Linux

One of the easiest ways to install a Python 3 distribution for iSpec on Linux without root permissions is by using Anaconda.

- Go to the Anaconda Python 3 Distribution site (you can `Skip registration`), pick `Distribution Installer` for `Linux`, download the `64-Bit Installer` and run it from a terminal (adapt the file name to the downloaded version):

```
bash Anaconda3-2025.06-0-Linux-x86_64.sh
```

Follow the instructions, anaconda will be installed in */home/your_user/anaconda3/* by default. The installer will ask if you want to initialize/modify **.bashrc** or **.bash_profile**, select 'yes'. The installer should have added to your *.bashrc* or *.bash_profile* files (in your home directory) something like this:

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/home/your_user/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/your_user/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/your_user/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/your_user/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

This will activate anaconda automatically every time you open a terminal.

- Close all the terminals and open a new one in order to re-load this new setup.

- Install a library that does not come by default and that it is needed by iSpec:

```
conda update conda
conda update --all
conda install lockfile cython
conda install -c conda-forge libgfortran=3 # required library by SME, which is obsolete in modern Linux distributions
```

- On Debian based GNU/Linux distributions such as Ubuntu or Linux Mint, the following packages should be installed by using the package manager (e.g. *sudo apt install package_name*) and its the only step that requires root permissions (fortunately, many servers already include these tools):

```
sudo apt install build-essential gfortran
```

- Download iSpec, and decompress it:

```
curl -O https://www.blancocuaresma.com/s/tmp/scratch/iSpec_v20250801.tar.gz
tar -zxvf iSpec_v20250801.tar.gz
mv iSpec_v20250801 iSpec
```

- Make sure all required Python packages are present in the system:

```
cd iSpec/
pip install -r requirements.txt
```

- Go into the iSpec directory and compile the radiative transfer codes (it will print many warnings, they can be ignored, however if there are errors, then you will not be able to synthesize spectra and determine parameters):

```
cd iSpec/
make clean
make spectrum
make turbospectrum
```

```
make moog
make moog-scat
```

- [OPTIONAL] Download an additional pre-computed grid:

```
cd iSpec/input/grid/
curl -O https://lweb.cfa.harvard.edu/~sblancoc/iSpec/grid/SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
tar -xvf SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
rm -f SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
```

- [OPTIONAL] Download an additional model atmosphere:

```
cd iSpec/input/atmospheres/
curl -O https://lweb.cfa.harvard.edu/~sblancoc/iSpec/atmospheres/MARCS.NLTE.tar
tar -xvf MARCS.NLTE.tar
rm -f MARCS.NLTE.tar
```

- Run the visual interface (GUI) for the first time, which will trigger some additional compilation:

```
cd iSpec/
./iSpec.command
```

Mac

One of the easiest ways to install a Python 3 distribution on MacOSX with most of the needed libraries for iSpec is by using Anaconda.

- Go to the Anaconda Python 3 Distribution site (you can `Skip registration`), pick `Distribution Installer` for `Mac` that corresponds to

your Mac CPU (as
a rule of thumb, starting 2020 Macs use Apple Silicon and not the Intel Chip), download the `Command Line Installer` and run it from a terminal (adapt the file name to the downloaded version):

```
bash Anaconda3-2025.06-0-MacOSX-arm64.sh
```

Follow the instructions, anaconda will be installed in */home/your_user/anaconda3/* by default. The installer will ask if you want to initialize/modify **.bashrc** or **.bash_profile**, select 'yes'. The i
nstaller should have added to your *.zshrc*, *.bashrc* or *.bash_profile* files (in your home directory) something like this:

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/Users/your_user/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/Users/your_user/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/Users/your_user/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/Users/your_user/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

This will activate anaconda automatically every time you open a terminal.

- Close all the terminals and open a new one in order to re-load this new setup.
- Check that python is correctly installed:

```
which python
```

```
python --version
```

which should show something similar to:

```
/Users/your_user/anaconda3/bin/python
```

```
Python 3.13.5
```

- Install a library that does not come by default and that it is needed by iSpec:

```
conda update conda
conda update --all
conda install lockfile cython
```

- Open a terminal, run `xcode-select --install`, and click the Install button to install the required **command line developer tools**. Don't worry if you see a message telling you the software cannot be
  installed because it is not currently available from the Software Update Server. This usually means you already have the latest version installed. Once installed, verify that this version of gcc is the on
  e being executed from your terminal:

```
which gcc
gcc --version
```

which should show something similar to:

```
/usr/bin/gcc
```

```
Apple clang version 17.0.0 (clang-1700.0.13.5)
Target: arm64-apple-darwin24.5.0
Thread model: posix
```

```
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
```

If your system already had these tools and it has gone through multiple updates, sometimes it breaks and it is required to fully remove the tools with `sudo rm -rf /Library/Developer/CommandLineTools/` a
nd trigger the install again with `xcode-select --install`.

- Install the official gfortran binaries, the latest verified version was gfortran 14.2. Onc
  e installed, verify that this version of gfortran is the one being executed from your terminal:

```
which gfortran
gfortran --version
```

which should show:

```
/usr/local/bin/gfortran
```

```
GNU Fortran (GCC) 14.2.0
Copyright (C) 2024 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Sometimes it is necessary to uninstall previously installed gfortran, for that you can run `sudo rm -rf /usr/local/gfortran/ /usr/local/bin/gfortran; sudo pkgutil --forget com.gnu.gfortran` and install it again.

- Download iSpec, and decompress it:

```
curl -O https://www.blancocuaresma.com/s/tmp/scratch/iSpec_v20250801.tar.gz
tar -zxvf iSpec_v20250801.tar.gz
mv iSpec_v20250801 iSpec
```

- Make sure all required Python packages are present in the system:

```
cd iSpec/
pip install -r requirements.txt
```

- Go into the iSpec directory and compile the radiative transfer codes (it will print many warnings, they can be ignored, however if there are errors, then you will not be able to synthesize spectra and determine parameters):

```
cd iSpec/
make clean
make spectrum
make turbospectrum
make moog
make moog-scat
```

- [OPTIONAL] Download an additional pre-computed grid:

```
cd iSpec/input/grid/
curl -O https://lweb.cfa.harvard.edu/~sblancoc/iSpec/grid/SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
tar -xvf SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
rm -f SPECTRUM_MARCS.GES_GESv6_atom_hfs_iso.480_680nm.tar
```

- [OPTIONAL] Download an additional model atmosphere:

```
cd iSpec/input/atmospheres/
curl -O https://lweb.cfa.harvard.edu/~sblancoc/iSpec/atmospheres/MARCS.NLTE.tar
tar -xvf MARCS.NLTE.tar
rm -f MARCS.NLTE.tar
```

- Run the visual interface (GUI) for the first time, which will trigger some additional compilation:

```
cd iSpec/
./iSpec.command
```

Optional

By default, anaconda will add to your `.bashrc`, `.bash_profile`, or `.zshrc` (in your home directory) something like this:

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/home/your_user/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/your_user/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/your_user/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/your_user/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

This will activate anaconda automatically every time you open a terminal. Personally, I prefer to have more control and I replace those lines with:

```
#------------------------------------------------------------------------------
path_remove() {
  # Delete path by parts so we can never accidentally remove sub paths
  if [ $PATH = $1 ]; then
      PATH=""
  else
      PATH=${PATH//":$1:"/":"} # delete any instances in the middle
      PATH=${PATH/#"$1:"/} # delete any instance at the beginning
      PATH=${PATH/%":$1"/} # delete any instance in the at the end
```

```
  fi
}

path_append() {
    # Insert path at the end if it exists and was not added before
    # (PATH should have been already exported before)
    if [ -d "$1" ] && [[ ":$PATH:" != *":$1:"* ]]; then
        PATH="${PATH:+"$PATH:"}$1"
    fi
}

path_insert() {
    # Insert path at the beginning if it exists and was not added before
    # (PATH should have been already exported before)
    if [ -d "$1" ] && [[ ":$PATH:" != *":$1:"* ]]; then
        PATH="$1${PATH:+":$PATH"}"
    fi
}

path_reappend() {
    # Remove path if it exists (it was added before)
    # Re-append path at the end
    path_remove $1
    path_append $1
}

path_reinsert() {
    # Remove path if it exists (it was added before)
    # Re-insert path at the beginning
    path_remove $1
    path_insert $1
}
#.....................................................................
```

```
# Anaconda
if [ -d "$HOME/anaconda3/bin" ]; then
    anaconda_activate() {
        path_insert "$HOME/anaconda3/bin"
        ANACONDA_NAME="anaconda3"
        if [[ "${PS1}" != *"[\${ANACONDA_NAME}]"* ]] ; then
            export _ANACONDA_OLD_PS1=$PS1
            export PS1="[\${ANACONDA_NAME}] ${PS1}"
        fi
    }
    anaconda_deactivate() {
        path_remove "$HOME/anaconda3/bin"
        if ! [ -z "${_ANACONDA_OLD_PS1+_}" ] ; then
            PS1="$_ANACONDA_OLD_PS1"
            export PS1
            unset _ANACONDA_OLD_PS1
        fi
    }
fi
```

This gives me more control since I can manually activate anaconda executing `anaconda_activate` and deactivate it with `anaconda_deactivate`. However, this option needs you to remember to run that activation manually every time you want to use anaconda.