A New Solution to the Normalization Problem

Mahdi Javadi (sjavadi@cecm.sfu.ca)

CECM, Simon Fraser University

Problem Statement

- We use Zippel's sparse interpolation to compute $g = \gcd(f_1, f_2)$.
- $f_1, f_2 \in F[x, y, \ldots]$.

Problem Statement

- We use Zippel's sparse interpolation to compute $g = \gcd(f_1, f_2)$.
- $f_1, f_2 \in F[x, y, \ldots]$.
- Normalization Problem. Example:
  - Suppose $g = (2y+1)x^2 + (y+2)$ and $p = 7$
  - The form is $g_f = (Ay+B)x^2 + (Cy+D)$
  - $g(y=1) = x^2 + 6$, $g(y=2) = x^2 + 1$
  - After solving the system of equations: $\{A=0, B=1, C=2, D=4\}$
  - The result is wrong.

Problem Statement

- We use Zippel's sparse interpolation to compute $g = \gcd(f_1, f_2)$.
- $f_1, f_2 \in F[x, y, \ldots]$.
- Normalization Problem. Example:
  - Suppose $g = (2y + 1)x^2 + (y + 2)$ and $p = 7$
  - The form is $g_f = (Ay + B)x^2 + (Cy + D)$
  - $g(y = 1) = x^2 + 6$, $g(y = 2) = x^2 + 1$
  - After solving the system of equations: $\{A = 0, B = 1, C = 2, D = 4\}$
  - The result is wrong.
- More precisely: When $\mathrm{lc}_x(g)$ has at least two terms, we can't use Zippel's method directly.

First Solution

- The first solution is presented by de Kleine, Monagan and Wittkopf in 2005.
- The idea is to scale each univariate image with an *unknown* scaling factor.

First Solution

- The first solution is presented by de Kleine, Monagan and Wittkopf in 2005.
- The idea is to scale each univariate image with an *unknown* scaling factor.
- Example:
  - Consider $g_f = (Ay^2 + B)x^3 + Cy + D$ and $p = 17$.
  - $g(y = 1) = m_1(x^3 + 12) = x^3 + 12$, $g(y = 2) = m_2(x^3 + 8)$ and $g(y = 3) = m_3(x^3)$.
  - m_2 and m_3 are unknowns. We set $m_1 = 1$.
  - Solve the system: $\{A = 7, B = 11, C = 11, D = 1, m_2 = 5, m_3 = 6\}$.

First Solution

- The first solution is presented by de Kleine, Monagan and Wittkopf in 2005.
- The idea is to scale each univariate image with an *unknown* scaling factor.
- Example:
  - Consider $g_f = (Ay^2 + B)x^3 + Cy + D$ and $p = 17$.
  - $g(y=1) = m_1(x^3 + 12) = x^3 + 12$, $g(y=2) = m_2(x^3 + 8)$ and $g(y=3) = m_3(x^3)$.
  - m_2 and m_3 are unknowns. We set $m_1 = 1$.
  - Solve the system: $\{A = 7, B = 11, C = 11, D = 1, m_2 = 5, m_3 = 6\}$.
- Suppose coefficients of g have term counts $n_1, \ldots, n_s$ and $n_{max} = \max(n_1, \ldots, n_s)$.
- The number of images needed is: $\max\left(n_{max}, \left\lceil \frac{(\sum_{i=1}^{s} n_i) - 1}{s-1} \right\rceil\right)$.

First Solution (contd.)

- Example: Let $g_f = (Ay^2 + B)x^2 + (Cyz^2 + D)x + Ez^2 + F$.

$$
\begin{bmatrix}
c & c & & & & & 1 & \\
c & c & & & & & & 1 \\
& & c & c & & & c & \\
& & c & c & & & & c \\
& & & & c & c & c & \\
& & & & c & c & & c
\end{bmatrix}
\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ 1 \\ m_2 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

First Solution (contd.)

- Example: Let $g_f = (Ay^2 + B)x^2 + (Cyz^2 + D)x + Ez^2 + F$.

$$\begin{bmatrix} c & c & & & & & 1 & \\ c & c & & & & & & 1 \\ & & c & c & & & c & \\ & & c & c & & & & c \\ & & & & c & c & c & \\ & & & & c & c & & c \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ 1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Using the trick the total cost is: $O(n_1^3 + \cdots + n_s^3)$.

First Solution (contd.)

- Example: Let $g_f = (Ay^2 + B)x^2 + (Cyz^2 + D)x + Ez^2 + F$.

$$\begin{bmatrix} c & c & & & & & 1 & \\ c & c & & & & & & 1 \\ & & c & c & & & c & \\ & & c & c & & & & c \\ & & & & c & c & c & \\ & & & & c & c & & c \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ 1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Using the trick the total cost is: $O(n_1^3 + \cdots + n_s^3)$.
- First problem: the systems of linear equations are now dependent to each other.
  - This reduces the parallelism.

Vandermonde Matrix

- In 1990, Zippel presented a trick to solve the systems of linear equations (monic case) in $O(n_1^2 + \cdots + n_s^2)$ time and linear space.
- This is a significant gain compared to $O(n_1^3 + \cdots + n_s^3)$ time and quadratic space.
- The trick is to choose the evaluation points such that the systems of equations are Vandermonde Matrices.

Vandermonde Matrix

- In 1990, Zippel presented a trick to solve the systems of linear equations (monic case) in $O(n_1^2 + \cdots + n_s^2)$ time and linear space.
- This is a significant gain compared to $O(n_1^3 + \cdots + n_s^3)$ time and quadratic space.
- The trick is to choose the evaluation points such that the systems of equations are Vandermonde Matrices.
- Example: Suppose $g_f = Ay^2x^2 + (Byz^2 + Cy^2z + D)x + Ez^2 + F$.
  - We need three univariate images.
  - For $\alpha = 2$ and $\beta = 3$ let $(y_0 = 1, z_0 = 1), (y_1 = \alpha, z_1 = \beta), (y_2 = \alpha^2, z_2 = \beta^2)$.

Vandermonde Matrix

- In 1990, Zippel presented a trick to solve the systems of linear equations (monic case) in $O(n_1^2 + \cdots + n_s^2)$ time and linear space.
- This is a significant gain compared to $O(n_1^3 + \cdots + n_s^3)$ time and quadratic space.
- The trick is to choose the evaluation points such that the systems of equations are Vandermonde Matrices.
- Example: Suppose $g_f = Ay^2x^2 + (Byz^2 + Cy^2z + D)x + Ez^2 + F$.
  - We need three univariate images.
  - For $\alpha = 2$ and $\beta = 3$ let $(y_0 = 1, z_0 = 1), (y_1 = \alpha, z_1 = \beta), (y_2 = \alpha^2, z_2 = \beta^2)$.

$$\begin{pmatrix} 1 & 1 & 1 \\ 18 & 12 & 1 \\ 324 & 144 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ k_1 & k_2 & k_3 \\ k_1^2 & k_2^2 & k_3^2 \end{pmatrix} and \begin{pmatrix} 1 & 1 \\ 9 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k_1' & k_2' \end{pmatrix}$$

Vandermonde Matrix (contd.)

- Finding inverse of a Vandermonde matrix:

$$\begin{pmatrix} 1 & k_1 & k_1^2 \\ 1 & k_2 & k_2^2 \\ 1 & k_3 & k_3^2 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Vandermonde Matrix (contd.)

- Finding inverse of a Vandermonde matrix:

$$\begin{pmatrix} 1 & k_1 & k_1^2 \\ 1 & k_2 & k_2^2 \\ 1 & k_3 & k_3^2 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

- The jth element of the top row of the product of these matrices is:

$$a_{1j} + a_{2j}k_1 + a_{3j}k_1^2 = P_j(k_1)$$

Vandermonde Matrix (contd.)

- Finding inverse of a Vandermonde matrix:

$$\begin{pmatrix} 1 & k_1 & k_1^2 \\ 1 & k_2 & k_2^2 \\ 1 & k_3 & k_3^2 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

- The jth element of the top row of the product of these matrices is:

$$a_{1j} + a_{2j}k_1 + a_{3j}k_1^2 = P_j(k_1)$$

- And the product above is:

$$\begin{pmatrix} P_1(k_1) & P_2(k_1) & P_3(k_1) \\ P_1(k_2) & P_2(k_2) & P_3(k_2) \\ P_1(k_3) & P_2(k_3) & P_3(k_3) \end{pmatrix}$$

Vandermonde Matrix (contd.)

- Using this method (monic case) the total cost for solving systems of linear equations is $O(n_1^2 + \cdots + n_s^2)$.

Vandermonde Matrix (contd.)

- Using this method (monic case) the total cost for solving systems of linear equations is $O(n_1^2 + \cdots + n_s^2)$.
- Second problem with scaling factors (non-monic case):
  - Since the systems are dependent and we are using scaling factors as unknows, Zippel's trick can not be used.

Vandermonde Matrix (contd.)

- Using this method (monic case) the total cost for solving systems of linear equations is $O(n_1^2 + \cdots + n_s^2)$.
- Second problem with scaling factors (non-monic case):
  - Since the systems are dependent and we are using scaling factors as unknows, Zippel's trick can not be used.
- Motivation: Find a solution to the normalization problem such that the systems of equations could be solved independently and in quadratic time.

New Solution

- We will use the fact that we know the form of the leading coefficient.

New Solution

- We will use the fact that we know the form of the leading coefficient.
- Example:
  - Suppose $g_f = (Ay^2 + B)x^2 + (Cy + D)x + (Ey^3 + Fy^2 + G)$ and $p = 13$.
  - Let $y_0 = 1, y_1 = 5, y_2 = 12$ and we force $A = 1$.
  - $g(y = y_0) = x^2 + 9x + 7, g(y = y_1) = x^2 + 9x + 12, g(y = y_2) = x^2 + x + 6$.

New Solution

- We will use the fact that we know the form of the leading coefficient.
- Example:
  - Suppose $g_f = (Ay^2 + B)x^2 + (Cy + D)x + (Ey^3 + Fy^2 + G)$ and $p = 13$.
  - Let $y_0 = 1, y_1 = 5, y_2 = 12$ and we force $A = 1$.
  - $g(y = y_0) = x^2 + 9x + 7, g(y = y_1) = x^2 + 9x + 12, g(y = y_2) = x^2 + x + 6$.
  - Since $\mathrm{lc}_x(g) = y^2 + B$, we must scale each image by this evaluated at the corresponding evaluation point.
    - $g_0 = (1 + B)x^2 + 9(1 + B)x + 7(1 + B)$.
    - $g_1 = (12 + B)x^2 + 9(12 + B)x + 12(12 + B)$.
    - $g_2 = (1 + B)x^2 + (1 + B)x + 6(1 + B)$.

New Solution

- We will use the fact that we know the form of the leading coefficient.
- Example:
  - Suppose $g_f = (Ay^2 + B)x^2 + (Cy + D)x + (Ey^3 + Fy^2 + G)$ and $p = 13$.
  - Let $y_0 = 1, y_1 = 5, y_2 = 12$ and we force $A = 1$.
  - $g(y = y_0) = x^2 + 9x + 7, g(y = y_1) = x^2 + 9x + 12, g(y = y_2) = x^2 + x + 6$.
  - Since $\mathrm{lc}_x(g) = y^2 + B$, we must scale each image by this evaluated at the corresponding evaluation point.
    - $g_0 = (1 + B)x^2 + 9(1 + B)x + 7(1 + B)$.
    - $g_1 = (12 + B)x^2 + 9(12 + B)x + 12(12 + B)$.
    - $g_2 = (1 + B)x^2 + (1 + B)x + 6(1 + B)$.
  - $\Rightarrow \{9(1 + B) = C + D, 9(12 + B) = 5C + D, (1 + B) = 12C + D\}$.
  - Solving the above system $\Rightarrow \{C = 2, B = 6, D = 9\}$ hence the correct leading coefficient is $y^2 + 6$.

New Solution (contd.)

- In general we can scale the images based on any coefficient and not just the leading coefficient.
- So our goal is to find the coefficient of g with minimum number of terms.
- WLOG assume $n_1 \leq n_2 \leq \cdots \leq n_s = M$.

New Solution (contd.)

- In general we can scale the images based on any coefficient and not just the leading coefficient.
- So our goal is to find the coefficient of g with minimum number of terms.
- WLOG assume $n_1 \leq n_2 \leq \cdots \leq n_s = M$.
- if $n_1 = 1$ we will scale all the images based on the coefficients of images corresponding to the term with $n_1 = 1$ terms.
- Otherwise, WLOG assume that the leading coefficient has n_1 terms.
- For any $k \geq 2$, we can use the coefficients corresponding to $n_1, n_2, \ldots, n_k$ to compute the leading coefficient.

New Solution (contd.)

- In general we can scale the images based on any coefficient and not just the leading coefficient.
- So our goal is to find the coefficient of g with minimum number of terms.
- WLOG assume $n_1 \leq n_2 \leq \cdots \leq n_s = M$.
- if $n_1 = 1$ we will scale all the images based on the coefficients of images corresponding to the term with $n_1 = 1$ terms.
- Otherwise, WLOG assume that the leading coefficient has n_1 terms.
- For any $k \geq 2$, we can use the coefficients corresponding to $n_1, n_2, \ldots, n_k$ to compute the leading coefficient.
- Turns out the minimum number of images needed is $N = \max\left(M, \left\lceil \frac{(\sum_{i=1}^{s} n_i)-1}{s-1} \right\rceil\right)$ which is the same as the first solution.
- Let $S_j = \left\lceil \frac{(\sum_{i=1}^{k} n_j)-1}{j-1} \right\rceil$. We choose $k \geq 2$ such that $S_{k-1} > N$ but $S_k \leq N$.

New Solution (contd.)

- The probability that we can find the leading coefficient using only two coefficients and with minimum number of univariate images ($k = 2$) is $\frac{1}{2}$.
- This means half of the time, we can find the leading coefficient only by solving a system of size $n_1 + n_2 - 1 < N$.

New Solution (contd.)

- The probability that we can find the leading coefficient using only two coefficients and with minimum number of univariate images ($k = 2$) is $\frac{1}{2}$.
- This means half of the time, we can find the leading coefficient only by solving a system of size $n_1 + n_2 - 1 < N$.
- In general, the probability that $k > i \geq 2$ is $\frac{1}{i}$.

New Solution (contd.)

- The probability that we can find the leading coefficient using only two coefficients and with minimum number of univariate images ($k = 2$) is $\frac{1}{2}$.
- This means half of the time, we can find the leading coefficient only by solving a system of size $n_1 + n_2 - 1 < N$.
- In general, the probability that $k > i \geq 2$ is $\frac{1}{i}$.
- The special case that $N > M$ happens with probability $\frac{1}{s}$ (not frequently).
  - In this case if we want to compute minimum number of images $\Rightarrow k = s$.

New Solution (contd.)

- The probability that we can find the leading coefficient using only two coefficients and with minimum number of univariate images ($k = 2$) is $\frac{1}{2}$.
- This means half of the time, we can find the leading coefficient only by solving a system of size $n_1 + n_2 - 1 < N$.
- In general, the probability that $k > i \geq 2$ is $\frac{1}{i}$.
- The special case that $N > M$ happens with probability $\frac{1}{s}$ (not frequently).
  - In this case if we want to compute minimum number of images $\Rightarrow k = s$.
- After solving the first system (to find the leading coefficient) we can scale the images and use Zippel's method to find the other coefficients.
  - Hence total cost is $O((n_1 + \cdots + n_k)^3 + n_{k+1}^2 + \cdots + n_s^2)$.

New Solution (contd.)

- The probability that we can find the leading coefficient using only two coefficients and with minimum number of univariate images ($k = 2$) is $\frac{1}{2}$.
- This means half of the time, we can find the leading coefficient only by solving a system of size $n_1 + n_2 - 1 < N$.
- In general, the probability that $k > i \geq 2$ is $\frac{1}{i}$.
- The special case that $N > M$ happens with probability $\frac{1}{s}$ (not frequently).
  - In this case if we want to compute minimum number of images $\Rightarrow k = s$.
- After solving the first system (to find the leading coefficient) we can scale the images and use Zippel's method to find the other coefficients.
  - Hence total cost is $O((n_1 + \cdots + n_k)^3 + n_{k+1}^2 + \cdots + n_s^2)$.
- Another advantage: We can further parallelize the algorithm after computing the leading coefficient by solving other systems independently.

Problems

- A problem with this method is that there might be a common factor among the set of the coefficients we choose to compute $\mathrm{lc}_x(g)$ with.

Problems

- A problem with this method is that there might be a common factor among the set of the coefficients we choose to compute $\mathrm{lc}_x(g)$ with.
- Example:
  - Let $g = (y^2+1)x^2 - (y^3+y)x + (y^3-2y+7)$ and $p = 17$.
  - We have the form of the gcd:
    $g_f = (Ay^2+B)x^2 + (Cy^3+Dy)x + (Ey^3+Fy+G)$ and we force $A = 1$.

Problems

- A problem with this method is that there might be a common factor among the set of the coefficients we choose to compute $\mathrm{lc}_x(g)$ with.
- Example:
  - Let $g = (y^2+1)x^2 - (y^3+y)x + (y^3-2y+7)$ and $p = 17$.
  - We have the form of the gcd:
    $g_f = (Ay^2+B)x^2 + (Cy^3+Dy)x + (Ey^3+Fy+G)$ and we force $A = 1$.
  - Use the following evaluation points: $\{y_0 = 1, y_1 = 7, y_2 = 15\}$.
  - Set of images: $\{g_0 = x^2+16x+3, g_1 = x^2+10x+4, g_2 = x^2+2x+4\}$.

Problems

- A problem with this method is that there might be a common factor among the set of the coefficients we choose to compute $\mathrm{lc}_x(g)$ with.
- Example:
  - Let $g = (y^2+1)x^2 - (y^3+y)x + (y^3 - 2y + 7)$ and $p = 17$.
  - We have the form of the gcd:
    $g_f = (Ay^2 + B)x^2 + (Cy^3 + Dy)x + (Ey^3 + Fy + G)$ and we force $A = 1$.
  - Use the following evaluation points: $\{y_0 = 1, y_1 = 7, y_2 = 15\}$.
  - Set of images: $\{g_0 = x^2 + 16x + 3, g_1 = x^2 + 10x + 4, g_2 = x^2 + 2x + 4\}$.
  - System of linear equations:
    $\{16(1+B) = C + D, 10(15+B) = 3C + 7D, 2(4+B) = 9C + 15D\}$ is under-determined.
  - This happens no matter how many evaluation points we choose.
  - The reason is the common factor $\gcd(y^2+1, y^3+y) = y^2+1$.

Problems (contd.)

- Suppose coefficients of g have term counts $n_1, \ldots, n_s$ and $n_1 \leq n_2 \leq \ldots n_s$.
- Suppose we choose the set $S = \{n_1, \ldots, n_k\}$ to find the leading coefficient and there is an *unlucky* factor.
- The proposed solution is to add n_{k+1} to the set S. If the problem still exists, keep adding more coefficients to S.

Problems (contd.)

- Suppose coefficients of g have term counts $n_1, \ldots, n_s$ and $n_1 \le n_2 \le \ldots n_s$.
- Suppose we choose the set $S = \{n_1, \ldots, n_k\}$ to find the leading coefficient and there is an *unlucky* factor.
- The proposed solution is to add n_{k+1} to the set S. If the problem still exists, keep adding more coefficients to S.
- Since $\mathrm{cont}_x(g) = 1$, if at the point where $S = \{n_1, \ldots, n_s\}$ there is still a common factor, it must be an *unlucky* content.
  - This unlucky content is caused by an unlucky choice of evaluation point or prime $\Rightarrow$ Start over.

Problems (contd.)

- Suppose coefficients of g have term counts $n_1, \ldots, n_s$ and $n_1 \leq n_2 \leq \ldots n_s$.
- Suppose we choose the set $S = \{n_1, \ldots, n_k\}$ to find the leading coefficient and there is an *unlucky* factor.
- The proposed solution is to add n_{k+1} to the set S. If the problem still exists, keep adding more coefficients to S.
- Since $\mathrm{cont}_x(g) = 1$, if at the point where $S = \{n_1, \ldots, n_s\}$ there is still a common factor, it must be an *unlucky* content.
  - This unlucky content is caused by an unlucky choice of evaluation point or prime $\Rightarrow$ Start over.
- Another problem with this method is that we still can not use Zippel's method to solve the first system of equations in quadratic time.

Problems (contd.)

- The first system looks like:

$$\begin{pmatrix} 1 & \cdots & 1 & \alpha_0 & \cdots & \alpha_0 \\ k_1 & \cdots & k_m & \alpha_1 k_{m+1} & \cdots & \alpha_1 k_{m+n} \\ k_1^2 & \cdots & k_m^2 & \alpha_2 k_{m+1}^2 & \cdots & \alpha_2 k_{m+n}^2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k_1^{m+n-1} & \cdots & k_m^{m+n-1} & \alpha_{m+n-1} k_{m+1}^{m+n-1} & \cdots & \alpha_{m+n-1} k_{m+n}^{m+n-1} \end{pmatrix}$$

- $\alpha_0, \ldots, \alpha_{m+n-1}$ are the second coefficients of the univariate images of the gcd.

Problems (contd.)

- The first system looks like:

$$\begin{pmatrix} 1 & \cdots & 1 & \alpha_0 & \cdots & \alpha_0 \\ k_1 & \cdots & k_m & \alpha_1 k_{m+1} & \cdots & \alpha_1 k_{m+n} \\ k_1^2 & \cdots & k_m^2 & \alpha_2 k_{m+1}^2 & \cdots & \alpha_2 k_{m+n}^2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k_1^{m+n-1} & \cdots & k_m^{m+n-1} & \alpha_{m+n-1} k_{m+1}^{m+n-1} & \cdots & \alpha_{m+n-1} k_{m+n}^{m+n-1} \end{pmatrix}$$

- $\alpha_0, \ldots, \alpha_{m+n-1}$ are the second coefficients of the univariate images of the gcd.
- Any suggestions?