计算物理作业 16

刘畅　　PB09203226

2012 年 12 月 18 日

[**作业 16**]: 设体系的能量为 $H = \frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}$ (以 kT 为单位), 采用 Metropolis 抽样法计算 $\langle x^2\rangle$, $\langle y^2\rangle$, $\langle x^2+y^2\rangle$, 并与解析结果进行比较. 抽样时在 2 维平面上依次标出 Markov 链点分布, 从而形象地理解 Markov 链.

1　算法

由于做一个 (线性的) 变量代换

$$x \to x/\sigma_x$$
$$y \to y/\sigma_y$$

就可以变到题目中的情形, 因此设 $\sigma_x = \sigma_y = 1$ 并不会影响一般性. 所以我们下面都采取这样的假定. 由于 H 用 kT 做单位, 因此 $\beta = 1$, 正则系综的配分函数为

$$Z = \int_{\mathbb{R}^2} \mathrm{e}^{-H}\,\mathrm{d}x\,\mathrm{d}y = \int_{-\infty}^{+\infty}\mathrm{d}x\int_{-\infty}^{+\infty}\mathrm{d}y\,\mathrm{e}^{-\frac{x^2}{2}-\frac{y^2}{2}} = 2\pi$$

这样系统处于态 (x,y) 的概率密度是

$$p(x,y) = \frac{\exp\left(-\frac{x^2}{2}-\frac{y^2}{2}\right)}{2\pi}$$

因此 x^2, y^2, x^2+y^2 理论上的平均值为

$$\langle x^2\rangle = \int_{\mathbb{R}^2} p(x,y)x^2\,\mathrm{d}x\,\mathrm{d}y = 1$$

同理

$$\langle y^2\rangle = 1$$
$$\langle x^2+y^2\rangle = 2$$

容易看出在 $\sigma_x \neq 1$, $\sigma_y \neq 1$ 的单位下上面的结果是

$$\langle x^2 \rangle = \sigma_x^2$$
$$\langle y^2 \rangle = \sigma_y^2$$
$$\langle x^2 + y^2 \rangle = \sigma_x^2 + \sigma_y^2$$

按照书上的算法, Metropolis 算法的步骤是

1. 选择一个初始构形 $\mathbf{R}_0 = (x_0, y_0)$
2. 在 $[-h, h] \times [-h, h]$ 区间内的均匀分布中随机选择一个点作为 $\Delta\mathbf{R}$, 计算新的构形 $\mathbf{R}'$.
3. 计算新的构形和旧的构形在正则分布概率密度下出现的概率的比值
$$\eta = \frac{p(x', y')}{p(x, y)}$$
4. 随机抽取 $[0, 1]$ 上的均匀分布 w, 如果 $\eta > w$, 那么 $\mathbf{R}_{i+1} = \mathbf{R}'$, 否则 $\mathbf{R}_{i+1} = \mathbf{R}_i$.
5. 重复直到足够多步

最后得到的点列 $\mathbf{R}_i$ 中, 为了减少误差, 通常舍去前面的一些点, 同时为了减少自相关系数的影响, 中间要跳过一些点, 最后得到 $f(\mathbf{r})$ 的值为

$$\langle f \rangle = \sum_{i=0}^{\text{nsteps}} f(\mathbf{R}_{n_0 + i \cdot n_d})$$

2 程序

按照惯例, 用一个结构来表示点 (x, y):

```
struct point {
    double x, y;
};
```

`rand_norm()` 和 `rand_unif()` 是两个均匀分布的抽样函数, 前面的作业用过好多次了. 由于前面的算法中只要用到 $p(x, y)$ 的相对值, 因此 $\frac{1}{2\pi}$ 系数是不需要的. $p(x, y)$ 函数实现在 `prob_dist()`:

```
/* Probability distribution for our canonical ensemble with
 * Hamiltonian H(x,y) = x^2/2 + y^2/2
 */
double prob_dist(struct point p)
{
    return exp(-p.x*p.x/2 - p.y*p.y/2);
}
```

generate_new_configuration() 函数用来实现算法的主要部分: 从旧的构形生成新的构形, 就是把前面的算法翻译成 C 语言, 这个没有什么好解释的:

```
/* generate a new configuration based
   upon the above accepting criterion */
struct point generate_new_configuration
                (double h, struct point old)
{
    double p;
    struct point new;
    double x = rand_unif(h);
    double y = rand_unif(h);
    new.x = old.x + x;
    new.y = old.y + y;
    p = prob_dist(new) / prob_dist(old);
    if (is_accepted(p))
        return new;
    else
        return old;
}
```

其中 is_accepted() 例程实现上面算法中的舍选.

最后只需要有选择地把 $\mathbf{R}_i$ 处的值加起来就得到了 $\langle f \rangle$. 这段代码在 metropolis(). 这代码中唯一需要解释的是宏 spit_all() 和 spit_hit(). 前者用来把每个 $\mathbf{R}_i$ 输出到文件, 后面把求 $\langle f \rangle$ 用到的 $\mathbf{R}_i$ 输出到文件.

3 结果

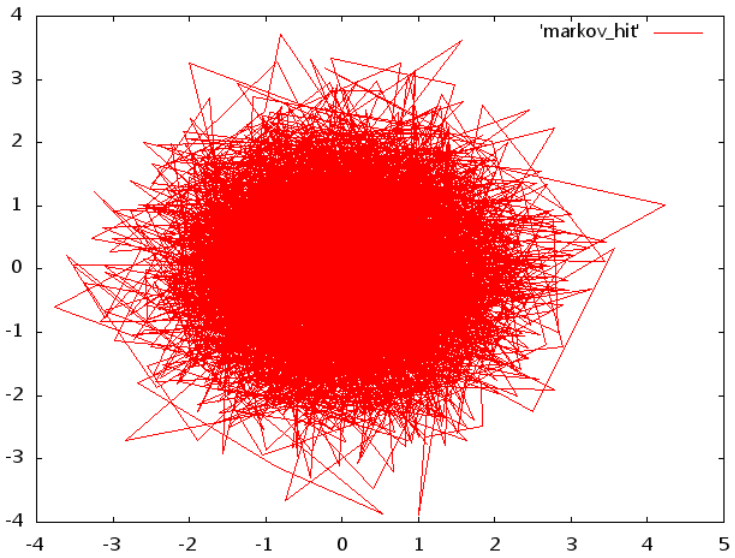
程序运行的结果为:

$$\langle x^2 \rangle = 1.050201266062$$

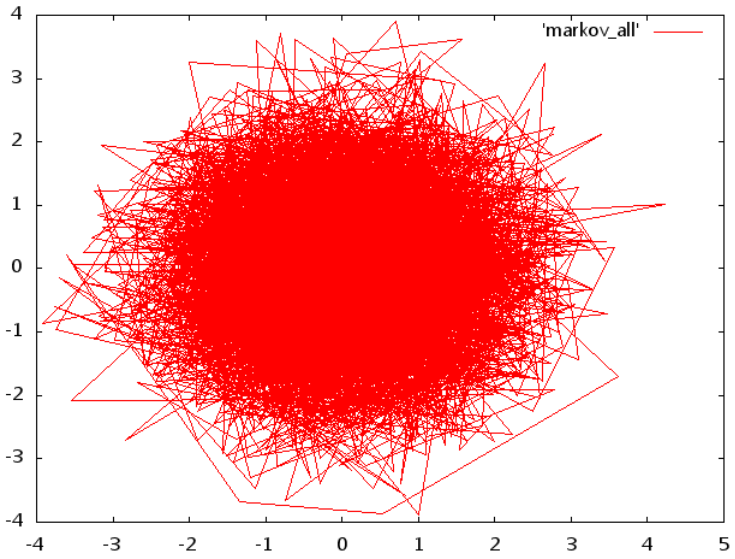
$$\langle y^2 \rangle = 0.940981966345$$

$$\langle x^2 + y^2 \rangle = 1.972662062136$$

可以看到和理论值是非常接近的. 每一步 Markov 链的图为:



这是 `spit_hit()` 的输出.



这是 `spit_all()` 的输出.