[Telepítés Windowsra 2](#_Toc79529114)

[Telepítés Linuxra 6](#_Toc79529115)

[A Juypter Notebook indítása a kód szerkesztéséhez és az executable elkészítése 7](#_Toc79529116)

[Linux: 7](#_Toc79529117)

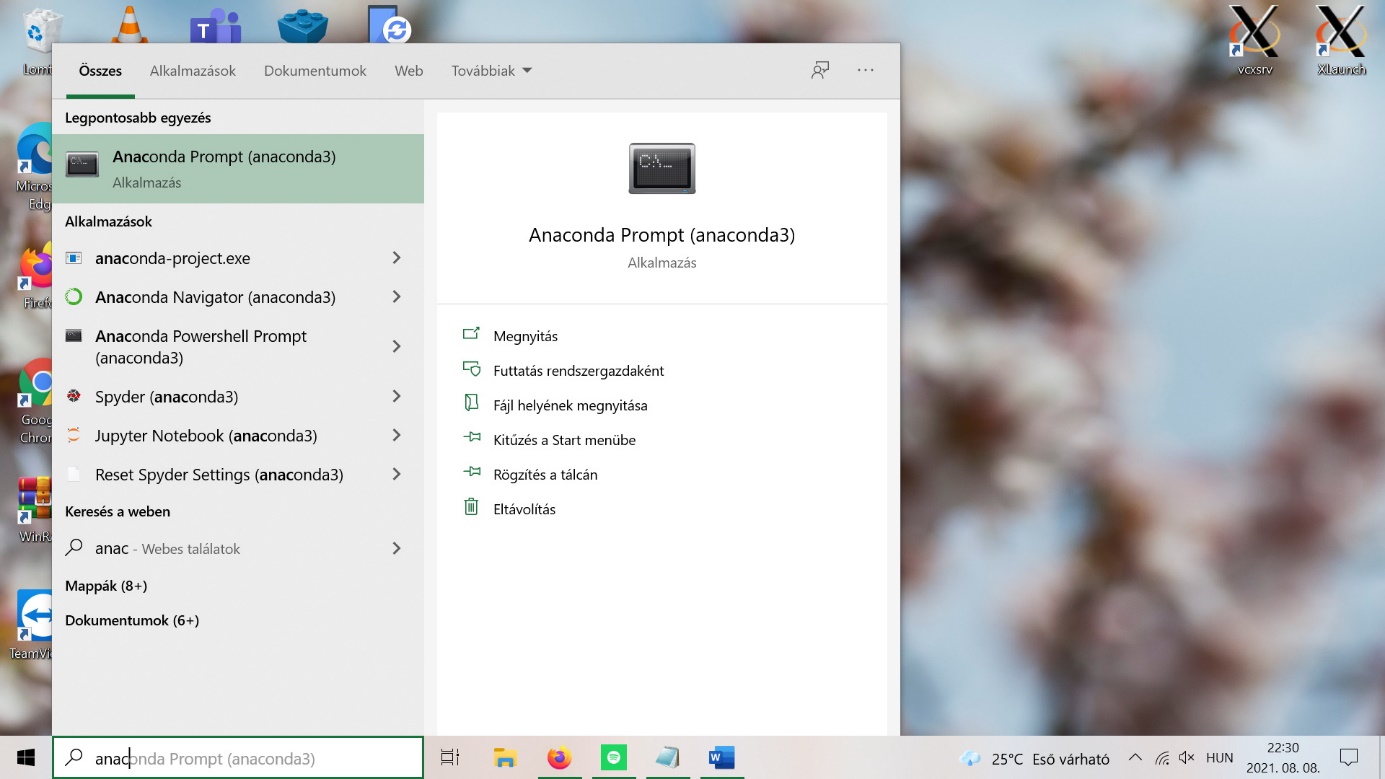
[Windows: 9](#_Toc79529118)

Telepítés Windowsra

1. töltsd le az exe-t innen: <https://drive.google.com/file/d/1vihMWkVK4HpgKhfaed75YVqpTTFych4Y/view?usp=sharing>
2. keresd ki a letöltésekből, kattints rá és elindul 😊

Ha ez valamiért nem működik, b) opció:

1. A CrystSimGUI.exe-t másold fel, lehetőleg a C:\Users\saját felhasználód\ mappába
2. Telepítsd az Anaconda programot ( <https://www.anaconda.com/products/individual#windows> ), semmi különleges opció nem kell, minden beállítás jó úgy, ahogy felajánlja
3. Nyisd meg az Anaconda prompt-ot



1. conda create --name myenv python=2.7.17 scipy matplotlib numpy

A képen szöveg látható

Automatikusan generált leírás

1.: ez kell megjelenjen utána

A képen szöveg látható

Automatikusan generált leírás

2. y, enter, várni amíg befejezi…

1. amikor befejezte és utána minden újraindításkor: conda activate myenv
2. a program indítása: .\CrystSimGUI.exe (ha máshova tetted: C:\ahol\a\program\van\CrystSimGUI.exe)

Indítás:

1. Anaconda Promptban conda activate myenv
2. .\CrystSimGUI.exe (vagy C:\ahol\a\program\van\CrystSimGUI.exe)
3. 1-2 perc és el is indul

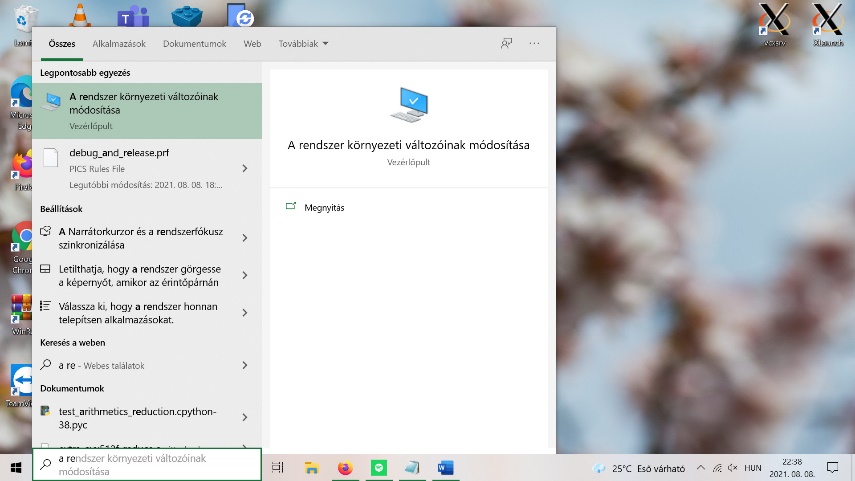
A képen szöveg látható

Automatikusan generált leírás

Ismert problémák a program indításánál:

* program nem indul el és ilyen hibát dob: LookupError: unknown encoding: cp65001

Megoldás: A rendszer környezeti változóinak módosítása🡪Környezeti változók🡪Rendszerváltozónál Új…

A képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírás

itt az új változó neve: PYTHONIOENCODING, értéke: UTF-8

ezután mindent le kell okézni és az anaconda prompt-ot újraindítani, utána működni fog

A képen szöveg látható

Automatikusan generált leírás

* nem dob semmilyen hibaüzenetet, de nem jelenik meg a grafikus felület (anacondában):

Megoldás: valószínűleg a PyQt könyvtár nincs telepítve: conda list pyqt ha nem ad ki semmilyen találatot, akkor conda install pyqt

ha van telepítve egy PyQt 5.x.x verzió és így sem indul el a grafikus felület, akkor meg lehet próbálkozni egy PyQt4 telepítésével (lásd [itt](#pyqt4); nem valószínű, hogy ez a hiba előjön a kész .exe fájl használatakor, de megeshet)

A képen szöveg látható

Automatikusan generált leírás

* nem jól jelennek meg a grafikus elemek (a grafikonok fele hiányzik, a betűméret túl nagy, véletlenszerűen átméreteződik az ablak)

A kezelőfelület egy 1920x1080-as monitorra van optimalizálva, ennél kisebb pl laptop monitoron nem biztos, hogy helyesen jelenik meg

* a szimuláció elindítása után azt írja ki, hogy a program nem válaszol

Igen. Miközben a szimuláció fut, a grafikus felület tényleg „lefagy”, nem lehet benne semmit sem csinálni, ezt a Windows úgy érzékeli, hogy a program nem válaszol. Ettől még a számolás fut a háttérben (a konzolon lehet ellenőrizni, hogy megjelennek az iterációs körökhöz tartozó cost értékek), ha befejezte, akkor a grafikus felület is visszaáll.

Telepítés Linuxra

elméletileg a Linux executable (itt: <https://drive.google.com/file/d/1l0d88YYUr8RH6KAl3xZrAbx4wBat7mzD/view?usp=sharing> ) használható anélkül is, hogy az Anaconda telepítve lenne (ellentétben a Windows verzióval)

A képen szöveg látható

Automatikusan generált leírás

3 a futtatható program dependenciái (WSL-Ubuntu 20.04)

ettől függetlenül, ha cd ahol/a/fájl/van és ./CrystSimGUI után nem indul el, akkor érdemes felállítani az anaconda environmentet és abban próbálkozni (lásd [itt](#conda_newenv))

A Juypter Notebook indítása a kód szerkesztéséhez és az executable elkészítése

Linux:

1. telepítsd a gfortran-t (sudo apt install gfortran)
2. telepítsd az Anacondát

curl -O https://repo.anaconda.com/archive/Anaconda3-2019.07-Linux-x86\_64.sh majd

bash Anaconda3-2019.07-Linux-x86\_64.sh

"Do you wish the installer to initialize Anaconda3 by running conda init?" yes

1. készíts egy új környezetet a megfelelő csomagokkal

conda create --name myenv scipy matplotlib numpy jupyter majd conda activate myenv

1. cd ahol/a/fcrystal/van, majd f2py -c -m fcrystal fcrystal.f90. Ez lefordítja az f90 kódot és egy Pythonba importálható csomagot csinál. Meg kell jelenjen egy fcrystal.cpython-37m-x86\_64-linux-gnu.so (vagy hasonló) fájl (ls -lh-val ellenőrizhető, hogy mikori a legutóbbi .so fájl) Ezt a Fortran kód minden módosítása után meg kell csinálni
2. ugyanitt jupyter notebook megnyitja a szerkesztőt egy új böngészőablakban és onnan futtathatóak az .ipynb fájlok

A képen szöveg látható

Automatikusan generált leírás

4 a Jupyter Notebook indítása.

Az executeable elkészítése:

1. először az .ipynb notebookot kell lefordítani .py Python scriptre. Ehhez az nbconvert szükséges (pip install nbconvert, ha még nincs telepítve): jupyter nbconvert --to script CrystSimGUI.ipynb
2. ezután, ha a környezetben minden csomag (matplotlib, stb…), amit használ a program, telepítve van, és a mappában, ahol állsz, ott van minden, amit a program beolvas (fcrystal, pcrystal, képfájlok, stb…) a Pyinstaller-el lehet elkészíteni a futtatható programot (pip install pyinstaller)

Ehhez valószínűleg szükség van arra, hogy a PyQt egy régebbi verzióját telepítsd pip install --user PyQt5==5.15.3 -al (ez a Pyinstaller verziójától is függ, de az 5.15.3-as PyQt biztosan működik). pyinstaller --onefile CrystSimGUI.py elkészíti a futtatható programot

!!! fontos: ha a Jupyter Notebookból lefutott a script, akkor ez is működni fog, kivéve: a %matplotlib inline-t mindenhonnan ki kell szedni (és bármi egyebet, ami nem UTF-8as karaktereket írna ki a konzolra, pl grafikonokat) és az egész legelejére szükséges egy from IPython import get\_ipython

1. a kész executable-t egy dist mappába fogja belerakni abban a mappában, ahonnan elindítottad, úgyhogy ./dist/CrystSimGUI -val elindítható a program

A képen szöveg látható

Automatikusan generált leírás

Ha olyan executable-t szeretnél csinálni, ami terjeszthető:

* Minden médiafájlra való hivatkozást (a háttérnek használt képek…) ki kell szedni
* Ha ragaszkodsz a képfájlokhoz:
  + valahova a kód elejére ezt másold be (innentől nem tudja a jupyter notebook lefuttatni, mert a \_\_file\_\_:-t nem kezeli rendesen):

import os

def resource\_path(relative\_path):

base\_path = getattr(sys, '\_MEIPASS', os.path.dirname(os.path.abspath(\_\_file\_\_)))

return os.path.join(base\_path, relative\_path)

* + a képfájlokat behívó kódot erre cseréld le:

image\_path = resource\_path("kep\_neve.png")

background\_image=tk.PhotoImage(file = image\_path)

* + futtasd le a jupyter nbconvert --to script CrystSimGUI.ipynb és a pyinstaller --onefile CrystSimGUI.py -t
  + nyisd meg a CrystSimGUI.spec fájlt, és add hozzá minden képfájlhoz ezt: a.datas += [('kep\_neve.png', '/home/ahol/a/kep/van/kep\_neve.png', 'DATA')]

A képen szöveg látható

Automatikusan generált leírás

5 ide illeszd a képfájlok elérési útját (ez éppen windowson, de értelemszerűen...)

* + mentsd el, majd pyinstaller --onefile CrystSimGUI.spec -el készítsd el a futtatható programot

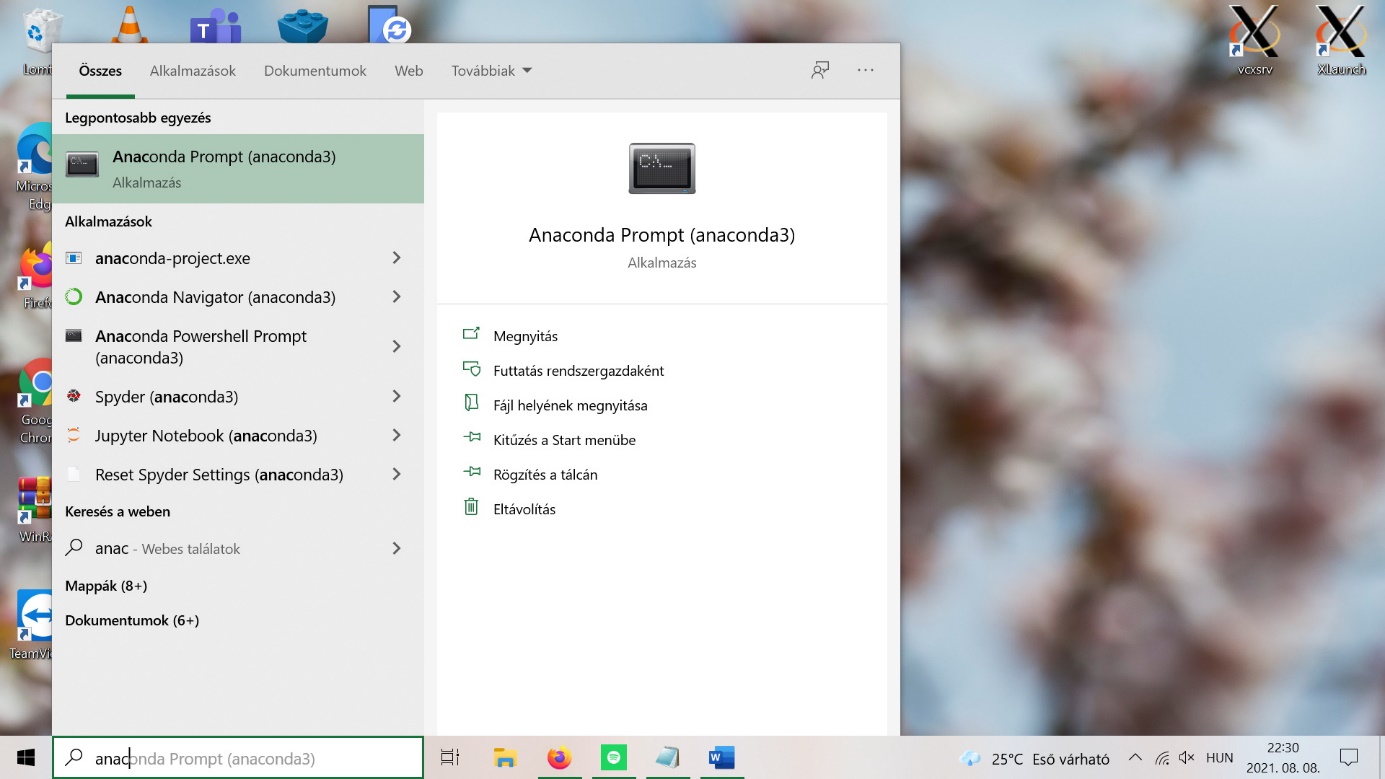
csak .png vagy .gif kiterjesztéssel működik !!!!

(forrás: <https://stackoverflow.com/questions/31836104/pyinstaller-and-onefile-how-to-include-an-image-in-the-exe-file> )

Windows:

Azoknak, akik szeretnék módosítani a programot, de nem feltétlenül foglalkoznának a futtatható fájllal:

1. opció:
2. telepítsd a WSL2-t, leírás itt: <https://altis.com.au/installing-ubuntu-bash-for-windows-10-wsl2-setup/>
3. megnyitva az Ubuntut: sudo apt install firefox
4. töltsd le a VcXsrv-t innen: <https://sourceforge.net/projects/vcxsrv/> és telepítsd fel, a disable access control opciót kell bejelölni, minden más opció default
5. a Windows Defenderben (Tűzfal és hálózatvédelem🡪Alkalmazás átengedése a tűzfalon) add hozzá kivételnek a VcXsrv-t magán és nyilvános hálózaton is
6. export DISPLAY=$(route.exe print | grep 0.0.0.0 | head -1 | awk '{print $4}'):0.0 (ha ezzel működik, akkor ezt érdemes hozzáadni a ~/.profile fájlhoz) (<https://stackoverflow.com/questions/61860208/wsl-2-run-graphical-linux-desktop-applications-from-windows-10-bash-shell-error> )
7. innentől a Linuxos futtatási leírást kell követni
8. opció:
9. Telepítsd az Anaconda programot ( <https://www.anaconda.com/products/individual#windows> ), semmi különleges opció nem kell, minden beállítás jó úgy, ahogy felajánlja
10. Nyisd meg az Anaconda prompt-ot



1. conda create --name myenv python=3.9 scipy matplotlib numpy jupyter

A képen szöveg látható

Automatikusan generált leírás

1.: ez kell megjelenjen utána

A képen szöveg látható

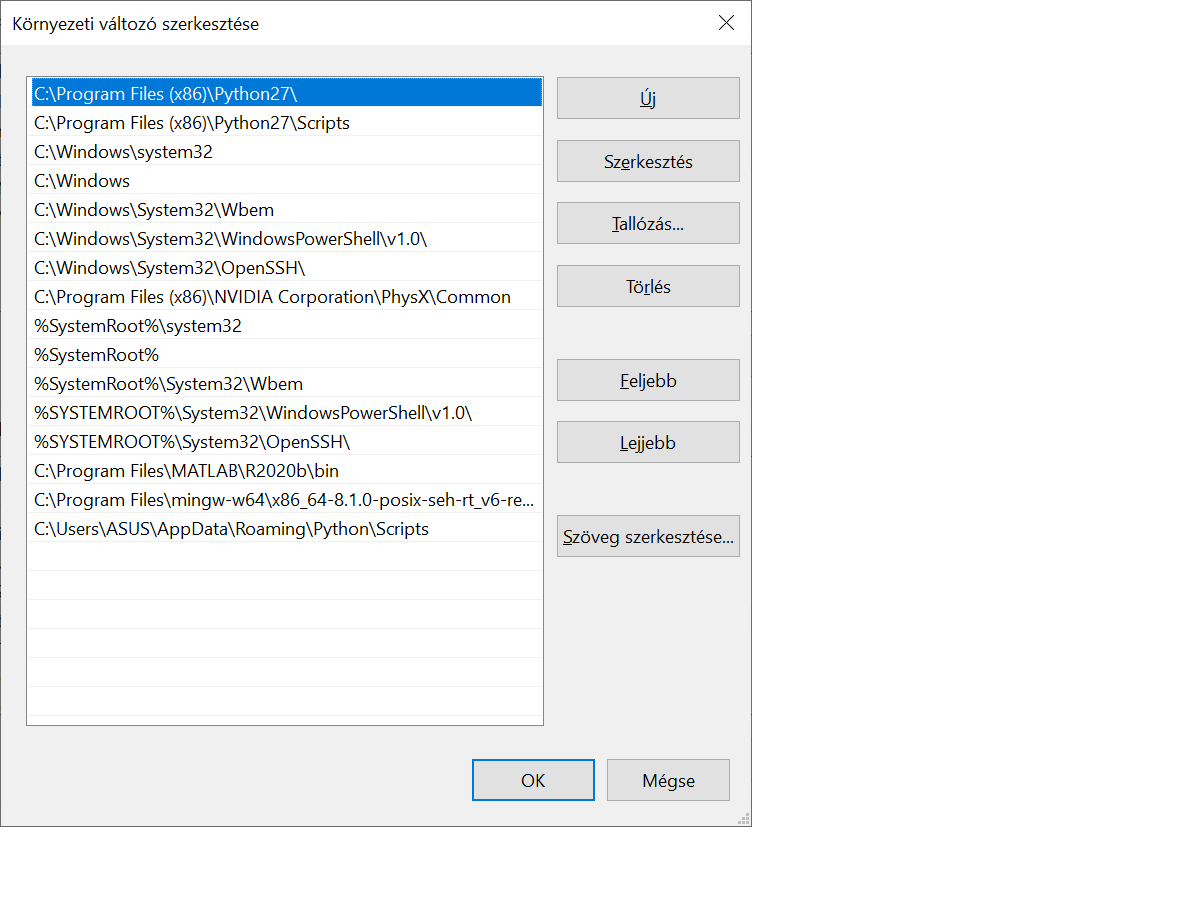
Automatikusan generált leírás

2. y, enter, várni amíg befejezi…

1. amikor befejezte és utána minden újraindításkor: conda activate myenv
2. jupyter notebook

Ha a jupyter notebook itt meg tud nyílni, akkor lehet szerkeszteni a kódot. Ahhoz, hogy a .ipynd notebook futtatható legyen, a fortran kódot le kell fordítani:

1. Telepítsd a MinGW-t innen: <http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/installer/mingw-w64-install.exe/download> (architecture: x86\_64, minden más opció default)
2. A képen szöveg látható

   Automatikusan generált leírásA környezeti változók között: A PATH-hoz add hozzá: C:\Program Files\mingw-w64\x86\_64-8.1.0-posix-seh-rt\_v6-rev0\mingw64\bin (…vagy ahol nálad van a mingw64 bin mappája) és hozz létre egy C\_INCLUDE\_PATH környezeti változót ami C:\Program Files\mingw-w64\x86\_64-8.1.0-posix-seh-rt\_v6-rev0\mingw64\include

6 Valahogy hasonlóan kell kinézzenek a környezeti változók

1. az fcrystal mappájában állva python C:\Users\ASUS\anaconda3\Scripts\f2py.exe -c --fcompiler=gnu95 --compiler=mingw32 -m fcrystal fcrystal.f90 (...vagy ahol a te Anacondádban az f2py van (lehet, hogy más a neve. f2py, f2py2e pl. lehet. valamikor az volt ebben a dokumentációban, hogy úgy is működik, ha f2py-script.py van az .exe helyett. ha nem megy le a sima f2py ki lehet próbálni azt is) Igen, mingw32, akkor is, ha a 64bites verziót telepítetted, az a biztos. Ha ez nem akar lefutni, hanem „error using mingw compiler in nt environment” jellegű hibával kilép, ami ezt meg kellene oldja: python setup.py build --compiler=gcc, majd az anaconda újraindítása (előfordulhat ilyenkor: hiányzó ftpmirror.py fájl. megoldás: Just add/copy missing file from https://svn.python.org/projects/python/trunk/Tools/scripts/ftpmirror.py to the path C:\path\to\python\Tools\Scripts)
2. most már a jupyter notebook paranccsal elindítható a Jupyter notebook ahonnan lehet szerkeszteni és futtatni az .ipynb fájlt

Ha ebből futtatható fájlt szeretnél:

1. opció:
2. hozz létre egy új felhasználót a gépen, amin nincs Anaconda
3. telepítsd a Pythont (nálam python 3.10-zel működött, Microsoft Storeban van, talán onnan a legkönnyebb)
4. parancssorban ellenőrizd hogy működik-e: python -v kiadja a verziót
5. telepítsd a csomagokat, amik kellenek: pip install numpy scipy matplotlib wheel pyqt5 pyinstaller pycairo cairocffi
6. telepítsd a MinGWt és állítsd be a környezeti változóit, aztán fordítsd le a fortran kódot az f2py-al
7. jupyter nbconvert --to script CrystSimGUI.ipynb -tel .py fájlt készíteni
8. pyinstaller –onefile CrystSimGUI.py, ez alkot egy exe-t de az nem fog működni, de innentől a spec fájl szerkesztése egyezik a Linuxos módszerrel, a dist mappában lesz a végső fájl
9. opció:

ugyanez, csak az Anacondán belül, új felhasználó létrehozása nélkül (a tapasztalat az, hogy nem biztos, hogy működik)

* Ha nem indul el a jupyter notebook (ezek a WSL-re jellemző hibák, de mindenhol előfordulhat)
  + Ha már fut egy firefox, zárd be
  + ha „Unable to init server: Broadway display type not supported” hibát ad: ellenőrizd, hogy a grafikus elemeket kezelő program kivételként hozzá van-e adva a windows defender tűzfalában
* Ha nem tudja az .ipynb kódból elindítani a grafikus felületet:
  + Valamelyik import hibás (nincs meg a fájl/telepítve a csomag)
  + A kód kommentjeiben olyan karakter van, amit nem tud feldolgozni (ékezetes betű… )
  + PyQt5 néhány verziója nem működik: 
    - amikor ezt python 2.7-tel próbáltam, akkor ez volt a megoldás:

Ha van pip, setuptools és wheel: <https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4> -ről letölteni egy PyQt4-hez tartozó wheel fájlt (nálam 64 bites Anacondához a PyQt4‑4.11.4‑cp27‑cp27m‑win\_amd64.whl volt a jó), majd pip install c:\Users\ASUS\Downloads\PyQt4‑4.11.4‑cp27‑cp27m‑win\_amd64.whl (valószínűleg a saját downloads folderedben lesz). Valószínűleg ez pyqt5 downgradelésére is működik.

* \_tkinter.TclError: couldn't open… és \_tkinter.TclError: couldn't recognize data in image file hibák: lásd [itt](#exe_kep). annyi módosítással, hogy csak .gif képfájlokat tud kezelni + Unicode karaktereket nem jól jeleníti meg (pl. nem ismeri meg, hogy az \u2103 a °C)

Win10, python 3.10.1 mellett a működő csomagok:

Package Version

------------------------- --------

altgraph 0.17.2

cairocffi 1.3.0

cffi 1.15.0

cycler 0.11.0

fonttools 4.28.3

future 0.18.2

kiwisolver 1.3.2

matplotlib 3.5.0

numpy 1.21.4

packaging 21.3

pefile 2021.9.3

Pillow 8.4.0

pip 21.3.1

pycairo 1.20.1

pycparser 2.21

pyinstaller 4.7

pyinstaller-hooks-contrib 2021.4

pyparsing 3.0.6

PyQt5 5.15.6

PyQt5-Qt5 5.15.2

PyQt5-sip 12.9.0

python-dateutil 2.8.2

pywin32-ctypes 0.2.0

scipy 1.7.3

setuptools 58.1.0

setuptools-scm 6.3.2

six 1.16.0

tomli 1.2.2

wheel 0.37.0

Ubuntu 20.04, conda 4.7.10, python 3.9.7 mellett a működő csomagok:

# Name Version Build Channel

\_libgcc\_mutex 0.1 main

argon2-cffi 20.1.0 py39h27cfd23\_1

async\_generator 1.10 pyhd3eb1b0\_0

attrs 21.2.0 pyhd3eb1b0\_0

backcall 0.2.0 pyhd3eb1b0\_0

blas 1.0 mkl

bleach 4.0.0 pyhd3eb1b0\_0

brotli 1.0.9 he6710b0\_2

ca-certificates 2021.10.26 h06a4308\_2

certifi 2021.10.8 py39h06a4308\_0

cffi 1.14.6 py39h400218f\_0

cycler 0.11.0 pyhd3eb1b0\_0

dbus 1.13.18 hb2f20db\_0

debugpy 1.5.1 py39h295c915\_0

decorator 5.1.0 pyhd3eb1b0\_0

defusedxml 0.7.1 pyhd3eb1b0\_0

entrypoints 0.3 py39h06a4308\_0

expat 2.4.1 h2531618\_2

fontconfig 2.13.1 h6c09931\_0

fonttools 4.25.0 pyhd3eb1b0\_0

freetype 2.11.0 h70c0345\_0

giflib 5.2.1 h7b6447c\_0

glib 2.69.1 h5202010\_0

gst-plugins-base 1.14.0 h8213a91\_2

gstreamer 1.14.0 h28cd5cc\_2

icu 58.2 he6710b0\_3

importlib-metadata 4.8.2 py39h06a4308\_0

importlib\_metadata 4.8.2 hd3eb1b0\_0

intel-openmp 2021.4.0 h06a4308\_3561

ipykernel 6.4.1 py39h06a4308\_1

ipython 7.29.0 py39hb070fc8\_0

ipython\_genutils 0.2.0 pyhd3eb1b0\_1

ipywidgets 7.6.5 pyhd3eb1b0\_1

jedi 0.18.0 py39h06a4308\_1

jinja2 3.0.2 pyhd3eb1b0\_0

jpeg 9d h7f8727e\_0

jsonschema 3.2.0 pyhd3eb1b0\_2

jupyter 1.0.0 py39h06a4308\_7

jupyter\_client 7.0.6 pyhd3eb1b0\_0

jupyter\_console 6.4.0 pyhd3eb1b0\_0

jupyter\_core 4.9.1 py39h06a4308\_0

jupyterlab\_pygments 0.1.2 py\_0

jupyterlab\_widgets 1.0.0 pyhd3eb1b0\_1

kiwisolver 1.3.1 py39h2531618\_0

lcms2 2.12 h3be6417\_0

ld\_impl\_linux-64 2.35.1 h7274673\_9

libffi 3.3 he6710b0\_2

libgcc-ng 9.1.0 hdf63c60\_0

libgfortran-ng 7.3.0 hdf63c60\_0

libpng 1.6.37 hbc83047\_0

libsodium 1.0.18 h7b6447c\_0

libstdcxx-ng 9.1.0 hdf63c60\_0

libtiff 4.2.0 h85742a9\_0

libuuid 1.0.3 h7f8727e\_2

libwebp 1.2.0 h89dd481\_0

libwebp-base 1.2.0 h27cfd23\_0

libxcb 1.14 h7b6447c\_0

libxml2 2.9.10 hb55368b\_3

lz4-c 1.9.3 h295c915\_1

markupsafe 2.0.1 py39h27cfd23\_0

matplotlib 3.5.0 py39h06a4308\_0

matplotlib-base 3.5.0 py39h3ed280b\_0

matplotlib-inline 0.1.2 pyhd3eb1b0\_2

mistune 0.8.4 py39h27cfd23\_1000

mkl 2021.4.0 h06a4308\_640

mkl-service 2.4.0 py39h7f8727e\_0

mkl\_fft 1.3.1 py39hd3c417c\_0

mkl\_random 1.2.2 py39h51133e4\_0

munkres 1.1.4 py\_0

nbclient 0.5.3 pyhd3eb1b0\_0

nbconvert 6.1.0 py39h06a4308\_0

nbformat 5.1.3 pyhd3eb1b0\_0

ncurses 6.3 h7f8727e\_2

nest-asyncio 1.5.1 pyhd3eb1b0\_0

notebook 6.4.6 py39h06a4308\_0

numpy 1.21.2 py39h20f2e39\_0

numpy-base 1.21.2 py39h79a1101\_0

olefile 0.46 pyhd3eb1b0\_0

openssl 1.1.1l h7f8727e\_0

packaging 21.3 pyhd3eb1b0\_0

pandocfilters 1.4.3 py39h06a4308\_1

parso 0.8.2 pyhd3eb1b0\_0

pcre 8.45 h295c915\_0

pexpect 4.8.0 pyhd3eb1b0\_3

pickleshare 0.7.5 pyhd3eb1b0\_1003

pillow 8.4.0 py39h5aabda8\_0

pip 21.2.4 py39h06a4308\_0

prometheus\_client 0.12.0 pyhd3eb1b0\_0

prompt-toolkit 3.0.20 pyhd3eb1b0\_0

prompt\_toolkit 3.0.20 hd3eb1b0\_0

ptyprocess 0.7.0 pyhd3eb1b0\_2

pycparser 2.21 pyhd3eb1b0\_0

pygments 2.10.0 pyhd3eb1b0\_0

pyparsing 3.0.4 pyhd3eb1b0\_0

pyqt 5.9.2 py39h2531618\_6

pyrsistent 0.18.0 py39heee7806\_0

python 3.9.7 h12debd9\_1

python-dateutil 2.8.2 pyhd3eb1b0\_0

pyzmq 22.3.0 py39h295c915\_2

qt 5.9.7 h5867ecd\_1

qtconsole 5.1.1 pyhd3eb1b0\_0

qtpy 1.10.0 pyhd3eb1b0\_0

readline 8.1 h27cfd23\_0

scipy 1.6.2 py39had2a1c9\_1

send2trash 1.8.0 pyhd3eb1b0\_1

setuptools 58.0.4 py39h06a4308\_0

sip 4.19.13 py39h2531618\_0

six 1.16.0 pyhd3eb1b0\_0

sqlite 3.36.0 hc218d9a\_0

terminado 0.9.4 py39h06a4308\_0

testpath 0.5.0 pyhd3eb1b0\_0

tk 8.6.11 h1ccaba5\_0

tornado 6.1 py39h27cfd23\_0

traitlets 5.1.1 pyhd3eb1b0\_0

tzdata 2021e hda174b7\_0

wcwidth 0.2.5 pyhd3eb1b0\_0

webencodings 0.5.1 py39h06a4308\_1

wheel 0.37.0 pyhd3eb1b0\_1

widgetsnbextension 3.5.1 py39h06a4308\_0

xz 5.2.5 h7b6447c\_0

zeromq 4.3.4 h2531618\_0

zipp 3.6.0 pyhd3eb1b0\_0

zlib 1.2.11 h7f8727e\_4

zstd 1.4.9 haebb681\_0