Position space calculation of a two-loop lattice diagram

1990 J. Phys. A: Math. Gen. 23 1575

(http://iopscience.iop.org/0305-4470/23/9/020)

J. Phys. A: Math. Gen. 23 (1990) 1575-1587. Printed in the UK

Position space calculation of a two-loop lattice diagram

Richard Friedberg† and Olivier Martin‡

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

‡ Physics Department, City College of City University, New York, NY 10031, USA

Received 27 April 1989

Abstract. We evaluate the finite portion, in the massless limit, of a two-loop graph built on a pair of Polyakov lines in scalar φ^3 theory on an infinite hypercubical lattice in $3+1$ dimensions. The actual calculation involves only three dimensions. We calculate in position space, using propagators obtained by a method previously described. By means of the Laplace difference equation we are able to obtain the answer as a function of r_0 (the separation of the Polyakov lines) for all r_0 in a large cube, in a single procedure. The CPU time (VAX/780) is about 6 ms per inequivalent value of r_0. Most of the time is taken by two two-dimensional discrete fast fourier transforms. However, the values calculated pertain to the infinite lattice. In our main calculation ($4\frac{1}{2}$ minutes, 45 760 inequivalent values) we obtain 8-decimal accuracy over most of the domain. This far surpasses the momentum space method in both speed and accuracy.

1. Introduction

In field theory, statistical mechanics, and non-relativistic quantum mechanics, the path integral provides a convenient starting point for calculations. Usually, it is possible to construct a perturbative series about a translation-independent soluble limit, via diagrams. These are simply expressed in Fourier (momentum) space, but of course it is possible to express them in position space also. After some manipulation, the diagrams are boiled down to integrals which often have to be treated numerically.

In continuum theories, the original multidimensional integrals can often be performed analytically by 'Feynman's trick', leaving numerical integrations only over a small number of auxiliary parameters. But in lattice physics, one is faced with a numerical integration over several d-dimensional momentum vectors. Unfortunately, the accuracy of numerical methods deteriorates rapidly with the dimension of the integral. For some diagrams this dimension is reduced by doing the computation in position space. Moreover, in lattice physics the integrals in position space are replaced by sums, which are performed much faster and produce no interpolation error.

Position space sometimes has additional advantages. The space may have boundaries or random inhomogeneities which destroy translation invariance. Or the external variables may be more naturally expressible in position space, as in the calculation of Wilson loops with complicated shapes.

This paper is part of a programme begun earlier [1] which explores the advantages of calculating in position space, especially in lattice-based theories. We consider here a simple example in lattice-based $(3+1)$-dimensional scalar φ^3 theory. We calculate a two-loop diagram (figure 1) relating two Polyakov lines extending from $-\infty$ to ∞ in

0305-4470/90/091575+13$03.50

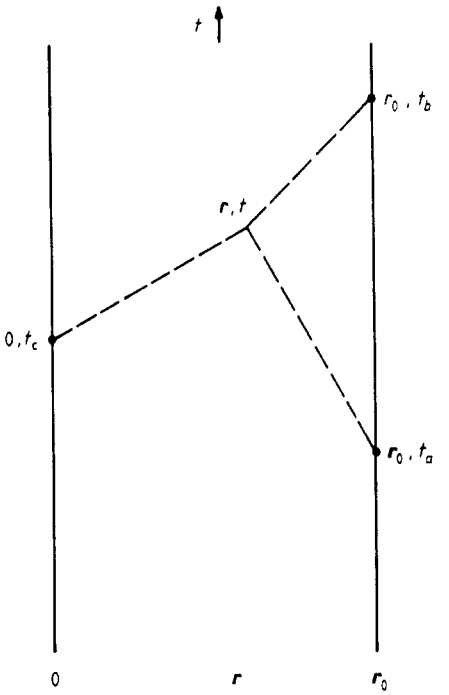

Figure 1. 'Spider' graph between two Polyakov lines parallel to the time direction.

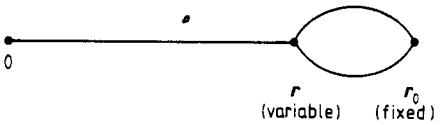

Figure 2. Three-dimensional graph equivalent to figure 1. The propagators in four dimensions have been summed over time to yield G_M of (3.2).

the 'time' direction, with arbitrary relative placement in three-dimensional 'space'. Since one is to sum over all locations on the lines, the propagators have zero momentum in the t direction and therefore reduce to the standard three-dimensional propagators in (x, y, z) space. Consequently, our task amounts to calculating the graph of figure 2 on a three-dimensional lattice. We take the lattice to be cubic and study the massless limit. We will show that if this problem is done in position space one can obtain results of far greater accuracy, in far less CPU time, than from a calculation in momentum space.

2. Statement of the problem

For non-zero mass M, the quantity to be evaluated would be

$$F_M(\boldsymbol{r}_0) = \oint \frac{d^3\boldsymbol{k}_a}{(2\pi)^3} \oint \frac{d^3\boldsymbol{k}_b}{(2\pi)^3} \frac{\exp[\mathrm{i}(\boldsymbol{k}_a + \boldsymbol{k}_b)\cdot \boldsymbol{r}_0]}{[Q(\boldsymbol{k}_a)+M^2][Q(\boldsymbol{k}_b)+M^2][Q(\boldsymbol{k}_a+\boldsymbol{k}_b)+M^2]} \tag{2.1}$$

where $\boldsymbol{r}_0$ (a vector with integer components) is the separation of the two fixed points, the integrals are from $-\pi$ to π, and

$$Q(\boldsymbol{k}) = \sum_{i=1}^{3} (2 - 2\cos k_i). \tag{2.2}$$

The corresponding continuum quantity would be

$$\bar{F}_M(\boldsymbol{r}_0) = \int \frac{d^3\boldsymbol{k}_a}{(2\pi)^3} \int \frac{d^3\boldsymbol{k}_b}{(2\pi)^3} \frac{\exp[\mathrm{i}(\boldsymbol{k}_a + \boldsymbol{k}_b)\cdot \boldsymbol{r}_0]}{[k_a^2 + M^2][k_b^2 + M^2][(\boldsymbol{k}_a + \boldsymbol{k}_b)^2 + M^2]} \tag{2.3}$$

where the integrals are now over all k-space. As $M \to 0$, one can show that

$$\bar{F}_M(\boldsymbol{r}_0) = \frac{1}{(4\pi)^2}\left[\ln\left(\frac{1}{3M|\boldsymbol{r}_0|}\right) - \gamma + 1\right] + \mathrm{O}[M] \tag{2.4}$$

where γ is Euler's constant. In view of the form of the infrared divergence, the term in $\ln M$ must be the same on the lattice. Therefore

$$F_M(\boldsymbol{r}_0) \simeq \frac{1}{(4\pi)^2}\left(\ln\frac{1}{M} + f(\boldsymbol{r}_0)\right) \tag{2.5}$$

as $M \to 0$. We wish to calculate f. (In lattice gauge theory, the infrared divergence cancels when one sums over all diagrams. Therefore it would suffice to calculate for each diagram the quantity analogous to f.)

3. Propagator in position space

The expression (2.1) can also be written as

$$F_M(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}} G_M(\boldsymbol{r})^2 G_M(\boldsymbol{r}_0 - \boldsymbol{r}) \tag{3.1}$$

where $\boldsymbol{r}$ is summed over the lattice, and

$$G_M(\boldsymbol{r}) = \oint \frac{d^3\boldsymbol{k}}{(2\pi)^3} \frac{\mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}}{[Q(\boldsymbol{k}) + M^2]}. \tag{3.2}$$

Since G_M satisfies

$$\mathrm{D}^2 G_M(\boldsymbol{r}) = -\delta_{\boldsymbol{r}} + M^2 G_M(\boldsymbol{r}) \tag{3.3}$$

where D^2 is the Laplace difference operator, we have

$$\mathrm{D}^2 F_M(\boldsymbol{r}) = -G_M(\boldsymbol{r})^2 + M^2 F_M(\boldsymbol{r}) \tag{3.4}$$

or, taking the limit $M \to 0$,

$$\mathrm{D}^2 f(\boldsymbol{r}) = -g(\boldsymbol{r})^2 \tag{3.5}$$

where

$$g(\boldsymbol{r}) = 4\pi \lim_{M\to 0} G_M(\boldsymbol{r}) = 4\pi \oint \frac{d^3\boldsymbol{k}}{(2\pi)^3} \frac{\mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}}{Q(\boldsymbol{k})}. \tag{3.6}$$

In a previous paper [1], we have shown how to calculate the whole array of values of g nearly to machine accuracy ($\sim$15 decimals in double precision) with a program requiring less than 100 μs CPU time on a VAX/780 for each value of $\boldsymbol{r}$. Thus we may regard g as given, over any desired range of $\boldsymbol{r}$.

4. Boundary value problem

We are to solve (3.5) with the condition that for $r \to \infty$, f should grow more slowly than r. This actually means that $f \approx \ln(1/3r) - \gamma + 1$, in agreement with the continuum solution (2.4). We considered several alternative methods.

The most direct way is to sum (3.1) for $M = 0$, making a term by term subtraction to cancel the infrared divergence. Apart from the complexity of separately evaluating the sum of the subtractions, the subtracted series still converges very slowly (error $\sim 1/R$) and one needs considerable time for high accuracy.

An attractive idea is to solve (3.5) in momentum space by means of the fast Fourier transform (FFT) [2]. However, this method always introduces some kind of boundary conditions—periodic, Dirichlet, or Neumann—on the surface of some cube surrounding the origin. This would seriously distort our solution since it would amount to placing a sublattice of image sources all over space. The quantity we are trying to calculate pertains to the *infinite* lattice. However, a simple modification of FFT allows us to solve (3.5) in a cube with *inhomogeneous* boundary conditions. Thus, if we know the true $f(r)$ on the surface—this can be found by an asymptotic series in $1/r$ which is much faster than a direct sum—we can find its values quickly and accurately inside the cube. This is the method we have used. It would also be possible, given the boundary values of f, to solve (3.5) by relaxation. But this is slower and less accurate than FFT.

5. Asymptotic series

For any well-behaved function, the Taylor expansion shows that

$$D^2 = \nabla^2 + \frac{1}{12}\left(\frac{d^4}{dx^4} + \frac{d^4}{dy^4} + \frac{d^4}{dz^4}\right) + \dots \tag{5.1}$$

successive terms falling off as powers of $1/r^2$. This allows us to construct the following table, valid for large r:

$$D^2 \ln\frac{1}{r} = -\left(\frac{1}{r^2} + (\tfrac{1}{10} - \tfrac{8}{5}\boldsymbol{P}_4(\hat{r}))\frac{1}{r^4} + \dots\right) \tag{5.2}$$

$$D^2 \frac{1}{r} = \tfrac{7}{2}\boldsymbol{P}_4(\hat{r})/r^5 + \dots \tag{5.3}$$

$$D^2 \frac{1}{r^2} = 2/r^4 + \dots \tag{5.4}$$

$$D^2 \frac{\boldsymbol{P}_4(\hat{r})}{r^2} = -18\boldsymbol{P}_4(\hat{r})/r^4 + \dots \tag{5.5}$$

$$D^2 \frac{\boldsymbol{P}_4(\hat{r})}{r^3} = -14\boldsymbol{P}_4(\hat{r})/r^5 + \dots \tag{5.6}$$

where

$$\boldsymbol{P}_4(\hat{r}) = r^{-4}(x^4 + y^4 + z^4 - 3x^2y^2 - 3x^2z^2 - 3y^2z^2) \tag{5.7}$$

is the unique spherical harmonic of order 4 possessing cubic symmetry and normalised to $\boldsymbol{P}_4(\hat{\boldsymbol{z}}) = 1$. From (3.6) and (3.3), $g(\boldsymbol{r})$ satisfies

$$\mathrm{D}^2 g(\boldsymbol{r}) = -4\pi\delta_{\boldsymbol{r}} \tag{5.8}$$

and vanishes far away; hence, by Gauss' law, $g \sim 1/r + \ldots$. Then comparing (5.3) and (5.6), we see that (5.8) is satisfied more nearly by

$$g(\boldsymbol{r}) = \frac{1}{r} + \tfrac{1}{4}\boldsymbol{P}_4(\hat{\boldsymbol{r}})/r^3 + \ldots \tag{5.9}$$

Squaring (5.9) and substituting into (3.5), we find by combining (5.2), (5.4) and (5.5) that (3.5) would be satisfied to $\mathrm{O}(1/r^4)$ if f were taken equal to

$$f_0(\boldsymbol{r}) = \ln\frac{1}{3r} - \gamma + 1 + \frac{(\frac{1}{20} + \frac{7}{60}\boldsymbol{P}_4(\hat{\boldsymbol{r}}))}{r^2}. \tag{5.10}$$

But since we are looking only at the part of space far from the origin, there is also a homogeneous solution $g(\boldsymbol{r})$. Therefore, we can only say that

$$\begin{aligned} f(\boldsymbol{r}) &= f_0(\boldsymbol{r}) + Ag(\boldsymbol{r}) + \ldots \\ &= \ln\frac{1}{3r} - \gamma + 1 + A/r + r^{-2}(\tfrac{1}{20} + \tfrac{7}{60}\boldsymbol{P}_4(\hat{\boldsymbol{r}})) + \frac{A}{4} r^{-3}\boldsymbol{P}_4(\hat{\boldsymbol{r}}) + \mathrm{O}[r^{-4}] \end{aligned} \tag{5.11}$$

where A is to be determined.

6. Determination of A

To find A we return to (3.1) and to the continuum analogue obtained from (2.3)

$$\bar{F}_M(\boldsymbol{r}_0) = \int \mathrm{d}^3\boldsymbol{r}\left(\frac{\mathrm{e}^{-Mr}}{4\pi r}\right)^2\left(\frac{\mathrm{e}^{-M|\boldsymbol{r}_0-\boldsymbol{r}|}}{4\pi|\boldsymbol{r}_0-\boldsymbol{r}|}\right). \tag{6.1}$$

By comparing (2.4) with (2.5), we see that $A/(4\pi)^2$ is the coefficient of $1/r_0$ in the leading term (for $r_0 \to \infty$) of

$$\Delta = \lim_{M\to 0}(F_M - \bar{F}_M). \tag{6.2}$$

This leading term can be found by replacing $|\boldsymbol{r}_0 - \boldsymbol{r}|$ by r_0 in both (3.1) and (6.1). (Note in particular that for $r \gg r_0$ the difference between the integrands of (3.1) and (6.1) diminishes as $(1/r^3)(1/r)(1/r)$, so that the contribution from large r is of order $\int \mathrm{d}^3\boldsymbol{r}/r^5$, or $1/r^2$.) Making this replacement, and taking $G_M(\boldsymbol{r}_0) \approx \mathrm{e}^{-Mr_0}/4\pi r_0$, we find that

$$A = \lim_{M\to 0}\frac{1}{4\pi}\left(\sum_{\boldsymbol{r}}(4\pi G_M(\boldsymbol{r}))^2 - \int \mathrm{d}^3\boldsymbol{r}(\mathrm{e}^{-Mr}/r)^2\right) \tag{6.3}$$

and we write $\sum (G_M(\boldsymbol{r}))^2$ as $H_M(0)$, where

$$H_M(\boldsymbol{R}) = \sum_{\boldsymbol{r}} G_M(\boldsymbol{r})G_M(\boldsymbol{R}-\boldsymbol{r}) = \oint \frac{d^3\boldsymbol{k}}{(2\pi)^3}\frac{\mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{R}}}{(Q(\boldsymbol{k}) + M^2)^2} \tag{6.4}$$

From an integration by parts one may obtain (see equations (2.5) and (2.6) of [1])

$$H_M(\boldsymbol{R} + \hat{\boldsymbol{x}}) - H_M(\boldsymbol{R} - \hat{\boldsymbol{x}}) = -R_x G_M(\boldsymbol{r}) \tag{6.5}$$

and therefore

$$\sum_r [G_M(r)]^2 = \sum_0^\infty (2n+1) G_M(2n+1, 0, 0). \tag{6.6}$$

On the other hand, a direct integration yields

$$\int d^3r \left(\frac{e^{-Mr}}{r}\right)^2 = \frac{4\pi}{2M} = 4\pi \sum_0^\infty e^{-(2n+1)M} + O(M). \tag{6.7}$$

Subtracting term by term, we can now write (6.3) as

$$A = \lim_{M\to 0} \sum_0^\infty [4\pi(2n+1) G_M(2n+1, 0, 0) - e^{-(2n+1)M}] \tag{6.8}$$

and since the series now converges uniformly in M (the summand falls as $e^{-(2n+1)M}/(2n+1)^2$), we may interchange limits, obtaining

$$A = \sum_0^\infty [(2n+1) g(2n+1, 0, 0) - 1] = 0.152\,859\,3250. \tag{6.9}$$

(Ten-decimal accuracy was obtained after 50 terms by a double Richardson extrapolation [3]).

7. Solution to the difference equation

Our algorithm actually uses FFT only in two directions, and switches to a faster one-dimensional method for the third. Consider a cube of radius $R = 2^m$ (side $2R$) centred at the origin. For $|x|$, $|y|$, $|z|$ all $< R$, let us express

$$[g(x, y, z)]^2 = 3 \sum_{k_x, k_y} s(k_x, k_y, z) \cos(k_x x) \cos(k_y y) \tag{7.1}$$

where k_x, k_y range over odd integers from 1 to $2R-1$. Suppose that for $|x|$, $|y|$ both $< R$, we know that

$$f(x, y, R) = 3 \sum_{k_x, k_y} b(k_x, k_y) \cos(k_x x) \cos(k_y y). \tag{7.2}$$

Then if for each k_x, k_y we can find, for $-R < z < R$, a function $\varphi(k_x, k_y, z)$ satisfying

$$\varphi(z+1) + \varphi(z-1) - 2\varphi(z) = \lambda\varphi(z) - s(z) \tag{7.3}$$

$$\varphi(-R) = \varphi(R) = b \tag{7.4}$$

where the arguments k_x, k_y are understood, and

$$\lambda = 4 - 2\cos k_x - 2\cos k_y \tag{7.5}$$

the function

$$\hat{f}(x, y, z) = \sum_{k_x, k_y} \varphi(k_x, k_y, z) \cos(k_x x) \cos(k_y y) \tag{7.6}$$

will satisfy $D^2\hat{f} = -\frac{1}{3}g^2$ inside the cube and will take the correct values of f for $|z| = R$, but will vanish for $|x| = R$ or $|y| = R$ because of the Fourier expansion used. The true function f is therefore

$$f(x, y, z) = \hat{f}(x, y, z) + \hat{f}(y, z, x) + \hat{f}(z, x, y). \tag{7.7}$$

We start by finding g inside the cube by the method of [1], and f on the surface from (5.11) with (6.9). We then determine s and b from (7.1) and (7.2) by FFT. (Our FFT routine has symmetries built in so that its intermediate arrays contain only R real numbers, not $2R$ complex numbers.)

We solve (7.3) and (7.4) by a one-dimensional Green's function method:

$$\varphi(z) = -\sum_{-(R-1)}^{R-1} q(z, z_1)\bar{s}(z_1) \tag{7.8}$$

where $\bar{s} = s + b$ if $|z| = R-1$, $\bar{s} - s$ otherwise, and (for $z_> > z_<$)

$$q(z_>, z_<) = q(z_<, z_>) = \frac{\sinh \mu(R - z_>) \sinh \mu(R + z_<)}{\sinh \mu \sinh 2\mu R} \tag{7.9}$$

with

$$2(\cosh \mu - 1) = \lambda. \tag{7.10}$$

This is done in a single sweep (hence, faster than FFT which requires m sweeps) by summing $\sum e^{n\mu}\bar{s}(R-n)$ and $\sum e^{-n\mu}\bar{s}(R-n)$ comulatively and using the partial sums to evaluate (7.8). Having found φ, we obtain f from (7.6) and (7.7) via a reverse FFT routine.

8. Results

Table 1 shows the computing time required for $R = 8$, 16, 32 and 64. The total time grows as $R^3 \log_2 R$ and is mainly consumed by FFT. This time (4s for $R = 16$, $4\frac{1}{2}$min for $R = 64$) is how long it takes to find f for the *whole cube* containing $R(R+1)(R+2)/6$ distinct sites unrelated by symmetry. The time per site thus grows only logarithmically (6 ms for $R = 64$).

Table 1. Computing time required for different values of R.

R = Radius of cube	Number of inequivalent sites	CPU time (s)	Time/site (ms)
8	120	0.8	6.7
16	816	4.0	4.1
32	5 984	30	5.0
64	45 760	264	5.8

We also note that our program receives no numerical input data of any kind. The CPU times given therefore include the time consumed by a preliminary program that computes g, as well as routines that find π (by an algorithm in [1]), γ (by double Richardson extrapolation) and A. These preliminary times are negligible compared to the total time for $R > 8$.

In table 2 we compare results for $r = 0$, for different values of R. The discrepancies are quite compatible with a residual error $1/R^4$ as expected in (5.11). This fact confirms the accuracy of both (5.11) and (6.9). We also compare our best value of $f(0)$,

Table 2. Results at $r=0$ for various different values of R.

	$f_R(0)$	$\Delta f = f_R(0) - f_\infty(0)$ $(\times 10^{-10})$	$R^4\Delta f$
$R=8$	2.380 231 9818	−76 212	−0.0312
$R=16$	2.380 239 1509	−4 521	−0.0295
$R=32$	2.380 239 5753	−277	−0.0291
$R=64$	2.380 239 6013	−17	−0.028 [6]
$R=\infty$ (extrap)	2.380 239 6030		
Independent calculation	2.380 239 6030		

extrapolated from $R=32$ and 64, with the result of an independent calculation in which $f(0)$ is written as

$$f(0) = \lim_{M\to 0}\left((4\pi)^2 \sum_r G_M(r)^3 - \ln\frac{1}{M}\right) = f_1 + f_2 \tag{8.1}$$

where

$$f_1 = \frac{1}{4\pi}\sum_r \left(g(r)^3 - \frac{1}{r^3}\right) \tag{8.2}$$

$$f_2 = \lim_{M\to 0}\left(\frac{1}{4\pi}\sum_r \frac{e^{-3Mr}}{r^3} - \ln\frac{1}{M}\right). \tag{8.3}$$

We approximate f_1 by summing (8.2) directly over a spherical volume centred on the origin. From (5.9) one sees that the summand diminishes as $1/r^5$ and that its leading term averages to zero on spherical integration. The leading spherically symmetric term goes as $1/r^7$, and hence contributes a truncation error of $1/R^4$ (R = radius of the sphere). But the coefficient of this term can be found and its contribution from outside the sphere estimated as an integral. When this contribution is added, the result exhibits a nearly drift-free variation with R, due to the graininess of the sum. This variation can be largely reduced by softening the truncation cut-off. Thus we obtain f_1 to ten decimal places from a sphere with a radius of about 45.

For f_2 we use the formula

$$f_2 = -1 + \frac{1}{7}\left(6\ln 2 + \sum_{m_1 m_2} C_{m_1 m_2} \ln\coth\left(\tfrac{1}{2}\pi\sqrt{m_1^2+m_2^2}\right)\right) \tag{8.4}$$

where m_1 and m_2 range over integers from $-\infty$ to ∞ with the exclusion of $m_1 = m_2 = 0$, and

$$C_{m_1 m_2} = 3 + 3(-1)^{m_1} + (-1)^{m_1+m_2} \tag{8.5}$$

obtained by one of us [4] through an elaboration of the methods of Madelung [5]. This formula sums over only two dimensions (in the dual space) and converges exponentially in the radius, so that we easily obtained ten decimal places.

The excellent agreement between the resulting value of $f(0)$ and that extrapolated from our main calculation once more confirms the accuracy of (5.11) and (6.9). We have also studied the error in $f(r)$ due to box size at points away from the origin. We find that the discrepancy between $R=32$ and $R=64$ is comparable to that at the origin except near the boundary of the smaller cube, where it becomes several times greater. (See table 3.)

Table 3. $\Delta f = f_{32} - f_{64}$ ($\times 10^{-9}$) for various positions.

n	$\Delta f(0, 0, n)$	$\Delta f(0, n, n)$	$\Delta f(n, n, n)$
0	−26	−26	−26
8	−27	−25	−23
16	−33	−18	0
24	−76	−7	+23
32	−307	−19	+14

Table 4. Values of f at various positions for $R = 64$.

n	$f(0, 0, n)$	$f(0, n, n)$	$f(n, n, n)$
0	2.380 239 601		
1	0.699 170 588	0.194 563 494	−0.058 629 810
2	−0.133 147 848	−0.558 648 209	−0.779 172 730
3	−0.599 097 757	−0.984 918 390	−1.197 086 332
4	−0.912 310 752	−1.282 349 719	−1.491 461 189
8	−1.634 780 550	−1.989 558 018	−2.195 089 390
12	−2.048 187 988	−2.399 617 137	−2.604 141 837
16	−2.339 585 215	−2.689 582 291	−2.893 632 281
24	−2.748 607 256	−3.097 321 521	−3.300 914 889
32	−3.038 010 480	−3.386 137 324	−3.589 509 000
40	−3.262 168 749	−3.609 960 048	−3.813 200 797
48	−3.445 159 367	−3.792 733 942	−3.995 888 273
56	−3.599 784 337	−3.947 207 502	−4.150 300 528
64	−3.733 669 485	−4.080 980 935	−4.284 028 213

In table 4 we show some results for $R = 64$. (These are selected from the whole array of 45 760 distinct numbers, all of which were calculated in a $4\frac{1}{2}$ min run.) On the basis of tables 2 and 3, we can make a conservative estimate that the absolute (not relative) error in these numbers is less than 3×10^{-9} for $|r| < 32$, and less than 3×10^{-8} everywhere.

We have not attempted to achieve greater accuracy in table 4 by extrapolation from smaller R, because we know that we could get several more decimal places anyway by extending (5.11) to higher order. Since this series is asymptotic there would eventually be diminishing returns at any fixed R, but a study of the results obtained by truncating (5.11) at lower orders indicates that at $R = 64$ we should get considerably more accuracy for at least the next few terms.

9. The momentum space integral

It is startling to compare our results with those of the usual momentum space methods. Consider a direct calculation of the six-dimensional integral (2.1). To regularise this expression as $M \to 0$, it is best to subtract out the leading singularities and integrate these analytically. A natural candidate is the integrand of (2.3), but the domain of integration is different.

In practice, it is not possible to get rid of all the singularities, and one is left with integrable singularities inside the domain of interest. This lowers the accuracy achievable through numerical integration: as one attempts to obtain higher accuracy, the computation is bogged down at these singularities. A second problem is the oscillating nature of the integrand: for r_0 not very small, it is necessary to include many points to take into account the cancellations. When the dimension of the integral is six or greater, this number of sampling points becomes prohibitive. Thus, it is just not possible with this method to get accurate values for large Wilson loops or lines. This was pointed out in [6] where Wilson loop averages were calculated by numerical integration, and even 6×6 loops could not be obtained to better than 1 percent accuracy. In addition, the integral (2.1) must be recalculated for each value of r_0, whereas in the position space method, all values of r_0 are treated in one shot.

In terms of CPU times, the position space calculation treats 10^4 values of r_0 with eight-decimal accuracy in about 6 ms of VAX/780 CPU time per r_0, whereas the direct evaluation of (2.1) with three-decimal accuracy at the origin takes seconds, and much more away from $r_0 = 0$.

A second approach to the momentum integral involves transforming the integral to a lower dimension. Equation (2.1) can be rewritten as

$$F_M(r_0) = \oint \frac{d^3k_a}{(2\pi)^3} \oint \frac{d^3k_b}{(2\pi)^3} \exp[i(k_a + k_b)\cdot r_0] \times \int_0^\infty\int\int d\lambda_1\, d\lambda_2\, d\lambda_3 \exp[-\lambda_1(Q(k_a)+M^2)] \times \exp[-\lambda_2(Q(\bar{k}_b)+M^2)] \exp[-\lambda_3(Q(k_a+k_b)+M^2)] \tag{9.1}$$

so that

$$F_M(r_0) = \int_0^\infty\int\int d\lambda_1\, d\lambda_2\, d\lambda_3 \times \exp\left(-(6+3M^2)\sum_1^3 \lambda_i\right) F(\{\lambda\}, x) F(\{\lambda\}, y) F(\{\lambda\}, z) \tag{9.2}$$

with $r_0 = (x, y, z)$ and

$$F(\{\lambda\}, u) = \oint \frac{dk}{2\pi} \oint \frac{dk}{2\pi} \exp(2\lambda_1 \cos k) \exp(2\lambda_2 \cos k') \times \exp[2\lambda_3 \cos(k+k')] \exp[i(k+k')u]. \tag{9.3}$$

In this last expression, one of the momentum integrals can be expressed as a Bessel function. It is thus possible to reduce (2.1) to a four-dimensional integral. The main advantage of this procedure, besides reducing the dimension of the integral, is to concentrate all of the oscillations into one dimension where they can be evaluated accurately, so that working with large r_0 is not so bad. One has to recalculate the three-dimensional integral for each choice of r_0, as in the previous treatment. Finally, it is necessary to regularise the integral by subtraction. This leads to integrable singularities as in the direct approach, and reduces the accuracy of the numerical calculation. Thus we find that even a single value of r_0 takes longer than to calculate the whole array by the real space method for comparable accuracy.

10. Extensions

In this section we shall merely sketch some possible procedures for treating harder problems along the lines of the present paper.

Suppose we wished to treat some more complicated theory, such as a lattice gauge theory. Our hope is that this would merely necessitate a new factor in the numerator of the vertex function. If this factor is trigonometric in the momenta, it can probably be replaced by a sum over nearest-neighbour values of the propagator. In that case the calculation becomes the same as for a scalar field, only repeated several times. Thus the spider graph on two Wilson lines, which in scalar theory required $4\frac{1}{2}$ min for all r_0 having components <64, might still require less than an hour (VAX/780 CPU) in gauge theory.

Returning to scalar theory, suppose we wished to treat the loop insertion diagram of figure 3. Writing t for the fourth lattice component (parallel to the two Wilson lines), and $G_{M(r,t)}$, $g_{(r,t)}$ for the four-dimensional generalisations of G_M and g, we have to replace (3.1) by

$$F_M(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}_1} \sum_{\boldsymbol{r}_2} H_M(\boldsymbol{r}_0 - \boldsymbol{r}_1 - \boldsymbol{r}_2) G_M(\boldsymbol{r}_1) G_M(\boldsymbol{r}_2) \tag{10.1}$$

where

$$H_M(\boldsymbol{r}) = \sum_{t=-\infty}^{\infty} G_M(\boldsymbol{r}, t)^2 \tag{10.2}$$

so that after reasoning similar to that of section 3 we have

$$\mathrm{D}^2(\mathrm{D}^2 f(\boldsymbol{r})) = h(\boldsymbol{r}) \tag{10.3}$$

where

$$h(\boldsymbol{r}) = \sum_{-\infty}^{\infty} g(\boldsymbol{r}, t)^2. \tag{10.4}$$

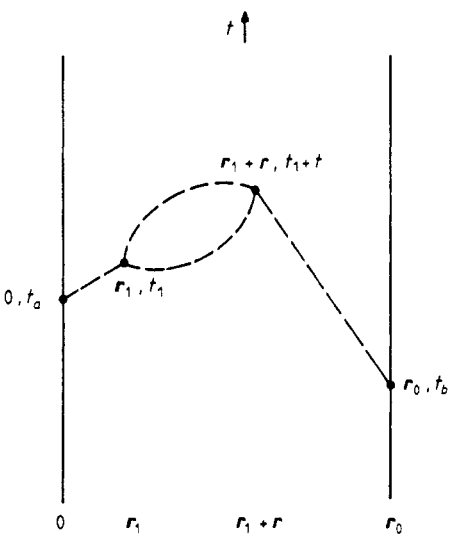

Figure 3. Loop insertion graph between two Polyakov lines. The two propagators terminating on the lines may be summed separately over time, but the two inner ones must be summed together as in (10.2).

Once $h(\boldsymbol{r})$ is known, (10.3) can be solved in three dimensions by the methods of the present paper, with hardly any more computing time if superfluous FFT are avoided. Thus the new problem is to compute (10.4) at each $\boldsymbol{r}$. Since the summand falls off as $(r^2+t^2)^{-2}$, one could obtain six-decimal accuracy by summing up to $t=100$ for each $\boldsymbol{r}$. But the time can be greatly reduced by summing the difference between $g(\boldsymbol{r}, t)^2$ and its asymptotic expression; then $t=16$ would be sufficient for six decimals. It would remain to evaluate $\sum_{-\infty}^{\infty}(r^2+t^2)^{-2}$ for each value of $\boldsymbol{r}$. (For $\boldsymbol{r}=0$, the origin may be excluded and the sum is $2\zeta(4)=\pi^4/45$.) But since this sum depends on $\boldsymbol{r}$ only through the integer r^2, the number of sums is reduced. Thus for $R=64$ in section 8, there are only about 10 000 distinct values of r^2. Morever, one can make the continuum approximation

$$\sum_{-\infty}^{\infty}(r^2+t^2)^{-2}\approx\int_{-\infty}^{\infty}\frac{\mathrm{d}t}{(r^2+t^2)^2}=\frac{\pi}{2r^3} \tag{10.5}$$

for all but the smallest values of r, since the error can be shown to fall off as $\pi^2\,\mathrm{e}^{-2\pi r}/r^2$. Taking all this into consideration, we would expect that the calculation for $R=64$ could be performed in less than half an hour of VAX/780 CPU time.

Now, returning to the spider graph, suppose we wish to compute a Wilson loop of arbitrary shape in four dimensions. Having found $g(r)$ in a four-dimensional box (r now denotes $(\boldsymbol{r}, t)$) we may compute

$$j(r)=\sum_{r_w} g(r-r_w) \tag{10.6}$$

where r_w varies over the loop. Then the quantity sought is $\sum j(r)^3$ which converges as $\int \mathrm{d}^4r/r^6$. But the leading term can be evaluated by other techniques, and the increment converges as $\int \mathrm{d}^4r/r^8$, so that one can obtain high accuracy from a box with $R=64$.

For rectangular loops one may save time by computing (10.6) for line segments of various lengths. If L is the maximum length, this requires only R^4L steps; then one has a library from which in R^4 steps one finds $j(r)$ for any rectangle and sums $j(r)^3$. In this way one might calculate a great number of rectangular loops in about 20 min of VAX/780 CPU time for each loop.

Even more efficiency might be obtained by writing the desired quantity as $\sum_{r_w} f(r_w)$ where

$$f(r)=\sum_{r} g(r-r)j(r)^2. \tag{10.7}$$

Since of any three points on a rectangle, two must be on the same or opposite sides, these two can be taken as the sources of j^2. The part of j^2 due to two points on the same line segment can be calculated for each segment length; this need not be repeated for each loop. The part due to two points on parallel segments might be done in three dimensions of position space and one (parallel to the segments) of momentum space; this momentum would appear as a mass in the three-dimensional propagator, which can be found by the methods of [1]. For wide loops only small masses would contribute.

Finally, consider the three-dimensional problem treated in the present paper, but within some irregular volume with Dirichlet boundary conditions. Here the methods of [1] cannot be used to find $g(\boldsymbol{r}, \boldsymbol{r}')$; it must be done by relaxation, possibly assisted by a FFT with artificial sources at the boundary. However, once $g(\boldsymbol{r}, \boldsymbol{r}')$ is found for all $\boldsymbol{r}'$ and a fixed $\boldsymbol{r}$, one can find $f(\boldsymbol{r}, \boldsymbol{r}')$ satisfying $\mathrm{D}^2f=-g^2$ by the same methods as presented in the previous sections.

11. Conclusion

We have explored some of the advantages of the position space method for calculating diagrams. In position space one can deal effectively with complicated Wilson loop shapes or irregular boundaries, but even in the absence of these complications the position space calculation may be faster and more accurate. We illustrated this last feature by the line-line correlation function in a scalar lattice theory (cf figure 1). We obtained an accuracy far beyond what can be hoped for in momentum space calculations, and achieved great speed by finding the value of the correlation function for many separations at once. In our calculation ($R = 64$) we obtained the value of figure 1 on the lattice with eight-decimal accuracy in 6 ms CPU time for each of 45 760 inequivalent values of $\boldsymbol{r}_0$. The accuracy was undiminished for most values of $\boldsymbol{r}_0$ with all components <64. It should be understood that position space will not *always* be the better method. In particular, for complicated diagrams with third-order vertices, the momentum space integral tends to be of lower dimension. But the enormous advantage of position space in both speed and accuracy in the calculation presented here makes it probable that the superiority of position space extends to a good many situations.

Acknowledgment

This research is sponsored in part by the Department of Energy of the USA.

References

[1] Friedberg R and Martin O 1987 *J. Phys. A: Math. Gen.* **20** 5095
[2] Brigham E O 1988 *The Fast Fourier Transform and its Applications* (Englewood Cliffs, NJ: Prentice-Hall)
[3] Hildebrandt F B 1956 *Introduction to Numerical Analysis* (New York: McGraw-Hill)
[4] Friedberg R 1986 unpublished
[5] Glasser M L and Zucker I J 1980 *Theoretical Chemistry: Advances and Perspectives* ed H Eyring and D Henderson (New York: Academic) p 67
[6] Wohlert R, Weisz P and Wetzel W 1985 *Nucl. Phys.* B **259** 85