Even Faster Polynomial Multiplication

Oisín Robinson
ICHEC, Dublin

Introduction

Multiplication of polynomials can be translated to multiplication of integers via 'Kronecker-Schönhage' (KS) substitution [1]. In this approach, each polynomial is encoded as an integer by packing the coefficients together, with enough zero-padding to allow for the size of any output coefficient. This is equivalent to evaluating each polynomial at base powers. The base (e.g. 10, or 2^k) is chosen so that we may assume its arithmetic is trivial.

Harvey [2] improved the substitution method by showing the original amount of zero-padding was unnecessary - in fact, the coefficients may be packed in half the space, with two half-size multiplications, or even $1/4$ the space, with four $1/4$-size multiplications. The output may be perfectly recovered with novel reconstruction algorithms, named KS2, KS3 and KS4.

In this work, we show how we can pack coefficients even more efficiently, and re-use the KS2 and KS3 methods to reconstruct the output, with 12 multiplications of $1/8$ size. In fact the new idea generalizes and in principle can reduce the operation to $4\left(2^n-1\right)$ multiplications of $1/2^{n+1}$-th size, at increasing expense of extra additions.

KS2

$$f = 41x^3 + 49x^2 + 38x + 29,$$
$$g = 19x^3 + 23x^2 + 46x + 21.$$

Then

$$f(10^2) = 41493829, \quad g(10^2) = 19234621,$$
$$f(-10^2) = -40513771, \quad g(-10^2) = -18774579.$$

Packed with *alternating signs* — still linear time. Two half-sized integer multiplications:

$$h(10^2) = f(10^2)g(10^2) = 798118074653809,$$
$$h(-10^2) = f(-10^2)g(-10^2) = 760628994227409.$$

If $h(x) = h^0(x^2) + xh^1(x^2)$, then

$$h^0(10^4) = \tfrac{1}{2}\left(h(10^2) + h(-10^2)\right) = 779373534440609$$
$$10^2 h^1(10^4) = \tfrac{1}{2}\left(h(10^2) - h(-10^2)\right) = 187445402132 00$$

KS3, KS4

KS3 is a variation of KS2 which packs the coefficients in *reversed* order. It also leads to two half-size multiplications. In fact, KS2 and KS3 are 'orthogonal', in that they may be cascaded. This is KS4, and reconstructs the output from four $1/4$-size multiplications.

KS5

KS5 is a variation of KS4 which also cascades KS2 and KS3, but which uses 12 multiplications of $1/8$-th size. When using KS4 (the faster of the three) we recall that the packing operation is equivalent to evaluating the polynomial at e.g. 10 and -10 (KS2), or $1/10$ and $-1/10$ (KS3). The new insight here is to evaluate f and g at $\sqrt{10}$, $\sqrt{-10}$, $\sqrt{1/10}$ and $\sqrt{-1/10}$ and use the same cascading idea as KS4. The difference is that when we evaluate at these points, we get an integer in a quadratic number field and must then multiply four such integers. This can be done with 12 componentwise multiplications.

$$f = 41x^3 + 49x^2 + 38x + 29.$$

Then $10^2 f(\sqrt{-\frac{1}{10}}) = 241 + 339\sqrt{-\frac{1}{10}}$.

$$\begin{array}{r} 290 \\ -49 \quad + \\ \hline 241 \end{array} \qquad \begin{array}{r} 380 \\ -41 \quad + \\ \hline 339 \end{array}$$

Note that $h(\sqrt{10}) = h^0(10) + h^1(10)\sqrt{10}$. We have 12 multiplications of $1/8$th the size, which precede the LHS below.

$$\left.\begin{matrix} h^0(10) \\ h^0(-10) \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{00}(10^2) & h^{00}(10^2) \\ h^{01}(10^2) & h^{00}(10^{-2}) \end{matrix}\right\} \xrightarrow{KS3} h^{00}(10^4)$$
$$\left.\begin{matrix} h^1(10) \\ h^1(-10) \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{10}(10^2) & h^{01}(10^2) \\ h^{11}(10^2) & h^{01}(10^{-2}) \end{matrix}\right\} \xrightarrow{KS3} h^{01}(10^4)$$
$$\left.\begin{matrix} h^0(\frac{1}{10}) \\ h^0(-\frac{1}{10}) \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{00}(10^{-2}) & h^{10}(10^2) \\ h^{01}(10^{-2}) & h^{10}(10^{-2}) \end{matrix}\right\} \xrightarrow{KS3} h^{10}(10^4)$$
$$\left.\begin{matrix} h^1(\frac{1}{10}) \\ h^1(-\frac{1}{10}) \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{10}(10^{-2}) & h^{11}(10^2) \\ h^{11}(10^{-2}) & h^{11}(10^{-2}) \end{matrix}\right\} \xrightarrow{KS3} h^{11}(10^4)$$

Complexity

We model the speedup over naive 'schoolbook' multiplication as

$$\frac{1}{m} \cdot k^2$$

where m is the number of multiplications, and ordinary KS uses integers k times as large. With this notation, the speedup of KS4 over KS is $4\times$, and of KS5 over KS is $5\,1/3\times$.

Example

We would like to see if it really works, so here is an example.

$$f = 41x^3 + 49x^2 + 38x + 29$$
$$g = 19x^3 + 23x^2 + 46x + 21.$$

$$\begin{aligned} h(\sqrt{10}) &= f(\sqrt{10})g(\sqrt{10}) = (519 + 448\sqrt{10})(251 + 236\sqrt{10}) \\ &= 1187549 + 234932\sqrt{10} = h^0(10) + h^1(10)\sqrt{10} \\ h(\sqrt{-10}) &= f(\sqrt{-10})g(\sqrt{-10}) = (-461 - 372\sqrt{-10})(-209 - 144\sqrt{-10}) \\ &= -439331 + 144132\sqrt{-10} = h^0(-10) + h^1(-10)\sqrt{-10} \\ 10^3 h(\sqrt{1/10}) &= 10 \cdot 10 f(\sqrt{1/10}) 10 g(\sqrt{1/10}) \\ &= 10 \cdot (339 + 421\sqrt{1/10})(233 + 479\sqrt{1/10}) \\ &= 991529 + 2604740\sqrt{1/10} = 10^3\left(h^0(1/10) + h^1(1/10)\sqrt{1/10}\right) \\ 10^3 h(\sqrt{-1/10}) &= 10 \cdot 10 f(\sqrt{-1/10}) 10 g(\sqrt{-1/10}) \\ &= 10 \cdot (241 + 339\sqrt{-1/10})(187 + 441\sqrt{-1/10}) \\ &= 301171 + 1696740\sqrt{-1/10} = 10^3\left(h^0(-1/10) + h^1(-1/10)\sqrt{-1/10}\right) \end{aligned}$$

$$\left.\begin{matrix} h^0(10) = 1187549 \\ h^0(-10) = -439331 \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{00}(10^2) = 374109 & 374109 \\ h^{01}(10^2) = 813440 & 64635 \end{matrix}\right\} \xrightarrow{KS3} h^{00}(10^4)$$
$$\left.\begin{matrix} h^1(10) = 234932 \\ h^1(-10) = 144132 \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{10}(10^2) = 189532 & 813440 \\ h^{11}(10^2) = 45400 & 345179 \end{matrix}\right\} \xrightarrow{KS3} h^{01}(10^4)$$
$$\left.\begin{matrix} h^0(\frac{1}{10}) = 991529 \\ h^0(-\frac{1}{10}) = 301171 \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{00}(10^{-2}) = 64635 & 189532 \\ h^{01}(10^{-2}) = 345179 & 215074 \end{matrix}\right\} \xrightarrow{KS3} h^{10}(10^4)$$
$$\left.\begin{matrix} h^1(\frac{1}{10}) = 2604740 \\ h^1(-\frac{1}{10}) = 1696740 \end{matrix}\right\} \xrightarrow{KS2} \left\{\begin{matrix} h^{10}(10^{-2}) = 215074 & 45400 \\ h^{11}(10^{-2}) = 45400 & 45400 \end{matrix}\right\} \xrightarrow{KS3} h^{11}(10^4)$$

This gives

$$h = 779x^6 + 1874x^5 + 3735x^4 + 4540x^3 + 3444x^2 + 2132x + 609.$$

Notice the upper and lower halves of coefficients appearing on the right (up to adjusting by carries which it is possible to do)

Implementation

Harvey's original implementation, zn_poly, was adapted to add the new cascade method. It works in practice and performance data is currently being gathered.

Applications

One exciting application in particular is reducing the memory requirement for stage 2 of the elliptic curve method of integer factorization. This carries out many large polynomial products, and requires up to gigabytes of memory. At present, it is possible to reduce required memory using the 'stage 2 blocks' method, however the number of blocks required to save more memory grows rapidly. With KS5, we have a straightforward way to split the memory required by eight, with 12 smaller computations for each product. This might be improved even further if the generalized method was implemented.

References

[1] J. von zur Gathen and J. Gerhard, *Modern computer algebra*.
Cambridge University Press, Cambridge, third ed., 2013.

[2] D. Harvey, "Faster polynomial multiplication via multipoint Kronecker substitution," *J. Symbolic Comput.*, vol. 44, no. 10, pp. 1502–1510, 2009.

[3] A. J. Devegili, C. ÓhÉigeartaigh, M. Scott, and R. Dahab, "Multiplication and squaring on pairing-friendly fields," *IACR Cryptology ePrint Archive*, vol. 2006, p. 471, 2006.

Contact Information

- Web: http://www.ichec.ie
- Email: oisin.robinson@ichec.ie

ICHEC
Irish Centre for High-End Computing

UCD DUBLIN