A new proposal for the fermion doubling problem

John P. Costella

Faculty of Mathematics, Mentone Grammar, 63 Venice Street, Mentone, Victoria 3194, Australia
jpcostella@hotmail.com; jpc@mentonegs.vic.edu.au; jpc@physics.unimelb.edu.au

(19 July 2002)

Abstract

In this paper I propose the use of a lattice derivative operator that is equivalent to the ideal SLAC derivative operator in all lattice calculations, but without the prohibitively expensive computational cost of the latter. A pedagogical motivation and derivation of the closed form of the SLAC derivative in position space is presented, and the proposed method for its cost-effective implementation is presented in detail.

1. Pedagogical overview of the fundamental reasons for fermion doubling

Perhaps the most important ingredient in any good lattice calculation is the fundamental building-in of as many of the symmetries of the continuum formalism being modelled as possible, so that the results respect the given symmetries identically, rather than only in the continuum limit. However, the conflicting requirements of these symmetries may, in turn, introduce subtle artefacts into the formalism that destroy its physical correspondence; and these artefacts can in some cases be difficult to exorcise.

The fermion doubling problem is a particularly notorious example of such an artefact. It arises from the apparently simple requirement that we construct a spatial derivative operator on the lattice—even in one dimension. In the first instance, one might think that the first-principles definition of the derivative taught to school children,

$$\frac{df}{dx} \equiv \lim_{a \to 0} \frac{f(x+a) - f(x)}{a}, \tag{1}$$

might be an ideal candidate for the lattice derivative operator: we need simply take a to be the lattice spacing, and the limit will automatically be taken when we extrapolate to the continuum limit of the calculations. In fact, in a huge number of computational applications in engineering, such a definition is perfectly acceptable, and is used every day without complication.

The problem arises if we wish to perform any calculation in quantum physics. Simple nonrelativistic quantum mechanics is enough to highlight the difficulties. There, we wish to construct a free-particle momentum operator $p \equiv -i\,d/dx$. (Throughout this paper, I use units in which $\hbar = 1$.) However, since we want the momentum to be an observable, its operator representation must (by the principles of quantum mechanics) be Hermitian. This, in turn, implies that the operator d/dx must be *anti-Hermitian*. What does this mean in the context of a one-dimensional lattice calculation? Think of the nonrelativistic wavefunction

$\psi(x)$ as a column vector, each element of which is a complex number, representing the value of the wavefunction at the corresponding lattice site. The conjugate wavefunction $\psi^*(x)$ can be represented as the Hermitian conjugate of this column vector (which we should probably denote $\psi^\dagger(x)$ instead). Operators are sandwiched between $\psi^\dagger$ and ψ, so must be matrices; each matrix element determines how ψ at some x is coupled to ψ^* at some other x'.

Anti-Hermiticity of the derivative operator then simply requires that its matrix representation be anti-Hermitian. The first-principles prescription (1), however, corresponds to an operator of the form

$$\frac{1}{a}\begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix},$$

which is clearly not antisymmetrical, as is required if a real matrix is to be anti-Hermitian.

A decision must therefore be made. Either one must give up the simple derivative operator (1), or else one must give up the Hermiticity of the momentum operator. The latter course of action would, of course, destroy the unitarity of the formalism; and, even though unitarity would (presumably) be retrieved in the continuum limit, the high importance afforded to this fundamental "symmetry" of quantum physics generally overrides any thought of doing so.

Thus, one generally chooses the former alternative, namely, the rejection of the first-principles operator (1). However, before we flush it away, we should first be aware of what we are discarding. Consider the discrete Fourier transform (namely, the momentum-space representation) of $-i$ times the operation represented by (1):

$$p_{\text{first-principles}} = \frac{e^{ipa}-1}{ia} \equiv e^{ipa/2}\,\frac{e^{ipa/2}-e^{-ipa/2}}{ia} \equiv \frac{2}{a}\sin\!\left(\frac{pa}{2}\right)e^{ipa/2}. \tag{2}$$

The final factor of $e^{ipa/2}$ is simply a phase factor, which encapsulates the non-Hermiticity of the operator—namely, the reason we wish to reject it. Let us ignore it for the moment—say, by defining

$$\tilde{p}_{\text{first-principles}} \equiv \frac{2}{a}\sin\!\left(\frac{pa}{2}\right).$$

This latter operator is perfectly well-defined. For small p, we find that $\tilde{p}_{\text{first-principles}} \approx p$, as would be expected for any reasonable momentum operator. At the Brillouin zone boundary, namely, at $p = \pm\pi/a$, we find $\tilde{p}_{\text{first-principles}} = \pm 2/a$; and $2/a$ is a large, nonzero value. The shape of the function is not quite right: instead of being a linear (namely, just p), it "bends over" more and more as one approaches the Brillouin zone boundary, and is stationary at the boundary. But it is nevertheless monotonic in p, and has no zeroes except at $p = 0$.

How does the first-principles momentum operator manage to have a discontinuity across the Brillouin zone boundary? It doesn't, of course: we have ignored the extra phase factor $e^{ipa/2}$, which has the effect of "twisting" the function away from the real axis: at the Brillouin zone boundary, it has twisted $p_{\text{first-principles}}$ around to be purely imaginary, with the $\pm 90°$ "twist" for $p = \pm\pi/a$ bringing the two ends of the function together.

In any case, we are rejecting this first-principles operator, on the grounds that we wish to preserve unitarity. What next? Obviously, we want a definition of the derivative that is more symmetrical about the position in question than (1). The obvious thing to try is

$$\frac{df}{dx} \equiv \lim_{a\to 0} \frac{f(x+a) - f(x-a)}{2a}, \tag{3}$$

again with a being taken to be the lattice spacing. The matrix corresponding to this operator now has the desired antisymmetry:

$$\frac{1}{2a}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots \\ -1 & 0 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}.$$

(There is a -1 element in the top-right corner, and $+1$ in the bottom-left, if we want to preserve translational invariance by imposing periodic boundary conditions.) It is remarkable—even if usually taken for granted—that the changes in "epsilontics" represented by the change from (1) to (3) can make the difference between violating and observing the unitarity of the underlying formalism.

It is the definition (3) that formed the basis of most early lattice calculations. However, this operator is still very naïve in its structure. (It is, for example, also taught to school students.) Moreover, by trying to preserve unitarity, in a naïve way, we have immediately introduced spurious properties into the derivative operator that are completely responsible for the fermion doubling problem. It is straightforward to see why this is the case. In the definition (3), we are no longer taking a finite difference between adjacent lattice sites, but we are rather *skipping* a site, and comparing the function at one site to the function two sites away from it. Now, recall that the highest-frequency normalised wave that can be represented on the lattice (at the boundary of the first Brillouin zone) simply oscillates $+1, -1, +1, -1, \ldots$ as we move along the lattice sites. The first-principles operator (1) compares adjacent sites, and finds a large (indeed, maximal) change of ± 2. The naïve operator (3), on the other hand, skips every other site, and compares $+1$ with $+1$, and -1 with -1, and concludes that the derivative of the wave is zero everywhere!

This vanishing of the naïve derivative operator at the Brillouin zone boundary would not, in itself, be problematical, if it were contained to only this one frequency, because the anomalous properties of one single mode out of the total number of momentum modes (equal to the number of lattice sites) would lead to negligible effects as the number of lattice sites is increased. Rather, the real problem arises in the region *near* the Brillouin zone boundary. If we take the discrete Fourier transform of the operator (3), we find

$$p_{\text{naïve}} = \frac{e^{ipa} - e^{-ipa}}{2ia} \equiv \frac{1}{a}\sin pa. \tag{4}$$

As promised, the unitarity-breaking phase factor of (2) is absent; and for small p, we find $p_{\text{naïve}} \approx p$, as desired. However, by skipping a site in position space, we have halved the

period of the function in momentum space, so that instead of approaching a large value at the Brillouin zone boundary, $p_{\text{naïve}}$ smoothly approaches zero! Indeed, the positive-p part of the first Brillouin zone has effectively been divided into two mirror halves: from $p = 0$ to $p = \pi/2a$, in which $p_{\text{naïve}}$ increases monotonically, and represents a reasonably faithful representation of the momentum operator; and then from $p = \pi/2a$ to $p = \pi/a$, in which $p_{\text{naïve}}$ actually *decreases* for increasing p. (The same description can be made for the negative-p part of the first Brillouin zone.)

It is these "mirror states" that are fundamentally responsible for the fermion doubling problem using the naïve derivative operator (3). The problem only arises for fermions, because the Dirac operator contains the first-derivative, whereas for bosons we only require the *second*-derivative, which can be reasonably approximated by Hermitian operator

$$\frac{d^2 f}{dx^2} \equiv \lim_{a \to 0} \frac{f(x+a) - 2f(x) + f(x-a)}{a^2}, \tag{5}$$

which does not skip any lattice sites, and hence exhibits neither a zero at the Brillouin zone boundary nor the "mirror states" phenomenon.

Thus, by saving unitarity, we have introduced spurious properties into the momentum operator, effectively leading to a doubling of the fermion species. It used to be widely believed that, by the Nielson–Ninomiya "no-go" theorem, such fermion doubling was essentially unavoidable in any lattice formalism of interest to high-energy physics. This belief is, however, founded on a misconception, as will be further discussed in Sec. 3.

2. The ideal (SLAC) derivative operator

If the naïve derivative operator, (3), suffers pathologically from the doubling problem, then how are we to define a derivative operator that still maintains unitarity?

Clearly, the ideal situation would be if the discrete Fourier transform of the derivative operator were to be simply

$$p_{\text{ideal}} = p \tag{6}$$

(the "SLAC" prescription of Drell, Weinstein and Yankielowicz.) Being real, such an operator is clearly Hermitian, and so unitarity would be preserved. Without spurious zeroes and mirror states in the first Brillouin zone, there is no doubling problem. Moreover, the shape of the Fourier transform, namely, perfectly linear, means that the errors involved in performing the derivative on a discrete lattice are minimised; such errors are manifested in the "bending" of the Fourier transform of the derivative error away from linearity.

There are several immediate objections to a derivative operator satisfying (6). The first is that, being a discontinuous function in momentum space at the Brillouin zone boundary, such an operator much necessarily be "nonlocal" in position space. This not only causes us concern from a conceptual point of view—we generally wish to study manifestly local gauge theories—but also implies a huge explosion in computational cost, because instead of using just two lattice sites to perform a derivative operation, we would need to make use of every single one of them.

The conceptual objections to "nonlocality" will be addressed in Sec. 3, and the avoidance of the explosion of computational cost will be addressed in Sec. 4. Let us, therefore, put these concerns to one side, for the moment.

A more subtle objection is that it is not immediately obvious what the discrete Fourier transform of (6) actually is. This does not present any *practical* barrier, of course, because for any particular lattice size, it is a simple enough computational task to perform the discrete Fourier transform of (6), numerically. However, such a way of proceeding would be conceptually unsatisfactory—to me, at any rate—because it would leave quite mysterious the question of what the operator corresponding to (6) is actually doing in position space.

Fortunately, there is a relatively simple trick that allows us to obtain the discrete Fourier transform of (6) analytically. To explain it in a way that is easy to understand, I must first digress into an analogous physical problem: the case of an engineer wanting to sample a signal—for example, in order to digitise an audio track and record it onto a CD. The engineer's "position space" is actually time, whereas for the lattice we are thinking of spatial dimensions, but the mathematics and the concepts are the same.

Now, the engineer does *not* simply pass his audio signal into an analogue-to-digital converter, sampling at a suitably high rate. If he did, he would, in general, find that there are frequency components in the audio signal that are outside the first Brillouin zone. (He calls it by a different name, but we won't worry about that.) When sampled, these higher-frequency components would be "folded back" into the first Brillouin zone (he calls it "aliasing"); and when the signal is reconstituted in the CD player, they would cause audible interference or degradation (depending on whether they are coherent sounds or just simply noise).

To avoid this phenomenon, the engineer first passes the audio signal through a "low-pass filter". Ideally, such a filter would allow through all frequencies in the first Brillouin zone, and block all frequencies outside it. In practice, the ideal filter can only be approximated; but we do not care about the building of circuits, so we can assume that we have in our possession the mathematically ideal low-pass filter.

The engineer's analysis of this filtering process is as follows. Allowing through the frequencies in the first Brillouin zone, and blocking those outside it, is equivalent to multiplying the momentum spectrum by a filter function that is equal to 1 inside the Brillouin zone, and 0 outside it. Multiplying in momentum space is equivalent to convolving in position space, so it is of great interest to the engineer to determine the Fourier transform of this low-pass filter. This is easy to do:

$$\frac{1}{2\pi}\int_{-\pi/a}^{\pi/a} dp\, e^{ipx} = \frac{e^{i\pi x/a} - e^{-i\pi x/a}}{2\pi i x} \equiv \frac{\sin(\pi x/a)}{\pi x}.$$

This function plays such a fundamental rôle in the engineer's work that it is given a special name:

$$\mathrm{sinc}(x) \equiv \frac{\sin \pi x}{\pi x}, \tag{7}$$

in terms of which the Fourier transform of the low-pass filter can be written

$$\frac{1}{a}\,\mathrm{sinc}\Big(\frac{x}{a}\Big). \tag{8}$$

The sinc function has a central peak at $x = 0$, and oscillates away on each side with an envelope that drops off like $1/x$. Its fundamental use to the engineer can be seen when we only consider values of x that are actually on the lattice sites that we wish to use, namely,

$x_n = na$ (n being an integer); in other words, we wish to take the sinc of integer values. For sinc(0) we note that $\sin \pi x \to \pi x$ for small x, and hence $\mathrm{sinc}(0) = 1$. On the other hand, for all nonzero integral values n, we find that $\mathrm{sinc}(n) = 0$, because $\sin nx = 0$ and the denominator is nonzero.

From (8) it can thus be seen that, in position space, the low-pass filter times the cell width (namely, a) is a representation of the Dirac delta function on the lattice, with the additional property that, when extended to all continuum values of x between the lattice sites, it possesses no frequency components outside the first Brillouin zone. Thus, when the engineer wishes to reconstitute the analogue sound track from the digitally sampled values recorded on the CD, he convolves the stream of sampled values with the sinc function—in other words, he places a copy of the sinc function, with a weight given by the amplitude of the sound signal at that sample, centred on each sample site in question; the reconstructed signal is the sum of all of these weighted sinc pulses. The resulting "smoothly" reconstructed signal is then guaranteed to have no frequency components outside the first Brillouin zone; and, indeed, if it were to be re-sampled at the same rate, the fact that the sinc function is zero for all lattice sites other than the central one means that there is no "crosstalk": the same sampled values would be obtained. For information contained within the first Brillouin zone, the entire process is completely lossless, without distortion, and reversible.

So what is the relevance of this interesting digression into the world of engineering? It is this: When putting a physical formalism onto a lattice, it is much better (conceptually, at least) to break the process into three parts. Firstly, constrain the formalism to the first Brillouin zone, using the equivalent of a low-pass filter. Secondly, consider what operators will be required, and *construct them explicitly*, from this low-pass (but still continuum) formalism. Only after this is completed should the third and final step be taken: the "sampling" of the resulting formalism onto the lattice.

Clearly, this is a general prescription, for the placing of any physical formalism whatsoever onto a lattice. The particular case of interest to us here is the construction of the spatial derivative operator. So how should we construct it, according to this advice?

First, we need to understand some things about the derivative operator in the original continuum, unfiltered formalism. We may consider the process of differentiation to be equivalent to be the effect of convolving the negative of the derivative of the Dirac delta function, $-\delta'(x)$, with the function we wish to differentiate, because

$$-\delta' \star f(x) \equiv -\int_{-\infty}^{\infty} dw\, \delta'(x-w) f(w) = +\int_{-\infty}^{\infty} dw\, \delta(x-w) f'(w) \equiv f'(x),$$

where in the middle step we have integrated by parts and noted that the surface term vanishes on account of the delta function. (We are assuming, as usual, that the functions $f(x)$ that we wish to act on are sufficiently analytical for their product with the Dirac delta function or its derivative to be unambiguous and well-defined.) Thus, the derivative operator is, effectively, the negative of the derivative of the Dirac delta function. Now, the Fourier transform of the Dirac delta function is just unity, and it is an elementary property of the Fourier transform that taking the derivative in position space is simply equivalent to multiplying by $-ip$ in momentum space; thus, the Fourier transform of the derivative operator is simply ip, or in other words the Fourier transform of $-i\,d/dx$ is just p, which is exactly what we know from nonrelativistic quantum mechanics.

Let us now apply the low-pass filter, to allow through only that part of the formalism contained within the first Brillouin zone. As noted above, the Fourier transform of the Dirac delta function is unity, but we now filter this so that it is only unity within the first Brillouin zone, and zero outside; the Fourier transform of this function is, of course, the sinc function described by (8).

Our above analysis then tells us that the derivative operator in the low-pass formalism should simply be taken to be the negative of the derivative of the delta function in the low-pass formalism, namely, the negative of the derivative of (8). By the above, the Fourier transform of this function will simply be p inside the first Brillouin zone, and zero outside it, which is exactly what we want. Now, it is an elementary task to compute the derivative of (8): we find

$$\begin{aligned} d_{\text{ideal}}(x) &\equiv -\frac{d}{dx}\left\{\frac{1}{a}\,\text{sinc}\left(\frac{x}{a}\right)\right\} = -\frac{1}{\pi}\frac{d}{dx}\left\{\frac{\sin(\pi x/a)}{x}\right\} \\ &= -\frac{\cos(\pi x/a)}{ax} + \frac{\sin(\pi x/a)}{\pi x^2}. \end{aligned} \tag{9}$$

This function looks a little unfamiliar, but it will be of crucial importance for us, so it is worth spending a little time understanding it. As with the sinc function, its behaviour around $x = 0$ takes a little work. We first combine the two terms over a common denominator:

$$d_{\text{ideal}}(x) = \frac{-\pi x\cos(\pi x/a) + a\sin(\pi x/a)}{a\pi x^2}.$$

We now expand the trigonometric functions as Taylor series:

$$\begin{aligned} d_{\text{ideal}}(x) &= \frac{1}{a\pi x^2}\left\{-\pi x\left[1 - \frac{1}{2}\left(\frac{\pi x}{a}\right)^2 + \mathcal{O}(x^4)\right] + a\left[\frac{\pi x}{a} - \frac{1}{6}\left(\frac{\pi x}{a}\right)^3 + \mathcal{O}(x^5)\right]\right\} \\ &= \frac{1}{a\pi x^2}\left\{-\pi x + \frac{\pi^3 x^3}{2a^2} + \mathcal{O}(x^5) + \pi x - \frac{\pi^3 x^3}{6a^2} + \mathcal{O}(x^5)\right\} \\ &= \frac{\pi^2 x}{3a^3} + \mathcal{O}(x^3). \end{aligned} \tag{10}$$

Thus the function is perfectly well-behaved (and vanishes linearly) around $x = 0$, as would be expected from the properties of the sinc function itself.

According to the philosophy outlined above, it is the low-pass-filtered derivative function, $d_{\text{ideal}}(x)$, that we should seek to "sample" onto the lattice. So let us immediately proceed to do so, by considering x-values $x_n \equiv na$ corresponding to lattice sites labelled by the integer n. From (9), and multiplying by the cell width of a, we find

$$\Delta_{\text{ideal}}(x_n) \equiv a\,d_{\text{ideal}}(x_n) = \frac{-\pi n\cos n\pi + \sin n\pi}{\pi a n^2}.$$

From (10) we have already established that $\Delta_{\text{ideal}}(x_0) = 0$, so we need simply consider the remaining cases $n \neq 0$. We now note that $\sin n\pi = 0$ and $\cos n\pi = (-1)^n$, so that we obtain the remarkably simply result

$$\Delta_{\text{ideal}}(x_n) = \begin{cases} 0 & \text{if } n = 0, \\ -(-1)^n/an & \text{otherwise.} \end{cases} \tag{11}$$

What does this mean? It means that the ideal way to compute the first derivative on the lattice is not to use the naïve prescription (3), but rather

$$\frac{df}{dx} \equiv \lim_{a\to 0} \frac{1}{a}\Big\{ \ldots - \frac{1}{4}f(x+4a) + \frac{1}{3}f(x+3a) - \frac{1}{2}f(x+2a) + f(x+a) - f(x-a) + \frac{1}{2}f(x-2a) - \frac{1}{3}f(x-3a) + \frac{1}{4}f(x-4a) - \ldots \Big\}, \tag{12}$$

which is the SLAC derivative in position space in closed form. We can recognise the naïve derivative operator contained in the middle two terms of this expression, but with twice the usual coefficient. The analysis above tells us that it is the omission of all of the other terms that causes the Fourier transform of the naïve operator to pick up the pathologies of fermion doubling and mirror states.

As noted above, the structure of this operator will be of concern to some. The first question that might be asked, however, is simply this: Where on Earth did all of those other terms really come from? We started with a representation of the Dirac delta function that was nonzero for only one lattice site. Somehow, when we differentiated this function, we ended up with contributions on *all* lattice sites (except, perhaps ironically, the central site itself). How can differentiating "nothing" give us "something"?

The answer is, again, contained in the careful way that we first passed the formalism through the low-pass filter, to contain it within the first Brillouin zone. The resulting sinc function vanished at all of the lattice sites except the central one; but *its derivative is nonzero at all of these other sites*. It is almost like the sinc function was "hiding" the bulk of itself from our latticised view—but we "revealed" it by applying the derivative operator, which effectively "slides" the function right and left and reveals its variation to us. This physical picture hardly needs elaborating, since the variational formulation of mechanics (upon which essentially all lattice calculations in quantum physics are based) is rooted in precisely such a conceptual framework!

Of greater concern to some will be the fact that the operation (11) is very "nonlocal". Of course, different workers in the field have concocted conflicting definitions of "locality" in the context of the lattice—a topic I shall return to in greater detail in the next section—but I believe that by *any* of these prior definitions of "locality" the operation (11) would be considered "very nonlocal". In absolute value, the terms fall off extremely slowly—like $1/x$. The oscillatory nature of the signs of the terms means that they might, in a sense, be considered to fall off a little more quickly, for the same reason that the sum of $1/n$ is logarithmically divergent but the sum of $(-1)^n/n$ is convergent. However, to compute the derivative of some other function $f(x)$ at any particular point x, we still need to take into account the nature of the function $f(x)$ itself. Clearly, if it were to itself, say, diverge linearly for large x, the sum represented by (11) would not be convergent (the terms would oscillate between finite values). But such divergent functions will not (usually) be within the domain of functions that we wish to consider (and if they are, we will in any case find that the definition of the Fourier transform will probably break down, or at least require medical attention).

The more serious questions are the following: Would a "nonlocal" operator such as (11) destroy the structure of manifestly local field theories? And how could anyone possibly im-

plement the operator (11) without an explosion in computational cost? These two questions will be dealt with in the next two sections, respectively.

3. "Local" and "nonlocal" can be misnomers on the lattice

Given the fundamental importance of locality in (continuum) interacting field theory, it is understandable that lattice workers are concerned that this property is not violated on the lattice. However, this concern has led, in my opinion, to a number of misconceptions, which I believe need to be corrected.

The naïve derivative operator (3) is generally described as being "local". The main justification for this description is that it only involves neighbouring lattice sites to the site in question, and so as the lattice spacing is taken to zero in the continuum limit, the points essentially become coincident.

A common generalisation of this definition of "locality" is to deem an operator "local" if it involves only a *finite number* of neighbouring lattice sites. Again, the idea is that a finite number of lattice spacings, multiplied by an infinitesimally shrinking lattice spacing distance, equates to an infinitesimal distance, and hence a "local" operation.

There are several problems with such definitions.

Firstly, it is generally assumed that any operator *not* satisfying any particular definition in question is, in fact, "nonlocal" in the continuum limit. This conclusion is too harsh. What one must do is determine whether, as the lattice spacing is shrunk in real space, the definition in question "picks out" the derivative of any suitably differentiable test function at the point in question. Clearly, operators only employing a finite number of neighbours will satisfy this requirement. *But the converse is not true:* an infinite number of lattice sites may be involved in the operation, and the operation may still be local in the continuum limit, provided that, as the lattice spacing is reduced, the contribution from any *finite interval* at any *finite distance* from the point in question vanishes in the limit. It can be shown that the ideal SLAC operator described in Sec. 2 satisfies this requirement (for suitably non-divergent test functions, at any rate).

The second fundamental problem with conventional definitions of a "local" operator on the lattice is that the impression is sometimes conveyed that the operator in question *is* actually local *for a finite lattice spacing distance*. This cannot, of course, be true. For a finite lattice spacing, we are undeniably linking fields at one lattice point to fields at other lattice points, which are finite distances away. These are manifestly nonlocal interactions. The point, of course, is that we wish to retrieve the locality of interactions *in the continuum limit*. Locality is one property of the continuum formalism that we cannot, by any trick, observe for finite lattice spacing.

This is made clearer if we consider a Taylor series expansion of the test function $f(x)$ about any point x in the continuum. Clearly, to find $f(x \pm a)$ at some finite distance a away from x, we would need to know the values of *all* of the (infinite number of) derivatives of f at the point x. By a reversion of series, the converse is equally true: to find the value of $f'(x)$ using only the values of f on lattice sites x_n, it is necessary to use *all* of the (infinite number of) values of f (for an infinite lattice). In effect, we can use the naïve operator (3) to give us a first-approximation to the derivative, but then we would need to use the higher-order finite differences in order to correct the contribution of the higher derivatives to the Taylor

series expansion.

This is precisely what is being achieved by the ideal SLAC operator (12).

4. The proposed derivative operator

The final—and valid—objection to the SLAC lattice derivative operator (12) is a practical one. How can one possibly implement it without blowing out the computational cost enormously? The operator (12) links a given (one-dimensional) lattice site to every other site in the lattice. Compare this to the naïve operator (3), which only links the given site to its two nearest neighbours. The time required to perform such a computation will increase by a factor of half the number of lattice sites!

The solution I propose to this problem recognises the nature of the calculations that we wish to perform on the lattice. We are not interested in looking at the numerical result of each and every application of the derivative operator to a given field. Rather, our calculations are such that we need to perform the derivative operation many, many times. These results are thrown together with the dynamics of the system we are trying to model, and at the end of the day we extract a small set of numbers that provide a good estimate (we hope) of some physical property of the system in question.

In effect, each such extracted value represents the integrating up of a huge number of elementary operations. In this context, it is reasonable to consider allowing some statistical uncertainty in the definition of each elementary operation. Even if each individual application of the operation does not manifestly exhibit the properties of the ideal SLAC operator, we can be confident that, *on the average*, the ideal operator will be faithfully represented. To do this, we need simply ensure that the *expectation values* of the properties of the individual operators being applied are equal to the properties of the desired ideal SLAC operator.

That is the general philosophy. Let me now make it concrete. Look back at the SLAC derivative operator (12). Inside it is contained twice the naïve derivative operator (3) that uses lattice sites that are one lattice spacing away from the point in question. My first stipulation is that, *every* time the derivative operator is applied, we *at least* compute twice the contribution due to the naïve operator. This provides our starting point. We are thus doing no worse, in some sense, than what we would do with the naïve operator.

We next consider the terms that involve sites that are *two* lattice spacings away. Apart from an extra minus sign, these two terms come in with a factor of 1/2. Now, instead of calculating this difference, and multiplying it by 1/2, I make the following stipulation: by random selection, only calculate this difference *half of the time*. If you calculate it, add it in completely. If you don't calculate it, don't add anything in.

We next turn to the two terms involving sites that are *three* lattice spacings away. The sign has flipped back to positive, and the terms come in with a coefficient of 1/3. Again, by random selection (independent of the above selection!), calculate this difference only *one-third* of the time, and add it in (completely) only if you do calculate it.

Continue this process, until you have gone right up to the limit of sites that are half a lattice away (using periodic boundary conditions where necessary).

The sum of the terms that you *did* calculate is now the value that you use for the derivative.

How much of a factor increase in computational time does such a proposal entail? In

terms of the actual computation of differences—the costly part—the number of calculations you will need to perform, on average, will go like the *logarithm* of the number of lattice sites, because it is simply $1 + 1/2 + 1/3 + \ldots$ up to half the number of sites, which (like the integral of $1/x$) is logarithmic. This is a lot better than being proportional to the number of lattice sites, as a naïve application of the SLAC derivative would entail!

The factor is not convergent, unfortunately, as the number of lattice sites goes to infinity; to make a convergent derivative operator of this sort, one would have to reduce the probabilities of the higher-distance differences, in some functional way, but compensate by multiplying each such difference by an increasingly large coefficient. Such a scenario sounds inherently unstable; and it may be quite possibly to prove that the fluctuations inherent in such a process would not reduce in variance over a large ensemble of such operations. In any case, it sounds like a bad way to proceed to me; living with a factor that is only logarithmic in the number of lattice sites is quite reasonable in any practical context.

The only remaining concern is the task of randomising the decision about whether to compute each difference at a given distance. Clearly, this selection task is the only part of the calculation that is *not* logarithmically sped up, because for each distance, we need to decide whether to compute or not compute. It would be nice if there were some simple way to spit out random numbers that told us *which* distances to compute; but I have not been able to think up any method that is quicker than a brute force loop.

Fortunately, such an explicit loop is relatively cheap in computational terms. Imagine we have at our disposal a (pseudo-)random bitstream. To decide whether to perform the distance-2 difference, we simply roll one bit off this stream. If it is 0, we do it; if it is 1, we don't. For distance-3, we roll off two bits. If the binary number created by these two bits is greater than 2, we try again. Otherwise, we check if the number is 0; if so, we do it; if it is 1 or 2, we don't. Likewise, for distance-4; we never need to re-try. We continue on in this fashion. Now, generating a pseudorandom bitstream takes only a simple addition per machine integer; comparing to the loop variable is a single instruction in machine code; and testing for zero is another single instruction in machine code. Incrementing the loop counter takes one further instruction. Thus, even though, using this approach, we need to loop through every single lattice site, the number of machine cycles (if written in assembly language) needed to make the decisions is four times the number of lattice sites in one dimension, divided by some efficiency factor (better than one-half!) for the case of discarded values. In most practical cases this should be a negligibly small cost compared to the difference operations themselves. (In cases where even this cost is prohibitive, it may be possible to "precompute" a large set of yes–no flags in a lookup table that could be re-used.)

5. Application to the second-derivative

Although, as noted earlier, it does not suffer from the pathologies of the fermion doubling and "mirror states" problems, the *second*-derivative operator is nevertheless of fundamental interest to gauge theory calculations. If one is already applying the first-derivative operator of the previous section to one's lattice formalism, one might ask whether a similar improvement to the action may be possible by applying the same principles to the second-derivative.

I believe that it can. All that we need to do is take the negative of the derivative of the

ideal (continuum but low-pass-filtered) derivative operation (9):

$$\begin{aligned} d^{(2)}_{\rm ideal}(x) &\equiv -\frac{d}{dx}\, d_{\rm ideal}(x) \\ &= -\frac{2\cos(\pi x/a)}{ax^2} + \frac{2\sin(\pi x/a)}{\pi x^3} - \frac{\pi\sin(\pi x/a)}{a^2 x}. \end{aligned} \tag{13}$$

Again, when we "sample" onto the lattice, and multiply by the size a of the lattice cell, we obtain

$$\Delta^{(2)}_{\rm ideal}(x_n) \equiv a\, d^{(2)}_{\rm ideal}(x_n) = -\frac{2\cos n\pi}{a^2 n^2} + \frac{2\sin n\pi}{a^2 n^3 \pi} - \frac{\pi\sin n\pi}{a^2 n}.$$

In this case we need to be careful when extracting the coefficient at $n = 0$; in the limit $n \to 0$, one finds

$$\Delta^{(2)}_{\rm ideal}(0) = -\frac{\pi^2}{3a^2}.$$

For $n \neq 0$, one can again use the identities $\sin n\pi = 0$ and $\cos n\pi = (-1)^n$. The overall result for the closed form of the "second SLAC derivative" is then

$$\Delta^{(2)}_{\rm ideal}(x_n) = \begin{cases} -\pi^2/3a^2 & \text{if } n = 0, \\ -2(-1)^n/a^2 n^2 & \text{otherwise.} \end{cases} \tag{14}$$

In this case, the fall-off of the coefficients goes like $1/n^2$ rather than $1/n$. The operator (14) does not contain precisely the naïve operator (5), but the structure of the three middle terms is similar, even if the numerical coefficients differ.

To implement the operator (14) in the "stochastic" fashion proposed in Sec. 4, it would probably be sufficient to calculate the three middle terms for every application of the derivative, and then to apply the higher-distance contributions (sums, now, rather than differences, but with the correct sign) with a statistical weight equal to $2/n^2$.

6. Conclusions

In this paper I have tried to explain, in a pedagogical way, why the SLAC prescription for the definition of the derivative operator is the "cleanest" from a fundamental point of view. I have shown that by very simple arguments it is possible to obtain the position-space form of the SLAC derivative in closed form. I have proposed that, by implementing this operator in a stochastic fashion, it will be possible to gain the advantages of the SLAC derivative without significantly more computational cost than using either the naïve derivative or the Wilson prescription for removing the fermion doubling problem.

Acknowledgments

I gratefully thank Tien D. Kieu for introducing me to this problem, the participants at the 2002 Congress of the Australian Institute of Physics for rekindling my interest in it, and Philippe de Forcrand and Christian Hoelbling for providing vital comments on the first version of this paper.