

# Stellar Formation and Evolution

## Supplementary Notes 01

---

Kinoshita Daisuke

21 March 2023

Institute of Astronomy, National Central University, Taiwan

# Sample Python scripts

- GitHub repository

- `https://github.com/kinoshitadaisuke`
- `https://github.com/kinoshitadaisuke/ncu\_stellarevolution\_202302`
- `https://github.com/kinoshitadaisuke/ncu\_stellarevolution\_202302/tree/main/s05`

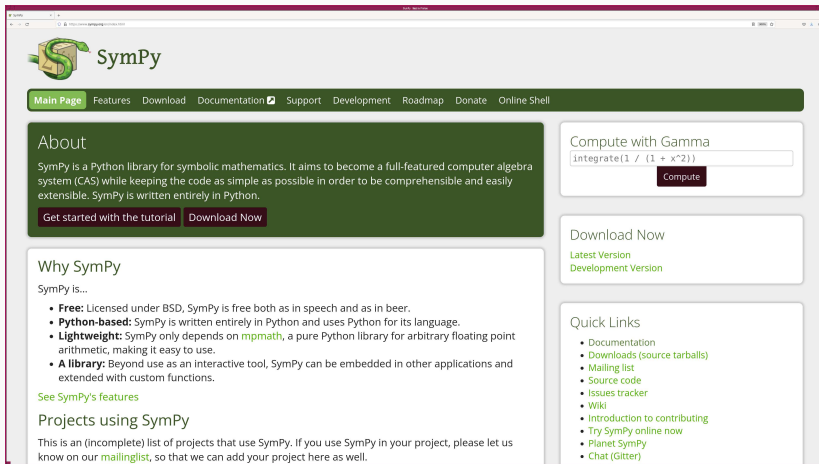
- Binder

- `https://mybinder.org/v2/gh/kinoshitadaisuke/ncu\_stellarevolution\_202302/HEAD`

# Python package “SymPy”

- SymPy
  - A Python package for symbolic mathematics.
  - <https://www.sympy.org/>

# Python package “SymPy”



The screenshot shows the SymPy website homepage. At the top left is the SymPy logo, a green 'S' with a snake-like shape. To its right is the text 'SymPy'. Below the logo is a dark green navigation bar with links: 'Main Page', 'Features', 'Download', 'Documentation', 'Support', 'Development', 'Roadmap', 'Donate', and 'Online Shell'. The 'Documentation' link is highlighted with a white checkmark.

The main content area is divided into several sections:

- About:** A dark green box containing the text: 'SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. SymPy is written entirely in Python.' Below this text are two buttons: 'Get started with the tutorial' and 'Download Now'.
- Why SymPy:** A white box with the heading 'Why SymPy' and sub-heading 'SymPy is...'. It contains a bulleted list:
  - **Free:** Licensed under BSD, SymPy is free both as in speech and as in beer.
  - **Python-based:** SymPy is written entirely in Python and uses Python for its language.
  - **Lightweight:** SymPy only depends on `mpmath`, a pure Python library for arbitrary floating point arithmetic, making it easy to use.
  - **A library:** Beyond use as an interactive tool, SymPy can be embedded in other applications and extended with custom functions.Below the list is a link: 'See SymPy's features'.
- Projects using SymPy:** A white box with the heading 'Projects using SymPy' and text: 'This is an (incomplete) list of projects that use SymPy. If you use SymPy in your project, please let us know on our [mailinglist](#), so that we can add your project here as well.'
- Compute with Gamma:** A white box with a text input field containing the code `integrate(1 / (1 + x^2))` and a dark red 'Compute' button.
- Download Now:** A white box with the heading 'Download Now' and two links: 'Latest Version' and 'Development Version'.
- Quick Links:** A white box with the heading 'Quick Links' and a bulleted list:
  - Documentation
  - Downloads (source tarballs)
  - Mailing list
  - Source code
  - Issues tracker
  - Wiki
  - Introduction to contributing
  - Try SymPy online now
  - Planet SymPy
  - Chat (Gitter)

<https://www.sympy.org/>

# Python package “SymPy”



**SymPy 1.11 documentation**

Q Search

- Installation
- Tutorials
- How-to Guides
- Explanations
- API Reference
- Contributing

Documentation Version

- SymPy 1.11 (latest version)
- SymPy 1.12-dev (dev version)

## Welcome to SymPy's documentation!

A PDF version of these docs is also available.

SymPy is a Python library for symbolic mathematics. If you are new to SymPy, start with the [introductory tutorial](#).

This is the central page for all of SymPy's documentation.

---

## Installation

Instructions on how to install SymPy.

## Tutorials

Tutorials are the best place to start for anyone new to SymPy or one of SymPy's features.

## How-to Guides

How-to guides are step-by-step instructions on how to do specific tasks.

<https://docs.sympy.org/>

# Calculating $\int_0^{10} 2x dx$

## sample Python script #00

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')

# function f(x)
f = 2 * x

# integration of f(x)
I = sympy.integrate (f, (x, 0, 10))

# printing result
print (f'I = {I}')
```

Calculating  $\int_0^{10} 2x dx$

### executing sample Python script #00

```
% chmod a+x star_202302_s05_00.py  
% ./star_202302_s05_00.py  
I = 100
```

# Calculating $\int_0^{\pi} \sin x dx$

## sample Python script #01

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')

# function f(x)
f = sympy.sin (x)

# pi
pi = sympy.pi

# integration of f(x)
I = sympy.integrate (f, (x, 0, pi))

# printing result
print (f'I = {I}')
```



Calculating  $\int_0^{\pi} \sin x dx$

### executing sample Python script #01

```
% chmod a+x star_202302_s05_01.py  
% ./star_202302_s05_01.py  
I = 2
```

# Calculating $\int_0^2 \sqrt{4 - x^2} dx$

## sample Python script #02

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')

# function f(x)
f = sympy.sqrt (4 - x**2)

# integration of f(x)
I = sympy.integrate (f, (x, 0, 2))

# printing result
print (f'I = {I}')
```

Calculating  $\int_0^2 \sqrt{4-x^2} dx$

### executing sample Python script #02

```
% chmod a+x star_202302_s05_02.py  
% ./star_202302_s05_02.py  
I = pi
```

# Calculating $\int_{-\infty}^{\infty} \exp(-x^2) dx$

## sample Python script #03

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')
# function f(x)
f = sympy.exp (-x**2)

# positive infinity
pinf = sympy.oo
# negative infinity
ninf = -sympy.oo
# integration of f(x)
I = sympy.integrate (f, (x, ninf, pinf))

# printing result
print (f'I = {I}')
```

# Calculating $\int_{-\infty}^{\infty} \exp(-x^2) dx$

## executing sample Python script #03

```
% chmod a+x star_202302_s05_03.py  
% ./star_202302_s05_03.py  
I = sqrt(pi)
```

# Calculating $\int_0^{\infty} x^3 \exp(-x^2) dx$

## sample Python script #04

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')

# function f(x)
f = x**3 * sympy.exp (-x**2)

# positive infinity
pinf = sympy.oo

# integration of f(x)
I = sympy.integrate (f, (x, 0, pinf))

# printing result
print (f'I = {I}')
```

Calculating  $\int_0^{\infty} x^3 \exp(-x^2) dx$

### executing sample Python script #04

```
% chmod a+x star_202302_s05_04.py  
% ./star_202302_s05_04.py  
I = 1/2
```

# Calculating $\int_0^{\infty} x^4 \exp(-x^2) dx$

## sample Python script #05

```
# importing sympy module
import sympy

# variable
x = sympy.Symbol ('x')

# function f(x)
f = x**4 * sympy.exp (-x**2)

# positive infinity
pinf = sympy.oo

# integration of f(x)
I = sympy.integrate (f, (x, 0, pinf))

# printing result
print (f'I = {I}')
```



Calculating  $\int_0^{\infty} x^4 \exp(-x^2) dx$

### executing sample Python script #05

```
% chmod a+x star_202302_s05_05.py  
% ./star_202302_s05_05.py  
I = 3*sqrt(pi)/8
```

# Document of “SymPy”

To know more about SymPy, read the official document.



The screenshot shows the SymPy 1.11 documentation website. On the left is a dark green sidebar with the SymPy logo (a green snake wrapped around a yellow cube) and the text "SymPy 1.11 documentation". Below the logo is a search bar and a list of navigation links: "Installation", "Tutorials", "How-to Guides", "Explanations", "API Reference", and "Contributing", each with a downward arrow. At the bottom of the sidebar are "Documentation Version" options: "SymPy 1.11 (latest version)" and "SymPy 1.12-dev (dev version)". The main content area has a white background with the heading "Welcome to SymPy's documentation!". Below this, it states "A PDF version of these docs is also available." and "SymPy is a Python library for symbolic mathematics. If you are new to SymPy, start with the introductory tutorial." It then says "This is the central page for all of SymPy's documentation." Below this are three sections: "Installation" with the subtext "Instructions on how to install SymPy.", "Tutorials" with the subtext "Tutorials are the best place to start for anyone new to SymPy or one of SymPy's features.", and "How-to Guides" with the subtext "How-to guides are step-by-step instructions on how to do specific tasks."

<https://docs.sympy.org/>