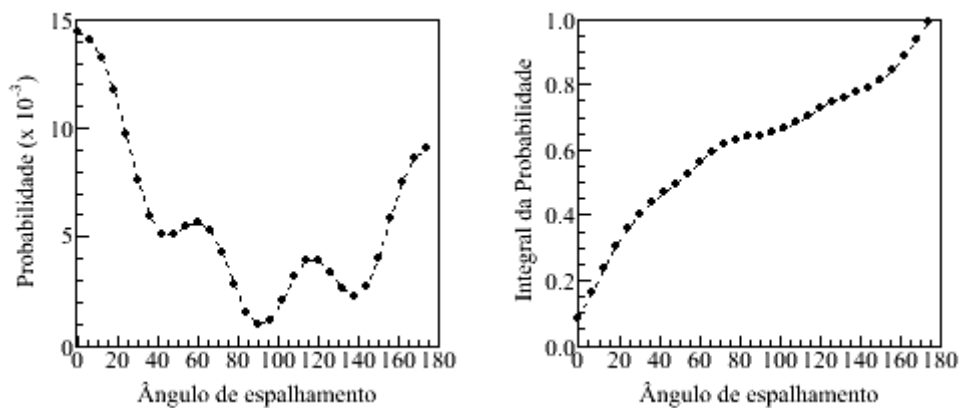


25 Espalhamento

Imagine que você está fazendo simulações sobre radioterapia com feixes de prótons, seguindo a trajetória de milhões de prótons à medida que atravessam os tecidos de um paciente. Para prosseguir com seus cálculos, você precisa saber o ângulo em que os prótons são espalhados pelos núcleos atômicos. Núcleos são sistemas com muitas partes internas que interagem seguindo leis físicas que às vezes não conhecemos muito bem ou que levam a equações diferenciais complicadas que seriam virtualmente impossíveis de serem resolvidas numericamente para um grande número de trajetórias. Como prosseguir?

Uma solução é sortear um ângulo de espalhamento que obedeça uma distribuição de probabilidades capaz de reproduzir os dados experimentais ou os valores calculados obtidos por outros pesquisadores. Se essa distribuição puder ser modelada por uma função matemática que possa ser integrada e o resultado invertido, podemos utilizar o método apresentado anteriormente para fazer o sorteio. Se isto não puder ser feito, o método pode ainda ser empregado numericamente.

A figura abaixo mostra, à esquerda, dados hipotéticos da probabilidade de espalhamento em função do ângulo de espalhamento de uma partícula por outra a determinada energia e, à direita, o valor da integral destas probabilidades.



O que a figura da esquerda nos diz é que a probabilidade de espalhamento a baixos ângulos é alta, tem um mínimo em torno de 90° e volta a subir, com algumas oscilações. Os dados

foram normalizados de modo que a integral de 0° a 180° (a área sob o gráfico), seja igual a 1, que é a probabilidade de ocorrer um espalhamento em qualquer ângulo. A figura da direita foi obtida multiplicando o valor da probabilidade obtida para cada ângulo pelo intervalo angular das medidas (6° , no caso, pois os 180° estão distribuídos em 30 pontos experimentais igualmente espaçados) e acumulando o resultado. Deste modo, temos a integral da função. Agora precisamos da inversão.

A inversão é simples: sorteie um número entre 0 e 1 — por exemplo 0,5; utilize este número como o valor das ordenadas (eixo y) do gráfico da integral da função e encontre o valor correspondente das abscissas (eixo x) — algo perto de 50° . Este é o ângulo desejado.

O script abaixo implementa essa estratégia de gerar números aleatórios que obedecem uma distribuição de probabilidade obtida experimentalmente.

exemplo-25-1.html

```
<script>
// Gerador de números aleatórios - dados exps
function DATARAND() {
  this.theta = new Array();
  this.prob = new Array();
  with (this) {
    theta[0] = 0; prob[0] = 14.39;
    theta[1] = 6; prob[1] = 14.03;
    theta[2] = 12; prob[2] = 13.24;
    theta[3] = 18; prob[3] = 11.75;
    theta[4] = 24; prob[4] = 9.71;
    theta[5] = 30; prob[5] = 7.60;
    theta[6] = 36; prob[6] = 5.95;
    theta[7] = 42; prob[7] = 5.13;
    theta[8] = 48; prob[8] = 5.10;
    theta[9] = 54; prob[9] = 5.48;
    theta[10] = 60; prob[10] = 5.69;
    theta[11] = 66; prob[11] = 5.31;
    theta[12] = 72; prob[12] = 4.25;
    theta[13] = 78; prob[13] = 2.82;
    theta[14] = 84; prob[14] = 1.55;
    theta[15] = 90; prob[15] = 0.95;
    theta[16] = 96; prob[16] = 1.20;
    theta[17] = 102; prob[17] = 2.12;
    theta[18] = 108; prob[18] = 3.19;
    theta[19] = 114; prob[19] = 3.90;
    theta[20] = 120; prob[20] = 3.92;
    theta[21] = 126; prob[21] = 3.35;
    theta[22] = 132; prob[22] = 2.63;
    theta[23] = 138; prob[23] = 2.30;
    theta[24] = 144; prob[24] = 2.77;
    theta[25] = 150; prob[25] = 4.06;
    theta[26] = 156; prob[26] = 5.82;
    theta[27] = 162; prob[27] = 7.51;
    theta[28] = 168; prob[28] = 8.64;
    theta[29] = 174; prob[29] = 9.09;
    for (i in prob) prob[i] = prob[i]/1000;
  }
  this.dtheta = this.theta[1] - this.theta[0];
  var totint = 0;
  this.integral = new Array(this.theta.length+1);
  this.integral[0] = 0;
  for (var i=1;i<this.integral.length;i++) {
```

```

    totint += this.prob[i-1] * this.dtheta;
    this.integral[i] = totint;
  }

  // para o caso da integral total não ser exatamente 1
  for (var i=0;i<this.prob.length;i++)
    this.prob[i] /= totint;
  for (var i=0;i<this.integral.length;i++)
    this.integral[i] /= totint;
}

// método que busca e devolve o ângulo aleatório
DATARAND.prototype.random = function() {
  var x = Math.random();
  with (this)
    for (var i=1;i<integral.length;i++)
      if ((x>=integral[i-1])&&(x<integral[i]))
        return theta[i-1] + (x-integral[i-1]) *
          dtheta/(integral[i]-integral[i-1]);
}

var xmin = 0;
var xmax = 180;
var ncan = 30;
var dx = (xmax-xmin)/ncan;
var dist = new Array(ncan);
for (i=0;i<ncan;i++) dist[i] = 0;

var dRand = new DATARAND();

var npts = 10000;
var x;
for (var i=0;i<npts;i++) {
  x = dRand.random();
  for (var n=0;n<ncan;n++) {
    if ((x>=(xmin+n*dx))&&(x<(xmin+(n+1)*dx))) {
      dist[n]++;
      break;
    }
  }
}

// Faz o gráfico da variável dist em modo texto
// Números na escala. Encontra o maior para usar
// a informação mais adiante, no alinhamento dos nros.
var label = new Array(ncan);
var maxlength = 0;
for (i=0;i<ncan;i++) {
  label[i] = (xmin+i*dx).toFixed(1);
  if (label[i].length>maxlength) maxlength = label[i].length;
}

// Acha o máximo da distribuição e faz com que
// seja representado por 50 asteriscos
var max = 0;
for (n=0;n<ncan;n++)
  if (dist[n]>max) max = dist[n];
var dy = max/50;

document.write("<pre style='font-size:8pt'>");

for (i=0;i<ncan;i++) {
  var nast = Math.round(dist[i]/dy);
  var str = "";

```

```

// preenche de brancos à esquerda do número qdo necessário
for (j=0;j<(maxlength-label[i].length);j++) str += " ";
str += label[i] + " ";
for (n=0;n<nast;n++) str += "*";
str += " " + dist[i] + "<br>";
document.write(str);
}

document.write("</pre>");
</script>

```

Resultado:

```

0.0 ***** 810
6.0 ***** 819
12.0 ***** 778
18.0 ***** 705
24.0 ***** 518
30.0 ***** 454
36.0 ***** 322
42.0 ***** 314
48.0 ***** 251
54.0 ***** 324
60.0 ***** 316
66.0 ***** 320
72.0 ***** 224
78.0 ***** 174
84.0 **** 73
90.0 *** 51
96.0 **** 70
102.0 ***** 117
108.0 ***** 170
114.0 ***** 242
120.0 ***** 224
126.0 ***** 190
132.0 ***** 161
138.0 ***** 150
144.0 ***** 170
150.0 ***** 245
156.0 ***** 358
162.0 ***** 452
168.0 ***** 491
174.0 ***** 507

```

O objeto `DATARAND` tem quatro propriedades: a matriz `this.theta`, que guarda os valores dos ângulos, a matriz `this.prob`, que guarda os valores da probabilidade de espalhamento em cada ângulo, a matriz `this.integral`, que guarda a integral da distribuição em função do ângulo e `dtheta`, o intervalo entre dois ângulos.

Para não sobrecarregar visualmente o arquivo do documento com enormes tabelas, caso um grande volume de dados experimentais seja utilizado, a definição do objeto poderia ser feita num arquivo `.js` à parte e carregado com o elemento `<script src="...">` (como no exemplo sobre o átomo de hidrogênio).

Note o uso, no construtor, da instrução `with` para evitar repetir o `this.` na frente de cada `theta` e `prob` da tabela. Note também que, por conveniência de visualização, após a atribuição dos valores às probabilidades, há um laço ainda dentro do escopo do `with` que divide-as por 1000 (se não fosse assim os valores das probabilidades seriam 0.01439, 0.01403 e assim por diante).

O laço seguinte acumula as probabilidades de modo a obter a integral da distribuição para os

valores dos ângulos. Observe que a propriedade `this.integral` é definida com um elemento a mais do que as propriedades `this.theta` e `this.prob`. Este elemento a mais é o primeiro, que recebe 0 (`this.integral[0] = 0`) para permitir que a interpolação feita no método `random()` funcione corretamente quando o valor uniforme sorteado corresponde ao primeiro canal da distribuição.

Ainda no construtor, o total geral (`totint`) é acumulado para renormalizar a distribuição, de modo que a integral seja de fato 1 e não um pouco menor devido a erros de arredondamento nos valores dos dados experimentais. Note o uso das construções `var1 += var2` e `var1 /= var2` que são equivalentes a `var1 = var1 + var2` e `var1 = var1 / var2`, respectivamente.

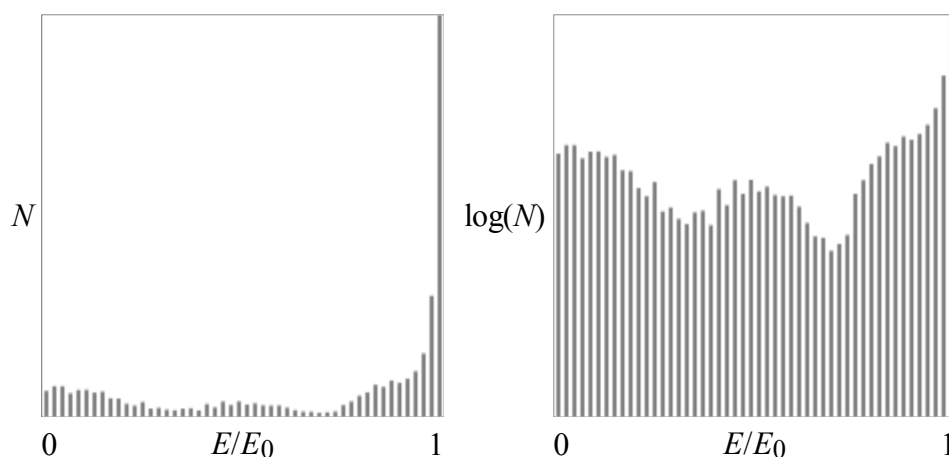
O método `random()` sorteia um número aleatório que obedece uma distribuição uniforme e que é valor de uma ordenada no gráfico da integral. A seguir procura, utilizando um laço `for`, a que valor das abscissas o valor sorteado corresponde neste gráfico. Quando encontra o canal correspondente, faz uma interpolação linear para obter o valor do ângulo que será retornado.

Finalmente, o script segue fazendo o que foi feito nos dois capítulos anteriores: imprimindo no modo texto o gráfico da distribuição sorteada. E, como você pode notar, ele simplesmente reproduz a distribuição de probabilidades da qual partimos.

Exercícios

- Um feixe monoenergético de partículas incide sobre um alvo e tem uma probabilidade de espalhamento dada pela distribuição descrita no texto. Supondo que a perda de energia no espalhamento é proporcional ao cosseno da metade do ângulo de espalhamento (ou seja, é nula para o espalhamento a 0° e total a 180°), qual a distribuição da energia das partículas espalhadas?

Nas figuras abaixo estão os resultados esperados. À esquerda, em uma escala vertical linear; à direita, a mesma distribuição com uma escala vertical logarítmica.



- Qual a distribuição angular das partículas que sofreram dois espalhamentos sucessivos segundo a distribuição acima? Seu algoritmo deve levar em conta o fato de que a distribuição acima refere-se a ângulos entre 0° e 180° que poderiam ser positivos ou negativos com relação à direção de incidência, com igual probabilidade.

