Anton Bobkov

Contact Information

Graduate Student
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90095-1555 USA

E-mail:
antongml@gmail.com
bobkov@math.ucla.edu
Website:
www.math.ucla.edu/~bobkov/
Phone: (408)813-6331

Education

University of California, Los Angeles *(graduate)* **Fall 2011 to present**
PhD, Mathematics (in progress)

Advisor: Matthias Aschenbrenner
Research interests: Mathematical logic, model theory, NIP theories, vc-density

University of California, Los Angeles *(undergraduate)* **Graduated Spring 2011**

- *B.S.* in Mathematics, *B.A.* in Physics
- Sherwood Prize
- Departmental Highest Honors in Mathematics, College Honors
- GPA: 3.82 (Magna Cum Laude)
- William Lowell Putnam Mathematics Competition
  - 2008 - score 30
  - 2009 - score 19

Undergraduate Research

Cryptography REU at Northern Kentucky University **Summer 2009**
Implemented a variant of MXL algorithm in computational algebra system MAGMA

Research assistant for Vladimir Vassiliev **2008 - 2011**
I did various numerical simulations in C++ for AGIS gamma-ray telescope. This included forward and inverse kinematics for Stewart platform, ray casting focusing simulations, and laser calibration. I have also worked on network interfacing with Gumstix boards using CORBA as well as installing and configuring a custom linux kernel.

Teaching

Intermediate C++ Programming, Linear Algebra, Calculus

Papers

Bobkov, A. *VC-density for trees*, in preparation

Software Experience

Unix-like systems
I am comfortable working in command line environment, including tasks such as

- installing and managing web-server, repository server, ssh server
- code building, editing, and version control

Languages	C++, C#, bash, Java, php, MAGMA
Code management	CMake, Makefile, git, subversion, Visual Studio, Unity3D
Standards	TCP/IP, .NET, CORBA

Independent Projects

Burn and Turn **2008 - 2011**

Cross platform arcade style video game featured on Kotaku and IndieGames. It was coded in C++ and used OpenGL as a backend for graphics. It was created by a team of three people over a course of four years and released on iOS and Android markets.

Self Balancing Robot **Summer 2012**

A vertical self-balancing robot ran by an arduino controller coded in C++. A numerical simulation was used to determine weight distribution. Robot's position is determined by data from an accelerometer and a gyroscope combined through a Kalman filter. Balancing is done through a DC motor using PID controller.

UCLA Graduate Student Wiki **Summer 2014**

Decentralized Networking Online Game **Fall 2014 - Present**