Matlab Middle Layer and AT 1.4

L. Nadolski,

Accelerator Physics Group, SOLEIL

https://sourceforge.net/projects/atcollab

https://sourceforge.net/projects/atcollab/
http://atcollab.sourceforge.net/index.html

Contents

- MML in a nutshell
- Matlab versions and compatibility
- How easy is the migration to AT 1.4?
- Conclusion

MML Genesis

Using Matlab for Accelerator Experimentation and Control
or
A Matlab "MiddleLayer" (MML)

Gregory J. Portmann
Jeff Corbett, Andrei Terebilo, James Safranek (SSRL)
Christoph Steier, Tom Scarvie, Dave Robin (ALS)
Laurent Nadolski (SOLEIL)

http://www2.als.lbl.gov/als_physics/portmann/MiddleLayer/

MML community around the word: a short list
Many users, few developers

North America: ALS, SSRL (SPEAR3), Duke FEL, NSLS2, (VUV or X-Ray rings), CLS, …

Europe: SOLEIL, LAL/THOMX (France), DIAMOND (England), ALBA (Spain), ANKA (Germany), ILSF (Iran), MAX-IV (Sweden), SOLARIS (Poland), …

Asia: PLS2 (Korea), SLS (Thailand), SSRF (China), NSRRC/TPS (Taiwan), …

Middle East: SESAME (Jordan)

Australia: ASP

Why Matlab?

- Only true available software available in late 90s
- Matrix programming language
  (variables default to a double precision matrix)
- Extensive built-in math libraries
- Active workspace for experimentation and algorithm development
- Easy of import/export of data
- Graphics library
- Compact code and good readability
- Adequate GUI capabilities
- Platform independents

Automating Physics Experiments
(without becoming a software engineer)

Goals

- Develop an easy scripting method to experiment with accelerators (accelerator independent)
  - Remove the control system details from the physicist (like Tango names and how to connect to the computer control system)
  - Easy access to important data (offsets, gains, rolls, max/min, etc.)

- Integrate simulation and online control. Make working on an accelerator more like simulation codes.

- Integrate data taking and data analysis tools

- Develop a software library of common tasks (orbit correction, tune correction, chromaticity, ID compensation, etc.)

- Develop a high level control applications to automate the setup and control of storage rings, boosters, transfer lines.

Matlab Toolbox Suite for Accelerator Physics

- **MiddleLayer + High Level Applications**
  1. Link between applications and control system or simulator.
  2. Functions to access accelerator data.
  3. Provide a physics function library.

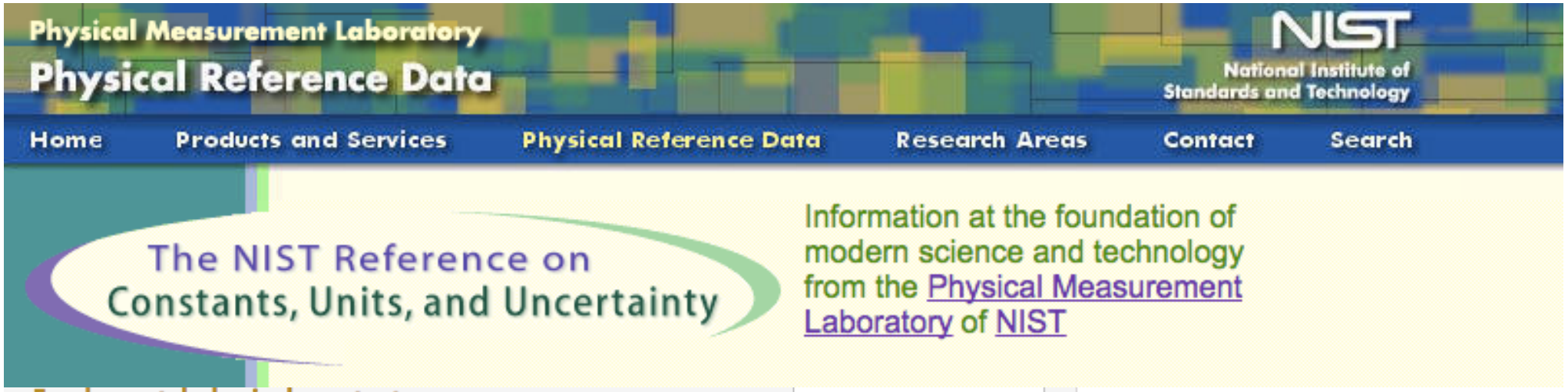
- MCA, LabCA, SCAIII – Matlab to EPICS links
- TANGO/Matlab binding

- Accelerator Toolbox for simulations

- LOCO – Linear Optics from Closed Orbits (Calibration)

- NAFF Library (frequency maps)

- Used for transfer lines, Booster, Storage Ring



Fundamental physical constants

by Jarek Luberek
22 May 2009

Functions that returns a struct() containing most fundamental physical constants.

Watch this File

Be the first to rate this file!

13 Downloads (last 30 days)
File Size: 7.39 KB
File ID: #24236
Version: 1.0

File Information

Description The struct has two levels. The first level is the name of the constant. The second level has fields: "value", "uncert" and "unit".
Example:

```
phc = fundamentalPhysicalConstantsFromNIST();
phc.speed_of_light_in_vacuum.value
```

returns

```
299792458
```

and

```
phc.speed_of_light_in_vacuum.unit
```

returns

```
ms^-1
```

Data was obtained from http://physics.nist.gov./cuu/index.html and (almost) automatically transfered to matlab syntax with the help of some c and awk programming.

The constants who's uncertainties are given av (exact), the value of 0 is returned.

MATLAB release MATLAB 7.8 (R2009a)

Fundamental Physical Constants

Defined as a Class for easy of use

Software Interconnection Diagram

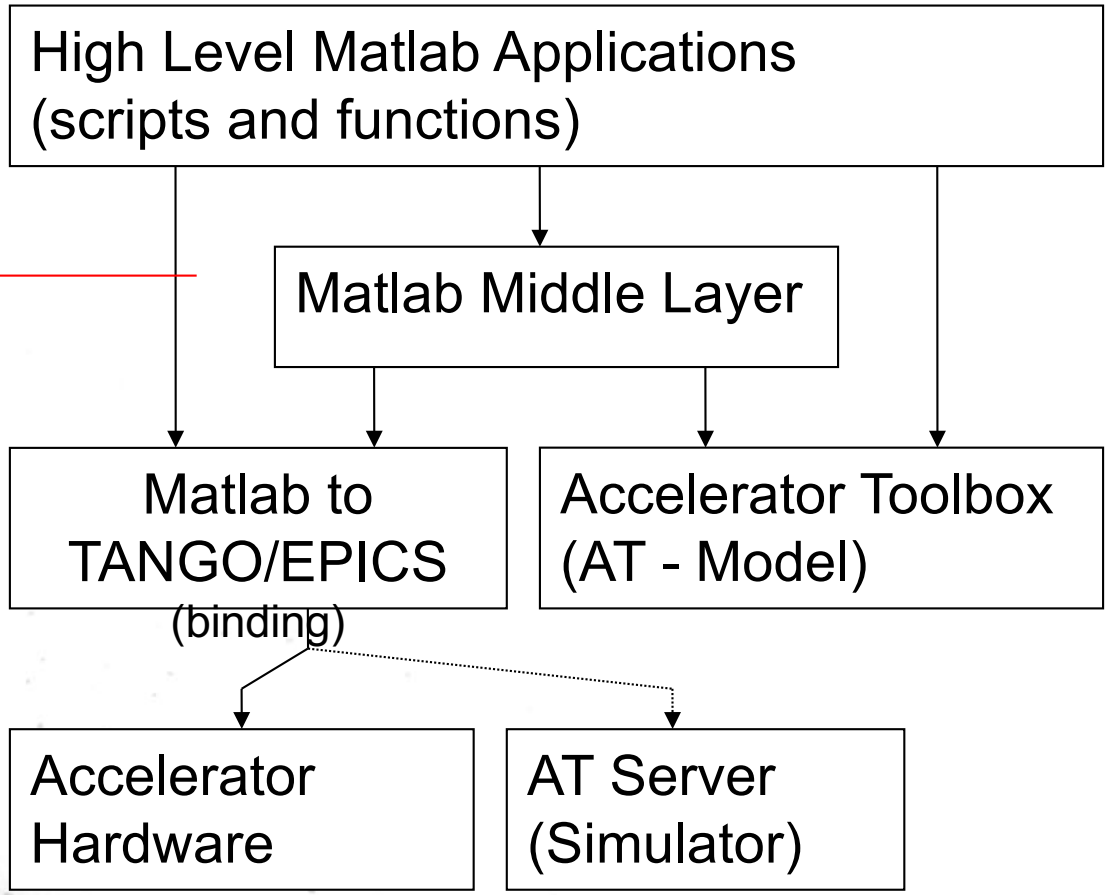
Hiden to users

- TANGO/EPICS system
- Naming convention

Device name:

ANS-C01/DG/BPM.2

Attribute name: XPosSA



Known features of MML/AT

- **Spirit and strength**
  - free of charge in our community
  - Sharing of development between labs
  - Avoid Matlab Toolbox

- **Robustness and reliability for operation**
  - For many: Machine dedicated shifts
  - For some labs: Daily operation

- **Different uses**
  - in control-rooms (online simulator)
  - Offices (simulation, optimization, design)

Various version of MML

- **Matlab Middle Layer**
  - Origin pot: ALS (G. Portmann)
  - Many forks and local development in most of the labs
    - add-ons and developments for extensive use
    - Home made functions
    - Use for controlling injector to front-ends of an accelerator facility
    - Tuned MML versions for commissioning
    - Dedicated/specific High Level Application (HLA/GUI) for accelerator physics (insertion, diagnostics, operation groups)
    - Consequence
      - Very few labs are in sync with ALS version (anyway: very few improvement and release)
      - Hundreds of Matlab scripts, applications written and interface with MML
      - Low use of ESRF AT version

- Can we do something to improve that?
- MML maintenance status?
- MML diffusion list status activity?

MIGRATION FOR YOUR LABORATORY OF AT TO THE NEW AT 1.4

Matlab compatibility with new AT version

- **Different Matlab versions used**
  - Two Matlab releases a year
  - Many evolutions of function interface since 2009 (figure return a structure, deprecated function, ...)

- **Matlab usage**
  - Development
  - Operation of the Accelerator in the control room (reliability/robustness are a MUST)

- **Legitimate Questions**
  - Is my MML setup compatible?
  - How much work to do a migration to AT 1.4?

Short Answer

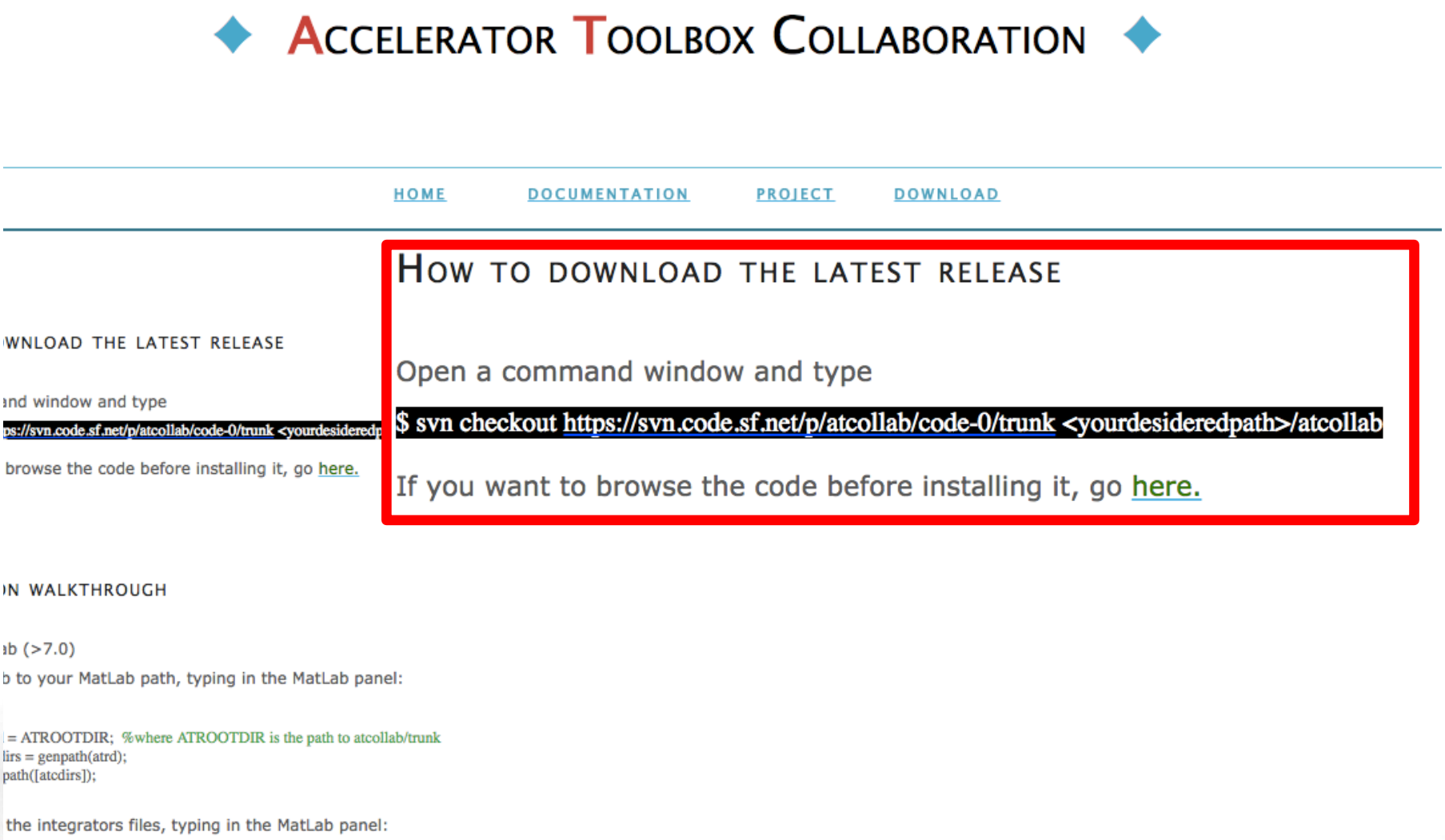
- **Very few showstoppers have been identified between AT 1.4 and MML**

- **Efforts have been made to keep**
  - The AT 1.2 interface with MML using global variable (for instance the RING structure is not a mandatory input argument)
  - Thorough tests have been done at SOLEIL

- **Simple procedure**
  - Download AT 1.4 **at http://atcollab.sourceforge.net/download.html**
  - In the past, in principle:
    - **MMLROOT/AT** hosted the AT version from SLAC
    - **MMLROOT/MML/AT** hosted your own AT development
  - Define a new path to AT 1.4 in your setpathmml fonction (MMLROOT/AT)
  - Recompile with : atmexall
  - **Its works seamlessly with 2009 to 2017a Matlab version at SOLEIL**

- **For Matlab 2009 version** and earlier
  - num2cell.m need to be overloaded when AT is loaded in your session

Atcollab web site: http://atcollab.sourceforge.net/download.html



Conclusion and Next step

- AT 1.4 is mature with lost of **new capabilities**
  - See list of features in Boaz's and Nicolas's talks
- AT 1.4 is an improved version with **better performance**
- **Let us increase the community of AT 1.4 users**
- When upgrade to recent Matlab versions
  - Be aware that some GUIs need some minor upgrades with recent versions of Matlab due to the figure output structure (numerical before)
  - Last version of MML (ALS 2016-02-08) included most of these small upgrades http://www2.als.lbl.gov/als_physics/portmann/MiddleLayer/