Computer Physics Communications 54 (1989) 109–124
North-Holland, Amsterdam

A VECTORIZED CODE FOR THE COMPUTATION OF THE TOPOLOGICAL CHARGE IN SU(2) LATTICE GAUGE THEORY

A.S. KRONFELD [a,1], M.L. LAURSEN [b,2], G. SCHIERHOLZ [a,c], C. SCHLEIERMACHER [d,3] and U.-J. WIESE [e,4]

[a] *Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, D-2000 Hamburg 52, Fed. Rep. Germany*
[b] *Institut für Physik der Johannes-Gutenberg-Universität, D-6500 Mainz, Fed. Rep. Germany*
[c] *Institut für Theoretische Physik der Universität Kiel, D-2300 Kiel, Fed. Rep. Germany*
[d] *Institut für Theoretische Physik der Universität Hannover, Appelstraße 2, D-3000 Hannover, Fed. Rep. Germany*
[e] *II. Institut für Theoretische Physik der Universität Hamburg, Luruper Chaussee 149, D-2000 Hamburg 50, Fed. Rep. Germany*

Received 26 September 1988

A vectorized code for calculating the topological charge of an SU(2) lattice gauge field is presented. The program is based on the combinatoric algorithm of Phillips and Stone. The present version works for hypercubic lattices with the gauge field stored according to the three-dimensional checkerboard scheme. Other storage schemes and simplicial lattices can be accommodated with minor modifications.

PROGRAM SUMMARY

Title of program: QUBIC

Catalogue number: ABHQ

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: Cray X-MP/48; *Installation*: Höchstleistungsrechenzentrum HLRZ, c/o KFA, Postfach 1913, D-5170 Jülich, Fed. Rep. Germany

Operating system: COS 1.16

Programming language used: FORTRAN 77

High speed storage required: size dependent; for a NS^3NT lattice the largest COMMON block occupies $16 * NS^3 * 4 * 27$ words

Number of bits in a word: 64

Number of lines in combined program and test deck: 1232

Keywords: lattice gauge theory, topology, instantons; fiber bundles, characteristic classes, second Chern number; vector processor

Nature of the physical problem
Four-dimensional SU(N) gauge fields are characterized by a topological charge [1], known as the second Chern number to mathematicians. This feature, not shared by Abelian gauge fields, is conjectured to be significant for the peculiar properties of quantized non-Abelian gauge theories. For example, the topology of the gauge field is known to be relevant to the resolution of the "$U_A(1)$ problem" [2], and the role of topology in the confinement mechanism needs clarification.

Method of solution
The problem of non-Abelian gauge fields is nonperturbative, and the most successful approach has been numerical simulations of the corresponding lattice gauge theory. For lattice gauge fields the topological charge can be obtained by reconstructing the underlying topological object, the coordinate bundle, from the lattice gauge field [3]. In the case of SU(2), the algorithm of Phillips and Stone reduces the computation of the topological charge to combinatorics [4].

[1] Present address: Theoretical Physics Group, Fermilab, P.O. Box 500, Batavia, IL 60510, USA.
[2] Present address: Instituut voor Theoretische Fysica, Valckenierstraat 65, NL-1018 XE Amsterdam, The Netherlands.
[3] Present address: Deutsche Bundeswehr, Flensburg, Fed. Rep. Germany.
[4] Present address: Gruppe Theorie der Elementarteilchen, Höchstleistungsrechenzentrum HLRZ, c/o KFA, Postfach 1913, D-5170 Jülich, Fed. Rep. Germany.

The present program uses this algorithm to compute the topological charge of a $NS^3 \times NT$ hypercubic lattice [5]; it is designed to be appended to the potential user's existing simulation programs. The test code sets up a configuration with $Q = 1$, to assist the user in installation.

Restrictions on the program

(a) The lattice size parameters NS and NT must be even.
(b) The lattice must have periodic boundary conditions.
(c) The user must provide program(s) to generate a sequence of SU(2) lattice gauge fields; QUBIC then determines the topological charge.

Typical running time

Compiled with the CFT compiler QUBIC needs 9.14 CPU-s on the Cray X-MP/48 to determine the topological charge of a 10^4 lattice. Despite considerable INTEGER arithmetic and several BLOCK-IF's, QUBIC attains a performance of 47 Mflops.

References

[1] A.A. Belavin, A.M. Polyakov, A.S. Schwartz and Y.S. Tyupkin, Phys. Lett. B 59 (1975) 85.
[2] G. 't Hooft, Phys. Rev. Lett. 37 (1976) 8; Phys. Rev. D 14 (1976) 3432.
[3] M. Lüscher, Commun. Math. Phys. 85 (1982) 29.
[4] A. Phillips and D. Stone, Commun. Math. Phys. 103 (1986) 599.
[5] A.S. Kronfeld, M.L. Laursen, G. Schierholz and U.-J. Wiese, Nucl. Phys. B292 (1987) 330.

LONG WRITE-UP

1. Introduction

Most modern theories of particle physics incorporate non-Abelian gauge fields in one way or another. Several models are patterned in some way after quantum chromodynamics (QCD), the leading candidate theory of the strong interactions. In these models nonperturbative aspects play a dominant role: for example, the quanta of the fundamental fields ought to be *confined*, and the physical particles are analogous to mesons and baryons. Confinement is inaccessible to perturbation theory, which is more or less similar to Abelian theories (like QED).

The most successful and systematic nonperturbative approach to field theory is the formulation on a spacetime lattice [1]. A special advantage of this framework is that it is amenable to numerical simulations [2]. However, the complexity of the models makes it imperative that efficient codes for supercomputers be developed. This paper describes a program – QUBIC – that measures the topological charge of lattice gauge fields with gauge group SU(2). The program is designed to be attached to users' other simulation programs of SU(2) lattice gauge theories; in conjunction with nontrivial methods to be described elsewhere, the program can be used for SU(3), the gauge group of QCD, and other SU(N). Although we have run primarily on a Cray X-MP, the code is standard FORTRAN 77, and hence portable. We have successfully run the test package on a Fujitsu VP100 and an IBM 3084. The present version and its predecessors have been used successfully in numerical simulations [3,4].

The structure of the program description is as follows. Sections 2 and 3 summarize some of the physical and mathematical background needed to appreciate the program. The specific algorithm employed is due to Phillips and Stone, and section 4 summarizes their ref. [5]. As it stands this algorithm applies to simplicial lattices; section 5 shows how to obtain a simplicial lattice from a hypercubic one. QUBIC is explained thoroughly in section 6; in particular, many technical details that were left out in section 4 appear here, so that the user can compare the theory to the code. Section 7 contains the results of test runs on two vector processors. The test package comes with subroutines to create a $Q = 1$ lattice gauge field; the construction is described in the appendix. Reasonable questions from users may be addressed to one of us (A.S.K.) by computer mail: ASK@FNAL.BITNET.

2. Topology of non-Abelian gauge fields

This and the following section are very brief. For a more thorough discussion, see ref. [6].

2.1. Instantons, tunnelling and the θ-vacuum

Non-Abelian gauge fields are described by a gauge potential A_μ taking values in a Lie algebra. The associated field strength $F_{\mu\nu}$ is given by

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu + [A_\mu, A_\nu]. \tag{2.1}$$

The dynamics are dictated by the Yang–Mills action

$$S = \frac{1}{2g^2}\int \mathrm{d}^4x \ \mathrm{Tr}\{F_{\mu\nu}F_{\mu\nu}\}, \tag{2.2}$$

with the trace taken in the fundamental representation. Here we restrict the discussion to $\mathfrak{su}(N)$ and the corresponding real, compact Lie group SU(N).

One feature that distinguishes SU(N) gauge theory from QED is the existence of the topological charge

$$Q = -\frac{1}{16\pi^2}\int \mathrm{d}^4x \ \mathrm{Tr}\{F_{\mu\nu}{}^*F_{\mu\nu}\}, \qquad {}^*F_{\mu\nu} = \tfrac{1}{2}\epsilon_{\mu\nu\rho\sigma}F_{\rho\sigma}. \tag{2.3}$$

Within each charge section the inequality,

$$\int \mathrm{d}^4x \ \mathrm{Tr}\left\{\left(F_{\mu\nu} \pm {}^*F_{\mu\nu}\right)^2\right\} \geq 0, \tag{2.4}$$

implies that the action is bounded below by

$$S \geq |Q|8\pi^2/g^2. \tag{2.5}$$

The inequality is saturated by solutions of the (Euclidean) equations of motion, called instantons. These are self-dual or anti-self-dual [7], $F_{\mu\nu} = \pm {}^*F_{\mu\nu}$. For example, the one instanton solution has a gauge potential

$$A_\mu = \frac{r^2}{r^2+\rho^2} g_1^{-1}\partial_\mu g_1, \tag{2.6}$$

where the size ρ of the instanton is an arbitrary scale, and

$$g_1 = (x_4 + \mathrm{i}\boldsymbol{\sigma}\cdot\boldsymbol{x})/r, \quad r^2 = x_4^2 + \boldsymbol{x}^2. \tag{2.7}$$

Note that the map g_1 cannot be reached by continuous deformations from the trivial map $g_0 = 1$.

Now consider the classical vacuum state $|0\rangle$ characterized by $A_\mu = 0$ (modulo gauge transformations). Gauss' law requires that $|0\rangle$ be invariant under infinitesimal gauge transformations, but it says nothing about homotopically nontrivial transformations such as g_1. Hence, there is an infinite sequence of classical vacua

$$|n\rangle = T_1^n|0\rangle, \tag{2.8}$$

where T_1 is the unitary operator implementing the gauge transformation g_1, and n is any integer. Due to tunnelling caused by the instantons, the quantum vacuum is described by a superposition of the $|n\rangle$, and since T_1 commutes with the Hamiltonian, one finds [8]

$$|\theta\rangle = \sum_n \mathrm{e}^{-\mathrm{i}\theta n}|n\rangle. \tag{2.9}$$

The vacuum parameter θ is a new and unexpected feature of QCD. If it is nonzero, the path integral is modified by the substitution $S \to S + \mathrm{i}\theta Q$. To understand the θ-vacuum one needs a nonperturbative method to compute Q; for recent progress in this area, see ref. [9].

2.2. The $U_A(1)$ problem and the Witten–Veneziano formula

Topologically nontrivial gauge fields become relevant to physics through the resolution of the $U_A(1)$ problem [10], which revolves around the flavor-singlet axial, i.e. $U_A(1)$, current:

$$J_\mu^5 = \sum_{f=1}^{N_f} \bar{\psi}_f \gamma_\mu\gamma_5\psi_f. \tag{2.10}$$

On the one hand, the symmetry generated by J_μ^5 ought to be spontaneously broken, like the flavor-nonsinglet symmetry SU(N_f) *. On the other hand, there is no meson in nature which can be interpreted as the associated Goldstone boson – the candidate η' (for $N_f = 3$) is much too massive. The first crucial step towards the resolution of the $U_A(1)$ problem is the chiral anomaly [11]. In the

* N_f is the number of "light" flavors.

quantum theory, J_μ^5 is no longer conserved, but rather

$$\partial_\mu J_\mu^5 = -2N_f q, \quad q = -\frac{1}{16\pi^2}\mathrm{Tr}\{F_{\mu\nu}{}^*F_{\mu\nu}\}. \tag{2.11}$$

The topological charge density appears on the right-hand-side of the (non)conservation law! Consequently, the anomaly *explicitly* breaks the $U_A(1)$ symmetry [12], Goldstone's theorem does not apply to J_μ^5, and the η' can assume the role of a run-of-the-mill meson.

The importance of topologically nontrivial fields is made more quantitative by the Witten–Veneziano formula [13,14]. In a certain large N limit, they find

$$\chi_t^{\text{quenched}} = \frac{f_\pi^2}{6}\left(m_{\eta'}^2 + m_\eta^2 - 2m_K^2\right), \tag{2.12}$$

where the *topological susceptibility* is given by

$$\chi_t \equiv \int d^4x \langle q(x)q(0)\rangle = \frac{\langle Q^2\rangle}{V}, \tag{2.13}$$

$$V = \int d^4x = \text{volume of spacetime},$$

and the superscript "quenched" indicates that the susceptibility to be inserted into eq. (2.12) is that of the pure gauge theory. Using the real world ($N = 3$) masses and decay constant f_π in eq. (2.12) yields $\chi_t^{\text{quenched}} = (180\ \text{MeV})^4$. This number should not be taken too seriously, because it is not clear how large various corrections might be.

3. Fiber bundles

3.1. Other expressions for the topological charge

For computations of the topological charge the expression in eq. (2.3) is not especially useful. In particular, the implication that Q depends on local properties of the field strength $F_{\mu\nu}$ is illusory. In fact, the integrand of eq. (2.3) is apparently a total divergence: $q = -(1/16\pi^2)\,\mathrm{Tr}\{F_{\mu\nu}{}^*F_{\mu\nu}\} = \partial_\mu \Omega_\mu$, where

$$\Omega_\mu = -\frac{1}{8\pi^2}\epsilon_{\mu\nu\rho\sigma}\,\mathrm{Tr}\left\{A_\nu\left(\partial_\rho A_\sigma + \tfrac{2}{3}A_\rho A_\sigma\right)\right\}, \tag{3.1}$$

and the gauge potential is assumed to be in some smooth gauge. However, the Chern–Simons form Ω_μ can develop singularities, leading to a nonzero topological charge. The situation is similar to a closed surface with nonzero extrinsic curvature, for which there is no smooth choice of coordinates without singularities. Similarly, there is no smooth choice of gauge without singularities for a non-Abelian gauge field with nonzero topological charge.

These problems of curvature are best handled by the formalism of fiber bundles. See ref. [16] for a review oriented towards physicists. In this formalism one considers the spacetime volume to be a union of cells

$$M = \mathbb{T}^4 = \bigcup_\alpha c_\alpha; \tag{3.2}$$

the 4-torus is convenient because of the eventual lattice simulations, but in the following all we require is $\partial M = 0$. In each cell c_α one can pick a smooth gauge potential $A_\mu^{(\alpha)}$. On the overlaps $c_{\alpha\beta} \equiv c_\alpha \cap c_\beta$ the gauge potentials are related by

$$A_\mu^{(\alpha)} = v_{\alpha\beta}\left(\partial_\mu + A_\mu^{(\beta)}\right)v_{\beta\alpha}, \quad v_{\alpha\beta} = v_{\beta\alpha}^{-1}. \tag{3.3}$$

The group-valued transition functions $v_{\alpha\beta}$ constitute the *coordinate bundle*. On double overlaps $c_{\alpha\beta\gamma} = c_\alpha \cap c_\beta \cap c_\gamma$, they must obey the cocycle condition:

$$v_{\alpha\gamma} = v_{\alpha\beta}v_{\beta\gamma}. \tag{3.4}$$

All of the information needed to determine the topological charge is encoded in the transition functions. This is shown by breaking the integral over M into a sum of integrals over the cells. Then eq. (3.1) can be applied and integration by parts is possible, because the Chern–Simons term $\Omega_\mu^{(\alpha)}$ is nonsingular in c_α, by hypothesis. On $c_{\alpha\beta}$ one then forms the gauge variation $\Omega_\mu^{(\alpha)} - \Omega_\mu^{(\beta)}$, and after considerable rearrangement one finds [15,17]

$$\begin{aligned} Q = &+\frac{1}{8\pi^2}\sum_{\alpha<\beta<\gamma}\int_{c_{\alpha\beta\gamma}} d^2x_{\mu\nu}\,\epsilon_{\mu\nu\rho\sigma} \\ &\times \mathrm{Tr}\left\{a_\rho^{(\gamma,\beta)}a_\sigma^{(\alpha,\beta)}\right\} \\ &-\frac{1}{24\pi^2}\sum_{\alpha<\beta}\int_{c_{\alpha\beta}} d^3x_\mu\,\epsilon_{\mu\nu\rho\sigma} \\ &\times \mathrm{Tr}\left\{a_\nu^{(\alpha,\beta)}a_\rho^{(\alpha,\beta)}a_\sigma^{(\alpha,\beta)}\right\}, \end{aligned} \tag{3.5}$$

where $a_\mu^{(\alpha,\beta)} = v_{\beta\alpha}\partial_\mu v_{\alpha\beta}$.

The expression for Q can be simplified even further in terms of the local section, which is a map g_α: $\partial c_\alpha \to \mathrm{SU}(N)$, obeying

$$g_\alpha g_\beta^{-1} = v_{\alpha\beta} \tag{3.6}$$

on $c_{\alpha\beta}$. In terms of the g_α one finds

$$Q = \sum_\alpha Q_\alpha,$$

where

$$Q_\alpha = -\frac{1}{24\pi^2}\int_{\partial c_\alpha} \mathrm{d}^3x_\mu\, \epsilon_{\mu\nu\rho\sigma}\, \mathrm{Tr}\left\{ b_\nu^{(\alpha)} b_\rho^{(\alpha)} b_\sigma^{(\alpha)} \right\},$$

$$b_\mu^{(\alpha)} = g_\alpha^{-1}\partial_\mu g_\alpha. \tag{3.7}$$

In eq. (3.7) the right-hand-side is an element of the homotopy group $\Pi_3(\mathrm{SU}(N)) = \mathbb{Z}$.

3.2. Efficient computation of the homotopy classes

Of the various formulae for the topological charge, eq. (3.7) is the most efficient to implement numerically, because it reduces the problem to the computation of homotopy classes of maps defined locally. This provides hope that the homotopy classes can be determined combinatorically, although a practical algorithm exists only for SU(2). In contrast, eq. (3.5) yields an integer after the sum over all cells, and the fractional contributions must be determined by sufficiently precise numerical integrations. Alternatively, the expression in eq. (3.5) can be reduced to local integers [18], but this involves explicit searching for singularities, which is also slower than a combinatoric method.

For the group SU(2) the homotopy group of interest is $\Pi_3(S^3)$, because of the isomorphism $\mathrm{SU}(2) = S^3$. In this case, as for all $\Pi_n(S^n)$, the homotopy class is given by the *winding number*, the number of times the map wraps around the sphere S^n. The winding number can be computed as follows. Pick an arbitrary "probe" $y \in \mathrm{SU}(2)$, and let $\underline{x}$ denote points on ∂c_α such that $g_\alpha(\underline{x}) = y$. Then the winding number is given by

$$Q_\alpha = \sum_{\underline{x}} \mathrm{sign} \left\| \frac{\partial g_\alpha}{\partial x}\bigg|_{\underline{x}} \right\|; \tag{3.8}$$

the summand is the orientation of the map $g_\alpha(\underline{x})$ relative to ∂c_α. The result is independent of the probe y, because each point of the target space is covered equally often (counting algebraically).

The meaning of eq. (3.8) is best explained in an R^4 notation. An element of SU(2) can be represented by a unit 4-vector, $u = u_4 + \mathrm{i}\boldsymbol{\sigma}\cdot\boldsymbol{u}$, $\boldsymbol{u}^2 + u_4^2 = 1$, and, since $x \in \partial c_\alpha \subset c_\alpha \subset R^4$, x and $\underline{x}$ are automatically four-vectors. Then

$$\left\| \frac{\partial g_\alpha}{\partial x} \right\| = \begin{array}{l}\text{Jacobian determinant}\\ \text{for the variable}\\ \text{transformation } x \mapsto g_\alpha.\end{array} \tag{3.9}$$

In the description of the program, we will generally speak of SU(2)-elements in the R^4 and S^3 language, because the geometry is simplest in this language *. For example, $\| \partial g_\alpha/\partial x \|$ is positive if the four-vectors $g_\alpha(x)$ and x have the same handedness, and negative otherwise.

3.3. Topology of lattice gauge fields

At first sight, the title of this subsection seems to be a contradiction in terms. The configuration space of lattice gauge theory is the product space

$$\underbrace{G \times G \times \ldots \times G}_{\text{one factor for each link}} \quad . \tag{3.10}$$

Since this is a discrete space, the natural topology is trivial. However, the natural topology is not the correct topology, if one wants to construct the continuum quantum field theory from the lattice theory. A correct topology for lattice gauge theory is one which reproduces the continuum topology for sufficiently smooth lattice gauge fields. Lüscher realized [15] that this is best achieved by reconstructing the coordinate bundle from the data of the lattice gauge field. In other words, one prescribes a set of transition functions, defined in terms of the link matrices. Equivalently, one can prescribe a set of local sections.

* The major exception will be when we refer to matrix multiplication, which is clearest in the matrix language.

4. Phillips and Stone algorithm

The algorithm that most elegantly embodies the above concepts is due to Phillips and Stone [5]. For SU(2) it is very fast. The essential feature is that the section maps certain polyhedra $D \subset \partial c_\alpha$ into SU(N) polyhedra. Let D have vertices x_i and its image $g_\alpha(D) \subset S^3$ have vertices $y_i = g_\alpha(x_i)$. For SU(2) the whole procedure can be embedded into R^4, as discussed in section 3.2. Then the question "$\exists \underline{x}$: $g_\alpha(\underline{x}) = y$?" can be answered by relatively straightforward geometry. $\underline{x}$ need not be determined explicitly, because the relative orientation of $g_\alpha(\underline{x})$ to ∂c_α is dictated by the relative orientation of the y_i to the x_i, and by some macroscopic features of the y_i.

This remainder of this section leans heavily on ref. [5], although most aspects of mathematical rigor (existence, continuity, etc.) are left out. Many details that directly affect the computation are deferred to section 6, so that the algorithm and code can be compared.

After introducing some notation, this section explains how Phillips and Stone construct transition functions. Their algorithm is based on two principles: it is inductive in the spacetime dimension, and the transition functions are chosen to be "as close as possible" to the lattice gauge field. The last subsection explains how to obtain the local sections from the transition functions.

4.1. Notation

The lattice Λ is composed of sites α, β, γ,.... The links of Λ form a complex of simplices $\sigma = \langle \alpha\beta \cdots \rangle$. The dimension of σ is d if the number of sites between the angle brackets is $d+1$. We will write $\tau \preccurlyeq \sigma$ if all of the vertices of τ are also in σ; clearly $d_\tau \le d_\sigma$. If $\tau = \sigma$ is to be precluded, then we write $\tau \prec \sigma$. For example, a four-dimensional lattice consists of 0-simplices $\langle \alpha \rangle$ (the sites), 1-simplices $\langle \alpha\beta \rangle$ (the links), 2-simplices $\langle \alpha\beta\gamma \rangle$ (triangles), 3-simplices $\langle \alpha\beta\gamma\delta \rangle$ (tetrahedra) and 4-simplices $\langle \alpha\beta\gamma\delta\epsilon \rangle$.

The algorithm requires a local ordering **o** of the vertices in each simplex. Thus, when we write $\sigma = \langle 012 \cdots d \rangle$, we imply that the vertices are **o**-ordered from left to right. For $\tau \prec \sigma$ the relative ordering in τ must be the same as in σ. Moreover, if $\tau \prec \sigma$ and $\tau \prec \rho$, then the ordering that τ inherits from σ must be the same as that inherited from ρ.

The cells of the discussion of section 3 will be the dual cells c_α. The aim is to construct transition functions on the overlaps $c_{\alpha\beta} = c_\alpha \cap c_\beta$. For further intersections of simplices we write $c_{\alpha\beta\cdots\gamma} = c_\alpha \cap c_\beta \cap \cdots \cap c_\gamma$. The algorithm works independently within each simplex σ, so it is convenient to write $c_\alpha^\sigma = c_\alpha \cap \sigma$, etc. If $\sigma = \langle \alpha\beta \rangle$ then $c_{\alpha\beta}^\sigma$ is the midpoint of $\langle \alpha\beta \rangle$, denoted by $p_{\alpha\beta}$.

Convenient coordinates for simplices are "barycentric coordinates", $t_0, t_1, \ldots, t_d$ for the d-dimensional simplex $\sigma = \langle 012 \cdots d \rangle$. An arbitrary point x is given by

$$x = \sum_{i=0}^{d} x_i t_i, \tag{4.1}$$

where x_α denotes the position of site α, and the t_i obey the constraint $\sum_i t_i = 1$. In terms of these coordinates, the simplices and dual cells are given by

$$\begin{aligned} \sigma &= \langle 012 \cdots d \rangle = \{ x \,|\, 0 \le t_i \le 1, \forall i \}, \\ c_\alpha^\sigma &= \{ x \,|\, 0 \le t_\beta, t_\gamma, \ldots \le t_\alpha; \langle \beta \rangle, \langle \gamma \rangle, \ldots \prec \sigma; \\ &\qquad \beta, \gamma, \ldots \ne \alpha \}, \\ c_\alpha &= \bigcup_{\sigma \succ \langle \alpha \rangle} c_\alpha^\sigma. \end{aligned} \tag{4.2}$$

Within c_α^σ the algorithm uses *modified barycentric coordinates* $s_\lambda = t_\lambda / t_\alpha \in [0, 1]$ for $\lambda \ne \alpha$ and $\langle \lambda \rangle \prec \sigma$. In these coordinates c_α^σ is a (hyper)cube. The face $c_{\alpha\beta}^\alpha$ has $s_\beta = 1$, in the modified barycentric coordinates of c_α^σ *, and hence is a cube of one dimension lower. It turns out to be useful to divide ∂c_{0d}^σ into two pieces:

$$\partial^1 c_{0d}^\sigma \equiv \bigcup_{\alpha=1}^{d-1} c_{0\alpha d}^\sigma = \bigcup_\alpha \{ \text{face with } s_\alpha = 1 \} \tag{4.3}$$

and

$$\partial^0 c_{0d}^\sigma \equiv \partial c_{0d}^\sigma \setminus \partial^1 c_{0d}^\sigma = \bigcup_\alpha \{ \text{face with } s_\alpha = 0 \}. \tag{4.4}$$

* Since $c_{\alpha\beta}^\alpha = c_{\beta\alpha}^\beta$, this face also has $s_\alpha = 1$, in the modified barycentric coordinates of c_β^σ.

The parallel transporter on link $\langle\alpha\beta\rangle$ will be denoted by $u_{\alpha\beta}$. On longer paths, parallel transporters are products of $u_{\alpha\beta}$, which will be abbreviated by $u_{\alpha\beta\cdots\gamma} = u_{\alpha\beta}u_{\beta *}\cdots u_{*\gamma}$.

4.2. Transition functions $v_{\alpha\beta}$

The algorithm is inductive. Let $\tau \prec \sigma$, and let d_τ (d_σ) be the dimension of simplex τ (σ). Then

$$v_{\alpha\beta}(c^\sigma_{\alpha\beta})\big|_\tau = v_{\alpha\beta}(c^\tau_{\alpha\beta}), \tag{4.5}$$

where the right-(left-)hand-side is given by the d_τ (d_σ) dimensional version of the algorithm. Thus, we present the prescription of ref. [5] in increasing dimension.

4.2.1. $d = 1$

At the midpoints of the link $\langle\alpha\beta\rangle$ the transition function is given by

$$v_{\alpha\beta}(p_{\alpha\beta}) = u_{\alpha\beta}. \tag{4.6}$$

This corresponds to choosing the smooth gauge $(x - x_\alpha)_\mu A^{(\alpha)}_\mu = 0$, called "radial gauge".

4.2.2. $d = 2$

Now the local ordering **o** plays a role. Set $\sigma = \langle 012\rangle$. When α and β are **o**-adjacent, the transition function remains constant on $c^\sigma_{\alpha\beta}$:

$$v_{01}(s_2) = u_{01} \text{ and } v_{12}(s_0) = u_{12}. \tag{4.7}$$

The arguments of the $v_{\alpha\beta}$ refer to the modified barycentric coordinates. For v_{02} there are two constraints, $v_{02}(s_1 = 0) = u_{02}$ from eq. (4.6) and $v_{02}(s_1 = 1) = u_{012}$ from eq. (4.7) and the cocycle condition eq. (3.4). Then on c^σ_{02}

$$v_{02}(s_1) = \mathfrak{g}_{012}(s_1), \tag{4.8}$$

where $\mathfrak{g}_{012}$ is the geodesic from u_{02} to u_{012}:

$$\mathfrak{g}_{012}(s_1) = u_{02}(u_{2012})^{s_1}. \tag{4.9}$$

4.2.3. $d = 3$

Set $\sigma = \langle 0123\rangle$. On the faces $\tau_i = \sigma\setminus\langle i\rangle$ the transition functions are already prescribed. For α and β **o**-adjacent, $v_{\alpha\beta}$ remains constant on $c^\sigma_{\alpha\beta}$. On c^σ_{02} and c^σ_{13} (i.e. when one vertex is **o**-between α and β) the interpolation is as in eq. (4.8):

$$v_{\alpha\beta}(s_\gamma, s_\delta) = \mathfrak{g}_{\alpha\gamma\beta}(s_\gamma), \tag{4.10}$$

where γ is the **o**-intervening vertex and δ is the remaining vertex.

On c^σ_{03} the interpolation must be two-dimensional, because v_{03} is prescribed already at p_{03} and on $\partial^1 c^\sigma_{03}$. The interpolation follows by exploiting the natural conical structure of c^σ_{03} with p_{03} as apex and $\partial^1 c^\sigma_{03}$ as base. Let $x \in \partial^1 c^\sigma_{03}$. Then the line segment $\overline{p_{03}x}$ is mapped to the shortest geodesic from $v_{03}(p_{03}) = u_{03}$ to $v_{03}(x) = v_{0\gamma}(x)\, v_{\gamma 3}(x)$, $\gamma = 1, 2$. In particular, $\partial^0 c^\sigma_{03}$ is mapped to the geodesics from u_{03} to u_{013} and to u_{023}. The resulting map will be denoted

$$v_{03}(s_1, s_2) = \mathfrak{h}_{0123}(s_1, s_2), \tag{4.11}$$

which, geometrically, is two geodesic triangles hinged at $s_1 = s_2$.

4.2.4. $d = 4$

Set $\sigma = \langle 01234\rangle$. Again, for α and β **o**-adjacent, $v_{\alpha\beta}$ remains constant on $c^\sigma_{\alpha\beta}$. Also, as for $d = 3$, if only γ **o**-intervenes α and β, then the algorithm prescribes geodesic interpolation in s_γ and is constant in s_δ and s_ϵ. If γ *and* δ are **o**-between α and β, then the two-dimensional interpolation described in section 4.2.3 is employed,

$$v_{\alpha\beta}(s_\gamma, s_\delta, s_\epsilon) = \mathfrak{h}_{\alpha\gamma\delta\beta}(s_\gamma, s_\delta). \tag{4.12}$$

On c^σ_{04} the conical structure $c^\sigma_{04} = p_{04} * \partial^1 c^\sigma_{04}$ is once again employed. The cocycle condition implies that $v_{04}(c^\sigma_{014}) = v_{01}(c^\sigma_{014})v_{14}(c^\sigma_{014})$ and $v_{04}(c^\sigma_{034}) = v_{03}(c^\sigma_{034})v_{34}(c^\sigma_{034})$ are both composed of two geodesic triangles, whereas $v_{04}(c^\sigma_{024}) = v_{02}(c^\sigma_{024})v_{24}(c^\sigma_{024})$ is the product of two one-dimensional maps like eq. (4.8). As before, for $x \in \partial^1 c^\sigma_{04}$ the line segment $\overline{p_{04}x}$ is mapped to the shortest geodesic from $v_{04}(p_{04}) = u_{04}$ to $v_{04}(x)$. Now the image

$$v_{03}(c^\sigma_{03}) = \mathfrak{k}_{01234}(s_1, s_2, s_3) \tag{4.13}$$

splits into four geodesic tetrahedra, and a pyramid with base $v_{04}(c^\sigma_{024})$.

4.3. Local sections g_α

Let us point out a few properties of the Phillips and Stone transition functions that are relevant to

the construction of the local sections. On $\partial^1 c^\sigma_{0d}$ the cocycle condition determines the transition function in terms of transition functions which are already prescribed by interpolations introduced for lower dimensional lattices. Moreover, the use of the conical structure guarantees that also on $\partial^0 c^\sigma_{0d}$ only lower dimensional interpolations are required. Consequently, but not obviously, the section for the d-dimensional lattice Λ can be obtained from the $d+1$-dimensional lattice $\hat{\Lambda}$. $\hat{\Lambda}$ is obtained from Λ by adding one new vertex χ, which is a *nearest-neighbor* to all $\alpha \in \Lambda$. On the new links $\langle \alpha\chi \rangle$ a new parallel transporter is defined $u_{\alpha\chi} = 1$, $\forall\alpha$. In the ordering of the vertices of a simplex, χ is taken to be the last vertex.

Let the transition function for $\hat{\Lambda}$ be denoted by V, and for $\sigma = \langle \alpha\beta\gamma\delta\epsilon \rangle \subset \Lambda$ write $\Sigma = \langle \alpha\beta\gamma\delta\epsilon\chi \rangle \subset \hat{\Lambda}$. From the inductive nature of the algorithm $V_{\alpha\beta} = v_{\alpha\beta}$, if $\alpha, \beta \neq \chi$. Now consider the transition functions V at double overlaps of the form $c^\Sigma_{\alpha\beta\chi}$. Note that $c^\Sigma_{\alpha\beta\chi}$ in $\hat{\Lambda}$ has the same modified barycentric coordinates as $c^\sigma_{\alpha\beta}$ in Λ. By the cocycle condition

$$v_{\alpha\beta} = V_{\alpha\chi} V_{\chi\beta} = V_{\alpha\chi} (V_{\beta\chi})^{-1}, \tag{4.14}$$

which is precisely eq. (3.6). Hence,

$$g_\alpha = V_{\alpha\chi} \tag{4.15}$$

is a section for the bundle defined by the $v_{\alpha\beta}$ for Λ.

For $d < 4$ the maps V are given by the maps $\mathfrak{g}$, $\mathfrak{h}$ and $\mathfrak{k}$ described in section 4.2. These maps are continuous and (piecewise) differentiable, so that the resulting section $g = V$ is defined on all of c^σ_α and not just on ∂c^σ_α. This implies that the winding numbers are all zero: there are no nontrivial SU(N) bundles over spacetimes with $d < 4$.

In four dimensions the maps V define a section on all of c^σ_α, unless α is the **o**-first vertex of σ. On $\partial^1 c^\Sigma_{0\chi}$, $V_{0\chi}$ is given (through the cocycle condition) by products $V_{0i} V_{i\chi}$, $i = 1, \ldots, 4$. These have nontrivial dependences on *three* modified barycentric coordinates, and hence the possibility that $V_{0\chi}(\partial^1 c^\Sigma_{0\chi}) = -1$ has nonzero measure. In this case, the geodesic from $V_{0\chi}(p_{0\chi}) = 1$ to -1 is not well defined. Under these conditions, the section has nonzero winding number.

The computation of the winding numbers will be discussed in section 6. The polyhedra D mentioned above are determined by the map $\mathfrak{k}$ on $\partial^0 c^\sigma_{0\chi}$, and by maps of the form $u\mathfrak{k}$, $\mathfrak{k}u$, $\mathfrak{g}\mathfrak{h}$ and $\mathfrak{h}\mathfrak{g}$ on $\partial^1 c^\sigma_{0\chi}$. The forms $\mathfrak{g}\mathfrak{h}$ and $\mathfrak{h}\mathfrak{g}$ describe two geodesic prisms, whereas $\mathfrak{k}$, $u\mathfrak{k}$ and $\mathfrak{k}u$ describe four geodesic tetrahedra and a pyramid. The central part of QUBIC, therefore, is a set of subroutines, which determine the existence and number of points $\underline{x}$, as well as the orientation of $g_\alpha(\underline{x})$.

5. Hypercubic lattices from simplicial lattices

The algorithm of Phillips and Stone relies on the geometry of a simplicial lattice, but most lattice gauge theory simulation programs use hypercubic lattices. To extend the algorithm to hypercubic lattices, it suffices to define a simplicial lattice gauge field that is as close as possible to the hypercubic lattice. This is done in two steps. First the hypercubes must be "sliced" into simplices in a way that fills space without any gaps; we choose the minimal procedure. Second one must assign parallel transporters to the new links created during the slicing step. Here we state only the results; for a more thorough discussion see ref. [3].

Let us label the corners of a hypercube from 0 to 15, as shown in fig. 1, using hexadecimal notation. The minimal triangulation requires 24

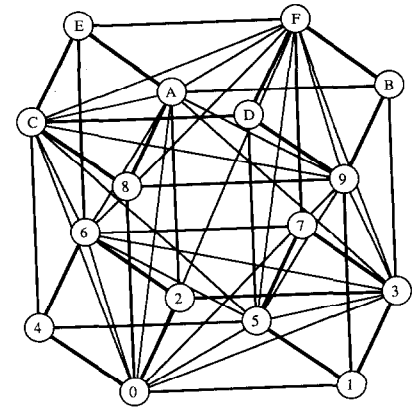

Fig. 1. Illustration of the minimal triangulation of a four-dimensional hypercube.

plaquette diagonals: ⟨03⟩, ⟨05⟩, ⟨06⟩, ⟨09⟩, ⟨0A⟩, ⟨0C⟩, ⟨35⟩, ⟨36⟩, ⟨39⟩, ⟨3A⟩, ⟨3F⟩, ⟨56⟩, ⟨59⟩, ⟨5C⟩, ⟨5F⟩, ⟨6A⟩, ⟨6A⟩, ⟨6A⟩, ⟨9A⟩, ⟨9C⟩, ⟨9F⟩, ⟨AC⟩, ⟨AF⟩, ⟨CF⟩; and one body diagonal: ⟨0F⟩. The hypercube is then sliced into the following simplices:

⟨01359⟩, ⟨0235A⟩, ⟨0456C⟩, ⟨089AC⟩,
⟨3567F⟩, ⟨39ABF⟩, ⟨59CDF⟩, ⟨6ACEF⟩,
⟨0359F⟩, ⟨036AF⟩, ⟨056CF⟩, ⟨09ACF⟩,
⟨0356F⟩, ⟨039AF⟩, ⟨059CF⟩, ⟨06ACF⟩. (5.1)

Note that the orientation of these simplices, relative to the hypercube, are +1, −1, +1, −1, +1, −1, +1, −1, +1, −1, +1, −1, +1, −1, +1, −1, respectively.

In slicing the whole lattice, one must take care to treat compatible the 3-cubes at the intersection between neighboring 4-cubes. Having sliced a given hypercube, one cannot translate it to one of its nearest neighbors; instead one must reflect about the common 3-cube. Successive reflections in all four dimensions will build up a 2^4 block, which can then be translated throughout space-time. This, incidentally, is the origin of the requirement that the lattice size parameters NS and NT be even.

The last step is to define parallel transporters for the new links. For the 24 plaquette diagonals one chooses the element of SU(2) such that parallel transport around each triangle is "half" that around the plaquette, i.e.

$$u_{03} = \left[u_{01}u_{13}u_{23}^{-1}u_{02}^{-1}\right]^{1/2}u_{02}u_{23}. \tag{5.2}$$

For the body diagonal there are now three "grand plaquettes" across which we can interpolate: [0, 3, F, C, 0], [0, 5, F, A, 0] and [0, 6, F, 9, 0]. One wants to pick the matrix that yields, on the whole, the smallest possible parallel transport around closed loops, which implies that one should apply a formula of the form eq. (5.2) to the grand plaquette with the *largest* parallel transport.

Of the two steps, only the second needs to be explicitly implemented on the computer. The first step is, however, reflected in the names of variables, e.g. in EVAL.

6. Description of program

This section provides a brief description of the various subroutines. The structure of the program is sequential, as one can see from a glance at the code for subroutine QUBIC. (See table 1 for an excerpt.) The complexity lies not in the flow, but in the actual computations themselves. Indeed, the subdivision by subroutine is guided by the conceptual nature of the tasks, rather than dictated by flow control.

The program assumes that the lattice gauge field is stored in common block LINKS0 according to the three-dimensional checkerboard scheme, with NCHECK colors. This implies that NS be a multiple of NCHECK as well as of 2; clearly NCHECK = 2 is the most natural choice.

The program is restricted to $NS^3 \times NT$ lattices with periodic boundary conditions. In the following we will occasionally speak of, e.g. IT ± 1, which is to be understood modulo NT.

6.1. MAIN

The are two installation dependent aspects: the random number generator RANF, and the elapsed CPU-time function SECOND. Users of other computers than the Cray must change these

Table 1
Excerpt of FORTRAN code from Q1 illustrating the structure of the program

```
      ICHERN = 0
      DO 100 IT = 1, NT
      DO 100 ICHECK = 1, NCHECK
      TIME2 = SECOND()
      CALL HCUBE(ICHECK,IT)
      TIME3 = SECOND()
      TIMING(3) = TIMING(3) + TIME3 - TIME2
      CALL INTERP
      TIME4 = SECOND()
      TIMING(4) = TIMING(4) + TIME4 - TIME3
      CALL EVAL
      TIME5 = SECOND()
      TIMING(5) = TIMING(5) + TIME5 - TIME4
      CALL CHRGE2
      TIME6 = SECOND()
      TIMING(6) = TIMING(6) + TIME6 - TIME5
100   CONTINUE
```

accordingly. (Note that RANF appears in an EXTERNAL statement.) Both of these aspects are irrelevant to production runs. In particular, the random numbers are used only by Q1 to construct the test configuration.

6.2. Q1

Although the intent of this program is to determine the topological charge of lattice gauge fields generated in the course of a Monte Carlo simulation, it is useful to have access to a configuration with $Q \neq 0$. The $Q = 1$ configuration which is described in the appendix is created by Q1. Neither this subroutine, nor those called by it, have been optimized. Q1 has two arguments. ISTOER is a flag: for ISTOER = 1 the ideal configuration described in the appendix is deformed slightly, to avoid run-time errors in later 4×4 determinants. RNDM is the user's favorite generator of a single random number, uniform on the interval (0, 1).

6.3. QUBIC

QUBIC is the controlling subroutine for the measurement of the topological charge. It contains outer loops over NT timeslices (labeled by IT) and the NCHECK checkerboard colors (labeled by ICHECK) (cf. table 1). The strategy is that HCUBE gathers all link matrices needed for evaluation of the Q_α for fixed IT and ICHECK.

The user is invited to modify QUBIC to suit his needs. In particular, those who use asynchronous I/O to reduce the core memory demands ought to insert the appropriate (system dependent) calls here. Since the common block LINKS0 will then contain only a subset of the lattice, the declaration

```
COMMON/LINKS0/
  U0(NSLICE,NU2,4,NCHECK,NT)        (6.1)
```

must be amended in HCUBE and inserted into QUBIC. Such users are advised that HCUBE requires timeslices IT and IT + 1. See also the remarks in section 6.6.

The remark concerning the timing function SECOND in section 6.1 also applies here. Moreover, in production runs users will probably want to erase these lines.

6.4. HCUBE

HCUBE gathers link matrices from timeslices IT and IT + 1 of the array U0 into the array U. This subroutine has not been fully optimized – readability and flexibility take priority here. The time spent in HCUBE is not great, so the latter concerns outweigh efficiency.

Some users prefer (or are constrained to) storage schemes other than the three-dimensional NCHECK-colored checkerboard. They will need to modify HCUBE. Retention of the general DO-loop structure and the indexing of U is imperative. The assignments to JX, JY, JZ and JT must also be left alone. Modifications must be restricted to indices labelling U0 – in the present case JSLICE and JCHECK. Indices for other schemes must be formed out of the spacetime lattice coordinates as presented in table 2. For example, if the gauge field U0 is declared as

```
COMMON/LINKS0/
  U0(NU2,4,NS,NS,NS,NT),            (6.2)
```

then U0 should be referenced as U0(IU2,1,IX,JY, JZ,JT) when "collecting the links in the 1-direction".

The orientations of the hypercubes and 4-simplices are also computed in HCUBE. Recall from section 5 that eight of the simplices within a hypercube are positively oriented and eight are negatively oriented (with respect to the hypercube). Furthermore, the alternating pattern of the sliced hypercubes introduces another orientation factor. These two orientation factors are coded into IRIENT.

Table 2
For each direction μ in the lattice, the array indices for U0 must be constructed from the tabulated variables

μ	Variables
1	IX, JY, JZ, JT
2	JX, IY, JZ, JT
3	JX, JY, IZ, JT
4	JX, JY, JZ, KT

6.5. INTERP and PLAQ

The construction of the minimal simplicial lattice from a hypercubic lattice, as described in section 5, calls for 24 plaquette diagonals and one body diagonal. HCUBE was designed so that the contents of LINKS2 can also be referred to by the more mnemonic nomenclature in INTERP. The arrays U*nm* are parallel transporters from site *n* to *m*, where *n* and *m* label the corners of the hypercube. The notation is that of fig. 1. INTERP calls PLAQ in a rather transparent way to construct the new parallel transporters of the simplicial lattice. The call

$$\text{CALL PLAQ(A,U03,U01,+1,U13,+1,U23,}\\ \text{-1,U02,-1),} \tag{6.3}$$

corresponds to

$$u_{03} = \left[u_{01}u_{13}u_{23}^{-1}u_{02}^{-1}\right]^{1/2}u_{02}u_{23},\\ A = \cos^{-1}\left[\mathrm{Tr}\left(u_{01}u_{13}u_{23}^{-1}u_{02}^{-1}\right)/2\right]; \tag{6.4}$$

eq. (6.4) is executed on a vector of length NSLICE, corresponding to the NSLICE hypercubes gathered into LINKS2.

6.6. EVAL

EVAL gathers the parallel transporters of the sliced hypercubes (stored in LINKS1) into a canonical ordering for the individual simplices. The parallel transporters of the simplices are passed via COMMON block LINKS2 to CHRGE2 for the computation of the winding numbers. LINKS2 is the largest COMMON block in the program (unless NT $\geq$ 42), and it can pose memory problems for large lattices. We have coded EVAL in a transparent, albeit brute force, manner in anticipation of users who will need to perform some sort of restructuring here. For example, one may want to devise a scheme with a loop over the 16 different simplices.

6.7. CHRGE2

CHRGE2 codes the information of table 4.1 in ref. [5]. The orientation factor IPS gives the relative orientation of the polyhedron *D*. Since the same polyhedra occur over and over again, but with different y_i, these are handled by subroutines, described below.

Users who work with simplicial lattices can discard the subroutines described above and call CHRGE2 directly. They should change the parameter NCELL in CHRGE2, in the other subroutines in this subsection, and in DCELL (section 6.8) to suit their needs. (NCELL is the number of simplices processed at a time.) These users must also compute the orientations of the simplices, i.e. IRIENT.

Especially for smooth configurations, the polyhedra *D* can become quite "small". It is therefore of paramount importance that the operations performed in the subroutines called by CHRGE2 are carried out in 64-bit arithmetic. Users who work with 32-bit words should therefore take care that this part of the program is changed to double precision.

Below we shall see that many of the arithmetic operations are in 4×4 determinants. Many of these involve the probe y. Since the winding number Q_α is independent of y, we choose $y = (0, 0, 1, 0)$, which reduces the determinants involving y to 3×3. This feature is built into all of the subroutines called by CHRGE2.

6.7.1. SIMPLX and SIMPL1

The most common structure in the algorithm is the tetrahedron, or 3-simplex. A tetrahedron has four vertices, whose images are given by y_1, y_2, y_3 and y_4, and four faces, which are the geodesic triangles defined by three out of the four y_i. $g_\alpha(D)$ contains the probe if

$$\begin{aligned}&\text{sign det}(y_1, y_2, y_3, y_4)\\ &\quad=\text{sign det}(y, y_2, y_3, y_4)\\ &\quad=\text{sign det}(y_1, y, y_3, y_4)\\ &\quad=\text{sign det}(y_1, y_2, y, y_4)\\ &\quad=\text{sign det}(y_1, y_2, y_3, y),\end{aligned} \tag{6.5}$$

because $\det(a, y_2, y_3, y_4)$ and $\det(b, y_2, y_3, y_4)$ have the same (opposite) sign if a and b are on the same (opposite) side of the spherical triangle defined by y_2, y_3 and y_4. If eqs. (6.5) hold, then

D yields a contribution to Q_α, equal to the orientation factors of D (IRIENT(IC) * IPS) multiplied by

$$\text{sign det}(y_1, y_2, y_3, y_4), \tag{6.6}$$

which is the orientation of $g_\alpha(D)$.

Frequently it turns out that $y_1 = (0, 0, 0, 1)$, in which case the determinants in eqs. (6.5) can be reduced further. SIMPL1 handles this situation.

6.7.2. PYRAMD and PYRAM1

The pyramid has the structure of a cone from the vertex x_0 (with image y_0) to a square base $R = [x_1, x_2, x_3, x_4]$ *. The sides of the cone are four triangular faces. The cone structures arises from the mapping $g_\alpha(D)$. From section 4, for $x_R \in R$, the line segment $\overline{x_0 x_R}$ is mapped to the geodesic from $g_\alpha(x_0) = y_0$ to $g_\alpha(x_R)$. The pyramid D contains the probe y whenever one of these geodesics $g_\alpha(\overline{x_0 x_R})$ passes through y.

In general, $g_\alpha(R)$ is not flat, but can be curved as illustrated in fig. 2. Nevertheless, the edges of $g_\alpha(R)$ can be extended to define a surface in S^3 denoted by cQ. The strategy is to determine the number of times the great circle through y_0 and y can intersect cQ; this can be up to two. The description is most natural in coordinates defined by y_1, y_2, y_3 and y_4. Then the intersection of the great circle with cQ is determined by the zeroes of a certain quadratic form (see ref. [5] for details). The number of real zeroes is the number of intersections. Moreover, if the zero is *positive*, then the probe is inside the cone defined by the four triangular faces.

The orientation of the section at the probe is IHZ * I0: IHZ is minus the slope of the quadratic form, evaluated at the zero at hand, and I0 is the relative orientation of the y_i, relative to the x_i. If both real zeroes are positive, then the two IHZ are opposite, and the pyramid gives no net contribution to Q_α.

The second DO-loop determines whether there is exactly one real positive root of the quadratic equation. It exploits the logic functions OR and XOR, which are bit-wise OR and exclusive-OR, respectively. Although almost all dialects of FORTRAN 77 have these functions, the names are not standard. For example, on Fujitsu and IBM machines OR and XOR are called IOR and IEOR, respectively.

PYRAM1 handles the common case $y_0 = (0, 0, 0, 1)$.

6.7.3. PRISM2 and PRISM3

The prisms have two triangular bases and three square sides. The vertices of $g_\alpha(D)$ are y_1, y_2, y_3, y_4, y_5 and y_6. The simplest way to determine the position of the probe relative to $g_\alpha(D)$ focuses on one triangular base, denoted Δ^2 and its image $g_\alpha(\Delta^2)$.

* Strictly speaking, R is a square only in the "modified barycentric coordinates", cf. ref. [5].

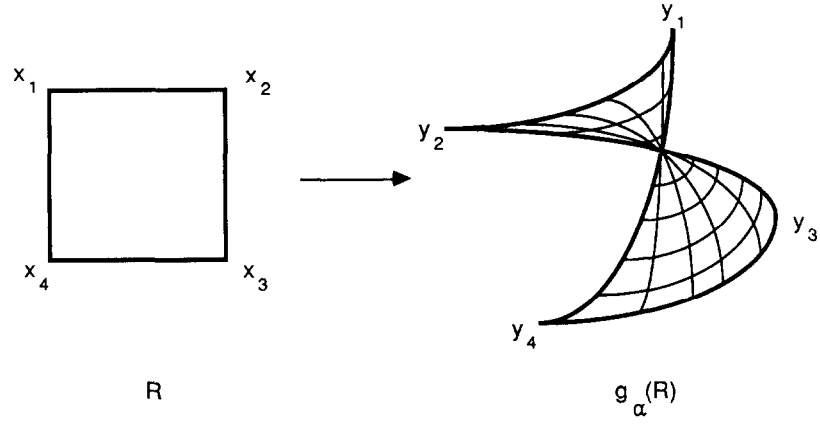

Fig. 2. $g_\alpha(R)$ lives in S^3, which is a curved space.

For the case handled by PRISM2, the section has the form $\mathfrak{g}\mathfrak{h}$:

$$g_\alpha(D) = g_\alpha(\Delta^1) g_\alpha(\Delta^2), \tag{6.7}$$

where $g_\alpha(\Delta^1)$ is the geodesic from u_{02} to u_{012}, and $g_\alpha(\Delta^2)$ is the geodesic triangle * with vertices y_1, y_5 and y_6. Let $z = u_{0210} y$. The probe is inside $g_\alpha(D)$ only if y and z are on *opposite* sides of $g_\alpha(\Delta^2)$, i.e. only if

$$\begin{aligned}&\text{sign det}(y,\ y_1,\ y_5,\ y_6)\\ &\quad = -\text{sign det}(z,\ y_1,\ y_5,\ y_6).\end{aligned} \tag{6.8}$$

If eq. (6.8) holds, one of y and z is between the surfaces defined by the images of the two triangular bases. If

$$\begin{aligned}&\text{sign det}(y,\ y_1,\ y_5,\ y_6)\\ &\quad = \text{sign det}(y,\ z,\ y_5,\ y_6)\\ &\quad = \text{sign det}(y,\ y_1,\ z,\ y_6)\\ &\quad = \text{sign det}(y,\ y_1,\ y_5,\ z),\end{aligned} \tag{6.9}$$

then y lies between the two surfaces, and furthermore $y \in g_\alpha(D)$.

The orientation of $g_\alpha(\underline{x})$ is determined by two factors. First one has the orientation relative to x_1, x_2, x_5 and x_6:

$$\text{sign det}(y_1,\ y_2,\ y_5,\ y_6). \tag{6.10}$$

Second, because of the intrinsic curvature of S^3, it is possible that $g_\alpha(\Delta^2) \cap u_{0120} g_\alpha(\Delta^2) \neq \emptyset$. The orientation of g_α changes at this intersection. Hence one has an additional orientation factor

$$\text{sign det}(y,\ y_2,\ y_3,\ y_4)\ \text{sign det}(y_1,\ y_2,\ y_3,\ y_4), \tag{6.11}$$

which is positive (negative) if y and y_1 are on the same (opposite) side(s) of $u_{0120} g_\alpha(\Delta^2)$, the geodesic triangle defined by y_2, y_3 and y_4.

For the case handled by PRISM3 the situation is similar. Now the section has the form $\mathfrak{h}\mathfrak{g}$:

$$g_\alpha(D) = g_\alpha(\Delta^2) g_\alpha(\Delta^1), \tag{6.12}$$

where $g_\alpha(\Delta^1)$ is the geodesic from 1 to u_{34}, and $g_\alpha(\Delta^2)$ is the geodesic triangle with vertices y_1, y_2 and y_3. Now let $z = y u_{43}$. Then $y \in g_\alpha(D)$ under the circumstances of eqs. (6.8) and (6.9) with the replacements

$$y_5 \leftrightarrow y_2 \text{ and } y_6 \leftrightarrow y_3. \tag{6.13}$$

Indeed, one subroutine could handle both types of prisms, but we have written two, so that CHRGE2 renders table 4.1 of ref. [5] more faithfully.

6.8. Utilities

There are two simple utility subroutines: MULT2C and DCELL.

6.8.1. MULT2C

A call of the form

```
CALL MULT2C(W,U,J,V,K,NMULT)
```

carries out the SU(2) matrix multiplication (J, $K = \pm 1$)

$$W = U^J V^K$$

for vectors of length NMULT. Note that J and K are scalars: the choice of SU(2) matrix or its inverse is assumed constant throughout the vector.

6.8.2. DCELL

A call of the form

```
CALL DCELL(DET,Y1,Y2,Y3,Y4)
```

computes 4×4 determinants:

$$\text{DET} = \det(y_1,\ y_2,\ y_3,\ y_4)$$

for vectors of length NCELL. DCELL is called seldom, because most of the 4×4 determinants have been reduced to 3×3, or even 2×2, determinants, and the code for such small determinants appears directly where needed.

7. Test results / performance

Typical output from the test package is displayed in table 3, in this case from the Cray X-MP/48 in Jülich with the CFT compiler. The output may not seem very informative, but we

* $g_\alpha(\Delta^2_{\text{this paper}}) = y_1 g_\alpha(\Delta^2_{\text{ref.[5]}})$.

Table 3
Output of the test package, giving times (in s) from a Cray X-MP/48

```
LATTICE SIZE IS NS = 10, NT = 10
   CREATED A FIELD WITH CHARGE 1
   PERTURBED THE FIELD A LITTLE BIT AWAY
   FROM IDEAL
TOPOLOGICAL CHARGE Q = 1
ELAPSED TIME IN CPU:
FOR Q1       : 4.81106189E+00  UNOPTIMIZED!!!
FOR QUBIC    : 9.13974316E+00  TOTAL
FOR HCUBE    : 1.57520124E+00
FOR INTERP   : 3.74265257E-01
FOR EVAL     : 9.27934496E-01
FOR CHRGE2   : 6.26228003E+00  INCLUDES SUB-
   SUBROUTINES
```

remind potential users that the package is meant to be built into a larger simulation of SU(2) gauge theory. Then all one really needs out of QUBIC is the topological charge Q of the sequence of Monte Carlo generated configurations.

We have tested the package on several machines. For two supercomputers the performance is listed in table 4. The listed Mflops number is computed from the run time and operation count of QUBIC only. On the Cray, we have used 46-bit INTEGER arithmetic, which is considerably faster than the default 64-bit arithmetic.

Table 4
Timings for two supercomputers, for a 10^4 lattice

Computer	Location	Time for QUBIC	Mflops
Cray X-MP/48	HLRZ, c/o KFA, Jülich, FRG	9.2 s	49
Fujitsu VP100	Universität Kaiserslautern, FRG	4.3 s	105

Table 5
Parameters appearing in QUBIC

NS	number of sites in the space directions
NT	number of sites in the time direction
NCHECK	number of colors in the (three-dimensional) checkerboard
NSLICE	number of sites in a timeslice on a particular checkerboard color = NS^3/NCHECK
NU2	number of REAL words needed to store an SU(2) matrix = 4
NCEPH	number of simplices in a sliced hypercube = 16
NCELL	number of simplices processed by CHRGE2

Table 6
COMMON blocks appearing in QUBIC

LINKS0	contains hypercubic lattice gauge field
LINKS1	contains part of simplicial lattice gauge field being processed
LINKS2	contains gauge field sorted into simplices
CHARGE	topological charge and orientation factors
COORDS	useful arrays used in HCUBE
TIMES	used for timing

Table 7
Important variables appearing in QUBIC

U0	contains hypercubic lattice gauge field
U	contains part of simplicial lattice gauge field being processed
V's	contains gauge field sorted into simplices
ICHERN	the topological charge Q
IPS	orientation of D with respect to the simplex
IRIENT	orientation of the simplex with respect to Λ
LA, LB, LC	useful arrays used in HCUBE
TIMING	used for timing

8. Variable lists

Tables 5, 6 and 7 contain lists of parameters, COMMON blocks, and important variables in QUBIC. Among the parameters, NU2 and NCEPH must *not* be changed. NCELL may only be changed in conjunction with some restructuring, e.g. of EVAL, cf. section 6.6, or for a simplicial lattice.

Acknowledgements

We would like to thank the Höchstleistungsrechenzentrum HLRZ at the KFA Jülich and the Universität Kaiserslautern for the opportunity to develop and run QUBIC. M.L.L. and C.S. thank R.D. Peccei for hospitality during visits to DESY; A.S.K. thanks P. Sauer for hospitality during a

visit to the Universität Hannover. Finally, we thank M. Kremer for the collaboration in ref. [4].

Appendix. A lattice gauge field with $Q = 1$

For installation of QUBIC it is useful to have access to a lattice gauge field with $Q = 1$, by construction. It follows from the general discussion of section 3, that "all" one needs to do is to introduce a nontrivial transition function at the overlap of two cells. One such construction, inspired by the instanton solution, was given in ref. [19]. There the nontrivial transition function is defined on the surface of a hypercube situated in the "middle" of the 4-torus. Here we employ an even simpler construction of Lüscher [20], where the transition function is defined on a surface of constant time.

First some notation. Consider the 4-torus of size L^3T and take the fundamental domain of the coordinates to be

$$-L/2 \le x_i \le L/2, \quad i = 1, 3;$$
$$-T/2 \le x_4 \le T/2. \tag{A.1}$$

The spacelike components form the vector $\boldsymbol{x}$, and write $\boldsymbol{z} = \boldsymbol{x}/L$, $z = |\boldsymbol{z}|$.

We wish to construct a gauge potential with the following boundary conditions:

$$A_\mu(\boldsymbol{x}, -T/2) = g(\boldsymbol{x})\left[\partial_\mu + A_\mu(\boldsymbol{x}, T/2)\right] g^{-1}(\boldsymbol{x}), \tag{A.2}$$

as well as strictly periodic in the spacelike directions. The winding number of g is then equal to the topological charge. Consider the following map:

$$g\colon \mathbb{T}^3 \to \mathrm{SU}(2) \approx S^3$$
$$\boldsymbol{x} \mapsto \exp[\mathrm{i}\boldsymbol{\sigma} \cdot \boldsymbol{z} f(z)], \tag{A.3}$$

where

$$zf(z) = \pi z(2 + 11z^2 - 72z^4 + 112z^6), \quad z \le 1/2$$
$$= 2\pi, \quad z \ge 1/2; \tag{A.4}$$

the polynomial has been chosen so that $zf(z)$ is twice differentiable, even at $z = 1/2$. As $\boldsymbol{x}$ varies over the domain indicated in eq. (A.1), $g(\boldsymbol{x})$ ranges over SU(2) exactly once – i.e. the map has unit winding number.

To pick a smooth gauge potential in the $Q = 1$ sector, we start with $A_\mu(\boldsymbol{x}, T/2) = 0$, and hence $A_i(\boldsymbol{x}, -T/2) = g(\boldsymbol{x})\partial_i g^{-1}(\boldsymbol{x})$, $A_4(\boldsymbol{x}, -T/2) = 0$. For $x_4 \in (-T/2, T/2)$ we take

$$A_i(x) = h(x_4) g(\boldsymbol{x}) \partial_i g^{-1}(\boldsymbol{x}),$$
$$h(x_4) = \frac{1}{2}\left[1 - \sin\left(\frac{\pi x_4}{T}\right)\right] \tag{A.5}$$

and $A_4(x) = 0$.

The potential A_μ is defined on the continuum torus. Now we need to set up the lattice gauge field, i.e. a set of parallel transporters defined on the links of a lattice embedded inside the torus. Neglecting transition functions, parallel transport is given by

$$U(x, y) = P \exp\left(\int_x^y \mathrm{d}x_\mu \, A_\mu\right), \tag{A.6}$$

where the integration is taken along some path connecting x and y, and P denotes path-ordering. A lattice gauge field consists of link matrices given by the parallel transport from x to $y = x \pm \hat{\mu} a$ along a straight-line path. These will be denoted $U_\mu(x) := U(x, x + \hat{\mu} a)$.

It is useful to visualize the cells of section 3 as the timeslices of the lattice. Then all transition functions are trivial, $v_{t,t+\hat{4}a} = 1$, *except* for $v_{T/2,-T/2} = g^{-1}$. With nontrivial transition functions, parallel transport is no longer given by eq. (A.6), but by a product of path-ordered exponentials (within cells) and transition functions (between cells). Hence, for the present case, the transition function only needs to be taken into account for the link matrices $U_4(\boldsymbol{x}, T/2)$. Since $A_4 = 0$ everywhere, the result is *

$$U_4(\boldsymbol{x}, T/2) = g(\boldsymbol{x}). \tag{A.7}$$

The other link matrices in the lattice gauge field do not notice the transition function, and thus are given by numerical integration of eq. (A.6). Since

* Note that the naive conclusion $U_4 = 1$ is just plain wrong, and it leads to a lattice gauge field with $Q = 0$.

$A_4 = 0$ everywhere, $U_4(\boldsymbol{x}, x_4 \neq T/2) = 1$. For the spacelike links, we cut the path (of length a) into N_{cut} smaller paths (of length $\delta = a/N_{\text{cut}}$) and compute the parallel transport there as follows. The time dependence is contained in an $\boldsymbol{x}$-independent factor, so we must consider the integral

$$I_i^k = \int_{x+\hat{\imath}k\delta}^{x+\hat{\imath}(k+1)\delta} \mathrm{d}x_i \; g(x)\partial_i g^{-1}(x), \text{ no sum on } i. \tag{A.8}$$

This can be approximated by

$$\begin{aligned} I_i^k \approx & \left[g(x+\hat{\imath}k\delta)g^{-1}(x+\hat{\imath}(k+1)\delta) \right. \\ & \left. - g^{-1}(x+\hat{\imath}(k+1)\delta)g(x+\hat{\imath}k\delta) \right]/2 \\ & + \mathcal{O}(\delta^3). \end{aligned} \tag{A.9}$$

On the short paths we then have

$$V_i^k = \exp\left(h(x_4)I_i^k\right), \tag{A.10}$$

from which we obtain the full parallel transport

$$U_i(x) = \prod_{k=0}^{N_{\text{cut}}-1} V_i^k. \tag{A.11}$$

Consistent with path-ordered the V_i^k with smaller k appear to the left.

References

[1] K.G. Wilson, Phys. Rev. D 10 (1974) 2445; also reprinted in ref. [2].

[2] Reprints of early work have been collected by C. Rebbi, Lattice Gauge Theories and Monte Carlo Simulations (World Scientific, Singapore, 1983).
For recent progress see Field Theory on the Lattice, eds. A. Billoire et al., Nucl. Phys. Proc. Suppl. B 4 (1988).

[3] A.S. Kronfeld, M.L. Laursen, G. Schierholz and U.-J. Wiese, Nucl. Phys. B 292 (1987) 330.

[4] M. Kremer, A.S. Kronfeld, M.L. Laursen, G. Schierholz, C. Schleiermacher and U.-J. Wiese, Nucl. Phys. B 305 [FS23] (1988) 109.

[5] A. Phillips and D. Stone, Commun. Math. Phys. 103 (1986) 599.

[6] A.S. Kronfeld, Nucl. Phys. Proc. Suppl. B 4 (1988) 329.

[7] A.A. Belavin, A.M. Polyakov, A.S. Schwartz and Y.S. Tyupkin, Phys. Lett. B 59 (1975) 85.

[8] C.G. Callan, R.F. Dashen and D.J. Gross, Phys. Lett. B 63 (1976) 334.
R. Jackiw and C. Rebbi, Phys. Rev. Lett. 37 (1976) 172.

[9] A.S. Kronfeld, M.L. Laursen, G. Schierholz, C. Schleiermacher and U.-J. Wiese, Nucl. Phys. B 305 (1988) 661.

[10] S. Weinberg, Phys. Rev. D 11 (1975) 3583.

[11] J. Schwinger, Phys. Rev. 82 (1951) 664.
S. Adler, Phys. Rev. 117 (1969) 47.
J.S. Bell and J. Jackiw, Nuov. Cim. 60A (1969) 529.

[12] G. 't Hooft, Phys. Rev. Lett. 37 (1976) 8; Phys. Rev. D 14 (1976) 3432.

[13] E. Witten, Nucl. Phys. B 156 (1979) 269.

[14] G. Veneziano, Nucl. Phys. B 159 (1979) 213.

[15] M. Lüscher, Commun. Math. Phys. 85 (1982) 29.

[16] M. Daniel and C.M. Viallet, Rev. Mod. Phys. 52 (1980) 175.

[17] P. van Baal, Commun. Math. Phys. 85 (1982) 529.

[18] M.L. Laursen, G. Schierholz and U.-J. Wiese, Commun. Math. Phys. 103 (1986) 693.

[19] I.A. Fox, M.L. Laursen, G. Schierholz, J.P. Gilchrist and M. Göckeler, Phys. Lett. B 158 (1985) 332.

[20] M. Lüscher, unpublished (1985).