Chapter 2
Numerical Methods

In our numerical analysis of quantised vortex tangle in random waves, we must be able to locate vortices numerically in three-dimensional complex fields, recovering both their local geometry well enough for comparison with known geometrical results and their global topology such that no information about how they wind about each other is missed. In this chapter we describe and explain the numerical techniques that we use to do so.

We begin in Sect. 2.1 by discussing the various techniques used in the literature to locate vortices in both two and three dimensions via numerical discretisation of the field in which they are present. Section 2.2 describes our own algorithm, extending these basic techniques with our own additions to create an algorithm that can efficiently trace vortex lines on large lengthscales by recursively focusing numerical effort on their local surroundings. In Sect. 2.3 we describe the necessary adaptations to this algorithm in order to best capture the nuances of the three different systems in which we will investigate random waves; the 3-torus, 3-sphere and quantum harmonic oscillator. Section 2.4 follows with the analytical and numerical details of recovering geometrical statistics of these vortices, accounting for the local details of each system, and in Sect. 2.5 we list all the details of the numerical datasets from which most of our later results are drawn.

2.1 Numerically Locating Vortices

A central problem in this thesis is the numerical detection and tracking of quantised vortices in random waves as they wind through three-dimensional space, accurately reconstructing their local space curve geometry, correctly resolving the way nearby vortex segments wind around one another, and doing so efficiently enough to sample large volumes of random vortex tangle. Similar problems in resolving properties of vortex filaments occur in many different areas of physics or mathematics, both related

A.J. Taylor, *Analysis of Quantised Vortex Tangle*,
Springer Theses, DOI 10.1007/978-3-319-48556-0_2

to our random waves and in entirely different contexts. We describe and compare here existing methods to do so in two and three dimensions, including an analysis of which techniques are most suited to locating vortices in our own complex scalar fields. This establishes the basic principles for our own tracking algorithm which extends some of these techniques, explained in Sect. 2.2.

The most directly similar methods in the literature include tracing nodal lines in random wave models [1, 2] or other numerically simulated fields [3], tracked vortices in optical beams [4–7], and certain simple discrete models of cosmic strings [8], as well as the detection of vortices in fields that are still complex scalar but with more complicated driving physics, such as vortices in Bose-Einstein condensates through simulations of the Gross-Pitaevskii equation [9, 10]. The same ideas appear more widely in understanding the unquantised vortices of classical turbulence [11] and in locating the more varied types of director field singularity in liquid crystals [12] or polarisation fields [13]. Detecting phase singularities is even important either physically or algorithmically to a range of physical image analysis problems, including understanding voltage patterns in the heart [14–16], interpreting MRI data [17, 18], or in reliably 'phase-unwrapping' noisy data that has been measured or stored modulo some constant [19]. Not all of these are directly relevant to our own work, but most make use of the same core ideas, which we will explore further below.

The simplest relevant problem is that of detecting vortices in a two-dimensional complex scalar field, where the vortices are point defects in the phase; if the field is a planar slice of a 3D system, these points are where one-dimensional vortex curves pass through the plane. Figure 2.1 shows an example area of a monochromatic Gaussian random field of the type discussed in Sect. 1.4; the side length of each image is 3.8 λ, covered with a numerical grid of side length 350, easily enough to show all

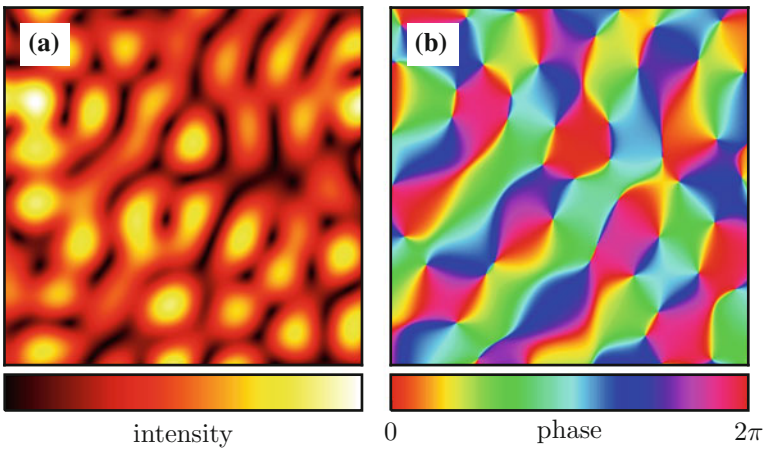

Fig. 2.1 A region of an example two-dimensional Gaussian random wavefunction. The side length of the given square is 3.8 λ, and **a**, **b** show the intensity and phase respectively. The numerical problem discussed in this section is how to locate the vortices in the phase efficiently and reliably

features of the field in high detail. The task of this Section is to locate its vortices with far fewer numerical sample points.

As discussed in Chap. 1, vortex lines are characterised both by their being nodal lines such that the value of the field is zero along their length, and by the complex phase winding around them some integer multiple of 2π times (in practice always once). It is possible in principle to use either of these properties to locate a vortex numerically, but in practice the changing phase is far more useful as in comparison to the intensity it can be more easily detected; the intensity may be near-zero across a wide area and can have a complex local structure, making gradient descent unreliable, but the phase change can be detected even far from the vortex as the total change remains the same about any path that encircles the vortex core, as long as no other vortices are involved. The detection of vortices is thus reduced to a phase unwinding problem, in which a closed path is sampled through the field via a set of sampling points, and the phase interpolated continuously between these samples. A vortex is present if the sum of phase differences between each consecutive pair of sampled lattice points is $2\pi m$, with any m in principle being resolvable with enough samples.

Figure 2.2 shows an example of this process using a circular path around a vortex in a Gaussian random field, with the path shown in (a) for the intensity and (b) for the phase of the field, and in (c) the integrated path beginning from the marked point and proceeding clockwise; we discuss the blue plot here, which uses 1000 sample points about the given contour, easily enough to observe variation at small distances. The phase along the path has a fairly complicated structure, changing sometimes extremely rapidly while in other places being relatively flat, but once it has been 'unwrapped' to remove the jump accounting for measurement modulo 2π the full phase change is equal to 2π and clearly detects the vortex. Although this example uses a circular sampling path about the vortex, any other shape would reveal the same result as long as only one vortex is present in the region it encloses (or multiple vortices with the same total phase sum). The vortex could in principle also be located by searching the intensity of the wavefunction for the nodal point at the vortex core, but this clearly cannot be so easily detected far from its real location—if we imagine for instance a gradient descent algorithm from some point on the contour, we might easily find only a local minimum or a different vortex line entirely.

Although robust, this initial method requires that very many sample points be selected in a path around the vortex, which is not practical for efficiently locating vortices whose position is otherwise unknown, such as in our random fields; it is hard to pick an appropriate integration contour without already knowing the position of a vortex. An ideal algorithm should instead locate vortices using the fewest possible sample points in the field. It is natural to base the procedure on a Cartesian grid, with the closed paths that may surround a vortex being the border of each square grid plaquette, and the sampled field values on this path being only the four grid points of the plaquette corners. The phase unwinding is reduced to just two rules; the phase change between each pair of adjacent sampling points is taken to be in the direction that minimises its absolute change modulo 2π, and a vortex is present if all four phase jumps have the same sign. This condition is not the most general, a vortex could be present if there are three large phase jumps in one direction even if the final

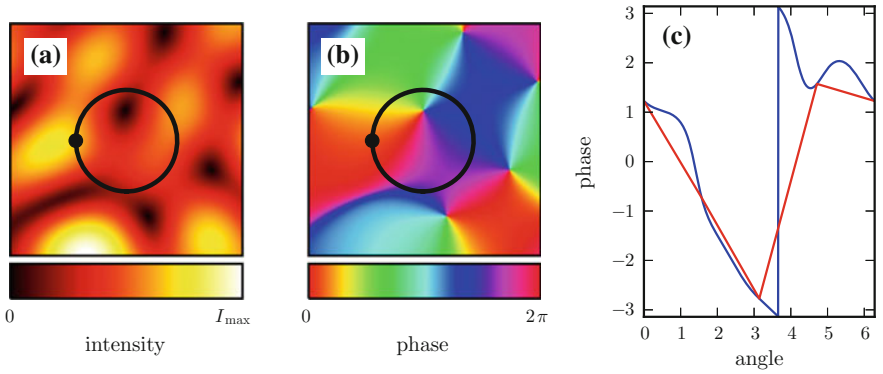

Fig. 2.2 Phase unwinding around a vortex. **a**, **b** Show the intensity and phase respectively of a $(1.27\ \lambda)^2$ area of a complex scalar Gaussian random wave with 12 component wavevectors, with a marked sampling contour proceeding clockwise from the marked point. **c** Shows the phase around this contour, sampled with 1000 points (*blue*) or just four points (*red*). The integrated change along both curves is 2π (though in local regions it may both increase or decrease) after unwrapping in both cases through the single large jump with magnitude greater than π that indicates a traversal of the phase periodicity from 2π to 0. The contour thus encircles a vortex. $I_{\max}$ is the maximum intensity within the visible region, not the entire field

jump is backwards, but this only occurs for an extremely anisotropic vortex and does not occur often. We discuss in the next Section how our own algorithm guarantees vortex continuity, which removes this omission as a concern.

Figure 2.2c shows in red the result of sampling the integration contour in (a) and (b) with only four (equidistant) points, linearly interpolating the phase between them. Although this discards almost all information about the local phase structure of the vortex, and despite the sample points being as far as 0.35 λ from the vortex core, applying the rules of the previous paragraph to these four points correctly detects that a vortex exists. It is no longer so visually clear where the phase jumps through 2π to 0, but the change is still detected following the rule that the total phase change (modulo 2π) is assumed to be less than π, and so the vortex is properly detected. This is only an example of how a very sparse sampling can detect vortices, we note that in plausible situations the sparse sampling would fail even though the dense sampling of 1000 points does not, and we discuss such situations further shortly.

In some models of vortex tangle, there is no more to the simulated field than this initial discretisation. For instance, in the $\mathbb{Z}_3$ model each vertex of the lattice is identified only by one of three values, which in this case we can consider as equidistant phases modulo 2π (for instance, 0, $2\pi/3$ and $4\pi/3$) [20], a notion related to the Potts model of statistical physics [21]. Assigning each vertex a random choice of these values creates a discrete phase field, around each plaquette of which we could search for vortices using the above algorithm based on just its four corners. In two dimensions this returns an unambiguous result (each plaquette simply does or

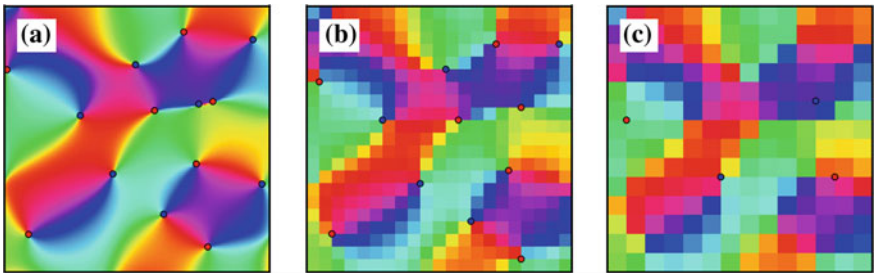

Fig. 2.3 Phase and vortex points of a two-dimensional random wave sampled at three different resolutions. **a–c** Show the same random wave sampled at Cartesian lattice spacings of 0.01 λ, 0.1 λ and 0.15 λ respectively, with the total grid side length 2.1 λ. Vortices are identified at each point via the phase change around the four vertices of a Cartesian sampling grid, with clockwise vortices coloured *blue* and anticlockwise vortices coloured *red*

does not contain a vortex), but we will return to this model later as an example of how further information about the vortices cannot be found in such a discrete model.

Returning to the question of how low resolution sampling of the field may introduce errors in the detected vortices, Fig. 2.3 shows an example of a function sampled on a Cartesian grid at resolutions of 0.01 λ, 0.1 λ and 0.15 λ. In all cases, vortices are detected solely from the phases of the four lattice points of each grid plaquette, as discussed above. At the highest resolution case, this procedure is extremely reliable and it is clear that every vortex has been detected correctly. This results from the smoothness of the field; when so far below the wavelength scale it cannot change too rapidly and four sample points are almost guaranteed to properly sample its path.

At a lattice spacing of 0.1 λ we begin to see how this algorithm is not always reliable, with the numerical search at this resolution shown in Fig. 2.3b but failing to detect precisely one vortex (ignoring the one example where vortex was closer to the edge than the lattice spacing). This is because two vortices approach closely, and the lower resolution is insufficient to distinguish them. Such an occurrence will sometimes lead to no vortex being detected as they have opposite signs and their phases cancel out, or in this case the phase anisotropy and sampling points are such that the double vortex appears instead as just one. It can also happen that the two vortices are both detected even though very close to one another, if the lattice sampling points happen to be placed correctly and the phase anisotropy is not too high, but there is no example of this in the Figure. Although the presence of the other vortices is detected, their apparent positions may be incorrect by up to approximately the lengthscale of the numerical lattice spacing; no smaller lengthscale is numerically sampled, so vortices can only be detected near to (but not exactly at) their real positions in the continuous field.

Decreasing the resolution slightly more to 0.15 λ lattice spacing leads to far more significant errors; almost all of the vortex points from the first two choices of resolution are no longer detected correctly, mostly being missing entirely. This appears to be because the scale has become large enough that the phase is commonly

anisotropic around the lattice plaquette, so it becomes increasingly a question of chance as to whether its four sampling points happen to have the correct position to see the presence of a vortex (though they are also now more likely to do so even if no vortex is present). Vortices could still be detected if the four lattice points were replaced with more samples in a path around the plaquette, the phase change still sums to 2π, but four points alone are no longer sufficient.

These overall constraints lead to our choosing a resolution around $0.1\,\lambda$ or smaller in our later numerical analysis of vortex lines in our random wave systems. The choice is somewhat heuristic rather than based on a specific property of the vortex recovery, but we are less concerned about missing vortices for reasons that will be explained later and so this basic resolution is sufficient. For similar reasons, similar Cartesian lattice methods are common throughout the literature [5, 10, 15, 17, 19].

The new algorithm presented in Sect. 2.2 resolves the detection failure problem through vortex continuity in the third dimension, but other modifications have been suggested in the literature. A first option is to use not a Cartesian grid but a triangular one, so that just three points encircle each singularity [22]; the local resolution is still low, but a regular triangle permits less ambiguity about the nature of the phase change, as it becomes totally unambiguous that all three triangle sides must have a phase changing in the same direction.

The quality of triangle-based sampling suggests another potential optimisation; since the phase of random waves varies only slowly below a certain scale, it is reasonable to *guess* that the point in the centre of each grid plaquette has a value that is the average of all the corner points. The plaquette can then be broken into four triangles, each individually searched for phase vortices. This method has the advantage that the spatial location of the vortex can be guessed on a scale lower than the sampling resolution, and in preliminary numerical experiments we have verified its stability. However, its accuracy is still not guaranteed and in practice our own three-dimensional method renders it unnecessary for fields where the full phase function is known, so we do not make use of it.

Using just four sampling points also means that vortices of higher orders, or two or more nearby vortices below the sampling distance, cannot be detected due to aliasing; the phase unwinding rules can only detect a 0 or $\pm 2\pi$ phase change. In random fields, nearby vortices are uncommon (the vortex-vortex correlation function goes to 0 at the origin [23]), and vortices of higher order generically do not occur in random waves.

These methods all relate to locating point vortices in two-dimensional phase fields, but our interest is in the three-dimensional tangle of one-dimensional vortex curves. The ubiquity of grid-based data and the robustness of phase unwrapping in 2D makes it desirable to extend these techniques; the 3D grid considered as an array of many cubic *grid cells*, with every face of every cell considered as a 2D phase unwrapping problem based on its four corners. A grid cell is marked as containing a vortex if a 2π phase change is detected around any of these faces, and since the vortex line must both enter and leave the cell, a local segment may be reconstructed by joining the two faces around which the phase circulation is found. Since the direction of

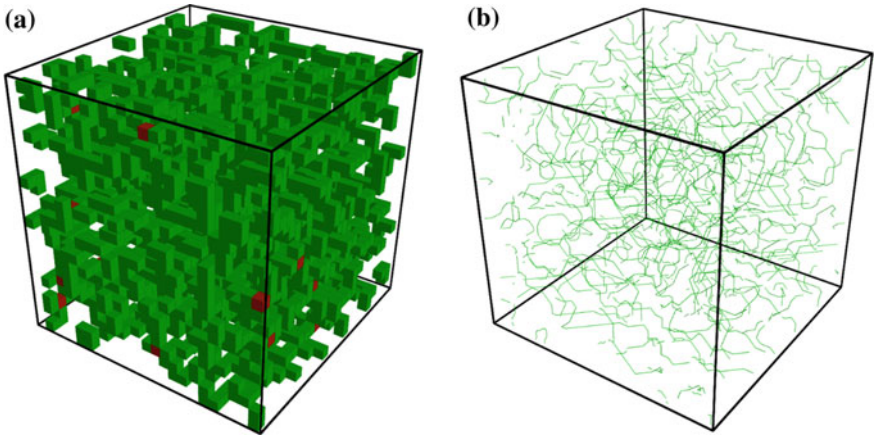

Fig. 2.4 Vortices in random wave fields, tracked in three dimensions through a basic phase unwinding method. In (**a**), Cartesian grid cells are marked in *green* if a vortex is detected through one or more of their sides, and in *red* if vortices are detected through three or more of their sides. In (**b**), the local line segments of vortices are approximated based on the cell sides through which they pass. The plotted cubic region has side length 4.33 λ

phase circulation must be consistent along the vortex, this orients the local segment according to the direction of phase change around these faces.

Figure 2.4 shows examples of this basic technique; Fig. 2.4a marks cells detecting a vortex through at least one face in green, and through at least three faces in red, denoting cells where the algorithm must be insufficient (other cells also incorrectly detect vortices, but not so obviously). Figure 2.4b shows the vortex segments from each cell, straight lines between the centres of the cell faces through which a vortex was detected. These cannot all be joined to continuous lines, as segments are missing where the basic algorithm failed to detect vortices. This can happen for all of the same reasons that apply to vortices in two dimensions (though the lengthscales are different between two-dimensional random waves and slices of three-dimensional volume tangle), but also for new reasons. For a vortex segment to be presented it must be detected not just once but twice, through both the two sides of the cell that it passes through. If it is not, we do not know its path. Vortices also cannot be resolved if more than one enters the same cell, in which case more than two cell sides have vortices detected and it is impossible to know which way the vortices are joined without more information. Slightly different mistakes arise if multiple vortices pass through the same face, in which case they will often not be detected properly (just as in two dimensions). This can even lead to previously-detected vortex paths being shown to be incorrect; even if two adjacent cells contain vortex segments that seem to be part of continuous chains, it is still possible that both vortices pass through the adjoining cell face and connect differently, but such that their passage is invisible in the discretised lattice.

This basic algorithm is the basis for three-dimensional vortex tracking in many of the systems mentioned above, where the vortices need only be approximately located [10, 19] or where experimental limitations do not permit a higher resolution [4]. However, the issues listed above mean it is not in general sufficient to precisely locate and distinguish vortices. This problem is well recognised and multiple solutions have been used; in a $2 + 1$D optical beam model, it was possible to 'guess' the correct local connectivity based on the smoothness of the field, a procedure applicable to our own work though such a statistical technique cannot always give the correct result [4, 5]. The issue also appears directly in the $\mathbb{Z}_3$ discrete phase lattice model introduced earlier, in which connectivity can only be resolved randomly if using a Cartesian lattice.

This ambiguity has also been resolved by using a tetrahedral grid [24] which directly clarifies vortex paths since tetrahedra with random phase vertices will never lead to multiple vortices passing through their volume, but applying this idea to continuous fields adds complexity and does not avoid problems when vortices approach closely or where numerical error means their passage through a grid cell wall is not detected. As with two dimensions, the use of tetrahedra may also be extended to split each cubic cell of a Cartesian grid, though rather than guessing the central phase (now an average of 8 corner points) it requires less interpolation to take advantage of the third dimension and split along four new planes of the cubic cell to give 4 individual tetrahedra in a sublattice of the Cartesian sampling; these choice of planes to split the cell is arbitrary, though a cell-by-cell comparison of each choice may help make the method robust. [25] similarly uses the centre point of the cell and of each of its faces to construct 24 different tetrahedra through whose sides the vortex is tracked with the same purpose. Again, these methods might improve the ability to distinguish and resolve vortices below the sampling resolution, but does not guarantee doing so in all otherwise-ambiguous cases.

This section has focused on locating vortices by a direct extension of two-dimensional phase unwrapping, but this is not the only method that has been employed in the literature. A somewhat different method focuses not directly on the phase, but on the *intersection* of two surfaces; the independent zero level surfaces of the real and imaginary components [2, 3] (discussed in Chap. 1. These are located by inspecting the corners of each grid cell in a Cartesian grid, as with the vortex phases above, but with a more involved scheme accounting for the triangulation of the zero surfaces rather than the location of one-dimensional line segments passing through the plane. Locating vortices reduces to testing the intersection of triangles in the surface reconstructions, a process which may be surprisingly efficient. Conveniently, there is no numerical ambiguity regarding vortex line closure, as the line is defined via isosurfaces whose triangulation is guaranteed to be continuous by the nature of the algorithms used—unlike the phase unwrapping above which may fail to detect a vortex under some conditions. However, this does not guarantee that the closure is physically correct if the resolution is low, or that all vortex loops are detected.

The main disadvantage of these techniques is that the surface triangulation is unnecessary to extract the location of vortex lines; if the same reconstruction can be

obtained via the phase alone, many of the same optimisations will be available, but the surface triangulation and intersection calculations may be omitted. For instance, our own algorithm in Sect. 2.2 could be modified to locate vortices via isosurface intersection, but we would anticipate identical results at the cost of a greater computation time.

A different method employing isosurfaces simply looks for contours of low intensity (i.e. for complex Ψ, the intensity $|\Psi|^2$). In principle, at some sufficiently low value these contours would only form tubes around vortex lines. In practice this performs very poorly for the same reason that the intensity is not useful for finding vortices in two dimensions; a very high resolution is necessary for the contour tube to closely surround the vortex, and if two vortices approach closely their contour tubes will merge.

A final technique from the literature is to try to avoid searching a full grid for vortices (in which many grid cells will be empty), instead locating an initial point on a vortex and iteratively marching along the curve [1, 3]. This has the advantage of efficiency, but the disadvantage that the location of vortices becomes statistical; it is impossible to guarantee finding at least one point on every vortex without searching almost all grid cells (a problem our method partly shares, but minimises by always searching the entire numerical grid for vortices). Additionally, walking along the vortex may be difficult or unstable, particularly where it approaches close to another (possibly unidentified) vortex curve. For this reason, we do not trace vortices this way; these techniques might be more suitable for approximately tracing curves such as in illustrating gradient flow [1], or if trying to follow a single vortex curve over a very large arclength rather than also tracking the whole local ensemble of curves.

2.2 Tracking Vortices with a Resampling Algorithm

None of the methods of Sect. 2.1 fully resolve the problem of three-dimensional vortex tracking, relying on assumptions and approximations that may not always hold, or which are insufficient in continuous fields. For these reasons, we introduce our own extension to this core algorithm, based instead on recursively *resampling* the field on smaller Cartesian grids of higher resolution wherever the vortex path is ambiguous. This allows us to locate and trace vortices efficiently with a relatively low initial sampling resolution but without sacrificing topological correctness, and returns piecewise linear representations of vortex curves with up to hundreds of vertices per wavelength. We will refer to this technique as the recursively resampled Cartesian grid method (RRCG). In this Section we give the details of the resampling algorithm, and of our geometrical optimisations to make precise vortex tracing possible.

The core new feature of the tracking algorithm implemented here is that a new, local sampling lattice is automatically generated in regions where the spatial details of the tangle are not resolved by the initial resolution of the sampling grid. This occurs mainly when two vortices enter the same grid cell, but also when an unusual vortex position and anisotropy means it is only detected to pass through one side of

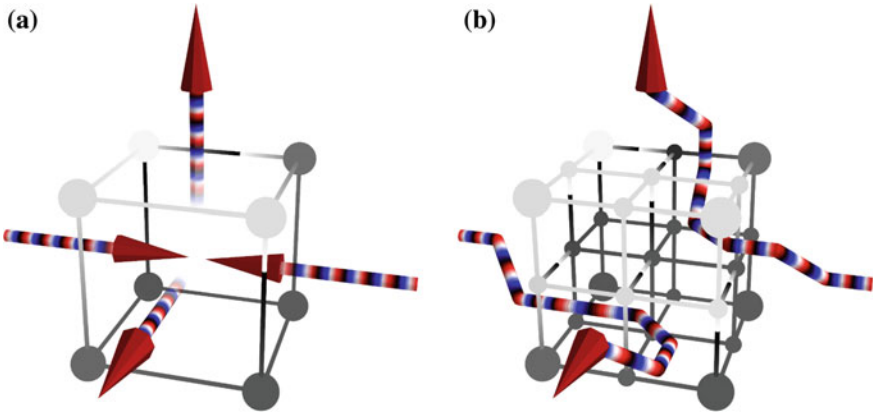

Fig. 2.5 Demonstration of the supersampling vortex tracking algorithm. Each sphere at a vertex of the sampling grid represents a phase voxel, with phase indicated by *greyscale*. Lines between the voxels are coloured following a linear interpolation of phase between them (following the shortest change mod 2π). The patterned lines represent vortex lines, the increase *red-white-blue* denoting the direction of the vortex line, from a right-handed increase of phase. These lines enter or leave the cell through faces around which the total phase change is 2π. This Figure originally appeared in [26]

the cell, while in reality we know that it must both enter and leave. In both cases, the topology within the grid cell is ambiguous, and cannot be perfectly resolved using any of the methods in the previous section.

Figure 2.5 shows an example grid cell where this resampling is necessary. In Fig. 2.5a, vortices are detected to pass through four of the cell faces, oriented by the direction of phase change (an arbitrary choice; we orient vortices such that the phase circulates clockwise around them). Without more information it is impossible to know which incoming and outgoing sides should be joined. In Fig. 2.5b, the 8 points of the local grid cell are replaced by a new $3 \times 3 \times 3$ grid; 8 new grid cells, each with side length one half of the original. Each cell of this new grid is examined for vortices in exactly the same way, this time fully distinguishing the paths of incoming and outgoing vortices as they do not approach closely on the scale of the new grid cells. The procedure is also *recursive*, if vortex topology is ambiguous on the scale of the new grid the resampling procedure is performed again, at an even higher local resolution, in the region of this specific smaller grid cell that still cannot be resolved.

To fully resolve all vortex filaments, we must both repeat this procedure and allow it to spread through the sampling lattice. The algorithm is as follows:

1. Check for vortex segments in all cells of the initial grid (i.e. looking for 2π phase changes around the border of each plaquette face). If there is no ambiguity in a grid cell, join the centres of the two faces where a phase change is detected with an oriented vortex line segment.
2. Resample all grid cells in which vortices are detected but the topology is ambiguous. The local grid cell sidelength of the resampling grid is changed to the

(initial sidelength)/N for integer N_{cell}, initially 2 but stored and incremented by one each time the procedure is run in a given cell.
3. If the local topology is still ambiguous, repeat step (2), further incrementing the local resolution.
4. Spread the new local resolution to all neighbouring cells bordering faces of the initial cell through which a vortex is detected. Each of these is also resampled to the same local resolution.
5. If step (4) changes the detected vortex configuration in any of the neighbouring cells, repeat (4) with neighbours of this new cell. Thus, the new resolution spreads until no ambiguity remains.
6. If the detected vortex configuration of *any* cell changed through steps (2–5), jump back to step (2).

In this fashion, volumes of higher resolution are generated where necessary, and spread through the initial grid until each ambiguity is resolved. Step (6) repeats the process, catching and correcting any regions through which a vortex was initially detected, but which resampling has revealed to be a numerical artefact rather than a real line segment. After following this procedure, every vortex is an unbroken chain of piecewise linear segments, forming either a closed loop within the simulated grid cell or terminating on its boundaries. Due to the recursive procedure, the topology of these curves can be correctly resolved even where they approach (in principle) arbitrarily closely, where multiple (even $\gg 2$) vortices enter the same cell, or even where multiple vortices pass through the same initial grid cell face and would normally be undetectable.

It is important to note that this algorithm does not just increase the resolution of the small number of cells where vortex detection errors can be seen, but also in all surrounding cells. This is vital because the initial vortex detection may simply be wrong, with particular phase anisotropy patterns and unfortunate grid spacing leading to the apparent presence of a vortex in the initial lattice even though a vortex does not pass through this region the continuous field being sampled. Allowing the resampled region to spread entirely resolves this problem, as it means that once a cell has been resampled (and its detected vortex contents possibly changed) its neighbours will always be checked for consistency with this new result, and in turn resampled themselves until the detected vortices are consistent everywhere. This also resolves most of the issues explained in the previous Section when two vortices are connected in an unseen way via the passage of both through a single grid plaquette. In practice this phenomenon tends to lead to high phase anisotropy and numerical failures that force a resampling, which in turn detects the true local geometry and topology. We are also able to optionally force this procedure every time different vortices occupy adjacent cells, but find that doing so almost never reveals new problems. Alternatively, the opposite may happen with the resampling grid containing new numerical errors, but in this case the resampling is simply repeated at a higher resolution and the local smoothness of the field means that all ambiguities are eventually resolved.

As a benchmark of standard performance under real use, several of our core datasets with 3-torus eigenfunctions use an initial grid cell sidelength of 0.087 λ;

at this resolution, around 4 % of grid cells contain vortices of which roughly 5 % must be re-resolved at least once and approximately 0.1 % re-resolved at five or more times the initial resolution. Sampling the initial grid at even twice the initial resolution of the original grid would have an 8-fold impact on calculation time, so the resampling procedure is essential to efficiently resolve the local topology. Full details of the different parameters used and numerical statistics are given in Sect. 2.5.

These options are also visualised in Fig. 2.6; (a) shows resampled regions in the same example cell as Fig. 2.4, with boxes showing regions resolved at two to seven times the initial resolution, the ratio denoted by the intensity of the colour. Only a small number of these seeded the resampling, and most of the cases of several adjacent resampled cells are due to the resampled region spreading by necessity as it reveals errors in adjacent cells and is necessary to ensure connectivity. This gives some insight into exactly how important the resampling procedure is to reconstructing the tangle. Figure 2.6b shows the same vortices but without the resampling indicator boxes; the steps of our algorithm guarantee that these vortices are now all continuous, unbroken space curves terminating on the boundaries of the volume.

Although this method topologically resolves vortex lines very precisely on the sub-wavelength scale, the individual linear segments do not accurately reproduce the spatial conformation of the vortex filament, as it is still assumed that each vortex line segment passes through the centre of the face where it are detected. Thus, even if all the sampled vertices are close to the true curve, the local tangents cannot be accurate. We add three further optional steps to improve the local fit to the curve, which are applied mainly in Chap. 3 where accurately resolving local geometry is most important.

First, once the vortex lines have been located, the resampling procedure is applied again to all the cells through which the vortex line passes, even if the topology is already well distinguished. This accurately traces the shape of the vortex curves without expending further numerical resources on the voids between. In principle, extremely small vortex loops below the sampling resolution may not be detected, but these are extremely rare and contribute a negligible amount to the total vortex arclength (Sect. 3.6). Figure 2.6d illustrates the result of this procedure applied to our previous example function, with the initial result of our algorithm in (b). Although the vortices in both cases have the same overall topology, the enforced resampling of the entire vortices in (d) can be seen to recover much more of their precise geometry. Applying such a locally high resolution to the initial grid would also track the vortices well even without the RRCG procedure, illustrated in Fig. 2.6c which shows the grid cubes in which vortices were detected even without the RRCG algorithm after searching the same function as in (a) but with four times its initial resolution applied uniformly across the lattice. Even viewing only the cells containing vortices it is now possible to follow the paths of individual vortex lines, but this calculation takes significantly longer than the RRCG resampling. Using this method to sample high energy eigenfunctions with an even higher resolution would quickly extend the computation time to days or weeks. Our algorithm focused on the vortices takes only hours at most (or more normally minutes, details in Sect. 2.5) and allows a local resolution along vortices of several hundred sample points per wavelength.

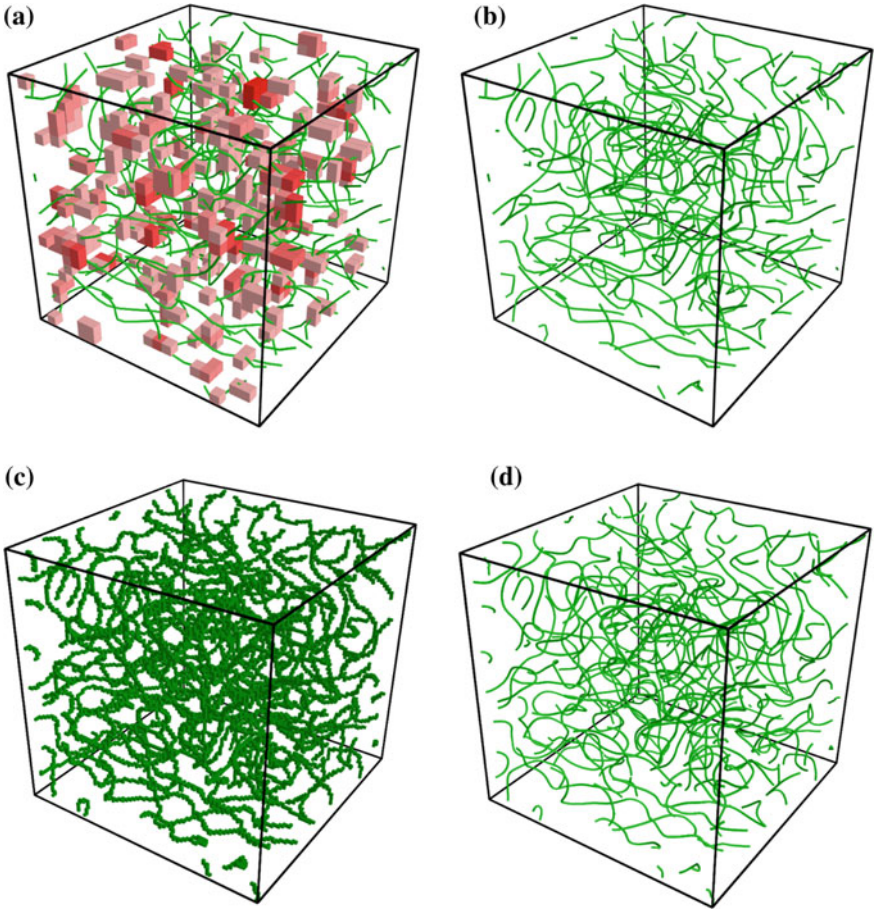

Fig. 2.6 Vortices detected in a 3-torus eigenfunction, showing the results of different algorithmic optimisations. The torus eigenfunction has energy 75, sampled on a grid of size 25^3 in (**a**), (**b**) and (**d**), an initial resolution of 0.1 λ. In (**a**), the *green lines* denote recovered vortex segments and the *red boxes* show the degree of local resampling—up to 4 times the initial resolution according to the intensity of the colour. **b** Shows the same vortex cell, but displaying only the vortices. **c** Shows the same vortex cell sampled at four times the original resolution, a lattice of size 80^3, showing only cells detected to contain one or more vortex segments. **d** Shows vortices detected with the lower initial resolution, but with an enforced resampling along the entire length of vortex curve to a local grid size of 0.025 λ, equivalent to (**c**) but applied only to vortex-containing cells rather than the entire volume. Of these cells, the method in (**d**) is used mainly in Chap. 3, while the simpler method of (**b**) is sufficient in Chap. 5. These cells all have side length 4.33 λ

Second, rather than assuming the vortex line penetrates the centre of the grid face through which it is detected, we instead treat the face plane as a new, two-dimensional vortex location problem. The intersection point is located by resampling the face with a 2D square grid of some multiple of the local 3D resampling resolution, and locating the square plaquette of this new grid around which there is still a 2π phase change. The centres of these new plaquettes in the 2D face are joined by straight lines in the 3D grid to approximate the real vortex curve. This optimisation is illustrated in Fig. 2.7, in which the initial four phase points of (a) reveal only that a vortex probably passes between them, while the 49 times higher resolution in (b) isolates the vortex well away from the point where the phases meet. Although this higher resolution is only necessary for the recovery of local geometrical quantities, the numerical cost of the face 2D resampling is relatively low (clearly lower than that of resampling an entire grid cell by a factor of the cube of the resolution multiplier), so we apply it frequently in datasets where only topology is relevant.

Finally, even with both of these optimisations the recovered curve is noisy on some small scale, thanks to remnants of the original lattice and the discretisation of the tangent vector. This poses problems for geometrical recovery as quantities such as the torsion (Sect. 3.3) are extremely sensitive to even small distortions. This remaining noise can be removed through smoothing and downsampling the recovered curve; since the recovered vertices cluster around the true curve, it will be better reconstructed by an appropriate *average* of their positions.

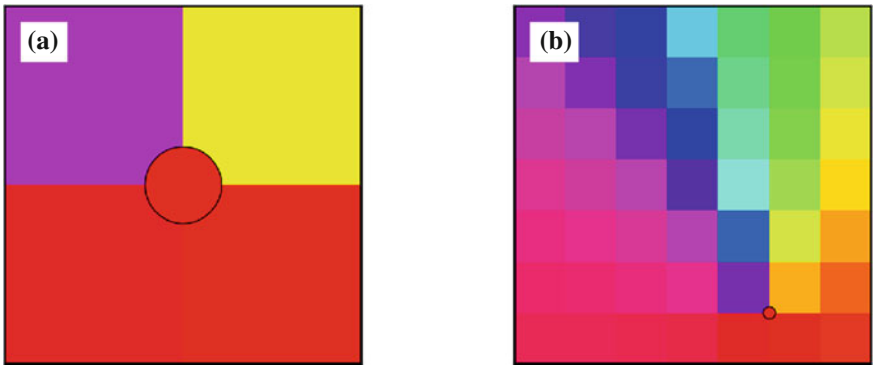

Fig. 2.7 Resampling of a face of a grid cell, gaining improved geometric information about the vortex position. In (**a**), the sampling resolution is 0.1 λ and there is only enough information to show that a vortex is probably present. In (**b**) the resolution is seven times higher (typical in our simulations), and the vortex can be located more precisely in the plane. The *red dot* denotes the (clockwise) vortex, numerically located via the phase changes between the adjoining four samples

We smooth the curves by keeping only every nth vertex for some integer n, and fitting with a piecewise polynomial spline curve interpolation. A forced perfect fit would introduce high frequency noise to the polynomial thanks to remnants of the original lattice, so we require only that the spline curve fit closely approximates the original vertices. We enforce that the sum of square distances between original and interpolated points does not exceed some small value; $\sum_v (\boldsymbol{r}_v - \boldsymbol{r}_\mathrm{i} i) < \delta L$, where $\boldsymbol{r}_\mathrm{v}$ represents the vertices of the original piecewise linear curve, $\boldsymbol{r}_\mathrm{i}$ is the interpolated spline curve at each of these points, δ is a fitting parameter, L is total length of the current vortex curve and the sum is over the vertices. This procedure resembles that of [27] in tracing superfluid vortices, though here we are concerned with accurately recovering the small scale geometry rather than removing noise on larger scales. We also optionally perform a direct smoothing by of the curve by convolving by a weighting distribution, though we find that this is largely redundant if combined with the polynomial spline fitting.

Figure 2.8 compares different choices of downsampling and δ parameters, along with their direct effect on the position of the peak of the curvature distribution for a larger line ensemble with these parameters applied (see Sect. 3.3, but used here only to test stability). If the polynomial fit is forced to pass perfectly through every point or no downsampling is performed, the peak is significantly higher (and the distribution would be highly distorted), but as both types of smoothing are increased the peak settles to approximately 0.15 ± 0.05. For this reason, we use $\delta = 0.004$, but the choice is somewhat arbitrary and we accept an inherent error on the order of 3 % of values recovered this way. Likewise, we downsample curves to every fifth

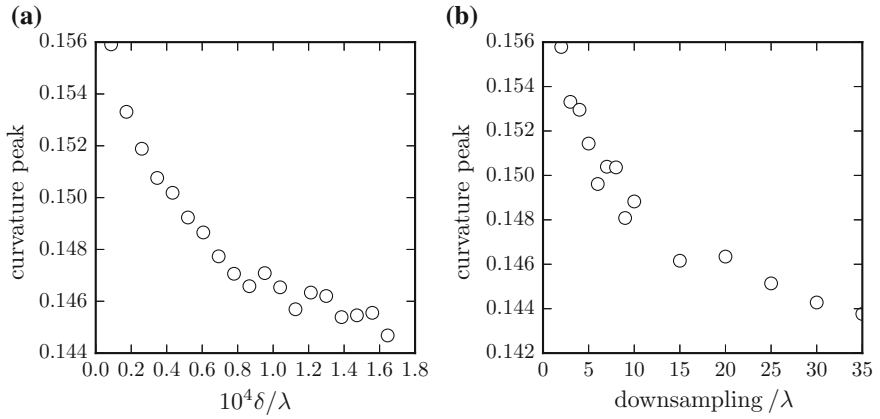

Fig. 2.8 The effect of different smoothing parameters on recovered vortex geometry. The vertical axis in both (**a**) and (**b**) shows the maximal value of the curvature PDF of the vortex curves (described fully in Sect. 3.3), i.e. the probability of the most probable curvature, an example of a geometrical parameter whose value depends on the details of smoothing. **a** Shows how this is affected by the δ parameter in our polynomial fit, and **b** the downsampling of the initial vortex curve

point before applying the polynomial fit. Although both patterns appear to be settling to a lower value of curvature, in fact this is due to the increased smoothing further distorting the curve.

2.3 Vortex Tracking in Different Systems

The algorithms described in Sects. 2.1 and 2.2 are easily adapted to vortex tracking in different systems by choosing parameters such that the area of interest is covered by a Cartesian grid. We cover here the details of doing so in the systems covered by this thesis; the 3-torus (periodic boundary conditions), the 3-sphere, and the quantum harmonic oscillator, as in Sect. 1.7.

For two of these systems this extension is trivial; with the quantum harmonic oscillator, the vortex tangle covers some statistically small region of real space, and the grid must only cover this. In principle, 'hairpin' vortices may extend arbitrarily far outside the classical radius where the potential is equal to the energy, but these are rare events that do not significantly affect topological statistics (Sect. 5.1), and the grid may easily be large enough to cover the interesting region. Our practical choices of parameters are discussed in Sect. 2.5.

When the eigenfunctions being studied are those of the 3-torus and so have Cartesian periodicity, the grid must itself must be made periodic, with plaquettes on one side identified with plaquettes on the opposite side, and lines joined continuously through these boundaries. We extend the algorithms above by allowing the 'spreading' stage to resample cells through periodic walls, ensuring a perfect vortex recovery. With this adjustment, the full extent of the manifold is appropriately sampled contingent only on an appropriate resolution according to the wavelength; our choices of parameters are again discussed in Sect. 2.5. Although lines may pass through the periodic sides many times, the full curve may be recovered by simply 'unwrapping' these passages, as in Fig. 2.9; each arc of the curve, terminating on the boundary of the simulation cell, is shifted by an appropriate multiple of the periodicity to form a single continuous curve. The closed loops may wrap a non-zero number of times around the torus and so do not form closed loops following this procedure, but this poses no numerical problem as long as the jump between their start and end points is ignored in geometrical measures. Such lines are topologically more complex discussed in Sect. 5.6.

Our third system is the degenerate eigenfunctions of the 3-sphere, S^3. These demand a further extension to the algorithm, as the 3-sphere cannot be tiled by regular quadrilaterals of arbitrary size. Other regular tilings are available, but are undesirable since (as discussed in Sect. 2.1) they add practical difficulty. It is more desirable to adapt our Cartesian grid method.

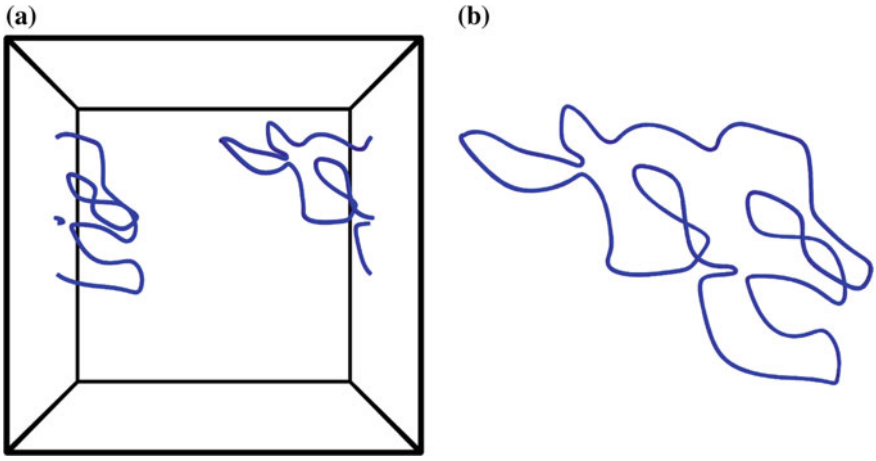

Fig. 2.9 A short vortex loop passing through the 3-torus boundaries in a simulated cell. **a** Shows its initial representation, with multiple components cut along the simulation boundary, while **b** shows the result of 'unwrapping' the line by shifting components by multiples of the periodicity in order to recover a trivial loop

We simulate the 3-sphere via the eight *octants* of the 3-cube (the *hypercube*), the shape in $\mathbb{R}^4$ with edges of equal lengths along each of four orthonormal directions. Its eight octants are the analogue of the six faces of a normal cube in $\mathbb{R}^3$. We cover each octant with a Cartesian grid on which the metric is not uniform. This is most clearly visualised by the analogous construction in two dimensions; consider a cube lying within a sphere, with both having their centre of mass at the origin. Any point of the cube may be projected to a unique point on the sphere along the straight line that begins at the origin and passes through that point. Each face of the cube can be assigned a natural Cartesian grid across its area, and any process on the surface of the sphere modelled by performing calculations on this grid while retrieving distances from the projection to the sphere. Although the metric is not uniform across each face, every face is identical, and the geometry of the sphere may be recovered by joining faces along appropriate edges. In this respect, the unfolded faces form a *net* of the 2-sphere.

In our case of the 3-sphere, the process is identical, but with the projection now via a straight line from the origin of $\mathbb{R}^4$ and the six 2D faces of the cube replaced by three-dimensional octants of the hypercube. Other properties remain the same; geometrical positions in each octant are identified with coordinates on S^3 by projection along a straight line passing through the origin in $\mathbb{R}^4$, and the geometrical connectivity of the 3-sphere is recovered by joining appropriate faces of each octant lattice.

Following the discussion in Sect. 1.7.2, the 3-sphere manifold satisfies

$$x^2 + y^2 + z^2 + t^2 = r^2. \tag{2.1}$$

and may be understood via three angular coordinates ψ, θ, ϕ ranging from $0 \to \pi$, $0 \to \pi$ and $0 \to 2\pi$ respectively. The projection to the 3-sphere from the hypercube requires inverting (1.20)–(1.23) to give

$$\psi = \arccos\left(\frac{t}{r}\right), \tag{2.2}$$

$$\theta = \arccos\left(\frac{z}{r\sin\psi}\right) = \arccos\left(\frac{z}{\sqrt{r^2 - t^2}}\right), \tag{2.3}$$

$$\phi = \arctan\left(\frac{y}{x}\right), \tag{2.4}$$

which project any point in $\mathbb{R}^4$ radially onto the 3-sphere of radius r.

By analogy with the 2-spherical example above, we simulate the 3-sphere by running our numerical algorithm from the previous sections once for each of the octants of the hypercube, covering each octant with a Cartesian lattice whose points are mapped to the hyperspherical angles via (2.2)–(2.4), allowing vortices to be traced without further modification of our numerical algorithm. Each face of each octant is associated with a face of another, which corresponds to the same plane on the 3-sphere. Vortices within each individual octant may enter or leave through one of its faces, but the full continuous vortex line is traced by joining it with the equivalent line leaving or entering the associated face of an 'adjacent' octant.

The main advantage of this decomposition is that octants are locally quite flat, the metric does not vary significantly across their volume, so our algorithm's assumptions about vortex density still hold to a good approximation. Other possible choices of simulation method do not perform so well; for instance, we could impose a Cartesian lattice on the stereographic projection of the 3-sphere and track vortices via the inverse map from stereographic coordinates to hyperspherical angles, but stereographic projection significantly distorts lengths and so vortex density varies significantly in the projected space. This latter choice makes the RRCG algorithm much less efficient, as it is difficult to concentrate numerical effort appropriately when large areas of the Cartesian grid correspond only to small hyperspherical volumes. Although this relatively low metric is sufficient for the RRCG algorithm to perform well, further corrections are necessary to recover local geometry of the resulting curves, discussed in Sect. 2.4.2.

Figure 2.10 demonstrates vortices traced numerically in each of our eight octants. The octants are placed adjacent to one another along identified faces where possible, but are not distorted from the Cartesian lattice recovered from the simulation. Since they represent a net of the 3-sphere, they could be folded together in $\mathbb{R}^4$ to construct the hypercube. The effect of the minimally distorted metric can be seen in the visually consistent vortex density within each octant; lengths of vortices measured in this discontinuous projection will correspond approximately (but not precisely) to distances on the 3-sphere, discussed further below.

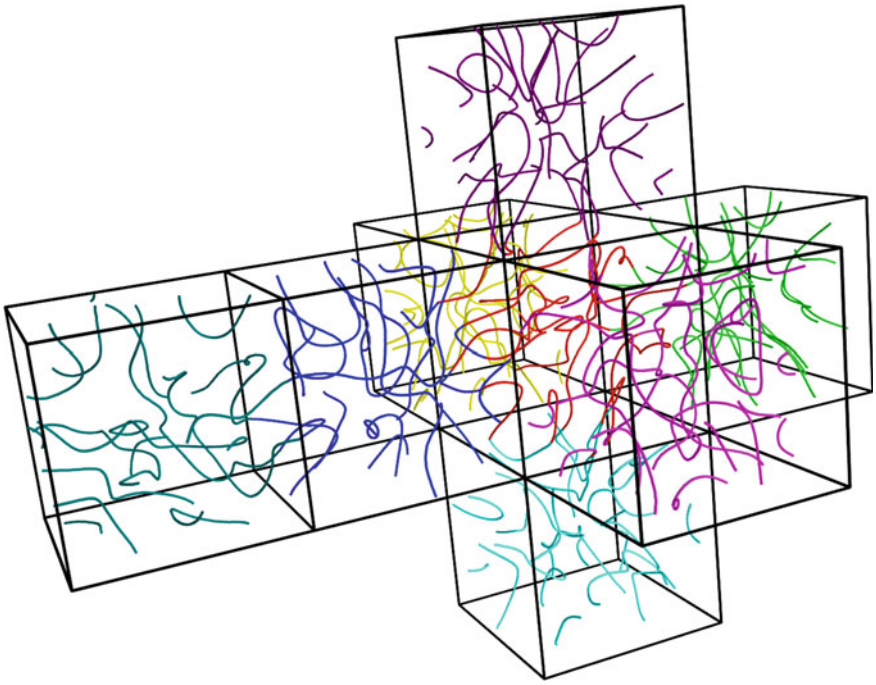

Fig. 2.10 Vortices traced in a hyperspherical eigenfunction via our octant method. The eigenfunction is a random hyperspherical harmonic of energy 80. Vortices in each octant are given a different colour, and octants positioned so that (where possible) moving through an octant wall maps to a continuous path in hyperspherical angles

Figure 2.11 shows the same vortices, coloured in the same way, but represented by stereographic projection. Most octants are now much more distorted, particularly the outermost in which the central point of the octant maps to the point at infinity; vortices passing close to this point may travel very far from the origin the projection. Despite this distortion, all vortices are now continuous, closed loops.

Although we have stressed the relative flatness of each octant, it is important to note that the metric does vary across their extent, and arclength as seen in Fig. 2.10 does not directly correspond to arclength as calculated properly in hyperspherical coordinates. Figure 2.12 demonstrates the ratio of metric distance to Cartesian distance (without metric correction) across the volume of each octant, via two planar slices; in the centre of the octant the hyperspherical distance is twice the local Cartesian distance, while at the edges this factor is reduced to approximately 0.88. This anisotropic distortion is not strongly visible in Fig. 2.10, but must be taken into account when calculating the true length. The details of this calculation are given in Sect. 2.4.

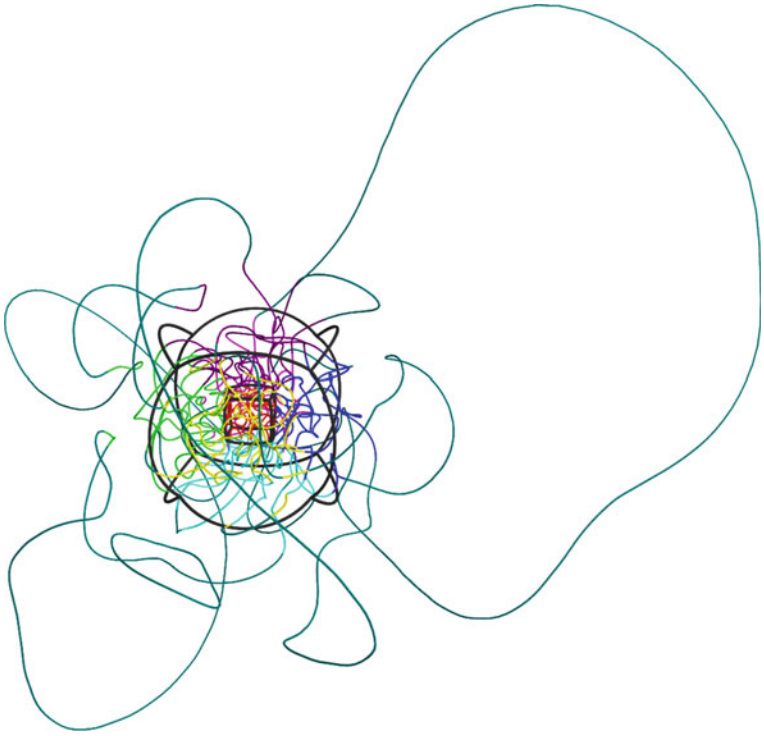

Fig. 2.11 Vortices traced in a hyperspherical eigenfunction via the octant method, projected stereographically. The eigenfunction is the same random eigenfunction as in Fig. 2.10, and vortices in each octant coloured identically, but now projected stereographically in their hyperspherical coordinates such that all vortices are continuous. The black lines show the azimuthal octant cell boundaries, also stereographically projected via the 3-sphere

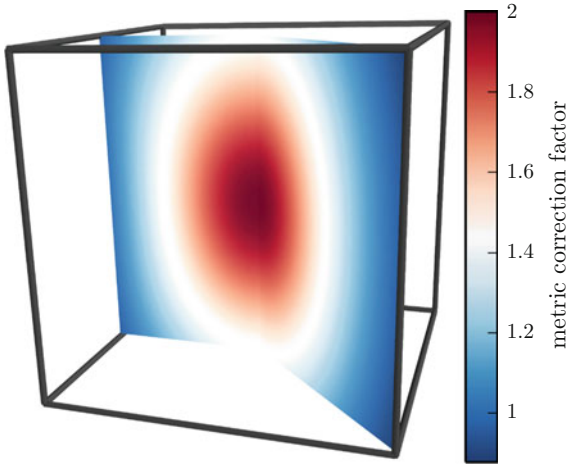

Fig. 2.12 Distance on 3-sphere as fraction of local distance on the Cartesian grid of a numerical octant. The distance is shown through two planes within the octant, one halfway through one axis, and the other following a diagonal. The distance factor ranges from minimum 0.8776 in the corners to maximum 2.0 in the centre

2.4 Geometrical Details in Different Systems

Performing different kinds of geometrical and topological analysis necessitates a consistent notion of lengths. We give here the details of calculating these quantities for vortices in the context of each of our three systems, accounting for specific properties including their periodicity, curved metric and varying background potential.

2.4.1 The 3-Torus

The 3-torus is flat (it admits trivial metric) and includes no potential, and so angles between vectors may be calculated just as in $\mathbb{R}^3$. The periodicity of the manifold does impose an extra condition on shortest distances between points; distances may not always be calculated directly between the two points, but (if relating to a larger section of vortex curve) must correctly take account of the periodic boundary conditions.

In practice, it is almost-always sufficient to 'unwrap' a vortex curve through the periodic walls, shifting each segment within the periodic cell by a multiple of the periodic side length corresponding to the number of times the boundaries have been traversed since an arbitrarily chosen starting point along the line; it is by this procedure that the vortex segments within a single cell of the 3-torus may be reconstructed to a continuous line as shown in Sect. 1.7.1. For loops that close with trivial homology, this map to $\mathbb{R}^3$ loses no information and lets the curves be analysed as normal. For loops that wrap some non-trivial number of times around the boundaries, the unwrapped curve will not close but instead remain open by some (potentially large) multiple of the cell side length according to its homology vector. The loop is still periodic across this gap, but this poses no practical problem as long as analysis does not treat this as a geometrical jump but only a representation of periodicity.

2.4.2 The 3-Sphere

The 3-sphere is unique among the systems studied here in exhibiting inherent background curvature; any calculation of distances or angles must take account of the path taken around the manifold.

The round metric on the 3-sphere in hyperspherical angular coordinates is given by

$$ds^2 = dx^2 + dy^2 + dz^2 = r^2 \left(d\psi^2 + \sin^2 \psi \left(d\theta^2 + \sin^2 \theta d\phi^2 \right) \right), \tag{2.5}$$

and distances along a given path $\Phi(s)$ may be obtained by integrating along its length with this metric. For a vortex curve L, the total length is the integral $\int_L ds$

In practice, the calculation is made easier by considering vortex curves as *piecewise-geodesic* on S^3, the generalisation of piecewise linear curves to arbitrary manifolds. Such a parameterisation is a natural result of the numerical algorithms in the previous sections, and the vortices of the sphere are specified in terms of a series of hyperspherical angles joined by minimal arcs. These minimal arcs (which are the geodesics between the points) are segments of *great circles* of the 3-sphere, defined as intersections of S^3 with a 2-plane passing through the origin in $\mathbb{R}^4$ [28]. Any two points of the 3-sphere sit on precisely one such great circle (unless they are antipodal in which case they sit on an infinite number of equivalent great circles), and they divide the circle into two segments, one of which is guaranteed to be the shortest distance between the points along the spherical metric.

Figure 2.13 demonstrates three such great circle segments on the 2-sphere (this could be a cross-sectional 2-sphere of the 3-sphere). The distance between each pair of points reduces to

$$d = r\kappa, \tag{2.6}$$

$$= r \arccos(\boldsymbol{v}_1 \cdot \boldsymbol{v}_2), \tag{2.7}$$

where r is the radius of the sphere and κ is the angle between the positions of the points in $\mathbb{R}^{n+1}$, given by the normalised vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$; in the case of the 3-sphere, these vectors are their positions in $\mathbb{R}^4$ as given by (1.20)–(1.23). Obtaining the lengths of vortices reduces to the sum of these geodesic distances along the vortex curve.

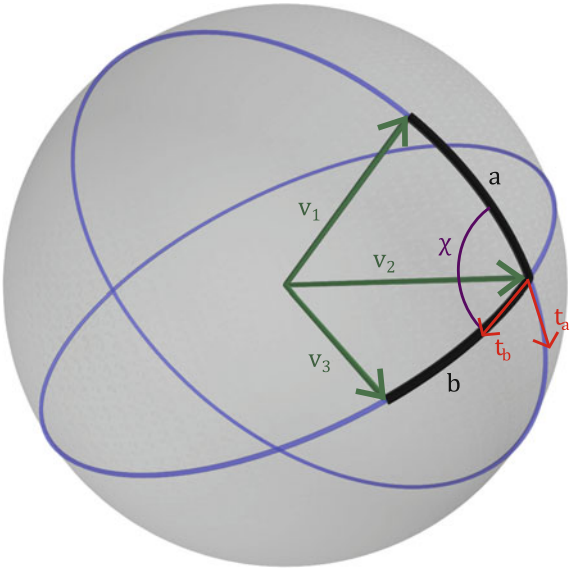

Fig. 2.13 Distances and angles between arcs on the 2-sphere. The 3-vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ denote points on the sphere, joined by geodesic arcs a and b which are segments of the great circles marked in blue. The length of each arc is given by (2.6), and the angle χ between the arcs is given by $\boldsymbol{t}_a \cdot \boldsymbol{t}_b$ in (2.8)

The calculation of angles must be treated similarly; since each geodesic segment is not simply a vector but an arc, the angle between each pair of segments is given by the angle between the arcs at the point of their intersection. This is illustrated at the intersection of arcs a and b in Fig. 2.13, they are separated by the angle

$$\chi = \arccos(\boldsymbol{t}_a \cdot \boldsymbol{t}_b) = \arccos\left(\frac{\boldsymbol{v}_2 - \boldsymbol{v}_1(\boldsymbol{v}_1 \cdot \boldsymbol{v}_2)}{|\boldsymbol{v}_2 - \boldsymbol{v}_1(\boldsymbol{v}_1 \cdot \boldsymbol{v}_2)|} \cdot \frac{\boldsymbol{v}_3 - \boldsymbol{v}_2(\boldsymbol{v}_2 \cdot \boldsymbol{v}_3)}{|\boldsymbol{v}_3 - \boldsymbol{v}_2(\boldsymbol{v}_2 \cdot \boldsymbol{v}_3)|}\right). \tag{2.8}$$

2.4.3 The Quantum Harmonic Oscillator

The quantum harmonic oscillator is outwardly most similar to the 3-torus; the eigenfunctions live in $\mathbb{R}^3$ without projection. However, the notion of distance still depends on the radius due to the changing potential in (1.32), which may be rewritten as

$$\nabla^2\Psi = (E - \frac{1}{2}(x^2 + y^2 + z^2))\Psi, \tag{2.9}$$

$$= k_{\text{eff}}^2\Psi, \tag{2.10}$$

where k_{eff} is an effective local wavevector, with an effective local wavelength $\lambda_{\text{eff}} = 2\pi/k_{\text{eff}}$. In this form the equation matches the random wave model of (1.6), but with a position-dependent wavelength that tends to infinity as the radius nears the boundary where $E = 1/2(x^2 + y^2 + z^2)$. Distances along vortex curves in eigenfunctions of the QHO may be calculated by integrating along the curve including a factor of this local wavelength.

This description only applies within the radius where the energy exceeds the potential. Outside this region, λ_{eff} becomes imaginary and is no longer a good descriptor of the local distance. When measuring the length of vortex curves we simply exclude these regions as not contributing to the geometry of the tangle, as vortices are confined to very simple behaviour that does not reflect the random wave model, being able only to travel straight to infinity (Sect. 1.7.3), or doing nothing more complex than reversing direction and re-entering the main tangle in what we will call hairpins.

Measuring arclength in this way is only an approximate descriptor of vortex arclength, taking account of the varying lengthscale of correlation in regions of different potential. However, it is not a complete description as it does not account for the anisotropic directional correlations near the classical radius. It is this phenomenon that encourages hairpins, with vortices rapidly changing direction despite the low local wavelength. A full parameterisation of arclength would need to instead take account of a local metric controlling this difference, but we do not do so here.

2.5 Summary of Numerical Results

We give here the details of all our core numerical datasets whose analysis comprises the rest of this research, including details of input parameters, examples of results, information on the practical efficiency of our algorithms, and basic statistics of the output vortices.

All statistics of timing refer to implementations of our algorithms via Python and Cython, running on a typical laptop machine at the time of writing (Intel Core i5-3320M CPU). It is likely that significant speed improvements could be made to the core algorithm, but our implementations are optimised only so far as necessary to retrieve appropriate results.

2.5.1 The 3-Torus

Our core datasets on the 3-torus have energies 75, 243, 675 and 1875 in (1.11). The numbers of simulated cells in each case, along with basic statistics, are given in Table 2.1.

The initial grid spacing in all cases is 0.0173 λ. Local regions are resampled at a higher resolution following Sect. 2.2. In the case of energy 75, an extra resampling stage is performed along the vortex lengths, forcing a local grid spacing of no greater than 0.00247 λ. In general, around 0.19 % of cells in the initial sampling grid contain a vortex segment of which about 0.15 % of these cases must be re-resolved. After this resampling and any further smoothing as in Sect. 2.2, each line is resolved with at least 50 sample points per wavelength, or at least 140 sample points per wavelength in the data with energy 75.

All other energies use a lower initial resolution, leading to around 3.9 % of grid cells containing vortices, with a further 4.6 % of these being re-resolved to guarantee

Table 2.1 Details of the simulations we have carried out on eigenfunctions of the 3-torus. The initial and resampled resolutions are only the minimal resolution per grid cell, and the local resolution may be higher if required by the algorithm. Errors given are standard deviations, reflecting the overall distribution of line lengths

Energy E	Number of cells simulated	Approx time per cell	Number of loops		Number of NTH lines	
			Total	Average	Total	Average
75	2628	2 h	39395	14.5 ± 11.1	6915	2.55 ± 1.54
243	19845	80 s	1211553	61.1 ± 27.4	66633	3.36 ± 1.56
675	9514	6 min	2580009	271.2 ± 83.3		
1875	51	20 min	54200	1062.7 ± 112.9	191	3.75 ± 0.83

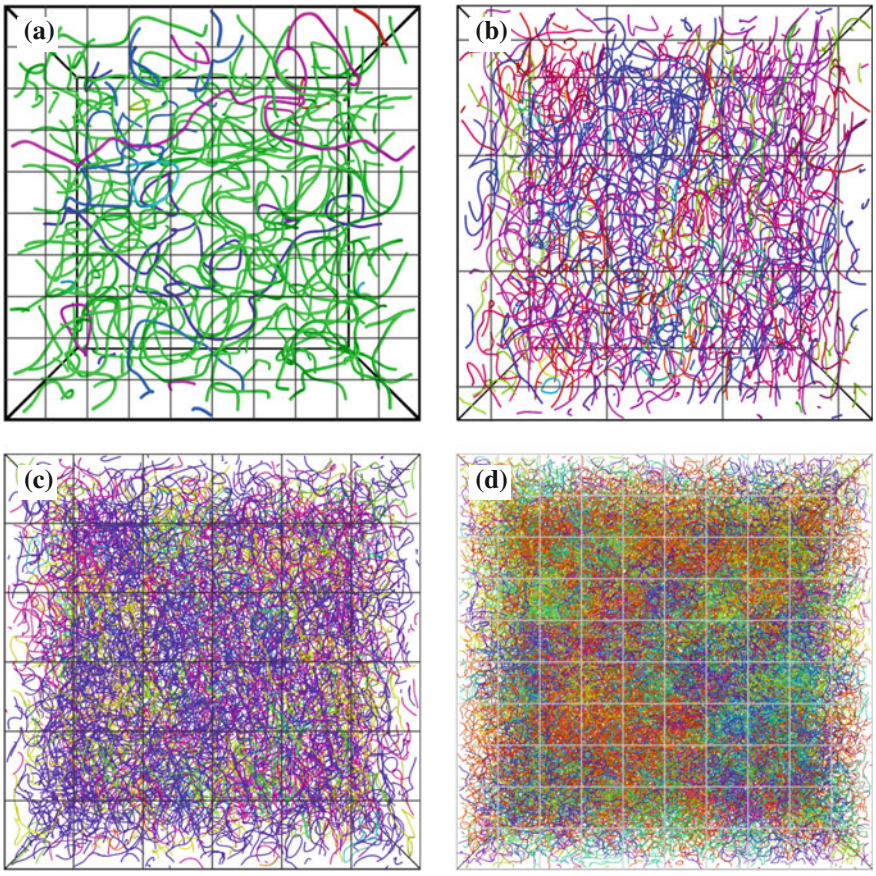

Fig. 2.14 Numerically tracked vortices in simulated cells of the 3-torus. **a**–**d** correspond to energies 75, 243, 675 and 1875 respectively. In each figure, every vortex is coloured by a different (random) hue. The grid on each image shows a spacing $25\times$ the grid spacing of the initial numerical sampling grid

local continuity; the lower initial resolution successfully shifts computational effort to the resampling procedure, rather than on closely examining empty regions of the grid.

Figure 2.14 demonstrates vortices in example simulation cells at each energy.

As a test of the recovered geometrical statistics at energies 243, 675 and 1875 in which the numerical resolution is relatively low, we have also simulated twice several eigenfunctions of energy 243. The first simulation was performed as above, and the second with the origin of the numerical grid shifted by $(\sqrt{2}, \sqrt{2}/2, 2\sqrt{2})$. The choice of perturbation is arbitrary, but intended to shift each axis by a non-trivial factor of the grid spacing. The recovered vortex curves should be topologically the same in both simulations, but the offset of the origin will expose geometrical

differences as the RRCG algorithm will not necessarily resample the same local regions of the vortices. The difference in the total recovered arclength (measured over the numerically recovered piecewise-linear vortex curves) will give an estimate of the error on the numerical recovery.

Over 49 eigenfunctions of energy 243, the average total arclength recovered was 2220 λ, and the average difference between the results of simulations on the same eigenfunction just 2.4 λ with a standard deviation of 1.8 λ; both on the order of just 0.1 % of the total arclength. The maximum difference found was 7.5 λ, and the minimum 0.009 λ. We conclude that the RRCG procedure gives consistent results even at the numerical resolutions we make use of, and although we must account for these errors in any later measurements of length, they are not likely to be significant. These numbers should also generalise without modification to our simulations at energies 675 and 1875, as although the total vortex arclength in a single cell will change, the fractional error of the RRCG algorithm will not.

2.5.2 The 3-Sphere

Our core datasets of the 3-sphere have energies 120 and 255 in (1.24). The numbers of simulated cells in each case, along with basic statistics, are given in Table 2.2.

All of these simulations are performed via a lattice of 50^3 points for each octant of the tesseract, as described in Sect. 2.3. This corresponds to a varying wavelength resolution depending on the energy of the eigenfunction and position in the cell due to the metric described in Sect. 2.4. We are not concerned with replicating the precise local geometry and this resolution is always sufficient for our algorithm to fairly efficiently distinguish vortices.

Figure 2.15 shows examples of the vortex tangle in 3-sphere simulations at each of these energies. In each case, only the regions surrounding the central seven cells of the tesseract (in stereographic projection) are plotted, with vortices outside this region being topologically and geometrically trivial in the projection but travelling far from this central volume.

Table 2.2 Details of the simulations we have carried out on eigenfunctions of the 3-sphere. Errors given are standard deviations, reflecting the overall distribution of line lengths

Energy E	Number of cells simulated	Approx time per cell/min	Number of loops	
			Total	Average
120	2949	5	30910	10.5 ± 3.8
255	5697	11	107856	18.9 ± 5.50

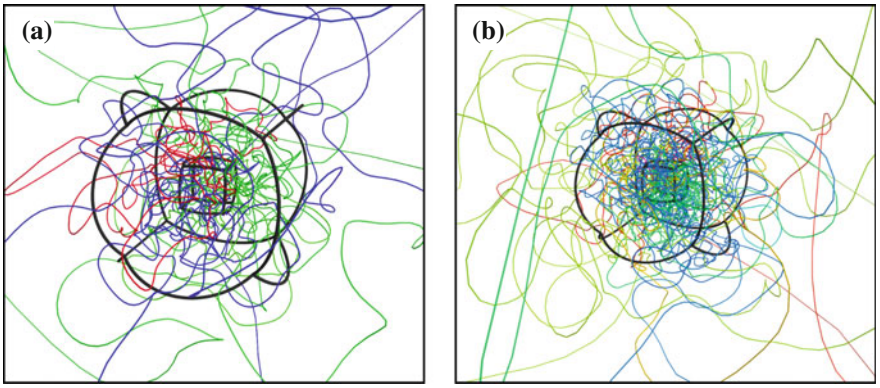

Fig. 2.15 Numerically tracked vortices in simulated degenerate eigenfunctions of the 3-sphere. **a**, **b** correspond to energies 120 and 255 respectively. The vortices are drawn in stereographic projection, with black lines showing some edges of the tesseract whose eight cells have been individually sampled, also plotted via projection to the 3-sphere. In each figure, every vortex is coloured by a different (random) hue. Only the area immediately surrounding the seven tesseract cells closest to the origin in projection is shown, with vortices outside this area corresponding to only a small true distance

2.5.3 The Quantum Harmonic Oscillator

Our core datasets of the QHO have energies $10 + \frac{3}{2}$, $15 + \frac{3}{2}$, $20 + \frac{3}{2}$ and $25 + \frac{3}{2}$ in (1.32). The numbers of simulated cells in each case, along with basic statistics, are given in Table 2.3.

The numerical grid size in all cases is identical, with vortices sampled in the region $-15 < x, y, z < 15$ at a resolution of 0.2 along every axis. The number of wavelengths that this resolution corresponds to varies both with position and with

Table 2.3 Details of the simulations we have carried out on eigenfunctions of the QHO. The 'number of infinite' column gives the number of vortices that extend to infinity (though becoming geometrically trivial outside the classical radius), while the 'number of loops' gives the number of vortices that close normally, largely within the classical radius. Errors given are standard deviations, reflecting the overall distribution of line lengths

Energy $E - \frac{3}{2}$	Number of cells simulated	Approx time per cell/min	Number of infinite		Number of loops	
			Total	Average	Total	Average
10	1032	2.5	10671	10.3 ± 2.9	3135	3.89 ± 2.15
15	296	3.5	4660	15.74 ± 3.84	2005	7.06 ± 3.65
20	5279	5.1	112384	21.29 ± 4.41	65318	12.40 ± 4.90
25	1413	7.5	37964	26.9 ± 4.9	22062	15.6 ± 6.3

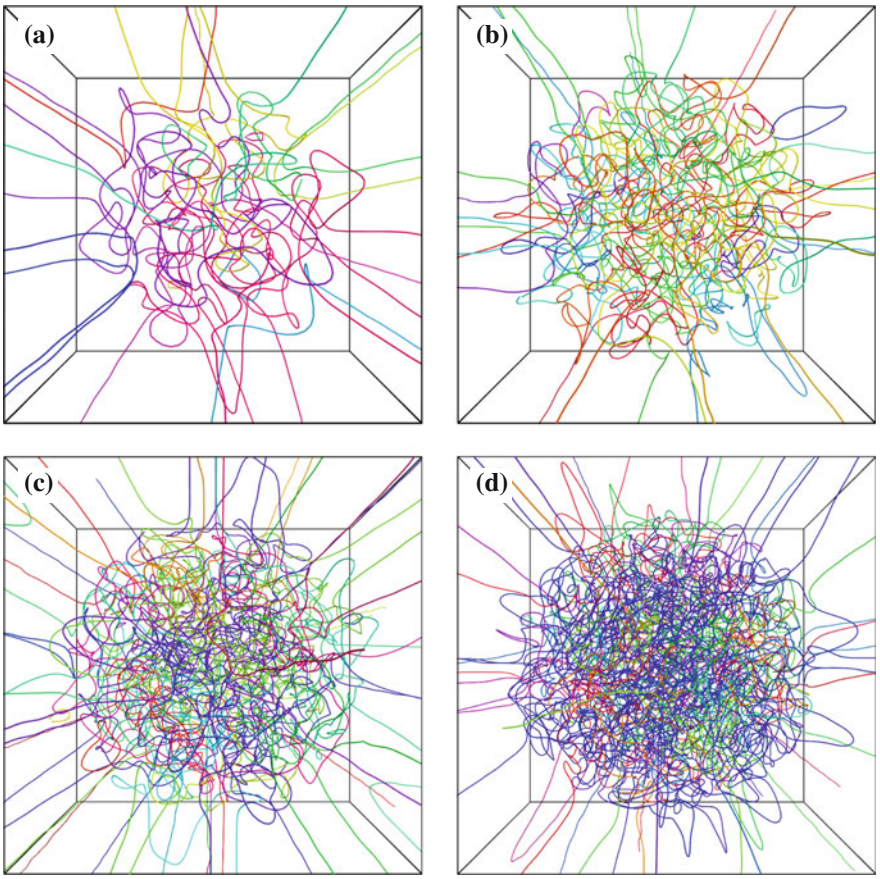

Fig. 2.16 Numerically tracked vortices in simulated degenerate eigenfunctions of the QHO. **a–d** correspond to energies E of 10, 15, 20 and 25 respectively. In each figure, every vortex is coloured by a different (random) hue. The drawn cell has total width 2.2 times the radius in the eigenfunction at which $E = 1/2(x^2 + y^2 + z^2)$, though the data tracks vortices over a larger volume to more reliably detect hairpins where vortices leaving this region return to the bulk of the tangle

energy (Sect. 2.4), but we are not concerned with geometrical correctness in these datasets and so the difference is not important.

Figure 2.16 shows examples of vortices in QHO cells at each of these energies.

References

1. K.-F. Berggren, P. Ljung, Nature of streamlines for Berry-type wave functions in open 3D cavities, in *Mathematical Modeling of Wave Phenomena*, *AIP Conference Proceedings*, vol. 1106 (2009), pp. 253–259

2. P. Ljung, A. Ynnerman, Extraction of intersection curves from iso-surfaces on co-located 3d grids, in *SIGRAD* (2003), pp. 23–28
3. J.-P. Thirion, A. Gourdon, The 3D marching lines algorithm. Graph. Model. Im. Proc. The 3D marching lines algorithm. **58**(6), 503–509 (1996)
4. K. O'Holleran, *Fractality and topology of optical singularities*. PhD thesis, Department of Physics and Astronomy, Faculty of Physical Sciences, University of Glasgow (2008)
5. K. O'Holleran, M.R. Dennis, F. Flossmann, M.J. Padgett, Fractality of light's darkness. Phys. Rev. Lett. **100**, 053902 (2008)
6. J. Leach, M.R. Dennis, J. Courtial, M. Padgett, Vortex knots in light. New J. Phys. **7** (2005)
7. K. O'Holleran, M.J. Padgett, M.R. Dennis, Topology of optical vortex lines formed by the interference of three, four and five plane waves. Opt. Express **14**, 3039–3044 (2006)
8. I.B. Halperin, Statistical mechanics of topological defects, in *Les Houches Session XXV–Physics of Defects*, ed. by R. Balian, K.l.éman, J.-P. Poirier (1981)
9. B.M. Caradoc-Davies, R.J. Ballagh, K. Burnett, Coherent dynamics of vortex formation in trapped bose-einstein condensates. Phys. Rev. Lett. **83**(5), 895–898 (1999)
10. B.M. Caradoc-Davies, *Vortex Dynamics in Bose-Einstein Condensates*. PhD thesis, University of Otago, Dunedin, New Zealand (2000)
11. M. Farge, K. Schneider, G. Pellegrino, A.A. Wray, R.S. Rogallo, Coherent vortex extraction in three-dimensional homogeneous turbulence: comparison between CVS-wavelet and POD-Fourier decompositions. Phys. Fluids **15**(10), 2886–2896 (2003)
12. G.P. Alexander, B.G. Chen, E.A. Matsumoto, R.D. Kamien, Colloquium: Disclination loops, point defects, and all that in nematic liquid crystals. Rev. Mod. Phys. **84**, 497 (2012)
13. F. Flossmann, K. O'Holleran, M.R. Dennis, M.J. Padgett, Polarization singularities in 2D and 3D speckle fields. Phys. Rev. Lett. **100**, 203902 (2008)
14. E. Matene, V. Jacquemet, Fully automated initiation of simulated episodes of atrial arrhythmias. Europace **14**, v17–v24 (2012)
15. K. Umapathy, K. Nair, S. Masse, S. Krishnan, J. Rogers, M. Nash, K. Nanthakumar, Phase mapping of cardiac fibrillation. Circ. Arrhythmia Electrophysiol. **3**, pp. 105–114 (2010)
16. J.M. Rogers, Combined phase singularity and wavefront analysis for optical maps of ventricular fibrillation. IEEE T Bio-Med. Eng. **51**(1), 56–65 (2004)
17. S. Chavez, Q.S. Xiang, L. An, Understanding phase maps in mri: a new cutline phase unwrapping method. IEEE Trans. Med. Imaging **21**(8), 966–977 (2002)
18. R. Yamaki, A. Hirose, Singularity-spreading phase unwrapping. IEEE T Geosci. Remote **45**(10), 3240–3251 (2007)
19. H.S. Abdul-Rahman, M.A. Gdeisat, D.R. Burton, M.J. Lalor, F. Lilley, C.J. Moore, Fast and robust three-dimensional best path phase unwrapping algorithm. Apply. Opt. **46**(26), 6623–6635 (2007)
20. T. Vachaspati, A. Vilenkin, Formation and evolution of cosmic strings. Phys. Rev. D **30**, 2036–2045 (1984)
21. F.Y. Wu, The Potts model. Rev. Mod. Phys. **54**, 235 (1982)
22. L.J. Rantner, L. Wieser, M.C. Stühlinger, F. Hintringer, B. Tilg, G. Fischer, Detection of phase singularities in triangular meshes. Method Inform. Med. **46**, 646–654 (2007)
23. M.R. Dennis, Nodal densities of planar Gaussian random waves. Eur. Phys. J. Spec. Top. **145**, 191–210 (2007). Conference on Nodal Patterns in Physics and Mathematics, Wittenberg, Germany, July 24–28, 2006
24. M. Hindmarsh, K. Strobl, Statistical properties of strings. Nucl. Phys. **B437**, 471–488 (1995)
25. J.F. Nye, Evolution of the hyperbolic umbilic diffraction pattern from airy rings. J. Opt. A **8**, 304–314 (2006)
26. A.J. Taylor, M.R. Dennis, Geometry and scaling of tangled vortex lines in three-dimensional random wave fields. J. Phys. A **47**(46), 465101 (2014)
27. A.W. Baggaley, C.F. Barenghi, Spectrum of turbulent Kelvin-waves cascade in superfluid helium. Phys. Rev. E **83**, 134509 (2011)
28. H. Manning, *Geometry of Four Dimensions*. Applewood Books (2011)

http://www.springer.com/978-3-319-48555-3

Analysis of Quantised Vortex Tangle

Taylor, A.J.

2017, XVI, 197 p. 95 illus., 84 illus. in color., Hardcover
ISBN: 978-3-319-48555-3