QUADRATIC SIEVE
FACTORIZING METHOD IN C
Outline

- Quadratic Sieve Algorithm
- Implementation
- Results
- Future Work
Quadratic Sieve

- The goal of the Quadratic Sieve is the factor a very large number in a short time.
- If \( N \) is the number to be factored, QS looks for \( x \) and \( y \) such that:
  - \( x \) is not congruent to \( +\pm y \mod n \)
  - \( x^2 \) is congruent to \( y^2 \mod n \)
Quadratic Sieve

- **Build Factor Base**
  - Small prime numbers
  - Legendre (N/P) = 1
  - “Smooth”
  - Recommended Base Size:
    - $E^{\sqrt{\ln(N) \times \ln(\ln(N))}}$
Quadratic Sieve

define:

- \( Q(x) = (x + \sqrt{n})^2 - n = x^2 - n \)
- Begin taking \( x \) values and computing \( Q(x) \)
- Determine if it factors factor base (if it does it has smoothness)
- if smooth then For each prime in factor base solve
  - \( Q(x) = s_1^2 \) is congruent to 0 mod \( p \)
  - \( s_2 = p - s_1 \)
  - Can Solve with Tonelli-Shanks Algorithm
Quadratic Sieve

- take subinterval
- put $Q(x_i)$ into array for each $x$
- for each $p$ start at $s_1$ and $s_2$
  - divide out highest power of $p$ for each element
  - record powers (mod 2) in an array
  - make vector for each factorable $Q(x)$
- (each entry corresponds to unique prime in factor base)
Quadratic Sieve

- Array elements == 1, factor over factor base.
- Put Vector of powers of primes into Matrix A
- find solutions to:
  - $Q(x_1)e_1 + Q(x_2)e_2 + ... + Q(x_k)e_k$
  - where the $e_i$ are either 0 or 1, so:
  - $a_1e_2 + a_22 + ... + akek$ is congruent to 0 mod 2
Quadratic Sieve

- Solve with Gaussian Elimination
- Check solution vectors if corresponding product produces factor.
Implementation

- Libraries:
  - GMP (GNU Multiple Precision)
  - MPFR (Multiple-Precision Floating point w/ correct Rounding)
  - GSL (GNU Scientific Library)
Results

- ***Collecting***
- -tonnelli solving $x^2 = 24961 \mod 23$
- $s1 = 12$
- $s2 = 11$
Future Work

- Fix Gaussian problem.
- Finish Application