

数値計算：確率的アルゴリズム

平井 慎一

立命館大学 ロボティクス学科

講義の流れ

- 1 乱数
- 2 モンテカルロ法
- 3 まとめ

乱数

コンピュータ上でサイコロを模擬する.

$$D = \begin{cases} 1 & X \in [0, 1/6) \\ 2 & X \in [1/6, 2/6) \\ 3 & X \in [2/6, 3/6) \\ 4 & X \in [3/6, 4/6) \\ 5 & X \in [4/6, 5/6) \\ 6 & X \in [5/6, 1) \end{cases}$$

X は区間 $(0, 1)$ 内の一様乱数

$$X \sim U(0, 1)$$

uniform.m

関数 rand 区間 $(0, 1)$ 内の一様乱数

```
for i=1:10
    x = rand;
    s = num2str(x);
    disp(s);
end
```

uniform.m

```
>> uniform
0.81472
0.90579
0.12699
0.91338
0.63236
0.09754
0.2785
0.54688
0.95751
0.96489
>> uniform
0.15761
0.97059
0.95717
0.48538
```

dice.m

```
function k = dice()
% simulating a dice
x = rand;
if x < 1/6.00      k = 1;
elseif x < 2/6.00 k = 2;
elseif x < 3/6.00 k = 3;
elseif x < 4/6.00 k = 4;
elseif x < 5/6.00 k = 5;
else              k = 6;
end
end
```

dice_run.m

```
for i=1:10
    s = [];
    for j=1:10
        k = dice();
        s = [s, ' ', num2str(k)];
    end
    disp(s);
end
```

dice_run.m

```
>> dice_run
2 4 6 5 6 3 3 4 2 5
4 4 2 1 3 3 5 5 5 1
1 4 2 6 6 2 5 6 4 6
6 4 5 4 1 3 4 4 3 6
5 6 3 4 6 6 5 2 4 1
3 5 6 5 3 5 3 6 6 6
3 3 2 5 6 6 4 4 1 6
3 2 6 5 6 2 5 4 1 3
2 5 2 6 5 3 3 5 6 4
4 2 3 5 6 5 1 5 3 3
>> dice_run
1 5 2 2 3 3 4 4 3 3
4 4 6 5 3 5 1 1 1 1
2 2 1 4 1 1 4 6 6 4
6 4 4 2 3 3 1 6 1 3
```

dice_validation.m

```
n = 12000;

for k=1:n
    a(k) = dice();
end

histogram(a);
fprintf("サイコロの目の頻度\n");
fprintf("一時停止：何かのキーを押してください\n");
pause;

d = a(2:n) - a(1:n-1);
histogram(d);
fprintf("差の頻度\n");
fprintf("一時停止：何かのキーを押してください\n");
pause;
```

一様乱数 $U(a, b)$

$U(a, b)$

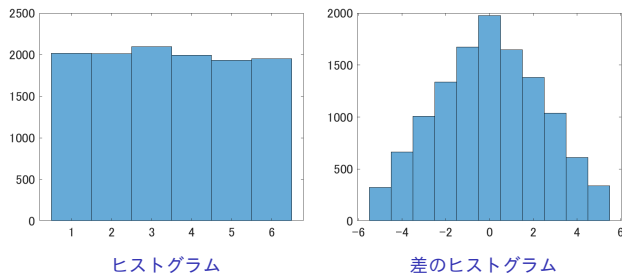
区間 (a, b) 内の一様乱数

$$X \sim U(0, 1)$$

$$Y \sim U(a, b)$$

$$Y = (b - a)X + a$$

dice_validation.m



正規乱数 $N(0, 1)$

$N(0, 1)$

平均0, 標準偏差1 (分散 $1^2 = 1$) の正規分布に従う乱数

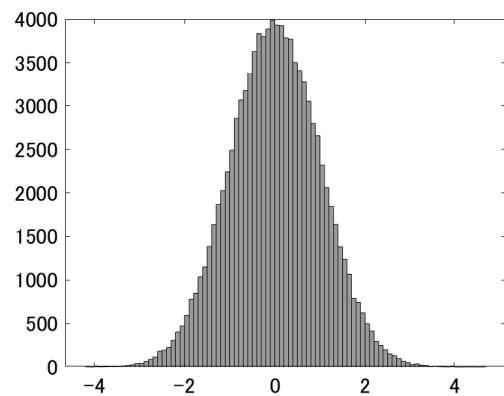
ファイル normal_histogram.m

```
a = randn(100000,1);
histogram(a);
saveas(gcf, 'normal_histogram.eps', 'eps');
```

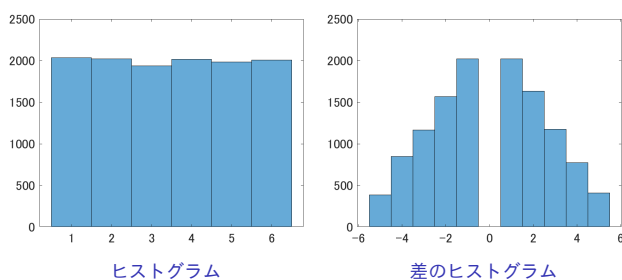
dice_false_run.m

```
>> dice_false_run
3 2 5 3 4 5 1 5 6 5
2 5 3 6 4 3 4 3 5 6
3 6 3 4 2 1 2 4 3 4
6 4 2 5 4 3 4 2 3 1
2 6 3 2 5 6 1 3 1 6
5 6 5 2 4 5 4 5 3 5
2 4 1 2 1 4 6 3 4 3
5 6 5 1 6 1 4 6 4 6
2 6 2 1 4 2 4 3 4 3
4 3 5 3 5 1 2 1 2 6
>> dice_false_run
4 1 6 4 2 3 6 5 2 1
6 2 3 1 4 6 5 3 2 3
5 1 3 1 2 6 4 2 1 4
1 4 3 4 5 2 3 4 1 6
```

normal_histogram.m



dice_false_validation.m



正規乱数 $N(\mu, \sigma)$

$N(\mu, \sigma)$

平均 μ , 標準偏差 σ (分散 σ^2) の正規分布に従う乱数

$$X \sim N(0, 1)$$

$$Y \sim N(\mu, \sigma)$$

$$Y = \sigma X + \mu$$

ノイズのシミュレーション

観測信号

$$x(t) = A \sin 2\pi ft + \varepsilon(t)$$

源信号 ノイズ

源信号のパラメータ $A = 2.0$, $f = 5$
ノイズの特性

$$\varepsilon(t) \sim U(-\alpha, \alpha)$$

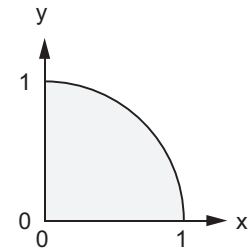
ノイズのパラメータ $\alpha = 1.2$
サンプリング時間 $T = 0.001$

定積分の計算

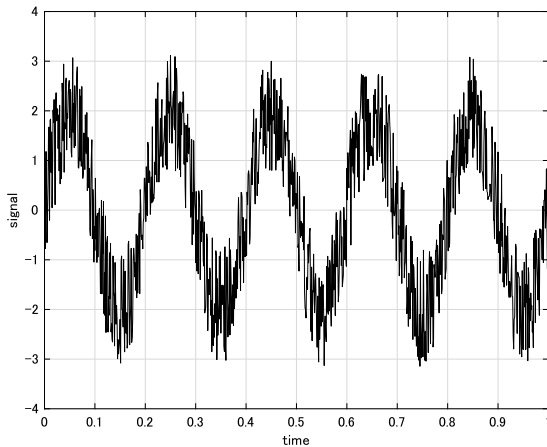
定積分

$$S_1 = \int_0^1 \sqrt{1-x^2} dx$$

被積分関数 $y = \sqrt{1-x^2}$



ノイズのシミュレーション

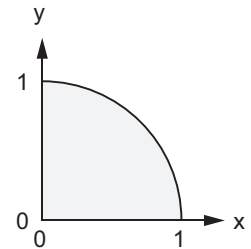
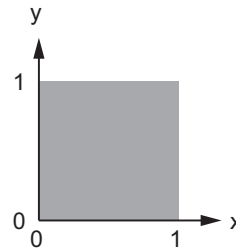


定積分の計算

$x \in [0, 1]$, $y \in [0, 1]$

点 (x, y) を領域 $0 \leq x, y \leq 1$ の中でランダムに発生させる。

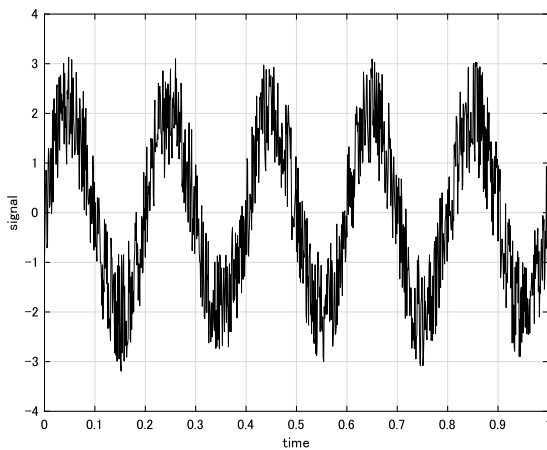
$$x \sim U(0, 1), \quad y \sim U(0, 1)$$



N : 発生させた点の個数

M : $y \leq \sqrt{1-x^2}$ を満たす点の個数

ノイズのシミュレーション



定積分の計算

$$M : N = S_1 : 1$$

$$\Downarrow$$
$$S_1 = \frac{M}{N}$$

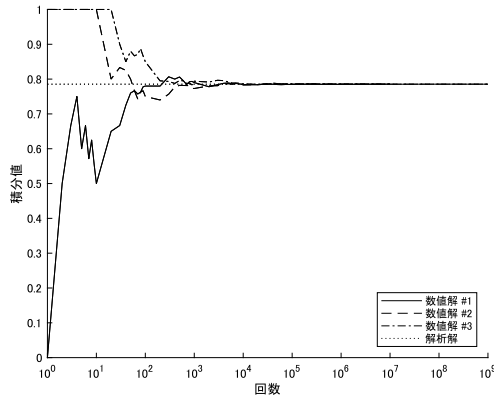
モンテカルロ法

計算に時間や手間を要する問題を、乱数を用いて近似的に解く手法

integral_S1.m

```
N = 100000000; M = 0; k = 1; step = 10;
for i=1:N
    x = rand;    y = rand;
    if x*x + y*y <= 1
        M = M+1;
    end
    if rem(i,step) == 0
        s = [num2str(i), ' ', num2str(M/i)];
        disp(s); k = k+1;
    end
    if k == 10
        step = step * 10; k = 1;
    end
end
end
```

定積分の計算



多重定積分の計算

多重定積分

$$S_n = \int \int \cdots \int_{D_n} \sqrt{1 - x_1^2 - x_2^2 - \cdots - x_n^2} dx_1 dx_2 \cdots dx_n$$

積分領域

$$D_n = \{(x_1, x_2, \dots, x_n) \mid x_1^2 + x_2^2 + \cdots + x_n^2 \leq 1, x_1, x_2, \dots, x_n \geq 0\}$$

被積分関数

$$y = \sqrt{1 - x_1^2 - x_2^2 - \cdots - x_n^2}$$

多重定積分の計算

$x_1, x_2, \dots, x_n \in [0, 1], y \in [0, 1]$
 点 $(x_1, x_2, \dots, x_n, y)$ をランダムに発生させる。
 $(0 \leq x_1, x_2, \dots, x_n, y \leq 1)$

$$x_1 \sim U(0, 1), x_2 \sim U(0, 1), \dots, x_n \sim U(0, 1), y \sim U(0, 1)$$

N : 発生させた点の個数

M : $y \leq \sqrt{1 - x_1^2 - x_2^2 - \cdots - x_n^2}$ を満たす点の個数

多重定積分の計算

$$M : N = S_n : 1$$

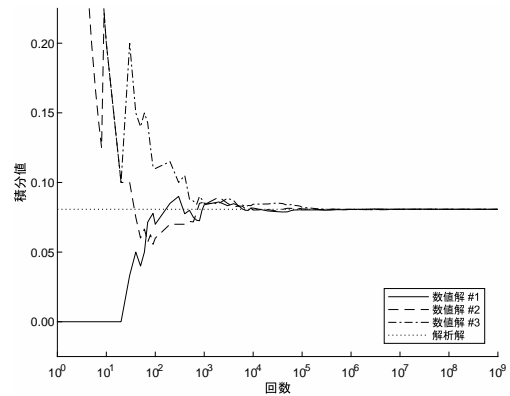
↓

$$S_n = \frac{M}{N}$$

integral_S5.m

```
N = 1000000000; M = 0; k = 1; step = 10;
for i=1:N
    x1 = rand; x2 = rand; x3 = rand; x4 = rand; x5 = rand
    y = rand;
    if x1*x1 + x2*x2 + x3*x3 + x4*x4 + x5*x5 + y*y <= 1
        M = M+1;
    end
    if rem(i,step) == 0
        s = [num2str(i), ' ', num2str(M/i)];
        disp(s); k = k+1;
    end
    if k == 10
        step = step * 10; k = 1;
    end
end
```

多重定積分 S_5 の計算



同じ誕生日

問題

n 人のグループで、誕生日が同じ人たちがいる確率を求める。

問題を簡単にするため、2月29日は考えない。
 誕生日の確率分布は、一様であると仮定する。

同じ誕生日

問題

n 人のグループで、誕生日が同じ人たちがいる確率を求める。

問題を簡単にするため、2月29日は考えない。
 誕生日の確率分布は、一様であると仮定する。

モンテカルロシミュレーションで解く。

- 1月1日, 1月2日, 1月3日, ..., 12月31日に, 1, 2, 3, ..., 365 を対応させる。
- 一様に値 1, 2, 3, ..., 365 を取る乱数を, n 個発生させる。
- n 個の値の中に同じ値があるか否かを調べる。
- 以上の試行を複数回繰返し, 同じ値がある頻度を数える。

same_birthday_simulation.m

```

n = 23; same = 0; diff = 0;
for iteration=1:1000
    birthday = ceil(365*rand(1,n));
    found = 0;
    for k=1:n
        if ismember(birthday(k), birthday(k+1:n))
            found = 1;
            break;
        end
    end
    if found
        same = same + 1;
    else
        diff = diff + 1;
    end
end

```

同じ誕生日

$n = 5$ で、頻度を 10 回数えた結果の例

5 人	同じ人がいる	24 回	すべて異なる	976 回
5 人	同じ人がいる	21 回	すべて異なる	979 回
5 人	同じ人がいる	28 回	すべて異なる	972 回
5 人	同じ人がいる	39 回	すべて異なる	961 回
5 人	同じ人がいる	26 回	すべて異なる	974 回
5 人	同じ人がいる	24 回	すべて異なる	976 回
5 人	同じ人がいる	24 回	すべて異なる	976 回
5 人	同じ人がいる	27 回	すべて異なる	973 回
5 人	同じ人がいる	29 回	すべて異なる	971 回
5 人	同じ人がいる	33 回	すべて異なる	967 回

同じ誕生日

$n = 23$ で、頻度を 10 回数えた結果の例

23 人	同じ人がいる	493 回	すべて異なる	507 回
23 人	同じ人がいる	496 回	すべて異なる	504 回
23 人	同じ人がいる	511 回	すべて異なる	489 回
23 人	同じ人がいる	514 回	すべて異なる	486 回
23 人	同じ人がいる	476 回	すべて異なる	524 回
23 人	同じ人がいる	507 回	すべて異なる	493 回
23 人	同じ人がいる	530 回	すべて異なる	470 回
23 人	同じ人がいる	519 回	すべて異なる	481 回
23 人	同じ人がいる	501 回	すべて異なる	499 回
23 人	同じ人がいる	531 回	すべて異なる	469 回

同じ誕生日

$n = 30$ で、頻度を 10 回数えた結果の例

30 人	同じ人がいる	708 回	すべて異なる	292 回
30 人	同じ人がいる	684 回	すべて異なる	316 回
30 人	同じ人がいる	717 回	すべて異なる	283 回
30 人	同じ人がいる	712 回	すべて異なる	288 回
30 人	同じ人がいる	713 回	すべて異なる	287 回
30 人	同じ人がいる	708 回	すべて異なる	292 回
30 人	同じ人がいる	701 回	すべて異なる	299 回
30 人	同じ人がいる	690 回	すべて異なる	310 回
30 人	同じ人がいる	716 回	すべて異なる	284 回
30 人	同じ人がいる	724 回	すべて異なる	276 回

同じ誕生日

$n = 50$ で、頻度を 10 回数えた結果の例

50 人	同じ人がいる	974 回	すべて異なる	26 回
50 人	同じ人がいる	968 回	すべて異なる	32 回
50 人	同じ人がいる	972 回	すべて異なる	28 回
50 人	同じ人がいる	964 回	すべて異なる	36 回
50 人	同じ人がいる	967 回	すべて異なる	33 回
50 人	同じ人がいる	962 回	すべて異なる	38 回
50 人	同じ人がいる	970 回	すべて異なる	30 回
50 人	同じ人がいる	981 回	すべて異なる	19 回
50 人	同じ人がいる	967 回	すべて異なる	33 回
50 人	同じ人がいる	982 回	すべて異なる	18 回

同じ誕生日

2月29日を考慮

⇒ 2月29日に値 366 を対応させる
乱数が 366 となる確率は、他の値の確率の 1/4

誕生日の確率分布を考慮

⇒ 確率分布を乱数の発生確率に反映させる

↓

モンテカルロ法は、様々な条件に対応することができる

二人零和ゲーム

プレイヤー A と B が対戦

各プレイヤーは手 P, Q を選択する。

プレイヤー A の利得

		B	
		P	Q
A	P	+5	-8
	Q	-2	+5

A が手 P, B が手 P → A が +5, B が -5

A が手 P, B が手 Q → A が -8, B が +8

A が手 Q, B が手 P → A が -2, B が +2

A が手 Q, B が手 Q → A が +5, B が -5

二人零和ゲーム

各プレイヤーは確率的に手を選ぶ

A が手 P を選ぶ確率 x → A が手 Q を選ぶ確率 $1 - x$

B が手 P を選ぶ確率 y → B が手 Q を選ぶ確率 $1 - y$

プレイヤー A の利得を求める。

zero_sum_game_simulation.m

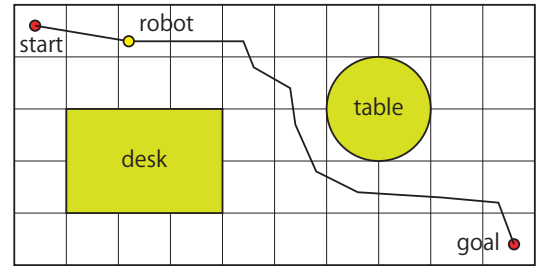
```

g = [ 5, -8; -2, 5 ];
x = 0.35; y = 0.50;

gainA = 0; result = [];
for iteration=1:10000
    if rand < x A = 1; else A = 2; end
    if rand < y B = 1; else B = 2; end
    gainA = gainA + g(A,B);
    result = [ result; iteration, gainA ];
end

plot(result(:,1), result(:,2));
    
```

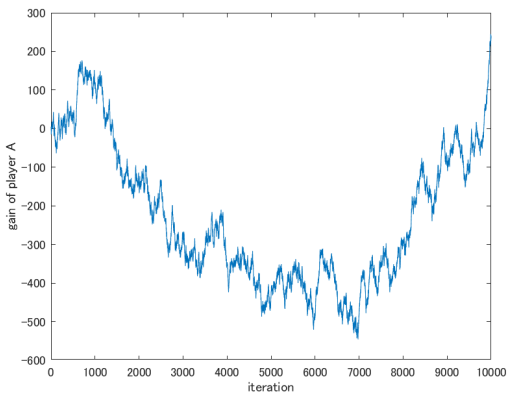
移動ロボットの経路計画



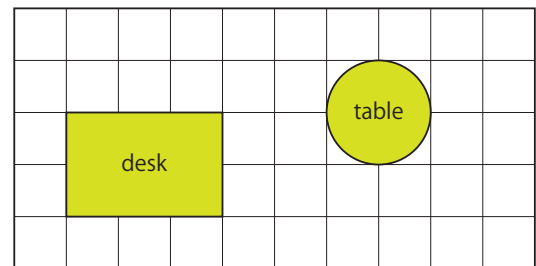
障害物 (机, テーブル) と干渉や接触をすることなく
スタートからゴールに至る経路を見つける

二人零和ゲーム

$x = 0.50, y = 0.50$



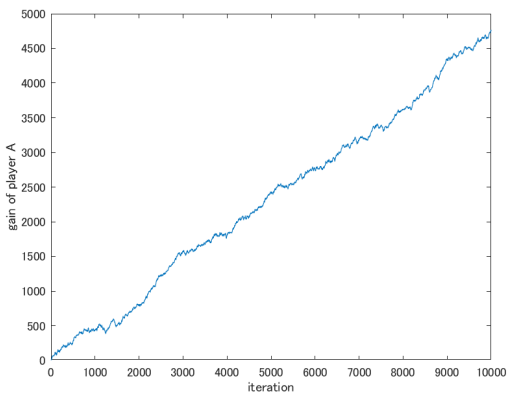
移動ロボットの経路計画



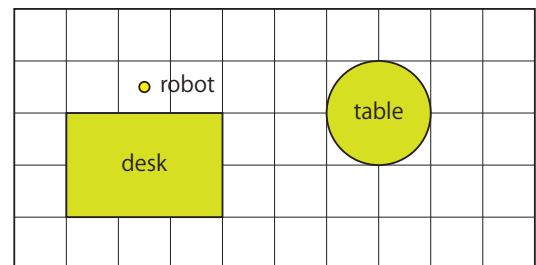
部屋の大きさ: 横 10m 縦 5m
机 (長方形 横 3m 縦 2m)
テーブル (円形 半径 1m)

二人零和ゲーム

$x = 0.35, y = 0.50$



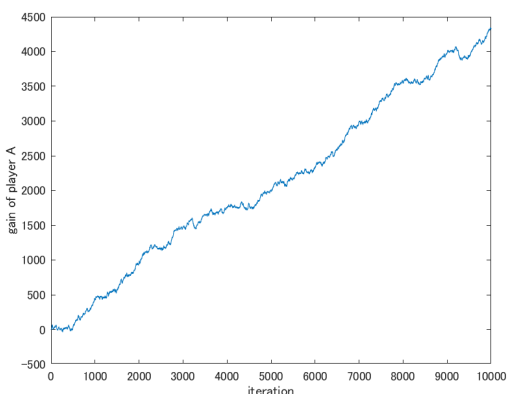
移動ロボットの経路計画



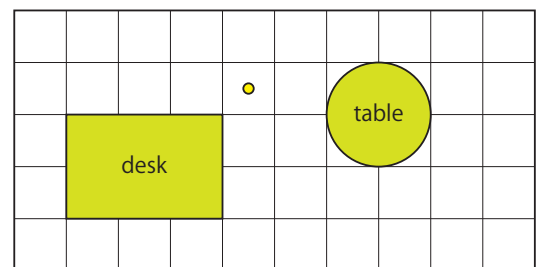
移動ロボット: 大きさを無視する → 点とみなす
部屋の左下角に原点を設定
→ 移動ロボットの位置 座標 (x, y)

二人零和ゲーム

$x = 0.35, y = 0.65$

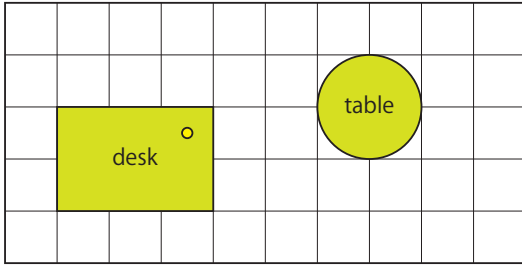


移動ロボットの経路計画



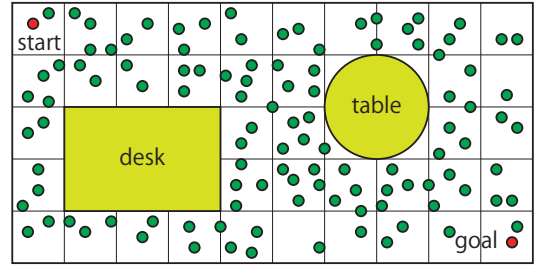
ロボットの位置 $(4.50, 3.50)$
自由 (free)

移動ロボットの経路計画



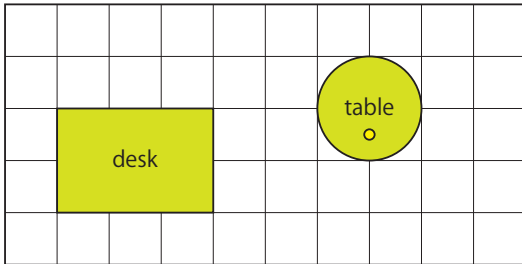
ロボットの位置 (3.50, 2.50)
干渉 (interfered)
→ 物理的に起こらない

PRM (Probabilistic Roadmaps)



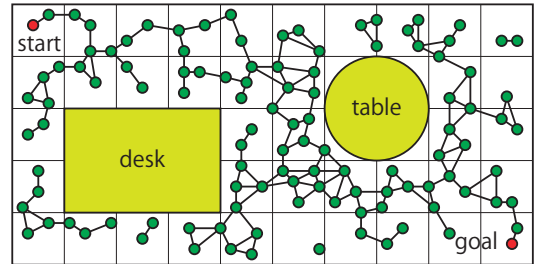
自由な点をランダムに生成

移動ロボットの経路計画



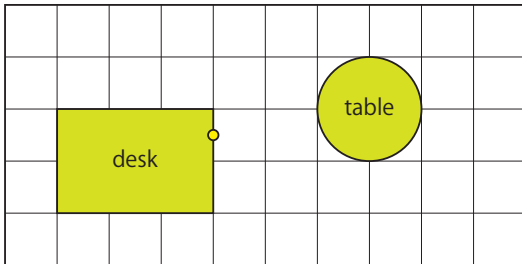
ロボットの位置 (7.00, 2.50)
干渉 (interfered)
→ 物理的に起こらない

PRM (Probabilistic Roadmaps)



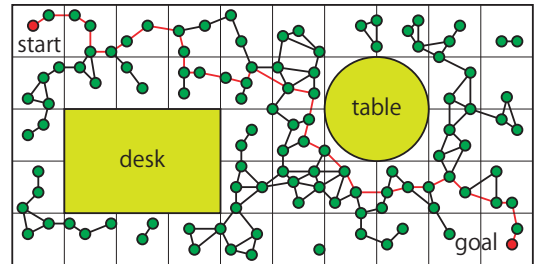
自由な短い線分をエッジで接続

移動ロボットの経路計画



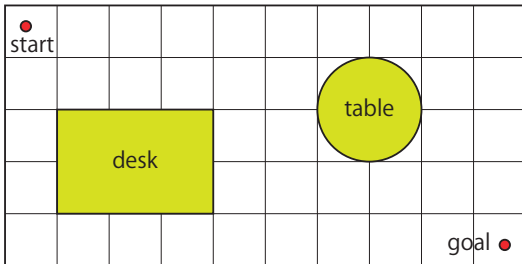
ロボットの位置 (4.00, 2.50)
接触 (contact)
→ 起こりうるが望ましくない

PRM (Probabilistic Roadmaps)



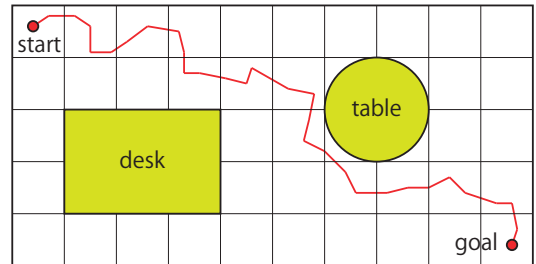
スタートからゴールへ至る経路

PRM (Probabilistic Roadmaps)



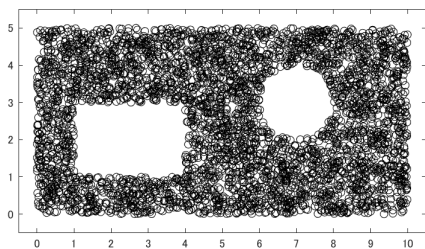
スタートとゴールの設定

PRM (Probabilistic Roadmaps)

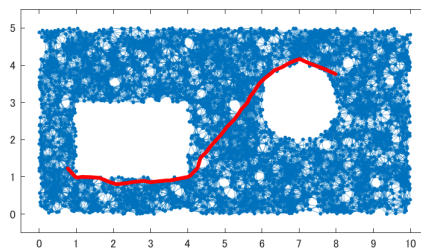


結果

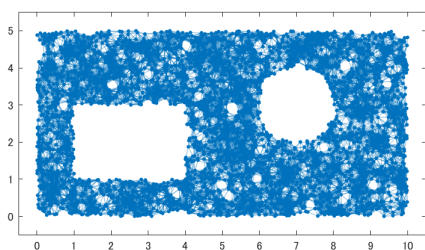
実行例（グラフを作成する）



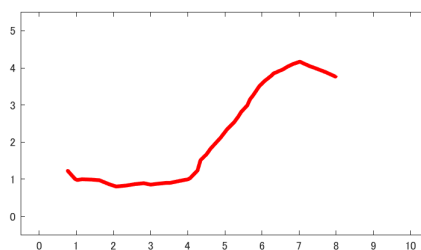
実行例（最短経路を見つける）



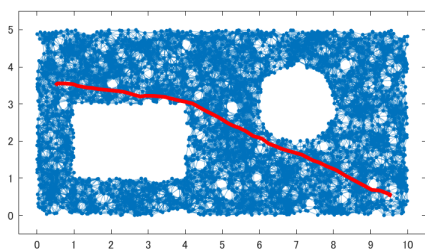
実行例（グラフを作成する）



実行例（最短経路を見つける）



実行例（最短経路を見つける）



まとめ

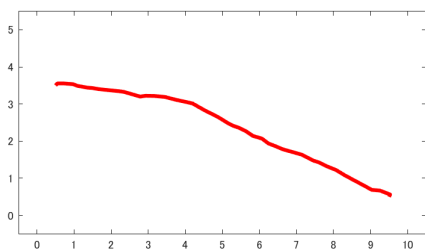
乱数

- 一様乱数 $U(0, 1)$
- 一様乱数 $U(a, b)$
- 正規乱数 $N(0, 1)$
- 正規乱数 $N(\mu, \sigma)$

モンテカルロ法

- 定積分の近似計算
- 多重定積分の近似計算
- 二人零和ゲームのシミュレーション
- 移動ロボットの経路計画

実行例（最短経路を見つける）



レポート (MATLAB Grader)

MATLAB grader 「確率的アルゴリズム」
締切：2024年7月1日（月曜）00:10 AM 6回提出可能

「関数」を書いたのちに、「関数の実行」をクリックすると、ヒストグラムやプロットが表示される。

ヒストグラムやプロットを見て、解答を確認した後に、「提出」をクリックする。

「評価」で検定を行っているため、正しい解答を誤りと判定することがある。そのときは、再度「提出」をクリックする。