The BBSOLVE User's Manual version 1.0

M. Blaskiewicz*
BNL, Upton NY 11973, USA

The Fortran program BBSOLVE numerically solves the Balbekov/Burov equation for single bunch transverse instability. It may be used for any particle type. Forces from space charge, resistive wall and multiple resonators are included. This note describes the algorithms and their implementation.

I. INTRODUCTION AND THEORY

Coherent instabilities are of significant concern for a wide variety of planned and existing accelerators. The theory of these phenomena has been advancing steadily for decades [1–14]. Recently, an ordinary differential equation has been developed by Burov [28] and refined by Balbekov [29, 30]. To derive this equation we begin by considering the equation for the evolution of the transverse centroid as a function of position, momentum, and time,

$$\left(\frac{\partial}{\partial t}+\omega_s u\frac{\partial}{\partial\theta}-\omega_s\theta\frac{\partial}{\partial u}\right)^2 X(\theta,u,t)+Q^2\Omega^2 X=2\Omega_0^2 Q_0\Delta Q(\theta)\left\{X(\theta,u,t)-\bar{X}(\theta,t)\right\}+\frac{G(\theta,t)}{\gamma m}. \tag{1}$$

In equation (1) t is time and Ω_0 is the revolution frequency of a synchronous particle. The longitudinal coordinate is θ with $s/R=\Omega_0 t-\theta$, with s the longitudinal Serret-Frenet coordinate. This is the opposite sign convention from [30] for ease of numerics later. The synchrotron frequency is ω_s and the momentum like variable is $u=(1/\omega_s)d\theta/dt$. The revolution frequency and tune for a given value of u are Ω and Q. The space charge tune shift as a function of longitudinal postion is $\Delta Q(\theta)\geq 0$ and the force due to wakefields is $G(\theta,t)$. Both these require the average postion along the bunch $\bar{X}(\theta,t)$. To obtain $\bar{X}$ we need the longitudinal phase space density $F(\theta,u)$ defined so that $Fd\theta du$ is the fraction of the beam in $d\theta du$. The normalized line density is $\rho(\theta)=\int duF(\theta,u)$ with $\int\rho(\theta)d\theta=1$. Then

$$\bar{X}(\theta,t)=\frac{\int duX(\theta,u,t)F(\theta,u)}{\rho(\theta)}, \tag{2}$$

where the u integral includes everywhere F is not zero.

To proceed one sets

$$X(\theta,u,t)=Y(\theta,u)\exp(-i\Omega_c t+i\chi\theta) \tag{3}$$

where $\Omega_c\approx Q_0\Omega_0$, and

$$\chi=\frac{\frac{d}{du}(\Omega Q)}{\frac{d}{du}\Omega}. \tag{4}$$

Next one takes $\Omega_x(u)=\Omega Q$ and approximates

$$d^2/dt^2+\Omega_x^2\approx(d/dt-i\Omega_x)(d/dt+i\Omega_x)\approx-2i\Omega_x(d/dt+i\Omega_x)\approx-2iQ_0\Omega_0(d/dt+i\Omega_0Q_0-iu\omega_s\chi).$$

Inserting all this in equation (1) and simplifying coefficients yields

$$\tilde{\omega}Y(\theta,u)+iu\frac{\partial Y}{\partial\theta}-i\theta\frac{\partial Y}{\partial u}=-\lambda(\theta)(Y-\bar{Y})+G(\theta). \tag{5}$$

where $\lambda(\theta)=\Omega_0\Delta Q(\theta)/\omega_s$, $\tilde{\omega}=(\Omega_c-\Omega_0Q_0)/\omega_s$ and

$$G(\theta)=\int_{-\hat{\theta}}^{\theta}d\theta_1\tilde{W}(\theta-\theta_1)\rho(\theta_1)\bar{Y}(\theta_1). \tag{6}$$

*Electronic address: blaskiewicz@bnl.gov

In equation (6) the half bunch length is $\hat{\theta}$ and

$$\tilde{W}(\theta-\theta_1)=\frac{NqR}{8\pi^2(E_T/q)\beta^2 Q_0 Q_s}W([\theta-\theta_1]/\Omega_0)e^{i(Q_c-\chi)[\theta-\theta_1]}, \tag{7}$$

where there are N particles of charge q in a machine of radius R. The total energy per particle is E_T with $\beta=v/c$. The betatron and synchrotron tunes are Q_0 and Q_s, respectively. Equation (5) is equation (17) in [30] except that my variable is θ (the negative of his θ). The reason for my choice of variables is that $G(-\hat{\theta})=0$ and numerical integration can proceed from $-\hat{\theta}$ with a positive integration step.

Next one approximates

$$Y(\theta,u)\approx y_0(\theta)+uy_1(\theta)+u^2y_2(\theta) \tag{8}$$

Note that

$$\bar{Y}(\theta)=y_0(\theta)+U^2(\theta)y_2(\theta) \tag{9}$$

with

$$U^2(\theta)=\frac{\int u^2F(\theta,u)du}{\int F(\theta,u)du}$$

Inserting (8) in (5) and equating powers of u yields 3 equations

$$\tilde{\omega}y_0-i\theta y_1-\lambda(\theta)y_2U^2-G=0 \tag{10}$$

$$\tilde{\omega}y_1+i\frac{dy_0}{d\theta}+\lambda y_1=2i\theta y_2\approx 0 \tag{11}$$

$$\tilde{\omega}y_2+i\frac{dy_1}{d\theta}+\lambda y_2=0 \tag{12}$$

To proceed set the right hand side of (11) to zero so that (11 and (12)) become

$$y_2=\frac{-iy_1'}{\lambda+\tilde{\omega}},\qquad y_1=\frac{-iy_0'}{\lambda+\tilde{\omega}},$$

where ' denotes differentiation with respect to θ. Inserting these into (10) yields

$$\tilde{\omega}y_0-\theta\frac{y_0'}{\lambda+\tilde{\omega}}=-\frac{\lambda U^2}{\lambda+\tilde{\omega}}\left(\frac{y_0'}{\lambda+\tilde{\omega}}\right)'+G. \tag{13}$$

Notice that equation (13) is in terms of y_0, not $\bar{Y}$ that appears in the expression for G. Using (9)

$$y_0=\bar{Y}-U^2y_2\approx\bar{Y}+\frac{U^2}{\lambda+\tilde{\omega}}\left(\frac{\bar{Y}'}{\lambda+\tilde{\omega}}\right)' \tag{14}$$

directly inserting this expression into (13) would yield a 4rth order ordinary differential equation. Balbekov argues that only the term proportional to $\tilde{\omega}$ needs expression (14) and the other instances can use $\bar{Y}\approx y_0$. This is because we are considering the case $\lambda(\theta)\gg 1$ which is true when $\Delta Q_{sc}\gg Q_s$. Making this approximation gives

$$\tilde{\omega}\bar{Y}+U^2\left(\frac{\bar{Y}'}{\lambda+\tilde{\omega}}\right)'-\theta\frac{\bar{Y}'}{\lambda+\tilde{\omega}}=G \tag{15}$$

with $G(\theta)$ given by equation (6) we have an integro-differential system for $\bar{Y}$. This equation is close to equation (20) in [30]. To make it exactly equal to Balbekov's result divide the right hand side of (15) by $1+\tilde{\omega}/\lambda$.

We now limit discussion to the case

$$\lambda(\theta)=\frac{\Delta Q_{sc}}{Q_s}(1-\theta^2/\hat{\theta}^2)^{\alpha}, \tag{16}$$

where $\Delta Q_{sc} > 0$ is the maximum space charge tune shift, Q_s is the synchrotron tune, and the bunch lies within $|\theta| < \hat{\theta}$. Note that while we define $\Delta Q_{sc} > 0$ the actual incoherent particle tunes are always depressed by the space charge force. The line density is $\rho(\theta) \propto (\hat{\theta}^2 - \theta^2)^\alpha$ and

$$U_\alpha^2(\theta) = \frac{\hat{\theta}^2 - \theta^2}{2\alpha + 2}.$$

With these functions equation (15) has regular singlular points at $\pm\hat{\theta}$. One of the allowed solutions is finite with continuous first and second derivatives and the other is singular. On physical grounds we reject the singular result and assume $\bar{Y}(\hat{\theta})$ and $\bar{Y}(-\hat{\theta})$ are finite.

For numerical purposes it is useful to define two first order differential equations

$$\frac{d}{d\theta}\bar{Y}(\theta) = (\lambda(\theta) + \tilde{\omega})Z(\theta) \tag{17}$$

$$U^2(\theta)\frac{d}{d\theta}Z(\theta) = \theta Z(\theta) + \tilde{\omega}\bar{Y}(\theta) + G(\theta) \tag{18}$$

Since Z' is finite at $\pm\hat{\theta}$ the boundary conditions are

$$0 = -G(-\hat{\theta}) = -\hat{\theta}Z(-\hat{\theta}) + \tilde{\omega}\bar{Y}(-\hat{\theta}) \tag{19}$$

$$0 = \hat{\theta}Z(\hat{\theta}) + \tilde{\omega}\bar{Y}(\hat{\theta}) + G(\hat{\theta}) \equiv E(\tilde{\omega}). \tag{20}$$

The numerical scheme is to assume an initial value of $\tilde{\omega}$. Then take $\bar{Y}(-\hat{\theta}) = 1$ and $Z(-\hat{\theta}) = \tilde{\omega}/\hat{\theta}$. Next we numerically integrate (17) and (18) to $\theta = \hat{\theta}$. In general $E(\tilde{\omega}) \neq 0$ because some of the singular solution will be present. Next we numerically compute $\partial\mathrm{Re}(E)/\partial\mathrm{Re}(\tilde{\omega})$, and $\partial\mathrm{Re}(E)/\partial\mathrm{Im}(\tilde{\omega})$ and the same for $\mathrm{Im}(E)$, yielding a 2 by 2 matrix. Given the value of E and its derivatives a linear solution is used to give an improved estimate of $\tilde{\omega}$. By starting on a grid of values of $\tilde{\omega}$ we can be confident that all the eigenvalues in a given range are found.

Since $U^2(\pm\hat{\theta}) = 0$ some care is needed in the numerical scheme. The problem starts by taking a uniform grid $\theta_n = n\Delta - \hat{\theta}$, where $n = 0, 1, \ldots N-1$ and defining $Z(\theta_n) = Z_n$ etc. Equations (17) and (18) become

$$\bar{Y}_{n+1} - \bar{Y}_n = (\Delta/2)\left[(\lambda(\theta_{n+1/2}) + \tilde{\omega})(Z_n + Z_{n+1})\right] \tag{21}$$

$$U^2(\theta_{n+1/2})(Z_{n+1} - Z_n) = (\Delta/2)\left[(\theta_{n+1/2})(Z_n + Z_{n+1}) + \tilde{\omega}(\bar{Y}_n + \bar{Y}_{n+1}) + 2G_n\right]. \tag{22}$$

The numerical approximation for G_n is the simplest possible

$$G_n = \sum_{k=1}^{n} \Delta\tilde{W}((n-k)\Delta)\rho(k\Delta)\bar{Y}_k. \tag{23}$$

Given (21) and (22) one obtains $\bar{Y}_{n+1}$ and Z_{n+1} from previous values. The error term is take as $E(\tilde{\omega}) = \hat{\theta}Z_N + \tilde{\omega}\bar{Y}_N + G_N$. One calculates $E(\tilde{\omega})$, $E(\tilde{\omega} + \delta\omega)$ $E(\tilde{\omega} + i\delta\omega)$ where $\delta\omega$ is initially an input parameter. After calculating the 2 by 2 matrix and $\delta\tilde{\omega}$ one compares $|\delta\tilde{\omega}|$ to another input parameter $|\delta\tilde{\omega}|_{max}$ one takes $\delta\tilde{\omega}_1 = \delta\tilde{\omega}\min(1, |\delta\tilde{\omega}|_{max}/|\delta\tilde{\omega}|)$, which limits the step size. One then takes $\tilde{\omega} \to \tilde{\omega} + \delta\tilde{\omega}_1$ and $\delta\omega \to \min(\delta\omega, |\delta\tilde{\omega}_1|)$, and iterates until $|E|$ is below some prescried initial value E_{thresh}. 500 steps are allowed for convergence, which is plenty if E_{thresh} is within the limits of numerical precision. After writing the eigenvalues and eigenvectors a new initial value of $\tilde{\omega}$ is chosen and the process repeats until all the initially prescribed values of $\tilde{\omega}$ are exhausted.

II. USING THE CODE

The program `bbsolve` uses the files `bbsolve.f` and the include file `bbsolve_c.f`. It compiles using the intel fortran compiler with line extensions in double precision with the command `ifort -132 -r8 -o bbsolve bbsolve.f`. The executable is `bbsolve`. The program is controlled by the input file `bbsolve.in`. An example input file is

```
22.8 3833. 40. -300.e3 360  gammat,circ,gamma0,vrf,nharm
28.1  0. 3.7e11 1. 1. 3. 6.e-9   tunex,chrom,pnumber,aatom,qatom,power,tauhat
3833. 72.e-8 0.036 0.   slenx,rhoe,bpipe,wstep
100.e6   scimped
```

```
1000 5 1 .5 .5 -1. npt,nr,ni,dqr,dqi,dqr0
1.e-6 1.e-4 .01  thresh00,thresh10,dtunemax0
2  npoley
0  -1.e14          2.e8     2.e9   wr,wi,sr,si
-1.e12    0          1.e8     0
```

The inputs used by the code are just the numbers. The characters to the right are comments to remind the user what's what. We will go through this input file line by line. The first line of actual input is `22.8 3833.  40.  -300.e3 360`. The value of the transition gamma gammat= 22.8 The circumference of the acclerator is circ=3833. meters. The Lorentz factor for the beam is gamma= 40. The rf voltage amplitude is 300 kV and being above transition a negative value is needed for stability vrf=-300.e3 Volts. The second line is the betatron tune 28.1=tunex, the chromaticity 0. = chrom, the number of ions per bunch 3.7e11=pnumber, the atomic mass aatom=1., the atomic number qatom=1., the value of α in equation (15) is 3.0=power, and $\hat{\theta}/\Omega_0$ = tauhat = $6.E-9$ s. Line 3 defines the resistive wall impedance and allows for a step function impedance. The resistive wall wake assume a length of pipe slenx=3833 meters, with resistivity rhoe=72.e-8 Ohm-meters, and pipe radius bpipe=0.036 meters. The value of the step wake is step = 0. volts/coulomb/meter. Line 4 is the space charge impedance scimped = 100.e6 Ohm/meter. Line 5 gives the number of incremetns in θ, npt=1000 and defines the initial values of $\tilde{\omega}$. The initial values for the real part are `dqr0 + k*dqr` for k=0,1,..nr. The initial values for the imaginary part is `j*dqi` for j = -ni,..,0,..ni. All j,k pairs are used. The 6th line of the input file has thresh00 = 1.e-6 as the initial value for the numerical derivative of E, $\delta\omega$ described just after equation (22). The value thresh01=1.e-4 is E_{thresh}, the error allowed when evaluating the boundary condition at $\theta = \hat{\theta}$. The value dtunemax0=0.01 is the maximum step size, $|\delta\tilde{\omega}|_{max}$. This value is smaller than needed but makes the output files clearer later on. The next line gives the number of resonator poles to add to the impedance, npoley=2. The input file concludes with npole=2 lines. The wake due to poles is

$$W_{pole}(t) = Re\left(\sum_{1}^{npole} w_k \exp(-s_k t)\right) \tag{24}$$

where $w_1 = -1.e14i$, $s_1 = 2.e8 + i2.e9$, $w_2 = -1.e12$, $s_2 = 1.e8$ The units for w_k are volts per coulomb per meter. The units for s_k are s^{-1}. To run the code I find it best to create a main directory with the executable and then sub directories for individual runs. In a subdirectory you need the input file and then type `../bbsolve`. You will see

```
$ ../bbsolve
 eta =   1.298668821175746E-003
 tunes =   7.715702877976159E-004
 head tail tune shift =   0.000000000000000E+000
 dqsc/qs =    64.4815537294324
$
```

where I've assumed the prompt is \$. The frequency slip factor is $eta = 1/gammat^2 - 1/gamma0^2$, tunes is the synchrotron tune, and head tail tune shift is $2\hat{\theta} * chrom/eta$ where $chrom = pdQ/dp$ is the un-normalized chromaticity (Q' in MAD). The ratio of the space charge tune shift in the center of the bunch to the synchrotron tune is dqsc/qs. There are 4 output files. Each row of `fort.70` contains $\theta_k, \mathrm{Re}(\tilde{W}(\theta_k)), \mathrm{Im}(\tilde{W}(\theta_k))$. The file `fort.21` gives the path of $\tilde{\omega}$ during iteration. It has 4 columns. Columns 1 and 2 are the real and imaginary parts of $\tilde{\omega}$ as the iteration proceeds. There are blank lines between different starting values of $\tilde{\omega}$ so you can plot it with lines using gnuplot. The third column of fort.21 gives the number of steps used in the iteration, beware if it's 500! The last column of `fort.21` is the value of $|E\tilde{\omega}|$. The output file `fort.66` has the initial and final values of $\tilde{\omega}$ when the iteration converges. These are the only values to trust. Columns 1 and 2 are the real and imaginary parts of the initial values. Columns 3 and 4 are the real and imaginary parts of the final values. A useful gnuplot plot command is

```
gnuplot> p 'fort.21' u 1:2, 'fort.66' u 3:4 w p 3
```

This command plots the points followed by all iterations in point type 1 (usually red) and the final points where convergence occured in point type 3 (blue). Figure 1 shows the figure generated by the above command. The final file is `fort.44`. This file contains the eigenvectors for eigenvalues that converged. There are 5 columns. The first is just the values of θ. The second and third columns are the real and imaginary parts of $\bar{Y}$. The fourth and fifth columns are the converged eigenvalues for the eigenvector. A useful plot is

```
gnuplot> p 'fort.44' u 1:2 w l, '' u 1:3 w l
```

This command overplots the real and imaginary parts of all the eigenvectors. Figure 2 shows the result of this gnuplot command.

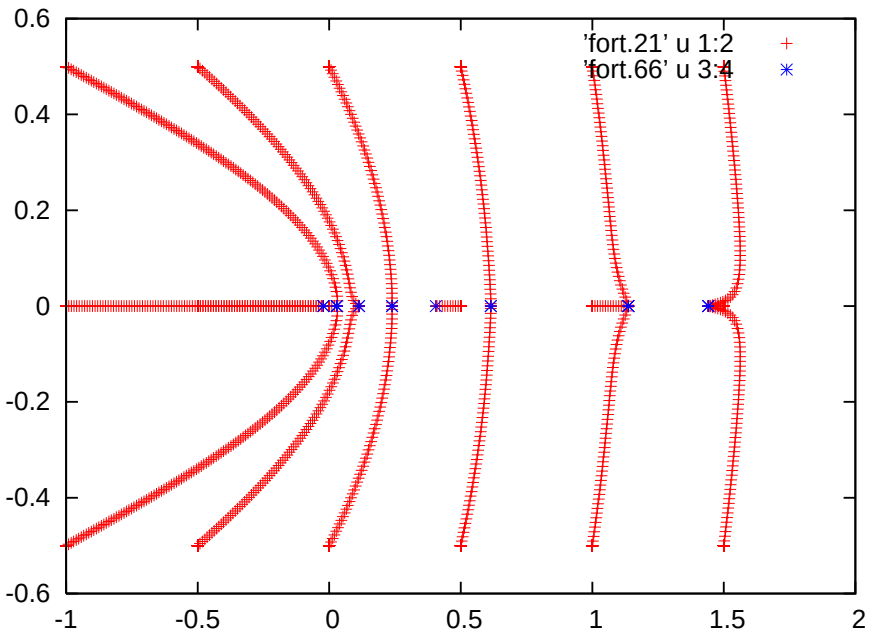

FIG. 1: Paths of iteration in $\tilde{\omega}$

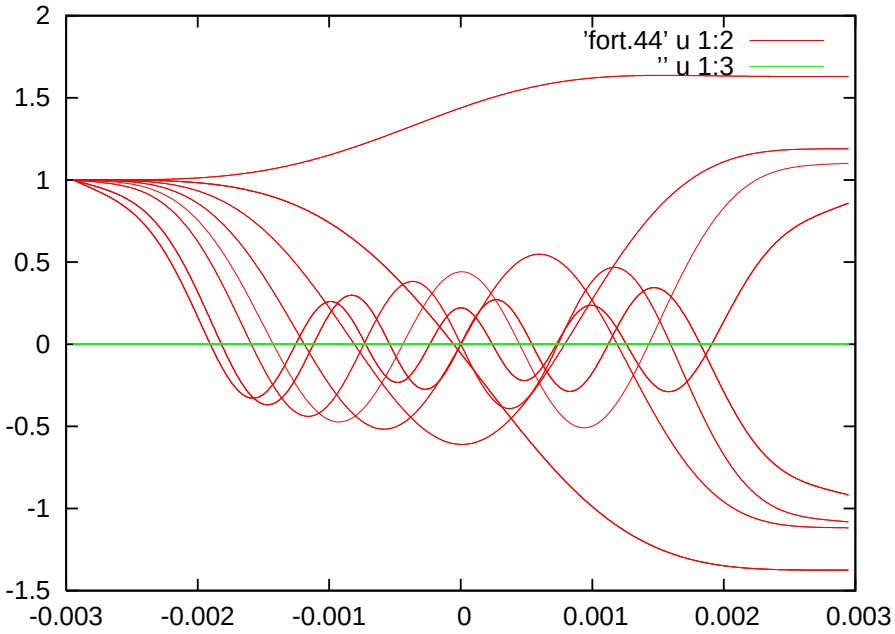

FIG. 2: Converged eigenvectors.

III. DIFFICULTIES AND CONCERNS

Sometimes the results analogous to Figure 1 show points of concentration for the paths (red) but fewer converged values (blue). In this case try

```
gnuplot set logscale y
gnuplot> p 'fort.21' u 4 w l
```

Figure 3 shows the results for our output file. Remember that one of the solutions to our differential equations is well behaved and the other is singular. It is only the nonzero value of $U^2(\hat{\theta} - \Delta/2)$ that keeps our value of E from going

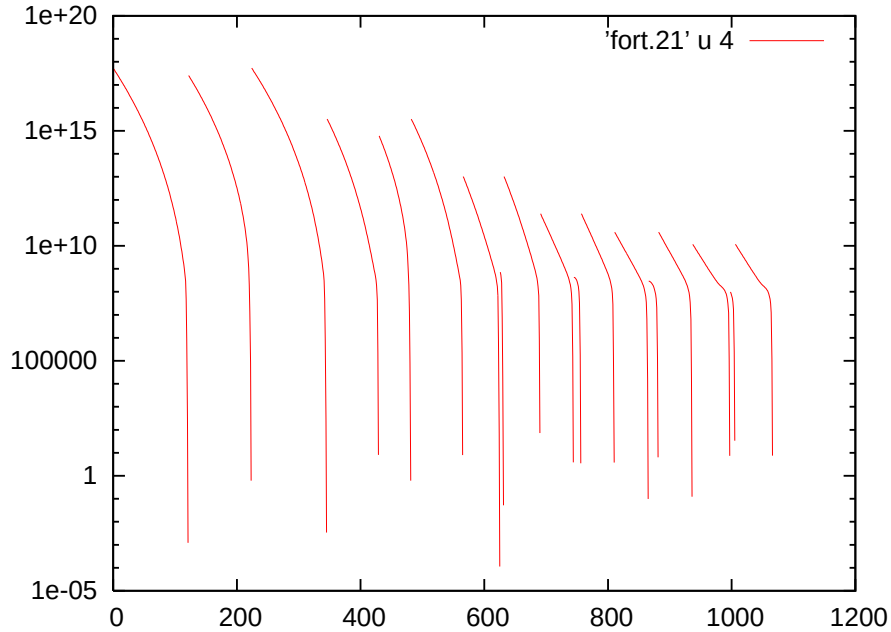

FIG. 3: Values of $|E|$ during iteration. All the iterations are plotted, one after the other.

to infinity for incorrect eigenvalues. Hence you may find that rather large values of `thresh10` are needed for things to converge.

This code is very new and there is a good chance of bugs. Any cross checks you can perform will improve things. I am particularly concerned that I might have messed up the coefficient in equation(7). The code is rather short, not even 300 lines. I would appreciate a few more pairs of eyes looking for problems!

IV. GETTING FREQUENCIES FOR MULTI-LINE PLOTS

The program `getfreq` reads `fort.66`, assumes that all tunes within 10^{-9} of each other are the same and writes out the results ordered with respect to $Re\tilde{\omega}$. It is useful for making multi-line plots.

[1] F.J. Sacherer, CERN/SI-BR/72-5, (1972).
[2] F.J. Sacherer, IEEE-TNS **NS-24**, # 3, p1393, (1977).
[3] A.W. Chao & C.Y. Yao, PEP-Note 321, (1979).
[4] J.L. Laclare, CERN 87-03, p 264 (1987).
[5] J.M. Wang, BNL 39057, (1986).
[6] A.W. Chao *Physics of Collective Beam Instabilities in High Energy Accelerators* Wiley, (1993).
[7] Y. Chin, K. Satoh & K. Yokoya, *Particle Accelerators* **13** , p45 (1983).
[8] Y. Chin, NIMA , **215** p501, (1983)
[9] K. Satoh & Y. Chin, *Nuclear Instruments and Methods* **207**, p309 (1983).
[10] G. Besnier, D. Brandt & B. Zotter, *Particle Accelerators* **17**, p51 (1985).
[11] S. Krinsky & J.M. Wang *Particle Accelerators* **17** , p109 (1985).
[12] Y. Chin, CERN SPS/85-2 (1985).
[13] K. Oide & K. Yokoya, KEK 90-10 (1990).
[14] M. Blaskiewicz, W.T. Weng PRE **50** p4030, (1994)
[15] P. Kernel, R. Nagaoka, J.-L. Revol, & G. Besnier p1133, EPAC00.
[16] R.H. Siemann, AIP Conf 127, p368, (1983).
[17] R.L. Gluckstern, J. van Zeijts, B. Zotter, PRE **47** # 1, p656 (1993).

[18] K. Yokoya *Particle Accelerators* **41** , p221 (1993).
[19] A. Burov & V. Danilov, PRL **82** , #11 p2286 (1999).
[20] R.D. Kohaupt DESY-85-139 (1985).
[21] J.S. Berg, CERN SL 97-72 (AP) (1997).
[22] Shane Rupert Koscielniak, PhD Thesis, University of Oxford, (1987).
[23] R.W. Hockney, J.W. Eastwood *Computer Simulation using Particles* Hilger, Bristol, (1988).
[24] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery *Numerical Recipes in Fortran 77* Cambridge University Press (1992).
[25] J. Wei *PhD thesis* State University of New York at Stony Brook (1990).
[26] A.W. Chao, M. Tigner *eds Handbook of Accelerator Physics and Engineering* World Scientific (1999).
[27] M. Blaskiewicz PRSTAB **4**, 044202 (2001).
[28] A. Burov PRSTAB, **12**,044202, (2009); **12**,044202, (2009)
[29] V. Balbekov PRSTAB, **12**,124402, (2009)
[30] V. Balbekov PRSTAB, **14**,094401, (2011)