Quality Control

Waldo MacKay

- The paper should be readable both in hard copy form and on the screen.

- Papers should stay within margins.
  - Outline templates on tranparencies make this check easy for the printed copy.

- Type 1 fonts should be used.
  - Bit mapped fonts usually look grainy on the screen and are hard to read.

- Formatting instructions should be followed.

- File size should not be too large.
  - Files should display on screen quickly.
  - Files should print efficiently (not longer than ~20s).
    Efficiency problems are caused by figures.
  - Not everyone has a fast computer with lots of memory.
  - Some large files will not display on all computers.
  - Files will downloaded from the web
  - Files will be downloaded over phone lines with slow modems.

Note: Some flexibility in the rules will be necessary to keep one’s sanity and finish the project in a reasonable amount of time.

File statistics:

PDF files:

Largest file	3.96 Mb
Smallest file	10.5 kb
Average size	171 kb

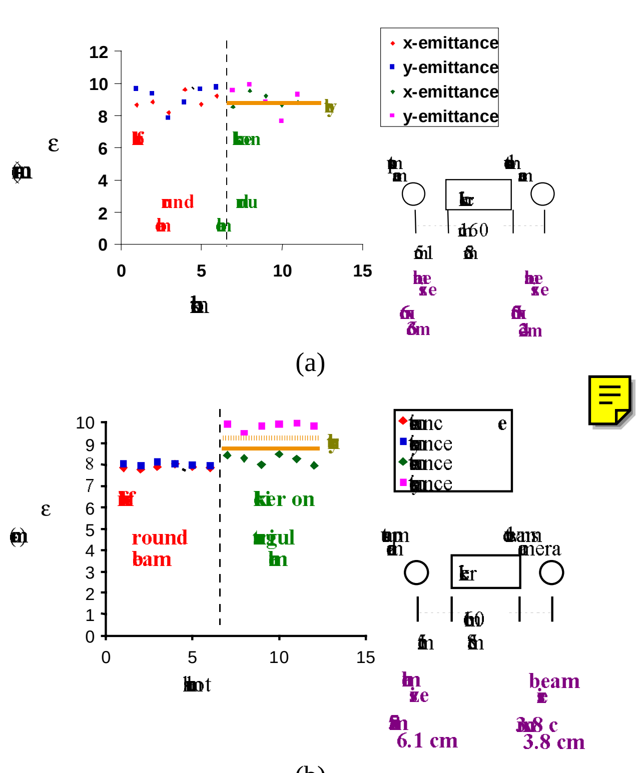
>2 Mb	4 files
1.5→2 Mb	2 files
1.0→1.5 Mb	18 files
0.75→1.0 Mb	16 files
500→750 kb	31 files
250→500 kb	138 files
100→250 kb	348 files
<100 kb	666 files
Total	1223 files

From comments in PostScript files:

MS Word	61%
LaTeX	32%
Unknown	6%
Other	1%
Remaining Type 3 fonts	30%

Fonts (Type 1 fonts required)

- For text:
  - **Adobe** standard fonts from basic 14 fonts:
    - **Times-Roman**
    - **Times-Italic**
    - **Times-Bold**
    - **Times-BoldItalic**
    - **Symbol**
    - **ZapfDingbats**
  - (Should not use MicroSoft’s NewTimes font set)
- For equations:
  - With LaTeX:
    - Type 1 Computer Modern and AMS from BlueSky or Bakoma sets
  - With MSWord:
    - Must have additional Type 1 fonts embedded to PostScript file for special characters
- For figures:
  - Vector graphics should use Type 1 Adobe Fonts from basic set of 14 fonts. Any other fonts should be included in the PostScript file.
  - Bitmap graphics
    - Preferably the above fonts should have been used, but as a minimum, the text should be legible when reduced to final size.
- Beware of font substitutions:
  - Missing fonts get substituted with Courier (as a default).
  - Bad encoding -- wrong characters or symbols.
  - Bad metrics -- overprinting or worse.



(a)

(b)

Fig. 6 Emittance measured downstream of the kicker with and without an applied voltage. (a) A round beam at the output yielded no emittance growth. (b) A larger beam yielded some emittance growth. The *y*-emittance is slightly larger than in *x* which is predicted by theory.

3.4 Kicker control system

A kicker control system that attempts to regulate beam motion with the dynamics of the kicker has been designed and tested on ETA-II. As shown in Fig. 7, the outer loop takes beam position data and calculates a desired pulser voltage waveform. The inner loop then attempts to match actual output of the pulser to the desired waveform.

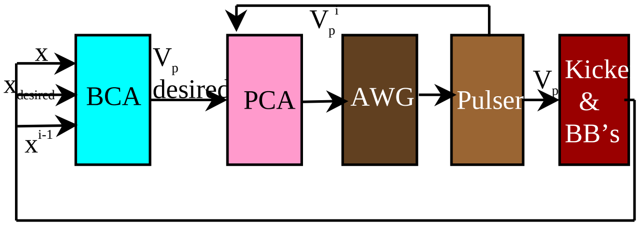


Fig. 7 The beam control algorithm (BCA) takes measured beam location to find the desired voltage needed for the next correction. It then feeds into the pulser control algorithm (PCA) which trys to produce the desired waveform at the output of the pulsers via the arbitrary waveform generators (AWG). The pulses are then sent to the kicker and beam position data is acquired from beam bugs (BB). Each pulser is controlled independently.

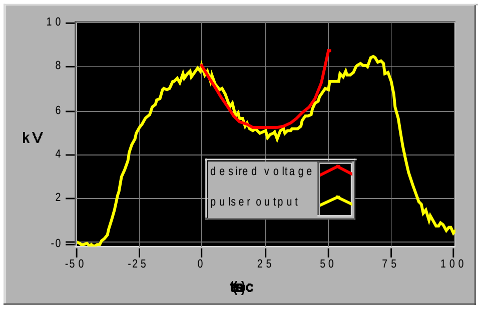


Fig. 8 The desired voltage is achieved at the output of the positive pulser (similar for the negative pulser) to within 2%. Notice the tail end of the waveform has a steep slope that the pulser cannot physically meet.

3.5 The septum magnet and split beam pipe

A septum magnet which is capable of generating two opposing dipole field regions was designed and built [6]. A split beam line and the transport section for two diverging beamlines, one straight-ahead and one at 15° were designed and tested in conjunction with the septum. Results of the experiment have lead us to believe that the aperture of the magnet is too restrictive for a low repetition rate machine such as DARHT. Hence, it is decided that a quad septum will be used in it's place (see fig. 2).

4 CONCLUSION

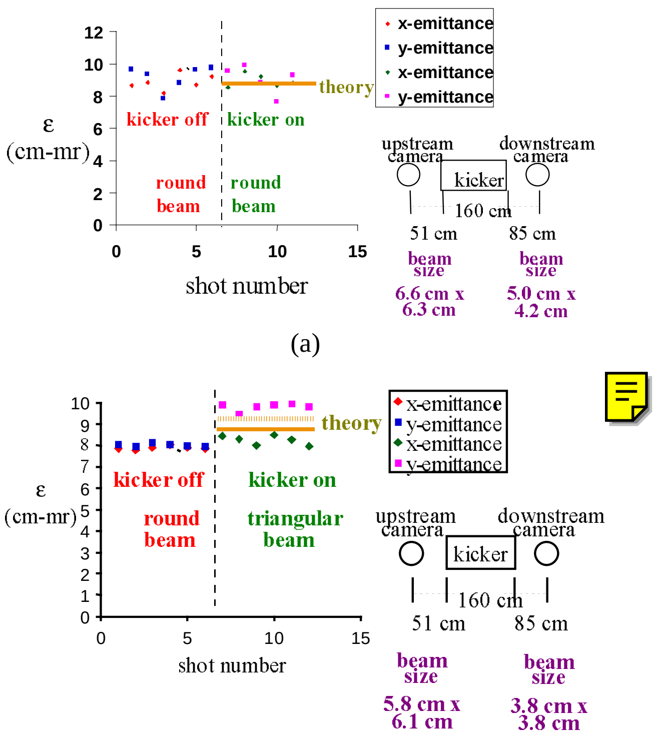
Development of a complete kicker system is well underway. The kicker structure itself and the septum magnet have been designed and tested on ETA-II. The two components are well-characterized. The kicker structure itself has shown success in beam switching and little emittance growth. The control system has been implemented and is undergoing further refinements. A new generation of solid state pulsers is being pursued. Long pulse precision beam position monitors for beam steering throughout the kicker system have been developed and scheduled to be tested on ETA-II later this year [9].

5 ACKNOWLEDGEMENTS

Thanks go to the ETA-II and Bechtel-Nevada staff for their support of the kicker experiments.

6 REFERENCES

[1] Chen, Judy, presented at LINAC98, Chicago, Aug. 1998.
[2] DeHope, W.J. et. al, to be presented at the 12th IEEE Int l. Pulsed Power Conf., Monterey, June 1999.
[3] Caporaso, G.J. et. al, 1997 Particle Accelerator Conference, 1861-3, Vancouver, May 1997.
[4] Chen, Judy et. al, "Fast kicker for high current, continuous beam", *to be submitted, preprint avail. upon request.*
[5] Weir, J.T. et al., this conference.
[6] Poole, B.D., et al., this conference.
[7] Paul, A.C.
[8] Wang, L.F. et al., this conference.
[9] Nelson, S.D. et al., this conference.



(a)

(b)

Fig. 6 Emittance measured downstream of the kicker with and without an applied voltage. (a) A round beam at the output yielded no emittance growth. (b) A larger beam yielded some emittance growth. The *y*-emittance is slightly larger than in *x* which is predicted by theory.

3.4 Kicker control system

A kicker control system that attempts to regulate beam motion with the dynamics of the kicker has been designed and tested on ETA-II. As shown in Fig. 7, the outer loop takes beam position data and calculates a desired pulser voltage waveform. The inner loop then attempts to match actual output of the pulser to the desired waveform.

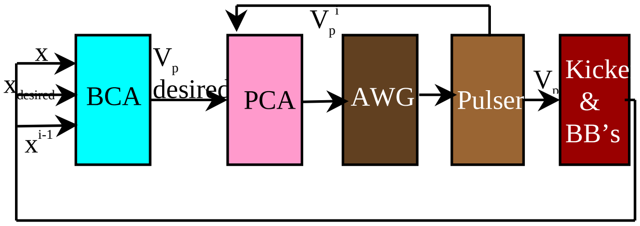


Fig. 7 The beam control algorithm (BCA) takes measured beam location to find the desired voltage needed for the next correction. It then feeds into the pulser control algorithm (PCA) which trys to produce the desired waveform at the output of the pulsers via the arbitrary waveform generators (AWG). The pulses are then sent to the kicker and beam position data is acquired from beam bugs (BB). Each pulser is controlled independently.

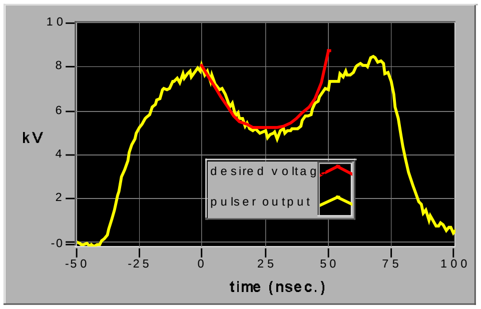


Fig. 8 The desired voltage is achieved at the output of the positive pulser (similar for the negative pulser) to within 2%. Notice the tail end of the waveform has a steep slope that the pulser cannot physically meet.

3.5 The septum magnet and split beam pipe

A septum magnet which is capable of generating two opposing dipole field regions was designed and built [6]. A split beam line and the transport section for two diverging beamlines, one straight-ahead and one at 15° were designed and tested in conjunction with the septum. Results of the experiment have lead us to believe that the aperture of the magnet is too restrictive for a low repetition rate machine such as DARHT. Hence, it is decided that a quad septum will be used in it's place (see fig. 2).

4 CONCLUSION

Development of a complete kicker system is well underway. The kicker structure itself and the septum magnet have been designed and tested on ETA-II. The two components are well-characterized. The kicker structure itself has shown success in beam switching and little emittance growth. The control system has been implemented and is undergoing further refinements. A new generation of solid state pulsers is being pursued. Long pulse precision beam position monitors for beam steering throughout the kicker system have been developed and scheduled to be tested on ETA-II later this year [9].

5 ACKNOWLEDGEMENTS

Thanks go to the ETA-II and Bechtel-Nevada staff for their support of the kicker experiments.

6 REFERENCES

[1] Chen, Judy, presented at **LINAC98**, Chicago, Aug. 1998.
[2] DeHope, W.J. et. al, to be presented at **the 12th IEEE Int'l. Pulsed Power Conf.**, Monterey, June 1999.
[3] Caporaso, G.J. et. al**, 1997 Particle Accelerator Conference**, 1861-3, Vancouver, May 1997.
[4] Chen, Judy et. al, "Fast kicker for high current, continuous beam", *to be submitted, preprint avail. upon request.*
[5] Weir, J.T. et al., this conference.
[6] Poole, B.D., et al., this conference.
[7] Paul, A.C.
[8] Wang, L.F. et al., this conference.
[9] Nelson, S.D. et al., this conference.

"untemplated" as soon as some concrete type is used for its arguments:

```
  int main() {
    SomeType x (2);
    SomeType y(3);
    Swap(x, y);
}
```

If class `SomeType` declares copy constructor and assignment operator private, the main program will not compile. This gives a typical example of template programming: most of templates are built with some assumptions (template constraints) for the template parameters. Note that C++ allows developers to create free template functions, template classes and template member functions in non-template classes.

2.2 Templates, Traits and Map Generation

Now we are ready to explain how we can generate maps, provided that we know how to integrate usual dynamic variables. Imagine that we have class `System` which has a template function `Advance()` for calculation of final value the vector describing the dynamical variables in 6-dimensional space. For simplicity, we will use a function propagating through a simple drift. The argument will be encapsulated in a vector container whose choice depends on a user:

```
  void template <class VecType>
  System::Advance(VecType& z) {
    double pz  = CalcPz(p);
    double lenOverPz = length/pz;
      p[4]+=lenOverPz*
       (InvReferenceBeta()+p[5]) +
       length*(1+Eta()*Delta_S)*
       InvReferenceBeta();
      p[2]+=p[3]*lenOverPz;
      p[0]+=p[1]*lenOverPz;
    }
```

`VecType` can be any of multiple containers for double vectors (`std::vector<double>`, `valaray<double>` etc.). Now we want to generalise this function so that it can treat not only double vectors, but also maps.

To make the receipe described in the previous section work, we have to overload `operator[](int)` in `DAMap` class so that it returns a corresponding DA vector. In addition, we assume that DA maps and vectors are implemented as template classes with type of coefficients being the template parameters, so that we can describe real or complex DA's used for normal form analysis. We also found advantageous to have dimension and order as template parameters, but for simplicity will omit them here. Thus, schematically the `DAMap` class will have the following structure:

```
template <class U> class DAMap {
      DAVector<U>* data;
    public:
      const DAVector<U>& operator[](int i) const
        {return data[i];}
      DAVector<U>& operator[](int i)
        {return data[I];}
    //etc.
};
```

Now, lets go back to generalisation of the `Advance()` function. Instead of `VecType` we now imagine more general template parameter which can include not only vectors, but also maps, since we made sure that `DAMap::operator[](int)` is defined. The first question coming to mind is what should we put instead of type declaration "`double`"? Evidently, this type should be the same as a type of variable returned by `operator [] (int)`. In case of a double vector, it should be `double`, in case of a DA map, this should be a DA vector! If we want to use one generalised `Advance()` function for both dynamic vectors and DA maps, how do we code this information so it works for both?

The solution is in using traits (see [4]). Traits are special classes, which provide mechanism to associate certain functions, values or types with particular classes and access them in a uniform way. For example, we often need to know what is the type of data is contained by the vector? In case of `std::vector<T>` it will be `T`. In case of `DAMap<U>`, it should be a `DAVector<U>`, since any map is just a vector of DA vectors. Similarly, we might want to know a size of the vector, which will be function `size()` for `std::vector<T>` and `DAMap<U>::Dimension()` for DA maps. This information can be encapsulated in the following traits classes. First we define a general template class which, being unspecialised, does not have anything:

```
  template <class U> struct UnaryTraits {};
```

Now we can partially specialise it for vectors and maps:

```
  template <class U>
  UnaryTraits <std::vector<U> > {
  //Define the type of data
    typedef U ValueType;

  //Get size of the vector
    static int GetSize(const std::vector<U>& u) {
       return u.size()}
};

template <class U>
  UnaryTraits <DAMap<U> > {
  //Define the type of data
    typedef DAVector<U> ValueType;

  //Get size of the vector
    static int GetSize(const DAMap<U>& u) {
      return u.Dimension();}
};
```

With traits defined, we can rewrite our `Advance()` function in such a way that it will treat both vectors of

Template Format:

Titles:

14pt Bold uppercase only. (Authors frequently ignore the uppercase requirement.)

Abstract Heading: 12pt Italic "*Abstract*", left justified.

Section Headings: 12pt Bold uppercase only and centered.

Subsection Headings: 12pt Italic mixed case and left justified.

Main text: 10pt Times-Roman.

References:

References are allowed to be in 8pt font.

Equations:

Equation numbers right justified in parentheses.

Recommend that format of equations should follow good formatting rules as used in textbooks (e.g. John Wiley, Academic Press, ...)
(Knuth's *TeX Book* treats good practices in depth.)

Symbols for units should be written in Times-Roman font, not Italic.

Figure Issues:

Poincaré and other point plots can be quite large.

- May need to be reprocessed from original files.
- Can edit **eps** file directly in some cases to reduce file size. (This can be very tedious.)
- Can convert to a bit map format, but may lose resolution -- particularly in labels.
- Can use a paint program on the bit map to erase labels -- replace labels either with paint program or by adding PostScript to over print labels on top of bit map.
- Similar tricks may be necessary with plots from some canned software such as MAFIA field plots and 3-d mesh generators.

Bit map figures do not always convert well into encapsulated PostScript files.

- *The LaTeX Graphics Companion* by Goosens et al. gives a lot of useful information for dealing with different formats.
- The *netpbm* package of software contains a lot of conversion programs for various formats.
  - Available for *nix platforms.
  - Comes with MikTeX for Windows platforms.
- Photos in jpeg format.
  - **jpeg2ps** can (sometimes) put a **eps** wrapper around a **jpeg** file. If it works the file usually distill nicely.

Bit maps do not always display well even though they print well.

- If you can zoom in to see the detail, then it is OK.
- Some figures will be illegible even in the original PostScript.

MAFIA

```
FRAME:    7            11/01/99 - 10:48:04            VERSION[V4.010]
```

```
SPTMP.DRC
```

```
FREQUENCY/HZ                       5.2381888000000E+08
MAXIMUM ERROR OF    CURLCURL-E     3.1319301575422E-02
MEAN     ERROR OF    CURLCURL-E     6.7151611438021E-04
MAXIMUM ERROR OF DIVERGENCE-D      1.5181150274657E-06
```

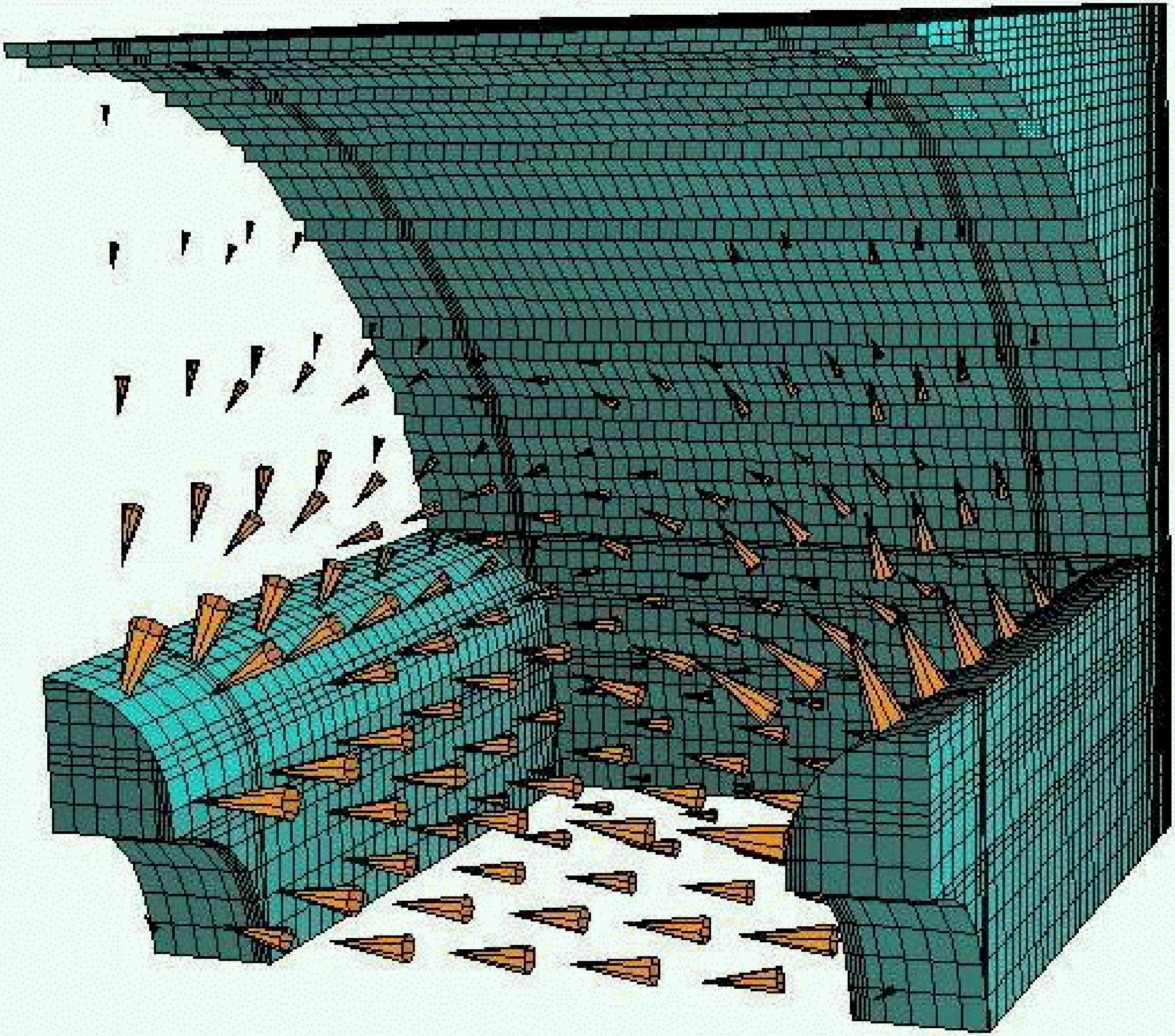
```
SPOKE LADER-BAR SC MODEL CAVITY      SPROUND.CAV
BETA=.5    FREQ = 500 MHZ    ROUND SPOKES
RCAV=.158 M  ZBAR=.3*ZCELL   APP=.03   BAR=.045
TIME HARMONIC ELECTRIC FIELD IN V/M
```

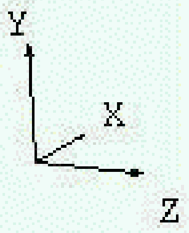
P--:4010

#3DARROW

```
COORDINATES/M
FULL RANGE / WINDOW
X[      .0000,    .16000]
 [      .0000,    .16000]
Y[      .0000,    .16000]
 [      .0000,    .16000]
Z[      .0000,    .16000]
 [      .0000,    .16000]
```

```
SYMBOL: E_1
TIME......:  3.00000E-03
MAX. ARROW.: 4.821E+01
INTERPOLATE=    1
LOGSCALE...= 0.000E+00
MATERIALS: 1,
```





Tables:

MS Word tables with the grid displayed usually exceed the margins.

(This obnoxious behavior seems to be the default with Word.)

Usually **Pitstop** can be used to grab the table and slide it towards inside the margins.

Processing steps for QA:

Before adding page numbers

1. Display **pdf** file with exchange.
2. Check display speed. Verify display speed is OK.
3. Check display quality.
4. Print file. Verify speed is OK.
5. Check margins in hard copy.
6. Compare text of printed copy with author's submitted paper copy.
    - Pay attention to special symbols in text and equations. (A familiarity with mathematical expressions helps.)
    - Check for font substitution and metric problems. (These can occur in figures.)

After adding page numbers:

1. Check display quality.
2. Check margins in hard copy.
3. Compare text of printed copy with author's submitted paper copy.

    Note: We only had to put page numbers manually in about 4 out of 1223 papers at PAC'99.
4. Check page numbers and any additional headers/footers.

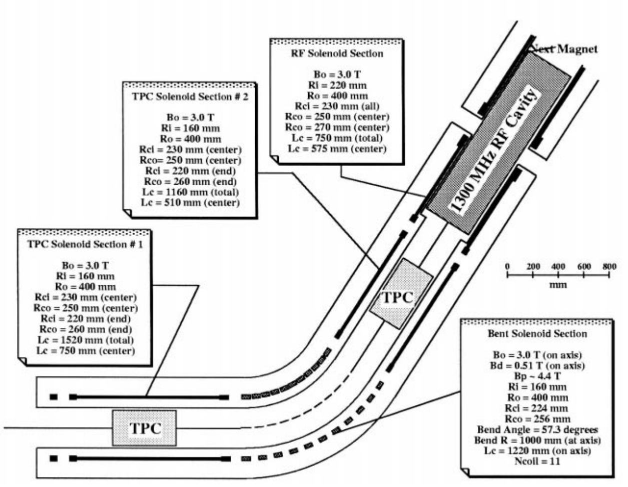


Figure 16: Emittance diagnostics via a bent solenoid spectrometer.

6 REFERENCES

[1] http://www.cap.bnl.gov/mumu/status_report.html

[2] http://www.cap.bnl.gov/mumu/book.html

[3] AIP Conf. Proc. **352** (1995).

[4] AIP Conf. Proc. **372** (1996).

[5] AIP Conf. Proc. **435** (1998).

[6] AIP Conf. Proc. **441** (1998).

[7] G.K. O'Neill, Phys. Rev. **102**, 1418 (1956).

[8] Yu.M. Ado and V.I. Balbekov, Sov. Atomic Energy **31**, 731 (1971).

[9] G.I. Budker, see AIP Conf. Proc. **352**, 4 (1996).

[10] A.N. Skrinsky, see AIP Conf. Proc. **352**, 6 (1996).

[11] http://xxx.lanl.gov/ps/hep-ex/9801004

[12] http://nicewww.cern.ch/˜autin/MuonsAtCERN/Neutrino.htm

[13] http://ftp.pac99.bnl.gov/Papers/Wpac/MOP86.pdf

[14] http://ftp.pac99.bnl.gov/Papers/Wpac/MOP98.pdf

[15] http://ftp.pac99.bnl.gov/Papers/Wpac/TUA147.pdf

[16] http://ftp.pac99.bnl.gov/Papers/Wpac/TUP101.pdf

[17] http://ftp.pac99.bnl.gov/Papers/Wpac/TUP129.pdf

[18] http://ftp.pac99.bnl.gov/Papers/Wpac/TUP154.pdf

[19] http://ftp.pac99.bnl.gov/Papers/Wpac/WEA163.pdf

[20] http://ftp.pac99.bnl.gov/Papers/Wpac/WEBR4.pdf

[21] http://ftp.pac99.bnl.gov/Papers/Wpac/WEBR5.pdf

[22] http://ftp.pac99.bnl.gov/Papers/Wpac/WEBR6.pdf

[23] http://ftp.pac99.bnl.gov/Papers/Wpac/WEP118.pdf

[24] http://ftp.pac99.bnl.gov/Papers/Wpac/THA130.pdf

[25] http://ftp.pac99.bnl.gov/Papers/Wpac/THP31.pdf

[26] http://ftp.pac99.bnl.gov/Papers/Wpac/THP32.pdf

[27] http://ftp.pac99.bnl.gov/Papers/Wpac/THP33.pdf

[28] http://ftp.pac99.bnl.gov/Papers/Wpac/THP34.pdf

[29] http://ftp.pac99.bnl.gov/Papers/Wpac/THP35.pdf

[30] http://ftp.pac99.bnl.gov/Papers/Wpac/THP36.pdf

[31] http://ftp.pac99.bnl.gov/Papers/Wpac/THP37.pdf

[32] http://ftp.pac99.bnl.gov/Papers/Wpac/THP38.pdf

[33] http://ftp.pac99.bnl.gov/Papers/Wpac/THP39.pdf

[34] http://ftp.pac99.bnl.gov/Papers/Wpac/THP40.pdf

[35] http://ftp.pac99.bnl.gov/Papers/Wpac/THP41.pdf

[36] http://ftp.pac99.bnl.gov/Papers/Wpac/THP42.pdf

[37] http://ftp.pac99.bnl.gov/Papers/Wpac/THP44.pdf

[38] http://ftp.pac99.bnl.gov/Papers/Wpac/THP44.pdf

[39] http://ftp.pac99.bnl.gov/Papers/Wpac/THP45.pdf

[40] http://ftp.pac99.bnl.gov/Papers/Wpac/THP46.pdf

[41] http://ftp.pac99.bnl.gov/Papers/Wpac/THP47.pdf

[42] http://ftp.pac99.bnl.gov/Papers/Wpac/THP48.pdf

[43] http://ftp.pac99.bnl.gov/Papers/Wpac/THP49.pdf

[44] http://ftp.pac99.bnl.gov/Papers/Wpac/THP50.pdf

[45] http://ftp.pac99.bnl.gov/Papers/Wpac/THP51.pdf

[46] http://ftp.pac99.bnl.gov/Papers/Wpac/THP52.pdf

[47] http://ftp.pac99.bnl.gov/Papers/Wpac/THP53.pdf

[48] http://ftp.pac99.bnl.gov/Papers/Wpac/THP54.pdf

[49] http://ftp.pac99.bnl.gov/Papers/Wpac/THP55.pdf

[50] http://ftp.pac99.bnl.gov/Papers/Wpac/THP56.pdf

[51] http://ftp.pac99.bnl.gov/Papers/Wpac/THP57.pdf

[52] http://ftp.pac99.bnl.gov/Papers/Wpac/THP58.pdf

[53] http://ftp.pac99.bnl.gov/Papers/Wpac/THP59.pdf

[54] http://ftp.pac99.bnl.gov/Papers/Wpac/THP60.pdf

[55] http://ftp.pac99.bnl.gov/Papers/Wpac/THP83.pdf

[56] http://ftp.pac99.bnl.gov/Papers/Wpac/THP85.pdf

[57] http://ftp.pac99.bnl.gov/Papers/Wpac/THP86.pdf

[58] http://puhep1.princeton.edu/mumu/target/targetprop.ps

[59] http://puhep1.princeton.edu/mumu/target/

[60] http://www.fnal.gov/projects/muon_collider/

[61] http://www.fnal.gov/projects/muon_collider/cool/cool.html