AT development and compatibility issues
Material for general discussion

Laurent S. Nadolski

Synchrotron SOLEIL

Contexts

- What is the situation?
- How is used Matlab Middle Layer (MML/AT) today?
- How many versions of AT?
- What are the constraints?
  - Development
  - Operation
  - Backwards compatibility
- Work methodology
- Towards solutions to increase benefits and reduce head-ache maintenance

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)

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

USA: ALS, Stanford (Spear3), Duke FEL, NSLS2, (VUV or X-Ray rings),

Canada: CLS

Europe: SOLEIL, 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

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

- Accelerator Toolbox for simulations

- LOCO – Linear Optics from Closed Orbits (Calibration)

- NAFF Library (frequency maps)

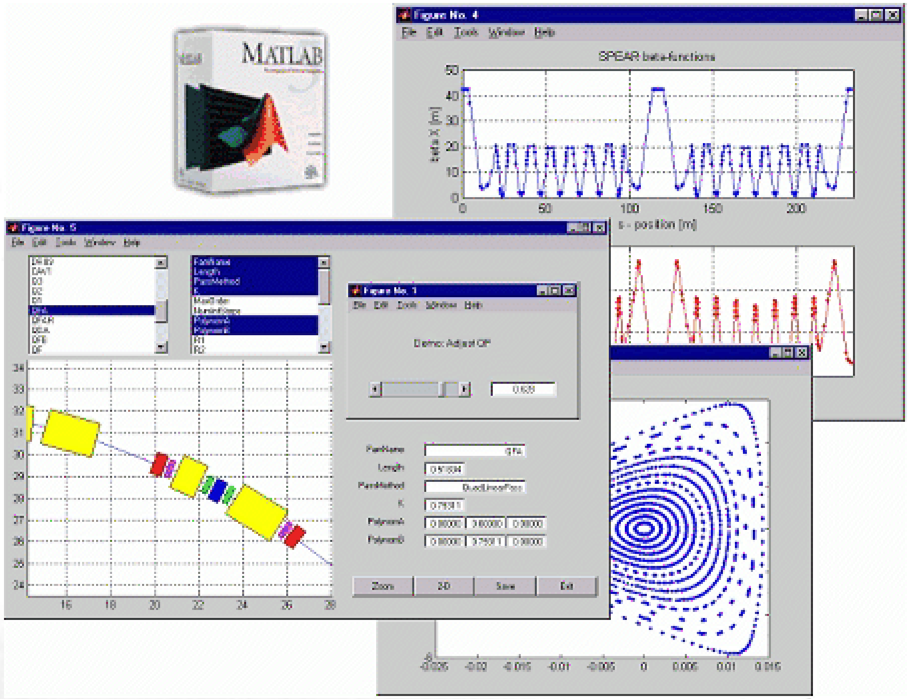
- Used for transfer lines, Booster, Storage Ring

AT - Accelerator Toolbox

Andrei Terebilo

MATLAB ® Toolbox for Particle Accelerator Modeling

Accelerator Toolbox is a collection of tools to model particle accelerators and beam transport lines in MATLAB environment. It is being developed by Accelerator Physics Group at Stanford Synchrotron Radiation Laboratory for the ongoing design and future operation needs of SPEAR3 Synchrotron Light Source.



What is Accelerator Toolbox

New in AT version 1.2

Download and Installation

Get Started

Collaboration

Publications

e-mail AT

Links

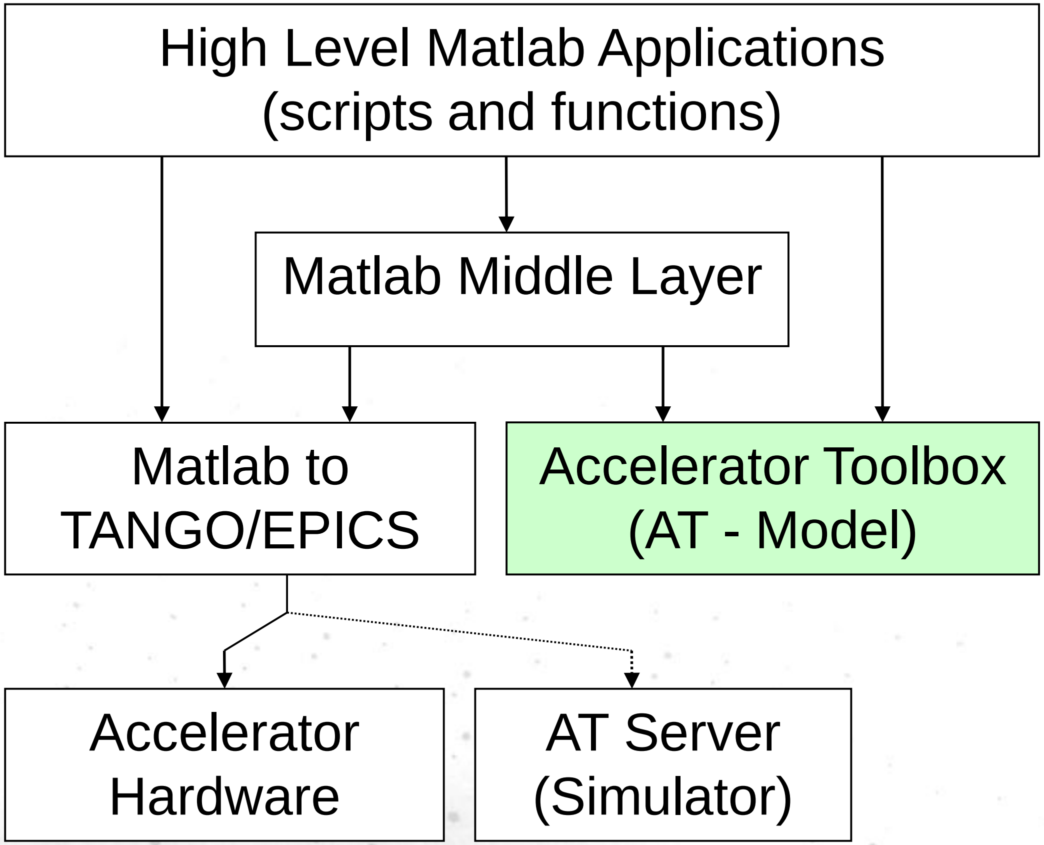
www-ssrl.slac.stanford.edu/at/welcome.html

http://www.slac.stanford.edu/~terebilo/at/

Various classes of users of AT

Use of AT at online simulator in MML

Use of AT as standalone application



High Level Matlab Applications (scripts and functions)

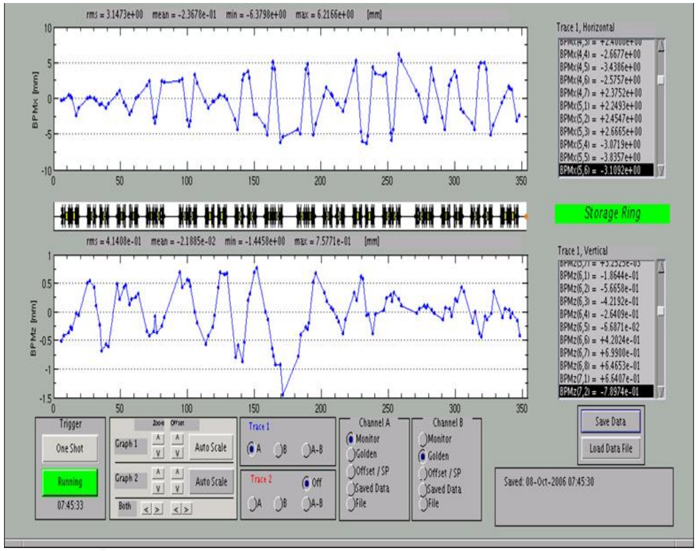
Matlab Middle Layer

Matlab to TANGO/EPICS

Accelerator Toolbox (AT - Model)

Accelerator Hardware

AT Server (Simulator)



```
rms = 3.1473e+00    mean = -2.3678e-01    min = -6.3798e+00    max = 6.2166e+00    [mm]
```

Trace 1, Horizontal

```
BPMx(4,4) = -2.6677e+00
BPMx(4,5) = -3.4386e+00
BPMx(4,6) = -2.5757e+00
BPMx(4,7) = +2.3752e+00
BPMx(5,1) = +2.2493e+00
BPMx(5,2) = +2.4547e+00
BPMx(5,3) = +2.6665e+00
BPMx(5,4) = -3.0719e+00
BPMx(5,5) = -3.8357e+00
BPMx(5,6) = -3.1092e+00
```

```
rms = 4.1408e-01    mean = -2.1885e-02    min = -1.4458e+00    max = 7.5771e-01    [mm]
```

Trace 1, Vertical

```
BPMz(6,1) = -1.8644e-01
BPMz(6,2) = -5.6658e-01
BPMz(6,3) = -4.2192e-01
BPMz(6,4) = -2.6409e-01
BPMz(6,5) = -6.6871e-02
BPMz(6,6) = +4.2024e-01
BPMz(6,7) = +6.9900e-01
BPMz(6,8) = +6.4653e-01
BPMz(7,1) = +6.6407e-01
BPMz(7,2) = -7.8974e-01
```

Trace 1: ● A ○ B ○ A-B

Trace 2: ● Off ○ A ○ B ○ A-B

Channel A: ● Monitor ○ Golden ○ Offset / SP ○ Saved Data ○ File

Channel B: ○ Monitor ● Golden ○ Offset / SP ○ Saved Data ○ File

Saved: 08-Oct-2006 07:45:30

AT ESRF Fork

- Great developments, major add-ons since AT birth by A. Terebilo (see previous presentation today)

- Enhanced flexibility

- But low consideration of its integration in MML (of used at ESRF)
  - Remove the use of global variables (THERING, FAMLIST)

  → Need modifications of interface of many functions to take the lattice as new input

MML/AT version and use

- AT:ESRF works well if use standalone

- MML/AT
  - 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

Known features of MML/AT

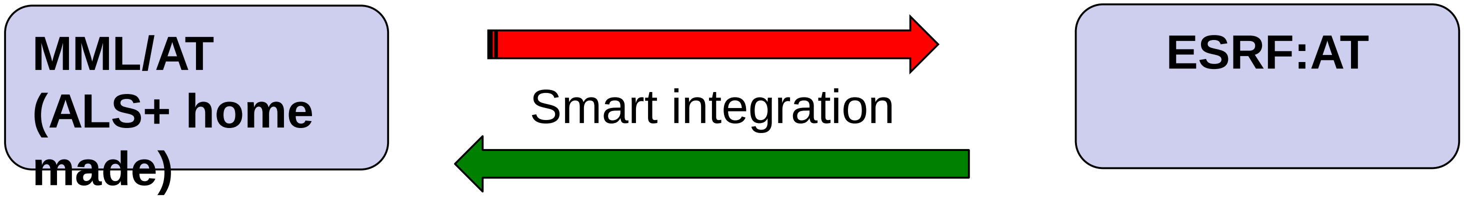
- **Spirit and strength**
  - free of charge in our community
  - Sharing of development between labs

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

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

- Many links between MML and AT since MML simplifies a lot the interface and make use of common nomenclatures for accelerator components

Most important rule for most of us: do not break the operation, existing development (backwards compatibility)



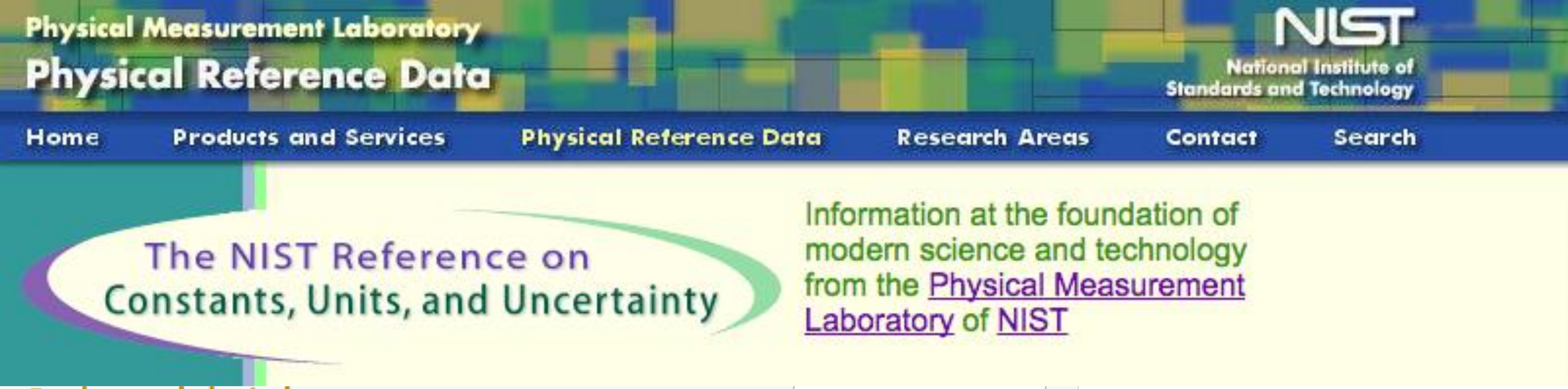
MML/AT (ALS+ home made)

Smart integration

ESRF:AT

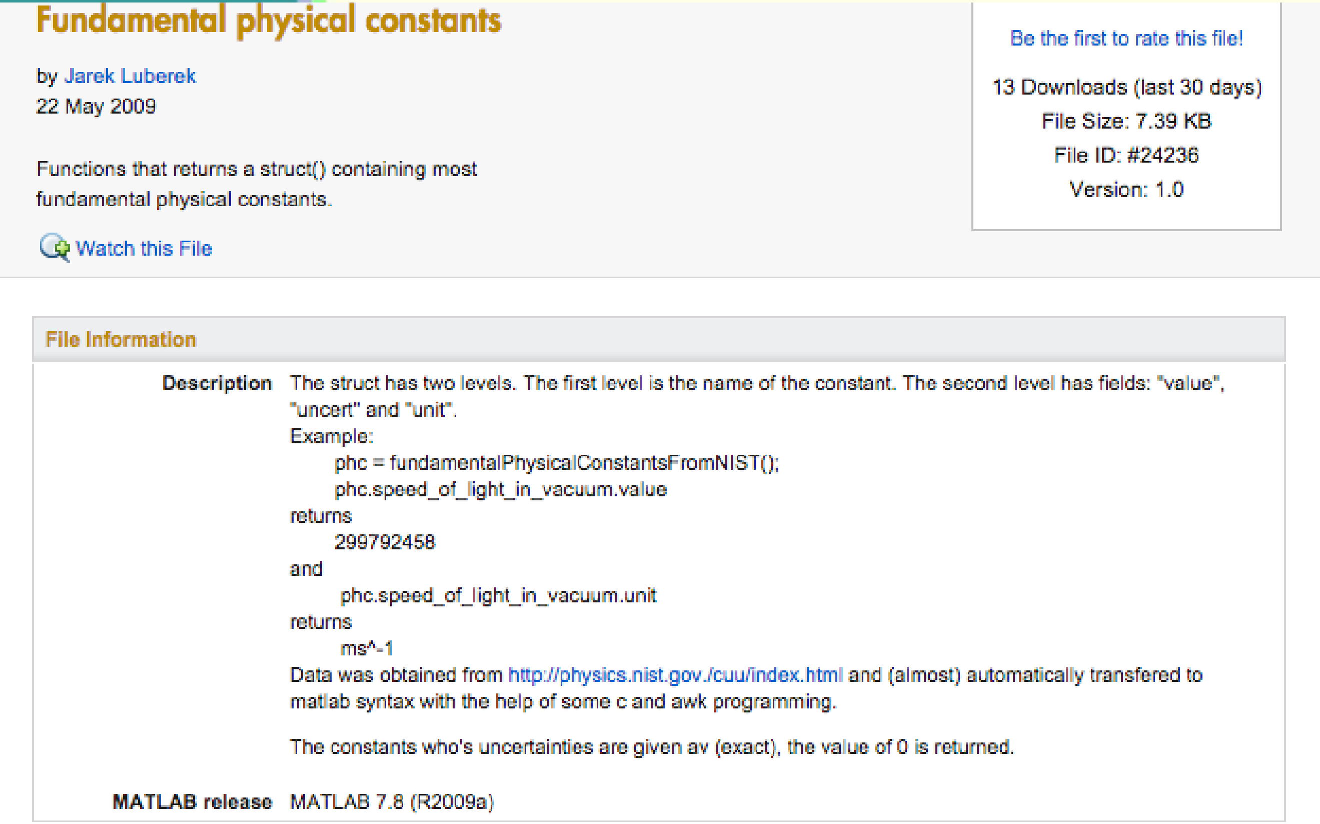
- Goal: get benefit from AT(ESRF)
- How to find a "magic" (smart) integration
  - Scenario 1: upgrade MML/AT (ALS version +home made dvpt) to make it compatible
    - Effort in many labs
    - Need human resources
    - Risk analysis: not uniform (some labs will not do it)
  - Scenario 2: upgrade AT (ESRF version) to enable it to work seamlessly with MML

THOUGHTS TO IMPROVE AT/MML



The NIST Reference on Constants, Units, and Uncertainty

Information at the foundation of modern science and technology from the Physical Measurement Laboratory of NIST



Fundamental physical constants

by Jarek Luberek
22 May 2009

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

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

AT improvements

- Passmethod (integration methods)
  - – Switchyard to select default method for each element for a given a lab
    - Consideration like fringe fields, small circumference ring, asymmetric edge focusing, etc.
- Atmatch
  - – Powerful but still a bit too much expert
  - – Is there any project to develop a simple interface, GUI for simple users
- Collective effect (ESRF, SOLEIL development)
- Large number of duplicated functions (atx, atsummary, etc.)
- Library of lattice parser between main codes (MADX, ELEGANT, TRACY, etc.)
  - – Many different classes of integration and element