Modelling Nonlinear Dynamics in the Beams of LEP

John Jowett
Accelerator Physics Group, SL Division, CERN
http://wwwslap.cern.ch/~jowett/

Collaborators:

Y. Alexahin, F. Barbarin, P. Beloshitsky,
K. Goral, F.C. Iselin, G. Roy, F. Ruggiero,
S. Tredwell, A.Verdier

Work during period 1993-1998

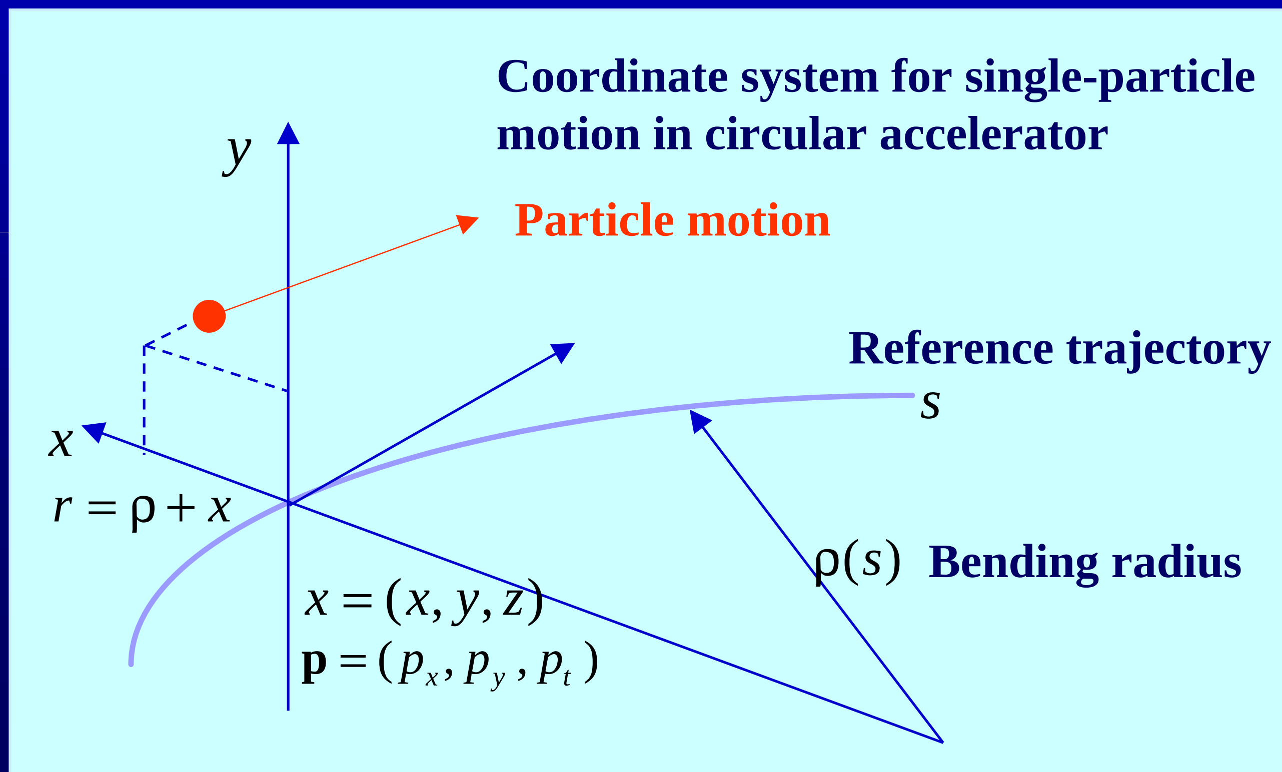
Plan of talk

- Storage ring particle dynamics in a nutshell
  - Motivation for LEP Collider at CERN
- Computational framework
  - Traditional approach
- Mathematical approach
  - Managing complexity
  - Some results
- Personal Vision

Storage Ring Particle Dynamics (1)

- Particles in beams are, by design:
  - bent to follow design trajectory (dipole magnets) ≈ **closed orbit**
  - focused (RF cavities, quadrupole magnets) according to a **beam optics**

Optics (β-functions, etc) characterise small amplitude **beta-synchrotron oscillations** (BSO) around closed orbit: 3 modes, 6-dim phase space.

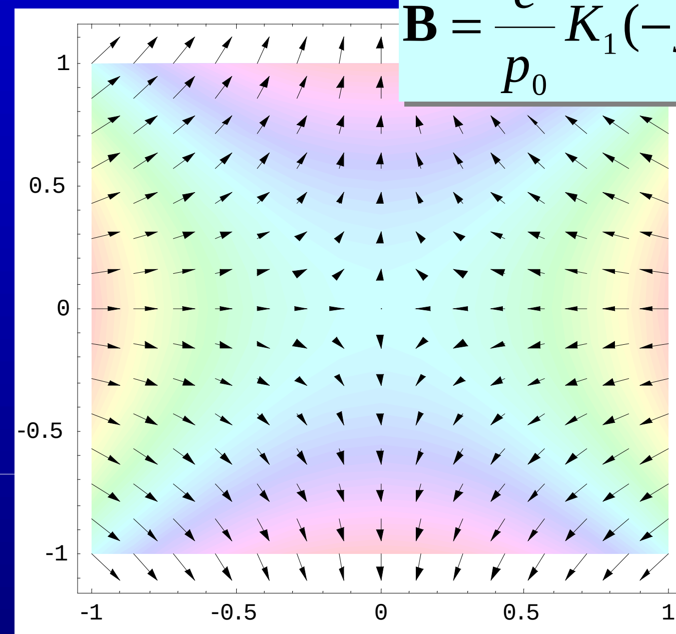


Storage Ring Particle Dynamics (2)

- Synchrotron radiation
  - Especially strong in LEP energy lost per turn up to 3 GeV out of 100 GeV
  - compensated by large RF accelerating system
  - Average "classical" part ⇒ **Radiation damping** of BSO
  - Random photon emission ⇒ **Quantum excitation** of BSO ⇒ Beam sizes (**emittances**)
  - Differences between e^+ e^-

Non-linear motion of particle in a quadrupole field

$$\mathbf{B} = \frac{e}{p_0} K_1(-y, x, 0)$$



$$x' \approx p_x \qquad \text{(for } y = 0\text{)}$$

$$p_x' \approx -p_0 K_1 x$$

$$- p\, p_x \left[c_1 K_1^2 x^2 + \sqrt{c_2} \left| K_1 x \right|^{3/2} \xi(s) \right]$$

$$c_1 \propto r_e, \qquad c_2 \propto r_e \hbar$$

Storage Ring Particle Dynamics (3)

- Real storage ring has imperfections
  - Errors and unwanted components of applied fields
  - misalignments of magnets
  - closed orbits deviate further from design trajectory
  - beam optics deviate from ideal
- Oscillations increasingly **non-linear** with amplitude
  - limited set of stable initial conditions in phase space **dynamic aperture**
  - Coupling effects $\Rightarrow$ vertical **emittance** optically corrected (orbit correction dipoles, sextupole magnets, tilted quadrupoles, …)

Optical Configuration affects Performance

- Dynamic aperture
  - allowable beam size (increases with energy), lifetime
- Energy reach of LEP increased with stronger focusing
- Luminosity (event rate) depends on vertical emittance

$$L = \frac{f_0 k_b N_+ N_-}{A} \quad \text{where}$$

f_0 = revolution frequency of ring

k_b = number of bunches circulating

N_+ = number of positrons in a bunch

N_- = number of electrons in a bunch

$A \approx 4\pi\sigma_x\sigma_y$ effective cross-sectional area of beam

Computational framework

- The MAD program
  - Accelerator design and modelling engine, used here to generate model of machine
  - compute parameters, track particles in optics
  - input is (rudimentary) programming language, output in various formats for historical reasons
- Monte-Carlo on ensemble of imperfect LEP machines
  - Predict typical effects of imperfections
  - Model the procedures used by operators to correct and compensate
  - Analyse, correlate many accelerator physics quantities

Building the LEP model in MAD

Efficient means to handle complexity of large accelerator structure, element classes, inheritance, etc.

Defining elements

```
!
MQ : QUADRUPOLE, L=1.600000
   QD : MQ, K1 = KQD
   QF : MQ, K1 = KQF
   QL4A.1 : MQ, K1 = KQL4A.
   QL4A.3 : MQ, K1 = KQL4A.
   QL4A.5 : MQ, K1 = KQL4A.
   QL4A.7 : MQ, K1 = KQL4A.
   QS2.2 : MQ, K1 = KQS2.2
   QS2.4 : MQ, K1 = KQS2.4
   QS2.6 : MQ, K1 = KQS2.6
   QS2.8 : MQ, K1 = KQS2.8
MQA : QUADRUPOLE, L=2.000000
   QL1.1 : MQA, K1 = KQL1.1
   QL1.3 : MQA, K1 = KQL1.3
   QL1.5 : MQA, K1 = KQL1.5
   QL1.7 : MQA, K1 = KQL1.7
   QL11.1 : MQA, K1 = KQL11
   QL11.3 : MQA, K1 = KQL11
   QL11.5 : MQA, K1 = KQL11
   QL11.7 : MQA, K1 = KQL11
```

Defining sequence of elements

```
! Sequence elements
!*************************************************
LEP : SEQUENCE
IP1 : IP, AT = 0.000000
   BEMI.QL1A.R1 : BEMIC, AT = 0.3065
   BSMRN.QL1A.R1 : BSM, AT = 12.067000
   ES.QL1A.R1 : ZL1A.R1, AT = 18.779000
   PUT.QL1A.R1 : BTPE, AT = 21.609000
QL1A.R1 : QL1.1, AT = 22.816000
QL1B.R1 : QL1.1, AT = 25.316000
   PU.QL1B.R1 : BPPE, AT = 26.485000
   CVA.QL1B.R1 : MCVA, AT = 26.804000, K
   PUQ.QL2A.R1 : BQPE, AT = 27.701000
   PUG.QL2A.R1 : BGPE, AT = 28.148000
QL2A.R1 : QL2A.1, AT = 29.316000
QL2B.R1 : QL2A.1, AT = 31.816000
   PU.QL2B.R1 : BPPE, AT = 32.984000
   CHA.QL2B.R1 : MCHA, AT = 33.254000, K
   SBT.QL2B.R1 : SBT1.1, AT = 34.924000,
   WIGMB.QL4A.R1 : BEUVB, AT = 58.854000
   WIG4M.QL4A.R1 : BEWI, AT = 59.934000,
```

Excitations of elements for an optics

```
KQF := .024675755912

! HIBL identical for all IPs : betay=69 betax=3
KQL6 :=  -3.108217E-02
KQL5 :=   2.761852E-02
KQL4 :=  -1.781627E-02
KQL2 :=  1.506392E-02
KQL1 := -1.293780E-02

KQL1.1 := KQL1 ; KQL1.3 := KQL1 ; KQL1.5 := KQL
KQL2A.1 := KQL2 ; KQL2A.3 := KQL2 ; KQL2A.5 :=
KQL4A.1 := KQL4 ; KQL4A.3 := KQL4 ; KQL4A.5 :=
KQL5.1 := KQL5 ; KQL5.3 := KQL5 ; KQL5.5 := KQL
KQL6.1 := KQL6 ; KQL6.3 := KQL6 ; KQL6.5 := KQL

! RFL

KDL := -.024818992513
KFL := .021399183901

KFL.1 := KFL ; KDL.3 := KDL ; KDL.5 := KDL ; KD
```

Simulating an Operational Correction

```
comment ***** CORRECT BETAY* *****
   Simulate the operational procedure of correcting betay* on one beam.

   How this works (following chat with Ghislain):
   We previously saved the ideal machine state in <optics>.ideal.pool.
   We now take note of the present betay* values in the imperfect
   machine that we have generated and pooldump its state as ....imp.p.pool
   The imperfect betay* values are saved in a file.
   We reload the perfect machine and re-match it to get the imperfect
   betay* values.  The increments are saved in a file.
   We poolload the imperfect machine again and apply the negative of
   that increment.

endcomment

savebeta,label=impEND,place=ENDLEP;savebeta,label=impIP2,place=IP2
savebeta,label=impIP4,place=IP4;savebeta,label=impIP6,place=IP6
savebeta,label=impIP8,place=IP8;
TWISS
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7.]..

Qx.imp=impEND[MUX];Qy.imp=impEND[MUY]
value Qx.imp, Qy.imp

set,bystar.IP2 , impIP2[bety]; set,bystar.IP4 , impIP4[bety]
set,bystar.IP6 , impIP6[bety]; set,bystar.IP8 , impIP8[bety]
value bystar.IP2,bystar.IP4,bystar.IP6,bystar.IP8
save,filename="IMPBYSTAR.mad",pattern="BYSTAR.*"
POOLDUMP,filename="unc.p.pool"
POOLLOAD,filename= "/afs/cern.ch/user/j/jowett/m6/ideal.pool"
value Qx.ideal, Qy.ideal, bystar.ideal
call,filename="IMPBYSTAR.mad"

! Define symmetric increments to the QS0 strengths
set,KQS0.L2.ideal,KQS0.L2 ; KQS0.L2 := KQS0.L2.ideal+DKQS0.2
set,KQS0.R2.ideal,KQS0.R2 ; KQS0.R2 := KQS0.R2.ideal+DKQS0.2
set,KQS0.L4.ideal,KQS0.L4 ; KQS0.L4 := KQS0.L4.ideal+DKQS0.4
```

- Physics masked by awkward procedure involving much file-management, copying values among variables, etc.
- Limit of reasonable usability of MAD language, starting to become very error-prone.

Earlier Approaches

- Many runs of MAD program to manage …
  - Many copies of similar input files, even more output files.
  - Unix scripts or enhancements thereof (awk, Perl, …)
  - Post-processing of output files in various formats
  - Analysis and visualisation with spreadsheets, CERN’s PAW package, …
  - Extend MAD itself
- Nevertheless, severe problems of data management and analysis
  - Lack of flexibility and interactivity
  - Rigorous evaluation of optics tedious
  - Limited human resources

Functional programming approach

- Develop functions in a higher-level interactive programming environment (Mathematica) to encapsulate the essential functions of the MAD program in this application.
  - Functions typically act on members of an ensemble of realisations of the imperfect LEPs.
    - E.g. the vertical emittance of a LEP
- Use mathematical operations to combine these functions
  - Easy in the Mathematica programming language and notebook interface
    - E.g. the correlation of vertical emittance with “dispersion”.

Benefits

- Natural and intuitive for the physicist/mathematician who is not a professional programmer.
- Manage complexity, forget (almost) about processes of generating results (running many batch jobs, gathering the data they provide).
- No loops.
- Results emerge as ensemble of mathematical functions or functional database.
- Objects in database can be anything from scalar values to surfaces in phase-space.
  - Never have to think about types or structure.

Setting up the ideal LEP

Ideal machine
no solenoids,
no RF, no radiation

→ Install vacuum chambers to limit aperture in quads

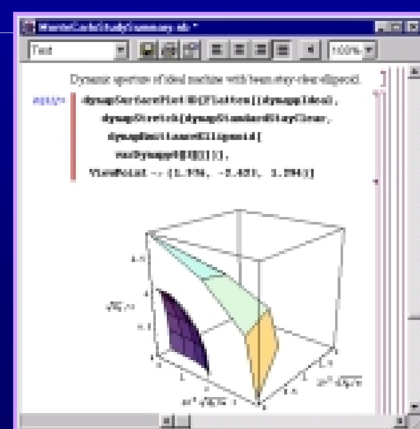
→ Compute solenoid compensation with tilted quads

→ Compute 6D linear machine, RF, radiation

$(x_n, p_n)_{1,2,3}, Q_{1,2,3},$
$U_0, \tau_{1,2,3}, \varepsilon_{1,2,3}, \ldots$

→ 6D dynamic aperture scan in 4D

(I_1, I_2, I_3, ϕ_3)



Treating one imperfect LEP

Generate ensemble of 30 imperfect machines (misalignments, tilts, field errors in all magnets)

Correct closed orbit without radiation
$\sqrt{\langle x^2 \rangle} = 0.6$ mm, $\sqrt{\langle y^2 \rangle} = 0.4$ mm

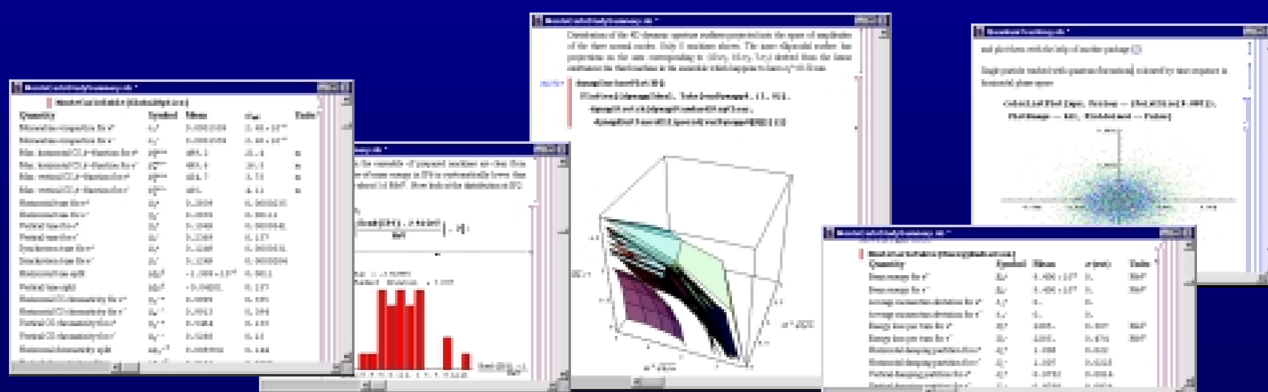
Solenoids, RF, radiation on
⇒ 6D machine for e+
Correct βy*, tunes

Compute 6D linear machine, RF, radiation
$(x_n, p_n)_{1,2,3}, Q_{1,2,3},$
$U_0, \tau_{1,2,3}, \varepsilon_{1,2,3}, \ldots$

6D dynamic aperture scan in 4D
(I_1, I_2, I_3, ϕ_3)

Reverse element sequence, build equivalent machine for e-

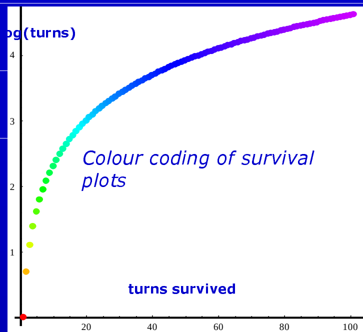
Tracking with quantum fluctuations, >200 τ_x
⇒ higher order effects on emittances



Particle Tracking and Dynamic Aperture

- One of the things MAD does
- Particles are launched with given initial conditions and tracked for many turns around the accelerator structure
  - If their orbits remain bounded, the initial conditions are inside the dynamic aperture.
- Search for boundary of dynamic aperture in 6 dim phase space ?
  - Many runs of MAD, re-launching particles
- Encapsulated in a single call of a Mathematica function
  - Returns a 3-4 dimensional dynamic aperture surface and data on particle survival
  - Mathematica package handles search procedure
  - MAD handles CPU-intensive tracking

Dynamic aperture shrinkage at high energy



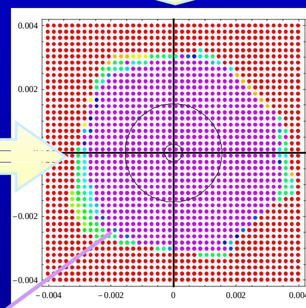
Survival plots of **initial conditions** in normal mode phase planes, units $\sqrt{\text{m}}$

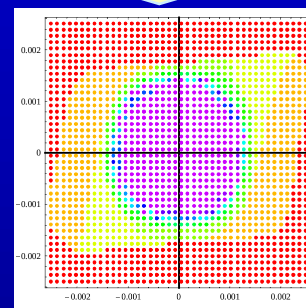
Mode 1 "horizontal betatron"

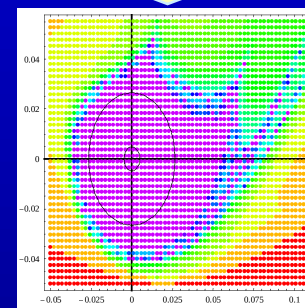
Mode 2 "vertical betatron"

Mode 3 "synchrotron"

"Comfortable" 94 GeV, J_x=1, V_{RF}=2.96 GV







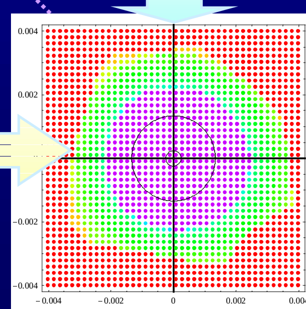
Integer resonance: $\frac{\partial Q_y}{\partial I_x}$

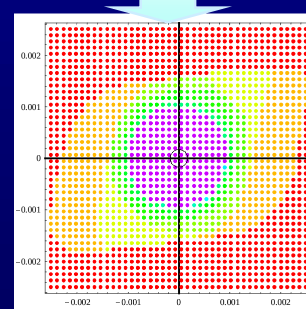
Reduction of DA by RBSC

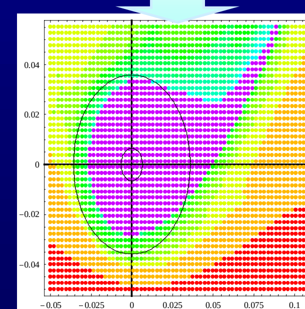
Reduction of DA by RBSC

Reduced "bucket height"

"Just sufficient" 100 GeV, J_x=1.5, V_{RF} =3.265 GV







Dynamic Programming Idiom

- See Section in Mathematica book
  `http://documents.wolfram.com/v4/MainBook/2_04/S2.4.9.html`
- "Functions that remember values they have found"

f[x_]:=f[x]= *<rhs>*

- Delayed evaluation replaced by immediate once the calculation has been done
  - Use to avoid repetition of expensive calculations
  - Saving the function f creates a database of results.
- Key to approach

Implementation

- One notebook manages the multi-step process of performing some thousands of MAD runs (few days) and distilling results into heterogeneous database of functions.
  - Functions return realisations of beam parameters in the ensemble of machines.
- “Viewer” notebooks perform statistical analysis, presentation of results
  - Answers to questions about machine performance
  - Exploratory studies
- Among these, a summary notebook, formatted appropriately, might provide a report for publication.
- Separation of these steps not essential.

LEP Imperfection Studies

John Jowett

Packages and functions needed

Setting-up an optics case

Generate the Ideal Machine

Dynamic Aperture of Ideal Machine

Generate Imperfect Machines

Dynamic Apertures of Imperfect Machines

Quantum tracking of imperfect machines

Other calculations on imperfect machines

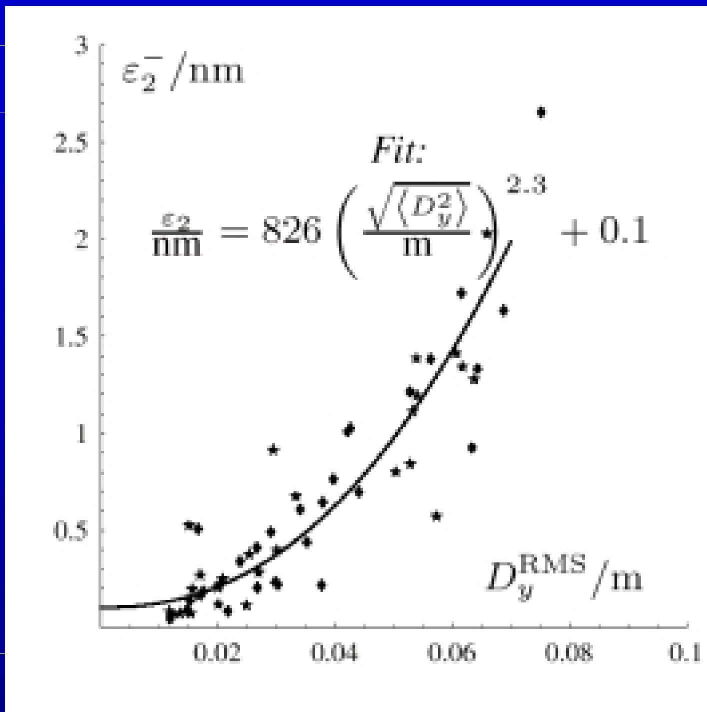
Build database of parameters of imperfect machines

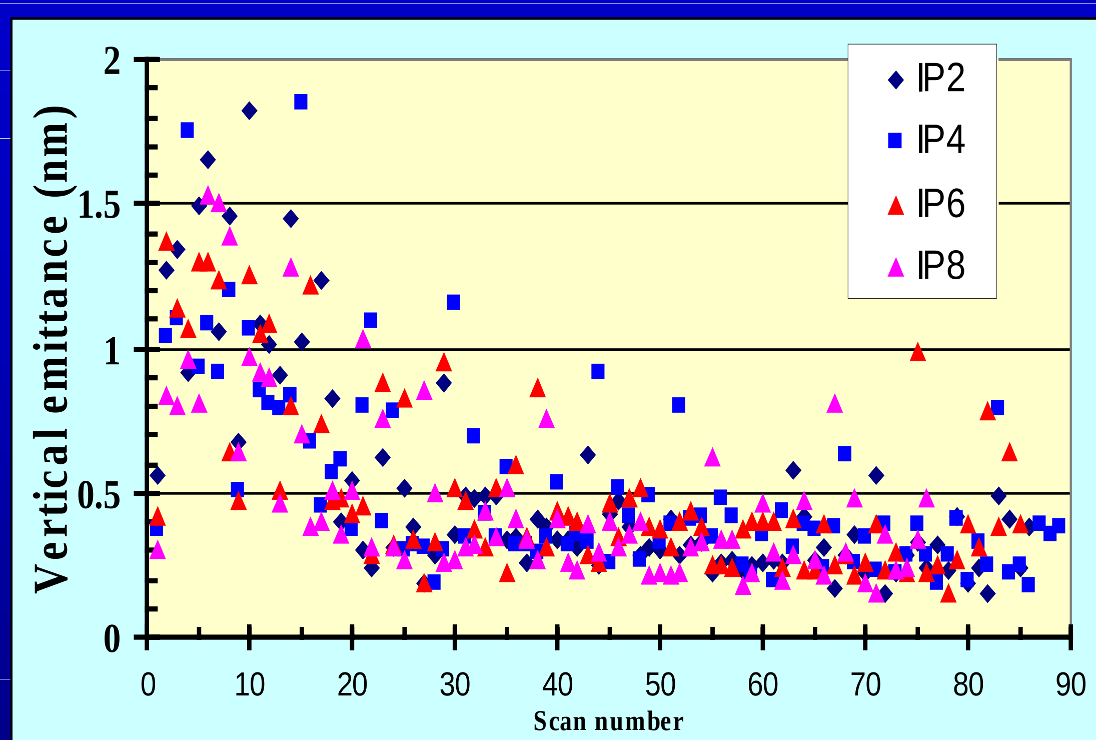
Utilities

Why is this better than writing a program?

- Just like doing mathematics (or theoretical physics).
- Flexibility: Interesting questions vary and evolve.
  - Much better to be able to explore them in an interactive environment even when heavy computation has to be done by a compiled program.
- Relief from planning a "campaign of simulations".
- Programming language is much simpler and vastly more expressive than, say, C++
  - No type declarations
  - No compilation, instant step-by-step debugging

Predicting vertical emittance





Predictions from a simulated ensemble of corrected machines

Measurements from LEP in 1997 (M. Lamont).
Fine-scanning of the vertical separation at the IPs, luminosity variation gives vertical beam size (translated here into emittance).
Vertical beam size at IP actually 3-4 μm

Later developments

- Absorb the limited MAD language in the Mathematica language:
  - Much higher-level programming structures
  - Mathematical/Graphical interfaces
- The new MAD Version 9
  - Complete rewrite of program in C++.
    - Incomplete, currently debugging.
  - Accelerator physics implemented in CLASSIC class library
- Parallelism with multiple Mathematica kernels
  - controlled from a single notebook

Owning up

- I have over-simplified.
- The ideal was not fully realised.
- This is how to do it if starting again.
- I am, however, confident that it is entirely feasible now.

Applicability

- When a problem requires ingredients from a heterogeneous computational environment
- Build set of Mathematica functions that return essential results by running external programs.
  - Externals are, generally, but not necessarily, the CPU-intensive computations.
- Use as a basis for higher-level functions that will exhaustively analyse problem.
- Judicious use of dynamic programming idiom to absorb results of heavy calculations in a functional database.

A Personal Vision

- How we do theoretical/mathematical/ computational science
  - Up to early 20th century
  - Mid 20th century
  - Late 20th century
  - Some time in the 21st century

Up to early 20th century

- Small science
  - One man apprehended the whole problem,
  - Understanding implemented in mathematical formalism with paper and pencil.
  - Pieces of problem solved on paper.
- Published paper was much the same as the work done.
  - Effort required to read it
  - But rather full understanding was gained
  - Not too hard to check the results

Mid 20th century

- Science got bigger
  - Many men got involved in a problem, some of them only apprehending their piece.
  - Pieces of problem solved by computer programs. Running programs became an activity in itself.
- Published paper no longer resembled the work done (*executive summary syndrome*)
  - Less effort to read it.
  - Less understanding conveyed.

Late 20th century

- Big science
  - Done with fewer people
  - Many were still running computer programs all day long
- e-Published papers galore
  - Less time to read them
  - Who really understood what ?
  - Who could check?
- But computer programs started to disappear into something that looked like mathematics again …

Some time in the 21st century

- Published paper will do the work
- Reader will understand with less effort by interaction with the paper