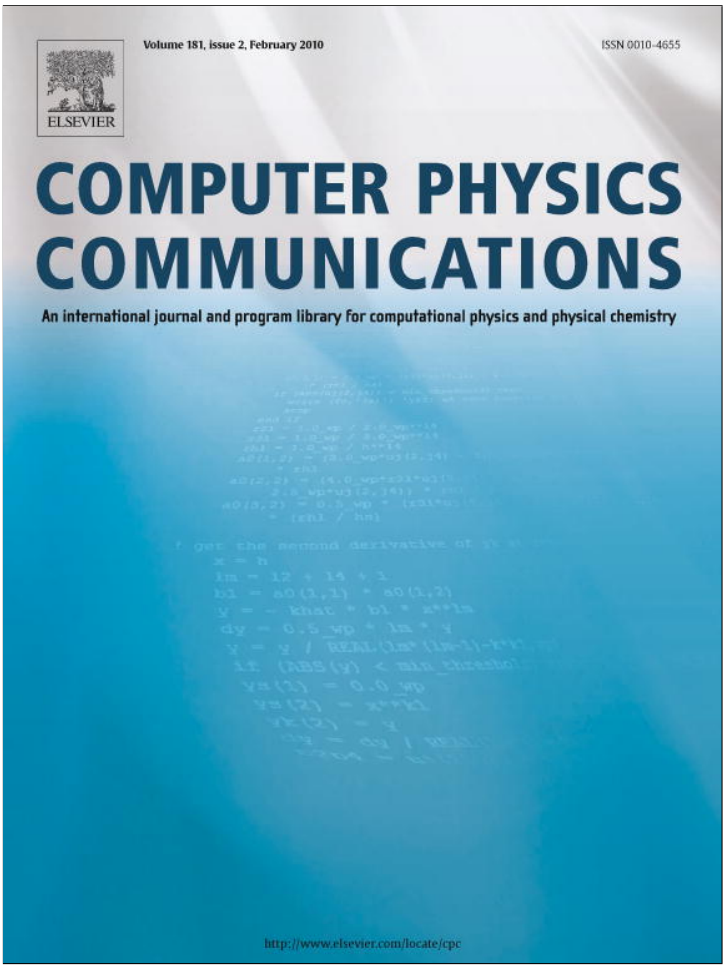

Computer Physics Communications 181 (2010) 313–316

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

A high-order fast method for computing convolution integral with smooth kernel

Ji Qiang

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

ARTICLE INFO

Article history:
Received 20 August 2009
Received in revised form 28 September 2009
Accepted 8 October 2009
Available online 9 October 2009

Keywords:
Convolution integral
Green's function
Simpson rule
FFT

ABSTRACT

In this paper we report on a high-order fast method to numerically calculate convolution integral with smooth non-periodic kernel. This method is based on the Newton–Cotes quadrature rule for the integral approximation and an FFT method for discrete summation. The method can have an arbitrarily high-order accuracy in principle depending on the number of points used in the integral approximation and a computational cost of $O(N\log(N))$, where N is the number of grid points. For a three-point Simpson rule approximation, the method has an accuracy of $O(h^4)$, where h is the size of the computational grid. Applications of the Simpson rule based algorithm to the calculation of a one-dimensional continuous Gauss transform and to the calculation of a two-dimensional electric field from a charged beam are also presented.

1. Introduction

Convolution has been used in solving some linear differential equations such as the Poisson equation, the Helmoltz equation, and the heat transfer equation based on the Green's function method under appropriate boundary conditions or initial conditions [1,2]. For example, the convolution between the Green's function of the Poisson equation and the density function has been used to obtain potential field under open boundary conditions in plasma physics, accelerator physics and cosmology simulations [3–5]. The direct numerical calculation of the convolution for potential field has a computational cost scaling as $O(N^2)$, where N is the number of grid points in the domain. Fortunately, the discretized convolution summation on an uniform grid can be calculated using a cyclic summation on a doubled computational domain using an FFT based method [5–7]. This reduces the computational cost from $O(N^2)$ to $O(N\log(N))$. However, the direct numerical convolution of the density function and the Green's function on the grid is equivalent to a relatively low-order quadrature rule (trapezoidal rule) approximation to the integral. In this paper, we propose approximating the convolution integral using the Newton–Cotes quadrature rule that can have arbitrarily high-order accuracy in principle. By defining a new density function on the grid, the resulting discrete summation can be calculated using the FFT based method. To be specific, in this paper, we will present the algorithm based on the Simpson rule approximation of the convolution integral. This algorithm has the accuracy $O(h^4)$ with a computational cost $O(N\log(N))$.

The organization of the paper is as follows: after introduction, we will present a one-dimensional algorithm in Section 2, a two-dimensional Simpson rule based algorithm in Section 3, and final discussions in Section 4.

2. One-dimensional convolution

In general, a one-dimensional convolution integral can be written as

$$I(x) = \int_0^L G(x-x')\rho(x')\,dx', \tag{1}$$

where the function G can be related to the Green's function of a differential equation, and the ρ can be associated with charge density or initial temperature. The above integral can be written as the summation of M equal subinterval integrals:

$$I(x) = \sum_{i=1}^{M} \int_{x'_i}^{x'_{i+1}} G(x-x')\rho(x')\,dx', \tag{2}$$

where $x'_i = (i-1)(L/M)$, $i = 1,\ldots,M$. For the integral between x'_i and x'_{i+1}, the closed Newton–Cotes formula of degree k at $k+1$ equally spaced points can be written as

$$\int_{x'_i}^{x'_{i+1}} G(x-x')\rho(x')\,dx' \approx \sum_{j=0}^{k} w_j G(x-x'_j)\rho(x'_j), \tag{3}$$

where $x'_j = x'_i + j\,dx$, $dx = L/(kM)$, and weights w_j are associated with integral of the jth Lagrange basis polynomial. For $k = 1$, this

E-mail address: JQiang@lbl.gov.

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2009.10.005

is known as the trapezoidal quadrature rule; for $k = 2$, this is the Simpson rule; for $k = 3$, Simpson's 3/8 rule; for $k = 4$, Boole's rule, etc. [8,9]. Substituting the above subinterval integral approximation into the summation equation (2), we obtain the extended Newton–Cotes quadrature rule approximation to the convolution integral as:

$$I(x) \approx \sum_{l=0}^{kM} G\left(x - x'_l\right)\rho\left(x'_l\right)W_l, \tag{4}$$

where $x'_l = l\,dx$; $W_l = w_j$, $j = \mathrm{mod}(l,k)$, $j \neq 0$; $W_l = w_0 + w_k$, $\mathrm{mod}(l,k) = 0$, $l \neq 0$, $l \neq kM$; $W_l = w_0$, $\mathrm{mod}(l,k) = 0$, $l = 0$; $W_l = w_k$, $\mathrm{mod}(l,k) = 0$, $l = kM$; and function $\mathrm{mod}(l,k) = l - k * \mathrm{int}(l/k)$. Defining a new density function $\bar{\rho}(x'_l) = \rho(x'_l)W_l$, the above discrete convolution summation at each point x_l can be written as

$$\bar{I}(x_l) = \sum_{l=0}^{kM} G\left(x_l - x'_l\right)\bar{\rho}\left(x'_l\right). \tag{5}$$

This discrete convolution can be calculated efficiently using the FFT based method. In the following, we will give a 4th-order accuracy algorithm based on the extended Simpson quadrature rule.

Using the extended three-point Simpson rule on an uniform grid, the above convolution integral can be approximated as

$$I(x) = \bar{I}(x) + O\left(h^4\right), \tag{6}$$

where h is the grid size, and $\bar{I}(x)$ is the numerical approximation of the convolution integral that is given by

$$\bar{I}(x) = \frac{1}{3}h\left(f_1 + 4\sum_{j=1}^{M} f_{2j} + 2\sum_{j=2}^{M} f_{2j-1} + f_N\right), \tag{7}$$

where the total number of grid points $N = 2^n - 1$, $M = 2^{n-1} - 1$, and the kernel function f is given by

$$f\left(x, x'\right) = G\left(x - x'\right)\rho\left(x'\right) \tag{8}$$

and $f_j(x) = f(x, x'_j)$ with $x'_j = (j-1)h$. Defining a new density function $\bar{\rho}$ such that

$$\bar{\rho}(x_j) = \begin{cases} \rho(x_j), & j = 1, \\ 4\rho(x_j), & j = 2l, \\ 2\rho(x_j), & j = 2l-1, \\ \rho(x_j), & j = N \end{cases} \tag{9}$$

the above convolution summation on grid point i can be rewritten as

$$\bar{I}(x_i) = \frac{1}{3}h\sum_{j=1}^{N+1} G(x_i - x_j)\bar{\rho}(x_j). \tag{10}$$

Here, we have added one zero value term ($\bar{\rho}(x_{N+1}) = 0$) to the original summation so that $N+1$ is a power of 2. The direct calculation of the above summation requires $O(N)$ operations for a single point x_i, where $i = 1, \ldots, N$. To obtain the convolution for all N points on the grid, the total computational cost will be $O(N^2)$. Fortunately, for the Green function given above, this summation can be calculated using an FFT based method on a doubled computational domain [5–7]. This reduces the computational cost of the original convolution for all grid points from $O(N^2)$ to $O(N\log(N))$.

To use the FFT based method, the convolution summation has to be replaced by a cyclic summation in the double-gridded computational domain:

$$\bar{I}_c(x_i) = \frac{1}{3}h\sum_{j=1}^{2N+2} G_c(x_i - x_j)\bar{\rho}_c(x_j), \tag{11}$$

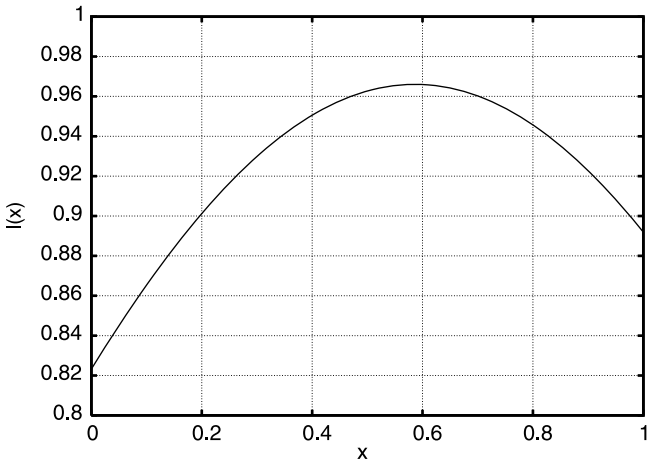

Fig. 1. The temperature distribution after the Gauss transform.

where $i = 1, \ldots, 2N+2$, $j = 1, \ldots, 2N+2$ and

$$\bar{\rho}_c(x_j) = \begin{cases} \bar{\rho}(x_j), & 1 \leqslant j \leqslant N+1, \\ 0, & N+1 < j \leqslant 2N+2, \end{cases} \tag{12}$$

$$G_c(x_k) = \begin{cases} G(x_k), & 1 \leqslant k \leqslant N+2, \\ G(-x_{2N+4-k}), & N+2 < k \leqslant 2N+2, \end{cases} \tag{13}$$

$$\bar{\rho}_c(x_j) = \bar{\rho}_c\left(x_j + 2(L+2h)\right), \tag{14}$$

$$G_c(x_k) = G_c\left(x_k + 2(L+2h)\right), \tag{15}$$

where $L = (N-1)h$. From above definition, one can show that the cyclic summation gives the same value as the convolution summation within the original domain, i.e.

$$\bar{I}(x_i) = \bar{I}_c(x_i) \quad \text{for } i = 1,\ N+1. \tag{16}$$

In the cyclic summation, the kernel is a discrete periodic function in the doubled computational domain. This cyclic summation can calculated using the FFT method, i.e.

$$\bar{I}_c(x_i) = \mathrm{Inv}\,FFT\left(\bar{I}_c(\omega)\right). \tag{17}$$

Here, Inv *FFT* denotes the inverse FFT of the function $\bar{I}_c(\omega)$ that is given by

$$\bar{I}_c(\omega) = G_c(\omega)\bar{\rho}_c(\omega), \tag{18}$$

where $G_c(\omega)$ and $\bar{\rho}_c(\omega)$ denote the forward FFT of the function G_c and the $\bar{\rho}_c$ respectively. The computational operation to calculate cyclic summation using above FFT method is of $O(N\log(N))$.

As an illustration of above algorithm, we calculate a one-dimensional Gauss transform of an initial temperature function $\rho(x') = x' + \sin^2(2\pi x')$, i.e.

$$I(x) = \int_0^1 \exp\left(-\frac{(x-x')^2}{2}\right)\rho\left(x'\right)dx'. \tag{19}$$

Such a transform can be used after some extension to solve the heat transfer equation with a given initial temperature distribution [10]. Fig. 1 shows the new function after the Gauss transform in the example.

To verify the accuracy of above algorithm, Fig. 2 shows the relative errors of the calculated convolution at $x = 0.5$ for a number of grid points. As a comparison, we also give the relative errors using the trapezoidal rule. It is expected that the Simpson rule FFT based method converges much faster than the standard trapezoidal rule based method.

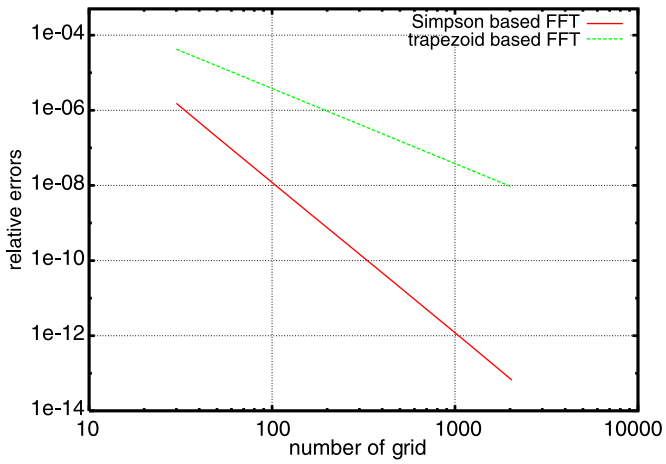

Fig. 2. The relative errors at $x = 0.5$ as a function grid points.

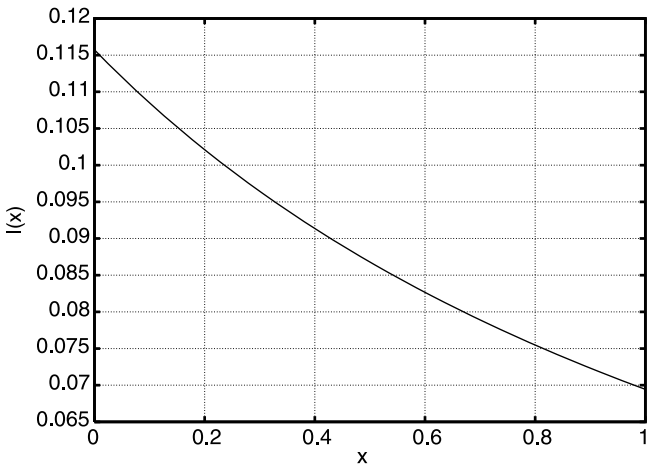

Fig. 3. Electric field as a function of x on-axis.

3. Two-dimensional convolution using the Simpson rule

Using the Simpson rule and the FFT method, the above algorithm can be readily generalized to two-dimensional or higher-dimensional convolution with a smooth kernel. For a two-dimensional convolution

$$I(x, y) = \int_0^a \int_0^b G(x - x', y - y')\rho(x', y')\, dx'\, dy', \tag{20}$$

the Simpson rule for this integral follows the tensor product of the one-dimensional algorithm yielding

$$I(x, y) = \bar{I}(x, y) + O\left(h_x^r h_y^s\right), \tag{21}$$

where $r + s = 4$, r and s are non-negative integer, and

$$\bar{I}(x, y) = \frac{h_x h_y}{9} \sum_{i'=1}^{N_x-1} \sum_{j'=1}^{N_y-1} c_{i',j'} G(x - x'_i, y - y'_j)\rho(x'_i, y'_j), \tag{22}$$

where $N_x = 2^n$, $N_y = 2^k$ and the coefficients $c_{i',j'}$ for $i' \neq 1$ and $i' \neq N_x - 1$, $j' \neq 1$ and $j' \neq N_y - 1$ are given by

$$c_{i',j'} = \begin{cases} 16, & i' = 2l \text{ and } j' = 2m, \\ 4, & i' = 2l - 1 \text{ and } j' = 2m - 1, \\ 8, & i' = 2l \text{ and } j' = 2m - 1; \text{ or } i' = 2l - 1 \text{ and } j' = 2m \end{cases} \tag{23}$$

and $c_{1,j'}, c_{N_x-1,j'}, c_{i',1}, c_{i',N_y-1} = 4$ for even number of i' or j', $c_{1,j'}, c_{N_x-1,j'}, c_{i',1}, c_{i',N_y-1} = 2$ for odd number of i' or j', and $c_{1,1}, c_{1,N_y-1}, c_{N_x-1,1}, c_{N_x-1,N_y-1} = 1$. As we did for one-dimensional case, we add one more grid at the boundary of the computational domain so that the density $\rho_{i',N_y} = 0$ and $\rho_{N_x,j'} = 0$ for all i' and j'. Defining a new density function $\bar{\rho}(x_{i'}, y_{j'}) = c_{i',j'}\rho(x_{i'}, y_{j'})$, the numerical approximation of the convolution integral on each grid point (i, j) can be rewritten as

$$\bar{I}(x_i, y_j) = \frac{h_x h_y}{9} \sum_{i'}^{N_x} \sum_{j'}^{N_y} G(x_i - x_{i'}, y_j - y_{j'})\bar{\rho}(x_{i'}, y_{j'}). \tag{24}$$

This convolution summation can be replaced by a two-dimensional cyclic summation in the same way as the one-dimensional case.

$$\bar{I}_c(x_i, y_j) = \frac{h_x h_y}{9} \sum_{i'=1}^{2N_x} \sum_{j'=1}^{2N_y} G_c(x_i - x_{i'}, y_j - y_{j'})\bar{\rho}_c(x_{i'}, y_{j'}), \tag{25}$$

where $i = 1, \ldots, 2N_x$, $j = 1, \ldots, 2N_y$, and

$$\bar{\rho}_c(x_i, y_j) = \begin{cases} \bar{\rho}(x_i, y_j), & 1 \leqslant i \leqslant N_x,\ 1 \leqslant j \leqslant N_y, \\ 0, & \text{otherwise}, \end{cases} \tag{26}$$

$$G_c(x_i, y_j) = \begin{cases} G(x_i, y_j), & 1 \leqslant i \leqslant N_x;\ 1 \leqslant j \leqslant N_y, \\ G(-x_{2N_x-i+2}, y_j), & N_x < i \leqslant 2N_x;\ 1 \leqslant j \leqslant N_y, \\ G(x_i, -y_{2N_y-j+2}), & 1 \leqslant i \leqslant N_x;\ N_y < j \leqslant 2N_y, \\ G(-x_{2N_x-i+2}, -y_{2N_y-j+2}), & N_x < i \leqslant 2N_x; \\ & N_y < j \leqslant 2N_y. \end{cases} \tag{27}$$

Within the original computational domain, the cyclic summation gives the same results as the convolution summation, i.e.

$$\bar{I}(x_i, y_j) = \bar{I}_c(x_i, y_j) \quad \text{for } i = 1,\ N_x;\ j = 1,\ N_y. \tag{28}$$

This cyclic summation can be calculated using the two-dimensional FFT in the same way as in the one-dimensional case. The computational cost of the above cyclic summation using the FFT scales as $O(N_x N_y(\log(N_x) + \log(N_y)))$ while the computational cost of the direct convolution summation scaling as $O(N_x^2 N_y^2)$.

As an application of the above algorithm, we calculated the electric field in a domain generated by a remote charged particle beam using a convolution of the Green function and the density function. This electric field is used in accelerator physics to study long-range colliding beam effects [11] and image charge effects of conducting photo cathode [12].

$$E_x(x, y) = \int_0^1 \int_0^1 \frac{(x_0 + x - x')}{(x_0 + x - x')^2 + (y_0 + y - y')^2} \rho(x', y')\, dx'\, dy', \tag{29}$$

where

$$\rho(x', y') = \exp\left(-\frac{1}{2}\left(\frac{(x' - x_c)^2}{\sigma_x^2} + \frac{(y' - y_c)^2}{\sigma_y^2}\right)\right), \tag{30}$$

where $x_c = y_c = 0.5$, $\sigma_x = \sigma_y = 0.16667$, $x_0 = 2$, and $y_0 = 0$.

Fig. 3 shows the on-axis electric field $E_x(x, 0.5)$ from the above convolution as a function of x. The electric field decreases quickly as a function of separation distance.

To verify the convergence of the two-dimensional algorithm, we also calculated the relative errors at $(0.5, 0.5)$ as a function of mesh grid points together with the conventional trapezoidal rule based FFT method. The results are given in Fig. 4. It is seen that the Simpson rule FFT based method converges much faster than the trapezoidal rule method as expected.

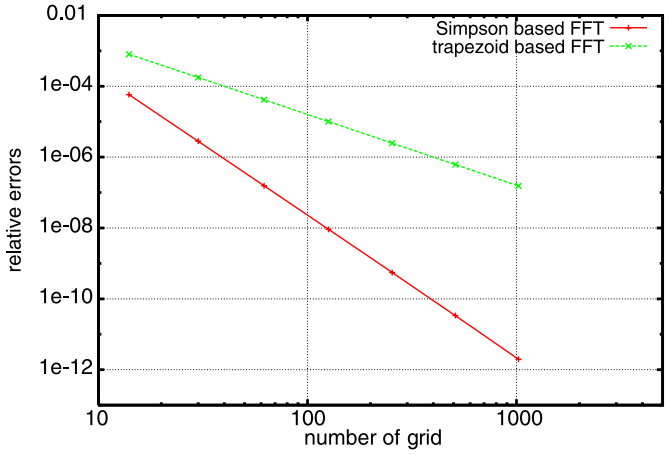

Fig. 4. The relative errors of electric field convolution at center as a function of grid points.

4. Discussions

In above sections, we showed that using the Simpson rule quadrature approximation for integral and the FFT method can lead to an efficient way to numerical calculation of a type of convolution with $O(h^4)$ accuracy and $O(N\log(N))$ computational cost in one- and two-dimensional convolutions. Using the tensor product of the Simpson rule, the above algorithm can be readily extended to three-dimensional convolution. Even though the algorithm given here is for calculation of convolution, it can be also used to calculate the cross-correlation of two functions after some variable transformation.

The above Newton–Cotes algorithm to calculate the convolution is illustrated using the Simpson quadrature approximation to an integral and has an accuracy of $O(h^4)$. This accuracy can be further improved by using an even higher-order Newton–Cotes quadrature rule such as Boole's rule for the approximation of the convolution integral. The downside of these even higher-order quadrature rules is that the new defined effective density function becomes increasingly complicated in two or three dimensions so that it may not be easily implemented into a computer program for practical applications.

Acknowledgements

We would like to thank Drs. J. Eastwood and J. Strain for helpful discussions. This research was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center.

References

[1] G. Barton, Elements of Green's Functions and Propagation, Oxford University Press, New York, USA, 1989.
[2] J.V. Beck, K.D. Cole, A. Haji-Sheikh, B. Litkouhi, Heat Conduction Using Green's Functions, Taylor & Francis, New York, NY, USA, 2003.
[3] J. Qiang, R. Ryne, S. Habib, V. Decyk, J. Comput. Phys. 163 (2000) 434.
[4] R.D. Ryne, J. Qiang, S. Habib, in: J. Rosenzweig, L. Serafini (Eds.), The Physics of High Brightness Beams, World Scientific, 2000, p. 91.
[5] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Adam Hilger, New York, 1988.
[6] J.W. Eastwood, D.R.K. Brownrigg, J. Comput. Phys. 32 (1979) 24.
[7] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed., Cambridge University Press, Cambridge, England, 1992.
[8] M. Abramowitz, I.A. Stegun (Eds.), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover, New York, 1972.
[9] http://mathworld.wolfram.com/Newton-CotesFormulas.html.
[10] http://en.wikipedia.org/wiki/Heat_equation.
[11] J. Qiang, M.A. Furman, R.D. Ryne, J. Comput. Phys. 198 (2004) 278.
[12] J. Qiang, S. Lidia, R.D. Ryne, C. Limborg-Deprey, Phys. Rev. ST Accel. Beams 9 (2006) 044204.