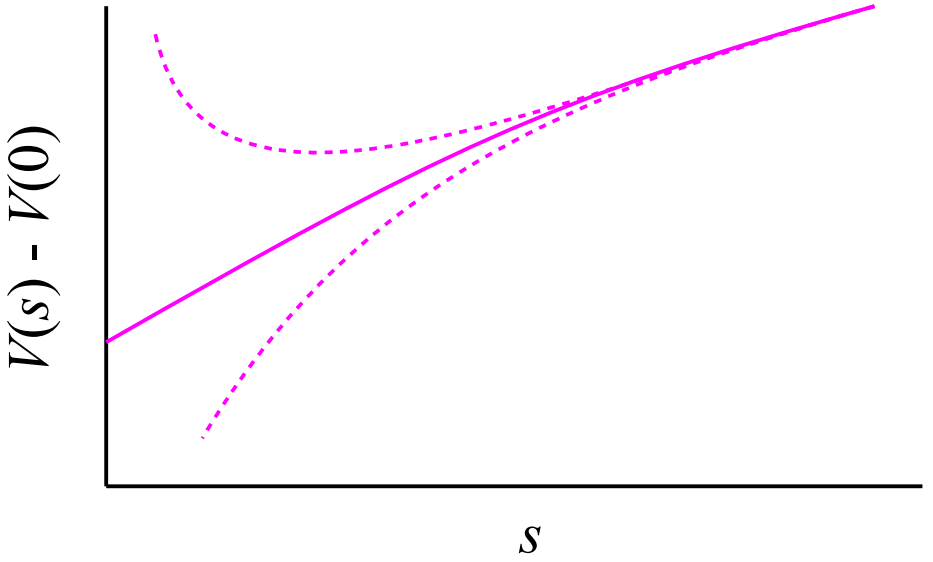
縮放尺寸 (scaling) 估計法

對於有限尺度的測量 $V(s)$

估計其在無窮系統的最佳值 $V(0)$

Scaling 的 assumption:

$$V(s)=V(0)+s^{\alpha}(A+sB+s^{2}C+\ldots)$$



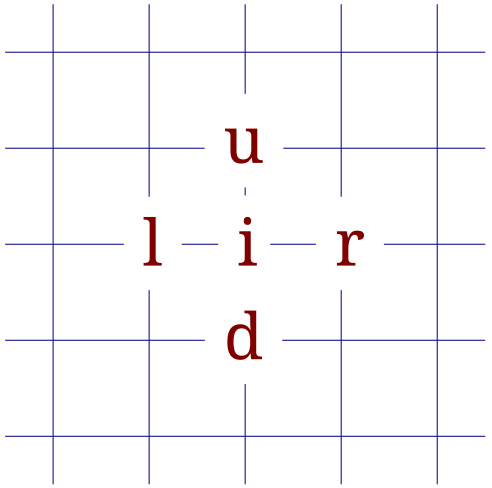
Poisson/Laplace 方程式

$$\nabla \varphi = -\rho$$

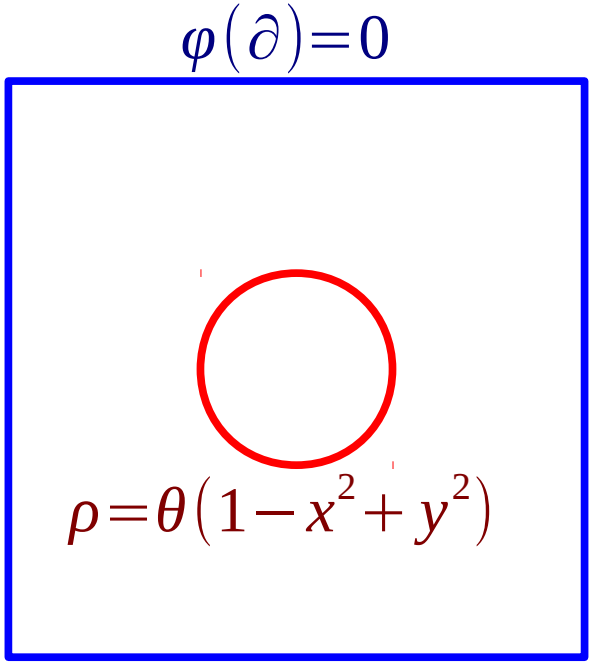
$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \varphi = -\rho$$

格點化

$$\approx \frac{1}{a^2} \left(\varphi_d + \varphi_u + \varphi_l + \varphi_r - 4\varphi_i \right)$$



$\rho = 0$



給定邊界值條件

例子:
不可壓縮和無旋條件的流體速度場
不隨時間變化的電場
熱傳導問題

邊界值問題的鬆弛法

- 設定系統邊界
- 重覆掃描系統
- 更新格點上的函數值
- 檢查誤差

$$\varphi_i = \frac{1}{4}\left(\varphi_d + \varphi_u + \varphi_l + \varphi_r + a^2 \rho_i\right)$$

例子:

Allocate system:

```
sz = (szx + 1) * (szy + 1);
phi = new double [sz];
rho = new double [sz];
nph = new double [sz];
```

Lattice:

```
0 < ix < szx, 0 < iy < szy
```

Boundary:

```
ix = 0, ix = szx, iy = 0, iy = szy
```

Coordinate:

```
x = 10 * ix / szx - 5;
y = 10 * iy / szy - 5;
```

Source:

```
rho[i] = x * x + y * y < 1.0 ? 1.0 : 0;
```

Update:

```
nph[i] = (phi[i-1] + phi[i+1] + phi[i-szy-1]
          + phi[i+szy+1] + a2 * rho[i]) / 4;
```

Find error:

```
double chg = fabs(nph[i] - phi[i]);
if (chg > error) error = chg;
```

Exchange memory:

```
t = phi;
phi = nph;
nph = t;
```

三維作圖

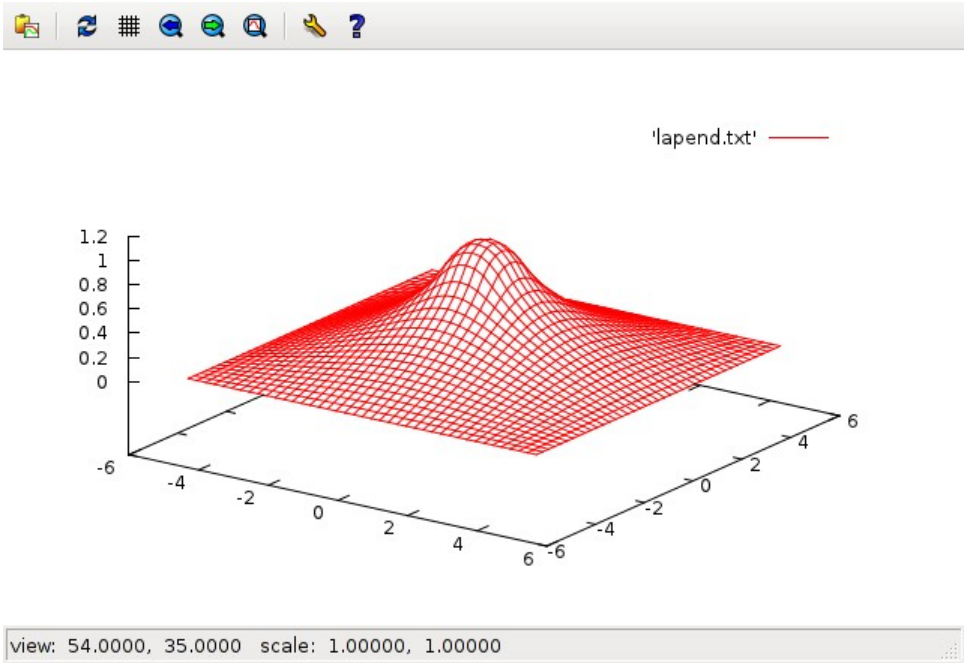
資料檔格式:

```
x1 y1 <z value>
x1 y2 <z value>
…
x1 yn <z value>
<empty line>
x2 y1 <z value>
...
```

多組資料的檔案:
以雙空行分隔

Gnuplot:

```
gnuplot> set st da lines
gnuplot> set hidden
gnuplot> splot 'lapend.txt'
gnuplot>
```



簡單的 Animation

Gnuplot 的 loop
檔案: “show_lap.gp”

```
splot 'lap.txt' index i
i = i + 1
pause 1
if (i < 13) reread
```

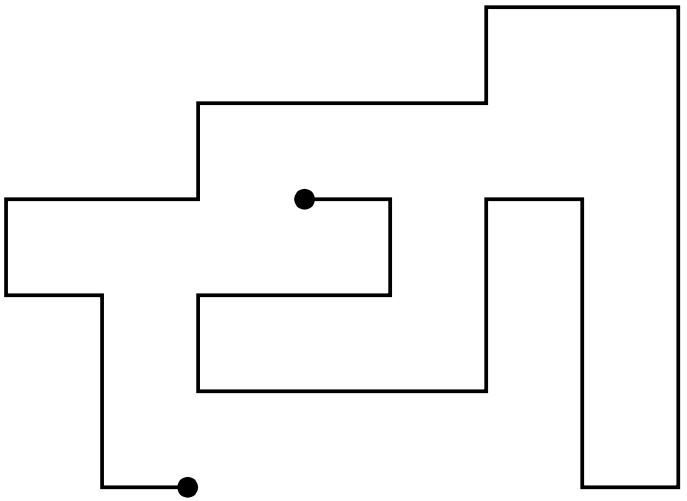
```
gnuplot> set st da lines
gnuplot> set zr [0:1.2]
gnuplot> i = 0
gnuplot> set hidden
gnuplot> load 'show_lap.gp'
```

自避型 (Self-avoiding) 隨機漫步

二維隨機漫步的問題:
- 平均位移
- 回到過出發點的機率
- 重覆到過的位置

有的問題須要存留完整的路徑軌跡

可能限制:
- 不能回頭走
- 不能重覆位置 (self-avoiding walk)
- 不能重覆路徑



Displacement: $\langle R^2 \rangle_n \sim n^{2v}$

Random walk: $v = 1/2$
2D self-avoiding walk: $v = 3/4$
3D self-avoiding walk: $v \approx 0.6$

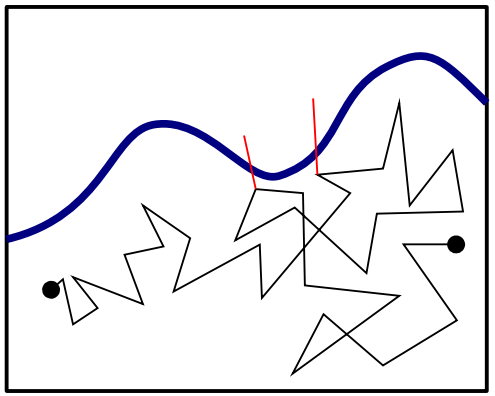
Number of walks: $c_n \propto \mu_n n^{\gamma-1}$

Random walk: $\gamma = 1$, $\mu_n = (2d)^n$
2D self-avoiding walk: $\gamma = 43/32$

Application:
聚合物 (polymer)
生物分子: DNA, 蛋白質
紐結 (knot) 理論

相依型 (Correlated) 取樣 (sampling)

非獨立的隨機點序列
例：用 random walk 作 MC 積分



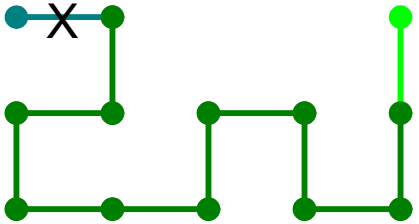
排拒 (Rejection) 法

1.取一許可的啓始點（系統狀態）
2.取樣
3.隨機變更此點來得取一新點
4.如新點不為許可點則重取此點
5.回到 2. 取樣

＊排拒後須重取此點，不然取樣會有偏差。

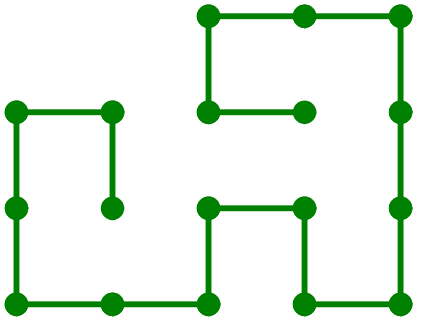
考慮一原數列受 n > 2 的限制:
4, 3, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 2, 3, 4, 5, 4, 3, 2, 3
在 rejection 的區間前後 3 都會被各取一次

游蛇 (slithering snake) 法
for self-avoiding walk



隨機變更:
任意由一端截去一段
在另一端的任意方向補回一段

＊有遍歷性 (ergodicity) 問題
無法到達的狀態:



自避漫步實作要點

- 用二維陣列來記錄被占據的 lattice sites
- 用一維陣列來存放 snake 所占據的 sites
- 將一維陣列視為環形（ snake 頭尾存位相鄰）
- 標記蛇頭在一維陣列中的存放位置
- 截頭補尾或截尾補頭僅須移動蛇頭標記

其他方法
- End-bond, kink-jump, crankshaft (Verdier-Stockmayer)
- Pivot algorithm

非定鍵長
- Bond-fluctuation model
- off-lattice models

本週作業

1. 用鬆弛法解 Poisson 方程式，邊界及電荷分佈如右。制作三維的電位圖。
2. 用游蛇法計算自避漫步在步數 $n = 100, 200, 400, 800, 1600$ 時的平均頭尾距離。
3. 分析 2. 所得數據，求 $\langle R^2 \rangle_n \sim n^{2\nu}$ 中的 ν 值。