Bachelor- / Masterarbeit

Numerische Stabilität großer Zahlen während der Gram-Schmidt Zerlegung und innerhalb des LLL-Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Institute for Scientific Computing

SCIENTIFIC COMPUTING

Dr. Michael Burger
Ansprechpartner

Alexanderstr. 2
64289 Darmstadt

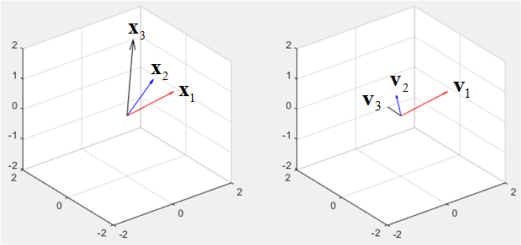
Tel.: +49 6151 16 - 76991
Fax: +49 6151 16 - 25345

michael.burger@tu-darmstadt.de

https://www.sc.informatik.tu-darmstadt.de/

Datum
7. Juli 2020

Im Rahmen Gitterbasierter Kryptographie und deren Kryptoanalyse müssen während des LLL-Algorithmus zur Verbesserung der Basisqualität [1] und der darin inhärenten Gram-Schmidt Orthogonalisierung (QR Zerlegung) große Zahlen verarbeitet werden. Dies führt bei Nichtbeachtung zum Absturz des Programms (z.B. divison by zero) oder zu falschen Ergebnissen.



Gram-Schmidt Orthogonalisierung in drei Dimensionen.

Angepasste Verfahren versuchen durch Neuordnung der Vektoren während der Orthogonalisierung (s.h. z.B. [2]) und durch andere algorithmische Anpassungen die Stabilität der Verfahren zu steigern.

Implementierungen mit beliebig großen Ganzzahldatentypen und Gleitkommazahlen mit erhöhter Genauigkeit sind numerisch relativ stabil, haben aber eine hohe Laufzeit. Daher existieren angepasste Implementierungen, die versuchen den Genauigkeitsverlust abzuschätzen und korrigierend eingreifen, indem sie zum Beispiel bei erwartetem Genauigkeitsverlust Iterationen mit Datentypen höherer Präzession zum „Glätten“ ausführen, während ansonsten hauptsächlich mit kleineren Datentypen gerechnet wird.

Im Rahmen der Arbeit sollen entsprechende effiziente und stabile Implementierungen für Orthogonalisierung und den LLL Algorithmus (sowie ggf. auch für Varianten von diesem) als Building Blocks zur Integration in ein größeres Framework erstellt werden. Hierzu ist zu Beginn eine Literaturrecherche über bestehende Algorithmen/Verfahren und existierende Implementierungen durchzuführen. Anschließend sind die dabei identifizierten vielversprechendsten Methodiken in Code umzusetzen und die Effizienz zu validieren und optimieren, sowie die Stabilität zu beweisen.

Empfohlene Vorkenntnisse:

- Gute Programmierkenntnisse in C++
- Erfahrung mit Performance-Analyse und -Tuning (Intel VTune, Valgrind)
- Grundkenntnisse in Linearer Algebra und Computerarithmetik

[1] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261(ARTICLE):515–534, 1982.

[2] James W Daniel, Walter Bill Gragg, Linda Kaufman, and Gilbert W Stewart. Reorthogonalization and stable algorithms for updating the gram-schmidt qr factorization. *Mathematics of Computation*, 30(136):772–795, 1976.