Accelerator Toolbox and atcollab: Experience and perspective on direction forward

B. Nash

ESRF

with many contributions from
ESRF beam dynamics group

Widely used Codes for electron rings

Elegant
APS-Borland et. al.

MAD-X/PTC
CERN- module structure
with organizers

AT
???

BMAD
Cornell- D. Sagan
http://www.lepp.cornell.edu/~dcs/bmad/

(OPA, Tracy, …?)

Accelerator Toolbox (AT)

Created by Andre Terebilo at SLAC in late 1990's
He left accelerator physics in 2009.

AT is a Matlab based electron tracking code, with integration routines written in C for reasons of speed. There are also many routines for calculation of ring parameters. Some single pass (linacs, transfer lines) utilities exist as well.

AT code is used at more than 20 different synchrotrons

phase space coordinates in AT

$$\vec{z} = \begin{pmatrix} x \\ x'(1+\delta) \\ y \\ y'(1+\delta) \\ \delta \\ cT \end{pmatrix}$$

$$\delta = \frac{E - E_0}{E_0}$$

atcollab

How to organize AT development across multiple labs, when it is already in use by so many?

Can we support an open source collaborative development for the AT code?

Needed parts of open source project:

*) Repository in version control system: SourceForge chosen with svn
*) communication amongst users and developers: (mailing list, wiki, website…)
*) documentation, bug reporting, etc
*) well defined releases

http://sourceforge.net/projects/atcollab/

considerations:
> copyright issues? (I asked A. Terebillo and he suggested we go ahead)
> branching and forking of code?
> How to find common ground and bring different development together?

atcollab communication

mailing list: atcollab-general@lists.sourceforge.net

27 members from 15 institutes

ESRF (5)
NSRC, Taiwan (3)
LBL (3)
IHEP, China (3)
ALBA (3)
SLAC
Australian Synchrotron
Mayo Clinic
INFN
Diamond
KIT, ANKA
SOLEIL
ELETTRA
Canadian Light Source
LAL

on Twitter:
@atcollab_tweets

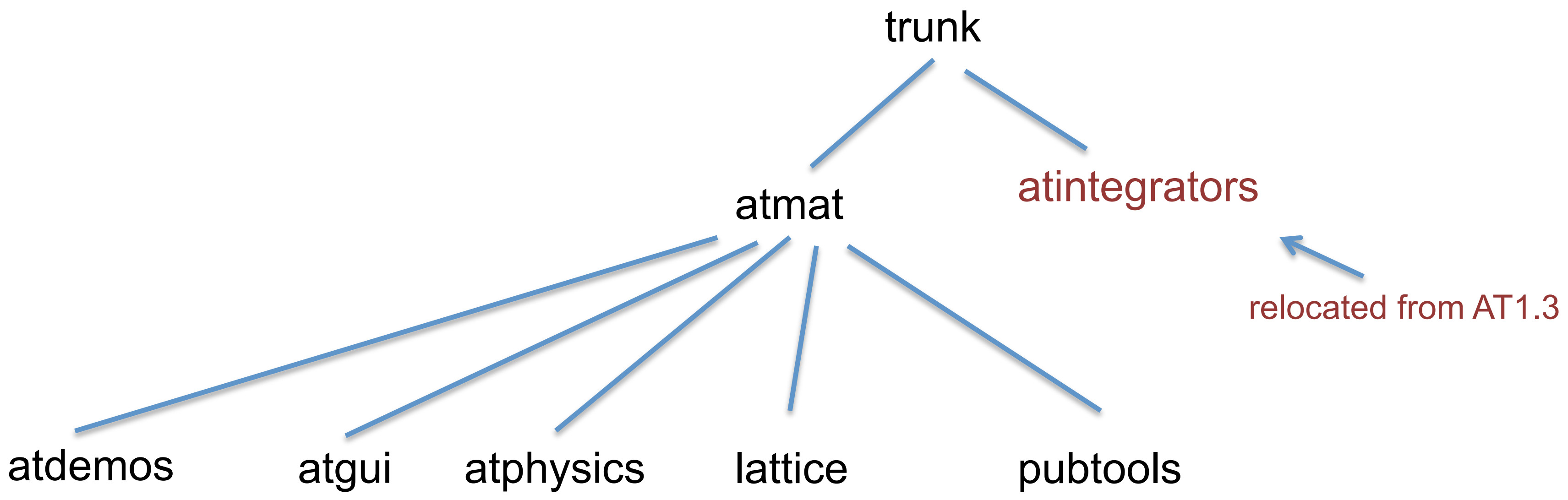
Archives and subscription info at

https://lists.sourceforge.net/lists/listinfo/atcollab-general

A better website is needed!

atcollab svn repository

http://sourceforge.net/projects/atcollab/



trunk

atmat

atintegrators

relocated from AT1.3

atdemos

atgui

atphysics

lattice

pubtools

get copy of repository with:

```
svn checkout https://svn.code.sf.net/p/atcollab/code-0/trunk atcollab
```

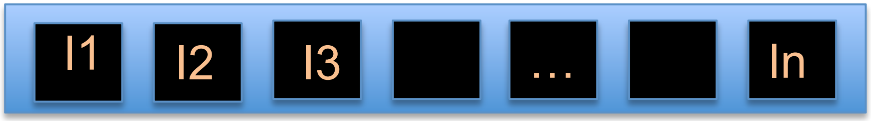
Having repository online facilitates email discussion. Can refer to code with URL. e.g. ringpass function is here:
http://sourceforge.net/p/atcollab/code-0/HEAD/tree/trunk/atmat/attrack/ringpass.m

AT is also distributed with MML by Greg Portmann.
als.lbl.gov/als_physics/portmann/MiddleLayer/Release/at

If possible, we should bring these versions together!

Tracking in AT

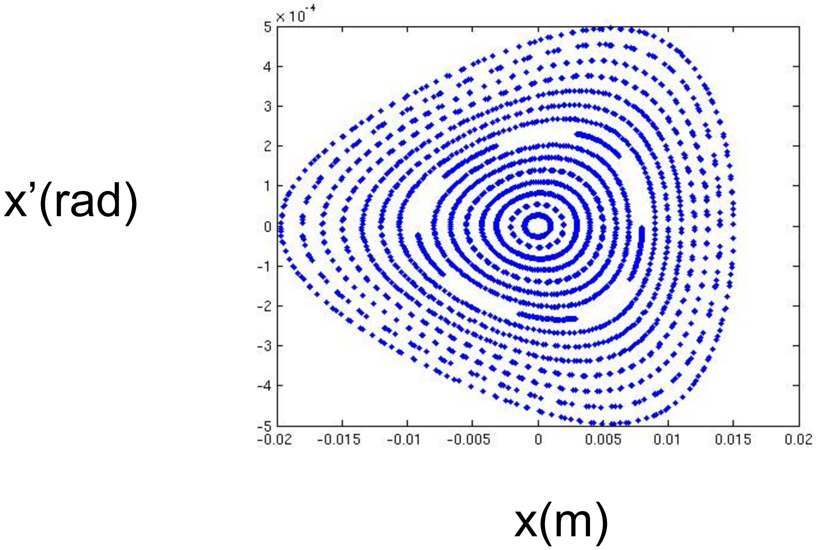
ringpass calls atpass via PassFunction



Integrators
called with
lattice parameters

can be called individually in Matlab
via MexFunction

```
>> z0=[0.001 0 0 0 0 0]'*(1:15);
>> zf=ringpass(ring,z0,200);
>> plot(zf(1,:),zf(2,:),'.b')
```



Integrators

- AT has the following integrators in the atintegrators directory:

AperturePass.c

BendLinearPass.c

BndMPoleSymplectic4E2Pass.c

BndMPoleSymplectic4E2RadPass.c

BndMPoleSymplectic4Pass.c

BndMPoleSymplectic4RadPass.c

CavityPass.c

CorrectorPass.c

DriftPass.c

EAperturePass.c

IdTablePass.c

IdentityPass.c

Matrix66Pass.c

QuadLinearPass.c

SolenoidLinearPass.c

StrMPoleSymplectic4Pass.c

WiggLinearPass.c

GWigSymplecticPass.c

Others have developed different integrators.
We should find the most accurate and fastest integrators.

Integrators

AT structure makes it very easy to see what integration routine is being used and to find the code.

e.g. DriftPass:

```
void DriftPass(double *r_in, double le, int num_particles)
/* le - physical length
   r_in - 6-by-N matrix of initial conditions reshaped into
   1-d array of 6*N elements
*/
{       int c, c6;
        double p_norm, NormL;
        for(c = 0;c<num_particles;c++)
                {       c6 = c*6;
                    if(!mxIsNaN(r_in[c6]))
                        {    p_norm = 1/(1+r_in[c6+4]);
p_norm = 1/(1+r6[4]);
NormL = le*p_norm;
r6[0] += NormL*r6[1];
r6[2] += NormL*r6[3];
r6[5] += NormL*p_norm*(r6[1]*r6[1]+r6[3]*r6[3])/2;
}
```

C code

mex function
passfunction

compiled with atmexall function

called from atpass via
passFunction

In-coming phase-space point

element

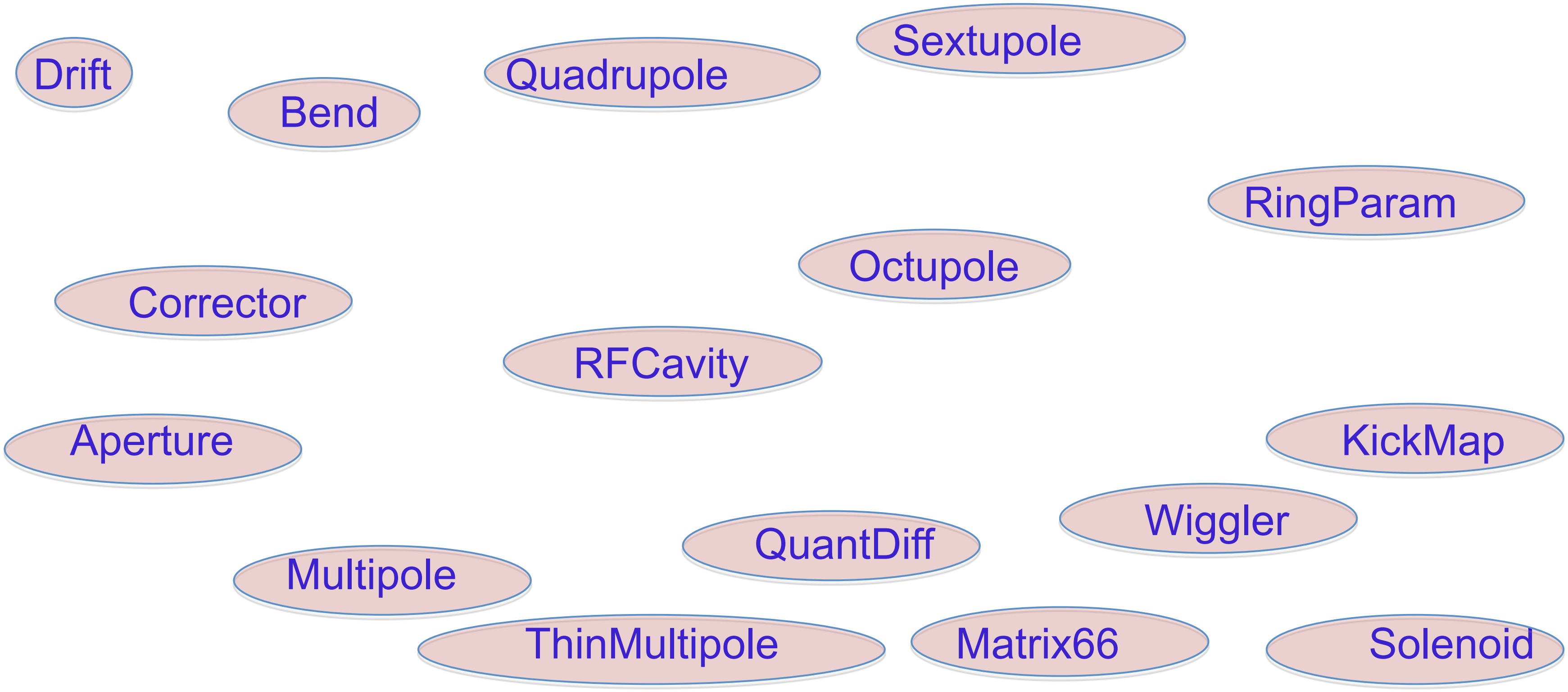
Integrators may also be called directly:

Calling syntax, via MexFunction: >>DriftPass(D2,[.001 0 0 0 0 0]')

Classes

Issue: how to write functions that identify particular elements that will work for lattices at different labs?

Solution: add a “Class” field to all elements



Drift

Bend

Quadrupole

Sextupole

RingParam

Octupole

Corrector

RFCavity

Aperture

KickMap

Wiggler

QuantDiff

Multipole

ThinMultipole

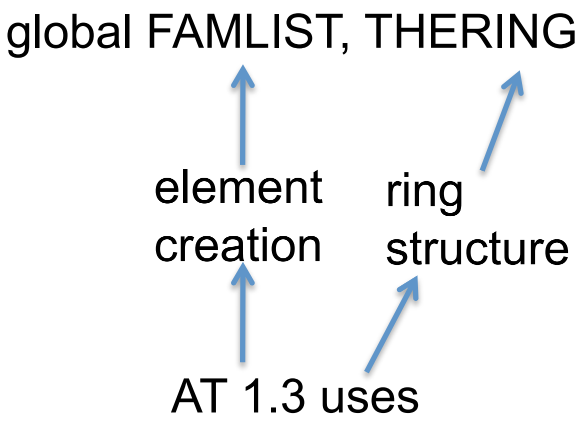
Matrix66

Solenoid

Example:

indsext=findcells(‘ring’,’Class’,’Sextupole’)
gives indices of all sextupoles

Moving away from global Variables Element and Ring Creation



global FAMLIST, THERING

element creation

ring structure

AT 1.3 uses

<span style="color:red">Issue</span>: We want to be able to create many lattices in one Matlab session

For the element creation, output elements directly. For example

```
DR1=atdrift(‘D1’,0.7)
QF=atquadrupole(’QF1’,0.94,0.39,’QuadMPoleFringePass’)
SH=atsextupole('SH',0,0,'StrMPoleSymplectic4Pass');
B1=atsbend(’B1',0,0,0,'BndMPoleSymplectic4Pass');
```

Now build lattice with

```
RING=[{DR1};{QF};{SH};{B1};{DR1};   …    ]
```

RingParam

Issue: if we want to define many different ring structures at the same time, we need a way to define global parameters for each one.

Idea: create an element in each ring to store global parameters for that ring.

ATRINGPARAM(FAMNAME,E0,NBPERIODS)

reference energy

periodicity of lattice

Not many functions yet developed to take advantage of this feature.

(IdentityPass needed as PassMethod for RingPass)

Documentation

- most functions already have pretty good documentation within Matlab
- A manual is needed for finding functions and clarifying conventions and best programming methods. Physics assumptions should be better documented, and references to other manuals and primary literature given.

Backwards compatibility and development issues

Example:
Global turn #

In AT1.3 turn number is not available to the integration routines.

Simple change to atpass:
pass turn number?

issue: need to make small change in pass_function of all pass methods to accept this.

In the end, we found a different solution to make turn number accessible without requiring changing argument list in each pass method.

Interface of electron beam with x-rays

example:
BMAD code (D. Sagan) is including both electrons and x-ray tracking.

Electron beam distribution plus bending magnet or undulator determines x-ray beam. There may be cases where mutual optimization of electron beam with x-ray optics could be beneficial.

Emittance and lattice functions may be adequate to define electron beam, but detailed coupled beam second moments may be useful in some cases.

Closed orbit and Sigma matrix output at requested locations should be adequate.

So, an alternative to following the BMAD approach is to define data formats at interface of electron beam and x-ray beam.

A project of a “dictionary” for x-ray beam lines has been started by Manuel Sanchez del Rio. It includes definitions of electron beam. In some cases, we may need to generalize to non-Gaussian electron beam.

Conclusions and direction forward?

- atcollab is started and mailing list and svn repository on Sourceforge established
- backwards compatibility with AT1.3 has been enforced to extent possible.
- contributors have added code to pubtools directory
- though encouraging contributions from numerous facilities, majority of atcollab is ESRF
- How to get a single version of AT?
- Interaction with MML and LOCO?
- Interaction with x-ray codes should be considered. Rather than direct integration, definition of data formats (e.g. Sigma matrix and positions) should be adequate
- Substantial work on creation of a manual still required
- Identification of most accurate and fastest pass methods available within the whole accelerator community should be continued.
- Well defined releases should be considered. AT 2.0?