Undergraduate Research Assistantship Spring 2004

Researcher: Peter Manning Jr.
Advisors: Maurice Hasson and Juan Restrepo

Research Topic: Development of a matrix preconditioner from a polynomial approximation of 1/x

Introduction

In many areas of scientific study, we have the need to solve the matrix equation $Ax = b$, where A is an nxn matrix. This equation arises in such fields as quantum mechanics, biology and many other physical systems. Due to the complexity of nature, the matrices that appear tend to be very large. The purpose of this project is to develop a method to solve this equation quicker than traditional methods. Traditionally, $Ax = b$ is solved either by Gaussian elimination or through finding A^{-1}, as $A^{-1}Ax = A^{-1}b$, and thus $Ix = A^{-1}b$, I being the identity matrix. Iterative methods that are commonly utilized in solving $Ax = b$ include the Jacobi method, the Gauss-Seidel method as well as Lanczos methods. The Lanczos method generally requires $O(n \log n)$ operations, the Jacobi method requires $n^3/3$ operations, and Gaussian elimination requires $9n^3$ operations. The two methods that require $O(n^3)$ are too slow for very large matrices. What this project is aiming to accomplish is efficiently find a matrix B such that $B \approx A^{-1}$. Once B is found, the matrix equation $BAx = Bb$ can be solved quickly using any of the iterative methods described above. This new method can be implemented in $O(n^2)$ operations. Although this is less expensive than the Lanczos method, it is a new, less complicated approach to the problem.

Background

The matrix B is called a preconditioner. The purpose of a preconditioner is to turn the matrix equation $Ax = b$ into an easier problem. Once a preconditioner matrix B is found, the new equation $BAx = Bb$ will have the same solution as $Ax = b$. In general, a good preconditioner must exhibit two properties: the preconditioned system must converge quickly and the construction of B must be efficient. Thus, once B is created, the linear system $BAx = Bb$ should converge quickly under iterative solving methods. The reason for this is the following. A preconditioning matrix B should be a very close approximation to A^{-1}. Thus, the left side of the equation is $BA \approx A^{-1}A \approx I$. Now, with a matrix $BA \approx I$, the spectrum of BA is very close to one in the complex plane. Thus, since the spectrum of BA is so close to one, $BAx = Bb$ can be solved iteratively very quickly.

Polynomial Approximation to 1/x

The theory behind this project is as follows. We will construct a polynomial P such that $P_{2n+1}(x) \approx 1/x$. This approximation is valid in the interval $[-a,-b] \cup [a,b]$.

Through the research of Maurice Hasson and Juan Restrepo, it has been determined that the function $1/x$ can be approximated on the interval $[-a,-b] \cup [a,b]$ by the following equation:

$$P_{2n+1}(x) = x\sum_{k=0}^{n}{}' \frac{2}{\pi}\int_0^{\pi} \frac{\cos(k\theta)}{\frac{1}{2}((b^2-a^2)\cos(\theta)+b^2+a^2)}\,d\theta \left(T_k\left(\frac{2x^2}{b^2-a^2} - \frac{b^2+a^2}{b^2-a^2}\right)\right)$$

$$x \in [-a,-b] \cup [a,b]$$

$T_k(u)$ are the Chebyshev polynomials and the ‘ indicates that the first term of the sum must be halved. Because the term inside of the sum is always an even function due to the fact that $T_k(u)$ yields a function of degree *2n*, multiplying the sum by x forces the polynomial to be odd. When approximating $1/x$ this is necessary, as $1/x$ is an odd function.

Now that we have a polynomial $P_{2n+1}(x) \approx 1/x$, we can apply it to the matrix A from $Ax = b$. The matrix $P_{2n+1}(A)$ is approximately A^{-1}. $|P_{2n+1} - 1/x| \leq \varepsilon$ with $x \in [-a,-b] \cup [a,b]$ and it can be proved that $\|P_{2n+1}(A) - A^{-1}\| \leq \varepsilon$ with the spectra of A in $[-a,-b] \cup [a,b]$. It can also be proved that $\|P_{2n+1}(x)A - I\| \leq \varepsilon$. The degree of approximation, ε , of order *2n+1* of $f(x) = \frac{1}{x}$ on $[-a,-b] \cup [a,b]$ is given by

$E_{2n+1}(f) \approx b\left(\frac{b-a}{b+a}\right)^n$. In theory it is possible to make $E_{2n+1}(f)$ as small as one wants, in practice it is more difficult.

Numerical Implementation

Maurice's research indicated that creating a polynomial of degree 19 would be sufficient to conduct experiments and would most likely be closest to the near best approximation of A^{-1}. It turns out that the integral

$$\frac{1}{\pi}\int_0^{\pi} \frac{\cos(k\theta)}{\frac{1}{2}((b^2-a^2)\cos(\theta)+b^2+a^2)}\,d\theta$$

is approximately $(-1)^k 3^{-k}$, $k = 0,1,2...n$. Thus, in constructing the polynomial of degree *2n+1*, the coefficients used were $(-1)^k 3^{-k}$, for $k = 0,1,2...9$ with the first coefficient halved. Below illustrates the polynomial that was constructed to test the method described. The polynomial *P* in the below case is represented by ψ. λ represents the polynomial of degree 2 inside of the Chebyshev polynomial.

```
i2 = 2
i1 = 1
λ = 2 x^2 / (i2^2 - i1^2) - (i2^2 + i1^2) / (i2^2 - i1^2)
a0 = 2 * (-1) ^0 * 0.33333333333333333333 ^0
a1 = 2 * (-1) ^1 * 0.33333333333333333333 ^1
a2 = 2 * (-1) ^2 * 0.33333333333333333333 ^2
a3 = 2 * (-1) ^3 * 0.33333333333333333333 ^3
a4 = 2 * (-1) ^4 * 0.33333333333333333333 ^4
a5 = 2 * (-1) ^5 * 0.33333333333333333333 ^5
a6 = 2 * (-1) ^6 * 0.33333333333333333333 ^6
a7 = 2 * (-1) ^7 * 0.33333333333333333333 ^7
a8 = 2 * (-1) ^8 * 0.33333333333333333333 ^8
a9 = 2 * (-1) ^9 * 0.33333333333333333333 ^9
ψ =
 x * (.5 (a0) + λ * (a1) + (a2) * (2 * λ^2 - 1) + (a3) * (4 λ^3 - 3 λ) + (a4) * (8 λ^4 - 8 λ^2 + 1) +
    (a5) * (16 λ^5 - 20 λ^3 + 5 λ) + (a6) * (32 λ^6 - 48 λ^4 + 18 λ^2 - 1) +
    (a7) * (64 λ^7 - 112 λ^5 + 56 λ^3 - 7 λ) +
    (a8) * (1 - 32 λ^2 + 160 λ^4 - 256 λ^6 + 128 λ^8) +
    (a9) * (9 λ - 120 λ^3 + 432 λ^5 - 576 λ^7 + 256 λ^9))

ψ = ψ / 2
```

When expanded, the polynomial of degree 19 is

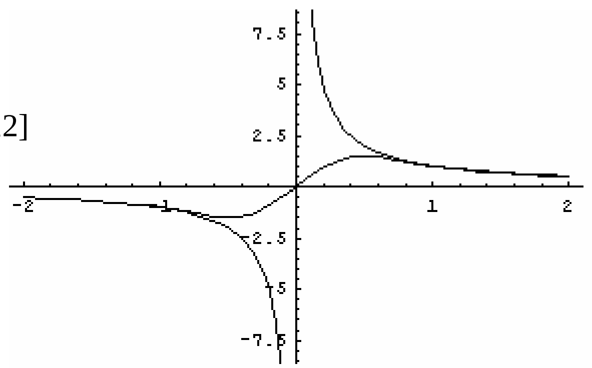
$$5.0625x - 11.21484375081669984x^3 + 14.31884765392467408x^5 - 11.67448425785245447x^7 + 6.35605429737609980x^9 - 2.342474354783027492x^{11} + 0.577686246222253878x^{13} - 0.0913463304208466889x^{15} + 0.00837341362191094648x^{17} - 0.000338319742299432181x^{19}$$

This is an approximation that was created by Mathematica and the number of digits is not significant. As one can already observe, the polynomial is odd as we expected.

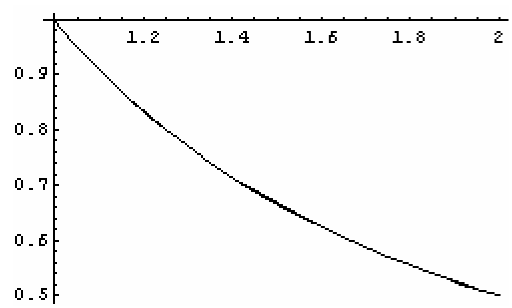
Numerical Examples

Below is a plot of the above polynomial and 1/x on the interval [-2,2] and [1,2] as marked. It is apparent that the polynomial is extremely close to 1/x.

[-2,2]

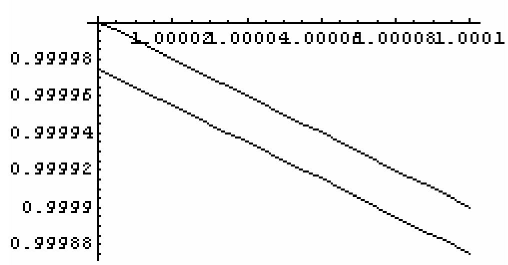


[1,2]



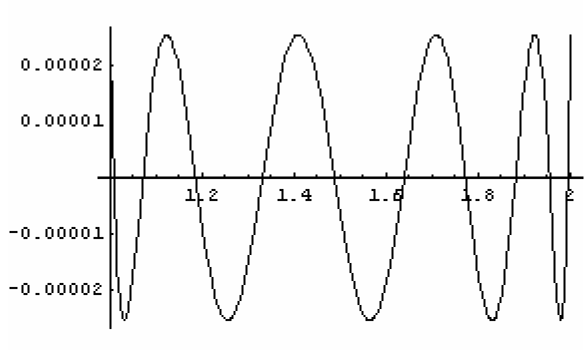
If we zoom in and plot 1/x and the polynomial in the interval [1,1.0001], the error between the two graphs is now visible.

[1,1.0001]



The error is approximately .00003, as can also be observed below in a plot of $1/x - \psi$.

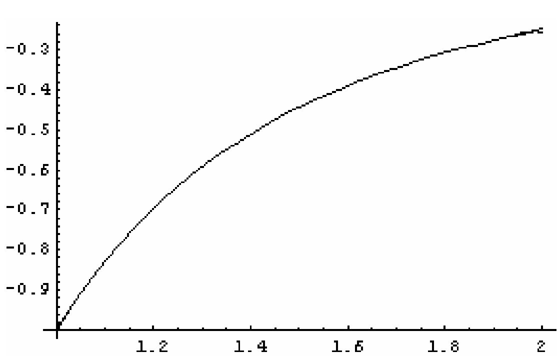
[1,2]



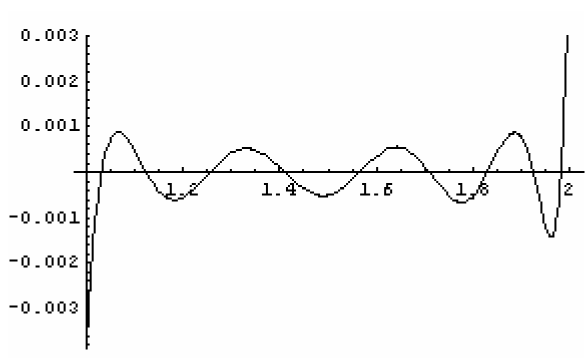
If we apply the error formula to $n = 9$, the maximum error is .0001. Therefore, the polynomial that we created has less error than was expected.

Although the approximation is extremely accurate, its derivatives are not. In certain matrices, this poses a problem. Though we will not go into what kind of matrices the derivatives of the polynomial cause a problem for, it is important to illustrate the error that arises. Below are the first, second, and third derivatives of the polynomial plotted with the first, second and third derivatives of 1/x on the interval [1,2] on the left with the error between the two graphs on the right.

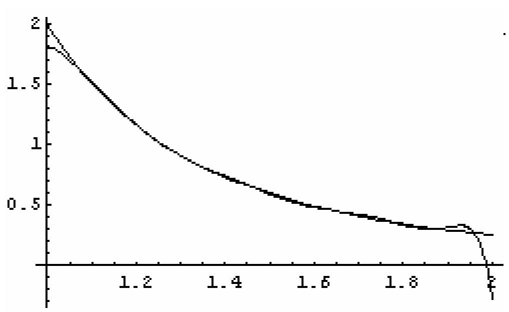
First derivative



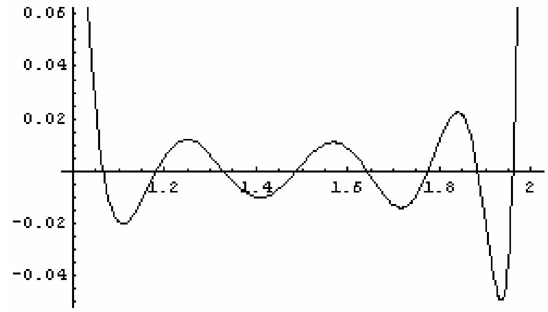
Error in first derivative



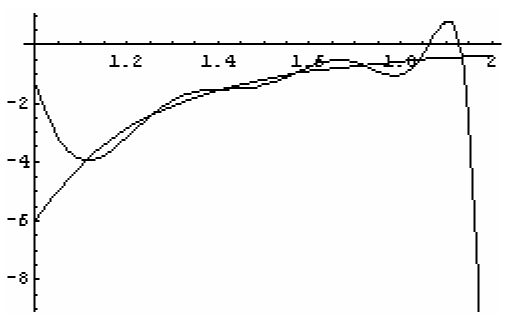
Second Derivative



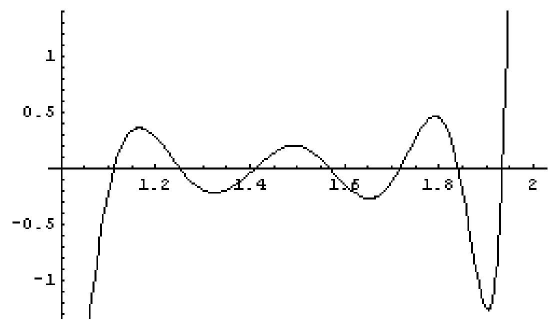
Error in second derivative



Third Derivative



Error in third derivative



As we take higher and higher derivatives, it is apparent that the accuracy of the polynomial approximation deteriorates. What is most obvious is how the polynomial's derivatives become extremely unstable at the boundaries of the interval [1,2]. This has serious repercussions for certain matrices of the form

$$\begin{array}{cccc} d & c & . & a \\ . & d & c & . \\ . & . & d & c \\ . & . & . & d \end{array}$$

The reason for this is unknown at this time.

Currently, these are all the numerical experiments that have been completed by the undergraduate researcher, although numerical experiments using symmetric matrices have been completed by Maurice Hasson. These experiments also indicate that the method described is accurate to four decimal places and closely approximate the matrix A^{-1}.

Conclusion

By utilizing polynomial approximation methods to create a near best approximation to 1/x, we are able to obtain an approximation for the matrix A^{-1} from the linear system $Ax = b$. Although this approximation is not close enough to A^{-1} to solve the linear system, it is a perfect preconditioning matrix. Now, instead of solving $Ax = b$, we are able to solve $BAx = Bb$, where B is the matrix obtained from the polynomial approximation $P_{2n+1}(A)$. Using this method, a polynomial of degree 19 was created and compared to 1/x. The maximum error of the approximation was .00003 and the minimum error was 0. Experiments done with symmetric matrices indicate that the L2 norm of the matrix $\left\|P_{2n+1}(A) - A^{-1}\right\| \le .00004$. Although only a few experiments have been completed, it is obvious that the desired results are being obtained. Next, it will be imperative that this method for solving linear systems is demonstrated for much larger matrices than the ones currently experimented with. Also, the problem with the derivatives of the polynomial approximation will also have to be resolved.

References

Bramley, Randall. “Iterative Methods for Linear Systems.” Class notes: Scientific Computing I. Spring 2004. <http://www.cs.indiana.edu/l/www/classes/p573/notes/pde/iter.html>.

Hasson, Maurice and Juan Restrepo “Preconditioning of Symmetric Matrices with Clustered Eigenvalues.” May, 2004.

Hendrickson, Bruce. “Support Theory for Preconditioning.” January, 2001. <http://www.cs.sandia.gov/~bahendr/support.html>.

Mulligan, S. "Iterative Methods for Linear Systems." Applied Computing for Technologists <http://www.comp.dit.ie/smulligan/Itertive.pdf>.