<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.R. Finnie</td>
<td>Modeller: A DSS for Experimental Study of Human-Computer Interaction</td>
<td>1</td>
</tr>
<tr>
<td>S.W. Postma</td>
<td>New CS Syllabus at Natal (Pmb)</td>
<td>7</td>
</tr>
<tr>
<td>C.C. Handley</td>
<td>Finding All Primes Less than Some Maximum</td>
<td>15</td>
</tr>
<tr>
<td>A-K. Heng</td>
<td>Some Remarks on Deleting a Node from a Binary Tree</td>
<td>21</td>
</tr>
<tr>
<td>A-K. Heng</td>
<td>A Summation Problem</td>
<td>23</td>
</tr>
<tr>
<td>S. Berman</td>
<td>A Persistent Programming Language - Preliminary Report</td>
<td>25</td>
</tr>
</tbody>
</table>

**BOOK REVIEWS**

14
33

An official publication of the Computer Society of South Africa and of the South African Institute of Computer Scientists

'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Africa en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes
Editor

Professor G. Wiechers
INFOPLAN
Private Bag 3002
Monument Park 0105

Editorial Advisory Board

Professor D.W. Barron
Department of Mathematics
The University
Southampton S09 5NH, England

Professor J.M. Bishop
Department of Computer Science
University of the Witwatersrand
1 Jans Smuts Avenue
Johannesburg, 2001

Professor K. MacGregor
Department of Computer Science
University of Cape Town
Private Bag
Rondebosch, 7700

Dr H. Messerschmidt
IBM South Africa
P.O. Box 1419
Johannesburg, 2000

Subscriptions

Annual subscription are as follows:

<table>
<thead>
<tr>
<th></th>
<th>SA</th>
<th>US</th>
<th>UK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuals</td>
<td>R10</td>
<td>$7</td>
<td>£5</td>
</tr>
<tr>
<td>Institutions</td>
<td>R15</td>
<td>$14</td>
<td>£10</td>
</tr>
</tbody>
</table>

Circulation and Production

Mr C.S.M. Mueller
Department of Computer Science
University of the Witwatersrand
1 Jan Smuts Avenue
Johannesburg, 2001

Quaestiones Informaticae is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.
A major concern in numerical analysis is the question how an error at one point in a calculation propagates, that is, whether its effect becomes greater or smaller as subsequent operations are carried out. Analysis of error in a numerical result is fundamental to any intelligent computation, whether done manually or with a computer. Every result in a computer must be expressed in some finite number of digits, and this simple appearing restriction completely changes many of the standard mathematical ideas.

Suppose we are given \( N \) positive numbers and we wish to find their sum. If we assume that these \( N \) numbers can be represented exactly in a machine (no conversion error, e.g. convert from decimal number to hexadecimal number), we can prove that if we arrange these numbers in increasing sequence, the total error due to roundoff will be reduced.

We define addition is EXACT if there is no error (truncation error, rounding error, ...) during adding of numbers. We are going to define a method in order to minimize the error if we wish to add \( N \) positive numbers.

Let

\[
a_1, a_2, a_3, \ldots, a_N
\]

be \( N \) positive numbers, these numbers can be represented exactly in a machine and the numbers are in increasing sequence.

Using the above assumptions, if we add from small number to large number, we know that there exists one \( L \) such that

\[
a_1 + a_2 + \ldots + a_L \quad \text{is exact and}
\]

\[
a_1 + a_2 + \ldots + a_L + a_{L+1} \quad \text{is not exact.}
\]

Normal addition from left to right, we will have \( N-L \) roundoff errors of additions. The problem now is to see whether we can reduce the number of roundoff errors.

The method we are going to use is what we call "Partition-Method". We will split the given addition into several partitions, denoted by \( P_I \) for the \( I \)-th partition. Now define

\[
P_0 = a_1 + a_2 + a_3 + \ldots + a_{K_0}
\]

\[
P_1 = a_{K_0+1} + a_{K_0+2} + \ldots + a_{K_1}
\]

\[
P_2 = a_{K_1+1} + a_{K_1+2} + \ldots + a_{K_2}
\]

\[
\vdots
\]

\[
P_{J} = a_{K(J-1)+1} + a_{K(J-1)+2} + \ldots + a_N
\]

If each partition is the maximum number of terms that the sum is exact, then adding up from \( P_0 \) to \( P_{J} \), we have reduced the total number of roundoff errors from \( N-L \) to \( J \), and the total number of partitions is also minimized.

Example 1: \( 1 + 2 + 3 + \ldots + 15000 \)

Working on IBM/3033 machine, this computer uses 4-bytes to represent a floating number (first byte is for the characteristic part in excess 64, the next three bytes are for the mantissa). Using binary search, we obtain the following partitions.
### Example 2:

To find the sum of the scalar product of \((1 \ 2 \ 3 \ldots \ n)(1 \ 2 \ 3 \ldots \ n)\) where \(n=925\)

We obtain 16 partitions:

<table>
<thead>
<tr>
<th>Partition</th>
<th>Terms</th>
<th>Hexadecimal</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>FROM 1 TO 5792</td>
<td>46FFFD50</td>
<td>0.16776528 E+08</td>
</tr>
<tr>
<td>P1</td>
<td>FROM 5793 TO 8192</td>
<td>4710012B</td>
<td>0.16782000 E+08</td>
</tr>
<tr>
<td>P2</td>
<td>FROM 8193 TO 10033</td>
<td>46FFFF49</td>
<td>0.16777033 E+08</td>
</tr>
<tr>
<td>P3</td>
<td>FROM 10034 TO 11585</td>
<td>46FFFC98</td>
<td>0.16776344 E+08</td>
</tr>
<tr>
<td>P4</td>
<td>FROM 11586 TO 12952</td>
<td>46FFEABA8B</td>
<td>0.16771723 E+08</td>
</tr>
<tr>
<td>P5</td>
<td>FROM 12953 TO 14188</td>
<td>46FF0F12</td>
<td>0.16773138 E+08</td>
</tr>
<tr>
<td>P6</td>
<td>FROM 14189 TO 15000</td>
<td>46B4D3EE</td>
<td>0.11850734 E+08</td>
</tr>
</tbody>
</table>

The exact solution = 476B4BA6(C) or 0.11250750 E+09
Normal calculation = 476B3ACC or 0.11243846 E+09
Partition method = 476B4BA4 or 0.11250745 E+09

The roundoff errors of additions have reduced from 9208 to 6!

### Example 3:

To find the value of \(p(x) = a_0x^n + a_1x^{n-1} + \ldots + a_n\)

with \(n=12, \ x=6.0\) and \(a_i=i-1\) for \(i=1, \ldots, 13\)

Normal calculation using Horner's rule, the result obtained is equal to 48BAD5AE in hexadecimal or 0.313456589 E+10 in decimal. The exact solution is equal to 48BAD5B0(A1) in hexadecimal or 0.3134566561 E+10 in decimal. If we split the polynomial into two parts:

\[
p(x) = \left[ a_1x^8 + a_2x^7 + \ldots + a_9x^4 \right] + \left[ a_{10}x^3 + a_{11}x^2 + \ldots + a_{13} \right]
\]

The value inside the square brackets is exact, the result has been improved to 48BAD5B1 in hexadecimal or 0.313456666 E+10 in decimal.