ADAPTIVE THREE-DIMENSIONAL RMS ENVELOPE SIMULATION IN THE SAD ACCELERATOR MODELING ENVIRONMENT*

C.K. Allen#, LANL, Los Alamos, NM 87545, U.S.A.
K. Furukawa, M. Ikegami, K. Oide, KEK, Tsukuba, Ibaraki 305-0801, Japan

Abstract

The capability for three-dimensional RMS envelope simulation, including space charge, has been implemented in the SAD (for Strategic Accelerator Design) [5] accelerator modeling environment used at KEK. The dynamics within the model are similar to that used by Trace3D [3] and TRANSPORT [1]. Specifically, the matrix of all second-order beam moments is propagated using a linear beam optics model for the beamline. However, the current simulation employs an adaptive space-charge algorithm. It maintains the integration step size as large as possible while enforcing a given error tolerance. We concentrate on the adaptive nature of the RMS simulation, since this is the novel feature.

BACKGROUND

The complete set of phase space coordinates for a beam particle, including both position and normalized momentum, at axis location s is given by $(x,x',y,y',z,\delta) \in \Re^6$. For conciseness, it is convenient to denote these points in phase space by a vector quantity $\mathbf{z}$; we have

$$\mathbf{z} \equiv (x, x', y, y', z, \delta) \in \Re^6 . \tag{1}$$

Consider now a distribution of particles forming a three-dimensional beam bunch and assume further that the distribution can be described by an s-dependent functional $f : \Re^6 \times \Re_+ \to \Re$ on phase space. Denote by $\langle\cdot\rangle : \Re^6 \to \Re$ the phase space moment operator with respect to f. That is, for any (possibly s-dependent) function g on phase space $\Re^6$ we have $\langle g\rangle \equiv (1/Q)\int g(\mathbf{z};s)f(\mathbf{z};s)d^6\mathbf{z}$ where $Q \equiv \int f(\mathbf{z};s)d^6\mathbf{z}$ is the total charge of the bunch.

The phase space coordinates are themselves functions on phase space and we may take their moments. The vector $\bar{\mathbf{z}} \equiv \langle\mathbf{z}\rangle$ contains all the first-order moments of the distribution f and its value is the centroid of the beam at location s. The symmetric, positive-definite matrix

$$\boldsymbol{\tau} \equiv \left\langle \mathbf{z}\mathbf{z}^T \right\rangle \tag{2}$$

contains all the second moments of f and is known as the *correlation matrix* of the distribution. The matrix $\boldsymbol{\tau}(s)$ describes the second-order evolution of f. It is our independent variable. Another quantity often seen in the literature is the *covariance matrix* $\boldsymbol{\sigma} \equiv \langle(\mathbf{z}-\bar{\mathbf{z}})(\mathbf{z}-\bar{\mathbf{z}})^T\rangle = \boldsymbol{\tau} - \bar{\mathbf{z}}\bar{\mathbf{z}}^T$. It is the matrix of central, second moments of f and analogous to the standard deviation, its univariate counterpart. The matrix $\boldsymbol{\sigma}$ describes the width of f and is the origin of the term "RMS envelopes". For the case of a centered beam where $\bar{\mathbf{z}} = \mathbf{0}$, we have $\boldsymbol{\sigma} = \boldsymbol{\tau}$, the case most often treated in the literature.

* Work supported by KEK under a foreign visiting scientist grant.
#ckallen@lanl.gov

Beam Dynamics

We use a linear beam optics model for the beamline where each beamline element n may be represented as a transfer matrix $\boldsymbol{\Phi}_n$. We let $\boldsymbol{\tau}_n$ denote the correlation matrix at the entrance to element n and $\boldsymbol{\tau}_{n+1}$ denote the correlation matrix at its exit. Then the dynamics equations for $\{\boldsymbol{\tau}_n\}$ are given by [3]

$$\boldsymbol{\tau}_{n+1} = \boldsymbol{\Phi}_{n,sc}\boldsymbol{\tau}_n\boldsymbol{\Phi}_{n,sc}^T , \tag{3}$$

where $\boldsymbol{\Phi}_{n,sc}$ is the transfer matrix for element n including space charge. It has the form

$$\boldsymbol{\Phi}_{n,sc} = \exp\left(L_n\mathbf{G}_n + L_n\mathbf{G}_{sc}\right), \tag{4}$$

where L_n is the length of element n, $\mathbf{G}_n$ is the generator matrix representing external forces exerted by element n, and $\mathbf{G}_{sc}$ is the generator matrix representing the internal forces of the beam. Equation (4) requires the assumption that both $\mathbf{G}_n$ and $\mathbf{G}_{sc}$ are constant. This is often the case for the matrix $\mathbf{G}_n$, but seldom true for the space charge matrix $\mathbf{G}_{sc}$; it depends strongly upon the correlation matrix $\boldsymbol{\tau}$. Thus, if we are to employ Eq. (3) in an accurate numerical algorithm, we are forced to step a smaller distance $h < L_n$, then recompute $\mathbf{G}_{sc} = \mathbf{G}_{sc}(\boldsymbol{\tau})$ as necessary.

Model Quantities

Our algorithm propagates through beamline element n in steps of varying lengths $h \le L_n$. For each beamline element n, assume that we are given separate transfer matrices $\boldsymbol{\Phi}_n(h)$ for the element without space charge and a transfer matrix $\boldsymbol{\Phi}_{sc}(h)$ for space charge alone, rather than the full transfer matrix with space charge $\boldsymbol{\Phi}_{n,sc}(h)$. Thus, $\boldsymbol{\Phi}_n(h)$ and $\boldsymbol{\Phi}_{sc}(h)$ are the partial transfer matrices through n for a distance h.

From the two separate transfer matrices $\boldsymbol{\Phi}_n(h)$ and $\boldsymbol{\Phi}_{sc}(h)$, we need to obtain the full transfer matrix $\boldsymbol{\Phi}_{n,sc}(h)$. We must first accept that we will not get an exact value for $\boldsymbol{\Phi}_{n,sc}(h)$. The transfer matrix $\boldsymbol{\Phi}_{sc}(h)$ is computed under the assumption that $\mathbf{G}_{sc}$ is constant over h, which it is not. The larger the value of h, the less accurate the value of $\boldsymbol{\Phi}_{sc}(h)$. There is no point in computing $\boldsymbol{\Phi}_{n,sc}(h)$ to high accuracy, since we are always limited by the length of h. We employ one particularly convenient fact,

$$\mathbf{\Phi}_{n,sc}(h) = \mathbf{\Phi}_{sc}(h/2)\mathbf{\Phi}_n(h)\mathbf{\Phi}_{sc}(h/2) + O(h^3)\,. \tag{5}$$

This formula gives us a transfer matrix $\mathbf{\Phi}_{n,sc}(h)$ from $\mathbf{\Phi}_n(h)$ and $\mathbf{\Phi}_{sc}(h)$ which is second-order accurate in h. Equation (5) can be proven with a direct application of the Zassenhaus formula for the product of matrix exponentials [7].

Computation of the Transfer Matrices

SAD provides the transfer matrix $\mathbf{\Phi}_n$ for each element n, but not the partial transfer matrix $\mathbf{\Phi}_n(h)$ that we require for space charge effects. We must first determine the generator matrix $\mathbf{G}_n = (1/L_n)\log(\mathbf{\Phi}_n)$ then compute $\mathbf{\Phi}_n(h) = \exp(h\mathbf{G}_n)$. The matrix logarithm is computed by an iterative technique based upon the fact that $\exp(-\mathbf{A}_i)\mathbf{\Phi}_n \to \mathbf{I}$ as $\mathbf{A}_i \to \log(\mathbf{\Phi}_n)$ [4]. This technique is efficient because repeated computation of the matrix exponential is still faster than direct computation of the matrix logarithm. It works well for when $\mathbf{\Phi}_n$ is symplectic, but convergence often fails for the general case.

The process for computing the $\mathbf{\Phi}_{sc}(h)$ is covered elsewhere in the literature (e.g., see [3]). However, we note some points of interest. Inferring from Eq. (4) we find $\mathbf{\Phi}_{sc}(h) = \exp(h\mathbf{G}_{sc})$. Letting $\mathbf{G}_{sc}{}^0$ denote the space charge generator matrix in the beam frame, then $\mathbf{G}_{sc} = \mathbf{L}_0{}^{-1}\mathbf{R}_0{}^{-1}\mathbf{G}_{sc}{}^0\mathbf{R}_0\mathbf{L}_0$ where $\mathbf{L}_0$ is the Lorentz transform to the rest frame and $\mathbf{R}_0$ is a rotation that aligns the beam ellipsoid to the coordinate axes. In this context $\mathbf{G}_{sc}{}^0$ is idempotent, that is, $(\mathbf{G}_{sc}{}^0)^2 = \mathbf{0}$. Since $\exp(\mathbf{L}_0{}^{-1}\mathbf{R}_0{}^{-1}\mathbf{G}_{sc}{}^0\mathbf{R}_0\mathbf{L}_0) = \mathbf{L}_0{}^{-1}\mathbf{R}_0{}^{-1}\exp(\mathbf{G}_{sc}{}^0)\mathbf{R}_0\mathbf{L}_0$, we have

$$\mathbf{\Phi}_{sc}(h) = \mathbf{L}_0^{-1}\mathbf{R}_0^T\left(\mathbf{I} + h\mathbf{G}_{sc}^0\right)\mathbf{R}_0\mathbf{L}_0\,, \tag{6}$$

from the Taylor expansion of the matrix exponential and the fact that $\mathbf{R}_0{}^T = \mathbf{R}_0{}^{-1}$. Application of (6) substantially reduces the computation cost of determining $\mathbf{\Phi}_{sc}(h)$.

PROPAGATION ALGORITHM

In many existing RMS envelope simulation codes the step size h is chosen *a priori*, typically as an integer divisor of L_n. There is no way of insuring solution accuracy in that case. In contrast we choose our step length h dynamically, depending upon an error criterion. The technique below picks the largest h while maintaining solution precision. It is based upon the procedure described in Press *et. al* [6].

Within a given beamline element n and with a given step size h we define our stepping operator $S_h : \Re^{6\times6} \to \Re^{6\times6}$ for the correlation matrix $\boldsymbol{\tau}$. Let $\mathbf{\Phi}(h) \equiv \mathbf{\Phi}_{sc}(h/2)\mathbf{\Phi}_n(h)\mathbf{\Phi}_{sc}(h/2)$ and define define S_h according to

$$S_h(\boldsymbol{\tau}) \equiv \mathbf{\Phi}(h)\boldsymbol{\tau}\mathbf{\Phi}^T(h)\,. \tag{7}$$

From Eq. (3) we see that S_h steps the correlation matrix $\boldsymbol{\tau}(s)$ at a position s within element n to position $s + h$. Let $\boldsymbol{\tau}(s+h)$ denote the exact solution of $\boldsymbol{\tau}$ for an advance from s to $s + h$. By Eqs. (3) and (5), $S_h[\boldsymbol{\tau}(s)]$ produces a second-order approximation to $\boldsymbol{\tau}(s+h)$, or more formally

$$S_h(\boldsymbol{\tau}(s)) = \boldsymbol{\tau}(s+h) + h^3\mathbf{C}\,, \tag{8}$$

where $\mathbf{C} \in \Re^{6\times6}$ is a constant matrix given by (an abuse of) Taylor's theorem $\mathbf{C} = \boldsymbol{\tau}'''(\hat{s})/3!$ for some $\hat{s} \in [s,s+h]$. The term $h^3\mathbf{C}$ is the remainder for the stepping process and our objective is to control the magnitude of this value. To do so, let $\|\cdot\|$ be any matrix norm on $\Re^{6\times6}$ then define

$$\varepsilon(h) \equiv \left\|h^3\mathbf{C}\right\| = h^3\left\|\mathbf{C}\right\|. \tag{9}$$

We recognize $\varepsilon(h)$ as the residual error in our approximation of $\boldsymbol{\tau}(s+h)$. Assume that we are given an *a priori* constraint on this error, say $\bar{\varepsilon}$. Then at each step i, our objective is to find the largest h_i such that $\varepsilon(h_i) \le \bar{\varepsilon}$. We accomplished this objective through step doubling.

Let $\boldsymbol{\tau}^1(s+2h) \equiv \mathbf{S}_{2h}(\boldsymbol{\tau}(s))$ be the result of taking one step of length $2h$ and $\boldsymbol{\tau}^2(s+2h) \equiv S_h[S_h(\boldsymbol{\tau}(s))]$ be the result of taking two steps of length h. Then we have

$$\begin{aligned}\boldsymbol{\tau}^1(s+2h) &= \boldsymbol{\tau}(s+2h) + (2h)^3\mathbf{C}\,,\\ \boldsymbol{\tau}^2(s+2h) &= \boldsymbol{\tau}(s+2h) + 2(h^3\mathbf{C})\,,\end{aligned} \tag{10}$$

Let $\boldsymbol{\Delta}(h) \equiv \boldsymbol{\tau}^1(s+2h) - \boldsymbol{\tau}^2(s+2h)$ so

$$\left\|\boldsymbol{\Delta}(h)\right\| = 6h^3\left\|\mathbf{C}\right\| = 6\varepsilon(h)\,. \tag{11}$$

Then the ratio of $\|\boldsymbol{\Delta}(h)\|$ for two potentially differing steps sizes h_i and h_{i+1} is given by

$$\left\|\boldsymbol{\Delta}(h_i)\right\| / \left\|\boldsymbol{\Delta}(h_{i+1})\right\| = h_i^3 / h_{i+1}^3\,. \tag{12}$$

This relation is the foundation for generating step lengths. Assume we are given a step h_i and we wish the next step h_{i+1} to maintain the error $\bar{\varepsilon}$, that is $\varepsilon(h_{i+1}) = \bar{\varepsilon}$. By Eq. (11) we have $\|\boldsymbol{\Delta}(h_{i+1})\| = 6\varepsilon(h_{i+1}) = 6\bar{\varepsilon}$. Substituting this into the above and rearranging yields the desired result,

$$h_{i+1} = h_i\left[6\bar{\varepsilon} / \left\|\Delta(h_i)\right\|\right]^{1/3}. \tag{13}$$

Interpretation of formula (13) goes as follows: although we used a length h_i to step $\boldsymbol{\tau}$ from s to $s + 2h_i$, we could have used a length of size h_{i+1} and still remained within error constraint $\bar{\varepsilon}$. Since there is no point in recomputing $\boldsymbol{\tau}$ for the different h_{i+1} (we are already at location $s + 2h_i$), the implication is we should try a step h_{i+1} for the next iteration.

There are a couple of additional points to note here before concluding the section. The most important is that if we find h_{i+1} to be less than h_i, we must roll back the

computation and re-step $\boldsymbol{\tau}$ using the smaller step size h_{i+1}. This procedure is necessary because the condition $h_{i+1} < h_i$ implies we have violated our error constraint $\varepsilon(h_i) \le \bar{\varepsilon}$. Another point of practical concern is that we should not change the step length if formula (13) suggests a very small change in h_{i+1}. If h_{i+1} is less than h_i by only a few percent, it may not be worth the trouble of recomputing $\boldsymbol{\tau}$ for a small gain in accuracy. Thus, we provide the caveat that, given the small "slackness parameter" δ_h, we actually update the next step value h_{i+1} only if $|h_{i+1} - h_i|/h_i \ge \delta_h$.

Finally, we note an additional procedure to the algorithm which provides a potentially modest gain in solution accuracy. Referring back to Eqs. (10), we can subtract four times the second equation from the first to yield $\boldsymbol{\tau} = (4/3)\boldsymbol{\tau}^2 - (1/3)\boldsymbol{\tau}^1 + O(h^4)$. Use of this formula is known as *internal extrapolation*. However, we cannot monitor its accuracy. Although it is higher order, it may not be higher accurate, we have no way of knowing. Its use seldom does harm.

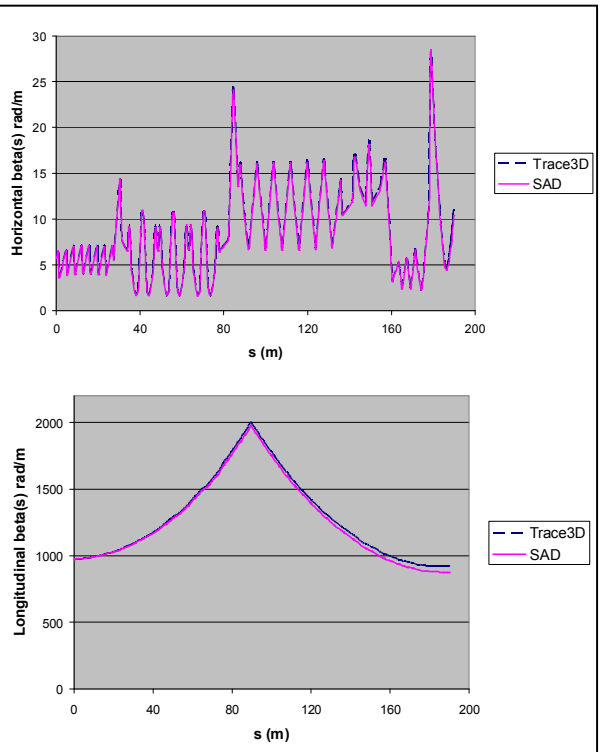

Figure 1: simulation results for SAD versus Trace3D

SIMULATION RESULTS

An RMS simulation module was developed for the SAD accelerator modeling environment based upon the principles here. The module was implemented primarily in SADScript [5], a scripting language similar in syntax to Mathematica [8].

Figure 1 compares the results of the SAD RMS envelope simulation and Trace3D for the case of the Japan Proton Accelerator Research Complex (J-PARC) at Tokai, Japan. The beamline being modeled is the transport section between the linear accelerator and the 3 GeV synchrotron. An H^- beam enters the transport at 181 MeV and 30 mA. The two plots in the figure show the horizontal and longitudinal β functions, respectively. In both simulations we have chosen $\bar{\varepsilon} = 10^{-5}$, $\delta_h = 0.05$, initial step size $h_0 = 3$ cm, and used the l_1 matrix norm. Trace3D uses a constant step size of $h = 1$ cm. In Figure 1 we see very good agreement in the horizontal plane and a small, but noticeable discrepancy in the longitudinal. Trace3D appears to produce more space-charge effect in the longitudinal direction. SAD uses a symplectic technique for generating the transfer matrices whereas Trace3D computes $\boldsymbol{\Phi}_n$ from the equations of motion, but the nature of this discrepancy is still unknown.

SUMMARY

We have described an adaptive stepping algorithm for propagating the correlation matrix $\boldsymbol{\tau}$ through beamline elements in the presence of space charge. It is based upon the discrete transfer equations (3) for $\boldsymbol{\tau}$ and the adaptive step sizing formula of (13). At each iteration i the use of formula (13) keeps our step size h_i as large as possible, however, it does require some overhead. We must compute three applications of Eq. (7) for a single iteration. Yet each iterate actually propagates the $\boldsymbol{\tau}$ a distance $2h_i$ and, consequently, must be compared to two single applications of Eq. (7). The resulting computational overhead is 150% of a single-step algorithm, plus the computation of the norm $\|\boldsymbol{\Delta}(h_i)\|$ and any roll-back incurred. Thus, the worst-case scenario takes at least 1.5 times the non-adaptive approach (h_i constant). What we gain from the non-adaptive approach is the guarantee of solution accuracy $\bar{\varepsilon}$. Moreover, we are also guaranteed a self-consistency space-charge calculation. In most cases, we expect a computational advantage of adaptive stepping. Considering the overall advantages contrasted with the small amount of additional code development, the adaptive stepping process appears as a clear benefit in RMS envelope simulation.

REFERENCES

[1] K.L. Brown and R. Rothacker, SLAC-91, Rev. 3, UC-28 (1983).

[2] D.C. Carey, *The Optics of Charged Particle Beams* (Harwood Academic Publishers, London, 1987).

[3] K.R. Crandall and D.P. Rusthoi, LANL Report LA-UR-97-886.

[4] This technique was suggested by E. Forest.

[5] K. Oide, http://acc-physics.kek.jp/SAD/sad.html

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, 2nd Ed. (Cambridge University Press, Cambridge, 1992), pp. 463-469.

[7] D.H. Sattinger and O.L. Weaver, *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics* (Springer-Verlag, New York, 1986), Chap. 3, p. 36.

[8] S. Wolfram, http://www.wolfram.com.