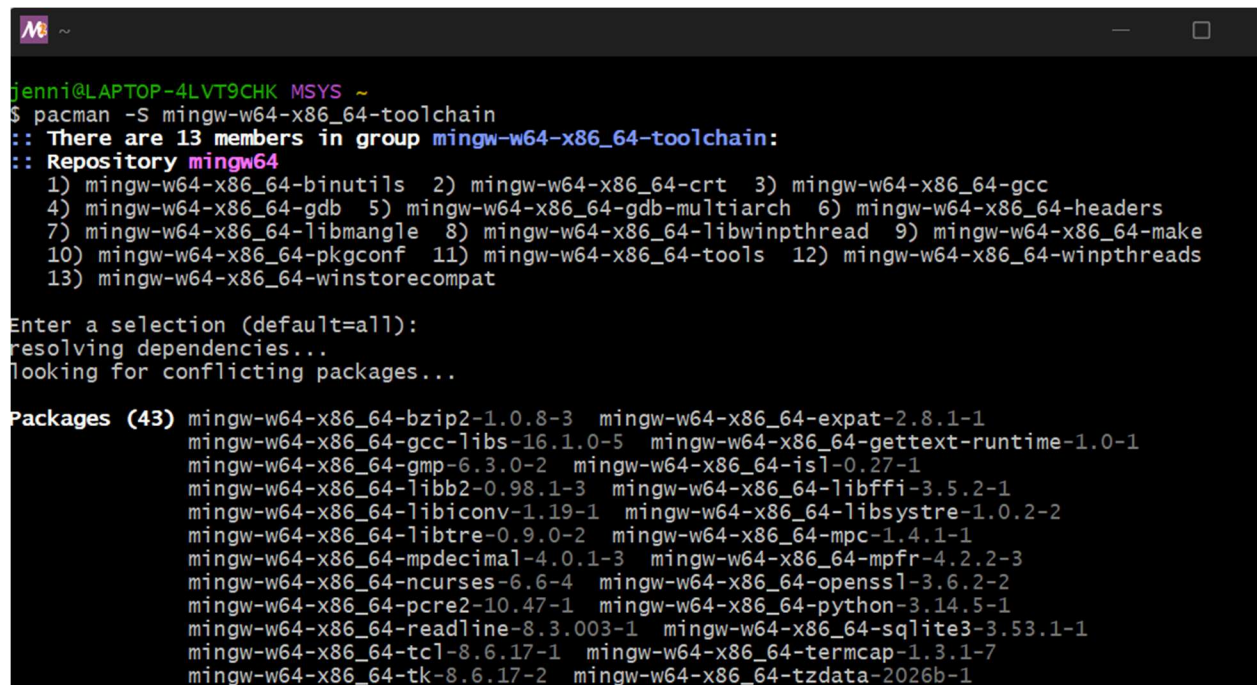
c) Type Y when asked to complete installation

d) Window will close. Reopen shell.

5) We can now use pacman to request the mingw64 toolchain package

a) Type `pacman -S mingw-w64-x86_64-toolchain`

b) Press `enter` to install all packages



```
jenni@LAPTOP-4LVT9CHK MSYS ~
$ pacman -S mingw-w64-x86_64-toolchain
:: There are 13 members in group mingw-w64-x86_64-toolchain:
:: Repository mingw64
 1) mingw-w64-x86_64-binutils 2) mingw-w64-x86_64-crt 3) mingw-w64-x86_64-gcc
 4) mingw-w64-x86_64-gdb 5) mingw-w64-x86_64-gdb-multiarch 6) mingw-w64-x86_64-headers
 7) mingw-w64-x86_64-libmangle 8) mingw-w64-x86_64-libwinpthread 9) mingw-w64-x86_64-make
10) mingw-w64-x86_64-pkgconf 11) mingw-w64-x86_64-tools 12) mingw-w64-x86_64-winpthreads
13) mingw-w64-x86_64-winstorecompat

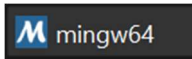
Enter a selection (default=all):
resolving dependencies...
looking for conflicting packages...

Packages (43) mingw-w64-x86_64-bzip2-1.0.8-3 mingw-w64-x86_64-expat-2.8.1-1
mingw-w64-x86_64-gcc-libs-16.1.0-5 mingw-w64-x86_64-gettext-runtime-1.0-1
mingw-w64-x86_64-gmp-6.3.0-2 mingw-w64-x86_64-isl-0.27-1
mingw-w64-x86_64-libb2-0.98.1-3 mingw-w64-x86_64-libffi-3.5.2-1
mingw-w64-x86_64-libiconv-1.19-1 mingw-w64-x86_64-libsystre-1.0.2-2
mingw-w64-x86_64-libtrem-0.9.0-2 mingw-w64-x86_64-mpc-1.4.1-1
mingw-w64-x86_64-mpdecimal-4.0.1-3 mingw-w64-x86_64-mpfr-4.2.2-3
mingw-w64-x86_64-ncurses-6.6-4 mingw-w64-x86_64-openssl-3.6.2-2
mingw-w64-x86_64-pcre2-10.47-1 mingw-w64-x86_64-python-3.14.5-1
mingw-w64-x86_64-readline-8.3.003-1 mingw-w64-x86_64-sqlite3-3.53.1-1
mingw-w64-x86_64-tcl-8.6.17-1 mingw-w64-x86_64-termcap-1.3.1-7
mingw-w64-x86_64-tk-8.6.17-2 mingw-w64-x86_64-tzdata-2026b-1
```

- 9) Once the toolchain is downloaded and installed, there is one last package to install:
  - a) `pacman -S make`
  - b) Type Y when asked if you want to proceed with installation
  - c) Once “make” is installed you can close the shell window (again, close the window by clicking the “x” instead of typing “exit” at the command line).

## II. Installation of Gfortran through MinGW64 app

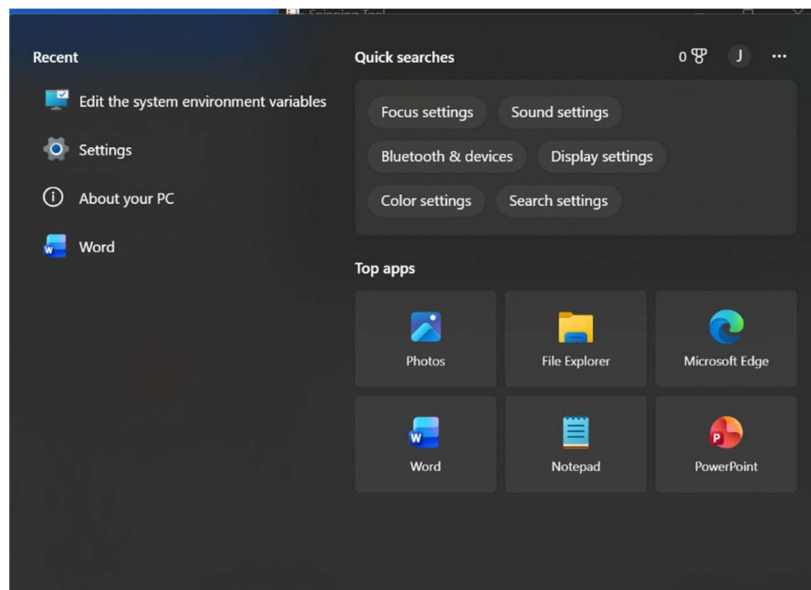
- 1) Now if you want to work with the 64-bit compilers and build environment you double-click on the mingw64 application from the msys64 directory.



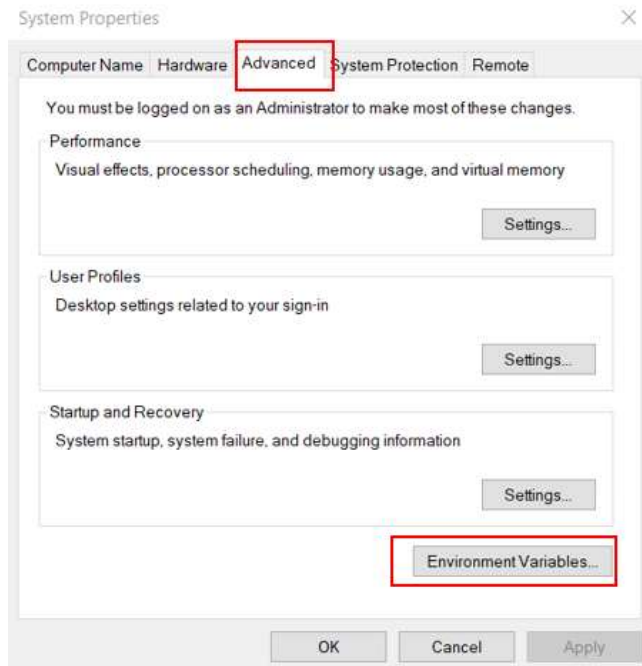
- 2) Once MINGW64 is opened type:
  - a) Type `pacman -S mingw-w64-x86_64-gcc-fortran`
  - b) You can confirm the successful installation with the following command

```
gfortran --version
```

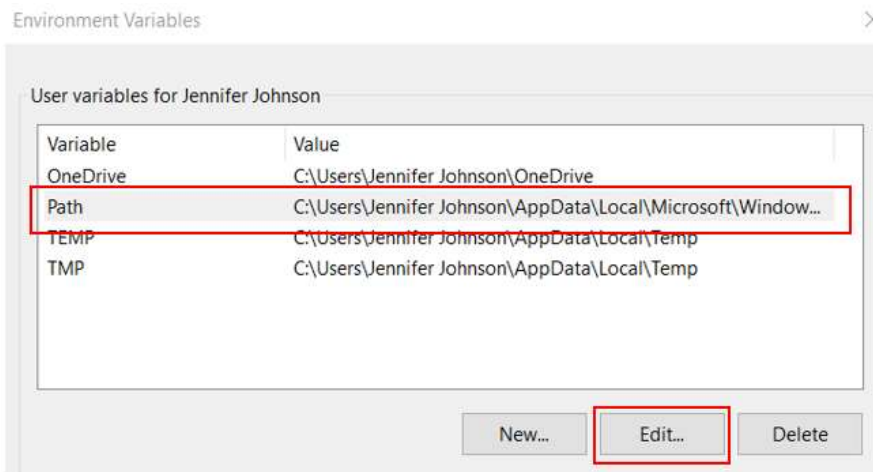
- 3) Once gfortran is installed Windows needs to make sure all the compilers and required binaries were known to the Windows PATH. To add to the PATH:
  - a) Type “Environment Variable” in the START search bar
  - b) Click on “EDIT THE SYSTEM ENVIRONMENT VARIABLES” in the CONTROL PANEL



c) Click “Environment Variables” under the “Advanced” tab.

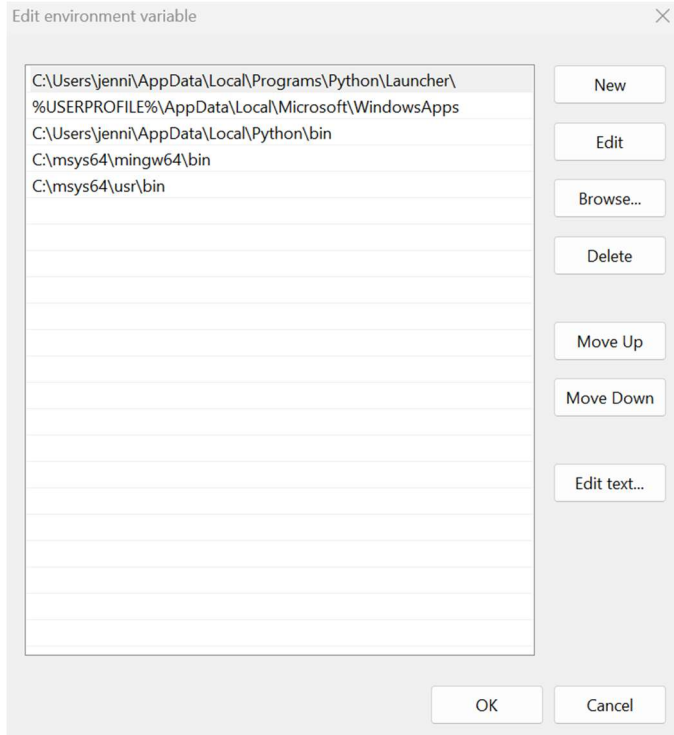


d) Under the “Path” variable click “Edit...”



e) Click “New” and map to “C:\msys64\usr\bin”. The bin folder should have Applications such ‘make’

f) Click “New” and map to “C:\msys64\mingw64\bin”. The bin folder should have Applications such “gfortran”



### III. Prepare AT for WINDOWS Compile

- 1) Download latest AT version (<https://oalib-acoustics.org/models-and-software/acoustics-toolbox/>). Unzip the folder in user home 'C:\Users\jenni'
- 2) Open "Makefile" in at folder with a text editor.
- 3) Prevent compiler warnings for large arrays exceeded size limits

- a) At the end of the 'export FFLAGS' lines under the 'export FC=gfortran' section add  
`-frecursive`

```
export FC=gfortran

export FFLAGS= -march=native      -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -
Wintrinsic-shadow -Wline-truncation -Wa,-q -std=f2008 -O3 -ffast-math -funroll-all-loops -fomit-frame-
pointer -mtune=native -frecursive

export FFLAGS= -march=native      -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -
Wintrinsic-shadow -Wline-truncation -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer
-frecursive
```

- 4) Comment out lines with “Find” commands under the “Clean” section
  - a) When building, it will look for these items, and fail if they do not find.

```
clean:
-rm -f bin/*.exe
# find . -name '*.dSYM' -exec rm -r {} +
# find . -name '*.png' -exec rm -r {} +
# find . -name '*.eps' -exec rm -r {} +
# find . -name '*.mod' -exec rm -r {} +
# find . -name '*.grn' -exec rm -r {} +
# find . -name '*.shd' -exec rm -r {} +
# find . -name '*.shd.mat' -exec rm -r {} +
# find . -name '*.prt' -exec rm -r {} +
(cd misc; make -k -i clean)
(cd tslib; make -k -i clean)
(cd Bellhop; make -k -i clean)
(cd Kraken; make -k -i clean)
(cd KrakenField; make -k -i clean)
(cd Krakel; make -k -i clean)
(cd Scooter; make -k -i clean)
(cd tests; make -k -i clean)
```

- 5) Open “field3d.f90” in C:\Users\jenni\at\KrakenField through a text editor.
  - a) Modify the bolded lines in the file to allow for allocatable arrays.

PROGRAM FIELD3D

! Generate file of replica vectors  
! Multiple receivers depths are handled inefficiently.

USE SourceReceiverPositions  
USE BeamPattern  
USE ElementMod  
USE ReadModes  
USE EvaluateGBMod  
USE Evaluate3DMod  
USE EvaluatepdqMod  
USE RWSHDFFile  
USE FatalError

IMPLICIT NONE

!INTEGER, PARAMETER :: SHDFile = 25, MaxM = 2380, MaxSet = 40000 ! MaxSet = 112000 for Channel Islands  
INTEGER, PARAMETER :: SHDFile = 25, MaxM = 7000, **MaxSet = 200000** ! MaxSet = 112000 for Channel Islands  
INTEGER :: M( MaxSet ), MS( 3 ), Msource, Mlimit, iostat, ifreq = 1  
INTEGER :: iCorner1, isx, isy, isz1, irz1, itheta, NSets, iRec, IElementSource, INode  
REAL (KIND=8) :: freq, freq0, atten = 0  
REAL (KIND=8) :: Rmin, Rmax ! Receiver ranges  
REAL (KIND=8) :: TStart, TEnd

!Moving k and phiR from static declared arrays to ALLOCATEABLE arrays

**!COMPLEX :: k( MaxM, MaxSet ), phiR( MaxM, MaxSet ), kS( MaxM, 3 ), phiST( MaxM, 3 )**

COMPLEX :: kS( MaxM, 3 ), phiST( MaxM, 3 )

CHARACTER (LEN= 7) :: Option

CHARACTER (LEN=80) :: Title, TitleEnv, FileRoot, SHDFFileName, PRTFileName

CHARACTER (LEN=10) :: PlotType = 'TL'

COMPLEX, ALLOCATABLE :: k( :, : ), P( :, : ), phiS( :, :, : ), phiR( :, : )

CALL CPU\_TIME( Tstart ) ! log the start time

! get the file root for naming all input and output files  
! should add some checks here ...

CALL GET\_COMMAND\_ARGUMENT( 1, FileRoot )

SHDFFileName = TRIM( FileRoot ) // '.shd'

PRTFileName = TRIM( FileRoot ) // '.prt'

! Open the print file

OPEN( UNIT = PRTFile, FILE = PRTFileName, STATUS = 'UNKNOWN', IOSTAT = iostat )

WRITE( PRTFile, \* )

WRITE( PRTFile, \* ) '\_\_\_\_\_'

WRITE( PRTFile, \* ) 'Running FIELD3D'

WRITE( PRTFile, \* ) 'Sums modes, producing pressure'

WRITE( PRTFile, \* )

CALL READIN( FileRoot, Title, Option, Mlimit, Rmin, Rmax ) ! Read in all the user input

ALLOCATE( phiS( MaxM, Pos%NSz, 3 ) )

**!New ALLOCATE statement for k and phiR arrays**

ALLOCATE( k( MaxM, MaxSet ), phiR( MaxM, MaxSet ) ) ! Allocated dynamically avoid limit on large matrices (mcmodel=small)

#### IV. Build AT using gfortran

- 1) Open command prompt through START menu
  - a) Type: gfortran
  - b) You should get a fatal error due to no inputs; this means that WINDOWS knows gfortran is there and is looking for input files.

```
Command Prompt
Microsoft Windows [Version 10.0.26200.8246]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jenni>gfortran
gfortran: fatal error: no input files
compilation terminated.

C:\Users\jenni>
```

- 2) cd into AT folder

```
Command Prompt

C:\Users\jenni>cd at

C:\Users\jenni\at>dir
Volume in drive C is OS
Volume Serial Number is ECFE-CFC3

Directory of C:\Users\jenni\at

05/26/2026  01:39 PM  <DIR>          .
05/26/2026  01:39 PM  <DIR>          ..
05/26/2026  01:38 PM                30,724 .DS_Store
05/26/2026  01:38 PM                950 at_init_matlab.m
05/26/2026  01:38 PM  <DIR>          Bellhop
05/26/2026  01:39 PM  <DIR>          doc
05/26/2026  01:38 PM                42,086 index.htm
05/26/2026  01:38 PM  <DIR>          Krakel
05/26/2026  01:39 PM  <DIR>          Kraken
05/26/2026  01:39 PM  <DIR>          KrakenField
05/26/2026  01:38 PM                35,129 LICENSE
05/26/2026  02:10 PM                14,017 Makefile
05/26/2026  01:39 PM  <DIR>          Matlab
05/26/2026  01:38 PM  <DIR>          misc
05/26/2026  01:39 PM  <DIR>          Scooter
05/26/2026  01:39 PM  <DIR>          tests
05/26/2026  01:38 PM  <DIR>          tslib
                5 File(s)                122,906 bytes
                12 Dir(s)    769,539,608,576 bytes free

C:\Users\jenni\at>
```

- 3) Once in the Acoustics Toolbox folder
  - a) Type: `make clean`
  - b) Make clean removes every file in the directory that end with `.exe`. This deletes items that have been created by running the AT, usually from tests packaged with the AT. This leaves only the source code; a clean slate.
- 4) Build Toolbox
  - a) Type: `make`
  - b) The program should be error and warning free. After complete, a message should appear

```

Command Prompt
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib EvaluateGBMod.f90
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib EvaluatepdqMod.f90
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib field3d.f90
gfortran -o field3d.exe -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wli
ne-truncation -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib ReadModes.o EL
ementMod.o Evaluate3DMod.o EvaluateGBMod.o EvaluatepdqMod.o ../misc/libmisc.a field3d.o -L../misc -lmisc
KRAKEN field routines built
*****

make[1]: Leaving directory '/c/Users/jenni/at/KrakenField'
# (cd Krakel; make -k all)
(cd Scooter; make -k all)
make[1]: Entering directory '/c/Users/jenni/at/Scooter'
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib factor_Mod.f90
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib backsub_Mod.f90
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib scooter.f90
gfortran -o scooter.exe -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wli
ne-truncation -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib factor_Mod.o b
acksub_Mod.o scooter.o ../misc/libmisc.a -L../tslib -lts -L../misc -lmisc
gfortran -c -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline-truncatio
n -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib sparc.f90
gfortran -o sparc.exe -march=native -Waliasing -Wampersand -Wsurprising -Wintrinsics-std -Wno-tabs -Wintrinsic-shadow -Wline
-truncation -Wa,-q -std=gnu -O3 -ffast-math -funroll-all-loops -fomit-frame-pointer -frecursive -I../misc -I../tslib factor_Mod.o bac
ksub_Mod.o sparc.o ../tslib/sourceMod.o ../misc/libmisc.a ../tslib/libts.a -L../tslib -lts -L../misc -lmisc
Scooter and Sparc built
*****

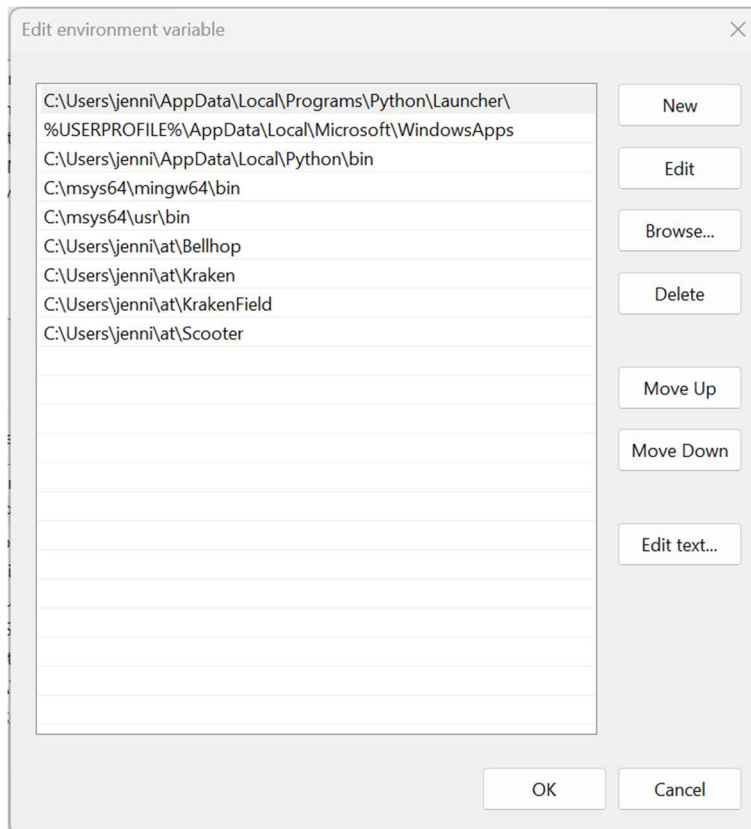
make[1]: Leaving directory '/c/Users/jenni/at/Scooter'

*****
**** Acoustics Toolbox built ****
*****

```

## V. Map Compiled Binaries to WINDOWS PATH

- 1) Open back the “Edit Environmental Variable” window
  - a) Map to each folder in the AT folder that contains “Application” files

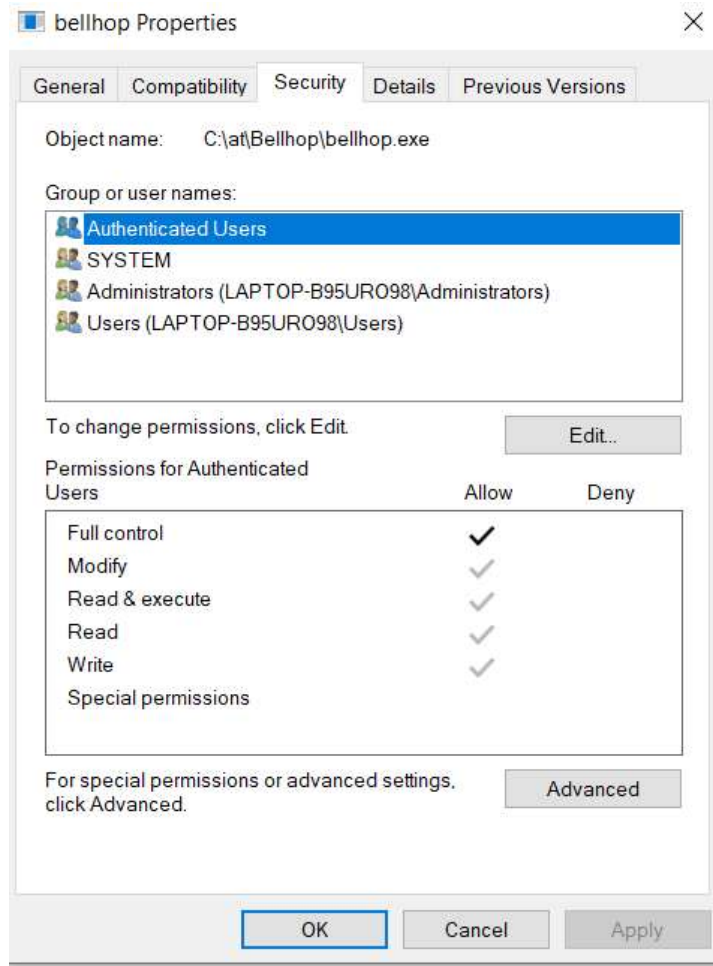


- 2) To test whether WINDOWS sees the compiled binaries
  - a) Open a new command prompt window
  - b) Type: bellhop
  - c) You should receive a "STOP FATAL ERROR" message.
  - d) Run 'runtests.m' in (C:\Users\jenni\at\tests) to see if everything is running correctly. This should run without errors.

**Congrats! You Have compiled the AT on Windows.**

## VI. Miscellaneous Notes

- 1) When running any tests in the “tests” folder and they fail, check the permissions.
  - a) Right click on the APPLICATION of interest (such as bellhop). Click on properties and under the “Security” tab make sure “Full control” is enabled for the designated user.



- 2) Each program in the AT can be called through MATLAB (like a terminal), however it is not recommended to add the AT to your permanent MATLAB PATH in some cases. If accessing the AT often then it could be useful. Otherwise, files/functions with the same name for example “distance” will be defaulted to those in the AT.