Eurographics Symposium on Rendering 2019
T. Boubekeur and P. Sen
(Guest Editors)

Volume 38 (2019), Number 4

Orthogonal Array Sampling for Monte Carlo Rendering: supplemental document

Wojciech Jarosz Afnan Enayet Andrew Kensler Charlie Kilpatrick Per Christensen
Dartmouth College Pixar Animation Studios

1. Further reading list

Below is our suggested sequence through a subset of the OA literature for graphics researchers to most quickly get ramped up on these topics.

Graphics

1. First become familiar with Chiu, Shirley, and Wang's original multi-jittered sampling paper [CSW94] as it is the conceptual precursor to orthogonal arrays.
2. Then read Kensler's correlated multi-jittered tech report [Ken13]. While the title suggests the main contribution is to incorporate favorable correlations, more importantly, this paper shows how to construct multi-jittered samples *in-place*, one sample at a time without requiring any storage.

Orthogonal array basics

3. The preface and first chapter of Hedayat, Sloane, and Stufken [HSS99] provides a great non-technical introduction to orthogonal arrays from the experimental design perspective.
4. Sections 10.3 and 10.4 of Owen [Owe13] provide a great Monte Carlo integration-centric introduction to Latin hypercubes and orthogonal arrays. We suggest reading it before diving into more detail in the original publications.
5. [Tan93] first introduced the idea of enforcing both OA stratification and Latin hypercube stratification with "orthogonal array-based Latin hypercube designs" also called "U-designs" (similar to Chiu, Shirley, and Wang's multi-jittered stratification, but for points of arbitrary dimension).
6. Many more general OA construction routines require arithmetic on finite/Galois fields. This is more involved than our routines using modular arithmetic, but many open-source libraries exist (particularly for binary finite fields) since these are used in cryptography. Hedayat, Sloane, and Stufken [HSS99] provides a good introduction to Galois fields within the context of orthogonal arrays.

Additional avenues

After becoming familiar with the basics from the main paper and the references above, there are several promising off-shoots that may be useful for graphics:

7. **Strong orthogonal arrays:** These enforce stratification in all dimensional projections, and are strongly related to (t,m,s)-nets from QMC. [HT12] originally proposed them, and there has been active work in this area since then [HCT18; HT14; LL15; Wen14].
8. **Nested orthogonal arrays/Latin hypercube designs:** These are a stepping stone to progressive sample sequences. They allow creating two or more layers of nested point sets [HHQ16; AZ16; HQ10; HQ11; QA10; WL13; ZWD19; RHVD10; Qia09; QAW09; YLL14; SLQ14].
9. **Correlation-controlled Latin hypercubes/orthogonal arrays:** Instead of independently randomizing the factors and levels of an orthogonal array, variance can be reduced by controlling the correlations [Owe94; CQ14; Tan98]. This is the same idea used by correlated multi-jittered sampling [Ken13].

2. Additional and alternate code listings

Here we include alternate code listings and additional visualizations and variance graphs.

Listing 1: *A modified version of Listing 1 from the main document which computes all dimensions (instead of just one) of an arbitrary sample from a Bose OA pattern.*

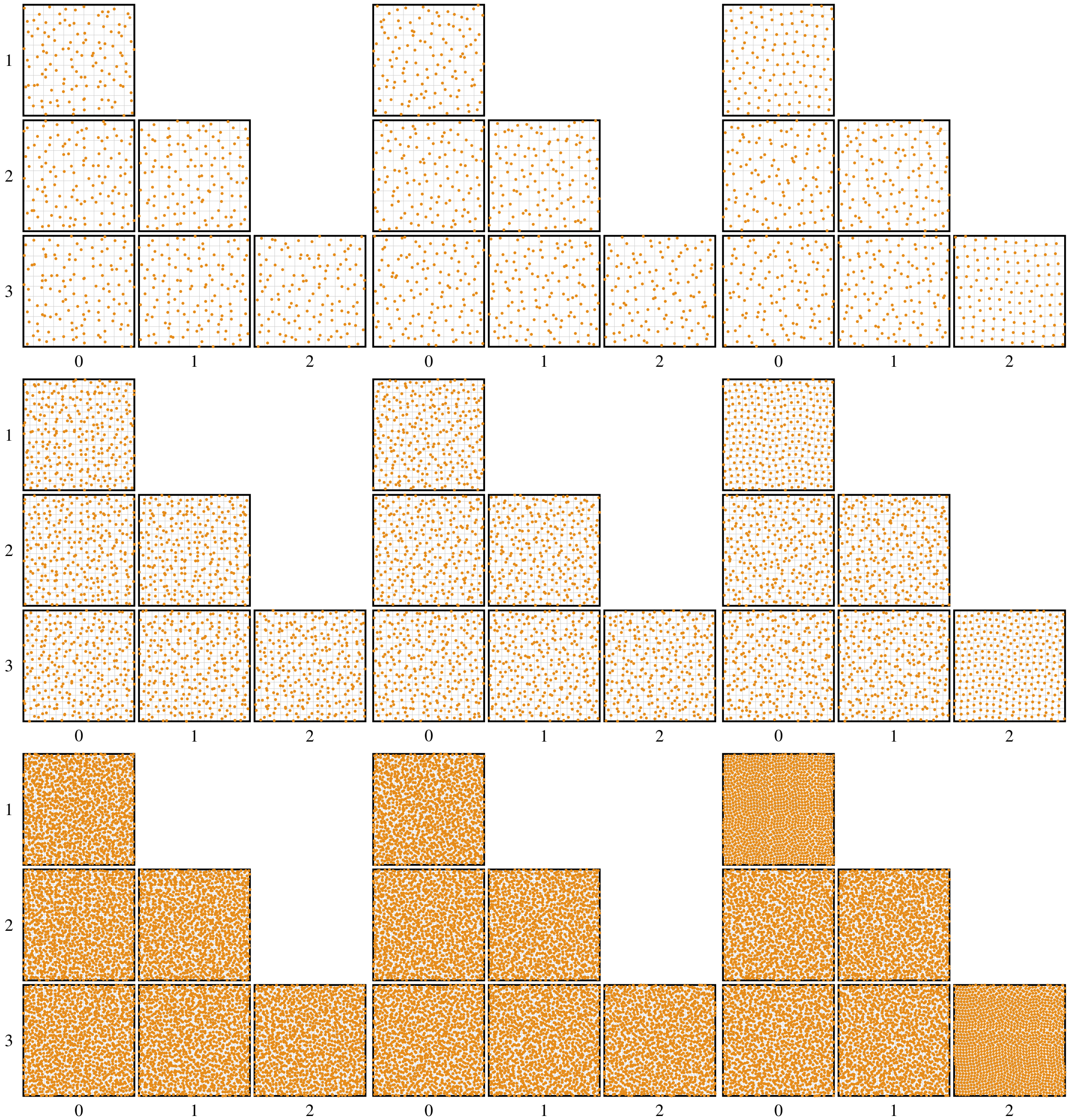
```
vector<float> boseOA(unsigned i,      // sample index
                     unsigned d,      // number of dimensions (<= s+1)
                     unsigned s,      // number of levels/strata
                     unsigned p[],    // array of d pseudo-random seeds
                     OffsetType ot) { // J, MJ, or CMJ
    vector<float> Xi(d);
    i                = permute(i % (s*s), s*s, p[0]);
    unsigned Ai0     = i % s;
    unsigned Ai1     = i / s;
    unsigned stratumX = permute(Ai0, s,     p[0] * 0x51633e2d);
    unsigned stratumY = permute(Ai1, s,     p[1] * 0x51633e2d);
    float sstratX    = offset(Ai0, Ai1, s, p[0] * 0x68bc21eb, ot);
    float sstratY    = offset(Ai1, Ai0, s, p[1] * 0x68bc21eb, ot);
    float jitterX    = randfloat(i,         p[0] * 0x02e5be93);
    float jitterY    = randfloat(i,         p[1] * 0x02e5be93);
    Xi[0]            = (stratumX + (sstratumX + jitterX) / s) / s;
    Xi[1]            = (stratumY + (sstratumY + jitterY) / s) / s;
    for (unsigned j = 2; j < d; ++j) {
        unsigned Aij      = (Ai0 + (j-1) * Ai1) % s;
        unsigned j2       = (j % 2) ? j-1 : j+1;
        unsigned Aij2     = (Ai0 + (j2-1) * Ai1) % s;
        unsigned stratumJ = permute(Aij, s,      p[j] * 0x51633e2d);
        unsigned sstratJ  = offset(Aij, Aij2, s, p[j] * 0x68bc21eb, ot);
        float jitterJ     = randfloat(i,         p[j] * 0x02e5be93);
        Xi[j]             = (stratumJ + (sstratJ + jitterJ) / s) / s;
    }
    return Xi;
}
```

Listing 2: *A modified version of Listing 2 from the main document which computes all dimensions (instead of just one) of an arbitrary sample from a Bush OA pattern.*

```
vector<float> bushOA(unsigned i,      // sample index
                     unsigned d,      // number of dimensions (<= s+1)
                     unsigned s,      // number of levels/strata
                     unsigned t,      // strength of OA (0 < t <= d)
                     unsigned p[],    // array of d pseudo-random seeds
                     OffsetType ot) { // J or MJ
    vector<float> Xi(d);
    unsigned N = pow(s, t);
    i = permute(i, N, p[0]);

    // get digits of ‘i‘ in base ‘s‘
    auto iDigits = toBaseS(i, s, t);

    unsigned stm = N / s;   // pow(s, t-1)
    for (unsigned j = 0; j < s; ++j) {
        unsigned k        = (j % 2) ? j - 1 : j + 1;
        unsigned phi      = evalPoly(iDigits, j);
        unsigned stratum  = permute(phi % s, s, p[j] * 0x51633e2d);
        unsigned sstratum = offset(i, s, stm,   p[j] * 0x68bc21eb, ot);
        float jitter      = randfloat(i,        p[j] * 0x02e5be93);
        Xi[j]             = (stratum + (sstratum + jitter) / stm) / s;
    }
    return Xi;
}
```

Listing 3: *A modified version of Listing 3 from the main document which computes all dimensions (instead of just one) of a CMJ pattern generalized to arbitrary dimensions d.*

```
vector<float> cmjdD(unsigned i,    // sample index
                    unsigned s,    // number of levels/strata
                    unsigned t,    // strength of OA (t=d)
                    unsigned p[] { // array of d pseudo-random seeds
    vector<float> Xi(t);
    unsigned N = pow(s, t);
    i = permute(i, N, p[0]);

    // get digits of ‘i‘ in base ‘s‘
    auto iDigits = toBaseS(i, s, t);

    unsigned stm1 = N / s;  // pow(s, t-1)
    for (unsigned j = 0; j < s; ++j) {
        unsigned stratum  = permute(iDigits[j], s, p[j] * 0x51633e2d);
        auto pDigits      = allButJ(iDigits, j);
        unsigned sstratum = evalPoly(pDigits, s);
        sstratum          = permute(sstrata, stm1, p[j] * 0x68bc21eb);
        float jitter      = randfloat(i,          p[j] * 0x02e5be93);
        Xi[d]             = (stratum + (sstratum + jitter) / stm1) / s;
    }
    return Xi;
}
```



(a) Jittered (b) Multi-Jittered (c) Correlated Multi-Jittered

Figure 1: *All six bivariate projections of a 4D strength-2 Bush-OA with 121, 289 and 1369 samples using jittered, multi-jittered, and correlated multi-jittered offsets.*

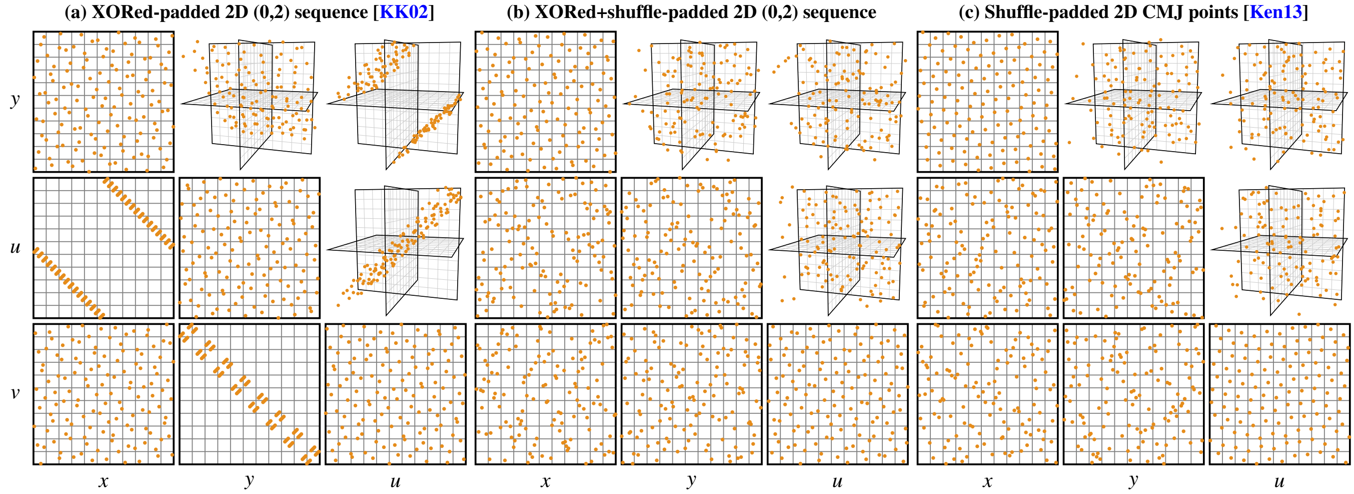


Figure 2: *Alternate version of Fig. 2 from the main document with 3D projections of the 4D pointsets instead of power spectra. A common way to create samples for higher-dimensional integration is to pad together high-quality 2D point sets. Kollig and Keller [KK02] proposed scrambling each dimension of a $(0,2)$ sequence by XORing it with a different random bit vector. While odd-even dimension pairings produce high-quality point sets, even-even (xu) or odd-odd (yv) dimensions are poorly distributed. These issues can be eliminated by additionally randomly shuffling the points across dimension pairs, but this decorrelates all cross dimensions, destroying 2D stratification.*

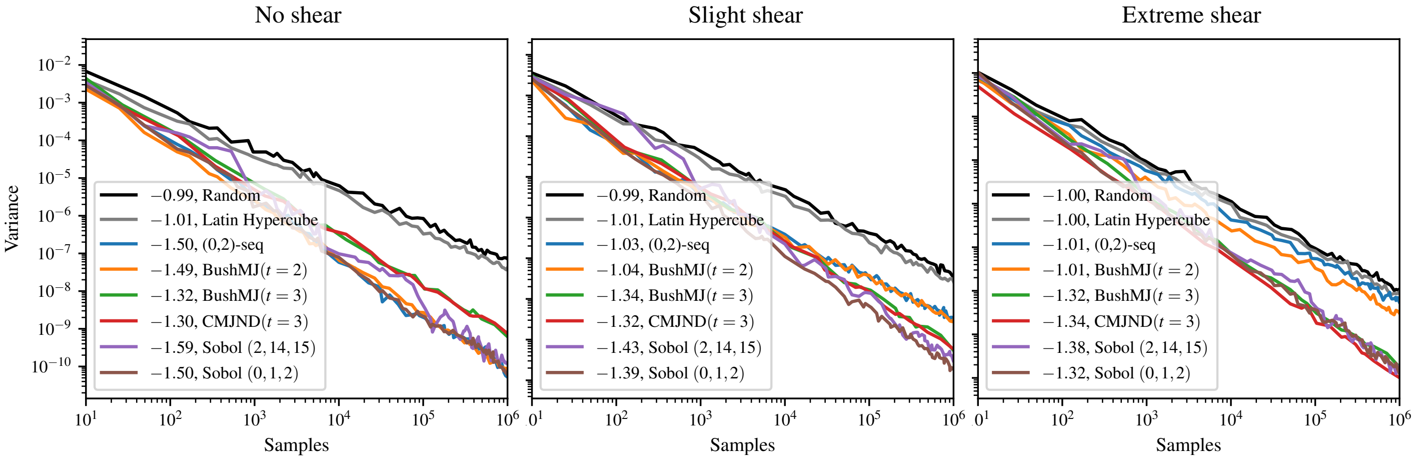


Figure 3: *Variance graphs for a cylinder sheared by different amounts. This integrand is defined by two circles that lie in the $z=0$ and $z=1$ planes with centers $(x_0,y_0,0)$ and $(x_1,y_1,1)$. The analytic integral is defined as $\int_0^1(x,y,z)=\pi r^2\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+1}$ where the radius r is an arbitrary constant. Since an extruded circle with no shear (left) is actually a two-dimensional integrand, samplers that have 2D stratification perform best in this case. Once any shear is applied, however, the integrand is 3D and, asymptotically, 2D stratification degrades to $\mathcal{O}(N^{-1})$ convergence. With only slight shear (middle), 2D stratified samplers initially perform best, but their variance graphs curve and start to perform worse than 3D stratification at higher sampler counts. For larger shear angles the tipping point starts earlier, with 3D stratification dominating even at lower sample counts.*

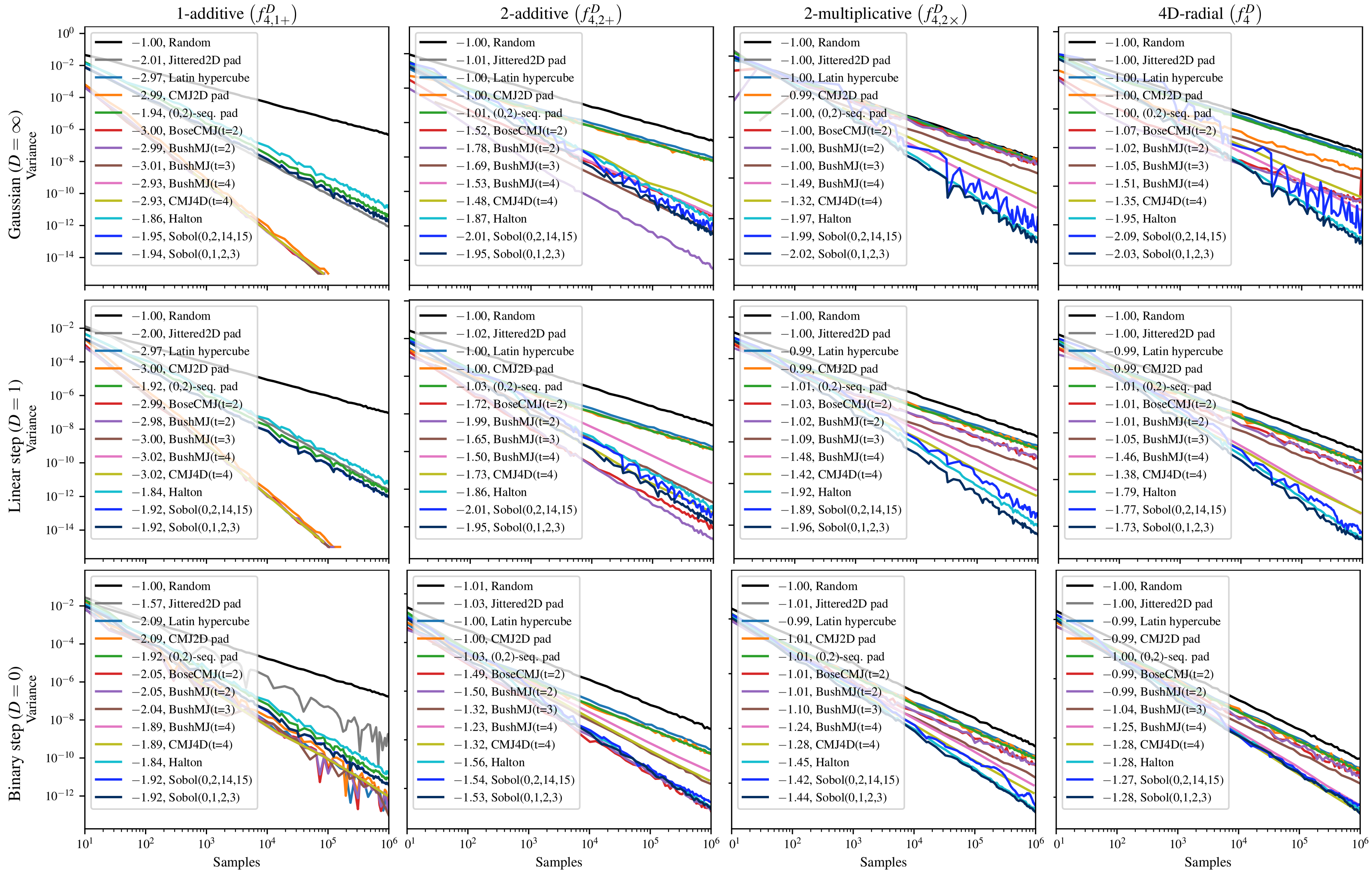


Figure 4: *Fig. 4 from the main paper extended with the Gaussian integrand. Variance behavior of 13 samplers on 4D analytic integrands of different complexity (columns) and continuity (rows). We list the best-fit slope of each technique, which generally matches the theoretically predicted convergences rates (Table 3). Our samplers always perform better than traditional padding approaches, but are asymptotically inferior to high-dimensional QMC sequences for general high-dimensional integrands. When strength $t < d$ (right two columns), convergence degrades to -1, but higher strengths attain lower constant factors.*

References

[AZ16] ATKINS, J. and ZAX, D. B. "Nested orthogonal arrays". *arXiv:1601.06459 [stat]* (Jan. 2016). arXiv: 1601.06459 [stat] 1.

[CQ14] CHEN, J. and QIAN, P. Z. "Latin hypercube designs with controlled correlations and multi-dimensional stratification". *Biometrika* 101.2 (Feb. 2014), 319–332. ISSN: 1464-3510. DOI: 10/f5549t 1.

[CSW94] CHIU, K., SHIRLEY, P., and WANG, C. "Multi-jittered sampling". *Graphics Gems IV*. Ed. by HECKBERT, P. S. San Diego, CA, USA: Academic Press, 1994, 370–374. ISBN: 0-12-336155-9 1.

[HCT18] HE, Y., CHENG, C.-S., and TANG, B. "Strong orthogonal arrays of strength two plus". *The Annals of Statistics* 46.2 (Apr. 2018), 457–468. ISSN: 0090-5364. DOI: 10/gfznb6 1.

[HHQ16] HWANG, Y., HE, X., and QIAN, P. Z. "Sliced orthogonal array-based Latin hypercube designs". *Technometrics* 58.1 (Jan. 2016), 50–61. ISSN: 1537-2723. DOI: 10/f8cjrk 1.

[HQ10] HAALAND, B. and QIAN, P. Z. G. "An approach to constructing nested space-filling designs for multi-fidelity computer experiments". *Statistica Sinica* 20.3 (2010), 1063–1075. ISSN: 1017-0405 1.

[HQ11] HE, X. and QIAN, P. Z. G. "Nested orthogonal array-based Latin hypercube designs". *Biometrika* 98.3 (Sept. 2011), 721–731. ISSN: 0006-3444. DOI: 10/cz9pj6 1.

[HSS99] HEDAYAT, A. S., SLOANE, N. J. A., and STUFKEN, J. *Orthogonal Arrays: Theory and Applications*. Springer-Verlag, 1999. ISBN: 978-1-4612-7158-1. DOI: bqh3jj 1.

[HT12] HE, Y. and TANG, B. "Strong orthogonal arrays and associated Latin hypercubes for computer experiments". *Biometrika* 100.1 (Dec. 2012), 254–260. ISSN: 1464-3510. DOI: 10/gfznb5 1.

[HT14] HE, Y. and TANG, B. "A characterization of strong orthogonal arrays of strength three". *The Annals of Statistics* 42.4 (2014), 1347–1360. ISSN: 00905364. DOI: 10/gfznb4 1.

[Ken13] KENSLER, A. *Correlated Multi-Jittered Sampling*. Tech. rep. 13-01. Pixar Animation Studios, Mar. 2013 1, 4.

[KK02] KOLLIG, T. and KELLER, A. "Efficient multidimensional sampling". *Computer Graphics Forum (Proceedings of Eurographics)* 21.3 (Sept. 2002), 557–563. ISSN: 0167-7055. DOI: 10/d2stpx 4.

[LL15] LIU, H. and LIU, M.-Q. "Column-orthogonal strong orthogonal arrays and sliced strong orthogonal arrays". *Statistica Sinica* (2015). ISSN: 1017-0405. DOI: 10/f7z56h 1.

[Owe13] OWEN, A. B. *Monte Carlo Theory, Methods and Examples*. To be published, 2013. URL: https://statweb.stanford.edu/~owen/mc/ (visited on 06/07/2019) 1.

[Owe94] OWEN, A. B. "Controlling correlations in Latin hypercube samples". *Journal of the American Statistical Association* 89.428 (Dec. 1994), 1517–1522. ISSN: 1537-274X. DOI: 10/gfzncf 1.

[QA10] QIAN, P. Z. G. and AI, M. "Nested lattice sampling: a new sampling scheme derived by randomizing nested orthogonal arrays". *Journal of the American Statistical Association* 105.491 (2010), 1147–1155. ISSN: 01621459. DOI: 10/fwjp8h 1.

[QAW09] QIAN, P. Z. G., AI, M., and WU, C. F. J. "Construction of nested space-filling designs". *The Annals of Statistics* 37.6A (Dec. 2009), 3616–3643. ISSN: 0090-5364, 2168-8966. DOI: 10/fr79bb 1.

[Qia09] QIAN, P. Z. G. "Nested Latin hypercube designs". *Biometrika* 96.4 (2009), 957–970. ISSN: 0006-3444. DOI: 10/cpmd77 1.

[RHVD10] RENNEN, G., HUSSLAGE, B., VAN DAM, E. R., and DEN HERTOG, D. "Nested maximin Latin hypercube designs". *Structural and Multidisciplinary Optimization* 41.3 (Apr. 2010), 371–395. ISSN: 1615-1488. DOI: 10/cpqtq3 1.

[SLQ14] SUN, F., LIU, M.-Q., and QIAN, P. Z. G. "On the construction of nested space-filling designs". *The Annals of Statistics* 42.4 (2014), 1394–1425. ISSN: 0090-5364. DOI: 10/gf3nmd 1.

[Tan93] TANG, B. "Orthogonal array-based Latin hypercubes". *Journal of the American Statistical Association* 88.424 (Dec. 1993), 1392–1397. ISSN: 1537-274X. DOI: 10/gfzncp 1.

[Tan98] TANG, B. "Selecting Latin hypercubes using correlation criteria". *Statistica Sinica* 8.3 (1998), 965–977. ISSN: 10170405, 19968507 1.

[Wen14] WENG, J. "Maximin Strong Orthogonal Arrays". M.Sc. Thesis. Simon Fraser University, 2014 1.

[WL13] WANG, K. and LI, Y. "Constructions of nested orthogonal arrays". *Journal of Combinatorial Designs* 21.10 (2013), 464–477. ISSN: 1520-6610. DOI: 10/gf2zf9 1.

[YLL14] YIN, Y., LIN, D. K., and LIU, M.-Q. "Sliced Latin hypercube designs via orthogonal arrays". *Journal of Statistical Planning and Inference* 149 (June 2014), 162–171. ISSN: 0378-3758. DOI: 10/gfzncs 1.

[ZWD19] ZHANG, T.-F., WU, G., and DEY, A. "Construction of some new families of nested orthogonal arrays". *Communications in Statistics - Theory and Methods* 48.3 (Feb. 2019), 774–779. ISSN: 0361-0926. DOI: 10/gf2zf4 1.