WAIT: Workshop on the Advances in Information Technology, Budapest, 2026

Fudging fudge dice

Péter Salvi

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Műegyetem rkp. 3., H-1111 Budapest, Hungary

Abstract

Dice with unequal probabilities are relatively rare, and their construction poses many interesting questions. Here we try to design a die approximating the result of a d4F roll.

1. Introduction

> "**fudge**, v. t. : to devise as a substitute"
>
> *Merriam–Webster Dictionary*[1]

Table-top role-playing games are well-known for using many dice of various types, usually denoted by *dN*, where *N* is the number of faces. Most common are *d4*, *d6*, *d8*, *d10*, *d12*, and *d20*, but even *d100* dice are used in some systems.[5] With the exception of *d10*, all the other common dice are Platonic solids—their regularity ensures uniform probabilities. The *d10* die is usually constructed as a pentagonal trapezohedron.

An interesting question is how to construct *d3* dice; the most common version uses curved, unstable connecting surfaces between the three planes. Another solution is to just use a regular die (i.e., a *d6*), and set the same labels to opposite faces. This is the approach taken in *fudge dice*, originally devised for the universal Fudge system,[4] but nowadays associated with the Fate Core system[2] and its descendants (Fate Accelerated, Fate Condensed etc.), and thus also called *fate dice*. The markings of the faces are: empty, $-$ and $+$, see Figure 1.

The Fate system uses a single type of roll, the *d4F*, i.e., four fudge dice, and computes the sum (interpreting the markings as 0, -1 and 1). The result is a discrete random variable in the $[-4,4]$ interval with the distribution shown in Table 1.

Figure 1: *Fudge dice.*

Value	Probability	Percentage
-4	1/81	1.23%
-3	4/81	4.94%
-2	10/81	12.35%
-1	16/81	19.75%
0	19/81	23.46%
1	16/81	19.75%
2	10/81	12.35%
3	4/81	4.94%
4	1/81	1.23%

Table 1: *The d4F distribution.*

When fudge dice are not available, regular dice are an easy substitute: just draw the diagonal over the dots on ⚁ and ⚂, thereby creating a $-$ sign; draw a square over the dots of ⚃ and ⚅, similar to a 0; and draw both diagonals over the dots of ⚀ and ⚄, creating a $+$ sign.[†]

[†] https://2d4chan.org/wiki/Fudge_dice

Figure 2: *Captain Toad and Toadette with a super gem.*

If we do not need to reconstruct the exact distribution, there are different options. Instead of a roll we can flip 10 coins, count the heads, and subtract 5, clamping the result into $[-4,4]$. This gives a clamped, centered binomial distribution that closely approximates a *d4F* roll. Its expected χ^2 value is $3.39 \cdot 10^{-3} \cdot N$, where N is the number of experiments. The degree of freedom being 8, the critical value is 15.507 (using 0.05 significance). It can be shown that with 1000 "rolls" a chi-squared test will not spot the difference in more than three-fourths of the cases.

But some people are in a hurry, and do not want to count—for them, the ideal solution would be a single die with the same distribution. Designing a non-uniformly distributed die, however, can be a challenge, as one should take into account various factors. The following section introduces our model and the computations behind it, while Section 3 shows some test results.

2. Modeling the die

Considering the characteristics of the distribution (2 pairs of unlikely results, 2 pairs of probable results, and a single most likely value), we decided to use a convex polyhedron that can be assembled from a frustum of a pyramid (top) and an inverted pyramid (bottom), a shape somewhat similar to the *super gems* in Captain Toad: Treasure Tracker (see Fig. 2). The design itself depends on four parameters (see Fig. 3):

Parameter	Min	Max	Description
a	1	2	aspect ratio
b	0.3	0.9	top shrinkage
c	0.1	1	top height
d	0.5	4	bottom depth

The model can be described by its vertices:

- Center ($v_1 \dots v_4$):
  $(-a,0,-1)$, $(a,0,-1)$, $(a,0,1)$, $(-a,0,1)$
- Top ($v_5 \dots v_8$):
  $(-ab,c,-b)$, $(ab,c,-b)$, (ab,c,b), $(-ab,c,b)$

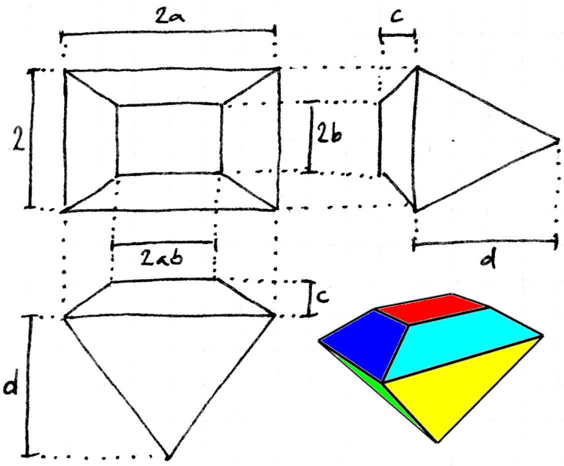

Figure 3: *Die design. The blue—cyan—green—yellow—red colors show increasing probability of landing on that face.*

- Apex (v_9):
  $(0,-d,0)$

... and its faces:

- Upper (f_1 and f_3: cyan; f_2 and f_4: blue):
  $[v_5,v_6,v_2,v_1]$, $[v_6,v_7,v_3,v_2]$, $[v_7,v_8,v_4,v_3]$, $[v_8,v_5,v_1,v_4]$
- Top (f_5: red):
  $[v_8,v_7,v_6,v_5]$
- Lower (f_6 and f_8: yellow, f_7 and f_9: green):
  $[v_9,v_1,v_2]$, $[v_9,v_2,v_3]$, $[v_9,v_3,v_4]$, $[v_9,v_4,v_1]$

Here f_5 is associated with the most probable value (19/81), the surrounding faces ($f_1 \dots f_4$) with the low probabilities (1/81 and 4/81), and the faces on the bottom part ($f_6 \dots f_9$) with the more probable ones (10/81 and 16/81). Note that when we roll the die, f_5 cannot be on top, so that face can be used for adding a logo or other design elements. The faces with lowest probability (f_2 and f_4) should be marked with ± 2, since these will be on top when the die lands on a face with (approximately) 10/81 probability. Similarly, e.g., the larger faces on the bottom (f_6 and f_8) should be marked with ± 3, etc. The roll of 0 is indicated by the apex being on top.

Now that the design is fixed, we need to find parameters that optimize the deviation from the target distribution. First we need to make sure that the center of mass, projected to the plane of each face, is always *inside* the polygons, otherwise the model would not be stable. Due to the symmetric construction, the center of mass will be on the y axis, and its y-coordinate is

$$\mathrm{COM}_y = \frac{c^2(3b^2+2b+1)-d^2}{4(c(b^2+b+1)+d)}. \qquad (1)$$

This assumes that the model is solid; if it is just a shell, the center of mass is the area-weighted mean of the face centroids, which leads to a somewhat more complex expression.

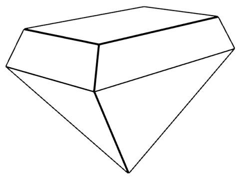

Figure 4: *A typical model based on solid angle optimization ($a \approx 1.61$, $b \approx 0.83$, $c \approx 0.56$, $d \approx 2.04$).*

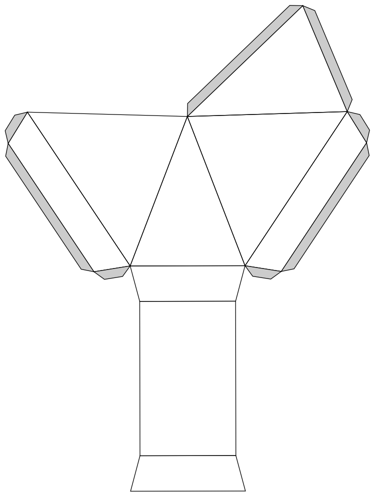

Figure 5: *Polyhedron net of the die in Figure 4.*

Our experiments showed that there is no substantial difference between the two in the interesting cases.

A good approximation of the relative probabilities of the die landing on each face comes from the *solid angle* seen from the center of mass. The objective function is the squared error weighted by the inverse probability:

$$\sum_{k=-4}^{4} \frac{(q_k - p_k)^2}{p_k}, \tag{2}$$

where p_k is the target probability of the value k, while q_k is the probability computed by dividing the solid angle by the sum of all angles (i.e., 4π sr). The optimization problem can be solved by some derivative-free method, such as the Nelder–Mead algorithm.[3] A typical result is shown in Figure 4. This, in theory, slightly favors the least probable faces, but otherwise the approximation is quite good, with the center of mass within the stable range.

Unfortunately, reality is not so simple. Performing experiments with a paper model (see Fig. 5), it turned out that the object, while stable in theory, can hardly stand on its smallest face, and actual probabilities also largely diverge from what was expected based on its solid angles. Probable causes include friction, as well as the complex dynamics of the bouncing object.

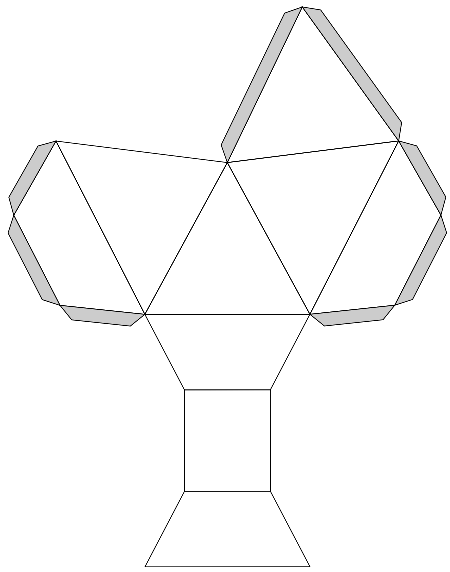

Figure 6: *Polyhedron net of the final 9-sided die.*

Based on these observations, we decided to determine the parameters by simulation, using a rigid body physics engine like OPEN DYNAMICS ENGINE[‡] or the BULLET PHYSICS SDK.[§] In this case we substitute q_k in Eq. (2) with $\hat{q}_k$, the empirical frequency of an event based on a large number of simulations.

In each simulation, the model is placed at a random position roughly 5 units above the ground, with random orientation and random initial angular velocity. Restitution is set to 0.3 (bouncy) and friction to 0.8 (moderate). When the object settles on the ground, the simulation ends and the face lying on the ground is determined. Each evaluation of the objective function makes 1000 simulations.

3. Results

After thorough testing, we arrived at our final model with the following parameters: $a = 1.18$, $b = 0.52$, $c = 0.72$, and $d = 1.41$. A sample of 30000 simulations shows that it slightly favors ± 1 compared to the target distribution. Actual rolls were also tested using another paper model (see Fig. 6), which resulted in the counts shown in Table 2. It was

‡ https://www.ode.org/

§ https://github.com/bulletphysics/bullet3

Value	Count	Target count	Relative error (%)
-4	0	5	-100
-3	20	20	0
-2	48	50	-4
-1	80	80	0
0	89	95	-6
1	83	80	4
2	66	50	32
3	17	20	-15
4	2	5	-60

Table 2: *Distribution of a sample of 405 rolls with the final model.*

not a particularly lucky series: extremal values (± 4) were severely underrepresented, while there was a large excess of 2s. Still, with a χ^2 value of 12.97 and a p-value of 0.11, it is not an unlikely sample even from real fudge dice.

We can also look at the total variation distance (TVD)

$$\frac{1}{2}\sum_{k=-4}^{4}|p_k - \hat{q}_k| \approx 0.047. \quad (3)$$

This is a very good match, especially if we take into account that the asymptotic expected value of this metric, assuming that P and $\hat{Q}$ have the same distribution, is around $\sqrt{\frac{K-1}{2\pi N}} = \sqrt{\frac{8}{2\pi \cdot 405}} \approx 0.056$, where K is the number of categories.

The Jensen–Shannon divergence (JSD)

$$\frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(\hat{Q}\|M) \approx 0.0068 \quad (4)$$

is another way to quantify the goodness of match. Here $M = \frac{1}{2}(P + \hat{Q})$ is the midpoint distribution, and

$$D_{KL}(P\|M) = \sum_{k=-4}^{4} p_i \ln \frac{p_i}{m_i} \quad (5)$$

is the Kullback–Leibler divergence from M to P. The result is between 2 and 3 times the expected value, which is around $\frac{K-1}{8N} = \frac{8}{8\cdot 405} \approx 0.0025$. This is generally considered a good match, although not an excellent one.

Conclusion

We have shown how to construct dice with uneven probabilities. In the process we have seen that there is a large discrepancy between the purely theoretical, idealized model, and real observations, which led us to base our model on simulated rigid body physics instead.

Figure 7: *The paper model used in the experiments.*

We have also conducted an experiment of 405 rolls, which can be considered a medium-sized sample. The results demonstrate a high degree of fidelity to the target distribution. Statistical analysis indicates that the observed deviations are consistent with sampling noise (χ^2, $p \approx 0.11$), with a negligible absolute probability gap (TVD $\approx$ 0.047) and strong structural alignment in information density (JSD $\approx$ 0.0068).

The model could be further enhanced by, for example, changing the *ab* side lengths into an independent *e* parameter. A 3D printed model and automated real-life testing would also be interesting.

Acknowledgements

The physical simulations were created by Anthropic Claude Sonnet 4.5 (Bullet Physics version) and Grok 4 (ODE version), while the solid angle based experiments were created by Gemini 3. I would also like to acknowledge the help of discussions with Gemini 3, ChatGPT 5.2 and Microsoft Copilot. A special thanks goes to my wife, who put up with my late-night dice-rolling.

References

1. *Merriam–Webster's Collegiate Dictionary*. Merriam–Webster, Inc., Springfield, MA, 10th edition, 1993.
2. *Fate Core System*. Evil Hat Productions, LLC., Silver Spring, MD, 2013.
3. Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. MIT Press, 2019.
4. Steffan O'Sullivan. *Fudge*. Grey Ghost Press, Inc., 10th anniversary edition, 2005.
5. Sandy Petersen and Lynn Willis. *Call of Cthulhu*. Chaosium, Inc., 5.6 edition, 1999.