The State of the ETEAPOT (*E*lectric *T*hin *E*lement *A*ccelerator *P*rogram for *O*ptics and *T*racking) Code, and its Context/Environment

J. Talman

January 15, 2013

Abstract

An overview of ETEAPOT is presented. A brief introduction continues with explicit instructions and examples of repeatable output. This allows simple direct verification of benchmarked work (both transverse and longitudinal). While the betatron/transverse propagation is well in hand ([1], [2]), the longitudinal/synchrotron is less established. Allowing new physicists to exploit this functionality fairly easily; I regard this as the main result of this project (to date). Physics implications can be expected to follow. For example, the need for reduced dispersion has already been suggested. Significant constraints remain. For example "modest curvature" lattice elements (-1 ≤ m ≤ 1) have turned out to be nearly equivalent for our purposes. However, some recent work ([3]) indicates that what's needed is m values more than 100 times this big! This make the "Bend, Kick, Bend" algorithm approach all the more essential. This report aims to flesh all this out.

1 Introduction

As seen at URL http://code.google.com/p/ual/source/list?num=25&start=25 the "Initial directory structure" for UAL was created May 6, 2008. N. Malitsky was the primary repository contributer until Aug 30, 2009 (UAL::243) or so. This is a very powerful/robust software architecture accomodating arbitrary accelerator lattice descriptions (assuming they permit a closed orbit). I have been the primary repository contributer (though in the restricted domain of all electric accelerators) since. "*E*lectric *D*ipole *M*oment latest" (UAL::272 - UAL::967 or so) is at URL http://code.google.com/p/ual/source/list . Roughly speaking (and this is all a little rough), Nikolay created/imported the original TEAPOT (mostly magnetic) functionality as UAL and I have been grafting on the electric lattice capability. The UAL architecture underpins all my work. Figure 1 is an iconic representation of its foundational nature.

1

Expanding slightly, the acronym <u>A</u>ccelerator <u>P</u>ropagator <u>D</u>escription <u>F</u>ormat enables any and all ($\forall$) lattice element algorithms!

Expanding slightly again, the acronym <u>S</u>tandard e<u>X</u>change <u>F</u>ormat enables any and all ($\forall$) lattice elements!

Probably the most important modification to the above synopsis is the central role of R Talman. He was central to the progenitor TEAPOT code, the philosophy of which circumscribes the UAL repository. Of course, Richard has been central to the entire UAL (all 966 revisions) development. Of course my involvement is due to him.

Probably the second most important modification to the above synopsis is the role of A Luccio and SPINK. The proof of concept of "adding" spin to the UAL architecture interested some important people.

Fanglei Lins name should be mentioned as well.

Nikolay is likely winding down his UAL involvement, and I might be, as well.

Again, my role has been to augment the existing magnetic functionality with electric (bend) functionality. Possible scenarios/trajectories for this effort are

(T1) I drop out, and the only result is the improved ability of Richard to help create a satisfactory electric storage ring

(T2) I drop out, and the code continues to help Richard, and possibly other physicists

(T3) I continue developing the code, but the effort remains mostly confined to me and Richard

(T4) I continue developing the code, become part of the relevant community, and the code develops a foothold

With an eye on handing off this work, this report targets scenario (T2) in/on the off chance that some fairly independent physicist(s) attempt to use it.

Possibly it helps me achieve (T3) and/or (T4).

2 Graphical, somewhat whimsical, restatement

Figure 2 presents an iconic portrayal of (T1): UAL, at least the electric part, closes. Presently, Richard moves on.

Figure 3 presents an iconic portrayal of (T2): UAL/ETEAPOT is but a small part of a general tool set. Physics is open ended (ie the primary domain).

Thus, with an eye on adding UAL/ETEAPOT to some physicists tool boxes, I turn to more concrete specification.

3 Computer Syntax for Repeatable Results

The exact steps:

```
svn co http://ual.googlecode.com/svn/trunk/ ual1
cd ual1
tcsh
setenv UAL 'pwd'
source setup-linux-ual
```

```
    make clean
    make >&! make.log

    cd examples/ETEAPOT
    cat README
```

and the inexact step:

```
    "follow the relevant instruction file"
```

are designed to lead to repeatable results.

For now, I will follow one of the three instruction files referred to in README,

runExampleFor_ETEAPOT_MltTurn

as it targets previously benchmarked longitudinal/synchrotron motion. It looks like:

```
cp Shell.cc $UAL/ext/UI/src/UAL/UI/Shell.cc
pushd $UAL/ext/UI
make clean
make
ll lib/linux           // e.g. 1154818 libUalUI.so
popd

cp $UAL/codes/ETEAPOT_MltTurn/src/ETEAPOT_MltTurn/Integrator/RFCavityTracker-mod.cc
    $UAL/codes/ETEAPOT_MltTurn/src/ETEAPOT_MltTurn/Integrator/RFCavityTracker.cc

pushd $UAL/codes/ETEAPOT_MltTurn
make clean
make
ll lib/linux           // e.g.  574307 libETEAPOT_MltTurn.so
popd

cp data/eteapot_MltTurn.apdf data/eteapot.apdf

make outputReady
make clean
make

// We recommend the lenovo ThinkPad line with ubuntu, or Scientific Linux
// On an ubuntu single processor vostro 200, 12/31/2012, this next command
// takes 10 minutes or so, and generates a 3G or so file
./ETEAPOT_MltTurn ./data/E_BM_M1.0.RF.sxf -1 40 1 500 >! OUT
// On an ubuntu single processor vostro 200, 12/31/2012, this next command
// takes 10 minutes or so, and generates a 3G or so file

// "look" at NikolayOut
perl SplitNikolayOut.pl NikolayOut >! IG

gnuplot
    gnuplot> l 'forGnuplot_Figure2'
```

3

4 Results

When these instructions are adhered to religiously (easier said than done), gnuplot gives Figure 4. The reader can take my word that this is identical to Figure 2 in [13], or he can verify it for himself. This latter is my preferred option as it facilitates ETEAPOT becoming part of the tool kit.

The original description in [13] is

```
Longitudinal dynamics in lattice BM M1.0.RF (having m=-1) is shown in Figure 2. (For the outermost
amplitude the initial conditions have been adjusted to eliminate the initial betatron amplitudes.) The
three benchmark lattices are compared in the similar plots of Figure 3. For this comparison only the
three innermost amplitudes are tracked. In these plots three particles start at the origin traveling in the
forward direction, with fractional energy offsets UAL = 0.000001, 0.000002, 0.000003. The three plots
are essentially identical. We interpret this to mean that the quadrupoles adjacent to the bend elements,
having been adjusted to restore the vertical tune to its standard value Qy = 0.2, effectively cancel the
focusing effect coming from non-zero field index m. In other words, as far as longitudinal dynamics is
concerned, the three benchmark lattices are equivalent.
```

A similar procedure (only difference being l 'forGnuplot_Figure11' in gnuplot) duplicates Figure 11.

The original description in [13] is

```
A possibly surprising aspect of the oscillation is illustrated by the bottom figure of Figure 11. The
tracked particle executes longitudinal oscillations of amplitude 0.5 m. This graph can therefore be
usefully compared with the outermost of the orbits shown in Figure 2. Both have a similarly large
ct-ranges. The tracked particle starts side-by-side with the reference particle, but with too little speed
(UAL = 105), as can be seen from Figure 2. Acting as if above resonance its shorter path length
dominates is too-low speed, and it takes the tracked particle only 25 turns to get to the front of the bunch,
as can be seen from Figure 11. But then it takes about 50 turns to get back level with the reference
particle, and 50 turns more to get to the tail of the bunch. The time distortion in Figure 11 therefore
corresponds to the up-down asymmetry in Figure 2
```

The preceding is only the (probably most important) tip of the iceberg.

Richard and I have generated a whole series of benchmark results ([9], [10], [11], [12], [13]). These can, in principle, all be recreated by the methods outlined above.

The instruction file

runExampleFor_ETEAPOT_MltTurn

is only one of three mentioned in file $UAL/examples/ETEAPOT/README

Others are

determineTwissCleanRunPlotM1.**0**_**sl4**

runExampleFor_orbitsWithSpin

5 Gotchas

This is all a little error prone. I'm working on improving it! The consensus seems to be that UAL is difficult, if not "dang near impossible" [2], to use.

This whole approach is predicated on the work of Munoz and Pavic [4]. The time of flight calculation, in particular, should be looked at more closely. One could take a different point of view, that I have been evaluating their work. Richard and I feel that it is at least mostly correct. There's a Munoz Evaluation section that goes into more detail on this.

My sense is that Richard and I will normalize to the lenovo ThinkPad line.

Furthermore, we may impose an operating system, e.g. Scientific Linux, as well.

There seems to be some (non show stopping) inconsistencies among the various potential UAL platforms. These we've learned to live with. There seems to be some (show stopping) inconsistencies among more widely divergent platforms. These have to be filtered out.

6 Discussion

Again, I am here targeting my "scenario (T2)" which involves other people potentially using this code and related material.

I have gathered all the documents cited below electronically. How long this will be available is anybodies guess!

Of course the google codes repository is likely to be around for a while! I may just check in all these supporting documents.

Again Figure 4 is simply recreatable, and duplicates previously benchmarked work. [13] shows that this "m=-1" ("saddle") lattice is nearly indistiguishable from "parallel" (m=0) and "toroidal" (m=1) lattices. This is within the constraint of small ($Q_y \approx 0.2$) values (so that the vertically focussing quads in the various lattices are quite different) for the vertical tune. Obviously very important if true! A corollary might be that m values more than 10 times these values might be called for ([14]). This in turn, might obsolete my ETEAPOT code as it uses a linearized equation for handling deviations from m=1. The relevant theory is in section 5.2 of [8].

These results, while repeatable/verifiable (or hopefully at least something to have a dialog over) are possibly incorrect. The time of flight per probe, per bend, is critically important. Our current results are a little controversial ([1]). The next section, Munoz Evaluation, goes into a little more detail.

7 TBD

I'm hoping/planning to make this a stand alone document. I feel that it is close now, though oriented for this presentation..

I'd like to better document the betatron/transverse capability and cross checking.

Figure 6 quotes transverse as "verified", and longitudinal as "ongoing". While I view this as correct, I should expand!

8 "Verified" Transverse Verification

"Near script" determineTwissCleanRunPlotM1.0_sl4 specifies a command line

./determineTwiss ./data/E_BM_M1.0_sl4.sxf -1 40 >! OUT

This implicitly uses a "split". File extractParameters.h has a parameter splitForBends. This manifests itself on lines 238 -253 of determineTwiss.cc

```
238   if( typeOutput=="Sbend        "){
239 // nonDrifts++;
240    totSplits=2*pow(2,splitForBends);
```

```
241 // std::cerr << "totSplits " << totSplits << "\n";
242    sBndDlta=(sX-sPrevious)/totSplits;
243 // sBndDlta=(sX-sPrevious)/(1+splitForBends);
244    for(int j=0;j<totSplits;j++){
245      std::cerr << "name " << nameOutput << " type " << typeOutput << " " << xX << " " << yX << "
246 //   std::cerr << "name " << nameOutput << " type " << typeOutput << " " << xX << " " << yX << "
247      algorithm<double,PAC::Position>::bend_m_elementName[bend]=nameOutput;
248      algorithm<double,PAC::Position>::bend_m_sX[bend++]=sPrevious+sBndDlta;
249      nonDrifts++;
250      sPrevious+=sBndDlta;
251 //   algorithm<double,PAC::Position>::bend_m_sX[bend++]=sX;
252    }
253    }
```

Following determineTwissCleanRunPlotM1.0_sl4 religiously gives the following table:

Table 1: Split Tabulation

splitForBends	Q_x	β_x	Q_y	β_y
0	0.460553	35.9195	0.200880	263.356
1	0.460479	35.9227	0.200401	263.975
2	0.460460	35.9235	0.200281	264.131
3	0.460455	35.9238	0.200251	264.169
4	XXXXXXXX	XXXXXXX	XXXXXXXX	XXXXXXX
4	0.460455	35.9245	0.200244	264.179

The line with Xs is the result of the "out of box" code (8192) with splitForBends = 4. To get the next row, one must modify algorithm.hh to e.g.

```
29     static std::string bend_m_elementName[32768];   // [16384];   // [8192];
30     static double bend_m_sX[32768];   // [16384];   // [8192];
```

and algorithm.icc to e.g.

```
7   std::string algorithm<Coordinate, Coordinates>::bend_m_elementName[32768];   // [16384];   //
8 template<class Coordinate, class Coordinates>
9 double algorithm<Coordinate, Coordinates>::bend_m_sX[32768];   // [16384];   // [8192];
```

9   Munoz Evaluation

TEAPOTs "Drift, Kick, Drift" approach for maintaining symplecticity and other reasons is well documented (see e.g. [6]). For ETEAPOT, this approach has been upgraded to "Bend, Kick, Bend". This upgrade is made necessary by the variation of particle speed in electric fields (unlike magnetic). The lattice element whose continuous evolution is to be modelled as a delta function impulse is assumed known exactly. Here we are trying a refinement where we know the spherical bend element and its kick exactly, and then kick correct that for elements whose propagation is not

known exactly (e.g. Cylindrical bend elements). Taking $\mathbf{F} \equiv -k\hat{\mathbf{r}}/r^2$ as our "to be kick corrected force", this gives

$$\mathbf{F} = -k\frac{\hat{\mathbf{r}}}{r^2} = \frac{k}{r^2}\frac{d\hat{\boldsymbol{\theta}}}{d\theta} \tag{1}$$

(using $d\hat{\boldsymbol{\theta}}/d\theta = -\hat{\mathbf{r}}$). This implies planar motion and conserved angular momentum L (per probe, per bend). Choosing polar coordinates (θ, ϕ) in this plane, $L = \gamma m_p r^2 \dot{\theta}$. Note that [8] calls γ, γ^I, to emphasize that it is inside a bend element. Since I am always inside a bend element here, I use the simpler notation. Defining $\mathbf{u} \equiv \gamma\mathbf{v}$ so that $\mathbf{p} = m_p\mathbf{u}$

$$\frac{d\mathbf{p}}{dt} = \frac{d\mathbf{p}}{d\theta}\frac{d\theta}{dt} = m_p\frac{d\mathbf{u}}{d\theta}\frac{d\theta}{dt} = \frac{L}{\gamma r^2}\frac{d\mathbf{u}}{d\theta} \tag{2}$$

The classical law of motion $\mathbf{F} = d\mathbf{p}/dt$ holds good including in the semi relativistic region we are working with ([7]). Thus

$$\frac{k}{r^2}\frac{d\hat{\boldsymbol{\theta}}}{d\theta} = \frac{L}{\gamma r^2}\frac{d\mathbf{u}}{d\theta} \tag{3}$$

so that

$$\frac{d\mathbf{u}}{d\theta} = \frac{k\gamma}{L}\frac{d\hat{\boldsymbol{\theta}}}{d\theta}. \tag{4}$$

Both the r dependence, and the t dependence have been hidden. Of course the exact time dependence is of critical importance in the phasing of particles relative to the RF Cavity source. This is the main concern, here.

Defining $\mathbf{h} = \mathbf{u} - \frac{k\gamma}{L}\hat{\boldsymbol{\theta}}$ gives

$$\frac{d\mathbf{h}}{d\theta} = -\frac{k}{L}\frac{d\gamma}{d\theta}\hat{\boldsymbol{\theta}}. \tag{5}$$

This form motivates the choice of $\mathbf{h}$ as it shows that it is conserved nonrelativistically ($\gamma = 1$, a constant). It turns out that it is conserved relativistically in circular orbits ([4], their Eq. 8). This makes it especially good for the near circular design orbits of the EDM experiment.

Then

$$h_\theta = u_\theta - \frac{k\gamma}{L} = \gamma v_\theta - \frac{k\gamma}{L} = \gamma r\dot{\theta} - \frac{k\gamma}{L} = \frac{L}{m_p r} - \frac{k\gamma}{L} \tag{6}$$

so that

$$\gamma = \frac{L^2}{km_p r} - \frac{Lh_\theta}{k} = \frac{L}{m_p r^2\dot{\theta}} = \frac{dt}{d\theta}\frac{L}{m_p r^2} \tag{7}$$

and

$$\frac{dt}{d\theta} = \frac{Lr}{k} - \frac{m_p r^2}{k}h_\theta = \frac{r}{k}(L - m_p r h_\theta). \tag{8}$$

7

This equation is interesting in its own right, and seems to be characteristic of this (Munoz) approach. Calculated quantities often seem to be the difference of 2 terms, each of similar size. Numerical imprecision becomes a major worry. Richard and I have been wrestling with these for at least a couple of years. We feel that we've handled/overviewed them pretty well.

From tracking, we think we know all these quantities (r, L, h_θ). L is constant (per probe, per bend). We think we have exact analytical formulas for r and h_θ right through each bend.

Various equivalent massaged expressions exist ([8] equations 148, 149, and 150).

We have tried a fair cross section of them, and always get quite similar answers. Still the possibility that we are making a mistake somewhere remains. A probably over pedantic list of error sources is:

- Is the Munoz formalism correct for this application?
- Did we calculate this formula correctly?
- Did we enter it into Maple correctly?
- Did Maple give the right answer(s)?
- Did I transcribe the answer(s) correctly?
- Should we have a "kick" correction in time, analogous to the kick correction in momentum?
- Is/are there simple "back of the envelope" approximation(s) that increase confidence?

Investigating some of these may be a future direction for me.

10 References

References

[1] Semertzidis, Y. Personal Communication

[2] Talman, R. Personal Communication

[3] Talman, R. Reduced Dispersion Proton EDM Storage Ring Lattices December 10, 2012

[4] Munoz, G., Pavic, I. A Hamilton-like vector for the special-relativistic Coulomb problem European Journal of Physics 27(2006) 1007-1018

[5] Boyer, T. Unfamiliar trajectories for a relativistic particle in a Kepler or Coulomb potential American Journal of Physics 72(2004) 992-997

[6] TEAPOT: A Thin-Element Accelerator Program For Optics And Tracking Schachinger, L. and Talman, R. Particle Accelerators 1987, Vol. 22, pp. 35 - 56

[7] Talman, R. Geometric Mechanics Page 360 John Wiley & Sons, Inc. 2000

[8] Malitsky, N., Talman, J., and Talman, R. Appendix UALcode: Development of the UAL/ETEAPOT Code for the Proton EDM Experiment Personal Communication December 15, 2012

[9] Talman, R. All-Electric Proton EDM Lattices for Benchmarking Simulation Codes Mar. 12, 2012

[10] Talman, J. and Talman, R. UAL/ETEAPOT Results for Proton EDM Benchmark Lattices April 27, 2012

[11] Talman, J. and Talman, R. UAL/ETEAPOT Proton EDM Benchmark Comparisons II: Transfer Matrices and Twiss Functions August 30, 2012

[12] Talman, J. and Talman, R. UAL/ETEAPOT Results (Augmented) for Proton EDM Benchmark Lattices October 29, 2012

[13] Talman, J. and Talman, R. UAL/ETEAPOT Proton EDM Benchmark Comparisons III: Dispersion, Longitudinal Dynamics and Synchrotron Oscillations January 10, 2013

[14] Talman, R. Reduced Dispersion Proton EDM Storage Ring Lattices December 10, 2012
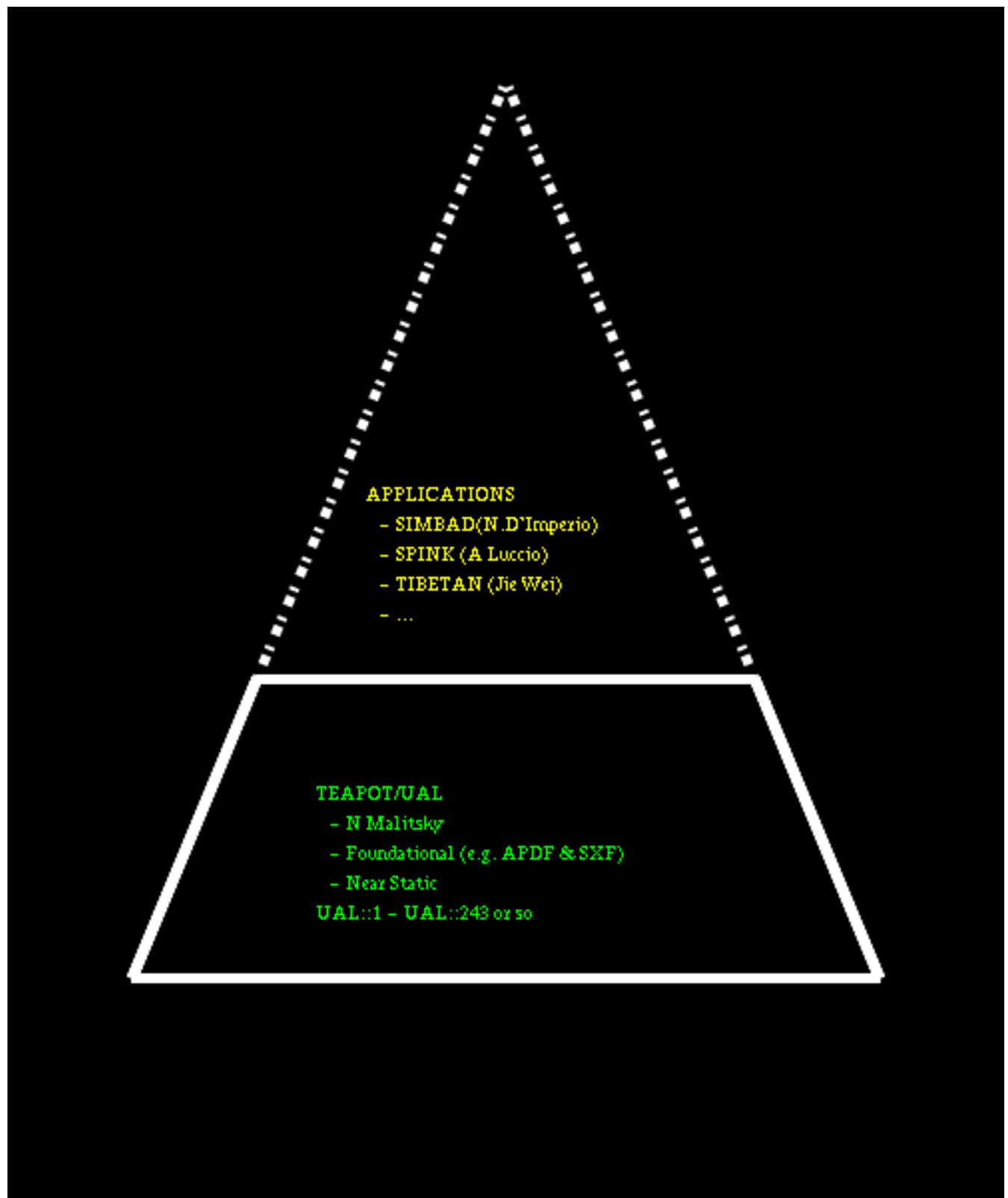
Figure 1: The "UAL Project View" (highly iconic representation of the UAL Architecture, and some successful prior projects)
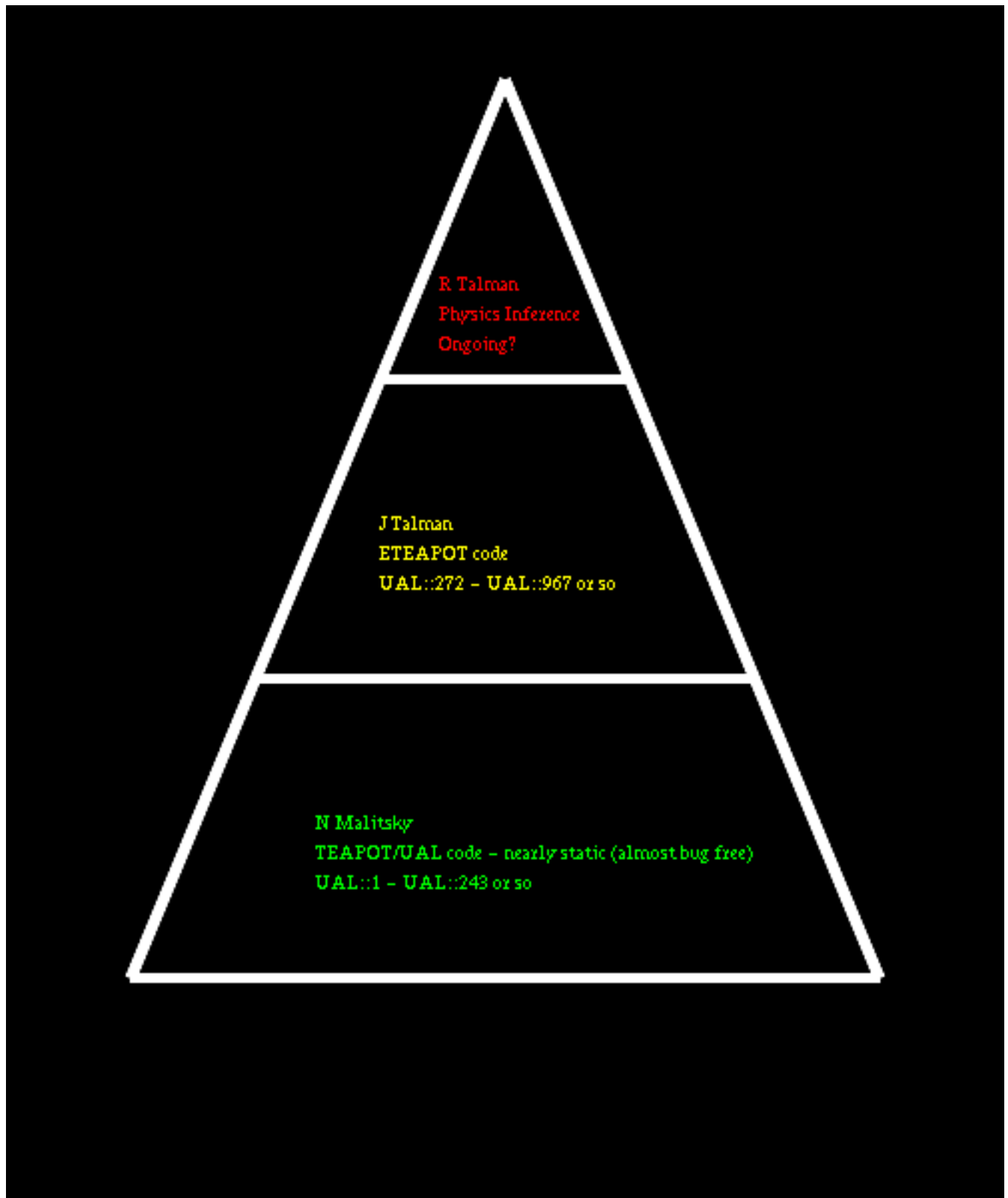
Figure 2: Johns ETEAPOT Project view of the UAL world. I've enabled/facilitated electric bend physics simulations for physicists, and I'm done! ;-)
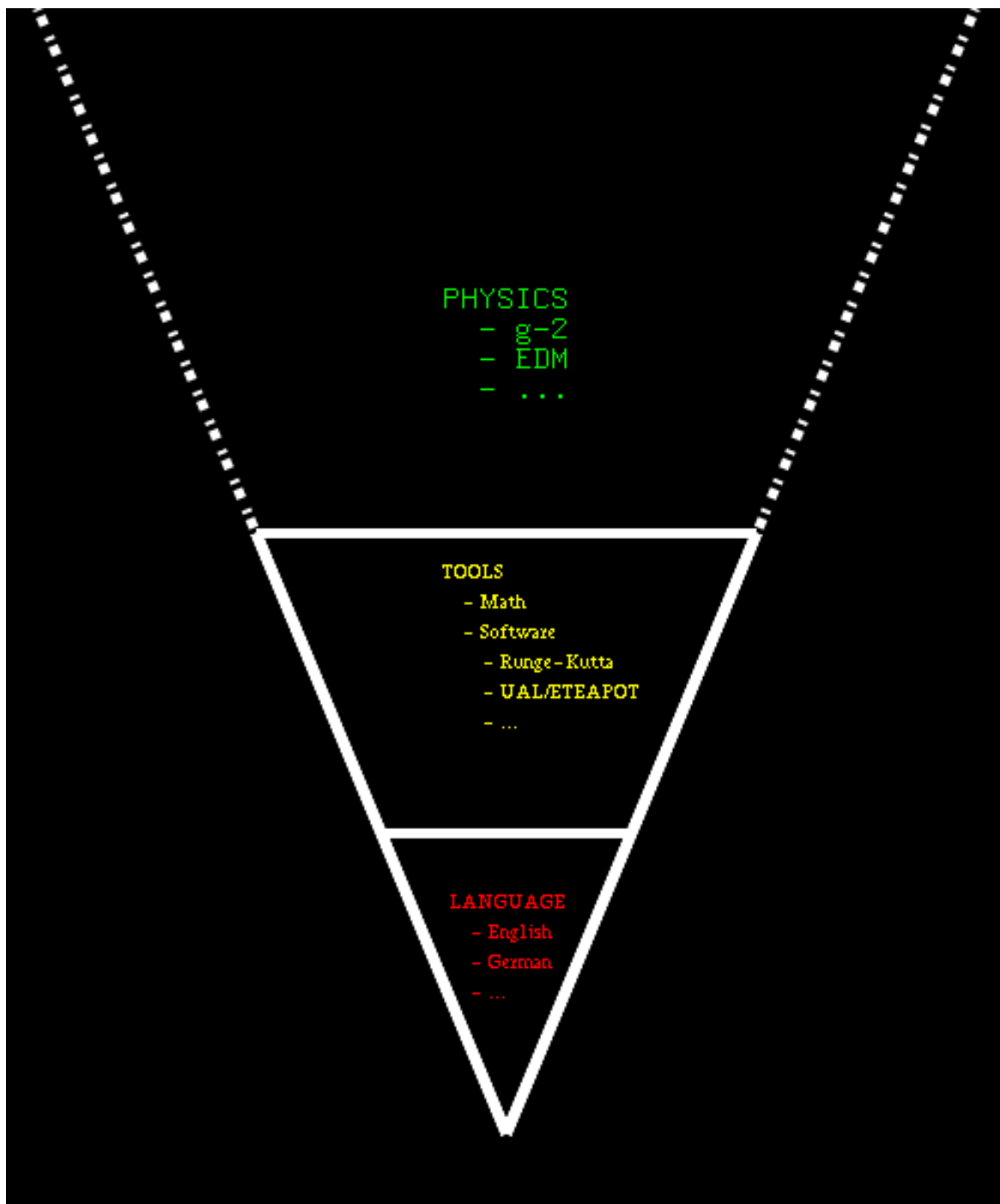
Figure 3: Johns Toolbox view of the physics world. I'm trying to add UAL/ETEAPOT to the tool box!
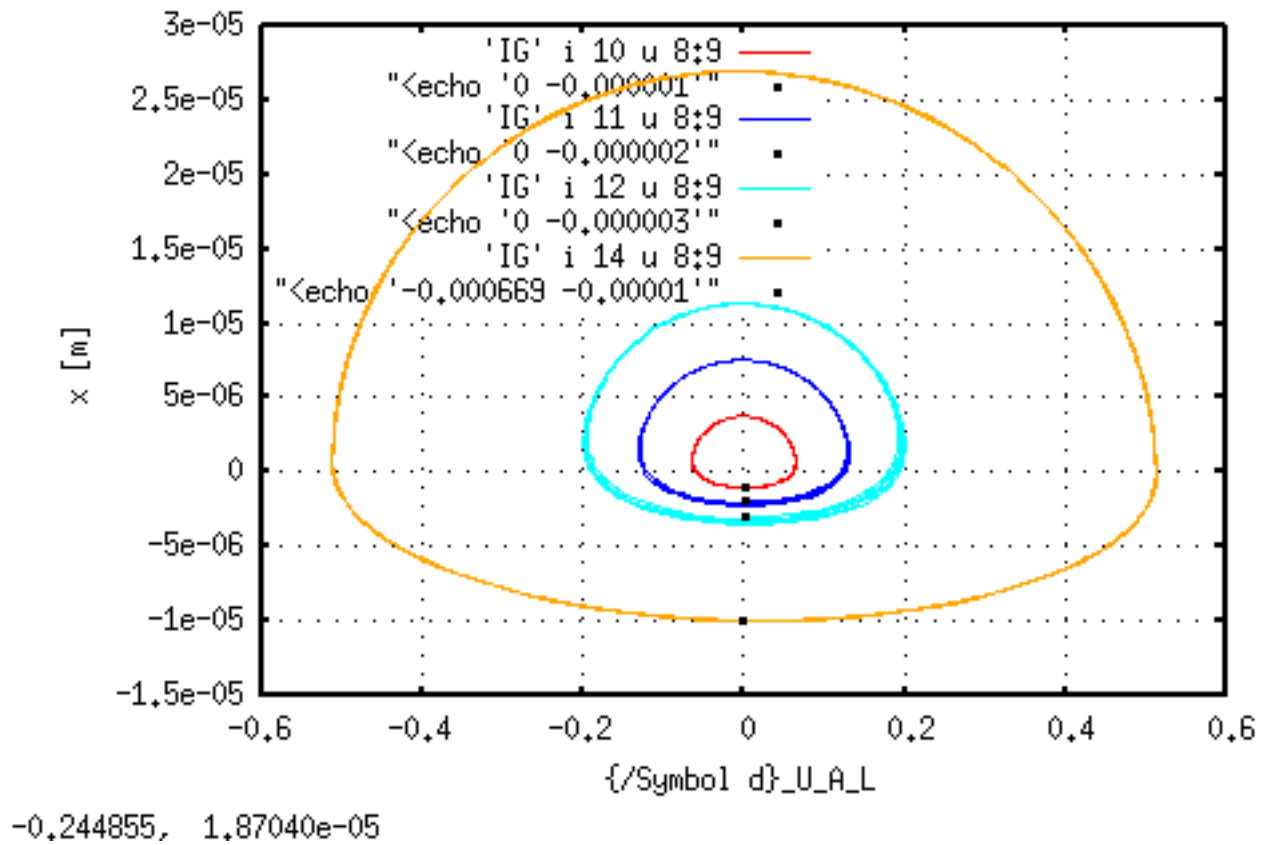
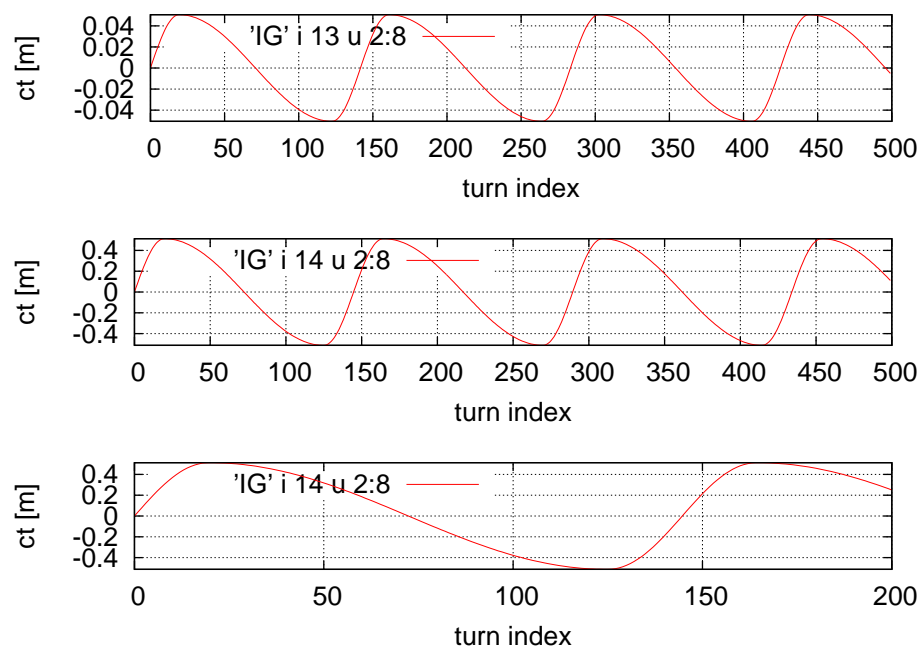Figure 4: Duplication of the Figure 2 from [13]
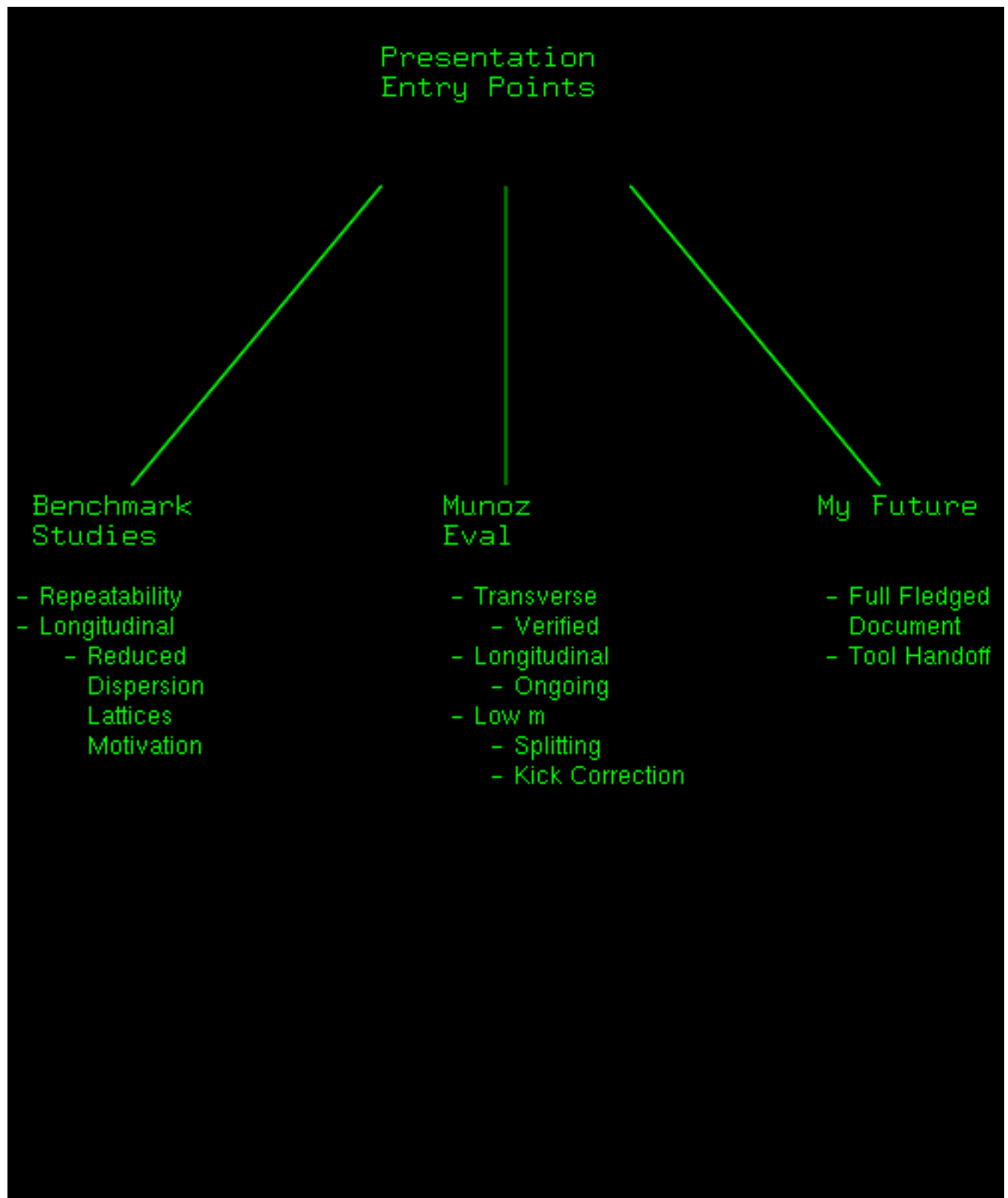
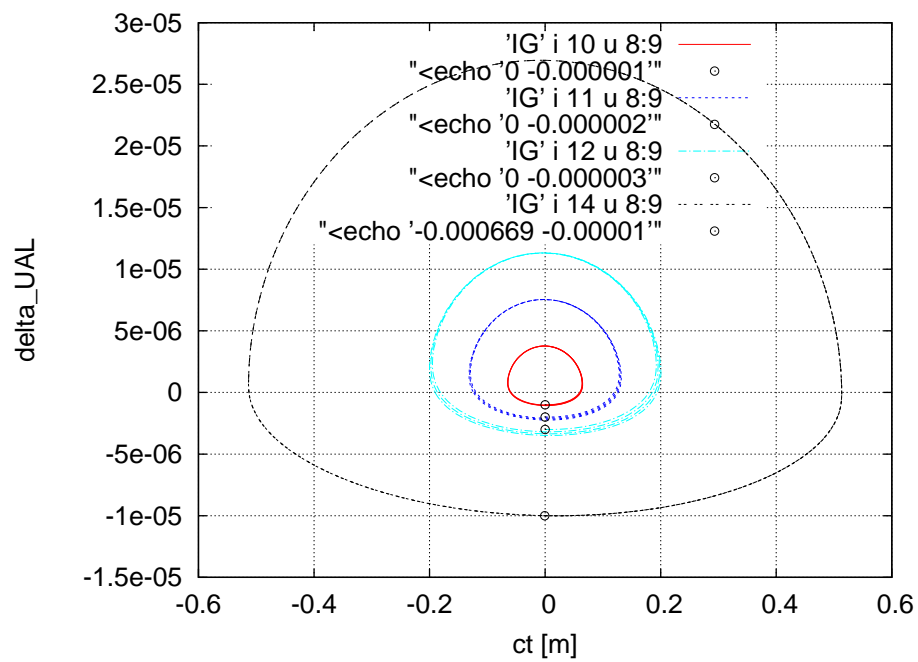Figure 5: Duplication of the Figure 11 from [13]

Figure 6: Duplication of the Figure 11 from [13]

Figure 7: Duplication of the Figure 11 from [13]