AN ABSTRACT OF THE THESIS OF

Douglas J. Limmer for the degree of Master of Science in Mathematics presented on June 8, 1993.

Title: Using p-adic Valuations to Decrease Computational Error

Redacted for Privacy

Abstract approved: _____

Robert O. Robson

The standard way of representing numbers on computers gives rise to errors which increase as computations progress. Using p-adic valuations can reduce error accumulation. Valuation theory tells us that p-adic and standard valuations cannot be directly compared. The p-adic valuation can, however, be used in an indirect way. This gives a method of doing arithmetic on a subset of the rational numbers without any error. This exactness is highly desirable, and can be used to solve certain kinds of problems which the standard valuation cannot conveniently handle. Programming a computer to use these p-adic numbers is not difficult, and in fact uses computer resources similar to the standard floating-point representation for real numbers. This thesis develops the theory of p-adic valuations, discusses their implementation, and gives some examples where p-adic numbers achieve better results than normal computer computation.

Using p-adic Valuations to Decrease Computational Error

by

Douglas J. Limmer

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed June 7, 1993

Commencement June 1994

APPROVED:

Redacted for Privacy

Professor of Mathematics in charge of major

Redacted for Privacy

Head of department of Mathematics

Redacted for Privacy

Dean of Graduate School

Date these is presented: June 8, 1993

TABLE OF CONTENTS

Using p-adic Valuations to Decrease Computational Error

INTRODUCTION

Using usual computer arithmetic and approximations, errors in numbers will grow, and often surprisingly quickly. A method to store and use numbers which has no error would be a help to anyone doing numerical computation. Failing that, it would be advantageous to have a system which adds less error than the usual computer arithmetic operations do.

There are methods of storing numbers exactly. However, these methods are often cumbersome. They require relatively large amounts of storage space, and they increase computational complexity. These sorts of solutions are impractical for computations which use a large amount of storage space, or whose algorithms innately require a lot of time to run.

Therefore, what is needed is a system of doing arithmetic which doesn't take an impractical amount of time, uses storage space similar to the normal representation for computer numbers, and decreases the amount of error in the computation. A study of valuation theory will show the methods of measuring error, and which valuations can decrease error. It will lead to the p-adic valuations, which will result in a method that can reduce error in computation.

This p-adic representation for numbers is surprisingly easy to program. Many of the operations turn out to be nearly the same as their floating-point counterparts. I have written a computer package which represents 2-adic numbers, and includes the arithmetic on them. These numbers can be used as a standard computer data type, and will give greatly reduced error, or no error, for some computations.

VALUATION THEORY

To determine how big the error is in a calculation, it is important to determine how big numbers are in general. The usual method of doing this is the absolute value. Valuation Theory studies a generalization of the absolute value, and its general properties will help in the study of errors.

DEFINITION: A *valuation* on a field $\mathbf{F}$ is a function ϕ from $\mathbf{F}$ into the real numbers $\mathbf{R}$ such that

$$(i) \qquad \phi(a) \geq 0 \ \forall a \in \mathbf{F} \text{ and } \phi(a) = 0 \iff a = 0$$

$$(ii) \qquad \phi(ab) = \phi(a) \cdot \phi(b)$$

$$(iii) \qquad \phi(a+b) \leq \phi(a) + \phi(b).$$

Note that the function $\phi(a) = |a|$ is a valuation for the field of rational numbers $\mathbf{Q}$, and for $\mathbf{R}$. In addition, the function ϕ defined by $\phi(0) = 0$ and $\phi(a) = 1$ if $a \neq 0$ is also a valuation, called the *trivial valuation*.

In computers, real numbers are generally stored as an approximated rational number, usually a rational number with a power of 2 as a denominator. Additionally, they are stored in a floating point format, which means that for large numbers, only the digits toward the higher places are kept as significant. This approximation causes error, and arithmetic on these numbers will increase the error.

For instance, suppose the real numbers r and s are approximated by the rational numbers $\tilde{r}$ and $\tilde{s}$. Suppose also that each approximation is within ϵ of the original number, i.e., $\phi(r-\tilde{r}) < \epsilon$ and $\phi(s-\tilde{s}) < \epsilon$, where ϵ is a small positive number. Then the approximate sum will be $\tilde{r}+\tilde{s}$, and the error in the result will be

$$\phi((r+s)-(\tilde{r}+\tilde{s})) = \phi((r-\tilde{r})+(s-\tilde{s})) \leq \phi(r-\tilde{r}) + \phi(s-\tilde{s}) \leq 2\epsilon.$$

Thus, the possible error has doubled. Similarly, in multiplication, the error is

$$\phi(rs - \tilde{r}\tilde{s}) = \phi(s(r-\tilde{r}) + \tilde{r}(s-\tilde{s})) \leq \phi(s)\phi(r-\tilde{r}) + \phi(\tilde{r})\phi(s-\tilde{s}) < \epsilon(\phi(s) + \phi(\tilde{r})).$$

This can potentially lead to a large amount of error.

We would like to determine all possible valuations on the rational numbers, to see if there are any valuations which give less error. In particular, a lower bound on the sum of two numbers would decrease potential error accumulation. To determine all rational valuations, some properties of valuations and additional definitions will be helpful. In general for any valuation,

$$\phi(1) = \phi(1 \cdot 1) = \phi(1)\phi(1)$$

$$\phi(1) = \phi(-1 \cdot -1) = \phi(-1)\phi(-1)$$

so that $\phi(1) = \phi(-1) = 1$. From this comes that $\phi(-a) = \phi(a)$ and $\phi(a-b) \leq \phi(a) + \phi(b)$. Call the absolute value valuation on the rational numbers ϕ_∞, so that $\phi_\infty(a) = |a|$. Similarly, call the trivial valuation on the rationals ϕ_0.

DEFINITION: Two valuations ϕ and ψ are *equivalent* if

$$\phi = \psi^\rho$$

for some real number $\rho > 0$.

The trivial valuation is only equivalent to itself, since $0^\rho = 0$ and $1^\rho = 1$ for any positive ρ.

Lemma 1. *Equivalence of valuations is an equivalence relation.*

Proof: $\phi = \phi^1$, so ϕ is equivalent to itself. If $\phi = \psi^\rho$, then $\psi = \phi^{\frac{1}{\rho}}$, and $\frac{1}{\rho} > 0$, so equivalence is symmetric. Also, if $\psi = \phi^{\rho_1}$ and $\theta = \psi^{\rho_2}$, then $\theta = \phi^{\rho_1\rho_2}$. The property of equivalence is transitive. Q. E. D.

DEFINITION: A valuation is called *non-archimedean* if

$$\phi(a+b) \leq \max\{\phi(a), \phi(b)\}$$

for all a,$b \in \mathbf{F}$. The valuation is called *archimedean* otherwise.

The trivial valuation is non-archimedean. The p-adic valuations ϕ_p on $\mathbf{Q}$ are also non-archimedean, as defined below.

DEFINITION: The *p-adic valuation* for a prime p on $\mathbf{Q}$ is defined as $\phi_p(p) = p^{-1}$ and $\phi_p(q) = 1$ for q prime and not equal to p. The valuation on the rest of the rationals is defined by unique factorization of integers, and the multiplicative property a valuation needs.

Lemma 2. *If ϕ is an archimedean valuation and ψ is a non-archimedean valuation, then ϕ and ψ are not equivalent.*

Proof: Since ϕ is archimedean, there are two elements m and n such that $\phi(m+n) > \max\{\phi(m), \phi(n)\}$. One of $\phi(m)$ and $\phi(n)$ must be larger. Without loss of generality, we may assume that $\phi(n) > \phi(m)$. Thus,

$$\phi(n)\phi(\frac{m}{n}+1) > \phi(n)\max\{\frac{\phi(m)}{\phi(n)}, 1\} = \phi(n),$$

so that $\phi(\frac{m}{n}+1) > 1$. However, ψ is non-archimedean, so

$$\psi(m+n) \leq \max\{\psi(m), \psi(n)\},$$

or

$$\psi(\frac{m}{n}+1) \leq \max\{\psi(\frac{m}{n}), 1\} = 1,$$

assuming ϕ and ψ are equivalent (so that $\psi(m) < \psi(n)$). If the two are equivalent, then $\phi(m+n) = (\psi(m+n))^{\rho}$ for some positive real ρ. Since one value is greater than one, and the other is not, there is no positive real number ρ which accomplishes this. Thus, the two valuations are not equivalent. Q. E. D.

Note that since $\phi_\infty(2) = \phi_\infty(1+1) = 2 > 1 = \max\{\phi_\infty(1), \phi_\infty(1)\}$, ϕ_∞ is archimedean, so that ϕ_∞ is not equivalent to the p-adic valuation ϕ_p for any prime p.

Lemma 3. *ϕ_p is equivalent to ϕ_q if and only if $p = q$.*

Proof: If $p = q$, it is trivially true that the two valuations are equivalent. If ϕ_p is equivalent to ϕ_q, then there is some positive ρ such that $\phi_p^\rho = \phi_q$. Thus, in particular, $\phi_p(p)^\rho = \phi_q(p)$. However, if $q \neq p$, then $\phi_q(p) = 1 \neq p^{-\rho} = \phi_p(p)^\rho$, contradicting that the valuations are equivalent. Hence, q must equal p. Q. E. D.

Lemma 4. *If a, b and c are real numbers such that $c^n \leq a + nb$ for all n, then $c \leq 1$.*

Proof: If $c > 1$, then $n(c-1) > a$ and $\frac{n-1}{2}(c-1)^2 > b$ for some sufficiently large n, so that

$$c^n \geq (c-1)^n \geq 1 + n(c-1) + \binom{n}{2}(c-1)^2 > a + nb.$$

Thus, by way of contradiction, $c \leq 1$. Q. E. D.

Lemma 5. *If $\phi(m \cdot 1) \leq 1$ for all positive integers m, then ϕ is non-archimedean.*

Proof: Let x be any element of $\mathbf{F}$ such that $\phi(x) \leq 1$. Then

$$(\phi(x+1))^n = \phi(\sum_{i=0}^{n} \binom{n}{i} \cdot x^i) \leq \sum_{i=0}^{n} \phi(\binom{n}{i} \cdot 1) \cdot (\phi(x))^i \leq (n+1) \cdot \sup_{m \in \mathbf{Z}_+} \{\phi(m \cdot 1)\}$$

since $\binom{n}{i}$ is a positive integer, and $\phi(x) \leq 1$. Also, since the sup term above equals 1 by the assumption, then by Lemma 4, $\phi(x+1) \leq 1$. Now, choose x and y in $\mathbf{F}$. Without loss of generality, we may assume that $\phi(x) \leq \phi(y)$. So, $\phi(x+y) = \phi(y)\phi(\frac{x}{y} + 1) \leq \phi(y)$, since $\phi(\frac{x}{y}) \leq 1$, so that $\phi(x+y) \leq \max\{\phi(x), \phi(y)\}$, or ϕ is non-archimedean. Q. E. D.

Note that this also says that if ϕ is archimedean, then $\phi(m \cdot 1) > 1$ for some positive integer m.

Theorem 6. *Any non-trivial valuation of* $\mathbf{Q}$ *is equivalent to exactly one of* ϕ_∞ *or* ϕ_p *for some prime* p.

Proof: (This proof, as well as the proofs for Lemmas 2 and 5, were modified from proofs in Endler [2].)

Case 1: Suppose a valuation ϕ is non-trivial and non-archimedean on $\mathbf{Q}$. Then, by Lemma 2, ϕ is not equivalent to ϕ_∞. By Lemma 3, if ϕ is equivalent to any ϕ_p, then it is equivalent to only one such valuation.

Let $I = \{a \in \mathbf{Z} | \phi(a) < 1\}$, where $\mathbf{Z}$ indicates the integers. Let $i,j \in I$ and $r \in \mathbf{Z}$. Then

$$\phi(i-j) \leq \max\{\phi(i), \phi(j)\} < 1,$$

so that I is a subgroup of $\mathbf{Z}$ under addition. ri is the sum of r copies of i, so that $\phi(ri) = \phi(i+i+\cdots+i) \leq \max\{\phi(i), \phi(i), \ldots, \phi(i)\} < 1$. Therefore, I is an ideal of $\mathbf{Z}$.

$\phi(n) = \phi(1+1+\cdots+1) \leq \max\{1,1,\ldots,1\} = 1$ for any $n \in \mathbf{Z}_+$. Suppose $I = \{0\}$. Then $\phi(n) = 1$ for all n (by definition of I), so $\phi(n^{-1}) = 1$ for all n, so that

$$\phi(\frac{a}{b}) = \frac{\phi(a)}{\phi(b)} = 1$$

for all non-zero fractions $\frac{a}{b}$. Thus, ϕ is the trivial valuation on $\mathbf{Q}$. By way of contradiction, $I \neq \{0\}$.

Suppose that $a, b \in \mathbf{Z}$ such that $ab \in I$. Then $\phi(ab) < 1$, or $\phi(a)\phi(b) < 1$. If both $\phi(a)$ and $\phi(b)$ are greater than or equal to 1, then their product must also be, hence one of $\phi(a)$ and $\phi(b)$ must be less than 1. Thus, either a or b is in I, so that I is a prime ideal, or $I = p\mathbf{Z}$ for some prime p.

$p \in I = p\mathbf{Z}$, so $0 < \phi(p) < 1$. Thus, there is some positive real number ρ such that $\phi(p) = p^{-\rho} = (\varphi_p(p))^\rho$. Also note that if $n \in \mathbf{Z}\backslash I$, then $\phi(n) \geq 1$. But it has been previously shown that $\phi(n) \leq 1$ for any integer, so that $\phi(\mathbf{Z}\backslash I) = \{1\}$.

By definition of ϕ_p, $\phi_p^\rho(n) = 1$ for any n not divisible by p. Now, any integer can be written as mp^k, where p does not divide m. $\phi(mp^k) = \phi(m)(\phi(p))^k = (\phi_p(m))^\rho(\phi_p(p))^{\rho k}$, so ϕ and ϕ_p^ρ are identical on the integers. Since $\phi(\frac{a}{b}) = \frac{\phi(a)}{\phi(b)}$, the two must also match on all the rationals. Hence, ϕ is equivalent to ϕ_p.

Case 2: Suppose a valuation ϕ is non-trivial and archimedean on $\mathbf{Q}$. Then, by Lemma 2, ϕ is not equivalent to any ϕ_p, since they are non-archimedean.

For all integers $m > 1$, $n > 1$ and $t \geq 1$ there is some integer $s \geq 0$ and $a_0, \ldots, a_s \in \{0, \ldots, n-1\}$ such that $a_s \neq 0$ and $m^t = a_0 + a_1 \cdot n + \cdots + a_s \cdot n^s$. Thus, $m^t \geq n^s$, so $s \leq t \log(m)/\log(n)$, and so

$$(\phi(m))^t \leq \sum_{i=0}^{s} \phi(a_i)\phi(n)^i \leq n \cdot \sum_{i=0}^{s} \phi(n)^i$$

$$\leq n \cdot (s+1) \cdot \max\{1, \phi(n)^s\} \leq n(\frac{\log m}{\log n} \cdot t + 1) \cdot (\max\{1, (\phi(n))^{\frac{\log m}{\log n}}\})^t.$$

Note that this uses $\phi(a_i) \leq n$ (true by $a_i < n$ and the triangle inequality), and that each $\phi(n)^i$ is less than either 1 or $\phi(n)^s$, depending on whether $\phi(n) < 1$ or not. By Lemma 4, this implies that

$$\frac{\phi(m)}{\max\{1, (\phi(n))^{\frac{\log m}{\log n}}\}} \leq 1,$$

or that $\phi(m) \leq \max\{1, (\phi(n))^{\frac{\log m}{\log n}}\}$.

Assume $\phi(n) \leq 1$ for some integer n. Then by the above statement, $\phi(m) \leq 1$ for all positive integers m, and by Lemma 5, that means ϕ is non-archimedean. By contradiction, $\phi(n)$ must be bigger than 1 for all $n > 1$. Therefore, by the above, $\phi(m) \leq (\phi(n))^{\frac{\log m}{\log n}}$, or

$$(\phi(m))^{\frac{1}{\log m}} \leq (\phi(n))^{\frac{1}{\log n}}.$$

Since this holds true for every integer greater than 1, in particular it holds for both $m < n$ and $m > n$, so that $(\phi(m))^{\frac{1}{\log m}}$ is constant for all $m > 1$, or,

since $\phi(m)$ is positive, there is some ρ such that $(\phi(m))^{\frac{1}{\log m}} = e^{\rho}$ for all $m > 1$. Because ϕ is archimedean, $\phi(m) > 1$ for some m, so ρ is positive.

Therefore

$$\phi(m) = m^{\rho} = \phi_{\infty}(m)^{\rho}$$

for all integers greater than 1. This is also true for $m \in 0, 1$, since $\phi(0) = 0, \phi(1) = 1$. So $\phi(m) = (\phi_{\infty}(m))^{\rho}$ for all non-negative integers and hence, by multiplication, is true for negative integers, and all rational numbers. Thus, ϕ and ϕ_{∞} are equivalent valuations on $\mathbf{Q}$. Q. E. D.

Therefore, all valuations on $\mathbf{Q}$ are equivalent to either the trivial valuation, the standard absolute value, or one of the p-adic valuations. If valuation theory is to help us find a valuation to reduce error, then one of these must be used. The standard absolute value gives us the representation we have been using, so that will not decrease error. The trivial valuation gives an error of 0 if the "approximation" is exactly correct, and 1 otherwise. It can be used to note of the answer is exact, but not for any sort of approximation. The only hope of getting a better valuation for approximations lies in using a p-adic valuation.

P-ADIC VALUATIONS

To help eliminate error in computation using valuation theory, what is left to look at are the p-adic valuations. Do these valuations actually reduce error? Since all p-adic valuations function in the same way, we will fix a prime p for the rest of this section. Recall from previous work that, given two real numbers r and s, and their rational approximations $\tilde{r}$ and $\tilde{s}$, within an error estimate of ϵ, the error on addition, $|(r+s)-(\tilde{r}+\tilde{s})|$, is less than 2ϵ. Also, the error in multiplication was $\epsilon(|s|+|\tilde{r}|)$.

However, given a p-adic valuation, the error turns out differently. Using the same approximations, with $\phi_p(r-\tilde{r}) < \epsilon$ and $\phi_p(s-\tilde{s}) < \epsilon$, for some $\epsilon < 1$,

$$\phi_p((r+s)-(\tilde{r}+\tilde{s})) = \phi_p((r-\tilde{r})+(s-\tilde{s})) < \max\{\phi_p(r-\tilde{r}), \phi_p(s-\tilde{s})\} \leq \epsilon.$$

So, for addition, the amount of error in the sum is no worse than the original error. In multiplication,

$$\phi_p(rs-\tilde{r}\tilde{s}) = \phi_p(s(r-\tilde{r})+\tilde{r}(s-\tilde{s})) < \epsilon \max\{\phi_p(s), \phi_p(\tilde{r})\},$$

so that the error estimate in multiplication is less than it was in the original case.

What is needed now is a way to represent numbers p-adically. This can be done by defining a metric on the rational numbers $\mathbf{Q}$, based on a p-adic valuation, and using that to define a completion of the rational numbers. Define a metric $d : \mathbf{Q} \times \mathbf{Q} \to \mathbf{R}$ by $d(x,y) = \phi_p(x-y)$. This is a metric since

$$(i) \qquad d(x,y) = 0 \iff x = y,$$

$$(ii) \qquad d(x,y) = d(y,x), \text{ and}$$

$$(iii) \qquad d(x,z) \leq d(x,y) + d(y,z).$$

This construction can be done for any valuation, not just a p-adic or non-archimedean one.

Next, we need to define the completion of the rational numbers. Define a relation $\sim$ on the set of Cauchy sequences[1] of rational numbers by $a \sim b$ if and only if $d(a_i, b_i) \to 0$ as $i \to \infty$. $d(a_i, a_i) = 0$ for all i, so $\sim$ is reflexive, and $d(a_i, b_i) = d(b_i, a_i)$, since $\phi_p(x) = \phi_p(-x)$, so that $\sim$ is symmetric. Also notice that $d(a_i, c_i) \leq d(a_i, b_i) + d(b_i, c_i)$, so that if $d(a_i, b_i)$ and $d(b_i, c_i)$ both approach zero, then their sum must also, so that $\sim$ is transitive, or an equivalence relation. Thus, the set of equivalence classes can be treated as a set of numbers. Call this set $\mathbf{Q}_p$. This is the completion of the rationals using the p-adic valuation, instead of the standard absolute value. In contrast, doing this same completion using the standard absolute value gives $\mathbf{R}$.

If q is a rational number, then the equivalence class of the constant sequence $\{q, q, q, \ldots\}$ is an element of $\mathbf{Q}_p$. Constant rational sequences are always a positive distance apart, so no two constant rational sequences are equivalent to each other in this equivalence relation. Arithmetic is defined term-wise on the sequences, so that the set of equivalence classes of constant sequences is isomorphic to the rational numbers. Thus the field of p-adic numbers has a subfield isomorphic to $\mathbf{Q}$. In fact, this set is dense in $\mathbf{Q}_p$, since $\mathbf{Q}_p$ completes $\mathbf{Q}$.

This new set $\mathbf{Q}_p$ forms a field, and the valuation ϕ_p can be extended to that field in the following way: If $\{a_n\}$ is a Cauchy sequence, then

$$\phi_p(a_n) = \phi_p(a_m + (a_n - a_m)) \leq \max\{\phi_p(a_m), \phi_p(a_n - a_m)\} = \phi_p(a_m)$$

for m, $n > N$ for some N. Similarly,

$$\phi_p(a_m) = \phi_p(a_n + (a_m - a_n)) \leq \max\{\phi_p(a_n), \phi_p(a_m - a_n)\} = \phi_p(a_n)$$

so that, for some N, if m, $n > N$, then $\phi_p(a_m) = \phi_p(a_n)$, or the sequence's terms eventually have some constant valuation p-adically. This constant value is the

[1] A sequence $\{x_0, x_1, x_2, \ldots\}$ is a Cauchy sequence if and only if for every $\epsilon > 0$ there is some N such that for all $m,n > N$, $d(x_m, x_n) < \epsilon$.

p-adic valuation of the Cauchy sequence. In fact, this also says that the p-adic valuation of any p-adic number is still some power of p, just like it was for the rational numbers.

There is an alternative representation for these p-adic numbers. The proof of this representation is modified from one in Bachman [1].

Theorem 7. *Any p-adic number a can be represented uniquely as*

$$a = \sum_{j=n}^{\infty} a_j p^j$$

for some a_j in $\{0, 1, \ldots, p-1\}$, with n such that $\phi_p(a) = p^{-n}$.

Proof: Let $a \in \mathbf{Q}_p$. If $a = 0$, $\phi_p(a) = 0$. Otherwise, since $\mathbf{Q}$ is dense in $\mathbf{Q}_p$, there is a Cauchy sequence $\{a_n\}$ of elements in $\mathbf{Q}$ converging to a, such that $\lim a_n = a$. By the p-adic valuation on p-adic numbers, $\phi_p(a_N) = \phi(a)$ for some N. Thus,

$$\phi_p(\mathbf{Q}_p) = \phi_p(\mathbf{Q}) = \{p^n | n \in \mathbf{Z}\}.$$

So, $\phi_p(a) = p^{-n} = \phi_p(p^n)$ for some integer n. Thus, $\phi_p(\frac{a}{p^n}) = 1$. Let $b = \frac{a}{p^n}$. Since $b \in \mathbf{Q}_p$, there is a sequence c_k in $\mathbf{Q}$ which converges to b. So, for some integer $N > 0$, $\phi_p(b - c_k) < 1$ for $k \geq N$. So,

$$\phi_p(c_N) = \phi_p(b + (c_N - b)) \leq \max\{\phi_p(b), \phi_p(c_N - b)\} = \phi_p(b) = 1$$

and

$$\phi_p(b) = \phi_p(c_N + (b - c_N)) \leq \max\{\phi_p(c_N), \phi_p(b - c_N)\} = \phi_p(c_N)$$

for some large value of N, so that

$$\phi_p(c_N) = \phi_p(b) = 1$$

for large values of N, and $c_N \in \mathbf{Q}$.

Let $c_N = \frac{e}{d}$ for some integers e and d, with both e and d relatively prime to p. (Note that otherwise, c_N has a power of p as a factor, and $\phi_p(c_N) \neq 1$.) So, d and p are relatively prime. This means that there are integers x and y such that

$$xd + yp = 1, \text{ or } xd \equiv 1 \pmod{p}.$$

Then

$$\phi_p(\frac{e}{d} - ex) = \phi_p(\frac{e(1-dx)}{d}) = \phi_p(\frac{eyp}{d}) = \phi_p(\frac{e}{d})\phi_p(yp) = \phi_p(yp) \leq \frac{1}{p} < 1.$$

Define a_n as ex, so that $\phi_p(a_n - b) < 1$, or

$$\phi_p(a_n p^n - bp^n) < \phi_p(p^n),$$

or

$$a = bp^n = a_n p^n + (b - a_n)p^n = a_n p^n + g$$

for $g = (b - a_n)p^n$, where $\phi_p(g) < \phi_p(p^n)$, so that $\phi(g) = \phi_p(p^m)$ for some $m > n$. This procedure can be continued, using g as a, in order to get, at stage k,

$$a = a_n p^n + a_{n+1}p^{n+1} + \cdots + a_{n+k-1}p^{n+k-1} + g_k,$$

with $a_i \in \mathbf{Z}$ and $\phi_p(a_i) = 1$ or $a_i = 0$. Since $\phi_p(p^{n+k}) \to 0$ as $k \to \infty$, then this sequence of sums converges to a, so that

$$a = \sum_{j=n}^{\infty} a_j p^j.$$

Any a_j can be written as $\tilde{a}_j + pl$ for some l, and this can be used to redistribute the a_j so that they are all in $\{0, 1, \ldots, p-1\}$. This is unique, since this cannot be redistributed. Q. E. D.

So, these p-adic numbers can be treated as infinite sums, or as a sequence $a = (a_n, a_n + 1, \ldots)$ of coefficients in $\{0, 1, \ldots, p-1\}$. This form is useful, since

it both determines the p-adic number uniquely, and can be represented in a computer more easily than can an equivalence class of Cauchy sequences.

The summation form of these numbers is very similar to the "summation" form of the decimal expansion of a real number. In the case of the real numbers, using base-10 notation, the form is

$$\sum_{j=n}^{\infty} a_j 10^{-j},$$

which has negative exponents instead of positive exponents. Addition and multiplication on these p-adic numbers work in a way similar to arithmetic on the decimal expansion of real numbers: addition by adding the lowest term, carrying to the right, and so on, and multiplication by shifting and multiplying.

There are, in some sense, advantages to doing this in the p-adic form: there is always a smallest, or leftmost, place to start in p-adic arithmetic, while in real numbers, there may be no smallest place. Also, for p-adic numbers, this representation is unique, while for the real number expansion, there are two possible expansions for some numbers, such as $1 = 0.9999\ldots$

In order to avoid confusion as to where the above sequence of coefficients starts, it cannot just be written as $(a_n, a_{n+1}, \ldots)$, because the power n will be lost. Instead, write a number a as

$$a = a_n a_{n+1} \cdots a_{-1} . a_0 a_1 a_2 \cdots,$$

with the period indicating the 0 term, identifying what the value n is. If $n > 0$, then add extra zero terms to the front to get the period in the correct location.

One important thing to notice, in order to do represent numbers, is that in the p-adic metric,

$$1 = \lim_{n \to \infty} 1 + p^n = \lim_{n \to \infty} (1-p)(1 + p + p^2 + \cdots + p^{n-1}),$$

so that

$$\frac{1}{(p-1)} = 1 + p + p^2 + p^3 \cdots$$

in the p-adic metric. So, given the expansion

$$a = 2 + 3p + p^2 + 3p^3 + p^4 + \cdots,$$

we find that

$$\begin{aligned} a &= 2 + 3p(1 + p^2 + p^4 + \cdots) + p^2(1 + p^2 + p^4 + \cdots) \\ &= 2 + (3p + p^2)(1 + p^2 + p^4 \cdots) = 2 + \frac{3p + p^2}{1 - p^2}. \end{aligned}$$

Thus, supposing that $p = 7$, $a = 2 + \frac{21+49}{1-49} = 2 - \frac{35}{24}$. So, 7-adically,

$$\frac{13}{24} = .231313131\ldots$$

and, similarly, other fractions can be found.

Since this notation is so similar to the regular real-number notation of decimal expansion, this can be used in computers to help reduce computational error. This "decimal" expansion of a p-adic number can be truncated in a way similar to that used for real-number binary expansions in computer representations. Arithmetic can be defined very similarly to the real-number case. This uses about the same storage space and computational complexity, and it will result in less error. To be usable, the computer must be able to "translate" between p-adic numbers and real numbers in an efficient manner. These obstacles can be overcome. The p-adic valuation on the rationals will result in a usable method of containing error.

FINITE SEGMENT P-ADIC NUMBERS

As seen in the last section, any p-adic number a can be written as a series

$$a = \sum_{j=n}^{\infty} a_j p^j.$$

However, for computing purposes, it is not possible to store all of the infinitely many coefficients of this sum. Therefore, numbers will be stored in a form which cuts off this sum at a certain point. So, the approximation to a will be $\tilde{a}$, which is defined as

$$\tilde{a} = \sum_{j=n}^{m} a_j p^j$$

for some number m. This is what is called a finite segment p-adic number. P-adically, this is a good approximation, since

$$\phi_p(a - \tilde{a}) = \phi_p(\sum_{j=m+1}^{\infty} a_j p^j) \leq \max_{j>m}\{\phi_p(a_j p^j)\}$$

$$= \max_{j>m}\{\phi_p(a_j)\phi_p(p^j)\} = \max_{j>m, a_j \neq 0}\{p^{-j}\} \leq p^{-m}.$$

This approximation will be better if a larger m is chosen.

To identify this number, the notation

$$a_m a_{m-1} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_n$$

will be used. This is backwards from the original decimal-like notation used before, but is used so that the parallels between arithmetic on these numbers and arithmetic on decimal expansions of real numbers are clearer. In fact, arithmetic on this expansion is almost identical to arithmetic on expansions of real numbers.

To see how to add two of these numbers together, look at the summation notation. For two approximations a, and b.

$$a + b = \sum_{j=n_1}^{m} a_j p^j + \sum_{j=n_2}^{m} b_j p^j = \sum_{j=n}^{m} (a_j + b_j) p^j$$

where $n = \min\{n_1, n_2\}$. This sum may not be in the standard form, since $a_j + b_j$ may be larger than $p - 1$. To fix that, it is necessary to redistribute the terms. For some integers d_j and c, $a_j + b_j = d_j + c \cdot p$, so that in redistributing, c gets "bumped up" or "carried" to the next term in the sum. This carrying is exactly the same as the carrying done in real numbers. This is not surprising, because our approximation to the p-adic number can be viewed as a base-p rational number. There is one slight difference, however. At the end of decimal arithmetic, there is sometimes a carry which goes into a place larger than one which was present in the original number. In the p-adic approximation, that carry would go into a p^{m+1} term, which is excluded from the representation, so that term is just dropped.

Multiplication is also done nearly the same as multiplication for decimal expansions. Again, this occurs because the p-adic approximation can be viewed as a base p rational number, and arithmetic is done the same way on both representations. However, just as before, any terms which are "too big" are just dropped. For instance, when multiplying two numbers a and b, with largest term p^m, there is a possibility of getting a p^{2m} term. All the terms from p^{m+1} to p^{2m} are dropped. The result is still a good approximation, since p-adically, these terms are very small.

Negation is defined differently than it is in the real number case. With real numbers, you just change the sign of the number to get the opposite of it. In p-adic arithmetic, there is no need for a sign change, since all numbers, including negative ones, can be written in sum form. Given a number a, define $-a$ as

$$-a = (p - an)p^n + \sum_{j=n+1}^{\infty} (p - (a_j + 1))p^j,$$

where

$$a = \sum_{j=n}^{\infty} a_j p^j.$$

Thus, the first term $a_n + (p - a_n) = p$, so the initial term is 0 and a 1 gets carried to the next term, and each term after that is $a_j + (p - (a_j + 1)) + 1 = a_j + p - a_j - 1 + 1 = p$, so that the jth term of the sum is also 0, and a 1 still gets carried to the next term.

Thus, $a + -a = 0$, which means $-a$ is the additive inverse of a. This number can be found, in the same way, up to the mth term without much difficulty. The result when the approximations are added will be p^{m+1}, which is close to 0 p-adically. Subtraction can then be defined as addition using the additive inverse of the second term.

Division can be done in a "long division" fashion. The procedure is as follows:

1. Get the multiplicative inverse modulo p of the p^n coefficient of the divisor, where n is the smallest integer where the p^n coefficient is non-zero. Call it b.
2. Get the lowest-term coefficient of the current partial remainder. (The partial remainder is initially the dividend.) Call the coefficient c_k at the kth step.
3. Find $a_k = b \cdot c_k$, reduced modulo p. This is the coefficient of the kth-lowest term in the quotient. (The lowest term in the quotient will be the $p^{\tilde{n}-n}$, where n is as above, and $\tilde{n}$ is similarly defined for the dividend.)
4. The new partial remainder is the old partial remainder, minus a_k times the divisor.
5. Repeat this process until the partial divisor is zero.

See Gregory and Krishnamurthy [3] for details on this process.

Of course, for the infinite-sum p-adic number, this procedure will not necessarily stop. But in the finite-segment case, where the p-adic number is approximated to the p^m term, the procedure will stop when that coefficient has been reached. This will stop in a finite amount of time, so this division algorithm can

be used in computer programs. Any larger terms which would come from this method are p-adically small.

These procedures will give a consistent system, but it may be unclear how this will reduce error. For instance, addition and multiplication are exactly the same as the real case, and if you look at the approximation formula, rational numbers are expressed in exactly the same form as they would be in the base-p system. In fact, with the approximation, it is the largest terms (in absolute value) which are being ignored. The answer lies in the differences between the p-adic and the standard valuations.

As an example, recall that to 6 places, the 7-adic representation of $\frac{13}{24}$ was .231313. In the reversed notation, this would be 313132. The number 53923 (decimal for 313132 base 7) will also have this representation. These two numbers are very close p-adically, but are very far apart in the real valuation. To see that the numbers are close in the p-adic metric, notice that

$$53923 - \frac{13}{24} = \frac{1294139}{24} = \frac{7^6 \cdot 11}{24},$$

which has 7-adic valuation of 7^{-6}, a small number. So, closeness of numbers p-adically tells us nothing about how close the numbers are in the standard sense. In fact, any two rational numbers whose difference has a large power of p in the numerator, and no power of p in the denominator, will be close p-adically. Since the denominator can be any number relatively prime to p, the numbers that are close in the p-adic sense make up a set which is dense in the standard sense. So, there seems to be no way to determine which real number a p-adic approximation is. Even if we know an estimate on the number we're looking for, there are still infinitely many numbers p-adically near the estimate.

However, there is a method of determining unique answers out of this system. By restricting which numbers are used, a one-to-one mapping between the

representable numbers and the p-adic approximations can be found. The set of representable numbers will be what is called the Farey fractions of order N, denoted $\mathbf{F}_N$, with the restriction that $2N^2 + 1 \leq p^{m+1}$. This restriction can make some computer computations more exact.

COMPUTER USE OF FINITE-SEGMENT 2-ADIC NUMBERS

As seen, finite-segment p-adic arithmetic is very similar to decimal-expansion real arithmetic, done in base p. So, in order to get the best results out of built-in computer arithmetic, which is done in base 2, the 2-adic numbers will give the best result, in terms of computational complexity. However, as stated earlier, to overcome the difficulties in p-adic valuations, the set of possible numbers must be restricted.

DEFINITION: The *Farey fractions of order* N, or $\mathbf{F}_N$, is defined as[2]

$$\mathbf{F}_N = \{\frac{a}{b} \in \mathbf{Q} : |a|, |b| \leq N\}.$$

The usefulness of these Farey fractions will be seen a little later.

The problem with the finite-segment approximation was that two largely different numbers could be the same in the approximation. To see why this occurs, let a and b be two p-adic numbers with representations the same up to some term m. Then

$$a - b = (\sum_{j=n}^{m} a_j p^j + \sum_{j=m+1}^{\infty} a_j p^j) - (\sum_{j=n}^{m} a_j p^j + \sum_{j=m+1}^{\infty} b_j p^j)$$

$$= (\sum_{j=m+1}^{\infty} (a_j - b_j) p^j = p^{m+1} \cdot \sum_{j=m+1}^{\infty} a_j p^{j-(m+1)}.$$

This number is divisible by p^{m+1}, or $a \equiv b \pmod{p^{m+1}}$. Thus, a finite segment p-adic number is only determined up to rational numbers divisible by p^{n+1}. The Farey fractions are useful for this reason: Let $\frac{a}{b}, \frac{c}{d} \in \mathbf{F}_N$ be different, but with

[2] The definition of the Farey fractions is slightly different in some sources; it is sometimes defined as all rational numbers between 0 and 1 with denominator up to N.

the same finite p-adic representation to the mth term. Then

$$\frac{a}{b} \equiv \frac{c}{d} \pmod{p^{m+1}}, \text{ and } |a|, |b|, |c|, |d| \leq N, c, d \neq 0.$$

From this comes that

$$ad \equiv bc \pmod{p^{m+1}}, \text{ or } ad - bc \equiv 0 \pmod{p^{m+1}}.$$

Since the two fractions were not equal, $ad - bc \neq 0$, so that $ad - bc \geq p^{m+1}$. Now, since $\frac{a}{b}$ and $\frac{c}{d}$ are Farey fractions of order N, $|ad - bc| \leq 2N^2$, so that $2N^2 \geq p^{m+1}$.

To keep two Farey fractions in $\mathbf{F}_N$ from having the same approximation, require that $2N^2 < p^{m+1}$, or that $2N^2 + 1 \leq p^{m+1}$. This requirement on N means that for $\mathbf{F}_N$, all the numbers will have unique p-adic approximations to the mth place.

In the particular case where $p = 2$, this means that the Farey fractions of order N have unique 2-adic approximations to the mth place if $2N^2 + 1 \leq 2^{m+1}$. If all numbers in a computation are approximated to the mth term 2-adically, and the true answer is a Farey fraction of order N, then the answer will be exactly correct, because each element of $\mathbf{F}_N$ has one, and only one, approximation. There will be absolutely no error on the result, since the unique Farey fraction with that approximation can be found. Even if intermediate results in the calculations give p-adic numbers which cannot be uniquely determined, when the final result is found, it will be accurate.

Of course, if the final answer is not in the Farey fraction set, then the answer will be incorrect. Sometimes, the answer will be the approximation of some Farey fraction, but the true answer will not be equal such a Farey fraction. In that case, confusion can occur since the computer gives an answer to a problem, and that answer is incorrect. In addition, there are many more possible representations than there are Farey fractions of order N, (There are p^m possible

representations, and less than N^2 Farey fractions, and $2N^2 < p^m$) so an answer which is not an appropriate Farey fraction may not match up to any Farey fraction approximation. In this case, some method of finding an answer should be found. However, there are an infinite number of possibilities for the number, and no way of finding which one is the correct answer. Perhaps the best solution would be to give an error, stating there is no solution, similar to computer errors from dividing by zero.

For actual computer storage and calculation, there are a few modifications which must be made to the procedure stated so far. For instance, every 2-adic number has a smallest term, the nth term, which is sometimes negative, but is finite. However, this n can be any integer. In order to store this number reasonably in a computer, floating point notation is used. The number is stored as the first m non-zero places (called the mantissa), with an exponent, which would store the value of n. This is roughly the same way that real number expansions are stored when floating-point numbers are used on computers. The 2-adic case will actually give a little more accuracy than the real-number case, since no storage is needed for the sign of the number. Multiplication and division are mostly unchanged; the mantissas are multiplied (or divided), and the exponents are added (or subtracted).

Negation of numbers is also mostly unchanged; just the mantissa is negated. However, addition must be slightly altered. When adding numbers, it must be certain that the numbers are added at the correct location. Thus, unless the exponents are the same, one of the mantissas must be shifted before addition can take place. The end result has the exponent which is the smallest of the two original exponents. The shifting should be done so that the highest parts of the mantissa are lost, as they are the smallest 2-adically.

This addition result will sometimes have a zero in the lowest place. Traditionally, this number would be shifted down (so as not to have such a zero), and the exponent altered. However, if that is done, the mantissa will be accurate to less than m places, and this loss of precision must be taken into account. Note that this is the only possible form of error which can occur in arithmetic with these numbers. If the zeroes are kept, then no precision is lost, but the division algorithm will be slightly more complex, since it needs to start at the first non-zero coefficient, which, if shifting, would always be in the lowest place in the mantissa.

Most of the arithmetic operations actually are quite easily done on a computer, if the m is chosen to be some common data size on the computer, such as 16, 32 or 64. In the computer package I wrote, I used a 32-bit size for the mantissa. The mantissa can (and should) be stored as an integer, in which case addition, multiplication and negation all turn out to be the same as the built-in integer operations. The extra carries in addition and multiplication are automatically truncated in the integer computation, and negation of the mantissa is exactly two's complement negation, which is the negation used in standard computer integer arithmetic. These routines give computational complexity comparable to regular arithmetic, since the routines to do most of this are mostly the same.

The routine I used for addition was the non-shifting method, which stores numbers with zeros in the last places. This keeps the accuracy without extra storage for the precision. This also makes the division algorithm as described more complex than otherwise, since the last bit isn't always a 1. However, the division algorithm I used wasn't the one given. I used a method which was more complex, but easier to code. Inversion (finding the multiplicative inverse) was defined first, and then division was multiplication by the inverse of the divisor.

Inversion was done by negating the exponent, and, using the mantissa m, finding the representation for the fraction $\frac{1}{m}$. This gives an inverse which is accurate to the required number of places. However, since the conversion routines are not very simple, the time needed to do this is too large. The above division algorithm should be used for time-intensive programming.

Converting to and from the 2-adic form was also necessary. Converting from 2-adic to rational form was not too difficult. Since all even denominators were factored into the exponent, the program looks at all possible odd denominators up to the bound N on the Farey fractions, (For 32 bits, this number was 46340) starting with 1. If some numerator came out also in the designated range, then that Farey fraction was the answer. If no such fraction came out, the fraction with the lowest numerator was given. This last answer is by no means necessarily accurate, and better results may occur if an error message were given instead.

Converting to 2-adic form takes a little more effort. First, all powers of two are factored out of the numerator and denominator, and used to make the exponent term. Then what is done is a version of the division algorithm as originally explained, which is made to work in particular on two odd integers, as opposed to two arbitrary 2-adic numbers. This procedure will give the correct 2-adic representation, since odd positive binary integers are stored the same way as their floating point 2-adic counterparts. If the fraction is negative, the result is negated at the end.

2-adic arithmetic can be useful in eliminating error. In addition, this arithmetic is done fairly easily on computers, so that the computations will take about the same amount of time as standard arithmetic. The storage space needed, compared to other methods of finding exact answers, is low. While not helpful as a way to reduce error in real numbers, p-adic arithmetic is still useful for reducing error in computation.

SUMMARY AND CONCLUSION

The p-adic valuations, and in particular the 2-adic valuation, can be used to reduce error, if the answer will be a rational number of a certain form. While this is a limitation, there are some difficult problems which can be solved using this method. The Hilbert matrix, for instance, is a matrix which has rational entries, and the usual methods of computation give very poor answers. However, if done 2-adically, the exact answer comes out, using far less storage space than would be necessary to get a close answer using normal computational approximation.

In general, 2-adic arithmetic is useful for finding exact answers to matrix problems, where the matrix has rational entries. In such a problem, however, it must be certain that the answer will fall into the set of Farey fractions of the correct order. In some problems, it may be possible to get an upper bound on the size of the numerator and denominator of the fractional answer. That will determine the order of the set of Farey fractions needed. Then, using the order determined, decide how big the 2-adic mantissa needs to be in order to be accurate. This method will allow exact answers to any such problem, but the precision of the 2-adic numbers must be able to be increased. Using computer packages which allow integers of any size, this can be done. Setting an upper bound on the size of the integers keeps the size of the number from getting too large, so that the amount of storage space needed is not prohibitive.

If a problem gives a real number solution, there still can be hope. By using rational approximations to the real numbers, the above procedure of determining how precise the 2-adic numbers need to be will work for the approximate problem. Then the only error in the answer will come from the original approximation. For instance, $\frac{22}{7}$ is a good approximation of π for many orders of the Farey

fractions. Using this estimate can give an answer which is accurate, with the exception of the error in the approximation of π. The error in π can accumulate, but the answer is at least as good as the real number floating-point calculation. In addition, standard arithmetic must use a denominator which is a power of 2. Denominators whic are not powers of 2 often give better approximations for numbers like π. For instance, $\frac{22}{7}$ is a much better approximation for π than $\frac{25}{8}$ is. Using p-adic arithmetic can give improved results for real arithmetic since it can handle different rational approximations.

Thus for many problems, particularly where exactness is important, the 2-adic valuation gives a method which can give exact answers, or close to exact answers, without a large amount of computational complexity or storage space. This can be a great help in many applications of computers, particularly in problems where a small change in the inputs to a calculation can give a large variance in the output. The 2-adic valuation can be very useful in computing answers to problems.

BIBLIOGRAPHY

[1] Bachman, George. *Introduction to p-Adic Numbers and Valuation Theory.* Academic Press, New York, 1964.

[2] Endler, Otto. *Valuation Theory.* Springer-Verlag, New York, 1972.

[3] Gregory, R. T. and E. V. Krishnamurthy. *Methods and Applications of Error-Free Computation.* Springer-Verlag, New York, 1984.

[4] Weiss, Edwin. *Algebraic Number Theory.* McGraw-Hill Book Company, Inc., San Francisco, 1963.