Cover Page for Proposal Submitted to the National Aeronautics and Space Administration

NASA Proposal Number

20-OSTFL20-0057

NASA PROCEDURE FOR HANDLING PROPOSALS

This proposal shall be used and disclosed for evaluation purposes only, and a copy of this Government notice shall be applied to any reproduction or abstract thereof. Any authorized restrictive notices that the submitter places on this proposal shall also be strictly complied with. Disclosure of this proposal for any reason outside the Government evaluation purposes shall be made only to the extent authorized by the Government.

SECTION I - Proposal Information

Principal Investigator	E-mail Address	Phone Number
Erik Tollerud	erik.tollerud@gmail.com	651-307-9409

Street Address (1)	Street Address (2)
3736 Tudor Arms Ave	

City	State / Province	Postal Code	Country Code
Baltimore	MD	21211-2245	US

Proposal Title : **Sustaining the Astropy Project**

Proposed Start Date	Proposed End Date	Total Budget	Year 1 Budget	Year 2 Budget	Year 3 Budget
07 / 19 / 2021	07 / 19 / 2024	634,590.00	182,655.00	195,030.00	256,905.00

SECTION II - Application Information

NASA Program Announcement Number	NASA Program Announcement Title
NNH20ZDA001N-OSTFL	E.7 Support for Open Source Tools, Frameworks, and Libraries

For Consideration By NASA Organization *(the soliciting organization, or the organization to which an unsolicited proposal is submitted)*

NASA , Headquarters , Science Mission Directorate , Cross Division

Date Submitted	Submission Method	Grants.gov Application Identifier	Applicant Proposal Identifier
01 / 19 / 2021	Electronic Submission Only		

Type of Application	Predecessor Award Number	Other Federal Agencies to Which Proposal Has Been Submitted
New		

International Participation	Type of International Participation
Yes	Collaborator

SECTION III - Submitting Organization Information

DUNS Number	CAGE Code	Employer Identification Number (EIN or TIN)	Organization Type
068870352	7AQW2		8H

Organization Name (Standard/Legal Name)	Company Division
Numfocus, INC.	

Organization DBA Name	Division Number

Street Address (1)	Street Address (2)
221 W 6TH ST 1550	

City	State / Province	Postal Code	Country Code
AUSTIN	TX	78701	USA

SECTION IV - Proposal Point of Contact Information

Name	Email Address	Phone Number
Erik Tollerud	etollerud@stsci.edu	651-307-9409

SECTION V - Certification and Authorization

Certification of Compliance with Applicable Executive Orders and U.S. Code

By submitting the proposal identified in the Cover Sheet/Proposal Summary in response to this Research Announcement, the Authorizing Official of the proposing organization (or the individual proposer if there is no proposing organization) as identified below:

- certifies that the statements made in this proposal are true and complete to the best of his/her knowledge;
- agrees to accept the obligations to comply with NASA award terms and conditions if an award is made as a result of this proposal; and
- confirms compliance with all provisions, rules, and stipulations set forth in this solicitation.

Willful provision of false information in this proposal and/or its supporting documents, or in reports required under an ensuing award, is a criminal offense (U.S. Code, Title 18, Section 1001).

Authorized Organizational Representative (AOR) Name	AOR E-mail Address	Phone Number
Nicole Foster	nicole@numfocus.org	512-831-2870

AOR Signature *(Must have AOR's original signature. Do not sign "for" AOR.)*	Date

[PAGES REDACTED DUE TO PERSONAL INFORMATION]

PI: Erik Tollerud
Co-I's: Tom Aldcroft, Lia Corrales, Matthew Craig, Kelle Cruz, Adam Ginsburg, Hans Guenther,
Pey Lian Lim, Adrian Price-Whelan
Collaborator: John Swinbank

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION VII - Project Summary

The Python language has seen increasingly wide use in astronomy, including astrophysics, cosmology, and planetary, solar, and space sciences. This has both led to and been influenced by the Astropy Project. Astropy was founded in 2011 with two major objectives: to develop a single core Python package to provide the foundational tools for astronomy research; and to build an ecosystem of usable, interoperable, and collaborative astronomy Python packages. With these goals, it has developed into a true community effort, with participants from a wide range of backgrounds, that is not the product of one particular institution, group, or mission. It has over 20,000 dependent repositories on GitHub alone; at least 5,000 peer-reviewed publications have used it; and it is critical for a range of NASA missions including Chandra, Hubble, and JWST. However, supporting such a wide user base through community effort alone presents a number of sustainability challenges. In particular, most development is done by a few people, infrastructure requires ongoing maintenance, and the number of active core developers is decreasing.

This proposal aims to address these concerns and improve the robustness of the Astropy Project's infrastructure by funding a set of tasks identified by the Project's contributors as particularly difficult to find volunteer effort to pursue. This includes a mix of Project-wide infrastructure improvements, targeted work on specific areas of the code that are in need of support, enhancements to the Astropy education materials to encourage community engagement and with the hope of fostering more contributions, and better support for the "affiliated packages," independent Python packages that collaborate with Astropy and are peer-reviewed for coding, documentation and testing standards. We propose to continue using our established funding model of primarily supporting contributors as part-time contractors, an approach that has been shown to successfully fund work by both existing and new contributors. In short, this proposal aims to provide support for the Astropy ecosystem and ensure that it is healthy and able to continue fulfilling its vision to provide a core package and an associated collaborative ecosystem. While the Project did not require dedicated funding to earn its role in the community, it does now need funding to continue to provide the production-level code and educational materials that the community has come to not just expect from us, but to rely on us to provide.

PI Name : **Erik Tollerud**	NASA Proposal Number 20-OSTFL20-0057
Organization Name : **Numfocus, INC.**	
Proposal Title : **Sustaining the Astropy Project**	

SECTION VIII - Other Project Information

Proprietary Information

Is proprietary/privileged information included in this application?

Yes

International Collaboration

Does this project involve activities outside the U.S. or partnership with International Collaborators?

Yes

Principal Investigator	Co-Investigator	Collaborator	Equipment	Facilities
No	**No**	**Yes**	**No**	**No**

Explanation :

Unfunded Collaborators have designated Astropy project roles that interact with the proposed work.

NASA Civil Servant Project Personnel

Are NASA civil servant personnel participating as team members on this project (include funded and unfunded)?

No

Fiscal Year	Fiscal Year	Fiscal Year	Fiscal Year	Fiscal Year	Fiscal Year
Number of FTEs	Number of FTEs	Number of FTEs	Number of FTEs	Number of FTEs	Number of FTEs

PI Name : **Erik Tollerud**	NASA Proposal Number 20-OSTFL20-0057
Organization Name : **Numfocus, INC.**	
Proposal Title : **Sustaining the Astropy Project**	

SECTION VIII - Other Project Information

Environmental Impact

Does this project have an actual or potential impact on the environment?	Has an exemption been authorized or an environmental assessment (EA) or an environmental impact statement (EIS) been performed?
No	**No**

Environmental Impact Explanation:

Exemption/EA/EIS Explanation:

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION VIII - Other Project Information

Historical Site/Object Impact

Does this project have the potential to affect historic, archeological, or traditional cultural sites (such as Native American burial or ceremonial grounds) or historic objects (such as an historic aircraft or spacecraft)?

No

Explanation:

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION IX - Program Specific Data

Question 1 : Short Title:

Answer: Sustaining the Astropy Project

Question 2 : Type of institution:

Answer: Non-profit Organization

Question 3 : Will any funding be provided to a federal government organization including NASA Centers, JPL, other Federal agencies, government laboratories, or Federally Funded Research and Development Centers (FFRDCs)?

Answer: No

Question 4 : Is this Federal government organization a different organization from the proposing (PI) organization?

Answer: N/A

Question 5 : Does this proposal include the use of NASA-provided high end computing (HEC)?

Answer: No

Question 6 : HEC Request Number

Answer:

Question 7 : Research Category:

Answer: 2) Data analysis/data restoration/data assimilation/Earth System modeling (including Guest Observer Activities)

Question 8 : Flight Services

Answer: No

Question 9 : Team members not confirmed via NSPIRES

Answer:

Question 10 : Does this proposal contain information and/or data that are subject to U.S. export control laws and regulations including Export Administration Regulations (EAR) and International Traffic in Arms Regulations (ITAR)?

Answer: No

Question 11 : I have identified the export-controlled material in this proposal.

Answer: N/A

Question 12 : I acknowledge that the inclusion of such material in this proposal may complicate the government's ability to evaluate the proposal.

Answer: N/A

Question 13 : Does the proposed work include any involvement with collaborators in China or with Chinese organizations, or does the proposed work include activities in China?

Answer: No

The National Environmental Policy Act (NEPA) obligates NASA to consider the potentialenvironmental effects of proposed projects, including those that NASA funds which areimplemented by grantees. The majority of grant-related activities are categorically excluded asresearch and development projects that do not pose adverse environmental impacts. These arecovered by a NASA Grants Record of Environmental Consideration (REC) available athttps://www.nasa.gov/nepa/grants. The following questions enable NASA to confirm that yourproposed activity falls within this blanket REC. Proposals that are not covered will requireadditional NEPA analysis if selected (e.g., filling out an Environmental Checklist) or the completionof NASA's Executive Order (EO) 12114 Checklist for an activity to be conducted abroad. "Yes"responses are not selection criteria, however, if a "Yes" response is marked, proposers shouldconsider NEPA and/or EO compliance in cost and schedule estimates.

Question 14 : Would the proposal involve any activity that includes:a.Construction of new facilities or modification to the footprint of an existing-facility, orb.Ground disturbance (e.g., excavation, clearing of trees, installation of equipment, etc.), orc.Outdoor discharges of water (e.g., waste water runoff), air emissions (e.g., ozone-depletingsubstances) or generation of noise exceeding 115 dBA (excluding those associated with aircraftoperations)?

Answer: No

Question 15 : Would the proposal involve any field activity that would:a.Release equipment (e.g., dropsondes, sensors, etc.) or chemicals (e.g., dyes, tracers, etc.) intothe air, bodies of water or on the ground, orb.Release a parachute or use equipment that would not be recovered, orc.Involve equipment or a payload that contains hazardous (e.g., petroleum, hypergols, oxidizers,solid propellants, etc.) or radioactive materials?

Answer: No

Question 16 : Would the proposal involve the launch of a payload, equipment, or instrument (e.g., via launchvehicle, sounding rocket, balloon, etc.)?

Answer: No

Question 17 : Would the proposal involve any activity to be conducted outside the United States or its territoriesexcluding travel for meetings or conferences?

Answer: No

Question 18 : Comments

Answer:

Question 19 : Does this proposal contain a citizen science component?

Answer: No

High Impact-High Risk: This year SMD will collect information from proposers and reviewers to assess (intellectual) risk and impact of ROSES proposals. Although generally not part of the peer review process, these are programmatic considerations that may be taken into account by the selection official. For the purposes of these questions, the definitions are as follows:A high impact research project is one in which, if confirmed/successful, would have a substantial and measurable effect on current thinking, methods or practice.A high risk research project tests novel and significant hypotheses for which there is scant precedent or preliminary data or that are counter to the existing scientific consensus. This type of risk is different from implementation risk which refers to the likelihood that the proposed research can be successfully conducted as proposed.

Question 20 : This proposal should be considered to have high impact.

Answer: Yes

Question 21 : Impact Explanation

Answer:

The code developed by the Astropy Project has been used by multiple Nobel Laureate groups, demonstrating it's clear potential for having a major impact on scientific thinking. Providing it NASA support for its sustainability will also send a clear message that such open source efforts are supported by the "institutional" players in astro and thus substantially impact how research software for astronomy is developed and funded in the future.

Question 22 : This proposal should be considered to have high (intellectual) risk.

Answer: No

Question 23 : Risk Explanation

Answer:

Question 24 : Please provide a link (URL) to the Open Source tools, frameworks, or libraries to be supported.

Answer: https://www.astropy.org/

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Cumulative Budget

Budget Cost Category	Funds Requested ($)			
	Year 1 ($)	Year 2 ($)	Year 3 ($)	Total Project ($)
A. Direct Labor - Key Personnel	0.00	0.00	0.00	0.00
B. Direct Labor - Other Personnel	0.00	0.00	0.00	0.00
Total Number Other Personnel	0	0	0	0
Total Direct Labor Costs (A+B)	0.00	0.00	0.00	0.00
C. Direct Costs - Equipment	0.00	0.00	0.00	0.00
D. Direct Costs - Travel	0.00	0.00	0.00	0.00
Domestic Travel	0.00	0.00	0.00	0.00
Foreign Travel	0.00	0.00	0.00	0.00
E. Direct Costs - Participant/Trainee Support Costs	0.00	0.00	0.00	0.00
Tuition/Fees/Health Insurance	0.00	0.00	0.00	0.00
Stipends	0.00	0.00	0.00	0.00
Travel	0.00	0.00	0.00	0.00
Subsistence	0.00	0.00	0.00	0.00
Other	0.00	0.00	0.00	0.00
Number of Participants/Trainees				0
F. Other Direct Costs	166,050.00	177,300.00	233,550.00	576,900.00
Materials and Supplies	0.00	0.00	0.00	0.00
Publication Costs	0.00	0.00	0.00	0.00
Consultant Services	166,050.00	177,300.00	233,550.00	576,900.00
ADP/Computer Services	0.00	0.00	0.00	0.00
Subawards/Consortium/Contractual Costs	0.00	0.00	0.00	0.00
Equipment or Facility Rental/User Fees	0.00	0.00	0.00	0.00
Alterations and Renovations	0.00	0.00	0.00	0.00
Other	0.00	0.00	0.00	0.00
G. Total Direct Costs (A+B+C+D+E+F)	166,050.00	177,300.00	233,550.00	576,900.00
H. Indirect Costs	16,605.00	17,730.00	23,355.00	57,690.00
I. Total Direct and Indirect Costs (G+H)	182,655.00	195,030.00	256,905.00	634,590.00
J. Fee	0.00	0.00	0.00	0.00
K. Total Cost (I+J)	182,655.00	195,030.00	256,905.00	634,590.00
Total Cumulative Budget				634,590.00

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2021	**07 / 18 / 2022**	**Project**	**1**

A. Direct Labor - Key Personnel

Name	Project Role	**Base Salary ($)**	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	**Fringe Benefits ($)**	Funds Requested ($)
Tollerud, Erik	**PI**	**0.00**				**0.00**	**0.00**	**0.00**
							Total Key Personnel Costs	**0.00**

B. Direct Labor - Other Personnel

Number of Personnel	Project Role	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	Fringe Benefits ($)	Funds Requested ($)
0	Total Number Other Personnel					Total Other Personnel Costs	**0.00**
						Total Direct Labor Costs (Salary, Wages, Fringe Benefits) (A+B)	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2021	**07 / 18 / 2022**	**Project**	**1**

C. Direct Costs - Equipment

Item No.	Equipment Item Description	Funds Requested ($)
	Total Equipment Costs	**0.00**

D. Direct Costs - Travel

	Funds Requested ($)
1. Domestic Travel (Including U.S. Territories and Possessions)	**0.00**
2. Foreign Travel (Including Canada and Mexico)	**0.00**
Total Travel Costs	**0.00**

E. Direct Costs - Participant/Trainee Support Costs

		Funds Requested ($)
1. Tuition/Fees/Health Insurance		**0.00**
2. Stipends		**0.00**
3. Travel		**0.00**
4. Subsistence		**0.00**
Number of Participants/Trainees:	Total Participant/Trainee Support Costs	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2021	**07 / 18 / 2022**	**Project**	**1**

F. Other Direct Costs

	Funds Requested ($)
1. Materials and Supplies	**0.00**
2. Publication Costs	**0.00**
3. Consultant Services	**166,050.00**
4. ADP/Computer Services	**0.00**
5. Subawards/Consortium/Contractual Costs	**0.00**
6. Equipment or Facility Rental/User Fees	**0.00**
7. Alterations and Renovations	**0.00**
8. Other:	**0.00**
9. Other:	**0.00**
10. Other:	**0.00**
Total Other Direct Costs	**166,050.00**

G. Total Direct Costs

	Funds Requested ($)
Total Direct Costs (A+B+C+D+E+F)	**166,050.00**

H. Indirect Costs

	Indirect Cost Rate (%)	**Indirect Cost Base ($)**	Funds Requested ($)
NumFOCUS De Minimis Rate	**10.00**	**166,050.00**	**16,605.00**
Cognizant Federal Agency: **None**		Total Indirect Costs	**16,605.00**

I. Direct and Indirect Costs

	Funds Requested ($)
Total Direct and Indirect Costs (G+H)	**182,655.00**

J. Fee

	Funds Requested ($)
Fee	**0.00**

K. Total Cost

	Funds Requested ($)
Total Cost with Fee (I+J)	**182,655.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2022	**07 / 18 / 2023**	**Project**	**2**

A. Direct Labor - Key Personnel

Name	Project Role	**Base Salary ($)**	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	**Fringe Benefits ($)**	Funds Requested ($)
Tollerud, Erik	**PI**	**0.00**				**0.00**	**0.00**	**0.00**
							Total Key Personnel Costs	**0.00**

B. Direct Labor - Other Personnel

Number of Personnel	Project Role	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	Fringe Benefits ($)	Funds Requested ($)
0	Total Number Other Personnel					Total Other Personnel Costs	**0.00**

Total Direct Labor Costs (Salary, Wages, Fringe Benefits) (A+B)	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057
Proposal Title : **Sustaining the Astropy Project**	

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2022	**07 / 18 / 2023**	**Project**	**2**

C. Direct Costs - Equipment

Item No.	Equipment Item Description	Funds Requested ($)
	Total Equipment Costs	**0.00**

D. Direct Costs - Travel

	Funds Requested ($)
1. Domestic Travel (Including U.S. Territories and Possessions)	**0.00**
2. Foreign Travel (Including Canada and Mexico)	**0.00**
Total Travel Costs	**0.00**

E. Direct Costs - Participant/Trainee Support Costs

		Funds Requested ($)
1. Tuition/Fees/Health Insurance		**0.00**
2. Stipends		**0.00**
3. Travel		**0.00**
4. Subsistence		**0.00**
Number of Participants/Trainees:	Total Participant/Trainee Support Costs	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 19 / 2022	**07 / 18 / 2023**	**Project**	**2**

F. Other Direct Costs

	Funds Requested ($)
1. Materials and Supplies	**0.00**
2. Publication Costs	**0.00**
3. Consultant Services	**177,300.00**
4. ADP/Computer Services	**0.00**
5. Subawards/Consortium/Contractual Costs	**0.00**
6. Equipment or Facility Rental/User Fees	**0.00**
7. Alterations and Renovations	**0.00**
8. Other:	**0.00**
9. Other:	**0.00**
10. Other:	**0.00**
Total Other Direct Costs	**177,300.00**

G. Total Direct Costs

	Funds Requested ($)
Total Direct Costs (A+B+C+D+E+F)	**177,300.00**

H. Indirect Costs

	Indirect Cost Rate (%)	**Indirect Cost Base ($)**	Funds Requested ($)
NumFOCUS De Minimis Rate	**10.00**	**177,300.00**	**17,730.00**
Cognizant Federal Agency: **None**		Total Indirect Costs	**17,730.00**

I. Direct and Indirect Costs

	Funds Requested ($)
Total Direct and Indirect Costs (G+H)	**195,030.00**

J. Fee

	Funds Requested ($)
Fee	**0.00**

K. Total Cost

	Funds Requested ($)
Total Cost with Fee (I+J)	**195,030.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057
Proposal Title : **Sustaining the Astropy Project**	

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 18 / 2023	**07 / 19 / 2024**	**Project**	**3**

A. Direct Labor - Key Personnel

Name	Project Role	**Base Salary ($)**	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	**Fringe Benefits ($)**	Funds Requested ($)
Tollerud, Erik	**PI**	**0.00**				**0.00**	**0.00**	**0.00**
							Total Key Personnel Costs	**0.00**

B. Direct Labor - Other Personnel

Number of Personnel	Project Role	Cal. Months	Acad. Months	Summ. Months	Requested Salary ($)	Fringe Benefits ($)	Funds Requested ($)
0	Total Number Other Personnel					Total Other Personnel Costs	**0.00**
						Total Direct Labor Costs (Salary, Wages, Fringe Benefits) (A+B)	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 18 / 2023	**07 / 19 / 2024**	**Project**	**3**

C. Direct Costs - Equipment

Item No.	Equipment Item Description	Funds Requested ($)
	Total Equipment Costs	**0.00**

D. Direct Costs - Travel

	Funds Requested ($)
1. Domestic Travel (Including U.S. Territories and Possessions)	**0.00**
2. Foreign Travel (Including Canada and Mexico)	**0.00**
Total Travel Costs	**0.00**

E. Direct Costs - Participant/Trainee Support Costs

		Funds Requested ($)
1. Tuition/Fees/Health Insurance		**0.00**
2. Stipends		**0.00**
3. Travel		**0.00**
4. Subsistence		**0.00**
Number of Participants/Trainees:	Total Participant/Trainee Support Costs	**0.00**

PI Name : **Erik Tollerud**	NASA Proposal Number
Organization Name : **Numfocus, INC.**	20-OSTFL20-0057

Proposal Title : **Sustaining the Astropy Project**

SECTION X - Budget

Start Date :	End Date :	Budget Type :	Budget Period :
07 / 18 / 2023	**07 / 19 / 2024**	**Project**	**3**

F. Other Direct Costs

	Funds Requested ($)
1. Materials and Supplies	**0.00**
2. Publication Costs	**0.00**
3. Consultant Services	**233,550.00**
4. ADP/Computer Services	**0.00**
5. Subawards/Consortium/Contractual Costs	**0.00**
6. Equipment or Facility Rental/User Fees	**0.00**
7. Alterations and Renovations	**0.00**
8. Other:	**0.00**
9. Other:	**0.00**
10. Other:	**0.00**
Total Other Direct Costs	**233,550.00**

G. Total Direct Costs

	Funds Requested ($)
Total Direct Costs (A+B+C+D+E+F)	**233,550.00**

H. Indirect Costs

	Indirect Cost Rate (%)	**Indirect Cost Base ($)**	Funds Requested ($)
NumFOCUS De Minimis Rate	**10.00**	**233,550.00**	**23,355.00**
Cognizant Federal Agency: **None**		Total Indirect Costs	**23,355.00**

I. Direct and Indirect Costs

	Funds Requested ($)
Total Direct and Indirect Costs (G+H)	**256,905.00**

J. Fee

	Funds Requested ($)
Fee	**0.00**

K. Total Cost

	Funds Requested ($)
Total Cost with Fee (I+J)	**256,905.00**

Sustaining the Astropy Project

Scientific/Technical/Management

1. Introduction, Background, and Capabilities

Over the last decade, the Python language has seen increasingly wide use in astronomy (including astrophysics, cosmology, and planetary, solar, and space sciences). It is easy to learn, powerful, free, and open source. Python's attributes have enabled astronomers to use, write, and share analysis tools at low cost and has resulted in a strong ecosystem of scientific Python packages, such as NumPy, SciPy, and Matplotlib.

The Astropy Project was founded in 2011 with two goals: to develop a single core Python package to provide the foundational tools for astronomy research; and to build an ecosystem of usable, interoperable, and collaborative astronomy Python packages. The Project has been successful in both of these objectives. The foundational tools provided by the core package are described in §1.1. Around this foundation, the community has built an ecosystem of around 50 closely related packages. These packages provide domain-specific functionality such as reducing CCD images and spectral data, querying astronomical web databases, and analyzing gamma-ray data. These packages now underpin a significant portion of astronomical research and have been used in the preparation of thousands of refereed publications. In Figure 1, we summarize other metrics indicating our strong development community and user base.

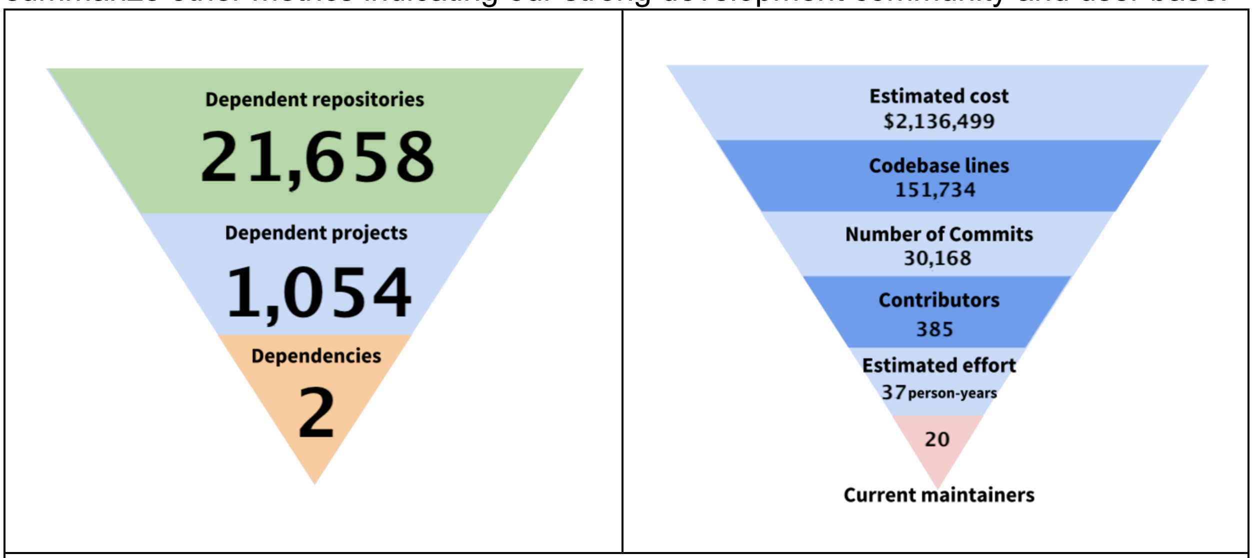

Figure 1. Astropy effort and dependency statistics. Dependent repositories are independent public repositories on GitHub that use `astropy`, dependent projects are a subset of these that are released on PyPI. *Source: GitHub and Openhub.*

The Astropy Project is a true community effort, with participants from a wide range of backgrounds and is not the product of one particular institution, group, or mision. The community includes researchers who volunteer their time, student interns, and software developers from various institutions. The interests of these contributors, and the time they are willing and able to volunteer, drive the direction of work on the Project. This presents a number of sustainability challenges, as described in §1.2.

The Project recently received funding from the Gordon & Betty Moore Foundation to develop a sustainable governance structure, to support the activities of a small number of key contributors, and to enable the Project to be attractive to federal funding agencies. That Moore grant, however, is not a source of *sustained* support. With this proposal, we are meeting one of the key deliverables of that grant and hope that it represents the beginning of the much-needed long-term support for the Astropy Project, a fundamental software and educational project impacting all corners of astronomy.

In the remainder of this section, we describe the capabilities of the Astropy Project and the challenges it faces. In §2, we describe our relevance to the NASA SMD science community and our relationship to NASA SMD's scientific vision and data strategic plan. In §3, we describe our governance and development workflow. Finally**, in §4, we describe the four major areas of the Astropy Project that we are requesting funds to support: 1) maintain and develop project-wide infrastructure; 2) to address key maintenance challenges in the core and coordinated packages; 3) and to ensure the Project's tutorials and documentation ecosystem remain current; and 4) support the maintenance and upkeep of the affiliated packages and the broader Astropy ecosystem.**

1.1. Capabilities Provided by the Astropy Project

Below we describe the range of capabilities the Astropy Project provides to the community. These capabilities include supporting infrastructure and documentation in addition to the functionality of the core software.

1.1.1. Functionality

The Astropy Project, including both the core package and its wider ecosystem, provides the following capabilities:

Manipulation of units, constants, and quantities: The `astropy` core package includes a robust system for associating units with numbers, or arrays of numbers, and tracking them through calculations. This is one of the most fundamental and widely used subpackages in the core package, and is used outside of the astronomical community (e.g., in plasma physics & biology). It is complemented by a selection of commonly-used constants from CODATA (Mohr, Newell & Taylor 2016), resolutions of the International Astronomical Union (Mamajek et al 2015a, 2015b) and other sources.

Representations of space and time: The `astropy` core package provides mechanisms for representing positions and velocities of objects. It is capable of handling different reference systems (e.g., ICRS, Galactic) and representations (spherical, cartesian), and provides convenient functionality for converting between them. It can also include time-dependent positions and velocities. Similar functionality is provided for time, including conversion between common time scales such as Universal Time, International Atomic Time, and Terrestrial Time, and representing times using different formats (e.g., Julian, ISO 8601). This functionality has been used to implement the current de-facto standard for representing world coordinate system (WCS) transformations (Calabretta & Greisen 2002).

Data containers and interfaces for reading and writing data: The `astropy` core package provides classes for representing and manipulating tabular and *n*-dimensional gridded data, with support for astronomical semantics, naming conventions, other domain-specific functionality and data formats like FITS, VOTable, and HDF5. Affiliated packages such as `ccdproc` built on this functionality to provide data reduction and analysis capabilities.

Computation, analysis, and visualization: The `astropy` core package provides astronomy-specific analysis tools, including a range of statistical functionality—such as calculation of periodograms and spatial correlation functions—and routines to assist in visualizing astronomical images.

1.1.2. Infrastructure in Support of Astropy Core and Affiliated packages and the Community

The Project maintains a number of critical infrastructure components and services, which are used not only by both Astropy's core and affiliated packages, but also more widely across the astronomical community and beyond. These include:

- A continuous integration service based on GitHub Actions and CircleCI.
- Plugins for the `pytest` testing framework and the Sphinx documentation system.
- Templates for setting up code repositories with appropriate packaging, testing, and documentation.
- Automated systems that interact with contributors on GitHub to efficiently manage and merge their contributions.

1.1.3. Educational Materials, including Documentation, Guides, Tutorials, and Workshops

The Astropy Project's position within the astronomical community makes it essential to provide appropriate educational materials and documentation. We have therefore developed the "Learn Astropy" ecosystem, which provides a range of *examples* (code snippets addressing a single task), *tutorials* (lessons which demonstrate the use of `astropy` modules within the context of astronomical research) and *guides* (longer "cookbooks" which teach data analysis methods alongside `astropy` functionality). The Learn Astropy team also conducts workshops, which introduce astronomers to the Astropy functionality and ecosystem. When these workshops are presented at meetings of the American Astronomical Society, they regularly draw ~100 attendees.

1.2. Key Sustainability Challenges

Our main sustainability challenges are the following:

- **Most development is done by a few people**: Despite having many—over 310—individual contributors, core development is dominated by a few developers: 75% of commits were contributed by ten developers. There are critical parts of the package that only one or two developers are properly familiar with.

- **Infrastructure requires ongoing maintenance**: The Project infrastructure for testing, building documentation, and creating new packages requires significant time to sustain. It has been challenging to find volunteers to work on these less high-profile tasks.
- **The number of active core developers is decreasing**: As the Project has matured and become more complex, it takes more effort for new contributors to become effective. At the same time, veterans are scaling-back their involvement or leaving the Project as their careers progress.

Development of open-source software benefits the entire scientific community, from individual researchers to future large missions. Without dedicated funding, it has become difficult to find people to work on critical, but less visible, parts of the Project. **Dedicated funding is now critical for the future sustainability of the Astropy Project and we describe four major areas in need of support in §4**.

2. Relevance to the NASA SMD Science Community and the Relationship to NASA SMD's Scientific Vision and Data Strategic Plan

2.1. Current Usage and Impact to the SMD Community

Software developed and maintained by the Astropy Project software is deeply embedded in the landscape of US ground- and space-based Astronomy. It constitutes basic infrastructure for our field and, without dedicated funding, represents a vulnerability to many missions. Investment in development of the Astropy Project impacts nearly every single federally funded Astronomy project.

2.1.1. Ground-Based Astronomical Observatories

The `astropy` core library has been widely adopted or used as a template for both pipeline development and user-facing analysis tools to support observatory operations and science. Examples of this include the Vera C. Rubin Observatory LSST Science Pipelines, Gemini (replacing IRAF), and the Daniel K. Inouye Solar Telescope (WCS-described *n*-dimensional data).

2.1.2. Space-based Observatories

`astropy` is a core dependency of a broad spectrum of flight-critical operations software for the **Chandra X-ray Observatory**. This includes software that is used to generate and validate the spacecraft command loads that execute the science mission. In addition, much of the spacecraft engineering and operations analysis uses an integrated Python analysis infrastructure that depends heavily on `astropy`.

For the **Hubble Space Telescope** (HST), newer developments such as the Cosmic Origins Spectrograph pipeline and recent user data analysis tools depend heavily on `astropy`. Use of community-wide packages has already reaped benefits: development done in support of one mission is now available to other missions. For example, HST benefitted from JWST contributions to the `photutils` Astropy coordinated package. Since these contributions were made in community software, the tools developed were also available for HST and the broader community, instead of being isolated to JWST.

The Kepler team has written the `lightkurve` package for data mining and visualization of star data from **Kepler** and **TESS,** using the `astropy` package and the `astroquery` coordinated package. This software's benefit to the Kepler and TESS missions is demonstrated by the dozens of participants that attended a `lightkurve` workshop presented at the latest meeting of the American Association of Variable Star Observers.

Both the data reduction pipeline for the **James Webb Space Telescope** (JWST) and the post-pipeline data analysis tools for the community depend heavily on `astropy`: it is the core library underlying almost all of the scientist-level software for JWST. This has led to significant code contribution to `astropy` as a part of the mission. Use of `astropy` also reduces user training requirements by relying on common interfaces that many users will already know. `astropy` is also used in JWST spacecraft operations engineering analysis via tools that provide fast and convenient access to mission telemetry within a Python analysis environment.

Development on the **Nancy Grace Roman Space Telescope** is at an early stage but the current intent is to build on the legacy of JWST. This means `astropy` is very likely to have a key role. Additionally, the Science Investigation Team model for development of some Roman scientific software implies a need for libraries that are shared between the facility and the astronomy community. This sort of "shared space" is exactly the goal of the Astropy Project as initially conceived and as implemented.

2.2. Relationship to SMD's Strategic Vision and Data Strategic Plan

The goals and functionality of the Astropy Project are strongly aligned with the NASA Science Vision for 2020–2024. Here, we describe how the project vision and management supports each of NASA's four pillars.

Exploration and Scientific Discovery: The core `astropy` library provides Python tools that are essential to every subfield of astrophysics: from analytical calculations supported by the `units` package to reading and writing FITS files and data tables, and querying publicly available NASA and Vizier archives. By providing a Python framework for astronomical research, the Astropy Project enables cross-disciplinary use of well-maintained Python tools to enhance discovery (Strategy 1.3).

Innovation: The Astropy Project goal to write and maintain research-grade code libraries through a community effort is a highly innovative endeavor (Strategy 2.1). The Project is forging a new path for research software development on many fronts, including use of industry-standard best practices (version control, rigorous testing), group collaboration, formal governance, and funding management (Strategy 2.4). The Astropy Project also fosters a culture of collaboration, respect, and a decision making process that values compromise in order to achieve group consensus (Strategy 2.2).

Interconnectivity and Partnerships: Software developed and maintained by the Astropy Project is used by numerous academic and national institutions, from ground based operations like the Vera C. Rubin and Gemini observatories to space based telescopes such as *Chandra* and JWST, and future missions like the Roman Space Telescope (Strategy 3.4). The Astropy Project is a sponsored project of NumFOCUS, a non-profit organization that specializes in supporting open source scientific computing software for the benefit of both research and industry (Strategy 3.3).

Inspiration: The Astropy Project places a high value on diversity and inclusion, supported both through our Code of Conduct and a dedication to work with an international team. With support from the Moore Foundation, the Project has supported initiatives on inclusion, diversity, and empowerment that include attending national diversity conferences, internships through Outreachy that specifically target diversity, and a peer-mentoring group for women-of-color coders (Strategy 4.1).

3. Project Management

3.1. Governance

Since its inception, the Astropy Project has been open and community driven. Anyone can participate in discussions or suggest changes to the code. Contributions are peer-reviewed to maintain strict standards for package quality and stability. This process is followed not just for code or educational materials, but also governance and organizational items.

We are in the process of formalizing our governance process as a Governance Charter (APE 0, Tollerud et al. 2020). The Project will continue to be directed by a four person Coordinating Committee, guided by the community of "voting members" of the Project. The criteria for becoming a voting member are described in the Charter. The Charter also describes the process for the Coordinating Committee to delegate decisions to committees and the voting members.

3.1.1. Management and Finance

The Astropy Finance Committee (FC) is tasked with the fiscal, staff management, and reporting tasks that are required to spend money. It communicates weekly with NumFOCUS, the fiscal sponsor, which is the legal entity executing contracts and administering grants for the Astropy Project. It also maintains a policy of openness with the Astropy community, and most decisions are made in consultation with the wider community. The FC orchestrates the community processes which determine funding priorities for the Astropy Project, organizes the selection and hiring of contractors (either directly or through a hiring committee for a particular position), provides oversight to ongoing engagements, authorizes the use of funds for travel, conferences, and purchases, and controls and approves all Astropy Project related invoices. In 2019, the Astropy Project was awarded a major grant from the Betty and Gordon Moore foundation. With this money, the FC has funded about a dozen of the current Astropy maintainers as contractors or through sub-grants, negotiated work statements, and has overseen work and invoices. Additionally, a Software Operational Support Specialist was identified through a formal job search process, who has been working under contract since August 1, 2020.

3.2. Development Workflow

The Astropy Project uses a workflow that is fairly common in the open-source community. The `astropy` source code is version controlled with the Git version control system and hosted in a public repository on GitHub. Most bug fixes, documentation enhancements, and new features are contributed through pull requests. Anyone can create a fork (essentially a local copy) of the project, implement changes, and submit a pull request. A continuous integration service runs the entire test suite on multiple platforms and Python versions on the changed code to ensure that the requested code change does not introduce any regressions. When there is general agreement and one of the current maintainers approves of the requested changes, they are merged into the main codebase. This process can take from a few hours for simple changes, to several months, depending on the complexity of the change, the availability of reviewers, and the responsiveness of the pull request author to any review comments.

A separate process exists for major changes, like the addition of an entire subpackage. In this case, the proposed change is described in an “Astropy Proposal for Enhancement” (APE). Each APE includes a summary of proposed changes, a decision rationale, and the proposed interface in pseudo-code. In this way, the community can discuss the APE without wasting time on coding up a change that might not be accepted in this way.

3.3. Licenses for Core `astropy` Package and Other Packages in the Ecosystem

The core `astropy` package and the coordinated packages are licensed under a BSD 3-clause license, which has very permissive terms for code reuse. This license was chosen early in the project in part to encourage use by others without requiring users to share their code in turn, as would be required under a GPL-style license. Other affiliated packages are free to choose any open-source license.

3.4. Metrics for Assessing Sustainability

There is not one metric to uniquely assess the sustainability for a software project and its community. We thus look at a number of indicators for different aspects. We track the adoption of `astropy` in astronomy research by looking at the number of citations in the NASA ADS (about 100 refereed citations per month at the end of 2020, about twice the number from three years earlier). The `astropy` core package has about 25 unique contributors per month (stable since 2014); for affiliated and coordinated packages, this number is harder to measure due to small-number statistics. We also look at the number of open and closed issues in the issue trackers and contributed pull requests, though these numbers have to be treated with caution. When new issues are opened at a faster rate than issues are closed, that could indicate that a package is not in a sustainable state and more attention is required. A large number of issues can signify growing involvement of the community, which submits more requests for features, but it could also be due to a growing back-log of unhandled bug reports. In summary, we use a holistic view of many various metrics to assess the current status of the Project and individual packages.

3.5. Inclusiveness and Community Development

The Astropy Project takes inclusiveness seriously and has taken steps like adopting a Code of Conduct and naming an independent ombudsperson early. The Project aims to be accessible to contributors and users worldwide and from all interested backgrounds. The open development model described in §3.2 and detailed development guides are meant to lower the bar for new contributors. Furthermore, the software operational source specialist hired via our Moore Grant is explicitly tasked with improving the new contributor experience.

At the same time, the project recognizes persistent challenges. The number of contributors for the core package is stable, but not growing, and heavily centered on major US institutions, with the majority of the contributors self-identifying as white males. Again, using funding from the Moore grant, the Astropy Project just started two major initiatives to address this situation: A peer mentoring for the Women of Color coding community is just getting started, and an intern, selected through the Outreachy project, which connects open source communities to underrepresented groups has been recruited.

The Astropy Project also provides significant infrastructure to help others in the community to set up, develop, and distribute specialized packages. Most of the packages are developed by the SMD community and directly foster SMD strategic objectives. The Astropy Project thus acts as a multiplier to reduce the work required for many SMD research projects.

This work of the Astropy Project is done in close collaboration with other SMD relevant projects. In particular, the Astropy Project is one of the founding members of OpenAstronomy, a collaboration of open-source astronomy related packages and work, code, and infrastructure is shared with other OpenAstronomy projects, most notably SunPy. This collaboration reduces the maintenance needs for all involved projects and is enabled by coordinating development plans and having contributors who are active in both projects.

4. Project Description: A Vision for Sustaining the Astropy Project

Below, we describe four major areas of the Astropy Project which require funds to maintain and meet the needs of the astronomy community: 1) project-wide infrastructure, 2) core and coordinated packages, 3) educational resources, and 4) affiliated packages. We propose to continue using the funding model of primarily supporting contributors as part-time contractors we initiated with our Moore grant. We have proven that this model works via the process to fill the Software Operations Support Specialist role. The number of high quality applications received for this position convinced us that there is a deep pool of expertise which can be tapped as contract-labor to address the challenges facing the Project.

Because of the presence of such expertise outside the traditional academic structure, we are not using the approach of subawards to explicitly named Co-I for this proposal. Instead, the decision of who will do the work will be done collaboratively as part of the Project's open decision-making model. Hence, we do not list specific named individuals at this time, but rather our budget calls for the work being done by contractors. The contractors selected will be contracted by NumFOCUS to do the work outlined here. We have used the funding from the Moore Foundation to prove that this model works.

4.1. Project-wide Infrastructure

As described below, we request a total of 420 hours in Year 1, 495 hours in Year 2, and 870 hours in Year 3 to maintain and improve the Project-wide infrastructure.

4.1.1. Support for all Project Maintainers

The Project is currently steered by a four-person Coordinating Committee and approximately 30 individuals who fill formalized roles in the Project. These roles include a community engagement coordinator, release coordinators, core maintainers for various parts of the `astropy` core package, among others. These team members perform the critical work of making the Project run and keeping it maintained. They are responsible for handling issues and providing reviews on pull requests. We will not be able to retain these team members without sustained funding. In order to retain key members of this team after the Moore grant ends, we require 125 hours of developer time in Year 2 and 500 hours in Year 3.

4.1.2. Continuous Integration

A critical part of the Astropy ecosystem is the infrastructure that we maintain for testing, documentation, and package management. This infrastructure requires regular maintenance as the underlying technology and services evolve. Continuous integration (CI) is used to run unit and regression tests across the project and its affiliated packages. Funding requested in this proposal will be used for both improvements and operations:

1. **Addressing issues as they occur**: from small (e.g., new Numpy feature breaking existing `astropy` tests) to big (e.g., refactoring the whole infrastructure when Travis CI dropped support for Open Source Software). These issues often affect the entire project because we have centralized CI support. While the amount of time it takes to address an issue varies widely (from an hour for a small issue to dozens of hours for a major refactor), it is important to have someone who can address these issues promptly as they arrive. On average, it will take 120 hours per year for this work.
2. **Improving the robustness of CI:** by reducing the runtime of the slowest tests and documentation page builds, improving existing workflows in GitHub Actions, and implementing GitHub Action templates (when supported). For an expert, progress can be made with 60 hours per year, and current personnel can help on-board a new hire. However, the effort required for this will depend on the developer's expertise and familiarity with the project.

3. **Improving the developer documentation related to CI**: updating testing instructions in `astropy`, keeping the package template in sync (§4.1.2), and creating a relevant FAQ. Work estimate for this is 40 hours in the first year and 20 hours per year thereafter.

4.1.3. Package Template

The Astropy package template is widely used within the Astropy community and beyond to simplify the setup process for creating new Python packages. The package template is designed to reduce barriers for scientists to package their code, thus encouraging open code sharing, fostering collaborations, and enabling the Astropy ecosystem to grow. For example, new packages can be set up with the Astropy package template using a user-friendly command-line interface that asks the user a series of yes/no questions, then automatically creates a package layout that meets the needs of each individual user. The package template also serves as a demonstration of best practices in Python packaging with an emphasis on common needs for scientific developers; it is therefore an important educational resource for the broader scientific community. The Astropy package template is currently used by over 300 packages hosted on GitHub and has been an important catalyst for the ever-growing list of Astropy affiliated packages that build upon the `astropy` core library to provide a more complete set of tools for all astronomers to use.

One of the principal challenges we have faced with providing a robust template for Python packages is that the official packaging tools and recommendations for best practices have changed significantly—and multiple times—over the past few years. We therefore have faced issues with (1) providing documentation that serves users with the right level of detail and context to update their packages to conform to new standards, and (2) maintaining the template itself and updating the template code to use these new standards. While we do not anticipate further major changes to the Python packaging landscape, we have a backlog of maintenance issues (~48 open GitHub issues) mainly related to improving the user-facing configuration step of running the package template setup and updating the package template documentation to provide better guidance to users who may want more information about how and why the template initializes a Python package layout. We estimate that these issues could be resolved with 100 hours of developer time per year (~2 hours per week).

One of our goals for the future of the Astropy package template is to continue to reduce the maintenance burden for other package maintainers (in this case, our “users”) for keeping their packages up-to-date with new tools and standards. For example, a significant opportunity for helping to keep packages that use the template in sync with the main template would be to disseminate updates to the template out to each derived package by automatically opening pull requests that implement the changes for the maintainers. This would be a substantial undertaking (both this proposal and one submitted to this call by the SunPy Project address this long standing need). This shared effort is intentional, not duplicative, and both teams would be able to simultaneously use the funding on this work. We estimate that implementing this functionality, and supporting new efforts to identify ways to continue improving the utility of the package template, requires an additional investment of 50 hours of developer time per year (~4 hours per month).

4.1.4. Contributor Support

Recent new contributors to Astropy Project software have expressed frustration with our contributor guide. It needs a clearer “quick start” section that supports getting new contributors up and running quickly. The longer contributing guide needs a substantial update to reflect the evolution in underlying technologies and regular maintenance afterward to keep it up to date as that technology continues to change. This effort requires 70 hours in Year 1 and 20 hours in Years 2 and 3.

4.2. Sustaining `astropy` Core and Coordinated Packages

While the Astropy Project provides a range of services, the most recognized part of the Project is the code we produce. These Python packages are called the Astropy Coordinated packages, the most prominent of which is the core `astropy` package. The Astropy Project commits to maintaining both the core package and the coordinated packages. Coordinated packages are more specialized than the core (e.g., `specutils`) or infrastructure packages that are used throughout the community (e.g., `astropy-sphinx`). As described below, we request a total of 447 hours in each of Years 1–3 to maintain and improve our key deliverable, the software itself.

4.2.1. Core Package: `io.fits`

The `io.fits` subpackage of the core astropy package is one of the most fundamental Python libraries in all of astronomy, just as CFITSIO is one of the most fundamental C packages in astronomy. Much of the reading and writing of FITS files—by far the dominant binary file format used for storing astronomical data—in Python is done with `astropy.io.fits`. The adoption of Python in the astronomy community as a whole and the NASA SMD community in particular could not have happened without a reliable way to read and write FITS files.

Just like FITS as a data format is used in many different ways, `astropy.io.fits` has many different, often competing, requirements. For example, speed and memory efficiency need to be optimized for reading or writing thousands of small FITS files, a large FITS file with thousands of extensions, or a large datacube. At the same time, the FITS standard often allows for different ways to encode the same information (e.g., binary or ASCII tables, scaled or unscaled images) and a large variety of data formats (e.g., integers and floats of different lengths, signed or unsigned) all of which need to be supported for a large range of computer architectures with different endianness and standard float precision. Extreme care must be taken that any code change is fully backwards compatible and does not change the existing behavior. The `astropy.io.fits` subpackage is therefore a challenge to maintain, where even seemingly small changes require deep knowledge of package internals. The end result is that undertaking substantial work on `astropy.io.fits` requires the attention of a dedicated developer; community contributions are insufficient.

The `astropy.io.fits` subpackage has many open issues and most of them require a thorough investigation of possible approaches, knowledge of the relevant FITS standards, and developing a community consensus on the approach taken. Among the open issues, we have identified several key areas that require attention. First, in order to lower the complexity of the code and to ease future maintenance, the package requires a redesign to take advantage of new Python data structures and NumPy functions. Another necessary critical improvement is a rewrite of the FITS verification code. (Approximately 20% of the open issues relate to problems with the current verification approach.) Two other areas of focus are a reworking of the way that scaled data is handled and better integration with the `dask` distributed array package for handling parallel read/write of FITS files where appropriate. This work requires 200 hours per year.

4.2.2. Core Package: Subpackage Maintenance and the Case of `cosmology`

In addition to high-use subpackages like `io.fits`, there are other subpackages with sustainability concerns that, while less fundamental, are still important to the Python ecosystem underpinning astronomy. As a collaborative, volunteer-driven project, a`stropy` subpackages sometimes end up with maintenance concerns as maintainers' focus changes. A prime example of this is the `cosmology` subpackage. It contains a range of functionality that is important for a broad set of astronomy users (e.g., redshift to distance conversions). At the outset of the Astropy Project, cosmology specialists volunteered to help implement this functionality because they enjoyed the prospect of creating the framework within `astropy`. However, those maintainers have left the Project. A recent discussion on the `astropy-dev` mailing list revealed that potential new contributors who might become maintainers are dissuaded by the prospect of learning a new framework without any incentive to do so. Hence, a modest amount of funding supporting some initial work could provide a "foot in the door" that leads to the recruitment of a new maintainer. While we describe a situation specific to `cosmology`, several other subpackages have similar sustainability concerns driven by the shift of Astropy from "new project with lots to build" to "needs ongoing maintenance." Based on the effort that maintaining other subpackages involves, we request 100 hours per year to facilitate the training and recruitment of potential subpackage maintainers.

4.2.3. Coordinated Package: `astroquery`

The `astroquery` package provides programmatic access to a wide variety of online archives. The number of supported archives has grown dramatically in the last year, and the maintenance burden has correspondingly grown. Software teams from other organizations, such as ESA, IRSA, Keck, NOAO, NRAO, STScI, and ALMA have made direct contributions that are seeing widespread use. However, each of these contributions has been accompanied by a major effort on the part of the maintainers to ensure compliance with the `astroquery` user interface, style, and testing guidelines. Additionally, each new feature requires significant time to review and adds to an ever-increasing codebase to be maintained. We are also expecting a slew of incoming new contributing agencies and archives, including the partly NASA-funded Gaia and Herschel missions.

There is room for substantial expansion of `astroquery`'s coverage as more and more archives come online. However, at the same time, as this burden grows, the existing maintainers' career stages and funding sources have changed, affecting their levels of contribution. Hence, we request 80 hours/year of support to recruit and onboard a new maintainer.

4.2.5. Coordinated Package: `ccdproc`

The coordinated package `ccdproc` provides essential functionality for optical/IR image reduction, combination, and management. There are three urgent needs for additional work: substantially improved documentation, a refactoring of the interface for image combination, and additional image combination functionality.

The current documentation is insufficient in a number of ways. It is difficult to navigate to find information for a specific task and lacking coverage for existing functionality (e.g, the functional interface for combining images by breaking the images into memory-size limited chunks). We request 27 hours/year to improve the `ccdproc` documentation.

The object-oriented interface for image combination needs to be refactored in order to make it easier to use in pipelines. Currently, the image combination object is built around a specific list or collection of images. The proposed interface will instead be built around the image combination methods (e.g. combining by average or median) which are callable or have methods for passing in a collection of images. For at least one major version, we will also need to maintain backward compatibility. The estimated effort for a refractor if `ccdproc` is 27 hours/year.

There is still some basic image combination functionality missing in `ccdproc`, primarily for combining science images based on WCS. Though the documentation currently contains instructions for doing this using other affiliated packages, it would make sense to provide some convenience functions in `ccdproc` for the common operations. The estimated effort for this work, including writing tests and documentation for the new functionality, is 13 hours/year.

4.3. Maintaining the Learn Astropy Ecosystem

Within Learn Astropy, there are many tutorials that cover fundamental astronomy research tasks and, together, make use of the nearly the entire Astropy ecosystem. Thus, this tutorial content often provides the first test of research flow functionality as the software is updated and provides excellent integration testing for the project. The users of the tutorials also test the fundamental operation of the Astropy ecosystem for astronomical tasks, and help us identify broken use cases, bugs, and needed patches to affiliated packages.

When the code within an individual Astropy tutorial breaks, the current CI infrastructure brings all other Learn Astropy development to a halt, including changes to non-broken tutorial content. Developing improvements to the CI infrastructure specific to the Learn Astropy content is fundamental to the continuance of the Learn Astropy ecosystem. This infrastructure task requires a skill set that is not typical of astronomers and is ideal for a contractor. We request funds to support 60 hours/year for a contractor to maintain the CI infrastructure for the Learn Astropy tutorials.

4.4. Supporting the Ecosystem of Astropy Affiliated Packages

While the `astropy` core library provides base-level functionality needed to support general scientific tasks, the success and growth of the Astropy Project is sustained by the ever-growing network of interoperable Astropy affiliated packages that build on the core package to add more domain-specific functionality. The Astropy community currently supports over 40 such affiliated packages that enable a wide range of analysis and computational efforts, from observation planning, to Galactic dynamics, to CCD data reduction tasks. The affiliated packages have been impactful because they expand the developer community and enable scientists and domain experts to share code and utilities with more specific goals or context, but with a common level of interoperability and coding standards. For example, a package may make heavy use of the `units` subpackage, meaning that users can expect to receive results as `units.Quantity` or related objects, which share a common and familiar interface.

More than half of the Astropy-affiliated packages are maintained by developers who are also active researchers in their respective scientific fields. A benefit of this structure is that the packages typically support best practices and commonly-needed functionality for the users and scientists who also work in these fields. However, this is also a challenge in that active researchers tend to have less time to invest in Python package maintenance and infrastructure needs, which include setting up and maintaining continuous integration services, installation on multiple platforms, documentation builds, and releases. While the Astropy Project has developed tools to reduce some of the burden of setting up and maintaining Python packages (§§1.1.2 and 4.1), and the vast majority of existing Astropy affiliated packages already make use of the Astropy package template (§4.1.3), keeping a Python package up to date with changes to the package template and landscape of packaging standards currently has to be done individually by each affiliated-package maintainer.

For a majority of the affiliated packages, maintenance is done primarily as volunteer effort. If affiliated-package maintainers change roles or move on to positions with less flexibility, packages can quickly become out-of-sync with community standards or infrastructure package versions, rendering it unusable.

In particular, the `astroplan` package requires maintenance in order to remove the code for Python 2 compatibility and improve the test coverage, particularly of the parts of the code that generate visual output like graphs and finding charts. In addition, the documentation build needs to be fixed and its content expanded to describe more of its functionality. We request 40 hours/year to dedicate to the needs of `astroplan`.

More broadly, maintainers of affiliated packages within the Astropy ecosystem need help adapting their package infrastructure and would benefit greatly from assistance with package releases. In order to sustain the ecosystem of affiliated packages that Astropy has grown into, we request funding to support their releases (1–2 releases per year for half of the affiliated packages, requiring ~2 hours of developer time per release;

60 hours/year). We additionally aim to provide consulting on maintenance issues and upkeep for any Astropy-affiliated package: Not all packages will require monthly assistance, so we estimate that a total of roughly 12 hours/month (140 hours/year) to fund maintenance consultants to sustain this need.

5. Summary

The Astropy Project is one the largest and the most recognized open-source projects in astronomy and space-science. The core package is depended upon by most major missions' analysis and data reduction software pipelines. The Project has grown from a pie-in-the-sky idea among a team of junior scientists, to a project envied by other major projects for its collaborative and extensive ecosystem. While the Project did not require dedicated funding to earn this role in the community, we do need funding to continue to provide the production-level code and educational materials that the community has come to not just expect from us, but to *rely* on us to provide.

As a NumFOCUS project, we recognize the importance of every project that is part of our open source scientific computing community. Though we would like for our work to be funded, we are committed to supporting and collaborating with other NumFOCUS projects that receive funding regardless of our own outcome. We believe that this attitude is crucial for the success of our community and the sustainability of open source projects. It is our hope that this sentiment will be taken into consideration when evaluating our proposal.

References Cited

Astropy Collaboration et al, 2018, Astronomical Journal accepted, arXiv:1801.02634

Astropy Collaboration et al. 2013, Astronomy & Astrophysics, 558, A33

Aldcroft, T., Craig, M., Cruz, K., Corrales, L., Crawford, S., Foster, N., Lim, P., Mumford, S., Price-Whelan, A., Robitaille, T., Shupe, D., Sipocz, B, Tollerud, E. 2020 (APE 0)

Calabretta, M.R., Greisen, E.W., 2002, Astronomy & Astrophysics 395, 1077–1122

Crawford, S., Earl, N., Ginsburg, A., & Tollerud, E. 2017, (APE 13), doi: 10.5281/zenodo.1117943

Cruz, K. L., Hagen, A., Robitaille, T., Tollerud, E., & Villaume, A. 2015, (APE 8), doi: 10.5281/zenodo.1043913

Cruz, K. L., Núñez, A., Burgasser, A. J., et al. 2018, Astronomical Journal, 155, 34

Faherty, J. K., Riedel, A. R., Cruz, K. L., et al. 2016, Astrophysical Journal Supplement, 225, 10

Gaia Collaboration, et al, 2016, Astronomy & Astrophysics, 595, A1

Greenfield, P., 2013, (APE 1) doi:10.5281/zenodo.1043886

Greenfield P., Owen R., Craig M. 2018, (APE 11), https://github.com/astropy/astropy-APEs/pull/14

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C and Oliphant, T. E. 2020, Nature 585, 357 - 362

Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90-95

Jenness, T., Bosch, J., Owen, R., et al. 2016, Proc. SPIE, 9913, 99130G. http://doi.org/10.1117/12.2231313

Lomb, N. R. 1976, Astrophysics and Space Science, 39, 447

Mamajek, E. E., Prsa, A., Torres, G., et al. 2015a, arXiv:1510.07674

Mamajek, E. E., Torres, G., Prsa, A., et al. 2015b, arXiv:1510.06262

McKinney, W., 2010, Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56

Mohr, P. J., Newell, D. B., & Taylor, B. N. 2016, Reviews of Modern Physics, 88, 035009.

Pedregosa, F., et al., 2011, Journal of Machine Learning Research, 12, 2825-2830

Robitaille, T., Greenfield, P., Craig, M. 2014 (APE 7) doi: 10.5281/zenodo.1043907

Robitaille, T., Tollerud, E., Mumford, S., Ginsburg, A. 2018 (APE 14) doi: 10.5281/zenodo.1188875

Scargle, J. D. 1982, Astrophysical Journal, 263, 835

VanderPlas, J. T., Ivezić, Z. 2015, Astrophysical Journal, 812,18

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C J, Polat, I, Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors 2020, Nature Methods, 17, 261 - 272

Zlotnick, F. 2017, GitHub Open Source Survey 2017, http://opensourcesurvey.org,
doi: 10.5281/zenodo.806811

Data Management Plan

Software Sustainability

The Astropy Project's major goal is to build a core software package to enable astronomy and to foster the growth of an ecosystem around it. The core and coordinated packages are therefore the fundamental deliverables of this work. We address their sustainability in two ways: archiving and preserving software releases for the future, and enabling their continued development.

The core `astropy` package and the coordinated packages are licensed under a BSD 3-clause license, which has very permissive terms for code reuse. This license was chosen early in the project in part to encourage use by others without requiring users to share their code in turn, as would be required under a GPL-style license. Other affiliated packages are free to choose any open source license.

The software is developed on GitHub. The Git log provides fine-grained tracking of software changes. Due to the distributed nature of Git, many copies of the code repositories exist as forks; the risk of substantial data loss due to GitHub becoming inaccessible is minimal. Migrating the code to an alternative hosting service (e.g., GitLab) would be straightforward should it become necessary.

The core package has a major release every ~6 months, with a "long-term support" release every 2 years and bugfix releases as needed. Releases are tagged in Git and published to PyPI (the Python Packaging Index), conda, and Zenodo. Coordinated package releases are handled in similar ways. Other Astropy products (e.g., tutorials, website) are also managed on GitHub.

Astropy papers, posters, and other publications are made available under a green open access model, and archived on GitHub and Zenodo. Exceptions may be made on a case-by-case basis and would require written justifications.

Data Products

The Astropy Project in general, and this proposal in particular, does not aim to create stand-alone data products. Some data may be incidentally included in or generated by test suites; where appropriate, this will be stored, version-controlled, distributed, and otherwise managed together with the software.

Dr. Erik J. Tollerud — Biographical Sketch

Contact Information

Space Telescope Science Institute
3700 San Martin Dr
Baltimore, MD 21218, USA
http://www.stsci.edu/~etollerud
ORCID: 0000-0002-9599-310X

Phone (mobile): 651-307-9409
Phone (office): 410-338-6761
Email: etollerud@gmail.com
Github: eteq

Research Positions

Assistant Astronomer, 2017-present
Data Analysis Tools Branch Project Scientist
Space Telescope Science Institute, Baltimore, MD, USA

Giacconi Fellow, 2015-2017
Space Telescope Science Institute, Baltimore, MD, USA

Hubble Fellow, 2012-2015
Yale University, New Haven, CT, USA

Education

University of California, Irvine Irvine, CA, USA.
Ph.D., Physics and Astronomy, June 2012

Advisor: James Bullock

Dissertation: "Local Group Dwarf Galaxies in the LCDM Cosmology: Theory Meets Observations"

University of California, Irvine Irvine, CA, USA
M.S., Physics and Astronomy, June 2008

University of Puget Sound, Tacoma, WA, USA
B.S., Physics, Math, May 2006

Research Interests

Astronomy software development and community management
Dwarf galaxy formation and evolution
Near-field cosmology (Local Group galaxies as a probe of dark matter)
Bayesian Statistics applied to Astronomy

Selected Publications

Mao, Y., ..., Tollerud, E.J., et al. 2020, "The SAGA Survey. II. Building a Statistical Sample of Satellite Systems around Milky Way-like Galaxies", arXiv:2008.12783, submitted to ApJ
Astropy Collaboration, ..., Tollerud, E.J., et al. 2019, "astroquery: An Astronomical Web-Querying Package in Python", AJ, 157, 3
Tollerud, E.J., et al. 2018, "Where Are All of the Gas-bearing Local Dwarf Galaxies? Quantifying Possible Impacts of Reionization", ApJ, 857, 45
Astropy Collaboration, ..., Tollerud, E.J., et al. 2018, "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package", AJ, 156, 3, 123
Tollerud, E.J., et al. 2012, "The SPLASH Survey: Spectroscopy of 15 M31 Dwarf Spheroidal Satellite Galaxies", ApJ, 752, 45
Najita, J., et al. 2017, "Maximizing Science in the Era of LSST: A Community-Based Study of Needed US Capabilities" arXiv:1610.01661

Selected Software

Coordination Committee Member, Astropy Project, 2011-present
Core Maintainer, `astropy.cosmology`, `astropy.utils`
Release Coordinator, `astropy`
Infrastructure Maintainer, Astropy Learn
Owner/Maintainer, `liberfa`
Contributor, Glue-viz
Contributor, Cython

Contributor, Scipy
Contributor, Matplotlib
Contributor, Sunpy
Contributor, Emcee
Additional open source contributions available at `https://github.com/eteq`

SELECTED CONFERENCES AND PRESENTATIONS

Invited Lecture, "Open Source Fundamentals: Open Development", Scipy 2020
Contributed Talk, "Data Analysis Tools for the James Webb Space Telescope: a Confluence of Academia, NASA, and Open Source", Scipy 2019
Invited Lecturer, "Astro Hack Week", Cambridge, UK, 2019
Invited Colloquium, UC Davis, 2018
Invited Astornomy Seminar, UC Irvine, 2018
Invited Lecturer, "LSST Data Science Fellows Program", JHU, Baltimore, MD, 2018
Invited Talk, "Mauna Kea Python Workshop", Hilo, HI, 2017
Invited Talk, "ADASS XXVII", Santiago, Chile, 2017
Workshop Organizer, "Astropy for Planetary Science", 48th DPS/11th EPSC, 2016
Invited Participant, ESO Dwarfs Workshop, Garching, Germany, 2016
Invited Webinar Speaker, LIneA, Rio de Janeiro, Brazil, 2015
Invited Colloquium, Queens University, 2015
Invited Colloquium, Wesleyan Colloquium, 2015
Invited Participant, "Local Group Astrostatistics", University of Michigan, 2015
Invited Talk, "Satellites and Streams in Santiago", Santiago, Chile, 2015
Invited Talk, "Python in Astronomy", Leiden, NL, 2015
Invited Talk, "The Life and Death of Satellite Galaxies", Leiden, NL, 2015
Invited Colloquium, UMass Amherst, 2014
Invited Colloquium, IfA Hawaii, 2013
Invited Colloquium, Stony Brook, NY, 2013

SELECTED AWARDS AND SERVICE ROLES

Winner, 2020 Royal Astronomical Society Group Award (Astropy Project team)
Founding Astropy Coordination Committee Member
OpenAstronomy Steering Council Member
Science PI: STScI Community Software Initiative
LOC Chair: Python in Astronomy 2019
SOC Chair: Large Surveys of the Great Andromeda Galaxy Lorentz Center Workshop (2017)
SOC Member: 9x conferences 2014-present
Grant Review panels: 5x (2014-present)
Hubble Space Telescope proposals awarded: 2x PI, 5x Co-I
Ground-based Telescope proposals awarded: Keck, Palomar, WIYN, Gemini
DELVE Survey Data Processing Coordinator

Additional Biographical Sketches

Dr. Thomas Aldcroft - Following completion of his Ph.D. in Physics in 1993 from Stanford University, Dr. Aldcroft had a postdoc from 1993-1995 at the Harvard-Smithsonian Center for Astrophysics studying X-ray properties of quasars with Dr. Martin Elvis. He then transitioned to the position of Astrophysicist in the Chandra X-ray Center (CXC) in the Operations Science Support team, initially focusing on the spacecraft hardware and ground software responsible for determining spacecraft attitude. In 2006 he was given an additional role as one of three Chandra mission Flight Directors, with broad responsibility for ensuring spacecraft safety during normal operations and directing activities during anomaly response and recovery. As part of his role supporting Chandra science operations, Dr. Aldcroft has (since 2008) led the development of a significant Python-based analysis environment which has become a flight-critical element of Chandra operations. During the early development of these tools, Dr. Aldcroft became involved in the local Python community at CfA. In 2011 he was part of an initial core group that recognized the need for a common Astropy package for the community, and helped organize the first official Astropy coordination meeting which was held at CfA that year. Since that time he has been an active contributor to the project, taking a lead role in development of three major core subpackages: table data structures, time conversions, and ASCII tabular data I/O. In 2016 he was appointed as one of the four Astropy Project Coordination Committee members, with responsibility for overall coordination and management of the Astropy Project.

Dr. Lia Corrales (Ph.D. Astronomy, Columbia University, 2014) — Dr. Corrales worked with Dr. Frits Paerels and was a NASA Earth and Space Science Fellow (2011–2014) during her thesis studies at Columbia University. She developed theory, computational resources, and observational techniques for modeling X-ray scattering halos from dust in the interstellar and intergalactic medium. As a Postdoctoral Associate at MIT Kavli Institute, she worked within the Chandra/HETG group. In addition to studying interstellar dust with high resolution X-ray spectroscopy and imaging, she worked with Frederick Baganoff to study the high resolution spectrum of Sgr A* in its quiescent state. Dr. Corrales held an Einstein Fellowship at the University of Wisconsin-Madison working with Dr. Sebastian Heinz from 2016 to 2018. She is now an Assistant Professor of Astronomy at the University of Michigan. Within the Astropy Project, Dr. Corrales has written several Astropy tutorials and has been the Tutorial Content Lead since 2016. She has served as an organizer and instructor at Astropy Workshops held at Winter Meetings of the American Astronomical Society (AAS) since 2017, and has organized the Astropy/NumFOCUS Exhibit Hall presentations at Winter AAS meetings since 2019. She is also the leader of WoCCode, an Astropy IDE initiative to develop a peer-mentoring network for women of color coders and other demographics that are underrepresented in open source software development.

Dr. Matthew Craig (Ph.D. Physics, UC Berkeley, 1997) - Dr. Craig's thesis was on structure formation in a cold dark matter universe working under Dr. Marc Davis. After completing the degree, and a short postdoc position at Berkeley, Dr. Craig began a tenure-track position at Minnesota State University Moorhead, a small, primarily undergraduate, university at which he is now a full professor. He began an observational astronomy program using the university's 0.4Matt telescope in 20212, and first contributed to astropy in 2013. Since then, he has co-developed the *ccdproc* package, maintained the astropy subpackage for gridded data, and worked to make astropy functionality accessible to introductory-level students through packages like *astrowidgets*, which present an interactive interface to astronomical images in Jupyter notebooks.

Dr. Kelle Cruz - Kelle Cruz is an Associate Professor of Physics and Astronomy at the Hunter College and a Research Associate at the American Museum of Natural History (AMNH). Her research interests include the study of very low mass stars and brown dwarfs. She received both her Bachelors and PhD from the University of Pennsylvania, where she was an NSF Graduate Research Fellow. She was an NSF Astronomy and Astrophysics Postdoctoral Fellow at AMNH and a Spitzer Postdoctoral Fellow at Caltech. She is the founder and Editor-in-Chief of the AstroBetter blog and wiki. She has served on the Coordination Committee of the Astropy Project since 2010. In this role, she has focused on user education via the Learn Astropy initiative and community engagement. She also currently serves on the Finance Committee of the Astropy Project.

Dr. Adam Ginsburg - Adam Ginsburg is Assistant Professor at the University of Florida. He obtained his PhD from the Unversity of Colorado, then was an ESO fellow in Garching and later a Jansky fellow at NRAO Socorro. His research is on high-mass star formation and the formation of the stellar initial mass function. He is co-PI and data reduction lead of the ALMA Large Program ALMA-IMF, which aims to measure the core mass function in the most actively star-forming parts of our Galaxy. He is on the astropy core development team, where he serves as lead developer of the astroquery archive querying tool and on the convolution subpackage, and he is a member of the spectroscopy coordinating committee. Ginsburg leads a vibrant young group at UF, with four graduate students, one postdoc, and several undergraduates involved in the observations and theory of high-mass star formation.

Dr. H. Moritz Günther – Dr. Günther obtained his PhD in physics/astronomy from the University of Hamburg, Germany in 2009. He was a postdoctoral associate at the Harvard-Smithsonian Center for Astrophysics for 5 years and joined the MIT Kavli

Institute in 2015. His PhD thesis about accretion and outflows in classical T Tauri stars is based mostly on X-ray spectroscopy. His research focuses on accreting young stars observed in the X-ray and UV range and he uses astropy in almost all his research projects. At MIT, Dr. Günther is responsible for the Chandra ray-trace code MARX and for ray-trace simulations of future mission proposals. Ray-traces of new missions are done in Python in a software packages called MARXS, which critically depends on astropy. Dr. Günther is one of the early contributors of the open source astropy-project and a co-maintainer for an astropy subpackage and an affiliated package. He is used to software development in a collaborative environment and applying best practices such as version control, continuous integration, and highly informative user documentation. He also serves the Astropy Community as affiliate package editor, responsible for reviewing packages asking to become affiliated with tha astropy community, because they share a common vision of code quality, interoperability, and high standards for software practices. Furthermore, Dr. Günther is a member of the Astropy Finance Committee, which performs financial planning, helps the community set funding priorities, and oversees all financial transactions for the Astropy project.
Some recent publications relevant to this proposal are:
[1] Günther, H. M. & Heilmann, R. "Lynx soft X-ray critical angle transmission spectrometer" 2019, JATIS, Volume 5, id. 021003
[2] Günther, Hans Moritz, Birnstiel, T., Huenemoerder, D. P et al. "Optical Dimming of RW Aur Associated with an Iron-rich Corona and Exceptionally High Absorbing Column Density" 2018, ApJ, 156, 56
[3] Astropy Collaboration, et al. "The Astropy Project: Building an Open-science Project and
Status of the v2.0 Core Package" 2018, AJ, 156, 123
[4] Gunther, H. M., Frost, J., & Theriault-Shay, A. "MARXS: A Modular Software to Ray-trace ¨
X-Ray Instrumentation" 2017, AJ, 154, 243

Pey Lian Lim (M.S. Astronomy, New Mexico State University, 2006) — Lim has been working at the Space Telescope Science Institute (STScI) since 2007, first as an analyst and then moved on to be a software engineer. Her work ranged from performing detector calibration to developing science software for both Hubble Space Telescope (HST) and James Webb Space Telescope (JWST). She has been involved with the Astropy Project from almost the beginning and is currently one of the most active contributors in many areas, in particular related to infrastructure, testing, and software maintenance. She also serves the Astropy Community as affiliate package editor, responsible for reviewing packages asking to become affiliated with the Astropy community, because they share a common vision of code quality, interoperability, and high standards for software practices.

Dr. Stuart Mumford - (Ph.D. Solar Physics, University of Sheffield, 2016) — Dr. Mumford is a research software engineer, primarily working on development of the calibrated data products and Python user tools for the Daniel K. Inouye Solar Telescope (DKIST). He has been involved in the SunPy project since it's 0.1 release in 2011, and with the exception of one year has been in the role of SunPy lead developer since 2012. In this time he has fostered close collaborations with the Astropy project, helping to develop functionality useful to both SunPy and Astropy such as coordinates, visualization and packaging infrastructure. He became an Astropy maintainer in 2019.

Dr. Adrian Price-Whelan (Ph.D. Astronomy, Columbia University, 2016) — Dr. Price-Whelan received his Ph.D. as an NSF graduate research fellow under the advisement of Dr. Kathryn V. Johnston in 2016. While a Ph.D. student, he became involved in early efforts to establish the Astropy project. He then moved to Princeton as a Lyman J. Spitzer, Jr. postdoctoral fellow at the Department of Astrophysical Sciences with mentor David Spergel (2016–2019). During his position at Princeton, Price-Whelan was involved in the cultural startup of the Center for Computational Astrophysics at the Flatiron Institute (Simons Foundation), where he is currently a Flatiron Research Fellow (2019–present). Throughout, Price-Whelan has continued to contribute to Astropy as a maintainter of the coordinates and units subpackages, as well as the lead maintainer of the Learn Astropy initiative, all while performing cutting-edge research in Galactic dynamics, Milky Way structure, and binary star science. In addition to Astropy, Price-Whelan maintains the astropy-affiliated package *gala*, for gravitational dynamics and orbit computation, along with open-source Python packages for binary-star science (*thejoker*) and for working with data from the Gaia mission (*pyia*).

John D. Swinbank (DPhil Astrophysics, University of Oxford, 2008) — Following a doctorate in high-redshift optical astronomy as the University of Oxford under the supervision of Drs Joanne Baker and Isobel Hook, Swinbank took up a role as a postdoctoral research assistant with the LOFAR Transients Key Science Project at the University of Amsterdam. During this period, he participated in commissioning and system integration of the LOFAR radio telescope, with a particular emphasis on the development of pipelines for detecting transient and variable sources. From 2010, he served as Software Project Manager for the AARTFAAC radio all-sky monitor at the University of Amsterdam. In 2014, he joined the Large Synoptic Survey Telescope (now Vera C. Rubin Observatory) team at Princeton University. As Technical Manager for LSST Data Release Production, he grew the team to around 15 people focused on developing the software pipelines which will be used to generate annual data releases based on the Rubin Observatory's survey. In 2017, he was appointed Research Professor at the University of Washington and Deputy Project Manager for the Rubin

Observatory Data Management Subsystem, working under Project Manager Dr William O’Mullane. In this role, he was responsible for supervising the work of around 100 people (60 FTEs) contributing to all aspects — software and hardware — of the observatory’s data processing system. He briefly served as Deputy Acting Associate Director for Rubin Observatory Operations. In November 2020, he returned to Europe, and is now Science Data Centre Program Manager at ASTRON, the Netherlands Institute for Radio Astronomy. Since late 2019, he has been a member of the Astropy Project’s Interim Finance Committee. He has extensive experience of all aspects of the software lifecycle, of managing large, distributed software development teams, and of financial management of budgets in excess of $100M.

Table of Personnel and Work Effort

Person and/or Role	Time Charged to this Proposal	Time Not Charged to this Proposal	Total Time per person/year
PI, Erik Tollerud	N/A	1 months/year	1 months/year
Co-I, Matt Craig	N/A	2 months/year	2 months/year
Co-I, Tom Aldcroft	N/A	0.2 months/year	0.2 months/year
Co-I, Lia Corrales	N/A	0.2 months/year	0.2 months/year
Co-I, Kelle Cruz	N/A	0.2 months/year	0.2 months/year
Co-I, Adam Ginsburg	N/A	0.2 months/year	0.2 months/year
Co-I, Hans Guenther	N/A	0.2 months/year	0.2 months/year
Co-I, Pey Lian Lim	N/A	0.2 months/year	0.2 months/year
Co-I, Adrian Price-Whelan	N/A	0.2 months/year	0.2 months/year
Collaborator, Stuart Mumford	N/A	*de minimus*	*de minimus*
Collaborator, John Swinbank	N/A	*de minimus*	*de minimus*
Unspecified open source contractor #1	2 months/year	N/A	2 months/year
Unspecified open source contractor #2	2 months/year	N/A	2 months/year
Unspecified open source contractor #3	2 months/year	N/A	2 months/year

Current and Pending Support

Erik Tollerud (PI)

- “COS-SAGA: The Circumgalactic Medium of Nearby Milky Way Analogs and their Satellites”. PI: Erik Tollerud (STScI), Hubble Space Telescope GO-15826.001-A, Space Telescope Grants Administration, gms_mail@stsci.edu. $198,790. Effort: 25%.
- “What drives the evolution of Luminous Compact Blue Galaxies in Clusters vs. the Field?”, Science PI: Gregory Wirth, Admin PI: Erik Tollerud (STScI), Hubble Space Telescope HST-AR-15058.007-A, Space Telescope Grants Administration, gms_mail@stsci.edu. 10/1/20 - 8/30/21. $71,303. Effort: 4%.
- “Sustaining and Growing the Astropy Project”. Science PI: Thomas Robitaille, Admin PI: Erik Tollerud (NumFOCUS). Gordon and Betty Moore Foundation DDD Grant, Robert Kirshner, robert.kirshner@moore.org. 10/1/19 - 9/30/22. $0 to Tollerud, $900k distributed across Astropy Project. Effort: 5%

Matt Craig (Co-I, Effort > 10%)

- “Sustaining and Growing the Astropy Project”. Science PI: Thomas Robitaille, admin PI: Erik Tollerud. Gordon and Betty Moore Foundation DDD Grant, Robert Kirshner, robert.kirshner@moore.org. $30k subaward to Co-I Craig. 5/1/20 - 4/30/21. Effort: 10%

Budget Justification

Basis of estimates

All direct costs in this proposal are for contractor expenses. In each instance the estimate of hours of effort is based on experience with similar tasks in the Astropy Project.

Budget summary

The following table provides a summary of the contractor budget (with only labor hours, as per salary redaction rules):

	Year 1	Year 2	Year 3
Area of support	Hours	Hours	Hours
Project-wide Infrastructure & core maintainers	420	495	870
Core and coordinated packages	447	447	447
Learn Astropy Infrastructure	60	60	60
Affiliated packages maintenance	180	180	180
Total	**1,107**	**1,182**	**1,557**

Year 1

The total effort for year 1 of the proposal is 1,107 hours.

- Infrastructure (420 hours)
    - Continuous integration (200 hours)
        - (100 hours) Provide consistent support to resolve ongoing issues as they arise.
        - (60 hours) Reduce runtime of tests, update automation for affiliated packages.
        - (40 hours) Rewrite developer documentation for continuous integration
    - Package template (150 hours)
        - (100 hours) Improve documentation of the template and close several outstanding issues reported by users of the template.
        - (50 hours) Bring older astropy affiliated packages up to date with the current package template and work on automating future updates.
    - Revision and maintenance of contributing guide (70 hours)

        - (50 hours) Rewrite of the current guide to contributing to astropy and affiliated packages. The current guide is very dated and needs extensive revision.
        - (20 hours) Update or add references to updated contributing guides in documentation for affiliated packages.
- Core and Coordinated Packages (447 hours)
    - Core astropy FITS subpackage (200 hours)
        - (160 hours) Begin rewriting the FITS verification part of the codebase.
        - (40 hours) Close open FITS bugs that are not related to verification. One bug takes roughly 10 hours to fix.
    - Other core astropy subpackages (100 hours)
        - (100 hours) The cosmology subpackage is one of a few subpackages without a dedicated maintainer. A minimum of 40 hours per year is required to maintain a mature subpackage. Turnover in maintainers has been high enough that we anticipate additional needs by the first year of the proposal.
    - Ccdproc (67 hours)
        - (30 hours) Extensive revision of package documentation.
        - (30 hours) Initial refactor of class-based interface for image combination.
        - (7 hours) Write a convenience function to combine science images with WCS reprojection, using other affiliated packages to handle the reprojection.
    - Astroquery (80 hours)
        - (40 hours) Increase test coverage for existing code. This level of effort should be sufficient to add coverage to two existing astroquery services.
        - (40 hours) Review and support contributions from several new data providers anticipated during this period.
- Learn Astropy support (60 hours)
    - (20 hours) Attend Learn Astropy telecons so as to understand current needs and emerging issues.
    - (40 hours) Maintain and improve continuous integration infrastructure and update tutorials as issues arise.
- Support for affiliated packages (180 hours)
    - (40 hours) Affiliated package modernization, including re-enabling regular documentation build.
    - (140 hours) Assistance with regular updates on affiliated package infrastructure each time a package has a release. This estimate is based on past release frequency, the number of affiliated packages and the time typically required to troubleshoot issues.

Year 2

The total effort for year 2 of the proposal is 1,182 hours.

Much of the justification is the same as Year 1. **Items which are different in Year 2 are in bold below.**

- Infrastructure **(495 hours)**
  - Continuous integration (200 hours)
    - (100 hours) Provide consistent support to resolve ongoing issues as they arise.
    - (60 hours) Reduce runtime of tests, update automation for affiliated packages.
    - (40 hours) Rewrite developer documentation for continuous integration
  - Package template (150 hours)
    - **(50 hours)** Improve documentation of the template and close several outstanding issues reported by users of the template.
    - **(100 hours) Work on automating future updates to packages that use the template.**
  - Revision and maintenance of contributing guide **(20 hours)**
    - **(20 hours) Maintain and update contributing guide.**
  - **Support for contributors in maintainer roles and coordinating committee (125 hours)**
    - **(125 hours) Partial year of support for work currently funded by Moore grant.**
- Core and coordinated packages (447 hours)
  - Core astropy FITS subpackage (200 hours)
    - **(40 hours) Finish remaining work on rewriting the FITS verification part of the codebase.** Estimate provided by current and former maintainers of the FITS subpackage.
    - **(80 hours)** Close open FITS bugs that are not related to verification. One bug takes roughly 10 hours to fix.
    - **(80 hours) Rewrite handling of scaled data. Estimate based on experience of prior maintainers.**
  - Other core astropy subpackages (100 hours)
    - (100 hours) A minimum of 40 hours per year is required to maintain a mature subpackage. Turnover in maintainers has been high enough that we anticipate additional needs by the first year of the proposal.
  - Ccdproc (67 hours)
    - **(10 hours) Maintenance of package documentation.**
    - **(40 hours) Further effort to revise the new class-based interface and begin removing support for the old interface.**
    - **(17 hours) Write a convenience function to align images based on stars using an existing affiliated package to do the alignment.**
  - Astroquery (80 hours)

        - (40 hours) Increase test coverage for existing code. This level of effort should be sufficient to add coverage to two existing astroquery services.
        - (40 hours) Review and support contributions from several new data providers anticipated during this period.
- Learn Astropy support (60 hours)
    - (20 hours) Attend every other Learn Astropy telecon so as to understand current needs and emerging issues.
    - (40 hours) Maintain and improve continuous integration infrastructure and update tutorials as issues arise.
- Support for affiliated packages (180 hours)
    - **(180 hours)** Assistance with regular updates on affiliated package infrastructure each time a package has a release. This estimate is based on past release frequency, the number of affiliated packages and the time typically required to troubleshoot issues, **and assume that the number of affiliated packages will have increased by this year.**

Year 3

The total effort for year 3 of the proposal is 1,557 hours.
Much of the justification is the same as Year 2. **Items which are different in Year 3 than in Year 2 are in bold below.**

- Infrastructure **(870 hours)**
    - Continuous integration (200 hours)
        - (100 hours) Provide consistent support to resolve ongoing issues as they arise.
        - (60 hours) Reduce runtime of tests, update automation for affiliated packages. Estimate based on past experience.
        - (40 hours) Rewrite developer documentation for continuous integration
    - Package template (150 hours)
        - (50 hours) Improve documentation of the template and close several outstanding issues reported by users of the template.
        - (100 hours) Work on automating future updates to packages that use the template.
    - Revision and maintenance of contributing guide (20 hours)
        - (20 hours) Maintain and update contributing guide.
    - **(500 hours)** Support for contributors in maintainer roles and coordinating committee. The Moore grants ends prior to year 3 of this proposal
- Core and coordinated packages (447 hours)
    - Core astropy FITS subpackage (200 hours)
        - **(120 hours)** Close open FITS bugs that are not related to verification. Estimate is that one bug will take roughly 10 hours.

        - **(80 hours) Better integrate with the dask package for distributed computing.**
    - Other core astropy subpackages (100 hours)
        - (100 hours) A minimum of 40 hours per year is required to maintain a mature subpackage. Turnover in maintainers has been high enough that we anticipate additional needs by the first year of the proposal.
    - Ccdproc (67 hours)
        - **(20 hours) Maintenance of package documentation.**
        - **(47 hours) Further effort to revise the new class-based interface and begin removing support for the old interface.**
    - Astroquery (80 hours)
        - (40 hours) Increase test coverage for existing code. This level of effort should be sufficient to add coverage to two existing astroquery services.
        - (40 hours) Review and support contributions from several new data providers anticipated during this period.
- Learn Astropy support (60 hours)
    - (20 hours) Attend every other Learn Astropy telecon so as to understand current needs and emerging issues.
    - (40 hours) Maintain and improve continuous integration infrastructure and update tutorials as issues arise.
- Support for affiliated packages (180 hours)
    - (180 hours) Assistance with regular updates on affiliated package infrastructure each time a package has a release. This estimate is based on past release frequency, the number of affiliated packages and the time typically required to troubleshoot issues, and assume that the number of affiliated packages will have increased by this year.

Facilities and Equipment

No specific facilities are to be used, and proposal personnel furnish their own computers for their preferred software development environment.

Detailed Budget

The total cost of the open source contractors provided by NumFOCUS is provided in the NSPIRES cover page budget in Section F line 3. Those costs are all labor and thus not shown here, consistent with ROSES rules regarding redaction.

“Sustaining the Astropy Project” Total Budget

The budget below is for the only budget line item in this proposal, consultant costs. It assumes a pay rate of $150/hour. NumFOCUS does not currently have a negotiated indirect cost rate with any cognizant agency therefore indirect costs are calculated based on the de minimis rate.

Budget Details

Budget Period:			Year 1			**Year 2**			**Year 3**	Total
Start Date:			7/19/2021			7/19/2022			7/19/2023	
End Date:			7/18/2022			7/18/2023			7/18/2024	
Personnel										
Key Personnel										
PI			$0.00			$0.00			$0.00	$0.00
Co-I			$0.00			$0.00			$0.00	$0.00
Total Salary			**$0.00**			**$0.00**			**$0.00**	**$0.00**
Other Direct Costs										
Consultant Services	**Rate**	**Hours**		**Rate**	**Hours**		**Rate**	**Hours**		
Infrastructure: continuous integration	$150	200.00	$30,000	$150	200.00	$30,000	$150	200.00	$30,000	$90,000
Infrastructure: package template	$150	150.00	$22,500	$150	150.00	$22,500	$150	150.00	$22,500	$67,500

Infrastructure: Contributing guide	$150	70.00	$10,500	$150	20.00	$3,000	$150	20.00	$3,000	$16,500
Infrastructure: Maintainers	$150	0.00	$0	$150	125.00	$18,750	$150	500.00	$75,000	$93,750
Core/coordinated: cosmology	$150	100.00	$15,000	$150	100.00	$15,000	$150	100.00	$15,000	$45,000
Core/coordinated: io.fits	$150	200.00	$30,000	$150	200.00	$30,000	$150	200.00	$30,000	$90,000
Core/coordinated: ccdproc	$150	67.00	$10,050	$150	67.00	$10,050	$150	67.00	$10,050	$30,150
Core/coordinated: astroquery	$150	80.00	$12,000	$150	80.00	$12,000	$150	80.00	$12,000	$36,000
Affiliated package support:	$150	180.00	$27,000	$150	180.00	$27,000	$150	180.00	$27,000	$81,000
Education and Docs	$150	60.00	$9,000	$150	60.00	$9,000	$150	60.00	$9,000	$27,000
Total Other Direct Costs			**$166,050**			**$177,300**			**$233,550**	**$576,900**
Total Direct Costs			**$166,050**			**$177,300**			**$233,550**	**$576,900**
Modified Total Direct Costs			$166,050			$177,300			$233,550	$576,900
Indirect Costs										
NumFOCUS De Minimis Rate			**$16,605**			**$17,730**			**$23,355**	**$57,690**
Budget Total			**$182,655**			**$195,030**			**$256,905**	**$634,590**