MATLAB vs. PYTHON from a MATLAbers Perspective

M.H. Trauth, 14. März 2022 10:47 Uhr

	MATLAB	PYTHON
Origins	The first version of MATLAB was an interactive matrix calculator, written in Fortran by Cleve Molar in the late 1970s. Together with Jack Little and Steve Bangert, he decided to make it a commercial product, founded MathWorks and released the first reversion of MATLAB written in C in 1984 (Source).	PYTHON was written by Guido van Rossum in the late 1980s and released as PYTHON 0.9.0 in 1991, PYTHON 2.0 in 2000 and PYTHON 3.0, not completely backward compatible, released in 2008. PYTHON 2 was discontinued in 2020 (Source).
Developer	MATLAB is the product of MathWorks Inc., which is a company with 5,000+ employees and a 1.05 billion USD revenue (2019). (Source). MATLAB, with 5 million+ users, is one of several products of the company, which also includes SIMULINK for simulation and model-based design of multidomain and embedded engineering systems (Source).	PYTHON is promoted, protected and advanced by the PYTHON Software Foundation (PSF), a nonprofit organization with a 4.503 million USD revenue (2019) (Source). There are different levels of membership, started from basic members supporting the PSF without a financial contribution, to supportive members, managing members, contributing members and fellows.
Business Model	MathWorks offers different license models for standard, education, home and student use of the software. The license includes the software, documentation and support with a 24-hour response time (Source).	PYTHON is free and comes with an online documentation, but without support (Source). However, there are numerous companies providing commercial services, consulting, trainings and tools with/for PYTHON (Source).
Features	MATLAB is both a programming language and a desktop programming environment, which includes an editor, a debugger, workspace browser, sophisticated graphics and a compiler for stand-alone applications. It can be run in a text terminal with an external editor, in the graphical user interface called MATLAB Desktop, as well as in a browser using MATLAB Online in a cloud environment. A mobile client called MATLAB Mobile exists to run it remotely on mobile devices such as smartphones and tablet computers.	PYTHON is used in a text terminal. However, Spyder, as an example, provides a development environment similar to MATLAB Desktop with an Editor, workspace browser, console (equivalent to the Command Window of MATLAB) and graphics. It can be installed together with PYTHON, or, which is recommended, as part of the Anaconda Distribution, which helps installing and managing the most important packages (see Setup) for PYTHON. Jupyter Notebook is a web-based interactive computing environment.
Setup	MATLAB is modular, with >90 toolboxes from MathWorks (Source) and a large number of user-provided toolboxes, partly commercial, others available for free available from the MathWorks File Exchange (Source). The File Exchange currently lists 42,967 user contributed scripts, functions and toolboxes as of March 2022, including 296 files contributed by MathWorks.	PYTHON is modular, with a large number of packages provided by independently operating teams, available for download from individual webpages. A list of PYTHON packages is available online on the Python Package Index. This index currently lists ~5.7M user contributed files from ~578k users as of March 2022.

Open Source	MATLAB is not open source. However, most MATLAB algorithms are open; others, so-called builtin functions, are not open but the Support provides all necessary information about the algorithm. Many of these, e.g. the FFT algorithm, which are not open, are standardized algorithms and can easily be tested for possible errors.	PYTHON is open source, i.e. the source code of PYTHON is fully open. The question is, of course, who, for example, actually looks up, understands and checks the code of the FFT in the source code before using it, remembering that there is nobody to ask about it.
Backwards Compatibility	MATLAB is is backwards compatible to the highest degree, as own tests with ~30 years old code show. For the adaptation and optimization of older code, software tools are provided or support helps.	The biggest challenge for long-time developers of PYTHON code was the transition from PYTHON 2 (2000–2020) to PYTHON 3 (since 2008). Since there is no support and you have to rely largely on searching solutions on the Internet (Stackoverflow, Google), you will need a lot of time filtering or translating PYTHON 2 solutions found on the Internet (Source).
Certification	MathWorks provides a IEC certification kit for ISO 26262 and IEC 61508.	PYTHON / the PSF does not offer certification of code.
Installation	MATLAB comes with an installer that you can use to install MATLAB together with the required toolboxes. Adding new toolboxes is achieved by simply re-running the installer. Toolboxes not provided by MathWorks can be installed by adding them to the search path of MATLAB or by running (if available) the installer of the toolbox (Source).	PYTHON comes with an installer for the command-line version of the software. Installing it with Spyder you get a minimum number of packages. Installing it with the Anaconda distribution, it comes with a large number of toolboxes. There is a free individual edition, but also commerical editions starting at $14.95/month.
Reproducibility	Clearly reproducible code as R2021b clearly says which version of the the ~90 preinstalled toolboxes are used. However, if you use toolboxes that are not offered by MathWorks, you get into an area that is very close to the PYTHON ecosystem.	PYTHON comes without packages or toolboxes. The installation of packages can lead to an environment which is hardly reproducible. However, using a distribution such as Anaconda and creating so-called environments helps to reproduce the setup used in a project.

Use without the GUI	You can run MATLAB as a command-line using
	`./matlab -nodesktop`
	which includes graphics, or
	`./matlab -nojvm`
	without graphics.
		PYTHON is a command-line tool by default (Source). You can run PYTHON by typing
		`python`
		into the text terminal or console.

Using it with a GUI	MATLAB comes with a graphical user interface called MATLAB Desktop.	PYTHON does not have a graphical user interface. However, you can run PYTHON within GUIs such as Spyder (Source).

Graphical Editor	MATLAB Live Editor to create scripts that combine code, output and formatted text, images, hyperlinks and equations created with an interactive WYSIWYG editor (Source).	Jupyter Notebook is a web-based interactive computing environment for PYTHON (Source).
Interactivity	Interaction with graphics simple and compatible across platforms.	Interactive code is platform and distribution dependent.
Online Computing	MATLAB Online (Source), MATLAB Drive (Source) and MATLAB Drive Connector (Source).	Jupyter Notebook is a web-based interactive computing environment for PYTHON (Source).
Self-paced Courses	MATLAB Academy (Source).	There is no professional elearning platform from the PSF but free and commerical alternatives exist.
Documentation	Complete and up-to-date documentation with simple examples, available online and local on a computer (Source). The documentation of MATLAB was quite confusing in the 2000s, when MATLAB grew very fast and many new toolboxes were developed, partly bought. This was also the time of the transition in graphics, which was completed around 2014. Since then, a lot of work has been done on the documentation, it has been unified, there are many cross-references and easy to understand examples.	Python documentation up to date (Source) but the documentation of packages of very mixed quality. The problem is that most users learn from examples rather than working through a cryptic explanation of the functions with their input and output arguments. The examples in the documentation are often far too complicated and specific for this, which is why users quickly start googling for solutions and in turn end up with specific, sometimes incorrect or outdated advice from other users, which also depends on specific environments and therefore often does not work.
Support	Professional support with 24 hour response time. This works great, with contact first from national offices of MathWorks, who forward the problem to headquarters if they can't solve it. In this way, I often came into contact with the actual developer of the particular algorithm.	None. Either use Google for help, browse free online resources provided by the PSF and other services such as Stackoverflow. Commercial solutions are provided by companies such as Stackoverflow Inc. ($135M revenue, 575 employees, source) and Anaconda Inc. ($11.61M revenue, 125 employees, source).
Book Program	MathWorks runs a Book Program for authors of MATLAB-based books (Source). The Book Program provides authors with free book development licenses and extra support.	The PSF says that they are not interested to get involved in the production of PYTHON-based books and they do not provide any kind of support (Source).
Books	MathWorks lists 2,000+ MATLAB-based books on their website (Source).	The PSF lists many PYTHON-based books on its website (Source).
Developer Conference	MATLAB Expo, in the US and elsewhere (Source).	PyCon Conferences (Source).

Syntax, Arrays	Very compact syntax, e.g.
	`A = [2 4 3 7; 9 3 -1 2; 1 9 3 7; 6 6 3 -2]`
		Very long syntax with lots of different brackets, e.g.
		`A = np.array([[2,4,3,7],[9,3,-1,2],[1,9,3,7],[6,6,3,-2]])`
		`print(A)`

Syntax, uint8	No rolling. `uint8(-3) = 0`	Rolling. `np.uint8(-3) = 253`
Syntax, Indexing	Indexing starts with (1,1). This has been widely criticized for having no (0,0) origin, but the advantages outweigh the disadvantages when working with arrays in science and engineering.	Indexing starts with (0,0) but also negative indices possible. This certainly has advantages in some applications, but in many others – especially in science and engineering – it leads to unexpected behavior and possible errors.

Syntax, Series	Get integers [0,10]
	c = 0 : 10
		Intervals open to the right are confusing, so this isn't [0,10] but [0,10[
		`c = np.arange(0,10,1)`
		`print(c)`
		`[0 1 2 3 4 5 6 7 8 9]`
		and you need to write
		`c = np.arange(0,10+1,1)`
		`print(c)`
		`[ 0  1  2  3  4  5  6  7  8  9 10]`
		to get the 10 included.

Syntax, Access	As expected:
	`c = 0 : 10`
	`c(1)`
	`c(2)`
	`c(1:2)`
	`ans =`
	`     0`
	`ans =`
	`     1`
	`ans =`
	`     0     1`
		Unexpected behaviour in the use of indices:
		`c = c = np.arange(0,10+1,1)`
		`print(c[0])`
		`print(c[0:1])`
		`print(c[0:2])`
		`0`
		`[0]`
		`[0 1]`

	Transposition:
	`c = 0 : 10;`
	`c'`
	`ans =`
	`0`
	`1`
	`2`
	`3`
	`4`
	`5`
	`6`
	`7`
	`8`
	`9`
	`10`
		The array
		`c = c = np.arange(0,10+1,1)`
		is a 0-dimensional array and therefore transposition has no effect:
		`import numpy as np`
		`c = np.transpose(c)`
		`print(c)`
		`[ 0  1  2  3  4  5  6  7  8  9 10]`

Transition from MATLAB to Python, and back.	MathWorks provides support to migrate PYTHON code to MATLAB and provides tools to run PYTHON code within MATLAB (Source). Numerous documents help PYTHON uses to get started with MATLAB (Source).	There are countless websites where you can get instructions on how to switch from MATLAB to PYTHON. Furthermore, there are more or less powerful tools for translating the code, z.B. MATLAB2PYTHON (Source).