Project proposal for `SciPy`

Title: User-oriented documentation and thorough restructuring

Author: Maja Gwóźdź

Contents: Motivation | Goals | User survey | Graphical guides | About me

Motivation

I intend to work on the refactoring of the existing documentation, so that it would be **easily accessible** by users with different needs. It goes without saying that a researcher is most likely interested in advanced and subtle features, whereas a user without prior expertise appreciates step-by-step guides and **diagrams**.

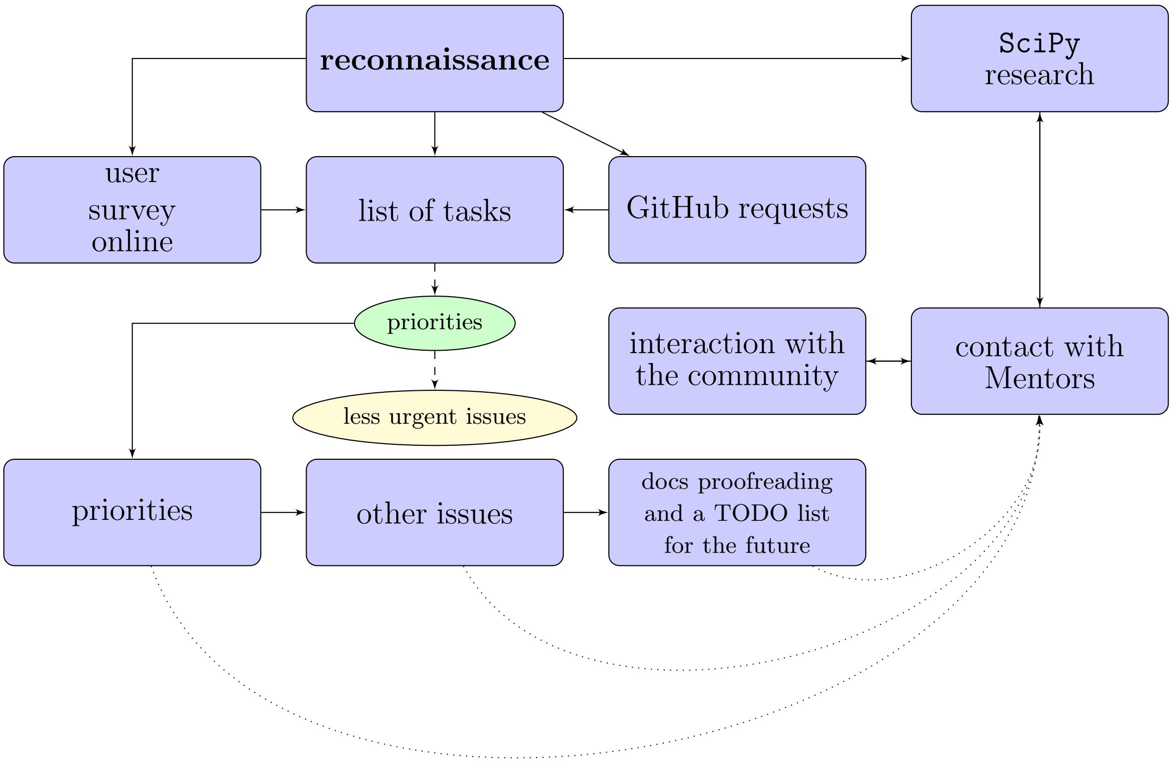
I am interested in pursuing this project for personal and professional reasons: first of all, I would like to contribute significantly to `SciPy` because my own research has greatly benefited from it and secondly, I encounter insufficient (or lacking) documentation all too often in other software and always wonder how much faster (if it all!) users could learn how to use the code had they been provided with a thorough guide.

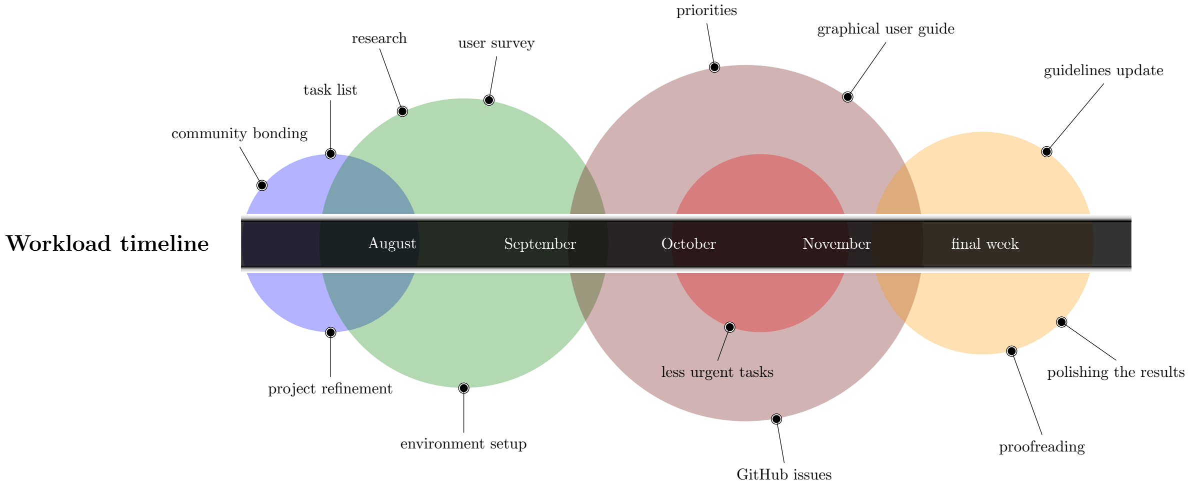
Goals

I aim to improve the existing `SciPy` documentation both content- and graphic-wise. The most important feature of my approach to this problem is the deployment and analysis of the **user survey**, that is to say, a concise survey conducted online enabling various users to voice their needs regarding the documentation. I strongly believe that their opinions should be the source of inspiration (how else can we create a more user-friendly documentation?).

As regards the realisation of the project itself, the **first phase** will involve designing and analysing the user survey, as well as tackling several **stylistic** issues I have noticed in the current documentation. For instance, lack of consistency (example: *2-dimensional arrays* occurring alongside *two-dimensional arrays*), convoluted sentences that ought to be rewritten, or the lack of alphabetical order in certain subpages. The **second phase** will focus on the introduction of **graphical** guides containing hyperlinks to the relevant topics (based on the survey results and other community requests). In the long run, I wish to achieve a satisfactory documentation tailored to different kinds of users. Moreover, I will attempt to render the tutorials more consistent both **linguistically** and **structurally**. Last but not least, I aim to write new tutorials (based on the current community needs).

The workflow and a tentative timeline are presented in the diagrams below, while the detailed descriptions of the user survey and graphical guides are included in the following sections.




Workload timeline

User survey

As regards the user survey, I propose to use Google Forms for several reasons. First of all, Google Forms is free and offers unlimited functionality (in terms of the number of respondents, questions, etc.), it has an appealing visual form, the most useful survey options (for instance, the customisable linear scale, checkboxes, and multiple choice), and, most importantly, the results can be easily exported for the purposes of statistical analysis. Based on online research, it appears that Google Forms is, at least for now, the best free tool for conducting surveys. On a less serious note, it would be a nice gesture to use a Google product in a Google-run initiative ☺.

I have created a preliminary survey with sample questions (it can be accessed here). A reasonable number of questions in the final version ought to be between ten and fifteen. In order to obtain concrete results, I suggest that we predominantly use multiple choice questions, a linear scale, and a few checkboxes. The linear scale should not resemble a full spectrum, though (it only causes confusion and the results are likely to suffer from high dispersion). There ought to be at maximum two open-ended questions, otherwise the results will be highly dispersed and not helpful at all. I reckon that even a very high number of responses would not be problematic due to the fact that the data can be easily exported and analysed automatically with statistical software. Assuming that the number of responses is, indeed, very high, the analysis of open-ended questions could be a little time-consuming but I presume that it will not be overwhelming. After all, an average user is not likely to write an essay about the state of the documentation. In the worst case scenario, some answers can be simply stored for future analysis.

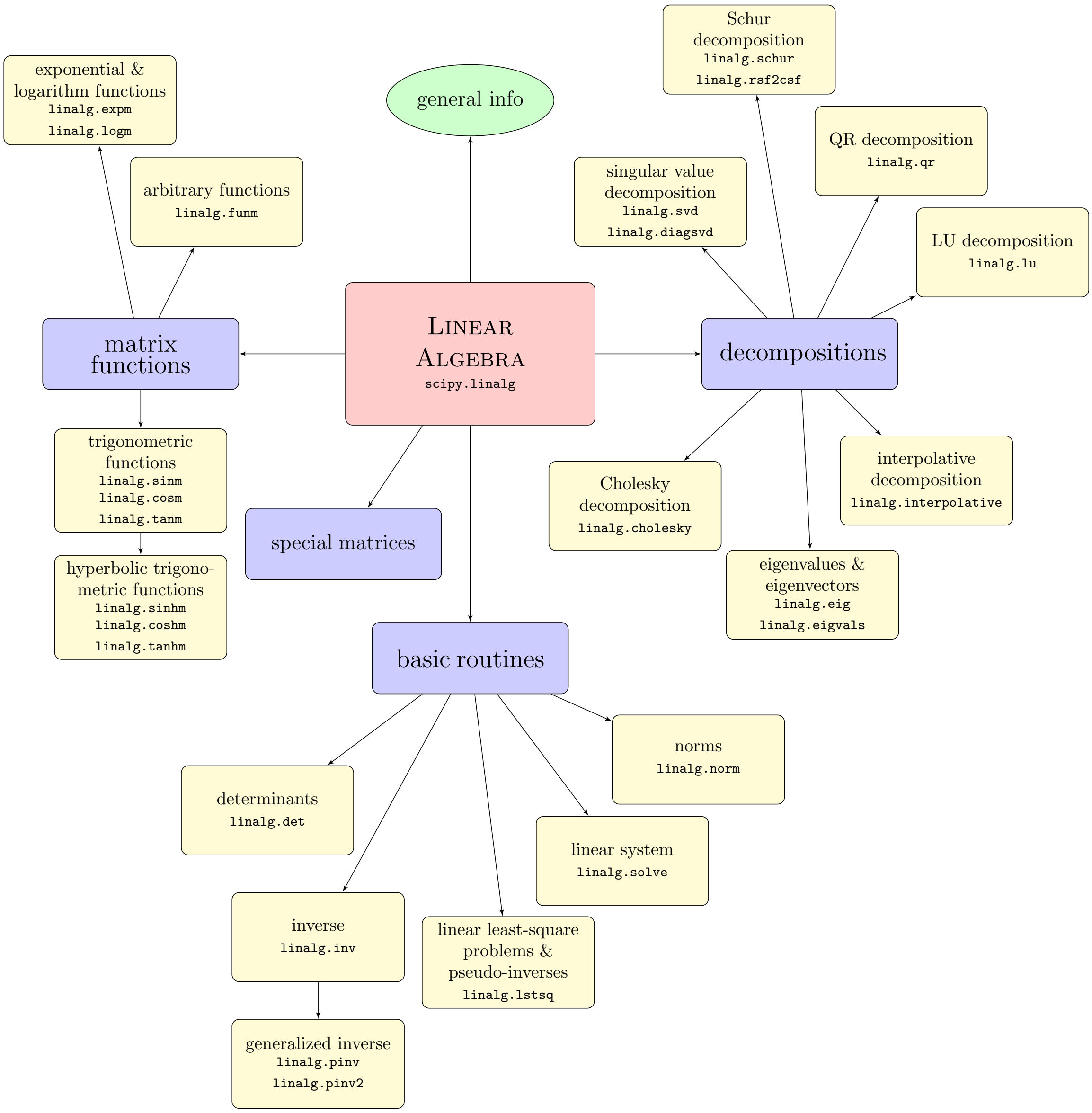
Graphical guides

My vision of the graphical guides (intended to serve as navigational tools) is based on a popular premise that (most) humans are better at processing straightforward visual structures rather than purely text-based information. Moreover, a thematically-oriented diagram with lines connecting similar topics of interest is, undoubtedly, a highly valuable asset for less experienced users (and not only).

As regards the implementation details, I propose to use the **TikZ package**. First and foremost, it is a powerful tool and does not seem to be at risk of being deprecated soon. It also offers high-quality output, has a really solid documentation, and is a frequent topic on TeX StackExchange and other mainstream forums. Most importantly, the integration of a TikZ file (more precisely, the numerous hyperlinks therein) with HTML documentation does not appear to pose significant problems due to the existence of various packages and fixes for embedding a TikZ picture in HTML (for instance, `TeX4ht`).

The question of future maintenance of the guides within `SciPy` can be easily solved by using, say, Overleaf (facilitates collaboration plus offers an instant preview) and predefined templates that I will supply. Basically, the graphical guides are not likely to differ hugely from one another. The structure, colour palette, and shapes are, more or less, going to be invariant, therefore subsequent re-shaping and further customisation will not be an issue. A rough sketch of such a guide (observe the counter-clockwise alphabetical order in the subcategories) is provided on the next page[1]. The complete diagram will, of course, contain hyperlinks to the respective sections in the documentation.

[1] Data based on: `https://docs.scipy.org/doc/scipy/reference/tutorial/linalg.html`.



About me

I completed a BA in **English Studies** with distinction (Jagiellonian University, Poland) and then obtained an MPhil in **Theoretical and Applied Linguistics** with distinction from the University of Cambridge. I then decided to pursue a BSc in **Computer Science** (at the Ludwig Maximilian University in Munich) and take additional courses in mathematics (so far, I have completed the following extension courses either from UC Berkeley or the University of Illinois: Elementary Number Theory, Calculus II, Precalculus, Python Programming). Other relevant technical courses I have taken so far are: Real Analysis, Linear Algebra, Introduction to Programming, Algorithms and Datastructures, Discrete Mathematics and Logic, Introduction to Functional Programming, Introduction to Artificial Intelligence, Logic, Computer Architecture. As regards machine learning, I have a working knowledge of statistics, the Multilayer Perceptron Classifier (especially its application to automatic speech recognition), and other popular Artificial Neural Networks.

While I am not a technical writer in the strictly professional sense of the word, I am familiar with Sphinx and I have performed the tasks of a technical writer on several occasions. For instance, I completed an internship at **Lufthansa CityLine**, where I was responsible for running penetration tests and writing a technical report on network vulnerabilities. I was also responsible for designing a JIRA / Confluence workflow and preparing a basic guide for internal users. I was a student at **GSoC** 2018 (it was a project on corpus linguistics involving, among other tasks, the creation of annotation guidelines) and I am currently a mentor at the same organisation (CLiPS, the University of Antwerp).

I am passionate about clear and logical communication of technical matters and I believe that this project suits my background perfectly because I have the required linguistic tools to convey complex ideas plus the necessary mathematical / computer science knowledge to comprehend the subject (or, at least, know how to ask the right questions about the given matter).

I can offer total commitment to the project. During the summer / autumn I will be able to devote forty hours per week to the project. I pay great attention to detail but, at the same time, try not to lose sight of the big picture. Whenever I notice that I spend too much time on a less urgent task, I quickly move on to the important phases, so as to meet the deadline (time permitting, I take care of the less urgent issues, of course). Getting stuck is the natural part of any creative process and it is, indeed, valuable but if it becomes a true obstacle I never hesitate to ask for help. This approach has worked very well in my previous projects and I intend to apply it to subsequent endeavours. In the interaction with supervisors and team colleagues, I particularly like constructive criticism and frequent feedback. While support and positive comments are undoubtedly important, I have never made significant progress based on praise alone. I enjoy challenging tasks and, as regards my approach to solving real-life software problems, I believe that actively listening to community members and global users is THE way to create excellent software. It would be an honour to work on `SciPy`.

Acknowledgments: I am grateful to Ralf Gommers for all useful suggestions and reading the drafts.