# Partial Differential Equation Solving, vol 2.

Marcin Chrząszcz
mchrzasz@cern.ch

University of Zurich^UZH

Monte Carlo methods,
12 May, 2016

# Announcement

There will be no lectures and class on 19th of May

# Dirichlet conditions:expected number of steps

⇛ find the function $u(x_1, x_2, ..., x_k)$ such that if fulfils the Laplace equation:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + ... + \frac{\partial^2 u}{\partial x_k^2} = 0, \quad (x_1, x_2, ..., x_k) \in D \subset \mathbb{R}^k$$

In the domain $D$, on the the $\Gamma(D)$ the $u$ function is given by:

$$U(x_1, x_2, ..., x_k) = f(x_1, x_2, ..., x_k), \quad (x_1, x_2, ..., x_k) \in \Gamma(D)$$

⇛ Now lets assume that the domain $D$ is a hyperball:

$$0 \leqslant \sum_{i=1}^{k} x_i^2 \leqslant r^2, \quad r = \text{const}$$

⇛ Now $\pi_\nu(x_1, x_2, ..., x_k)$ is a probability that a particle starting from $(x_1, x_2, ..., x_k)$ will end up on the edge after $\nu$ steps. The $\kappa(x_1, x_2, ..., x_k)$ is the estimated number of steps for this trajectory.

$$\pi_0 = \begin{cases} 1, & (x_1, x_2, ..., x_k) \in \Gamma(D) \\ 0, & (x_1, x_2, ..., x_k) \in D \end{cases} \tag{1}$$

$$\pi_\nu = \frac{1}{2k} \sum' \pi_\nu(x_1\prime, x_2\prime, ..., x_k\prime)$$

# Dirichlet conditions:expected number of steps

⇛ From Eq. 1 and 2 one gets:

$$\kappa(x_1, x_2, ..., x_k) = \sum_{\nu=1}^{\infty} \nu \pi_{\nu}(x_1, x_2, ..., x_k)$$

one gets:

$$\kappa(x_1, x_2, ..., x_k) = \frac{1}{2k} \sum_{\nu=1}^{\infty} \left[ \nu \sum^{\prime} \pi_{\nu-1}(x_1, x_2, ..., x_k) \right]$$

$$= \frac{1}{2k} \sum_{\nu=1}^{\infty} \left[ (\nu-1) \sum^{\prime} \pi_{\nu-1}(x_1{\prime}, x_2{\prime}, ..., x_k{\prime}) \right] + \frac{1}{2k} \sum_{\nu=1}^{\infty} \sum^{\prime} \pi_{\nu-1} x_1{\prime}, x_2{\prime}, ..., x_k{\prime})$$

⇛ From which we get:

$$\kappa(x_1, x_2, ..., x_k) = \frac{1}{2k} \sum^{\prime} \kappa(x_1{\prime}, x_2{\prime}, ..., x_k{\prime}) + 1$$

⇛ Now this is equivalent of the Poisson differential equation:

$$\frac{\partial^2 \kappa}{\partial x_1^2} + \frac{\partial^2 \kappa}{\partial x_2^2} + ... + \frac{\partial^2 \kappa}{\partial x_k^2} = -2k, \text{ b. con. } \kappa(x_1, x_2, ..., x_k) = 0, \quad (x_1, x_2, ..., x_k) \in \Gamma(D)$$

# Dirichlet conditions:expected number of steps

⇛ From previous equation: $\kappa(x_1, x_2, ..., x_k) = \psi(x_1, x_2, ..., x_k) - \sum_{i=1}^{k} x_i^2$ we get the for the $\psi$ function the Laplace equation:

$$\frac{\partial^2 \psi}{\partial x_1^2} + \frac{\partial^2 \psi}{\partial x_2^2} + ... + \frac{\partial^2 \psi}{\partial x_k^2} = 0$$

because on the border ($\Gamma(D)$):

$$\psi(x_1, x_2, ..., x_k) = r^2 = \mathrm{const}$$

so also inside the $D$: $\psi(x_1, x_2, ..., x_k) = r^2 = \mathrm{const}$ ⇛ From which we can estimate the number steps in the random walk:

$$\kappa(x_1, x_2, ..., x_k) = r^2 - \sum_{i=1}^{k} \leqslant r^2$$

## Important conclusion:

The expected number of steps in the random walk (the time of walk) from the point $(x_1, x_2, ..., x_k)$ till the edge od the domain can be estimated by $r$ number (the LINEAR! size). It is completly independent of the $k$!

# Dirichlet conditions as linear system

⇛ In the discrete form we can write the Dirichlet conditions as (2-dim case):

$$u(x,y) = \frac{1}{4}\left[u(x-1,y) + u(x+1,y) + u(x,y-1) + u(x,y+1)\right], \quad (x,y) \in D$$
$$u(x,y) = f(x,y), \quad (x,y) \in \Gamma(D)$$

⇛ Now we can order the grid ($(x,y) \in D \cup D$), we can represente the above equations as a linear system:

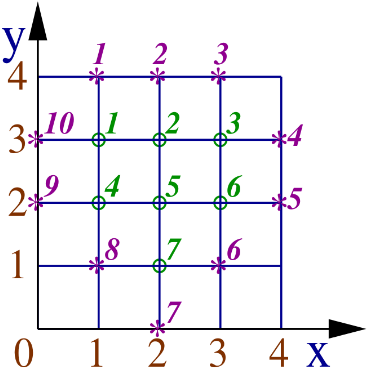
$$u_i = a_i + \sum_{j=1}^{n} h_{ij} u_j, \quad i = 1, 2, ...., n$$

## The trick:

So to solve a differential equation with Dirichlet boundary condition we can use all the methods of solving linear equation systems such as Neumann-Ulam or Wassow.

# Dirichlet conditions as linear system - example

- To do this we act as following: we number separately the points inside the $D$ domain and on the border $\Gamma(D)$.
- We write for each point inside the domain the Laplace equation as system of linear equations:



$$
\begin{array}{llllllll}
u_1 & -u_2/4 & & -u_4/4 & & & & = (f_1+f_{10})/4 \\
-u_1/4 & u_2 & -u_3/4 & & -u_5/4 & & & = (f_2)/4 \\
 & -u_2/4 & u_3 & & & -u_6/4 & & = (f_3+f_4)/4 \\
-u_1/4 & & & u_4 & -u_5/4 & & & = (f_8+f_9)/4 \\
-u_1/4 & & & -u_4/4 & u_5 & -u_6/4 & -u_7/4 & = 0 \\
 & & -u_3/4 & & -u_5/4 & u_6 & & = (f_5+f_6)/4 \\
 & & & & -u_5/4 & & u_7 & = (f_5+f_6)/4
\end{array}
$$

# Dirichlet conditions as linear system - example

⇛ The above equation we can transform the above equation into the iterative representation:

$$\overrightarrow{u} = \overrightarrow{a} + \mathbf{H}\overrightarrow{u}$$

where $\overrightarrow{u} = (u_1, u_2, ..., u_7)$ is the vector which represent the values of the function inside the $D$ domain, $\overrightarrow{a}$ is the linear combinations of the boundary values. In our example:

$$\mathbf{H} = \begin{pmatrix} 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \end{pmatrix}$$

⇛ To find the solution to aka $\overline{u}$ one can use the methods we already know: Neumann-Ulam and Wasow, etc.

⇛ There are tricks and tips one can use to make this problem faster as each of the entry is $\frac{1}{4}$.

# Neumann-Ulam method

⇛ We put the particle in $(x, y)$.
⇛ We observe the trajectory of the particle until it reaches the boundary. Point $P_k$ is the last point before hitting the boundary.
⇛ For each trajectory we assign a value that the arithmetical mean of the boundary points that are neighbours of the point $P_k$.
⇛ We repeat the above $n$ times and calculate the mean.
⇛ The example solution for $20$ trajectories:

$$u(2,2) = 1.0500 \pm 0.2756$$

⇛ E 10.1 Solve the above linear system using the Neumann-Ulam method for an assumed boundary conditions.

# Dual Wasow method

⇛ We choose the boundary conditions with arbitrary chosen probability p.d.f. $p(Q)$ the starting point.

⇛ We choose with equal probability the point inside $D$ where the particle goes.

⇛ With equal probability we choose the next positions and so on until the particle hits the boundary in the point $Q'$.

⇛ We count all trajectories $N((x_1, x_2, x_3, ..., x_k)$ that that have passed the point $(x_1, x_2, x_3, ..., x_k)$.

⇛ For the point $(x_1, x_2, ..., x_k)$ we calculate:

$$w(x_1, x_2, ..., x_k) = \frac{1}{2k} N(x_1, x_2, ..., x_k) \frac{f(Q)}{p(Q)}$$

⇛ The above steps we repeat $N$ times.

⇛ After that we take the arithmetic mean of $w$.

# Random walk with different step size

⇛ If $u(x, y)$ is a harmonic function that obeys the Laplace equation and $S_r(x, y)$ is a circle in with the middle point $(x, y)$ and radius $r$. Then a theorem states:

$$S_r(x,y) = \frac{1}{2\pi} \int_0^{2\pi} u(x + r\cos\phi, y + r\sin\phi) d\phi$$

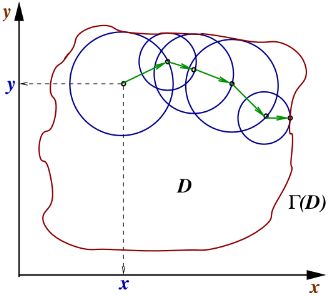
⇛ The above is true for in all the dimensions.
⇛ The E.Muller method:

- At the begging we set the point in the initial point: $(x_1, x_2, ..., x_k)$.
- We construct a $k$ dimensional sphere with center $(x_1, x_2, ..., x_k)$ and radius $r$. The $r$ has to be choosen in a way that the whole is inside the $D$: $S_r(\overrightarrow{x}) \in D$. We choose a random point from $\mathcal{U}(0, 2\pi)$ on the sphere which is our new point.
- We stop the walk when the point is on $\Gamma(D)$.

⇛ We repeat this $N$ times.
⇛ The final result if the arithmetical mean of all trajectories and is equal of the $u(x_1, x_2, ..., x_k)$.

# Muller method

⇛ The method is faster the faster the particle reaches the edge.
⇛ In order to do so we choose the radius that it is the maximal one that allows the sphere to be inside the domain $D$.



⇛ There is a problem!!!! The probability that we choose a point on the edge is $0$!!!!
⇛ An approximation has to be made: we choose a small number $\delta$ and we consider that the particle reached the border when the distance is with $\delta$.
⇛ We can always choose the $\delta$ such that the estimator error of function is smaller then a given $\epsilon$.

# Muller method, example

⇛ An example solution of Laplace equation on square $(0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1)$ with the boundary conditions: $u(0,y) = 1,\ u(1,y) = u(x,0) = u(x,1) = 1$

| Method | Points $(x,y)$ | N. trajectories | Ave.n.of.steps | Time $[s]$ | Solution |
|---|---|---|---|---|---|
| Cons. step<br>$(h = 0.05)$ | $(0.3, 0.3)$ | 2000 | 89.87 | 42.0 | 0.396 |
| | $(0.5, 0.1)$ | 2000 | 46.05 | 21.5 | 0.075 |
| | $(0.5, 0.5)$ | 2000 | 115.83 | 54.1 | 0.247 |
| Muller met. | $(0.3, 0.3)$ | 2000 | 6.06 | 17.9 | 0.398 |
| | $(0.5, 0.1)$ | 2000 | 6.04 | 18.0 | 0.078 |
| | $(0.5, 0.5)$ | 2000 | 5.07 | 14.5 | 0.255 |

# Other boundary conditions

⇛ Find the solution to the Laplace equation:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + .. + \frac{\partial^2 u}{\partial x_k^2} = 0, \quad (x_1, x_2, ..., x_k) \in D \subset \mathbb{R}^k$$

inside the $D$ domain if on the edge $\Gamma(D)$ the function fulfils the equation:

$$f(x_1, x_2, ..., x_k)\frac{\partial u(x_1, x_2, ..., x_k)}{\partial n} + g(x_1, x_2, ..., x_k)u(x_1, x_2, ..., x_k) = h(x_1, x_2, ..., x_k)$$

where $\frac{\partial u(x_1, x_2, ..., x_k)}{\partial n}$ is there derivative in the direction of normal to the $\Gamma(D)$ in the direction inside $D$.

⇛ The cases:

- $f = 0$. ⇛ Dirichlet boundary condition (1st class condition).
- $g = 0$. ⇛ Neumann boundary condition (2nd class condition).
- others. ⇛ General case (3rd class condition).

# Other boundary conditions

⇛ In 2-dim:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (x, y) \in D \subset \mathbb{R}^2$$

with the boundary condition:

$$f(x, y)\frac{\partial u(x, y)}{\partial n} + g(x, y)u(x, y) = h(x, y), \quad (x, y) \in \Gamma(D)$$

⇛ And the discrete differential equation:

$$u(x, y) = \frac{1}{4}\left[u(x - h, y) + u(x + h, y) + u(x, y - h) + u(x, y + h)\right]$$

## Reminder:

If at moment $t$ the point is in $(x, y)$ then in the $t + 1$ time the particle moves with equal probability to one of the following points: $(x - h, y)$, $(x + h, y)$, $(x, y - h)$, $(x, y + h)$.

# Random walk for boundary points

⇛ The boundary point $Q$ has only one internal neighbour point $P$.

- If the normal is parallel to the grid axis in the point $Q$:

$$f(Q)\frac{u(P)-u(Q)}{h}+g(Q)u(Q)=h(Q)$$

- Solving the above to get $u(Q)$ we get:

$$u(Q)=\frac{f(Q)u(P)}{f(Q)-hg(Q)}-\frac{h(Q)}{f(Q)-hg(Q)}$$

- To help we assign a temporary values:

$$\phi(Q)=\frac{f(Q)}{p\left[f(Q)-hg(Q)\right]},\quad \psi(Q)=-h\frac{h(Q)}{(1-p)\left[f(Q)-hg(Q)\right]}$$

⇛ So:

$$u(Q)=p\phi(Q)u(P)+(1-p)\psi(Q)$$

# Random walk for boundary points

⇛ The boundary point $Q$ has only one internal neighbour point $P$.

- If the normal is parallel to the grid axis in the point $Q$:

$$f(Q)\frac{u(P)-u(Q)}{h}+g(Q)u(Q)=h(Q)$$

- Solving the above to get $u(Q)$ we get:

$$u(Q)=\frac{f(Q)u(P)}{f(Q)-hg(Q)}-\frac{h(Q)}{f(Q)-hg(Q)}$$

- To help we assign a temporary values:

$$\phi(Q)=\frac{f(Q)}{p\left[f(Q)-hg(Q)\right]},\quad \psi(Q)=-h\frac{h(Q)}{(1-p)\left[f(Q)-hg(Q)\right]}$$

⇛ So:

$$u(Q)=p\phi(Q)u(P)+(1-p)\psi(Q)$$

⇛ Interpretation: $u(Q)$ can be seen that with probability $p$ it is equal $\phi(Q)u(P)$ and with provability $(1-p)$ is equal to $\psi(Q)$.

# Random walk for boundary points, continued

⇛ The boundary point $Q$ has only one internal neighbour point $P$.

⇛ The algorithm:

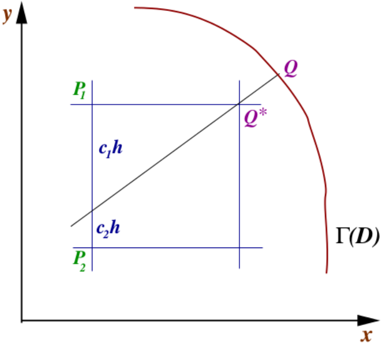
- We start the walk from a internal point $(X, Y)$ and we assign to it a weight: $W = 1$.
- If a particle at a given moment is sitting on the boundary then with probability $p$ it goes back to previous point $P$ and gets a weight $W \cdot \phi(Q)$ and with probability $(1-p)$ it finishes the walk and gets a weight of $W \cdot \psi(Q)$.
- For each trajectory we assign a value equal to the weight of the last point. So for example if the trajectory: $Q^{(1)}, Q^{(2)}, Q^{(3)}, ..., Q^{(k)}$ we will assign the number:

$$\phi(Q^{(1)})\phi(Q^{(2)})\phi(Q^{(3)})...\phi(Q^{(k-1)})\psi(Q^{(k)})$$

⇛ One again this is only for 1 neighbour point $P$ and that the normal of the boundary is parallel to the grid!

⇛ The general case is more difficult!

# More general case



⇛ The boundary conditions:

$$f(Q)\frac{1}{h\sqrt{1+c_1^2}}\left[c_2u(P_1)+c_1u(P_2)-u(Q^*\right]+$$
$$g(Q)u(Q^*)=h(Q)$$

⇛ The trick:

$$\phi_1(Q^*)=\frac{c_1f(Q)}{p_1\left[f(Q)-h\sqrt{1+c_1^2}\right]}$$

$$\phi_2(Q^*)=\frac{c_2f(Q)}{p_2\left[f(Q)-h\sqrt{1+c_1^2}\right]}$$

$$\psi_3(Q^*)=-h\frac{\sqrt{c_1^2+1}h(Q))}{p_3\left[f(Q)-h\sqrt{1+c_1^2}\right]}$$

⇛ Putting above new variables we get:

$$u(Q^*)=p_1\phi_1(Q^*)u(P_1)+p_2\phi_2(Q^*)u(P_2)+p_3\psi(Q^*)$$

⇛ We will interpret the $p_1$, $p_2$, $p_3$ numbers as probability.

# More general case, continuation

⇛ The rules of random walk:

- The particle starts in $(X, Y)$ inside the domain with weight: $W = 1$.
- If at some point in time the particle hits the boundary in point $Q^*$:
  - With probability $p_1$ it goes to point $P_1$ and the weight is $W \cdot \phi_1(Q^*)$
  - With probability $p_2$ it goes to point $P_2$ and the weight is $W \cdot \phi_2(Q^*)$
  - With probability $p_3$ it stops the walk and the weight is $W \cdot \psi(Q^*)$
- For each trajectory we assign the weight at the end point.

# Backup