Simulation on Rank-1 Lattices

H. Dammertz, A. Keller and S. Dammertz
Institute of Media Informatics, Ulm University, Germany

Abstract

Rank-1 lattices are available in any dimension for any number of lattice points and because their generation is so efficient, they often are used in quasi-Monte Carlo methods. Applying the Fourier transform to functions sampled on rank-1 lattice points turns out to be simple and efficient if the number of lattice points is a power of two. Considering the Voronoi diagram of a rank-1 lattice as a partition of the simulation domain and its dual, the Delauney tessellation, as a mesh for display and interpolation, rank-1 lattices are an interesting alternative to tensor product lattices. Instead of classical criteria, we investigate lattices selected by maximized minimum distance, because then the Delauney tessellation becomes as equilateral as possible. Similar arguments apply for the selection of the wave vectors. We explore the use of rank-1 lattices for the examples of stochastic field synthesis and a simple fluid solver with periodic boundary conditions.

1 Introduction

Many simulations are evaluated on Cartesian tensor product lattice structures although their sampling efficiency is not optimal [PM62]. Rank-1 lattices allow for a better sampling efficiency when selected carefully. We develop and review the basic tools like meshing, fast Fourier transform, lattice cell access, and interpolation for simulation on rank-1 lattices. In addition we give insight how to choose suitable rank-1 lattice parameters. We illustrate our new techniques for generating ocean waves as stochastic field and a simple fluid dynamics simulation. The most prominent example is the simulation of the ocean surface [AR86, Tes00] by random fields as used in the movies Titanic, Waterworld, or The Devil's Advocate [Ent]. The same principle has been applied to modeling of turbulent wind fields and various other phenomena [SF91, SF93, Sta95, Sta97].

2 Rank-1 Lattices

A discrete subset of $\mathbb{R}^s$ that contains $\mathbb{Z}^s$ and is closed under addition and subtraction is called a lattice [SJ94]. Rank-1 lattices

$$L_{n,\mathbf{g}} := \left\{ \frac{l}{n}\mathbf{g} + \Delta : \Delta \in \mathbb{Z}^s; l = 0, \ldots, n-1 \right\}$$

are defined by using only one suitable generator vector $\mathbf{g} \in \mathbb{N}^s$ for a fixed number $n \in \mathbb{N}$. Often it is more useful to consider their restriction

$$L_{n,\mathbf{g}} \cap [0,1)^s = \left\{ \mathbf{x}_l := \frac{l}{n}\mathbf{g} \bmod 1 : l = 0, \ldots, n-1 \right\}$$

to the s-dimensional unit torus $[0,1)^s$ (see the example in Figure 1). A notable advantage of rank-1 lattices over tensor product lattices is that they exist for any number n of points in any dimension s.

An $s \times s$ matrix V is called basis of the lattice $L_{n,\mathbf{g}}$, if $L_{n,\mathbf{g}} = \{\mathbf{x} = V\mathbf{l} : \mathbf{l} \in \mathbb{Z}^s\}$. Of all possible bases the Minkowski-reduced bases [AEVZ02] are the most useful for our purpose. Such a basis

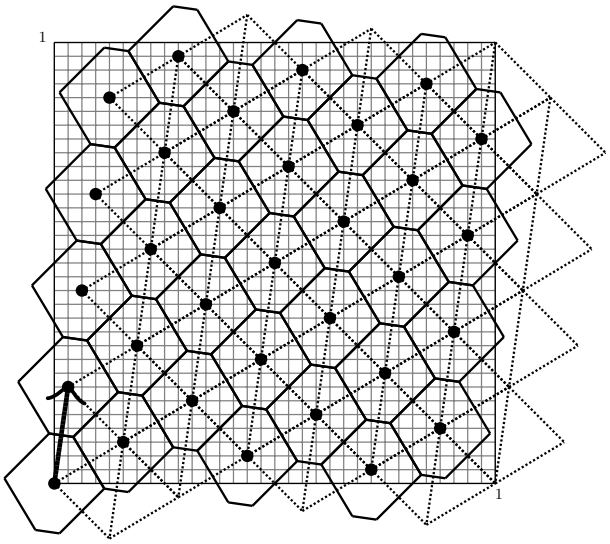


Figure 1: Illustration of the geometry of rank-1 lattices for an example in $s = 2$ dimensions with $n = 32$ points and the generator vector $\mathbf{g} = \binom{1}{7}$ (see the solid arrow from the origin). The solid lines depict the Voronoi diagram. Each cell is generated by a lattice point and contains the points of the unit torus, which are closer to this lattice point than to any other. In addition the lattice points are the centroids of the Voronoi cells. The dashed lines represent the dual graph, which is the Delauney tessellation.

contains the s shortest linearly independent vectors and can be found by a computer search over all n points $\mathbf{x}_l$ using their shortest distance to the origin on the unit torus. These vectors actually describe the Delauney tessellation and can be used for accessing neighboring lattice points.

2.1 Fast Fourier Transform

The fast Fourier transform is a versatile tool in simulation. Usually the transform has to be performed for each coordinate once. Instead of the standard tensor product algorithm, rank-1 lattices in s dimensions allow for transforming the data using only the one-dimensional Fourier transform [LH03], which is simpler to implement and a little bit more efficient for the same number of lattice points.

For this purpose the set $K_n := \{\vec{k}_0, \ldots, \vec{k}_{n-1}\} \subset \mathbb{Z}^s$ of wave vectors has to be selected such that each wave vector

$$\vec{k}_j \in Z_j := \{\vec{k} \in \mathbb{Z}^s : \vec{k} \cdot \vec{g} \equiv j \pmod{n}\}, \tag{1}$$

where $\mathbf{g}$ is the generator vector of the rank-1 lattice $L_{n,\mathbf{g}}$ under consideration. Hence,

$$\vec{k}_j \cdot \vec{x}_l = \vec{k}_j \cdot \frac{l}{n}\vec{g} = (j + r_j n)\frac{l}{n} = \frac{jl}{n} + r_j l$$

for some integer $r_j \in \mathbb{Z}$. Given Fourier coefficients $\hat{\vec{f}}(\vec{k}_j)$, synthesizing a function $\vec{f}$ on the lattice $L_{n,\mathbf{g}}$ by

$$\vec{f}(\vec{x}_l) \quad = \quad \sum_{j=0}^{n-1} \hat{\vec{f}}(\vec{k}_j) e^{2\pi i \vec{k}_j \cdot \vec{x}_l} = \sum_{j=0}^{n-1} \hat{\vec{f}}(\vec{k}_j) e^{2\pi i (\frac{jl}{n} + r_j l)} = \sum_{j=0}^{n-1} \hat{\vec{f}}(\vec{k}_j) e^{2\pi i \frac{jl}{n}} \tag{2}$$

in fact turns out to be a one-dimensional finite Fourier series independent of the dimension s, because $r_j l$ is integer and therefore $e^{2\pi i r_j l} = 1$.

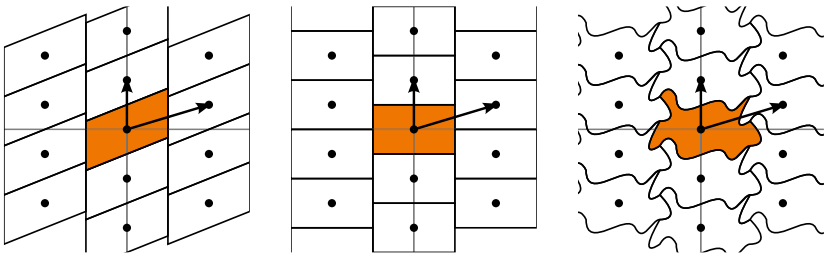


Figure 2: Examples of monohedral tilings of the 2-dimensional plane with the dual lattice and its basis. The arrows show the Minkowski-reduced basis vectors.

For n being a power of two, the fast inverse Fourier transform can synthesize the function in all lattice points most efficiently. Given a function $\vec{f}(\mathbf{x}_l)$ the fast Fourier transform can be used for the analysis, too:

$$\hat{\vec{f}}(\mathbf{k}_j) = \sum_{l=0}^{n-1} \vec{f}(\mathbf{x}_l) e^{-2\pi i \frac{jl}{n}}.$$

2.2 Choosing the Wave Vectors

There are infinitly many choices of wave vectors as defined by the sets Z_j in equation (1). Understanding the connection between the wave vectors and the structure of a rank-1 lattice helps to choose the best wave vectors for a given problem and provides a way to construct these wave vectors.

The dual lattice

$$L^{\perp}_{n,\mathbf{g}} := \{\mathbf{k} \in \mathbb{Z}^s : \mathbf{k} \cdot \mathbf{g} \equiv 0 \pmod{n}\} = Z_0$$

of a rank-1 lattice $L_{n,\mathbf{g}}$ has the basis $U = (V^{-1})^T$. Now the set K_n of wave vectors is U periodic [PM62], i.e. it has the property that the sets

$$K'_n := K_n + U\mathbf{l} \qquad \mathbf{l} \in \mathbb{Z}^s$$

are valid sets of n wave vectors with $K'_n \cap K_n = \emptyset$ as well. Consequently, any tile that results in a monohedral tiling of $\mathbb{Z}^s$ (see the illustration in Figure 2) with periodicity U can be used as a set of wave vectors [LH03].

Once such a tiling is chosen all integer vectors in the interior of one cell are the wave vectors. Note that the choice of the tiling is arbitrary and the elements of a single tile are not necessarily connected. However, one can use known spectral properties of the function that should be synthesized or analyzed. If no spectral properties are known a reasonable assumption for practical problems is an isotropic spectrum (i.e. no preferred direction) and that low frequencies are most important. This results in choosing the wave vectors in the fundamental (i.e. including the origin) Voronoi cell of $L^{\perp}_{n,\mathbf{g}}$, which is illustrated in Figure 3.

The above assumptions also provide a criterion for choosing the generator vector $\mathbf{g}$ of the rank-1 lattice: Choose $\mathbf{g}$ so that the in-circle of the fundamental Voronoi cell of the dual lattice is maximized. This is equivalent to maximizing the sampling efficiency

$$\eta := \frac{R}{P}$$

as defined by Petersen [PM62], where R is the volume of the in-circle of the Voronoi region and P is the volume of the fundamental Voronoi cell. The sampling efficiency measures how many of the important frequencies are actually captured by sampling with a given lattice.

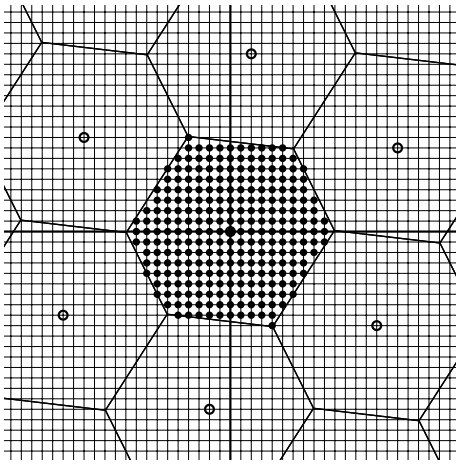
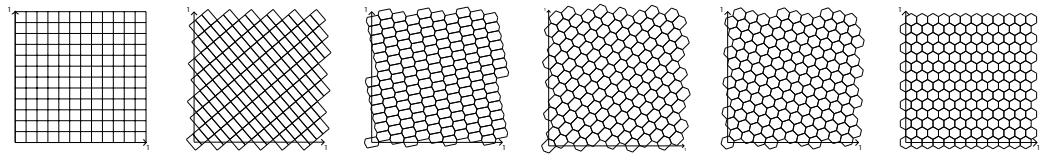


Figure 3: Dual lattice (circles) of a rank-1 lattice with $n = 256$ and $\mathbf{g} = \binom{1}{30}$. The set of wave vectors K_n (solid disks) in the fundamental Voronoi cell is highlighted.



rect. lattice $\eta = 78.5\%$ | $\mathbf{g} = \binom{1}{42}$ $\eta = 46.3\%$ | $\mathbf{g} = \binom{1}{67}$ $\eta = 56.7\%$ | $\mathbf{g} = \binom{1}{89}$ $\eta = 69.8\%$ | $\mathbf{g} = \binom{1}{33}$ $\eta = 87.3\%$ | hex. lattice $\eta = 90.7\%$

Figure 4: Sampling efficiency η of different lattices with $n = 144$. Note that the Fibnacci lattice with $\mathbf{g} = \binom{1}{89}$ is not the best choice with respect to sampling efficiency.

For rank-1 lattices this ratio can be maximized by choosing the generator vector such that the minimal distance between any two points of the dual lattice is maximized. Figure 4 shows the Voronoi diagrams of different rank-1 lattices, where a Cartesian tensor product and hexagonal lattice are shown for comparison. The hexagonal lattice is optimal with respect to the sampling efficiency in two dimensions and maximizing the minimum distance in a rank-1 lattice yields a good approximation. With increasing number of points the sampling efficiency of rank-1 lattices approaches the sampling efficiency of the hexagonal lattice.

Of course, if other spectral properties are known the rank-1 lattice search can be adapted for a better approximation of this kind of functions.

3 Applications in Computer Graphics

We illustrate the idea of simulation on rank-1 lattices by implementing two examples from the domain of computer graphics and animation in the new framework.

3.1 Spectral Synthesis of Ocean Waves

Along the method used by Tessendorf [Tes00], a periodic ocean tile (see Figure 5) is realized as a stochastic field using Fourier synthesis on a rank-1 lattice. Using the Fourier coefficients

$$\hat{h}(\mathbf{k}, t) := \hat{h}_0(\mathbf{k}) e^{i\omega(\mathbf{k})t} + \hat{h}_0^*(-\mathbf{k}) e^{-i\omega(\mathbf{k})t}$$

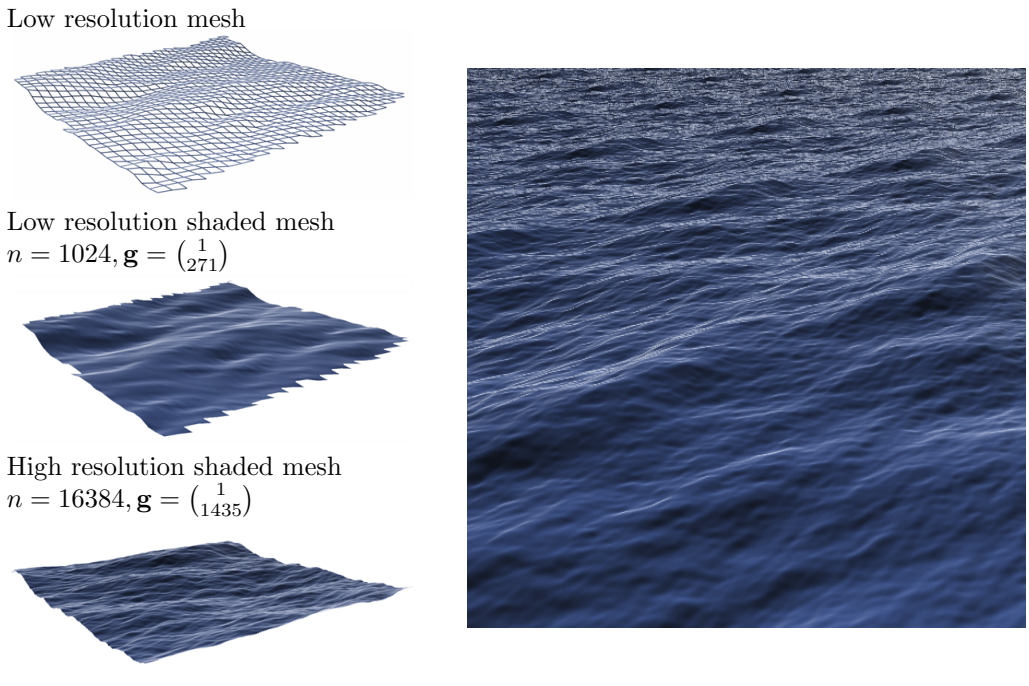


Figure 5: Left: Synthesized periodic $50m \times 50m$ tiles in two resolutions. Right: Modeling a larger piece of the ocean by tiling the periodic patches.

the height field

$$h(\mathbf{x}_l, t) := \sum_{j=0}^{n-1} \hat{h}(\mathbf{k}_j, t) e^{2\pi i \frac{jl}{n}}$$

becomes periodic in time t and real. For deep water the speed of a wave is given by the dispersion relation $\omega(\mathbf{k}) = \sqrt{g\|\mathbf{k}\|}$, where g is the gravitational constant. Based on observations from oceanography waves can be modeled statistically independent and normally distributed. Therefore, the amplitudes

$$\hat{h}_0(\mathbf{k}) := \frac{1}{\sqrt{2}}(\xi_r + i\xi_i)\sqrt{P_h(\mathbf{k})}$$

are realized using Gaussian random numbers ξ_r and ξ_i modulated by a spectrum. Out of many alternatives we chose the Phillips spectrum

$$P_h(\mathbf{k}) := A\frac{e^{-\frac{1}{(\|k\|L)^2}}}{k^4}|\mathbf{k}\cdot\mathbf{w}|^2$$

A	Phillips constant
$L = \frac{v^2}{g}$	Largest wave for windspeed v
$\mathbf{w}$	wind direction

which considers parameters like wind speed and direction. For the sake of completeness we mention that the gradient vector of the height field can be computed using the Fourier transform as well. This yields more precise normals for shading as those computed by finite differences.

3.1.1 Implementation

The synthesis of stochastic fields on rank-1 lattices consists of the following choices and decisions:

Number n of lattice points: Although rank-1 lattices exist for any number of points in any dimension, the fast Fourier transform is most efficient for n being a power of 2.

Generator vector g: In order to maximize the sampling efficiency we select a generator $\mathbf{g}$ that maximizes the minimum distance of the dual lattice. If the generator vector has Korobov form, i.e. $\mathbf{g} = (1, a, a^2, a^3, \ldots)$, the spectral test [Knu77] can be used to efficiently compute the minimum distance for each candidate a. While not true in general, for dimension $s = 2$ and n as a power of 2, one of the two components of the generator vector $\mathbf{g} = \binom{a_1}{a_2}$ has to be odd (w.l.o.g. $\gcd(a_1, n) = 1$). Otherwise points would coincide resulting in an obviously useless minimum distance of 0. Then for every j with $\gcd(j, n) = 1$ every vector $\mathbf{x}_j = j\mathbf{g} \bmod n$ is a generator vector of the same lattice (generator of the cyclic group), too, and there must exist an $l \in \{1, 3, 5, \ldots, n-1\}$ with $\mathbf{x}_l = \binom{1}{a}$. Thus a generator vector in Korobov form exists that obtains maximized minimum distance and the spectral test can be used. A list of all parameters for 2-dimensional maximized minimum distance rank-1 lattices is found in Table 1.

Basis V: V is determined as a Minkowski-reduced basis, which defines the Delauney triangulation that is used as the triangle mesh. Table 1 lists these basis vectors, however, multiplied by n. Given a generator vector in Korobov form, the first coordinate of each of these integer basis vectors is the increment or decrement to find the index of a neighboring lattice point.

Wave vectors K_n: We enumerate all wave vectors in a conservative bounding box of the fundamental Voronoi cell of the dual lattice and select the shortest ones. As a simple conservative convex hull we chose the axis-aligned bounding box determined by the direct lattice point neighbors of the origin. A much more involved approach is to compute the fundamental Voronoi cell in the dual lattice and rasterize it on the integer lattice.

3.2 Stable Simulation of Fluids

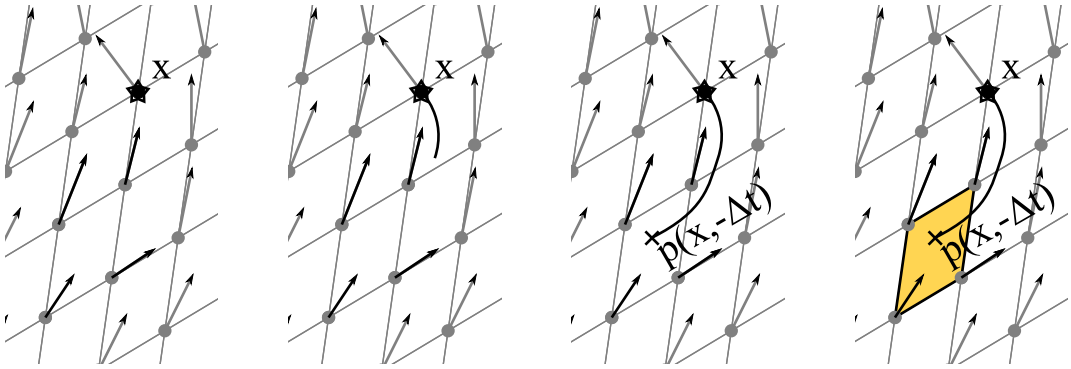
The stable fluids algorithm by Stam [Sta99] is a practical way of simulating incompressible fluids for animation. Note that the algorithm focuses on realtime simulation rather than on precision. The simulation is based on the Navier-Stokes equations

$$\operatorname{div} v = 0 \tag{3}$$

$$\frac{\partial v}{\partial t} = -(v \cdot \nabla)v + \nu \Delta v + f \tag{4}$$

for incompressible fluids, where v is the velocity field, ν the viscosity, and f are the external forces. The solution strategy is to simulate equation (4) and remove the divergence (3) at the end of each time step by using a projection based on the Helmholtz-Hodge decomposition $w = v + \nabla q$, which states that a vector field w can be decomposed into a divergence free part v and the gradient of a scalar field q. The velocity field for the next time step $t + \Delta t$ is computed in four steps [Sta99]:

1. The external forces are added: $v_1(\mathbf{x}_l) := v(\mathbf{x}_l) + \Delta t \cdot f(\mathbf{x}_l)$
2. The advection is computed by tracing back a particle back in time starting from point $\mathbf{x}_l$ according to the velocity field. The position $p(\mathbf{x}_l, -\Delta t)$ is computed by dividing the time step Δt into smaller time steps and performing the Euler rule for each of the small time steps (see the illustration). The velocities $v_2(\mathbf{x}_l) := v_1(p(\mathbf{x}_l, -\Delta t))$ are linearly interpolated using the closest lattice points. Due to linear interpolation the method is named stable, because the computed velocities never can exceed the old ones in magnitude.



3. The diffusion by the Laplace operator is efficiently computed as low pass filter in the Fourier domain. The Fourier coefficients are computed by

$$\hat{v}_2(\mathbf{k}_j) := \sum_{l=0}^{n-1} v_2(\mathbf{x}_l) e^{-2\pi i \frac{jl}{n}}$$

and filtered

$$\hat{v}_3(\mathbf{k}_j) := \frac{\hat{v}_2(\mathbf{k}_j)}{1 + \nu \Delta t \cdot (\mathbf{k}_j \cdot \mathbf{k}_j)}.$$

4. The divergence is removed using the projection

$$\hat{v}_4(\vec{k}_j) := \hat{v}_3(\vec{k}_j) - \frac{(\mathbf{k}_j \cdot \hat{v}_3(\vec{k}_j))\mathbf{k}_j}{(\mathbf{k}_j \cdot \mathbf{k}_j)}$$

and the velocity field at time step $t + \Delta t$ is synthesized by

$$v_4(\vec{x}_l) := \sum_{j=0}^{n-1} \hat{v}_4(\vec{k}_j) e^{2\pi i \frac{jl}{n}}.$$

3.2.1 Implementation

The stable fluids scheme has been implemented for two- and three-dimensional velocity fields as illustrated in Figure 6. The Fourier transformation techniques are the same as in the previous application example, except that they have to be performed for each component of the vector fields.

For linear interpolation (as required in step 2 of the algorithm) on a rank-1 lattice, the Minkowski-reduced basis V is used to access neighboring lattice points.

Scaling all lattice points by n results in integer coordinates for all $\mathbf{x}_l$ and the basis V.

Frame: Representing the velocity field v in the basis V avoids a transformation during interpolation. In this case external forces usually have to be transformed into the basis V.

Accessing lattice cells: The backtracking step requires to compute the index of a lattice cell containing a given point, which is simple if the lattice is given in Korobov form: Multiplying the basis matrix V by the integer parts of the coordinates of the point modulo n yields a lattice point in Cartesian coordinates. Obviously the first component is the lattice point or cell index. Neighboring lattice points for interpolation now are found as described in the previous example.

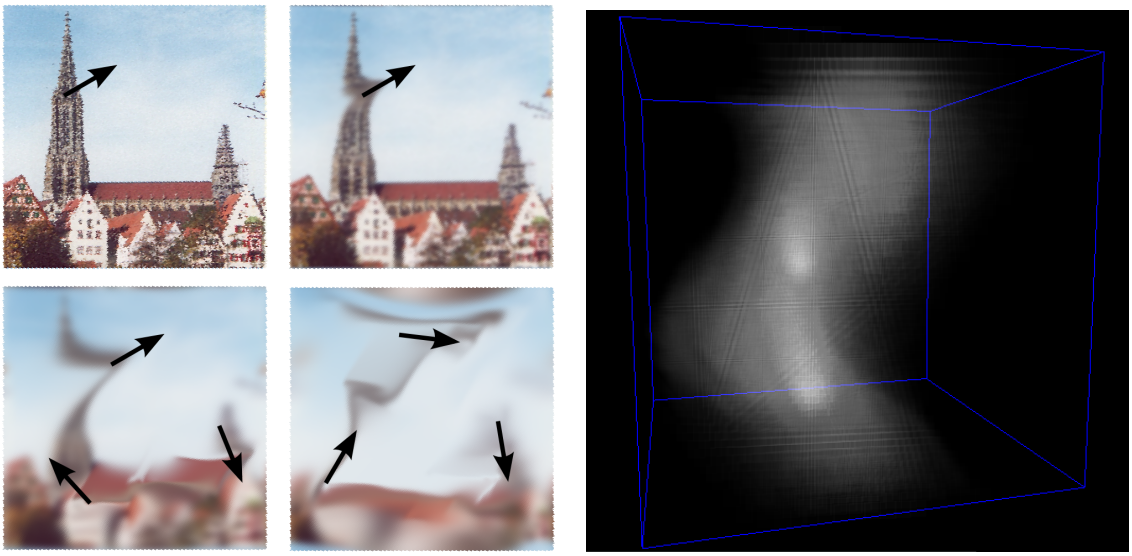


Figure 6: Left: The four images are subsequent snapshots of the stable fluid simulation on a rank-1 lattice. The arrows indicate the external forces applied to the periodic fluid. The fluid transports the background image and the effects of advection and diffusion, i.e. blur, are clearly visible. Right: Snapshot of a wind field simulation in three dimensions for the lattice $L_{32768,\mathbf{g}}$ with $\mathbf{g} = (1, 10871, 10871^2)$, where smoke is transported in a velocity field.

4 Conclusion

Spectral synthesis and simulation on rank-1 lattices can be implemented efficiently. Independent of the dimension s only a one dimensional Fourier transform is needed. Additionally the approximation can be more accurate than on a tensor product lattice [LH03, KSW04, DKKS]. It is also notable that the isotropic measure of maximized minimum distance can replace the classical measures like e.g. discrepancy (see the sampling efficiency of the Fibonacci lattice in Figure 4) often used in connection with rank-1 lattices. This measure also provides best visual quality.

In the future we like to extend our ideas to hierarchical approaches using lattice sequences and explore optimizations for anisotropic spectra. Moreover we will explore non-periodic boundaries.

Acknowledgments

The authors would like to thank mental images GmbH for support and for funding of this research. The second author is very thankful for the very fruitful communication with Li Dong, Fred Hickernell, and Peter Schröder. Note that the article entirely includes the technical report 307/01 "Random Fields on Rank-1 Lattices" by Alexander Keller, University of Kaiserslautern, that is based on the presentation "Solutions of P.D.E.s using Spectral Methods with Integration Lattices" by Li Dong and Fred Hickernell at the 4th International Conference on Monte Carlo and quasi- Monte Carlo Methods in Scientific Computing, Hong Kong Baptist University in 2000.

References

[AEVZ02] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[AR86] A.Fournier and W. Reeves. A simple model of ocean waves. In *SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, pages 75–84, New York, NY, USA, 1986.

[DKKS] J. Dick, P. Kritzer, F. Kuo, and I. Sloan. Lattice-Nyström Method for Fredholm Integral Equations of the Second Kind. *Submitted to the Journal of Complexity.*

[Ent] Areté Entertainment. http://www.areteentertainment.com.

[Knu77] D. Knuth. *The Art of Computer Programming, Volume 2, Seminumerical Algorithms.* Addison Wesley, 1977.

[KSW04] F. Kuo, I. Sloan, and H. Woźniakowski. Lattice Rules for Multivariate Approximation in the Worst Case Setting. *Monte Carlo and Quasi-Monte Carlo Methods 2004*, (H. Niederreiter and D. Talay, eds.), Springer-Verlag, pages 289–330, 2006.

[LH03] D. Li and F. Hickernell. Trigonometric spectral collocation methods on lattices. *Recent Advances in Scientific Computing and Partial Differential Equations, AMS Series on Contemporary Mathematics*, pages 121–132, 2003.

[PM62] D. Petersen and D. Middleton. Sampling and reconstruction of wave-number-limited functions in N-dimensional Euclidean spaces. *Information and Control*, 5(4):279–323, 1962.

[SF91] J. Stam and E. Fiume. A multiple-scale stochastic modelling primitive. In *Proceedings of Graphics Interface '91*, pages 24–31, 1991.

[SF93] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. *Computer Graphics*, 27(Annual Conference Series):369–376, 1993.

[SJ94] I. Sloan and S. Joe. *Lattice Methods for Multiple Integration.* Clarendon Press Oxford, 1994.

[Sta95] J. Stam. *Multi-Scale Stochastic Modelling of Complex Natural Phenomena.* PhD thesis, University of Toronto, 1995.

[Sta97] J. Stam. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 16(3):C159–C164, 1997.

[Sta99] J. Stam. Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 121–128, 1999.

[Tes00] J. Tessendorf. Simulating ocean water. In *SIGGRAPH 2000 Course Notes: Course 25.*

i	$n = 2^i$	generator $\mathbf{g}$	basis vectors $V = (v_1\, v_2)$
2	4	(1, 2)	(2, 0), (1, 2)
3	8	(1, 3)	(2, -2), (1, 3)
		(1, 5)	(2, 2), (1, -3)
4	16	(1, 4)	(4, 0), (1, 4)
		(1, 12)	(4, 0), (1, -4)
5	32	(1, 7)	(4, -4), (5, 3)
		(1, 9)	(4, 4), (3, -5)
		(1, 23)	(4, -4), (3, 5)
		(1, 25)	(4, 4), (5, -3)
6	64	(1, 28)	(7, 4), (2, -8)
		(1, 36)	(7, -4), (2, 8)
7	128	(1, 12)	(11, 4), (1, 12)
		(1, 116)	(11, -4), (1, -12)
8	256	(1, 30)	(9, 14), (17, -2)
		(1, 226)	(9, -14), (17, 2)
9	512	(1, 200)	(18, 16), (23, -8)
		(1, 312)	(18, -16), (23, 8)
10	1024	(1, 271)	(34, -2), (15, -31)
		(1, 495)	(2, -34), (31, -15)
		(1, 529)	(2, 34), (31, 15)
		(1, 753)	(34, 2), (15, 31)
11	2048	(1, 592)	(45, 16), (7, 48)
		(1, 1456)	(45, -16), (7, -48)
12	4096	(1, 70)	(59, 34), (58, -36)
		(1, 4026)	(59, -34), (58, 36)
13	8192	(1, 1530)	(91, -34), (75, 62)
		(1, 6662)	(91, 34), (75, -62)
14	16384	(1, 1435)	(57, -125), (137, -13)
		(1, 6291)	(125, -57), (13, -137)
		(1, 10093)	(125, 57), (13, 137)
		(1, 14949)	(57, 125), (137, 13)
15	32768	(1, 15936)	(183, -64), (146, 128)
		(1, 16832)	(183, 64), (146, -128)
16	65536	(1, 25962)	(260, -88), (53, -270)
		(1, 39574)	(260, 88), (53, 270)
17	131072	(1, 49531)	(172, -348), (217, 323)
		(1, 62899)	(348, -172), (323, 217)
		(1, 68173)	(348, 172), (323, -217)
		(1, 81541)	(172, 348), (217, -323)
18	262144	(1, 1990)	(527, 154), (395, -382)
		(1, 260154)	(527, -154), (395, 382)
19	524288	(1, 86592)	(775, -64), (442, 640)
		(1, 437696)	(775, 64), (442, -640)
20	1048576	(1, 195638)	(134, 1092), (879, -662)
		(1, 852938)	(134, -1092), (879, 662)
21	2097152	(1, 193293)	(1226, -958), (217, 1541)
		(1, 715835)	(958, 1226), (1541, -217)
		(1, 1381317)	(958, -1226), (1541, 217)
		(1, 1903859)	(1226, 958), (217, -1541)
22	4194304	(1, 1120786)	(363, -2170), (1699, 1398)
		(1, 3073518)	(363, 2170), (1699, -1398)
23	8388608	(1, 1671221)	(1807, -2533), (3097, 301)
		(1, 3288547)	(2533, 1807), (301, -3097)
		(1, 5100061)	(2533, -1807), (301, 3097)
		(1, 6717387)	(1807, 2533), (3097, -301)
24	16777216	(1, 7605516)	(2903, -3308), (1414, 4168)
		(1, 9171700)	(2903, 3308), (1414, -4168)
25	33554432	(1, 1905545)	(405, -6211), (5582, -2754)
		(1, 14462279)	(6211, 405), (2754, 5582)
		(1, 19092153)	(6211, -405), (2754, -5582)
		(1, 31648887)	(405, 6211), (5582, 2754)
26	67108864	(1, 22282116)	(6391, -6052), (8436, 2512)
		(1, 44826748)	(6391, 6052), (8436, -2512)
27	134217728	(1, 58928436)	(9147, 8444), (2740, -12144)
		(1, 75289292)	(9147, -8444), (2740, 12144)
28	268435456	(1, 86198508)	(682, 17592), (15577, 8204)
		(1, 182236948)	(682, -17592), (15577, -8204)
29	536870912	(1, 8370742)	(11737, 21958), (24885, 814)
		(1, 528500170)	(11737, -21958), (24885, -814)
30	1073741824	(1, 78999493)	(10221, -33695), (24071, 25699)
		(1, 284281075)	(33695, 10221), (25699, -24071)
		(1, 789460749)	(33695, -10221), (25699, 24071)
		(1, 994742331)	(10221, 33695), (24071, -25699)
31	2147483648	(1, 940574718)	(38453, 31638), (8176, -49120)
		(1, 1206908930)	(38453, -31638), (8176, 49120)

Table 1: Parameters of all maximized minimum distance lattices in two dimensions with $n = 2^i$ points for $i = 2, \ldots, 31$. Note that the basis vectors are given in integer precision and have to be divided by the number of lattice points.