Universiteit Antwerpen

Masterproef scriptie

Unbiased Monte Carlo for Recursive Integrals

Auteur: *Isidoor Pinillo Esquivel*

Promotor: *Wim Vanroose*

Academiejaar 2022-2023

Contents

Abstract

We will write this at the end.

1 Introduction

1.1 Introductory Example

To get familiar with Monte Carlo for estimating recursive integrals we demonstrate it on following problem:

$$y' = y, y(0) = 1. \tag{1}$$

By integrating both sides of (1) following integral equation can be derived:

$$y(t) = 1 + \int_0^t y(s)ds. \tag{2}$$

Equation (2) is a recursive integral equation or to be more specific a linear Volterra integral equation of the second type. By naively using Monte Carlo on the recursive integral of equation (2) one derives following estimator:

$$Y(t) = 1 + ty(Ut).$$

where $U = \text{Uniform}(0, 1)$. If y is well behaved then $E[Y(t)] = y(t)$ but we can't calculate $Y(t)$ without accesses to $y(s), s < t$. Notice that we can replace y by a unbiased estimator of it without changing $E[Y(t)] = y(t)$ by the law of total expectance ($E[X] = E[E[X|Z]]$). By replacing y by Y itself we obtain a recursive expression for Y:

$$Y(t) = 1 + tY(Ut). \tag{3}$$

Equation (3) is a recursive random variable equation. If you would implement equation (3) with recursion it will run indefinitely. A biased way of around this is by approximating $Y(t) \approx 1$ near $t = 0$. Later we discuss Russian roulette (refRussian roulette) which can be used as an unbiased stopping mechanism.

Python Code 1.1.1 (implementation of (3))

```python
from random import random as U
def Y(t, eps): return 1 + t*Y(U()*t, eps) if t > eps else 1
def y(t, eps, nsim):
    return sum(Y(t, eps) for _ in range(nsim))/nsim
print(f"y(1) approx {y(1,0.01,10**3)}")
# y(1) approx 2.710602603240193
```

An issue with (1.1.1) is that the variance explodes when t increases. Which we later solve in the section on ODEs. Note that (1.1.1) keeps desirable properties from unbiased Monte Carlo methods such as: being embarrassingly parallel, robustness and simple error estimates.

1.2 Contributions

We write this at the end.

1.3 Related Work

work on

- alternative methods for recursive integrals
- MC work on ODEs
- MC work on PDEs
- WoS

This is just to give a general overview we probably reference specific ideas when we first introduce them.

2 Background

2.1 Modifying Monte Carlo

Introduces Russian roulette, splitting, control variates, importance sampling and maybe quasi Monte Carlo with the $y' = y$ example.

2.2 Monte Carlo Trapezoidal Rule

comparing normal vs Monte Carlo trapezoidal rule and highlighting the "half variance phenomenon". + maybe integrating polynomials for intuition

2.3 Unbiased Non-Linearity

Example 2.3.1 ($y' = y^2$)
see python note book

Example 2.3.2 ($e^{E[X]}$)
see python note book

2.4 Recursion

Example 2.4.1 (coupled recursion)
example with $y' = y$ (I need to redo this example)

Example 2.4.2 (recursion in recursion)
maybe induction in induction proof example

Example 2.4.3 (tail recursion)
discuss problems with implementing recursion and solutions.
inverse problem example

2.5 Green Functions

green function stuff that we will be needing, we aren't sure in how much detail we're going to go.

Example 2.5.1 (numerical green functions)
There will be probably some green functions that we need that don't have an analytic expression yet.

3 1-Dimensional Recursive Integrals

3.1 Linear Recursive Integrals

We have algo in mind for this case based on coupled recursion on disjunct sets.

3.2 IVPs ODEs

An IVP example probably using DRRMC maybe compare it to parareal. Maybe also non-linear algo

3.3 BVPs ODEs

A BVP example using yet another algo that hopefully has the half variance phenomenon.

4 Higher Dimensional Recursive Integrals

4.1 Complicated Geometry

Example 4.1.1 (nasty 2D integral)
2D integral that is difficult because of its geometry

4.2 Recursive Brownian Motion

WoS like way to simulate Brownian motion which is related to the green function of the heat equation

Example 4.2.1 (recursive Brownian motion)
see period5

4.3 Heat Equation

a geometric robust way to solve the heat equation and maybe a higher order method to solve the heat equation

4.4 Wave Equation

probably won't get to it

5 Appendix

Derivation of the green functions and some expressions.