

HAZEL2: "synthesis" mode

Shuo Wang

Dept. of Astronomy, NMSU

DKIST Ambassador

[NSO 5th NCSP DKIST Data-Training Workshop: He I Diagnostics in the Solar Atmosphere](#)

What is HAZEL?

- HAnle and ZEeman Light (HAZEL)
- A code tool for the synthesis and inversion of Stokes profiles caused by the joint action of atomic level polarization and the Hanle and Zeeman effects in some spectral lines, such as 10830 Å.
- Written by [Andrés Asensio Ramos](#)

What can HAZEL do?







- I have an atmospheric model, and I want to see how it influences my observables (spectral lines).

➡ **synthesis**

- I have observations, and I want to know the physical properties such as vector magnetic field and velocities of the atmosphere from where they originate.

➡ **inversion**


What can HAZEL do?

Atmospheric Model 	HAZEL	Observables 
Vector magnetic field LOS velocity Thermal velocity Optical depth Damping	 synthesis   inversion 	Line profiles of Stokes I, Q, U, V

Hazel Versions

- HAZEL2
<https://github.com/aasensio/hazel2>
for Python user (recommended)
- HAZEL
for IDL user

What lines can HAZEL2 handle?

- Chromospheric lines
 - **He I 10830 Å line** (Let's focus on this line.)
 - He I D3 5876 Å line
 - Ca II 8542 Å line
- [Photospheric lines](#) 
- Telluric lines

Basic Functions of HAZEL2

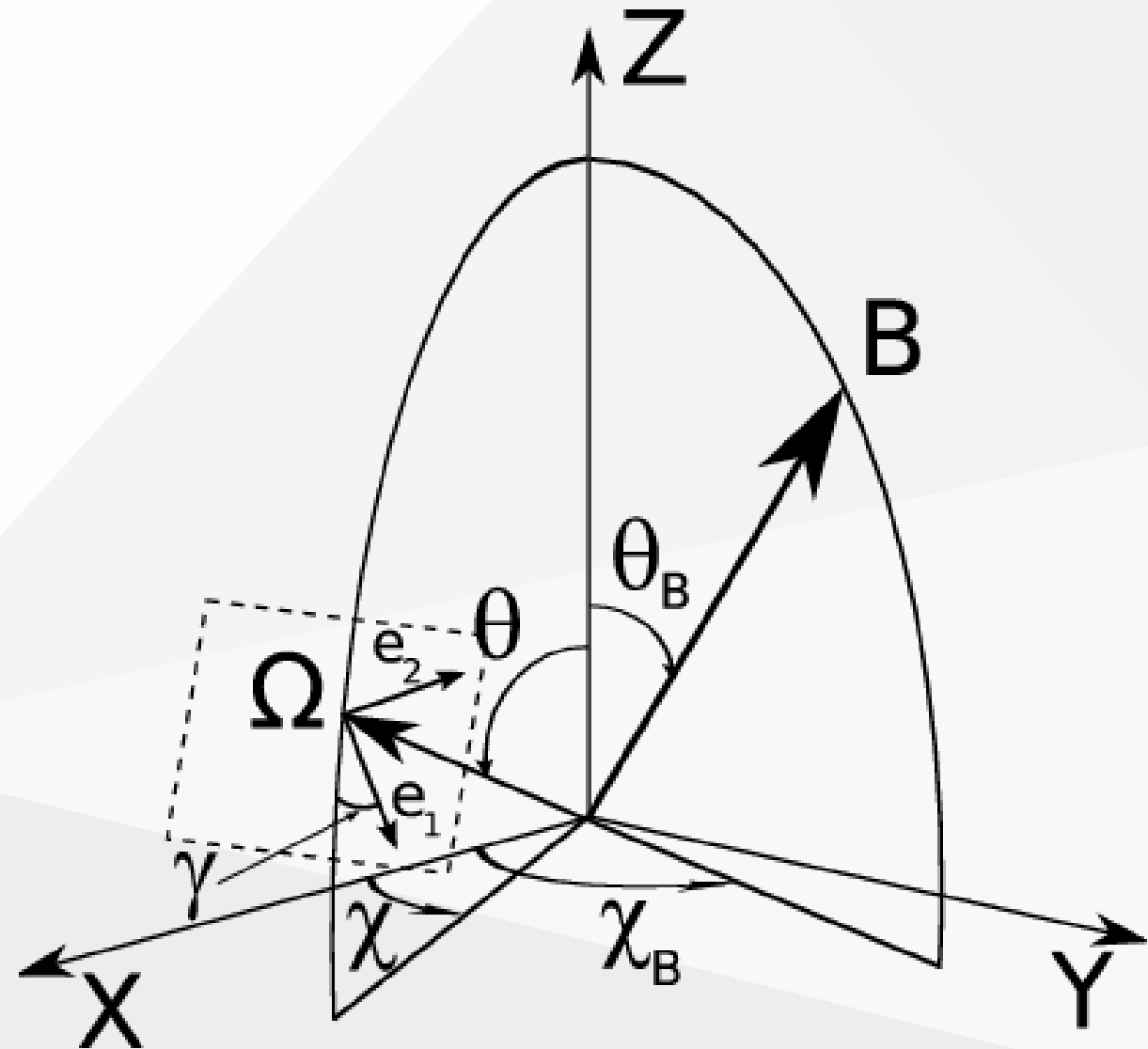
- **Synthesis** (Core function)
- **Inversion** (Call synthesis repeatedly, and return the values with best match.)

 Inversion will be discussed the next day (Wed) in more detail.

Geometry

- Observer angles (θ, χ, γ)

Symbol	Definition
Z axis	\perp solar atmosphere
Ω	LOS
X axis	in Z- Ω plane, $\chi = 0$
e1	Stokes Q+ direction



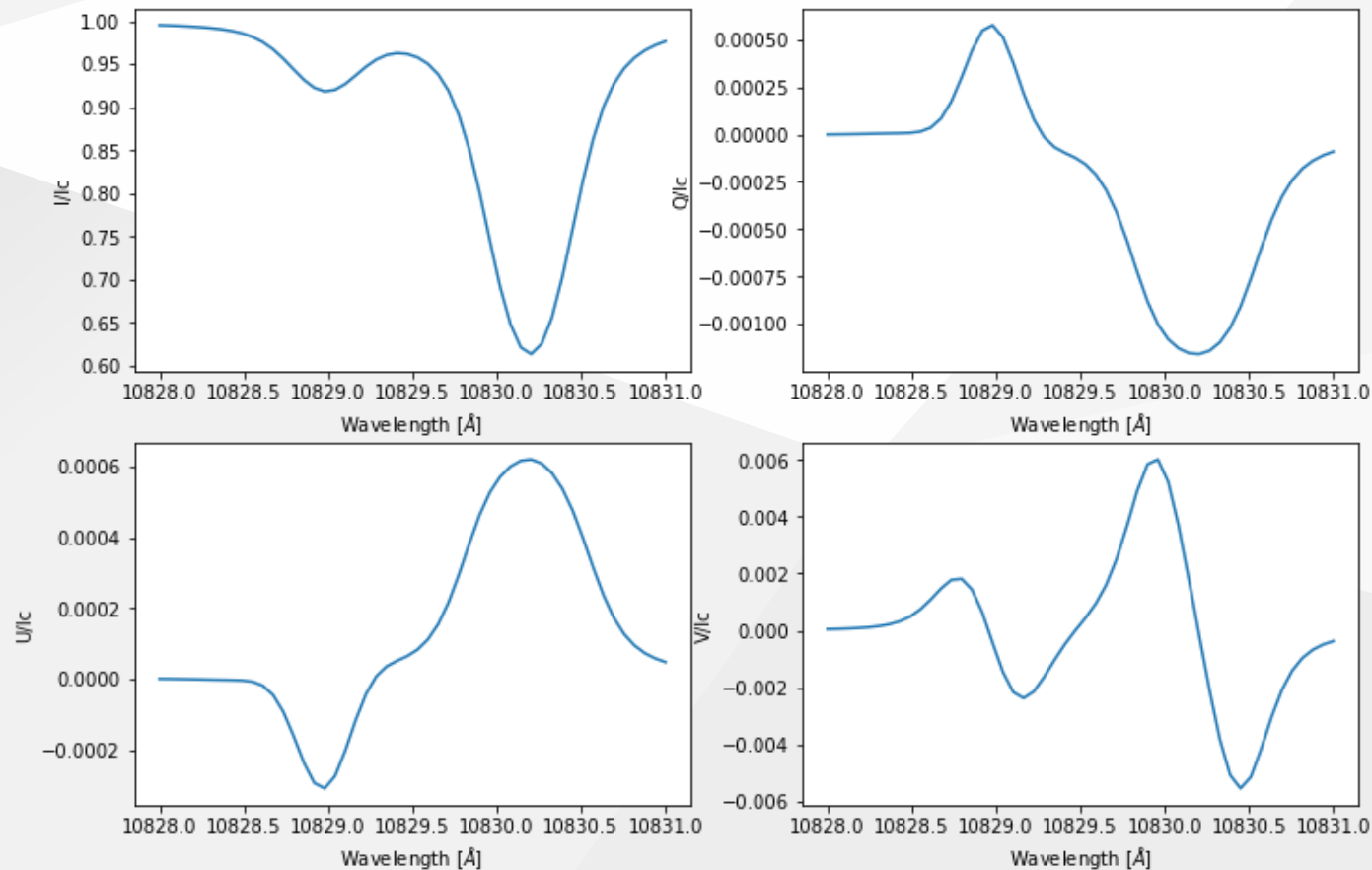
Input Synthesis

- Magnetic field vector (B_x, B_y, B_z)
- Thermal velocity affecting the width of the line
- Bulk velocity of the plasma leading to a redshift / blueshift
- Optical depth of the line
- Line damping parameter
- **Observer angles**

! The **Observer angles** depend on choice of coordinate system. They are defined in HAZEL as shown in the previous slide.

Synthesis Output

- Stokes I, Q, U, V profiles



How to install HAZEL2?

Follow instructions in the following links:

- [Hazel2 documents > 1. Installation](#)
- [NSO hazel 2 tutorial](#)

How to use HAZEL2?

- GUI (Synthesis only)
- Python code (Both synthesis and inversion)

Synthesis code (Python 3)


0 Import packages

```
import hazel  
import matplotlib.pyplot as plt
```

1 Construct model.

```
mod = hazel.Model(working_mode='synthesis')
```

- Working_mode: 'synthesis' or 'inversion'

 We will do this exercise just after this Intro. So that there will be ample time to test all these commands.

2 Add spectral region .

```
mod.add_spectral({'Name': 'spec1', 'Wavelength': [10828, 10831, 50],  
                 'topology': 'ch1', 'LOS': [0, 0, 90], 'Boundary condition': [1, 0, 0, 0]})
```

- Name: Define the name of the spectral region. Used in **3**.
- Wavelength: For sampling of line profiles [lower (Å), upper (Å), number of points]
- Topology: Combination of atmospheric layers. Use the value(s) of 'Name' in **3**.
- LOS: Observer angle in degrees [θ , ϕ , γ]. θ ranges from 0 (disk centrer) to 90 (limb). ϕ is χ in the figure of geometry.

2 Add spectral region .

```
mod.add_spectral({'Name': 'spec1', 'Wavelength': [10828, 10831, 50],  
                 'topology': 'ch1', 'LOS': [0, 0, 90], 'Boundary condition': [1, 0, 0, 0]})
```

- Boundary condition: The Stokes profiles that are entering the slab from below.





Defines the boundary condition normalized to the continuum intensity on the quiet Sun at disk center.

[Stokes I/Ic(mu=1), Q/Ic(mu=1), U/Ic(mu=1), V/Ic(mu=1)]

 [Hazel2 Documents > 5.2. How to deal with boundary conditions](#)

Topology Example

```
'topology': 'ph1 + ph2 -> ch1 -> te1 -> st1'
```

Light Direction	Layers
 st1	st: straylight contamination
 te1	te: telluric contamination
 ch1	ch: chromosphere
 ph1+ph2	ph: photosphere

->: Right arrow shows light path direction.

+: Atmospheres are added with a filling factor.

 [Hazel2 Documents > 3.3 Topology](#)

 We will do mostly one chromosphere in this workshop.

3 Add chromosphere 🌞.

```
mod.add_chromosphere({'Name': 'ch1', 'Spectral region': 'spec1', 'Height': 3, 'Line': '10830'})
```

- Name: Define the name of the chromosphere. Used in **5**.
- Spectral region: Use the value of 'Name' in **2**.
- Height: Try 3 arcsec for AR filament; 10-30 for quiescent filament.
- Line: Use '10830' for He I 10830 Å.

4 Set up model.

```
mod.setup()
```

5 Set atmospheric parameters.

```
mod.atmospheres['ch1'].set_parameters([-200, 50, 100, 1, -3, 6, 1, 0.5], 1)
```

[Bx, By, Bz, τ (optical depth) , v_{LOS} , v_{thermal} , β , a (damping)], ff

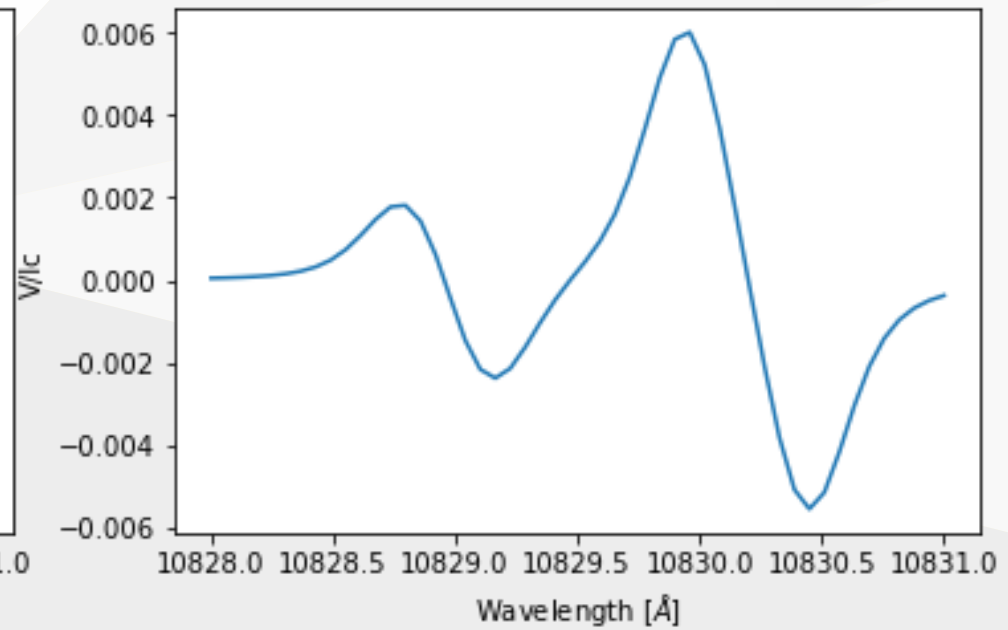
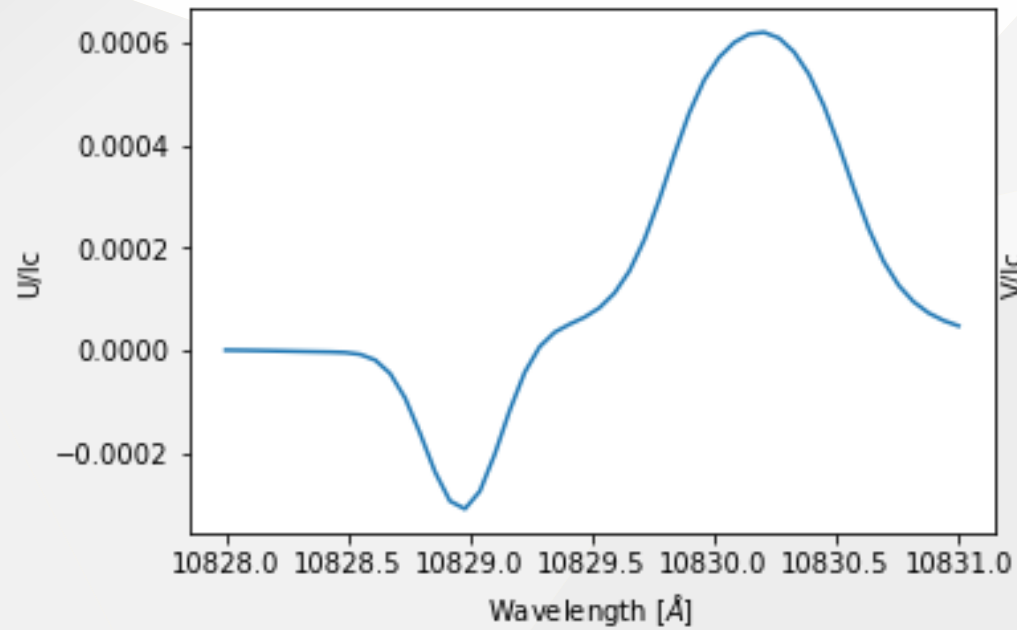
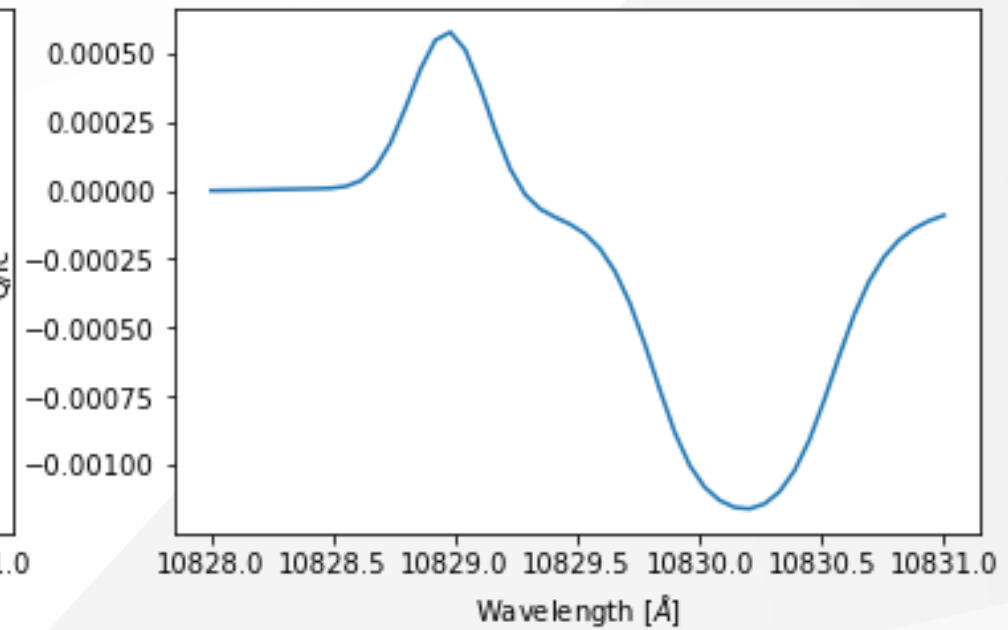
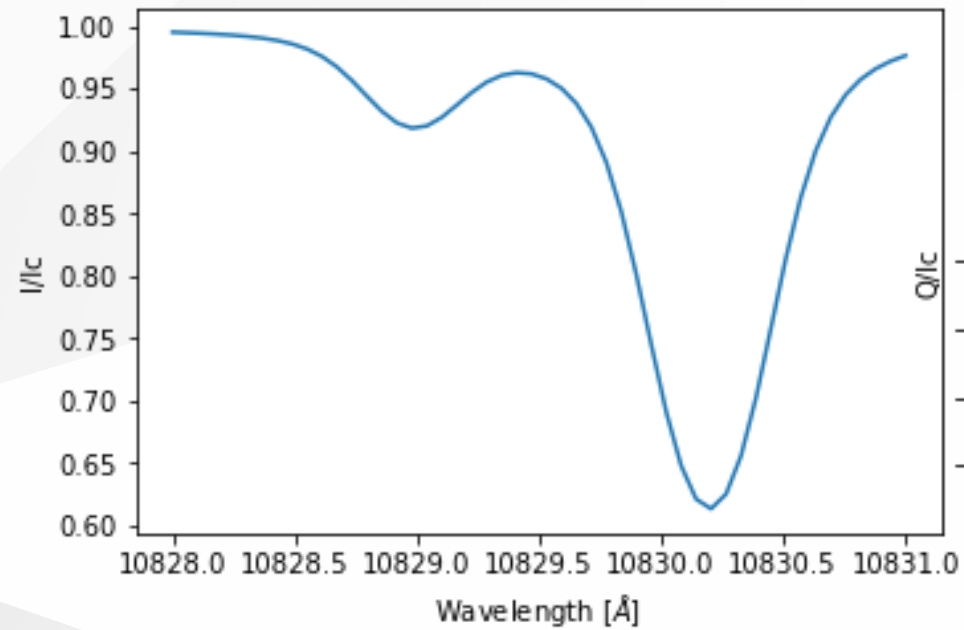
- Components of magnetic field are in Gauss
- Velocities are in km s^{-1}
- β : Enhancement factor for source function in radiative transfer equation. Always use 1, unless in strange situations (i.e., not only in absorption / on disk).
- ff: Filing factor. Ratio of light sensitive area versus total area of a pixel. We will mostly use 1.

6 Run.

```
mod.synthesize()
```

7 Plot results.

```
iq = 'IQUV'  
ms = mod.spectrum['spec1']  
plt.figure()  
for i in range(4):  
    plt.subplot(221+i)  
    plt.plot(ms.wavelength_axis, ms.stokes[i])  
    plt.xlabel('Wavelength [ $\text{\AA}$ '])  
    plt.ylabel(iq[i]+' / Ic')
```



Synthesis GUI

In terminal, type:

```
hazelgui
```



We will not use the GUI in the workshop.

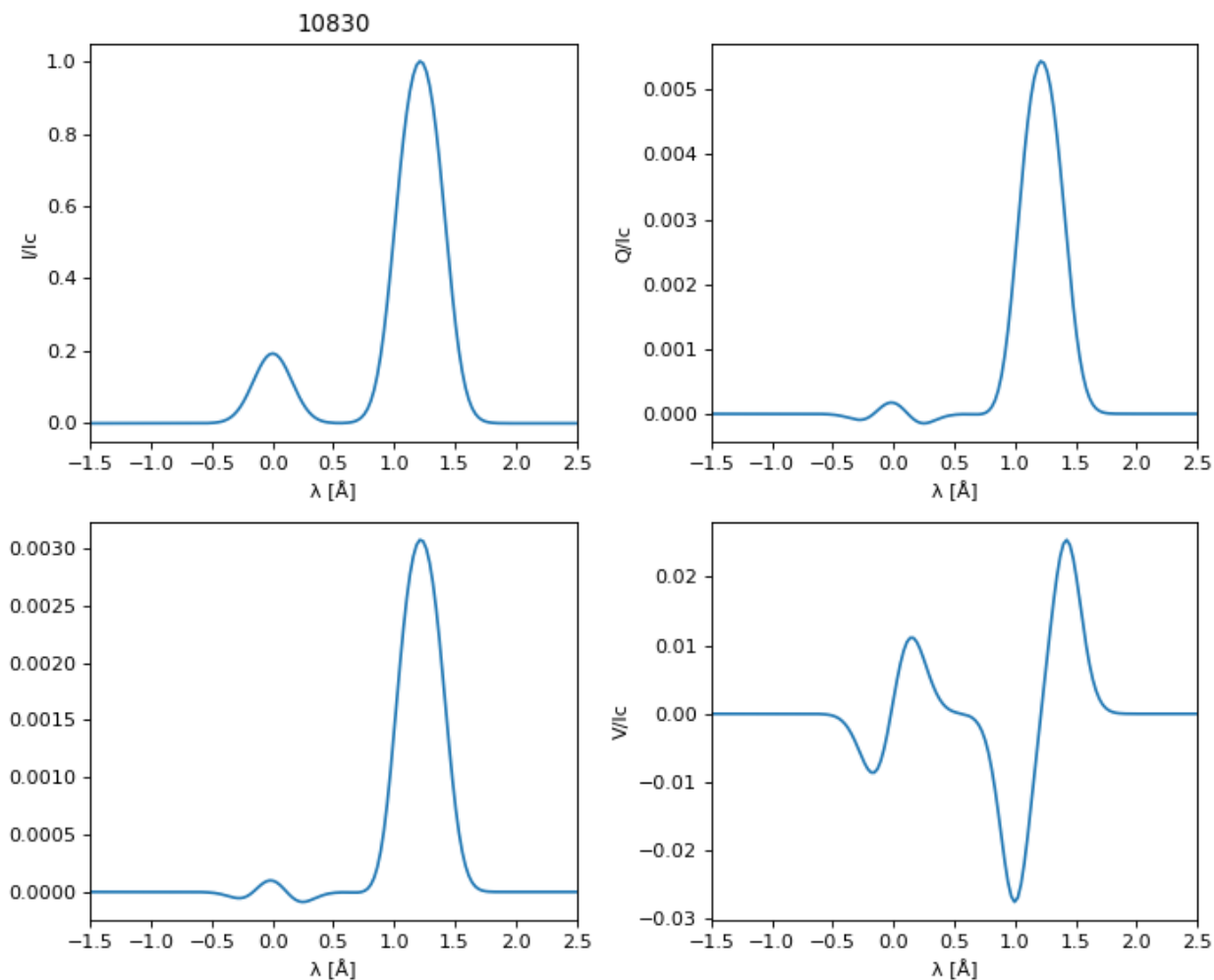
Synthesis GUI

In case you see the following error message:

```
backend = Qt5Agg
...
File ".../lib/python3.9/site-packages/hazel-2018.9.22-py3.9-linux-x86_64.egg/EGG-INF0/scripts/hazelgui", line 22, in <module>
    from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg as FigureCanvas
ModuleNotFoundError: No module named 'matplotlib.backends.backend_qt4agg'
```

Replace line 22 of the file mentioned above with the following:

```
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
```



Component 1

B [G]	0	<input type="range"/>	1000	160.9
θ_B [deg]	0	<input type="range"/>	180	80.0
Φ_B [deg]	-180	<input type="range"/>	180	41.0
width [km/s]	0.10	<input type="range"/>	20	6.0
v [km/s]	-10	<input type="range"/>	10	0.0
τ	0.0	<input type="range"/>	10	1.0

Component 2

B [G]	0	<input type="range"/>	1000	0.0
θ_B [deg]	0	<input type="range"/>	180	0.0
Φ_B [deg]	0	<input type="range"/>	180	0.0
width [km/s]	0	<input type="range"/>	15	6.0
v [km/s]	-10	<input type="range"/>	10	0.0
τ	0	<input type="range"/>	3	1.0

Observation

θ [deg]	<input type="range"/>	90.0
Φ [deg]	<input type="range"/>	0.0
γ [deg]	<input type="range"/>	90.0
Height [arcsec]	<input type="range"/>	3.0
Damping	<input type="range"/>	0.0
Fill fraction	<input type="range"/>	0.0
β_1	<input type="range"/>	1.0
β_2	<input type="range"/>	1.0
Multiplet	<input type="range"/>	10830

Calculate

<input type="button" value="Calculate"/>
<input type="button" value="Save profile"/>
<input type="button" value="Load observation"/>
<input type="button" value="Reset observation"/>

Loaded: none

Normalization

☒ Normalize maximum
 ☐ Normalize peak

 Use Allen I0: Q0: U0: V0: N. slabs: Left wavelength: Right wavelength: N. wavelengths: