

1

# Hit and Run and Stuff

2

Brian Cohn            May Szedlák

3

March 10, 2015

4

**Abstract**

5

# 1 Author Summary

## 2 Introduction

## 3 Results

## 4 Discussion

## 5 Materials

### 5.1 Hit-and-Run

The Hit-and-Run algorithm used for sampling in a convex body  $K$ , was introduced by Smith in 1984 [6]. The mixing time is known to be  $\mathcal{O}^*(n^2 R^2 / r^2)$ , where  $R$  and  $r$  are the radii of the inscribed and circumscribed ball of  $K$  respectively [1, 4]. In the case of the muscles of a limb, we are interested in the polygon  $P$  that is given by the set of all possible activations  $\mathbf{a} \in \mathbb{R}^n$  that satisfy

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0, 1]^n,$$

where  $\mathbf{f} \in \mathbb{R}^m$  is a fixed force vector and  $A = J^{-T} R F_m \in \mathbb{R}^{m \times n}$ .  $P$  is bounded by the unit  $n$ -cube since all variables  $a_i$ ,  $i \in [n]$  are bounded by 0 and 1 from below, above respectively.

Consider the following  $1 \times 3$  example.

$$1 = \frac{10}{3}a_1 - \frac{53}{15}a_2 + 2a_3$$

$$a_1, a_2, a_3 \in [0, 1],$$

the set of feasible activations is given by the shaded set in Figure 1.

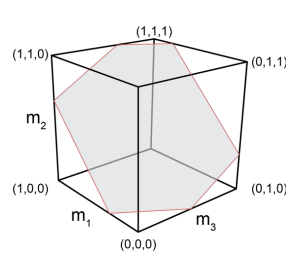


Figure 1: Feasible Activation

The Hit-and-Run walk on  $P$  is defined as follows (it works analogously for any convex body).

1. Find a given starting point  $\mathbf{p}$  of  $P$  (Figure 2) .

- 24 2. Generate a random direction through  $\mathbf{p}$  (uniformly at random over all directions)
- 25 (Figure 3).
- 26 3. Choose the next point of the sampling algorithm uniformly at random from the
- 27 segment of the line in  $P$  (Figure 4).
- 28 4. Repeat from (b) the above steps with the new point as the starting point (Figure
- 29 5).

30 The implementation of this algorithm is straight forward except for the choice of the  
 31 random direction. How do we sample uniformly at random (u.a.r.) from all directions  
 32 in  $P$ ? Suppose that  $\mathbf{q}$  is a direction in  $P$  and  $p \in P$ . Then by definition of  $P$ ,  $\mathbf{q}$  must  
 33 satisfy  $\mathbf{f} = A(\mathbf{p} + \mathbf{q})$ . Since  $\mathbf{p} \in P$ , we know that  $\mathbf{f} = A\mathbf{p}$  and therefore

$$\mathbf{f} = A(\mathbf{p} + \mathbf{q}) = \mathbf{f} + A\mathbf{q}$$

34 and hence

$$A\mathbf{q} = 0.$$

35 We therefore need to choose directions uniformly at random from all directions in  
 36 the vectorspace

$$V = \{\mathbf{q} \in \mathbb{R}^n | A\mathbf{q} = 0\}.$$

37 As shown by Marsaglia this can be done as follows [5].

- 38 1. Find an orthonormal basis  $b_1, \dots, b_r \in \mathbb{R}^n$  of  $A\mathbf{q} = 0$ .
- 39 2. Choose  $(\lambda_1, \dots, \lambda_r) \in \mathcal{N}(0, 1)^n$  (from the Gaussian distribution).
- 40 3.  $\sum_{i=1}^r \lambda_i b_i$  is a u.a.r. direction.

41 A basis of a vectorspace  $V$  is a minimal set of vectors that generate  $V$ , and it is  
 42 orthonormal if the vectors are pairwise orthogonal (perpendicular) and have unit length.  
 43 Using basic linear algebra one can find a basis for  $V = \{A\mathbf{q} = 0\}$  and orthogonalize it  
 44 with the well known Gram-Schmidt method (for details see e.g. [2]). Note that in order  
 45 to get the desired u.a.r. distribution the basis needs to be orthonormal. For the limb  
 46 case we can safely assume that the rows of  $A$  are linearly independent and hence the  
 47 number of basis vectors is  $n - m$ .

## 48 5.2 Stopping Time

49 In this section we discuss the stopping time of the Hit and Run algorithm. How many  
 50 steps are necessary to reach an approximate uniform distribution? The theoretical bound  
 51  $\mathcal{O}^*(n^2 R^2 / r^2)$  given in [4] has a very large hidden coefficient which makes the algorithm  
 52 almost infeasible in lower dimensions.

53 These bounds hold for general convex sets. For convex polygons, as in our case, Ge  
 54 et al. showed experimentally that up to about 40 dimensions, 10 million random points

55 suffice to get a close to uniform discussion [3]. For our case we generate 10 million points  
 56 and also test whether the mean of each coordinate converges and whether the upper and  
 57 lower bounds for each coordinate are met. In detail for the mean we see that it converges  
 58 after ?? steps. For the upper and lower bounds of the activation we can solve two linear  
 59 program for each coordinate of  $\mathbf{a}$  to find the upper and lower bounds of each  $a_i$ . We see  
 60 that those theoretical bounds match the experimentally obtained bounds.

## 61 6 Acknowledgments

## 62 References

- 63 [1] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for ap-  
 64 proximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium*  
 65 *of Theory of Computing*, pages 375–381, 1989.
- 66 [2] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.
- 67 [3] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of  
 68 convex polytopes. *Preprint: arXiv:1401.0120*, 2013.
- 69 [4] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.
- 70 [5] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*,  
 71 43:645–646, 1972.
- 72 [6] R.L. Smith. Efficient monte-carlo procedures for generating points uniformly dis-  
 73 tributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

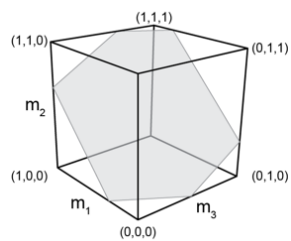


Figure 2: Inner Point

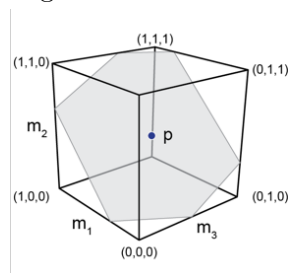


Figure 3: Direction

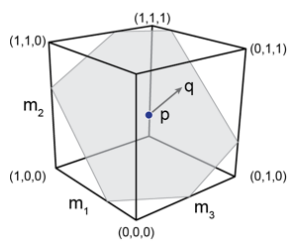


Figure 4: Endpoints

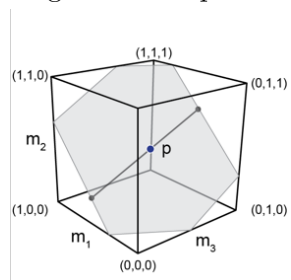


Figure 5: New Point

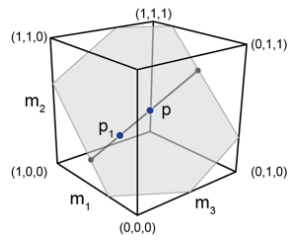


Figure 6: Uniform Distribution

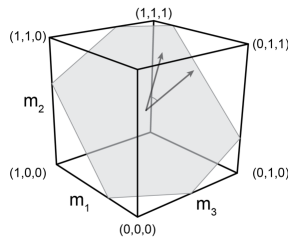


Figure 7: Some Basis

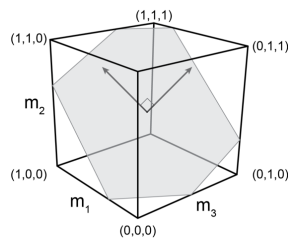


Figure 8: Direction