Monte Carlo in Movie Production

Ray Tracing and Sampling

Slavomir Kaslev
slavomir.kaslev@gmail.com

December 17, 2016

About me

About me

- My name is Slavomir Kaslev

About me

- My name is Slavomir Kaslev
- Software engineer

About me

- My name is Slavomir Kaslev
- Software engineer
- I work at Worldwide FX as Head of R&D

About Worldwide FX

About Worldwide FX

- Founded in 2001

About Worldwide FX

- Founded in 2001
- The first VFX studio in Bulgaria

About Worldwide FX

- Founded in 2001
- The first VFX studio in Bulgaria
- $\sim$ 250 visual artists

About Worldwide FX

- Founded in 2001
- The first VFX studio in Bulgaria
- $\sim$ 250 visual artists
- More than 115 feature film projects completed

Studio demo reel

Software we use

Software we use

- Linux on workstations, farm nodes and servers

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations
- Avid, Blender, Mudbox, Mari, Maya, Houdini, Katana, RenderMan, Nuke, RV, Massive, Photoshop, Unity, ZBrush, 3DEqualizer, ...

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations
- Avid, Blender, Mudbox, Mari, Maya, Houdini, Katana, RenderMan, Nuke, RV, Massive, Photoshop, Unity, ZBrush, 3DEqualizer, ...
- In-house tools

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations
- Avid, Blender, Mudbox, Mari, Maya, Houdini, Katana, RenderMan, Nuke, RV, Massive, Photoshop, Unity, ZBrush, 3DEqualizer, ...
- In-house tools
- “Software is eating the world”

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations
- Avid, Blender, Mudbox, Mari, Maya, Houdini, Katana, RenderMan, Nuke, RV, Massive, Photoshop, Unity, ZBrush, 3DEqualizer, ...
- In-house tools
- "Software is eating the world" Why?

Software we use

- Linux on workstations, farm nodes and servers
- Mac OS and Windows on some workstations
- Avid, Blender, Mudbox, Mari, Maya, Houdini, Katana, RenderMan, Nuke, RV, Massive, Photoshop, Unity, ZBrush, 3DEqualizer, ...
- In-house tools
- "Software is eating the world" Why? Hint: Dependent Types

Rendering

Rendering

- Renderer is a software that generates an image from 3D scene description

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc
- In-house: Hyperion at Disney, Manuka at Weta Digital, Plume at ILM, Blue Sky, etc

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc
- In-house: Hyperion at Disney, Manuka at Weta Digital, Plume at ILM, Blue Sky, etc
- Ray tracing won at least in the movie industry.

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc
- In-house: Hyperion at Disney, Manuka at Weta Digital, Plume at ILM, Blue Sky, etc
- Ray tracing won at least in the movie industry. Why?

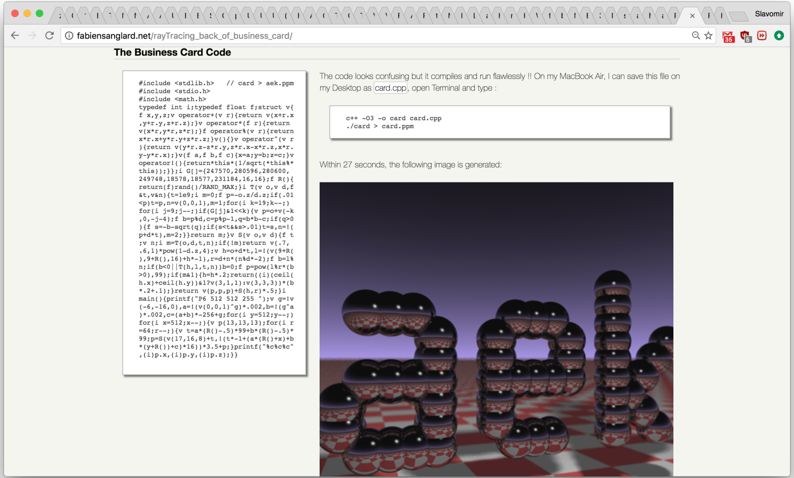
Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc
- In-house: Hyperion at Disney, Manuka at Weta Digital, Plume at ILM, Blue Sky, etc
- Ray tracing won at least in the movie industry. Why? Computer time is cheap, people time is expensive.

Rendering

- Renderer is a software that generates an image from 3D scene description
- 3delight, Arnold, RenderMan, V-Ray, mental ray, Maxwell, Redshift, etc
- Open source: Cycles, pbrt, appleseed, Mitsuba, LuxRender, etc
- In-house: Hyperion at Disney, Manuka at Weta Digital, Plume at ILM, Blue Sky, etc
- Ray tracing won at least in the movie industry. Why? Computer time is cheap, people time is expensive.
- "Physically Based Rendering: From Theory to Implementation" by Matt Pharr and Greg Humphreys

Bussiness card ray tracer



The Rendering Equation

$$L(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L(t(p, \omega_i), -\omega_i) \left|\cos\theta_i\right| d\omega_i$$

Analytical solution: Operator formulation

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

- What is the solution?

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

- What is the solution?

$$\begin{aligned} (\boldsymbol{I} - \boldsymbol{T})L &= L_e \\ L &= (\boldsymbol{I} - \boldsymbol{T})^{-1} L_e \end{aligned}$$

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

- What is the solution?

$$(\boldsymbol{I} - \boldsymbol{T})L = L_e$$
$$L = (\boldsymbol{I} - \boldsymbol{T})^{-1} L_e$$

- Power series representation: $L = L_e + \boldsymbol{T} L_e + \boldsymbol{T}^2 L_e + \dots$

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

- What is the solution?

$$\begin{aligned}(\boldsymbol{I} - \boldsymbol{T})L &= L_e \\ L &= (\boldsymbol{I} - \boldsymbol{T})^{-1} L_e\end{aligned}$$

- Power series representation: $L = L_e + \boldsymbol{T} L_e + \boldsymbol{T}^2 L_e + \ldots$

$$L = \sum_{n=0}^{\infty} \boldsymbol{T}^n L_e$$

Analytical solution: Operator formulation

- The light transport operator

$$L = L_e + \boldsymbol{T} L$$

- What is the solution?

$$(\boldsymbol{I} - \boldsymbol{T}) L = L_e$$
$$L = (\boldsymbol{I} - \boldsymbol{T})^{-1} L_e$$

- Power series representation: $L = L_e + \boldsymbol{T} L_e + \boldsymbol{T}^2 L_e + ...$

$$L = \sum_{n=0}^{\infty} \boldsymbol{T}^n L_e$$

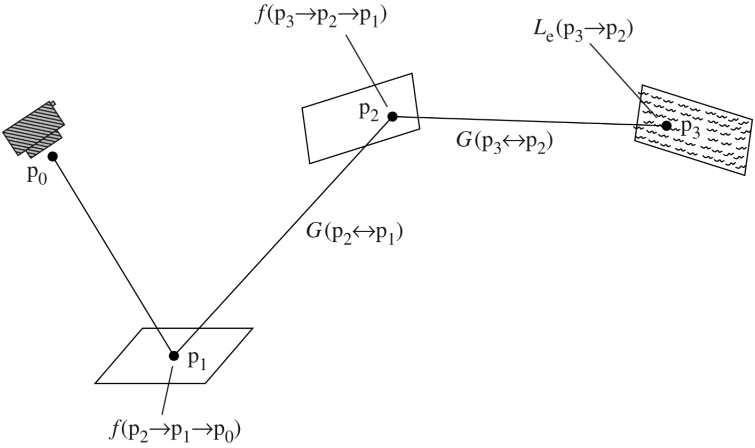
- The solution operator

$$\boldsymbol{S} = (\boldsymbol{I} - \boldsymbol{T})^{-1}$$
$$L = \boldsymbol{S} L_e$$

Numerical solution: Integral over paths

$$L(p' \to p) = L_e(p' \to p) + \int_A f(p'' \to p' \to p) L(p'' \to p') G(p'' \leftrightarrow p') dA(p'')$$

Generating paths



Strategies for generating paths

Strategies for generating paths

- Light tracing

Strategies for generating paths

- Light tracing
- Path tracing

Strategies for generating paths

- Light tracing
- Path tracing
- Bidirectional path tracing

Strategies for generating paths

- Light tracing
- Path tracing
- Bidirectional path tracing
- Vertex connection and merging (VCM)

Strategies for generating paths

- Light tracing
- Path tracing
- Bidirectional path tracing
- Vertex connection and merging (VCM)
- Still an active topic of research

Monte Carlo method

Monte Carlo method

- Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results

Monte Carlo method

- Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results
- Enrico Fermi first experimented with the Monte Carlo method while studying neutron diffusion in the 1930s

Monte Carlo method

- Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results
- Enrico Fermi first experimented with the Monte Carlo method while studying neutron diffusion in the 1930s
- The modern version of the Monte Carlo method was invented in the late 1940s by Stanislaw Ulam

Monte Carlo method

- Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results
- Enrico Fermi first experimented with the Monte Carlo method while studying neutron diffusion in the 1930s
- The modern version of the Monte Carlo method was invented in the late 1940s by Stanislaw Ulam
- Monte Carlo methods were central to the simulations required for the Manhattan Project

Monte Carlo method I

Suppose we want to find the value of I where

$$I = \int_{x \in [0,1]^s} f(x) dx$$

Monte Carlo method I

Suppose we want to find the value of I where

$$I = \int_{x \in [0,1]^s} f(x) dx$$

and also suppose we have a random variable $X \in [0,1]^s$ with probability density function $p(x)$.

Monte Carlo method I

Suppose we want to find the value of I where

$$I = \int_{x \in [0,1]^s} f(x) dx$$

and also suppose we have a random variable $X \in [0, 1]^s$ with probability density function $p(x)$.

What can we say about the expected value of $f(X)$?

Monte Carlo method I

Suppose we want to find the value of I where

$$I = \int_{x \in [0,1]^s} f(x)dx$$

and also suppose we have a random variable $X \in [0,1]^s$ with probability density function $p(x)$.

What can we say about the expected value of $f(X)$?

Well by definition of expected value we know that

$$\mathbf{E}[f(X)] = \int_{x \in [0,1]^s} f(x)p(x)dx$$

Monte Carlo method II

What about the expected value of $\frac{f(X)}{p(X)}$?

Monte Carlo method II

What about the expected value of $\frac{f(X)}{p(X)}$?

Let's see

$$\begin{aligned}\mathbf{E}[\frac{f(X)}{p(X)}] &= \int_{x\in[0,1]^s} \frac{f(x)}{p(x)} p(x)dx \\ &= \int_{x\in[0,1]^s} f(x)dx = I\end{aligned}$$

Monte Carlo method II

What about the expected value of $\frac{f(X)}{p(X)}$?

Let's see

$$\mathbf{E}[\frac{f(X)}{p(X)}] = \int_{x \in [0,1]^s} \frac{f(x)}{p(x)} p(x) dx$$
$$= \int_{x \in [0,1]^s} f(x) dx = I$$

Therefore

$$I = \mathbf{E}[\frac{f(X)}{p(X)}]$$

Monte Carlo method III

Monte Carlo method III

The general idea is to approximate the integral we're interested in

$$I = \int_{x \in [0,1]^s} f(x) dx$$

Monte Carlo method III

The general idea is to approximate the integral we're interested in

$$I = \int_{x \in [0,1]^s} f(x)dx$$

by a sum

$$\widetilde{I_N} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

Monte Carlo method III

The general idea is to approximate the integral we're interested in

$$I = \int_{x \in [0,1]^s} f(x) dx$$

by a sum

$$\widetilde{I_N} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

It can be shown that $\widetilde{I_N}$ is an unbiased estimator of I which converges to the correct value as $N \to \infty$

Monte Carlo method III

The general idea is to approximate the integral we're interested in

$$I = \int_{x \in [0,1]^s} f(x)dx$$

by a sum

$$\widetilde{I_N} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

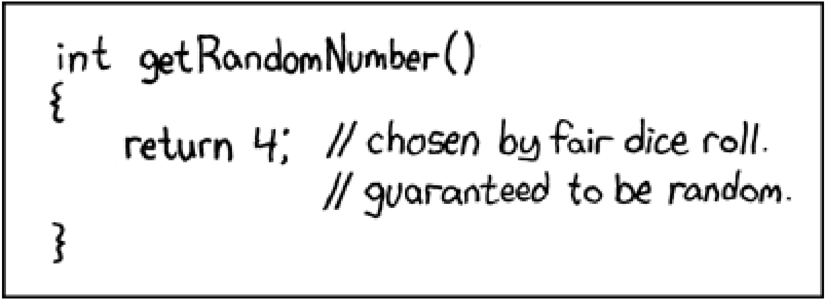
It can be shown that $\widetilde{I_N}$ is an unbiased estimator of I which converges to the correct value as $N \to \infty$

$$\mathbf{E}[\widetilde{I_N}] = I$$

$$\mathbf{V}[\widetilde{I_N}] = \frac{1}{N} \mathbf{V}[\frac{f(X)}{p(X)}]$$

$$\sigma[\widetilde{I_N}] = \frac{1}{\sqrt{N}} \sigma[\frac{f(X)}{p(X)}]$$

Random Numbers: xkcd #221



Quasi-Monte Carlo Method

Quasi-Monte Carlo Method

- The Koksma-Hlawka inequality

$$\left| \widetilde{I_N} - I \right| \leq \mathbf{V}[f] D_N^*(x_1, ..., x_N)$$

Quasi-Monte Carlo Method

- The Koksma-Hlawka inequality

$$\left| \widetilde{I_N} - I \right| \leq \mathbf{V}[f] D_N^*(x_1, ..., x_N)$$

- Roughly speaking the star discrepancy of a point set $D_N^*(x_1, ..., x_N)$ is a measure of how uniformly the sequence $x_1, ..., x_N$ is distributed in $[0, 1]^s$

Quasi-Monte Carlo Method

- The Koksma-Hlawka inequality

$$\left|\widetilde{I_N} - I\right| \leq \mathbf{V}[f] D_N^*(x_1, \ldots, x_N)$$

- Roughly speaking the star discrepancy of a point set $D_N^*(x_1, \ldots, x_N)$ is a measure of how uniformly the sequence $x_1, \ldots, x_N$ is distributed in $[0, 1]^s$
- The main discrepancy conjecture

$$D_N^*(x_1, \ldots, x_N) \geq c_s \frac{(\ln N)^s}{N}$$

Quasi-Monte Carlo Method

- The Koksma-Hlawka inequality

$$\left|\widetilde{I_N} - I\right| \leq \mathbf{V}[f] D_N^*(x_1, ..., x_N)$$

- Roughly speaking the star discrepancy of a point set $D_N^*(x_1, ..., x_N)$ is a measure of how uniformly the sequence $x_1, ..., x_N$ is distributed in $[0,1]^s$
- The main discrepancy conjecture

$$D_N^*(x_1, ..., x_N) \geq c_s \frac{(\ln N)^s}{N}$$

  - Proved for $s \leq 2$ by W. M. Schmidt. The problem is stil open in higher dimensions.

Quasi-Monte Carlo Method

- The Koksma-Hlawka inequality

$$\left| \widetilde{I_N} - I \right| \leq \mathbf{V}[f] D_N^*(x_1, ..., x_N)$$

- Roughly speaking the star discrepancy of a point set $D_N^*(x_1, ..., x_N)$ is a measure of how uniformly the sequence $x_1, ..., x_N$ is distributed in $[0, 1]^s$
- The main discrepancy conjecture

$$D_N^*(x_1, ..., x_N) \geq c_s \frac{(\ln N)^s}{N}$$

  - Proved for $s \leq 2$ by W. M. Schmidt. The problem is stil open in higher dimensions.
- Low discrepancy sequences: van der Corput, Halton, Hammersley, Sobol and others

$$D_N^*(x_1, ..., x_N) \leq C \frac{(\ln N)^s}{N}$$

Let's design a sampler: fract demo

Let's design a sampler: fract demo

- Python, NumPy, Z3 Theorem prover

Let's design a sampler: fract demo

- Python, NumPy, Z3 Theorem prover
- Z3 is an open source Satisfiability Modulo Theories (SMT) solver by Microsoft Research

Let's design a sampler: fract demo

- Python, NumPy, Z3 Theorem prover
- Z3 is an open source Satisfiability Modulo Theories (SMT) solver by Microsoft Research
- https://github.com/skaslev/fract/blob/master/mc-production.ipynb

Let's design a sampler: fract demo

- Python, NumPy, Z3 Theorem prover
- Z3 is an open source Satisfiability Modulo Theories (SMT) solver by Microsoft Research
- https://github.com/skaslev/fract/blob/master/mc-production.ipynb
- https://github.com/skaslev/pyman

Future

Future

- "It's hard to make predictions, especially about the future."

Future

- "It's hard to make predictions, especially about the future."
  Niels Bohr

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$?

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe.

Future

- "It's hard to make predictions, especially about the future."
  Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning
- Better mathematical models

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning
- Better mathematical models
  - Renderer based on Quantum Electrodynamics (QED) rather than classical approximations

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning
- Better mathematical models
  - Renderer based on Quantum Electrodynamics (QED) rather than classical approximations
  - "QED: The Strange Theory of Light and Matter" by Richard Feynman

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning
- Better mathematical models
  - Renderer based on Quantum Electrodynamics (QED) rather than classical approximations
  - "QED: The Strange Theory of Light and Matter" by Richard Feynman
  - "Richard Feynman Lecture on Quantum Electrodynamics: QED" https://www.youtube.com/watch?v=LPDP_8X5Hug

Future

- "It's hard to make predictions, especially about the future." Niels Bohr
- Faster Monte Carlo convergence
  - Can we do better than $O(N^{-1/2})$? Maybe. Maybe not.
  - "Advanced mathematical methods for scientists and engineers" by Carl Bender
- Noise filtering and machine learning
- Better mathematical models
  - Renderer based on Quantum Electrodynamics (QED) rather than classical approximations
  - "QED: The Strange Theory of Light and Matter" by Richard Feynman
  - "Richard Feynman Lecture on Quantum Electrodynamics: QED" https://www.youtube.com/watch?v=LPDP_8X5Hug

The Feynman Algorithm

The Feynman Algorithm

- Write down the problem

The Feynman Algorithm

- Write down the problem
- Think real hard

The Feynman Algorithm

- Write down the problem
- Think real hard
- Write down the solution

$$Ax = b$$

$Ax = b$

- Write down the problem

$Ax = b$

- Write down the problem
- Rewrite the problem as $Ax = b$

$Ax = b$

- Write down the problem
- Rewrite the problem as $Ax = b$
- Use off the shelf solver or write a specialized one

$Ax = b$

- Write down the problem
- Rewrite the problem as $Ax = b$
- Use off the shelf solver or write a specialized one

"All problems in computer graphics can be solved with a matrix inversion." Jim Blinn

Eigenvalue problems

Eigenvalue problems

- Write down the problem

Eigenvalue problems

- Write down the problem
- Rewrite the problem as $Ax = \lambda x$

Eigenvalue problems

- Write down the problem
- Rewrite the problem as $Ax = \lambda x$
- Use off the shelf solver or write a specialized one

Perturbation theory

Perturbation theory

- Write down the problem

Perturbation theory

- Write down the problem
- Insert a "small" parameter ϵ in so that the problem is trivial when $\epsilon = 0$

Perturbation theory

- Write down the problem
- Insert a “small” parameter ϵ in so that the problem is trivial when $\epsilon = 0$
- Guess the solution has the form $x = \sum_{n=0}^{\infty} a_n \epsilon^n$

Perturbation theory

- Write down the problem
- Insert a "small" parameter ϵ in so that the problem is trivial when $\epsilon = 0$
- Guess the solution has the form $x = \sum_{n=0}^{\infty} a_n \epsilon^n$
- Plug that back in the equation and solve for a_n

Perturbation theory

- Write down the problem
- Insert a “small” parameter ϵ in so that the problem is trivial when $\epsilon = 0$
- Guess the solution has the form $x = \sum_{n=0}^{\infty} a_n \epsilon^n$
- Plug that back in the equation and solve for a_n
- Sum the series and substitue $\epsilon = 1$

Further reading

- "Physically Based Rendering: From Theory to Implementation" by Matt Pharr and Greg Humphreys
- "Robust Monte Carlo Methods for Light Transport Simulation", Eric Veach, Ph.D. dissertation
- "Light Transport Simulation with Vertex Connection and Merging" by Iliyan Georgiev, Jaroslav Kivnek, Tom Davidovi, Philipp Slusallek
- "Random Number Generation and Quasi-Monte Carlo Methods" by Harald Niederreiter
- "Quantum Mechanics and Path Integrals" by Richard Feynman
- "An Introduction to the Analysis of Algorithms" by Robert Sedgewick and Phillipe Flajolet
- "Mathematical Physics", Carl Bender, https://www.youtube.com/watch?v=LYNOGk3ZjFM
- "fract", source code from this presentation, https://github.com/skaslev/fract

Questions?

Thank you