Fast evaluation of lattice Green functions

1987 J. Phys. A: Math. Gen. 20 5095

(http://iopscience.iop.org/0305-4470/20/15/026)

J. Phys. A: Math. Gen. **20** (1987) 5095-5111. Printed in the UK

Fast evaluation of lattice Green functions

Richard Friedberg†‡ and Olivier Martin§‖

† Department of Physics, Barnard College and Columbia University, New York, NY 10027, USA

§ Department of Physics, University of Illinois at Urbana-Champaign, Loomis Laboratory, 1110 W Green Street, Urbana, IL 61801, USA

Received 11 December 1986, in final form 16 March 1987

Abstract. Numerical methods for obtaining lattice Green functions are presented. The whole array to arbitrary distance on an infinite lattice can be generated in approximately 50 μs CPU (VAX) per site. The accuracy is close to machine error. For small mass, the coefficients of the power series expansion can be generated similarly. Results are given in two and three dimensions.

1. Introduction

Many problems involve the free field propagator

$$f(k)=\frac{1}{\Sigma_\mu k_\mu^2+M^2} \tag{1.1}$$

which on a lattice is modified to

$$f(k)=\frac{1}{\Sigma_\mu (2-2\cos k_\mu)+M^2}. \tag{1.2}$$

These problems include perturbative calculations of effects of weak disorder [1]¶ or of Wilson loops in lattice theories [2]. (There may also be numerators depending on k, but the main difficulties come from the denominators.)

Typically an n-loop calculation in D dimensions requires a numerical integration over nD momentum variables. In many cases, the computing time can be cut drastically by converting the integrals to sums in position space. For instance, a number of two-loop integrals can be reduced to sums over a single position in D space. Moreover, a sum is free of the interpolation error inherent in evaluating an integral. Since in principle one is dealing with an infinite lattice, there is a truncation error as the sum must be cut off somewhere, but the error can be clearly estimated.

It is further possible to reduce the truncation error by calculating the difference between the lattice diagram and some known continuum quantity; this difference converges faster with the cutoff. We shall discuss such devices in a forthcoming paper.

‡ Work supported in part by DOE.

‖ Work supported in part by NSF.

¶ Note equations (1.6.4) and (1.6.10). We write M^2 for $\Omega_0^2-\omega^2$, and take D_{l_i} to be $\varepsilon_i(2-2\cos lc_i)$.

Here, we address ourselves to the problem of determining the lattice Green functions (LGF) or position-space propagators defined by

$$\Psi(x)=\frac{1}{(2\pi)^D}\int_0^{2\pi}\prod \mathrm{d}k_\mu \exp\left(\mathrm{i}\sum_\mu k_\mu x_\mu\right) f(k) \tag{1.3}$$

where $f(k)$ is given by (1.2).

Clearly, if (1.3) must be done as a D-dimensional numerical integral at each lattice site, all advantage of the position method is lost. To date, the best evaluations of (1.3) have been done either (a) by fast Fourier transform in a finite volume [3] or (b) by reducing (1.3) to a one-dimensional integral of a product of Bessel functions [2] (see § 5.4).

In method (a), the work required grows logarithmically faster than the volume, and more importantly, the answer contains artefacts due to periodicity. Also the Fourier transforms do not give accurate results when the LGF is very small, e.g., close to machine accuracy.

Method (b) deals directly with the infinite volume case (though it can be adapted to finite volume) and is very accurate. However, it is much slower, requiring a numerical integral at each lattice site. It is ideally suited for problems in which one needs the LGF only at a few points, since the work is limited to those points.

We present here a method which, like the fast Fourier transform, generates a whole array of LGF, but in an efficient way. It is even faster than the FFT, as the work per site remains fixed no matter how large the array. Like the Bessel function method, it is absolutely free of finite-volume effects and can be extended in principle arbitrarily far from the origin. Even at large distances the relative error is not much more than machine rounding (10^{-16} in double precision).

The heart of our method is that, in addition to the Laplace difference equation, the LGF exactly satisfy an auxiliary set of first-order difference equations. By means of them, the LGF can be found quickly all over D space if they are known on a one-dimensional tube of cross section $2^{(D-1)}$. Moreover, the auxiliary equations can be combined with the Laplace equation to yield a *self-contained* set of difference equations determining the LGF on the tube. In § 2, we derive the auxiliary equations and show how to extend the tube solution to D space. This procedure requires about 50 μs CPU per site on a VAX.

The solution of the tube problem is more complicated than the extension to D space, but for large arrays the computing time is small by comparison since the number of sites involved is only linear in the radius of the array to be computed. In §§ 3 and 4 we describe a sifting procedure for obtaining extremely accurate solutions. For $M \neq 0$ these methods require a radius much larger than M^{-1}, so that at very small M it seems worthwhile to express the LGF as a power series in M at each site. In § 5, we show how this can be done using the $M=0$ values as a starting point. Section 6 gives our conclusions.

2. Reduction to one dimension

2.1. The gradient equations

As is well known, the LGF defined by (1.3) satisfy the difference equation

$$\sum_\mu [\Psi(x+\mu)+\Psi(x-\mu)]-(2D+M^2)\Psi(x)=-\delta_{x0} \tag{2.1}$$

which is the lattice analogue of Laplace's equation with unit source

$$\nabla^2\Psi_c - M^2\Psi_c = -\delta(x) \tag{2.2}$$

in the continuum, where Ψ_c is the Fourier transform of (1.1).

Now, the continuum Green function Ψ_c, being isotropic, also satisfies

$$\nabla\Psi_c = \hat{r}\frac{\partial}{\partial r}\Psi_c \tag{2.3}$$

which yields (by factoring out $(1/r)\partial\Psi_c/\partial r$) a set of $D-1$ linear homogeneous equations

$$\frac{1}{x_1}\frac{\partial\Psi_c}{\partial x_1} = \frac{1}{x_2}\frac{\partial\Psi_c}{\partial x_2} = \dots \tag{2.4}$$

in the derivatives of Ψ_c.

Likewise, Ψ, although not isotropic, satisfies an equation like (2.3). We have, for any fixed μ,

$$\begin{aligned}\Psi(x+\mu)-\Psi(x-\mu) &= \frac{1}{(2\pi)^D}\int\prod \mathrm{d}k_\nu \frac{2\mathrm{i}\sin k_\mu \exp(\mathrm{i}\Sigma k_\nu x_\nu)}{\Sigma_\nu(2-2\cos k_\nu)+M^2}\\ &= \frac{1}{(2\pi)^D}\int\prod_{\nu\neq\mu}[\mathrm{d}k_\nu \exp(\mathrm{i}k_\nu x_\nu)]\int_0^{2\pi} -2\mathrm{i}\exp(\mathrm{i}k_\mu x_\mu)\\ &\quad\times\frac{\mathrm{d}\cos k}{\Sigma_\nu(2-2\cos k_\nu)+M^2}\\ &= x_\mu\Phi(x)\end{aligned} \tag{2.5}$$

by integration by parts over k_μ, where

$$\Phi(x) = \frac{1}{(2\pi)^D}\int\prod_\nu \mathrm{d}k_\nu \exp\left(\mathrm{i}\sum k_\nu x_\nu\right)\ln\left(\sum_\nu(2-2\cos k_\nu)+M^2\right) \tag{2.6}$$

is the same for all μ. Hence

$$\frac{\Psi(x+\mu_1)-\Psi(x-\mu_1)}{x_1} = \frac{\Psi(x+\mu_2)-\Psi(x-\mu_2)}{x_2} = \dots. \tag{2.7}$$

We shall call these $D-1$ equations the *gradient equations.*

2.2. Error-prone extension from a cube

By combining (2.7) with (2.1), we obtain D linear equations at each site. Thus the LGF appear to be overdetermined since there are D times as many equations as unknowns; but of course the equations at neighbouring sites contain the required number of redundancies. Nevertheless the system of (2.7) with (2.1) is stronger than (2.1) alone in that fewer boundary conditions are needed.

In fact, if the LGF are given on a 2-cube (2^D adjacent sites forming a cube), they can be determined from (2.7) and (2.1) on the whole space. To see this, consider a corner of the 2-cube. This site has $2D$ neighbours of which D are in the cube. Therefore the D equations (2.7) and (2.1) contain just D unknowns (the neighbours not in the cube) and can be solved. Thus knowledge of the LGF is extended to all contiguous sites and, by extension through consecutive 2-cubes, all over the lattice.

This procedure, however, is not useful because the system of (2.7) and (2.1) taken all over the lattice actually admits 2^D independent solutions ((2.1) alone admits an infinite number if $D>1$), at least one of which grows exponentially in any given direction. Even if one knew the exact LGF on the original 2-cube, machine rounding would introduce small amounts of 'bad' solution which would grow rapidly (roughly times $4D-2$ at each step) and accuracy would soon be lost [4].

2.3. Error-free extension from a tube

A much better result is obtained from the following observation. Let a 'tube' signify an infinite one-dimensional series of consecutive 2-cubes. The tube has cross section 2^{D-1} sites per unit length and may accordingly be resolved into 2^{D-1} parallel infinite straight lines. Now if the LGF are given on a whole tube, they can be extended to the whole space by the gradient equations *alone*. Moreover, if the tube passes through the origin, this can be done without any magnification of machine error.

Let us explain this first in two dimensions. Let any point be labelled by the coordinates (n, m) and let the tube consist of the two parallel lines $m=0$ and $m=1$. For any n, m_0, the gradient equation at (n, m_0) gives

$$\Psi(n, m_0+1)-\Psi(n, m_0-1)=(m_0/n)[\Psi(n+1, m_0)-\Psi(n-1, m_0)] \tag{2.8}$$

so that Ψ can be determined on the line $m=m_0+1$ if it is known for $m=m_0-1$ and $m=m_0$. (For $n=0$, (2.8) is indeterminate, but that does not matter as $\Psi(0, m+1)=\Psi(m+1, 0)$ is already known.) Thus, from the lines $m=0$ and $m=1$, we generate $m=2$, $m=3$, and so forth. Because of the symmetry $\Psi(n, m)=\Psi(m, n)$, it suffices to use (2.8) only to generate the half-quadrant $n>m>0$.

Now suppose there is a rounding error $\varepsilon(n_0, m_0)$ in the determination of $\Psi(n_0, m_0)$. This error will propagate to other sites (n, m) for which $|n-n_0|\leqslant m-m_0$. The total error at site (n, m) will be $\Sigma_{n_0 m_0}\varepsilon(n_0, m_0)G_{nm}^{n_0 m_0}$, where G is determined by the gradient equation

$$G_{n\,m+1}^{n_0 m_0}-G_{n\,m-1}^{n_0 m_0}=\frac{m}{n}(G_{n+1\,m}^{n_0 m_0}-G_{n-1\,m}^{n_0 m_0}) \qquad m\geqslant m_0 \tag{2.9}$$

and the initial conditions

$$G_{n m_0}^{n_0 m_0}=\delta_{n n_0} \qquad G_{n\,m_0-1}^{n_0 m_0}=0. \tag{2.10}$$

These equations have the solution

$$G_{nm}^{n_0 m_0}=\tfrac{1}{2}\pi n_0\,\mathrm{Re}(\mathrm{i}^{n+m-n_0-m_0})g_{nm}^{n_0 m_0} \tag{2.11}$$

where

$$g_{nm}^{n_0 m_0}=\int_0^\infty J_n(u)J_{n_0}(u)[J_m(u)N_{m_0-1}(u)-N_m(u)J_{m_0-1}(u)]\,\mathrm{d}u. \tag{2.12}$$

(The divergence at small u for $m>n+n_0+m_0$ is related to the indeterminacy of (2.9) for $n=0$. It does not concern us since we use (2.8) only to generate Ψ for $m\leqslant n$. The symmetry $g_{nm}^{n_0 m_0}=g_{n_0 m}^{n m_0}$ can also be derived directly from a hopping expansion.)

We have evaluated G numerically by direct use of (2.9) and (2.10) and find $|G|\leqslant 1$ in all cases examined, as long as $m\leqslant n$. This is plausibly consistent with (2.12) for points close to the origin. For points far from the origin, a saddle-point treatment of (2.12) shows that $g_{\Lambda\nu_0\Lambda\mu_0}^{\Lambda\nu_0\Lambda\mu_0}$ diminishes as $\Lambda^{-3/2}$ for large Λ. If the rounding introduces

an uncorrelated mean square error $\langle \varepsilon^2 \rangle$ at each site, the total mean square error at $(n, m) = (\Lambda\nu, \Lambda\mu)$ will be

$$\Delta_{nm}^2 \sim \langle \varepsilon^2 \rangle \sum_{n_0 m_0} (G_{nm}^{n_0 m_0})^2 \sim \langle \varepsilon^2 \rangle \int_0^{\mu} \Lambda \, \mathrm{d}\mu_0 \int_{\nu-(\mu-\mu_0)}^{\nu+(\mu-\mu_0)} \Lambda \, \mathrm{d}\nu_0 (\Lambda\nu_0 g_{\Lambda\nu\Lambda\mu}^{\Lambda\nu_0\Lambda\mu_0})^2 \tag{2.13}$$

which grows linearly as Λ. Thus even at 10^4 sites away from the origin, Δ_{nm} is only two decimals greater than machine error.

The extension to higher dimensions is easy because each equation of (2.7) involves only four coplanar sites. Therefore, within each plane, the recursion (2.8) will generate the whole plane if two adjacent lines are given, one passing though the origin. Consequently, the whole D space is generated if two adjacent $(D-1)$ spaces are given, one passing through the origin. This is done by dividing D space into planes perpendicular to the $(D-1)$ spaces. By induction on D, it follows that all D space can be generated from a tube. (The process is made quite clear by considering $D=3$.) Since a whole plane is generated at each step, the error propagation is governed by exactly the same equations (2.9) and (2.10), and the total error remains small as in two dimensions. We have naturally checked that the numerical results we have obtained are in agreement with this analysis.

Since the gradient equation does not involve M, the foregoing analysis holds equally whether the mass is large, small or zero. However, to find Ψ on the tube, one needs a variety of methods for these different cases.

3. Tube solution for $M = 0$

3.1. The cubical constraint

In the massless case there is a special relation that follows from the 'Laplace' and gradient equations taken together. We consider a pair of adjacent 2-cubes, with 2^{D-1} sites in common and call one of the common sites x. This site has $D-1$ neighbours that do not belong to either cube (external neighbours). Each external neighbour can be expressed in terms of three internal neighbours by one of the equations (2.7). Now we write (2.1) at x, but replace the external neighbours by their expressions in terms of internal neighbours and sum the result over the 2^{D-1} possibilities for x. We now have a linear homogeneous equation in the $3 \times 2^{D-1}$ variables belonging to the pair of cubes. It turns out that this equation can be written as

$$C \text{ (cube 1)} = C \text{ (cube 2)} \tag{3.1}$$

where C is a linear combination of Ψ at the corners of the cube, with coefficients depending on the position of the cube but not on which neighbouring cube was used to derive (3.1). Therefore by proceeding from cube to cube, we find that C is the same for all cubes. Let us call this the cubical constraint.

The expression for C can be written in terms of the position vector $\bar{x}$ of the centre of the cube

$$C = \sum_{x \text{ at corners}} \left(4 \sum_{\mu} \bar{x}_\mu (x_\mu - \bar{x}_\mu) + D - 2 \right) \Psi(x). \tag{3.2}$$

In this form C may be recognised as the lattice analogue of the continuum quantity

$$C_c = 2^D (\boldsymbol{r} \cdot \boldsymbol{\nabla} + D - 2) \Psi_c(\boldsymbol{r}) \tag{3.3}$$

which of course is also constant, when $M = 0$.

Since C and C_c become the same far from the origin, they must be the same everywhere. This gives

$$C = 0 \qquad \text{for } D > 2 \tag{3.4}$$

which will play an important role in our sifting procedure.

In two dimensions, C simplifies, but the asymptotic value is no longer zero. We obtain

$$2(n+m+1)[\psi(n+1, m+1) - \Psi(n, m)] + 2(n-m)[\Psi(n+1, m) - \Psi(n, m+1)] = -4/2\pi \tag{3.5}$$

and in particular for $n = m$

$$\Psi(n+1, n+1) - \Psi(n, n) = -\frac{1}{(2n+1)\pi}. \tag{3.6}$$

Now in two dimensions the LGF become infinite in the limit $M \to 0$, but their differences remain finite. To study the differences, we may as well set

$$\Psi(0, 0) = 0. \tag{3.7}$$

Then

$$\Psi(1, 0) = \Psi(0, 1) = -\tfrac{1}{4} \tag{3.8}$$

because of (2.1), and from (3.6)

$$\Psi(1, 1) = -1/\pi. \tag{3.9}$$

Since Ψ is now determined on the four corners of a square, it can be extended to the whole plane by using (2.1) and (2.7); one obtains

$$\Psi(n, m) = \pi^{-1} p(n, m) - q(n, m) \tag{3.10}$$

where the rational numbers p, q are given explicitly by recursion. (Thus $\Psi(2, 0) = 2/\pi - 1$.)

Unfortunately, this method is ineffective as explained in § 2.3: p and q become large away from the diagonal, and accuracy is lost in the subtraction. Hence we make no use of (3.5) except as an afterthought, to check our calculation of $\Psi(1, 1)$.

3.2. Sifting procedure for $D = 2$

Let us take the case of two dimensions for simplicity. For $n = 1, 2, \ldots$, let

$$f_n = \Psi(n, 0) \qquad g_n = \Psi(n, 1) = \Psi(n, -1). \tag{3.11}$$

Then f_{n+1} is given by (2.1) in terms of f_{n-1}, f_n and g_n. The single gradient equation does not yield any relation at $(n, 0)$. However, at $(n, 1)$, it can be combined with the Laplace equation to eliminate $\Psi(n, 2)$; this gives an expression for g_{n+1}. The results are

$$f_{n+1} = 4f_n - f_{n-1} - 2g_n \tag{3.12}$$

$$(n+1)g_{n+1} = 4ng_n - (n-1)g_{n-1} - 2nf_n. \tag{3.13}$$

These equations involve only the LGF in the tube; hence the problem is now one dimensional.

Suppose one iterates the 4×4 system. At large n, the iteration is approximately

$$\begin{pmatrix} f_{n+1} \\ g_{n+1} \\ f_n \\ g_n \end{pmatrix} \simeq \begin{pmatrix} 4 & -2 & -1 & 0 \\ -2 & 4 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_n \\ g_n \\ f_{n-1} \\ g_{n-1} \end{pmatrix}. \tag{3.14}$$

Diagonalising this matrix gives four eigenvalues: 1, 1, λ_a and λ_a^{-1} where $\lambda_a = 3+\sqrt{8} \simeq 5.8$. Therefore if the initial conditions are not right, the solution will blow up as λ_a^n. The eigenvalues 1 are associated with the true solution (growing logarithmically) and the constant solution. The constant has been fixed by (3.7); in fact, we might as well express (3.12) and (3.13) in terms of differences $f_n - f_{n+1}$, $f_n - g_n$, so that the transfer matrix is only 3×3 and has eigenvalues 1, λ_a, λ_a^{-1}.

The initial conditions involve f_0, g_0, f_1 and g_1. The first three are determined by (3.7) and (3.8). But g_1 is unknown since we are not using (3.9). Instead it will be determined by requiring that the solution *not* blow up as λ_a^n.

If there were no rounding error we might proceed as follows. Generate a solution of (3.12) and (3.13) (call it Ψ^a) by starting with $g_1 = 1$, $f_0 = g_0 = f_1 = 0$. Of course Ψ^a will blow up as λ_a^n. Cut the process off at some convenient value $n = n_1$. Then generate a solution $\Psi^{(0)}$ by imposing (3.7), (3.8) and $g_1^{(0)} = 0$. This will also blow up. By setting $\alpha_1 = f_{n_1}^{(0)}/f_{n_1}^a$, and starting with $g_1^{(1)} = -\alpha_1$ instead of 0, one generates an improved solution

$$\Psi^{(1)} = \Psi^{(0)} - \alpha_1 \Psi^a \tag{3.15}$$

which is guaranteed to have $f_{n_1}^{(1)} = 0$ and therefore cannot be very different from the true Ψ. Since the true f_n is not quite zero, $\Psi^{(1)}$ will also blow up as λ_a^{n-m} when continued beyond n_1. When we reach some value n_2, we set $\alpha_2 = f_{n_2}^{(1)}/f_{n_2}^a$ and $g_1^{(2)} = g_1^{(1)} - \alpha_2$, obtaining a solution

$$\Psi^{(2)} = \Psi^{(1)} - \alpha_2 \Psi^a \tag{3.16}$$

that behaves well out to n_2, and so on.

In the presence of a rounding error, the first correction will work if we take n_1 so that $\Psi^a(n_1, 0)$ is near the inverse of the machine error. However, at the second correction the value of α_2 will be as small as the machine error; hence g_1 will not be corrected accurately, and $\Psi^{(2)}$ will still blow up at $n \sim n_2$. Subsequent corrections will not change g_1 at all.

We therefore modify the procedure. Having calculated α_2 at n_2, we use (3.16) directly to compute $\Psi^{(2)}$ for $n \leqslant n_1$. This gives good values at $n \sim n_1$ which can be used in (3.12) and (3.13) to start the recursion. The resulting $\Psi^{(2)}$ behaves well up to n_2. When it blows up at n_3, we repeat the process, obtaining $\Psi^{(3)}$ by direct subtraction for $n \leqslant n_2$ and by recursion from n_2 onward. We have found that the resulting solution is accurate nearly to machine error even hundreds of sites from the origin.

If we were to use (3.8) at the beginning, it would only make the starting solution as good as $\Psi^{(1)}$; the subsequent corrections would still be necessary. We did not do so because we wished to see whether our recursion would calculate π all by itself. Indeed we obtained

$$-1/g_1 = 3.141\,592\,653\,589\,793\,2(27) \tag{3.17}$$

agreeing with the tabulated value of π until the bracketed decimals. The discrepancy is less than one unit of machine error.

The calculation of π can be reduced to the algorithm

$$\pi = 4 \lim_{n\to\infty} (M_{12}^{(n)} + M_{22}^{(n)})/(M_{11}^{(n)} + M_{21}^{(n)})$$

$$M^{(n)} = \begin{pmatrix} 1 & 0 \\ 0 & (n+1)^{-1} \end{pmatrix} \left[\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & n \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix} M^{(n-1)} \right] \qquad M^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.18}$$

The first three approximations are $\frac{16}{5}=3.20$, $\frac{236}{75}=3.147$, $\frac{1216}{387}=3.1421$. The error diminishes by about $3+\sqrt{8}$ at each step.

3.3. Sifting for $D>2$

Consider $D=3$. The tube consists of four sites for each n, but only three are independent because $\Psi(n,1,0)=\Psi(n,0,1)$. The recursion is

$$\begin{aligned} f_{n+1} &= 6f_n - f_{n-1} - 4g_n \\ (n+1)g_{n+1} &= 6ng_n - (n-1)g_{n-1} - 2n(f_n + h_n) \\ (n+2)h_{n+1} &= 6nh_n - (n-2)h_{n-1} - 4ng_n \end{aligned} \tag{3.19}$$

where $f_n=\Psi(n,0,0)$, $g_n=\Psi(n,1,0)$, $h_n=\Psi(n,1,1)$. The eigenvalues at large n are $1, 1, \lambda_a, \lambda_a^{-1}, \lambda_b, \lambda_b^{-1}$ where $\lambda_a=5+\sqrt{24}$, $\lambda_b=3+\sqrt{8}$. The unit eigenvalues select the true LGF ($\sim n^{-1}$) and the constant.

The initial conditions are determined by the eight corners of a starting cube at the origin, but by symmetry these yield only four independent quantities: f_0, $g_0=f_1$, $h_0=g_1$ and h_1. By Laplace's equation at the origin

$$g_0 = f_0 - \tfrac{1}{6}. \tag{3.20}$$

Unlike those for $D=2$, the LGF here are finite, and we do not wish to set $f_0=0$. Instead, the additive constant must be determined so that $\Psi\to 0$ at large n. By means of the cubical constraint this condition can be introduced exactly via the starting cube, where (3.2) and (3.4) give

$$f_0 = 3g_1 + 2h_1. \tag{3.21}$$

In view of (3.20) and (3.21), there remain only two unknown parameters g_1 and h_1. These will be fixed by requiring the solution to be free of both the λ_a and the λ_b mode.

To avoid inaccurate subtractions, the sifting can be done as follows. First generate Ψ^a as in § 3.2. Then generate a different solution Ψ^b (also homogeneous at the origin) with successive corrections to eliminate the λ_a mode. The final Ψ^b grows only as λ_b^n. Now we are ready to generate the true solution. Suppose we have a solution $\Psi^{(l)}$ with (3.19) which behaves well until $n=n_l$ and blows up as $\lambda_a^{n-n_l}$ after that. At the next cutoff point n_{l+1}, we evaluate α_{l+1} and make a correction from Ψ^a as in § 3.2 so that the new solution $\Psi^{(l)\prime}$ has $f_{n_{l+1}}$ very small. However $g_{n_{l+1}}$ and $h_{n_{l+1}}$ will be comparable to their values in Ψ^b. The three solutions Ψ^a, Ψ^b, $\Psi^{(l)\prime}$ now have quite different ratios $g_{n_{l+1}}/f_{n_{l+1}}$. Therefore we can accurately find a new α_{l+1} and a β_{l+1} so that

$$\begin{aligned} \alpha_{l+1} f^a_{n_{l+1}} + \beta_{l+1} f^b_{n_{l+1}} &= f^{(l)\prime}_{n_{l+1}} \\ \alpha_{l+1} g^a_{n_{l+1}} + \beta_{l+1} g^b_{n_{l+1}} &= g^{(l)\prime}_{n_{l+1}} \end{aligned} \tag{3.22}$$

since the determinants used to solve (3.22) involve no large cancellations. The new solution $\Psi^{(l+1)}$ is now given by

$$\Psi^{(l+1)} = \Psi^{(l)\prime} - \alpha_{l+1}\Psi^a - \beta_{l+1}\Psi^b \tag{3.23}$$

for $n<n_l$, and by (3.19) thereafter.

For $D>3$ the analogous procedure goes through smoothly. There are $2D$ eigenvalues, of which $D-1$ are >1. The starting cube has 2^D corners but only $D+1$ are independent by symmetry. Imposing the cubical constraint and the Laplace equation at the origin fixes two of these. The remaining $D-1$ parameters are determined by sifting out the $D-1$ growing modes.

4. The tube solution for $M \neq 0$ and numerical results

When the mass is not zero, the quantity C of (3.2) is not constant. This is not surprising since (3.3) is not constant either. Therefore, there is no cubical constraint and we have D unknown parameters instead of only $D-1$. At the same time, the eigenvalues 1 of the asymptotic transfer matrix become $e^{\pm M}$, so that there are now D instead of $D-1$ growing modes. (The extra one was, for $M=0$, the constant solution, which was eliminated by the cubical constraint, or for $D=2$, by setting $\Psi(0,0)=0$.) Thus the sifting method is again applicable; one must simply sift out an additional growing mode. Actually, if one sifts from outside towards the origin, one has again only $D-1$ bad modes. We have found that an extra bad mode affects the relative accuracy very little, which remains near machine error. There are mainly two complications in the quasimassless case. The mode of interest decaying exponentially, there is sometimes a loss of accuracy when one does subtractions since the result is much smaller than the value started with. This problem is cured by restarting the iteration far enough back where the subtraction has not affected the Green function too much. The second problem is that for small mass, one must take the lattice sufficiently large ($\gg 1/M$), so that the e^{+M} mode can be separated from the e^{-M} one. Thus the sifting method does become more time consuming as $M \to 0$. However, there is no critical slowdown, in contrast to relaxational methods for solving the tube solution, e.g. successively over-relaxed iteration.

The actual speed of the tube part of the program in three dimensions was about 100 μs per site in the tube for the first mode to be sifted, and slightly more than n times that for the nth mode. The two-dimensional program was a bit faster. However, the overall speed of the programs was dominated by the extension to space since there are far more sites, and this took 50 μs per site on a VAX 780 in double precision arithmetic.

The accuracy of our method is excellent. Let us first consider the case of two dimensions. We have already illustrated how small the error is by computing π to seventeen digits using real *8 arithmetic on a VAX. It is most natural to ask how the accuracy of the Green functions depends on the mass and the distance from the origin. We have thus converted the programs to real *16 to be able to determine the errors given by the real *8 arithmetic. In practice, we find that the on-axis values have an error in the seventeenth digit near the origin, and that this error grows as one goes further away, but very slowly. Typically two digits are lost by the time one is 100 sites away from the origin, whatever the mass is. This shows that the method is extremely reliable. These features are illustrated by some examples in table 1. In three dimensions, the situation is similar. The accuracy of the tube solution near the origin is machine accuracy and deteriorates very slowly with distance. The number of growing eigenvectors of the iteration (which must be sifted out) does not much affect the accuracy, so we believe this method can be extended to arbitrary dimension with the same high quality results. Again, table 1 gives some numerical results comparing our double precision results to 'exact' results from the quadruple precision programs.

What about the extension from the tube? In § 2, our analysis showed that the use of the gradient equations does not magnify the error as one goes further from the tube. This was indeed confirmed numerically. Again this property does not depend on dimensionality as shown in § 2 and in table 1.

It is necessary to compare our method with previously existing ones. If we start with the integral (1.3), one has to approximate the integration by transforming it to a

Table 1. The Green function at various sites in two and three dimensions for two masses. Underlined digits show where the double precision program disagrees with the exact answer.

Site	Mass = 0.0	Mass = 0.1
(00)	0.000 000 000 000 0000	0.641 559 978 667 7016
(10)	−0.250 000 000 000 0000	0.393 163 878 614 3709
(11)	−0.318 309 886 183 7907	0.326 062 134 078 6916
(20)	−0.363 380 227 632 4187	0.282 902 906 418 5426
(21)	−0.386 619 772 367 5813	0.260 590 700 213 4057
(50)	−0.512 902 329 078 9229	0.147 556 310 490 9589
(10, 0)	−0.623 675 571 215 7085	$6.707\,471\,046\,828\,0910 \times 10^{-2}$
(10, 10)	−0.679 036 325 006 0300	$3.798\,274\,904\,899\,9015 \times 10^{-2}$
(20,0)	−0.734 095 688 013 8674	$1.813\,960\,744\,162\,1715 \times 10^{-2}$
(20, 20)	−0.789 304 497 364 1300	$6.737\,951\,950\,499\,6528 \times 10^{-3}$
(50, 0)	−0.879 955 915 479 8970	$5.884\,157\,957\,517\,5710 \times 10^{-4}$
(50, 50)	−0.935 122 777 651 6508	$6.264\,938\,664\,782\,4272 \times 10^{-5}$
(000)	0.252 731 009 858 6630	0.244 661 638 355 1214
(100)	$8.606\,434\,319\,199\,6337 \times 10^{-2}$	$7.840\,274\,108\,571\,3314 \times 10^{-2}$
(110)	$5.519\,143\,368\,773\,7317 \times 10^{-2}$	$4.772\,745\,412\,853\,1611 \times 10^{-2}$
(111)	$4.357\,835\,439\,772\,5526 \times 10^{-2}$	$3.624\,690\,936\,758\,6074 \times 10^{-2}$
(200)	$4.288\,931\,454\,236\,5747 \times 10^{-2}$	$3.562\,901\,905\,588\,9131 \times 10^{-2}$
(500)	$1.610\,107\,533\,393\,9829 \times 10^{-2}$	$9.825\,868\,803\,661\,5105 \times 10^{-3}$
(10, 0, 0)	$7.978\,261\,541\,929\,4058 \times 10^{-3}$	$2.943\,911\,427\,371\,9748 \times 10^{-3}$
(10, 10, 0)	$5.625\,268\,083\,901\,7879 \times 10^{-3}$	$1.365\,720\,776\,473\,2662 \times 10^{-3}$
(10, 10, 10)	$4.591\,851\,010\,273\,0959 \times 10^{-3}$	$8.104\,563\,230\,164\,6043 \times 10^{-4}$
(20, 0, 0)	$3.981\,378\,573\,047\,7163 \times 10^{-3}$	$5.399\,530\,381\,165\,7203 \times 10^{-4}$
(20, 20, 0)	$2.813\,270\,223\,465\,5466 \times 10^{-3}$	$1.661\,349\,241\,246\,0903 \times 10^{-4}$
(20, 20, 20)	$2.296\,884\,543\,905\,7603 \times 10^{-3}$	$7.177\,536\,645\,238\,7367 \times 10^{-5}$
(50, 0, 0)	$1.591\,708\,769\,407\,9974 \times 10^{-3}$	$1.075\,258\,905\,457\,8923 \times 10^{-5}$

sum. Since the integral is periodic, the error induced by the discretisation is of order e^{-N} for N steps in each direction of k. There will also be a rounding error due to the machine, which goes as $N^{D/2}$ times machine accuracy. This is negligible in practice. Suppose one wants the Green function at a single lattice site: to obtain, say, 10^{-10} accuracy, one needs $N \sim 20$, leading to $(20)^D$ operations. Thus for $D = 3$ this single site might require as much time for 10^{-10} accuracy as our method would take to generate an array of ~ 1000 inequivalent sites (all those with $|x| + |y| + |z| \leqslant 33$) to 10^{-16} accuracy.

To improve the direct method, one can convert the D-dimensional integral to a one-dimensional one over products of Bessel functions:

$$\psi(x) = \frac{1}{(2\pi)^D} \int_0^\infty \exp[-(2D + M^2)\lambda] \prod_i I_{x_i}(2\lambda)\, d\lambda. \tag{4.1}$$

The standerd algorithm for calculating Bessel functions consists of doing a recursion [5] relating $I_n(2\lambda)$ to $I_{n-1}(2\lambda)$. The whole lattice can thus be generated rather efficiently, but at each site, a one-dimensional integral must be done. Again a discretisation has to be made, with errors going as N^{-p} with p an integer depending on the discretisation method. In practice, it is not possible to keep N small while taking p large, so one often settles for $p = 4$. In this case, for an accuracy of 10^{-10}, N is of the order of 300. Each of the N steps at a site requires D multiplications and an addition, assuming the Bessel functions were tabulated in advance. However, our method (in the 'extension' stage) needs only *one* step for each site, consisting of three additions, a

multiplication and a division. Thus the Bessel function method is still not competitive, except possibly as an alternative to our sifting method for the initial tube solution. Even there, our method is more accurate and usually faster, especially for small or zero mass where the Bessel function integral converges slowly.

Another method which is often used to generate the Green function on the whole lattice is fast Fourier transform. This is done for a finite lattice of size L with periodic boundary conditions and requires $2DL^D \ln L$ operations in D dimensions. L must be taken much larger than $1/M$ to minimise finite-size effects. This is the most severe problem with this method; it is just not well suited to the infinite lattice. Since one cannot take advantage of the symmetry of the lattice as in our method, the FFT is of the order of $D!2^D D \ln L$ slower than our method, even neglecting the problem of accuracy. But to reduce finite-size effects to relative $O(1/N)$ in FFT, one must slow it down (by increasing the volume) by another factor N^2 if $D=2$ or 4, or N^3 if $D=3$. Thus FFT is not competitive except for crude accuracy. The other side of the coin is that, if it is the finite volume Green functions that are of interest, our method is not easily applicable and the FFT is a very natural and good way to proceed.

When the mass becomes very small, all of these methods become time consuming. If one needs the Green function at a fixed distance as $M \to 0$, it seems advantageous to determine the series expansion in powers of the mass. We do this in the next section and find it to be both fast and accurate.

5. Analytic expansion in M^2

5.1. General considerations

Considered as analytic functions of M^2, the continuum and lattice Green functions have the same type of singularity at $M^2=0$, since this is governed by small momentum. Letting x be a position vector in D dimensions, the LGF are given by

$$\Psi(x, M^2) = F(x, M^2)S(M^2) + H(x, M^2) \tag{5.1}$$

where F, H are analytic in M^2 and

$$\begin{aligned} S(M^2) &= M^{D-2} \ln \frac{1}{M^2} && D \text{ even} \\ &= M^{D-2} && D \text{ odd.} \end{aligned} \tag{5.2}$$

Writing

$$\begin{aligned} F &= \sum_l A_l(x) M^{2l} \\ H &= \sum_l B_l(x) M^{2l} \end{aligned} \tag{5.3}$$

our problem is to determine the A and B.

We first observe that, since the gradient equations do not contain M, each set of A or B must satisfy them separately, i.e. $[A_l(x+\mu) - A_l(x-\mu)]/x_\mu$ and $[B_l(x+\mu) - B_l(x-\mu)]/x_\mu$ must both be independent of μ for each l. Therefore it suffices to find the A and B on a tube and extend to D space by the method of § 2.

Next, we differentiate (2.6) with respect to M^2, obtaining

$$\frac{\mathrm{d}}{\mathrm{d}M^2}\Phi(x) = \Psi(x) \tag{5.4}$$

which can be substituted into (2.5) to give

$$(l+\tfrac{1}{2}D-1)(A_l(x+\mu)-A_l(x-\mu))=x_\mu(x)A_{l-1}(x) \tag{5.5}$$

$$\begin{aligned} l(B_l(x+\mu)-B_l(x-\mu)) &= x_\mu B_{l-1}(x) && D \text{ odd} \\ &= x_\mu\left(\frac{1}{l}A_{l-D/2}(x)+B_{l-1}(x)\right) && D \text{ even.} \end{aligned} \tag{5.6}$$

These can be used by recursion on l to find the A and B everywhere on the tube if they are known on a starting cube at the origin. To obtain values on the starting cube, we use different methods for the A and B.

5.2. Direct calculation of the A coefficients

When Ψ is expressed as an integral as in (1.3) and M^2 is allowed to vary over the complex plane, there is a branch cut on the negative real line from 0 to $-2D$ as that is where the pole of $f(k)$ crosses the domain of integration over k. The discontinuity across the cut is, by (5.1) and (5.2),

$$\begin{aligned} \Psi(-s^2-\mathrm{i}\varepsilon)-\Psi(-s^2+\mathrm{i}\varepsilon) &= 2\pi\mathrm{i}(\mathrm{i}s)^{D-2}F(-s^2) && D \text{ even} \\ &= -2(\mathrm{i}s)^{D-2}F(-s^2) && D \text{ odd} \end{aligned} \tag{5.7}$$

where s is real and positive and the argument x has been suppressed for brevity.

The left-hand side of (5.7) can be evaluated using

$$\frac{1}{z-\mathrm{i}\varepsilon}-\frac{1}{z+\mathrm{i}\varepsilon}=2\pi\mathrm{i}\delta(z) \tag{5.8}$$

which leads to

$$\begin{aligned} &\Psi(-s^2-\mathrm{i}\varepsilon)-\Psi(-s^2+\mathrm{i}\varepsilon) \\ &\quad=2\pi\mathrm{i}\frac{1}{(2\pi)^D}\int\prod \mathrm{d}k_\mu\,\exp\left(\mathrm{i}\sum k_\mu x_\mu\right)\delta\left(\sum(2-2\cos k_\mu)-s^2\right) \end{aligned} \tag{5.9}$$

so that (writing $\hat{F}$ for F if D is even and for $-F/\mathrm{i}\pi$ if D is odd)

$$\begin{aligned} (2\pi)^D(\mathrm{i}s)^{D-2}\hat{F}(-s^2) &= \int\prod \mathrm{d}k_\mu\,\exp\left(\mathrm{i}\sum k_\mu x_\mu\right)\delta\left(\sum 4\sin^2\tfrac{1}{2}k_\mu-s^2\right) \\ &= \int\prod_\mu \frac{\mathrm{d}s_\mu}{(1-s_\mu^2/4)^{1/2}}\prod_\mu[(1-s_\mu^2/4)^{1/2}+\tfrac{1}{2}\mathrm{i}s_\mu]^{2x_\mu}\delta\left(\sum s_\mu^2-s^2\right) \\ &= s^{D-2}\frac{\pi^{D/2}}{\Gamma(\frac{1}{2}D)}\left\langle\prod_\mu L(\tfrac{1}{2}\mathrm{i}sn_\mu, x_\mu)\right\rangle \end{aligned} \tag{5.10}$$

where $s_\mu=2\sin\frac{1}{2}k_\mu$, the average is over a unit vector n and

$$\begin{aligned} L(\sigma, x) &= (1+\sigma^2)^{-1/2}[(1+\sigma^2)^{1/2}+\sigma]^{2x} \\ &= \sum_{l=0}^{\infty}\frac{(2x+2l-1)!!}{(2x-2l-1)!!}\sigma^{2l}/(2l)! \end{aligned} \tag{5.11}$$

plus terms odd in σ.

Letting

$$a_l(x)=\frac{(2x+2l-1)!!}{(2x-2l-1)!!}=\prod_1^l[4x^2-(2j-1)^2] \tag{5.12}$$

we have

$$\left\langle \prod_\mu L(\tfrac{1}{2}Mn_\mu, x_\mu) \right\rangle = \sum_{l_1 \ldots l_D} \prod_\mu \frac{a_{l_\mu}(x_\mu) M^{2l_\mu}}{(2l_\mu)! 2^{2l_\mu}} \left\langle \prod_\mu n_\mu^{2l_\mu} \right\rangle$$

$$= \sum_{l_1 \ldots l_D} \prod_\mu \left(\frac{(2l_\mu - 1)!!}{(2l_\mu)! 2^{2l_\mu}} a_{l_\mu}(x_\mu) M^{2l_\mu} \right) \left[\left(d - 2 + 2 \sum l_\mu \right)!! \right]^{-1}$$

$$= \sum_{l_1 \ldots l_D} \prod_\mu (a_{l_\mu}(x_\mu) M^{2l_\mu} / l_\mu!) / 2^{3 \Sigma l_\mu} \left(d - 2 + 2 \sum l_\mu \right)!! \tag{5.13}$$

so that (5.10) yields

$$A_l(x) = c_0(D) \frac{1}{2^{3l}(d-2-2l)!!} \sum_{l_1 \ldots l_D}^{\Sigma l_\mu = l} \frac{1}{\Pi_\mu l_\mu!} \prod_\mu a_{l_\mu}(x_\mu) \tag{5.14}$$

where

$$c_0(D) = \frac{F}{(2\pi)^D \mathrm{i}^{D-2} F} \frac{\pi^{D/2}}{\Gamma(\frac{1}{2}D)}$$

$$= (-1)^{(D-1)/2} / (4\pi)^{D/2} (\tfrac{1}{2}D - 1)! \qquad D \text{ even}$$

$$= \tfrac{1}{4}(-1/\pi)^{(D-1)/2} / (D-2)!! \qquad D \text{ odd.} \tag{5.15}$$

From (5.14) the A_l can be obtained directly for any site, but the summation may be relatively time consuming. The method best combining speed and accuracy is to use (5.14) throughout the tube (bypassing (5.5) which tends to accumulate error) and extend to the rest of space as in § 2.3.

5.3. Starting values for B coefficients

If we use (5.6) to eliminate $B(2, 0, \ldots, 0)$, $B(2, 1, 0, \ldots, 0), \ldots, B(2, 1, \ldots, 1)$, then the Laplace equation (2.1) yields $D+1$ equations for the $(D+1)$ independent unknowns

$$\beta_l(n) = B_l(0, \underset{D-n}{\ldots}, 0, 1, \underset{n}{\ldots}, 1). \tag{5.16}$$

These equations can be written

$$2(D-n)\beta_l(n+1) + 2n\beta_l(n-1) - 2D\beta_l(n)$$

$$= (1 - n/l)\beta_{l-1}(n) \qquad D \text{ odd} \tag{5.17}$$

with an extra term related to $A_l - D/2$ on the RHS when D is even.

Unfortunately (5.17) does not determine the β_l from the β_{l-1}, because all $(D+1)$ equations are unaffected by adding a common term $\Delta\beta_l$ (independent of n) to all the β_l. Thus we need one more equation.

The other side of the coin is that, if (5.17) is multiplied by $\binom{D}{n} = D!/n!(D-n)!$ and summed over n, the β_l cancel out and we have a new equation relating the β_{l-1}. Replacing $l-1$ by l, the new equation is

$$\sum_{n=0}^{D} \binom{D}{n} (l+1-n)\beta_l(n) = 0 \qquad D \text{ odd}$$

$$= \sum_{n=0}^{D} \binom{D}{n} \frac{n^2}{l+1} \alpha_{l-(D+1)/2}(n) \qquad D \text{ even} \tag{5.18}$$

where the α_l are derived from the A_l by an equation like (5.16).

From (5.18), the ambiguous common term $\Delta\beta_l$ can be determined except in one case: when D is even and $l = D/2-1$. In that case, the coefficients on the LHS of (5.18) add up to zero, so that (5.18) is also unaffected by the addition of $\Delta\beta_l$.

In this 'intractable' case, we may note that, if the LHS of (5.17) is multiplied by $n\binom{D}{n}/4$ and summed over n, the result is the LHS of (5.18). Hence, the $\beta_{D/2-1}$ may be eliminated, giving an extra constraint on the $\beta_{D/2-2}$. For $D=4$ the extra constraint reduces to

$$\Psi(0,0,0,0)-2\Psi(1,1,0,0)+\Psi(1,1,1,1)=5/\pi^2 \tag{5.19}$$

when $M=0$. This is a check on the numerical calculation of § 3.3. However, it brings us no nearer to fixing the additive term for $\beta_{D/2-1}$.

5.4. The 'intractable' constant for even D

The preceding sections show how to find all the A_l and B_l by a combination of analytic and numerical methods, except (when D is even) for a single constant. Determining this constant is equivalent to finding the coefficient of M^{D-2} in $\Psi(0)$. We have not found any way to do this by difference equations and so we fall back on the Bessel function representation.

Writing (1.2) as

$$f(k)=\int_0^\infty \exp(-\lambda M^2)\prod_\mu \exp[-2\lambda(1-\cos k_\mu)]\,d\lambda \tag{5.20}$$

we obtain from (1.3) for $x=0$

$$\Psi(0,M^2)=\int_0^\infty \exp[-(2D+M^2)\lambda]I_0(2\lambda)^D\,d\lambda. \tag{5.21}$$

Matching this to (5.1) and differentiating $D/2-1$ times with respect to M^2, we obtain

$$(-1)^{-D/2-1}\int_0^\infty \exp[-(2D+M^2)\lambda]I_0(2\lambda)^D\lambda^{D/2-1}\,d\lambda$$
$$=(\tfrac{1}{2}D-1)!\left(A_0(0)\left(\ln\frac{1}{M^2}-\sum_1^{D/2-1}n^{-1}+B_{D/2-1}(0)\right)\right]+O\left(M^2\ln\frac{1}{M^2}\right). \tag{5.22}$$

If $M^2 \ll 1$ we may split the integral into $\lambda < 1/M$ and $\lambda > 1/M$, so that it becomes (omitting small terms)

$$\int_0^{1/M}\exp(-2D\lambda)I_0(2\lambda)^D\lambda^{D/2-1}\,d\lambda+\int_{1/M}^\infty e^{-M^2\lambda}\frac{1}{(\sqrt{4\pi\lambda})^D}\lambda^{D/2-1}\,d\lambda \tag{5.23}$$

and the second term (replacing $\lambda\to\lambda/M^2$) is

$$\frac{1}{(4\pi)^{D/2}}\int_M^\infty e^{-\lambda}\frac{d\lambda}{\lambda}=\frac{1}{(4\pi)^{D/2}}\left(\ln\frac{1}{M^2}-\int_M^{1/M}(1-e^{-\lambda})\frac{d\lambda}{\lambda}\right)$$
$$+\int_{1/M}^\infty e^{-\lambda}\frac{1}{(\sqrt{4\pi\lambda})^D}\lambda^{D/2-1}\,d\lambda. \tag{5.24}$$

Combining terms and making further negligible adjustments, (5.23) becomes

$$\frac{1}{(4\pi)^{D/2}}\left(\ln\frac{1}{M^2}+P(D)\right) \tag{5.25}$$

where

$$P(D)=\int_0^\infty [(\sqrt{4\pi\lambda}\ \mathrm{e}^{-2\lambda}I_0(2\lambda))^D-1+\mathrm{e}^{-\lambda}]\frac{\mathrm{d}\lambda}{\lambda}. \tag{5.26}$$

Matching the $\ln(1/M^2)$ term to (5.22), we obtain

$$A_0(0)=(-1)^{D/2-1}/(4\pi)^{D/2}(\tfrac{1}{2}D-1)! \tag{5.27}$$

in agreement with (5.15). The constant term gives

$$B_{D/2-1}(0)=\frac{1}{(4\pi)^{D/2}(\frac{1}{2}D-1)!}\left(\sum_1^{D/2-1} n^{-1}+(-1)^{D/2-1}P(D)\right) \tag{5.28}$$

which is the desired formula. The integral (5.26) must be calculated numerically.

For $D=2$ (5.28) gives $B_0(0)=P(2)/4\pi$. In this case an analytic result can be found directly from (1.3). Integrating over k_2 we have

$$\Psi(0,M^2)=\frac{1}{2\pi}\int_0^{2\pi}\mathrm{d}k_1[(2+M^2-2\cos k_1)(6+M^2-2\cos k_1)]^{-1/2}. \tag{5.29}$$

Setting $\cos k_1=1+\frac{1}{4}M^2(1-z)$ this becomes

$$\begin{aligned}\Psi(0,M^2)&=\frac{1}{4\pi}\int_1^{8M^{-2}+1}\frac{\mathrm{d}z}{\{(z^2-1)[(1+M^2/8-(M^2z/8)^2]\}^{1/2}}\\
&\approx\frac{1}{4\pi}\int_1^{8/M^2}\frac{\mathrm{d}z}{(z^2-1)^{1/2}[1-(M^2z/8)^2]^{1/2}}\\
&\approx\frac{1}{4\pi}\int_1^{1/M}\frac{\mathrm{d}z}{\sqrt{z^2-1}}+\int_{1/M}^{8/M^2}\frac{\mathrm{d}z}{z([1-(M^2/8)z]^2)^{1/2}}\\
&=\frac{1}{4\pi}\left\{\cosh^{-1}\frac{1}{M}+\ln[\sqrt{8}/M+(8/M^2-1)^{1/2}]\right\}\\
&\approx\frac{1}{4\pi}\left(\ln\frac{2}{M}+\ln\frac{16}{M}\right)=\frac{1}{4\pi}\ln\frac{32}{M^2}\end{aligned} \tag{5.30}$$

so that

$$B_0(0)=\frac{1}{4\pi}\ln 32. \tag{5.31}$$

5.5. Results

The determination of the A_l requires summing l^{D-1} terms (neglecting symmetry) and this is time consuming if l and d are large. We have thus limited the use of (5.14) to the tube. Then the calculation is again dominated by the extension to space, so that at each order in l, the amount of CPU time used is again 50 μs per site. We give in table 2 the first few terms in this small mass expansion for $d=2$ and 3. The coefficients grow as one goes further from the origin, but the series is convergent everywhere nevertheless. By comparing the exact values with the series, we have checked that the series converges to the right answer, so that one rapidly (in l) obtains machine accuracy.

Table 2. Coefficients of the mass expansion of the Green function in two and three dimensions. The first- and second-order terms are given and the underlined digits show where the double precision program disagrees with the exact answer.

Site	$l=1$	$l=2$
A[0, 0]	$-\frac{1}{8}$	0.019 531 250 000
A[1, 0]	$\frac{1}{8}$	−0.011 718 750 000
A[1, 1]	$\frac{3}{8}$	−0.011 718 750 000
A[10, 0]	$24+\frac{7}{8}$	151.582 031 250 000
A[10, 10]	$49+\frac{7}{8}$	615.644 531 250 000
A[20, 0]	$99+\frac{7}{8}$	2481.269 531 250 000
A[20, 20]	$199+\frac{7}{8}$	9962.519 531 250 000
B[0, 0]	−0.024 527 128 580 608 353	0.003 210 664 844 267 3390
B[1, 0]	0.044 421 496 467 095 270	−0.002 921 117 300 884 7492
B[1, 1]	0.073 581 385 741 825 059	0.000 809 076 677 831 547 68
B[10, 0]	−4.674 772 966 616 816	−16.052 591 905 483 16
B[10, 10]	−12.143 991 674 120 86	−100.291 439 009 499 1
B[20, 0]	−29.857 361 996 679 28	−542.824 843 834 596 5
B[20, 20]	−70.796 874 882 694 12	−2733.504 997 870 967
A[0, 0, 0]	$9.947\,183\,943\,243\,4585\times10^{-3}$	$-1.367\,737\,792\,195\,9755\times10^{-3}$
A[1, 0, 0]	$-3.315\,727\,981\,081\,1528\times10^{-3}$	$2.901\,261\,983\,446\,0087\times10^{-4}$
A[1, 1, 0]	$-1.657\,863\,990\,540\,5764\times10^{-2}$	$6.216\,989\,964\,527\,1616\times10^{-4}$
A[1, 1, 1]	$-2.984\,155\,182\,973\,0375\times10^{-2}$	$-3.730\,193\,978\,716\,2969\times10^{-4}$
A[2, 0, 0]	$-4.310\,446\,375\,405\,4987\times10^{-2}$	$-2.694\,028\,984\,628\,4367\times10^{-3}$
A[5, 0, 0]	−0.321 625 614 164 8718	−0.357 808 495 758 4199
A[10, 0, 0]	−1.316 344 008 489 218	−6.400 722 741 278 821
B[0, 0, 0]	$-1.216\,415\,858\,302\,2644\times10^{-2}$	$8.377\,624\,060\,630\,5671\times10^{-4}$
B[1, 0, 0]	$2.995\,767\,639\,342\,1190\times10^{-2}$	$-1.189\,597\,357\,774\,0506\times10^{-3}$
B[1, 1, 0]	$5.101\,859\,388\,164\,3107\times10^{-2}$	$1.541\,432\,309\,485\,0445\times10^{-3}$
B[1, 1, 1]	$6.554\,471\,201\,421\,8283\times10^{-2}$	$7.003\,491\,644\,003\,2347\times10^{-3}$
B[2, 0, 0]	$7.390\,018\,460\,897\,3693\times10^{-2}$	$1.581\,660\,060\,277\,3652\times10^{-2}$
B[5, 0, 0]	0.196 891 774 579 3256	0.385 781 737 478 5738
B[10, 0, 0]	0.396 886 132 191 7714	3.257 859 903 616 002

6. Conclusions

We have presented a very fast and accurate method for obtaining Green functions on an infinite lattice. The CPU requirements on a VAX 780 are 50 μs per site and the accuracy is essentially that of the machine. This numerical method can be used both for the massless and massive case. The numerical values of the Green function can be used in a real space evaluation of lattice graphs, e.g. for calculating the conductivity in the presence of weak disorder or Wilson loops at weak coupling. In addition, we have derived the small mass expansion of the Green function and have found it to be particularly useful near the origin.

Acknowledgments

Richard Friedberg thanks the Department of Physics of the University of Illinois at Urbana-Champaign for its hospitality during the final stages of this work. The work of Olivier Martin was supported by the NSF grant no PHY-82-01948.

References

[1] Doniach S and Sandheimer E H 1974 *Green's Functions for Solid State Physicists* (Reading, MA: Benjamin/Cummings)
[2] Wohlert R, Weisz P and Wetzel W 1985 *Nucl. Phys.* B **259** 85
[3] Brigham E 1974 *The Fast Fourier Transform* (Englewood Cliffs, NJ: Prentice-Hall)
[4] Hanning R W 1962 *Numerical Methods for Scientists and Engineers* (New York: McGraw-Hill)
[5] Mathews J and Walker R L 1964 *Mathematical Methods of Physics* (Reading, MA: Benjamin/Cummings) p 179