PROCEEDINGS OF THE SEVENTH SIAM CONFERENCE ON

PARALLEL PROCESSING FOR SCIENTIFIC COMPUTING

Edited by David H. Bailey
NASA Ames Research Center

Petter E. Bjørstad
University of Bergen

John R. Gilbert
Xerox Palo Alto Research Center

Michael V. Mascagni
Supercomputing Research Center,
Institute for Defense Analyses

Robert S. Schreiber
Research Institute for Advanced
Computer Science

Horst D. Simon
NASA Ames Research Center
Center for Supercomputing

Virginia J. Torczon
Rice Univeristy

Layne T. Watson
Virginia Polytechnic Institute and
State University

siam. Philadelphia

Society for Industrial and Applied Mathematics

PARALLEL ALGORITHMS FOR FINDING TRIGONOMETRIC SUMS *

PRZEMYSLAW STPICZYŃSKI † AND MARCIN PAPRZYCKI ‡

Abstract. Parallel versions of Goertzel and Reinsch algorithms for finding trigonometric sums are introduced as a special case of effcient parallel algorithms for solving linear recurrence systems. The results of the experiments performed on a 20-processors Sequent Symmetry are presented and discussed.

Key words. Trigonometric sums, linear recurrence systems, parallel algorithms, shared-memory parallel computers, speedup.

1. Introduction. In several mathematical problems such as trigonometric interpolation we need to compute sums

$$C(x) = \sum_{k=0}^{n} b_k \cos kx, \qquad S(x) = \sum_{k=1}^{n} b_k \sin kx. \tag{1}$$

There are two well-known algorithms for finding solutions of (1): Reinsch's algorithm [3], which works for any value of x, and Goertzel's algorithm [3], which can be applied for $|x|$ not too close to zero. These algorithms transform the original problem (1) to the solution of a linear recurrence system of order 2:

$$x_k = \begin{cases} 0 & \text{if } k \le 0 \\ f_k + a_{k,k-2}x_{k-2} + a_{k,k-1}x_{k-1} & \text{if } 1 \le k \le N. \end{cases} \tag{2}$$

Such a recurrence system can be efficiently solved on parallel computers using slight modifications of the recently proposed parallel algorithms [2, 4, 5, 6].

2. Parallel algorithms. In case of the Goertzel algorithm we need to compute the following linear recurrence system with constant coefficients:

$$S_k = \begin{cases} 0 & \text{for } k = n+1, n+2, \\ b_k + 2S_{k+1}\cos x - S_{k+2} & \text{for } k = n, n-1, \ldots, 1, \end{cases} \tag{3}$$

and then we compute $C(x) = b_0 + S_1 \cos x - S_2$ and $S(x) = S_1 \sin x$.

In Reinsch algorithm we set $S_{n+2} = D_{n+1} = 0$ and if $\cos x > 0$ then we solve

$$\begin{cases} S_{k+1} = D_{k+1} + S_{k+2} \\ D_k = b_k + eS_{k+1} + D_{k+1}, \end{cases} \tag{4}$$

for $k = n, n-1, \ldots, 0$, where $e = -4\sin^2 \frac{x}{2}$. If $\cos x \le 0$ then we solve

$$\begin{cases} S_{k+1} = D_{k+1} - S_{k+2} \\ D_k = b_k + eS_{k+1} - D_{k+1}, \end{cases} \tag{5}$$

where $e = 4\cos^2 \frac{x}{2}$. Finally, we compute $C(x) = D_0 - \frac{e}{2}S_1$ and $S(x) = S_1 \sin x$.

* This paper is an extended abstract.

† Department of Numarical Analysis, Marie Curie-Sklodowska University, Pl. Marii Curie-Sklodowskiej 1, 20-031 Lublin, POLAND (przem@hektor.umcs.lublin.pl).

‡ Department of Mathematics and Computer Science, University of Texas of the Permian Basin, Odessa, TX 79762 (paprzycki_m@gusher.pb.utexas.edu).

The proposed methods for solving (3), (4) and (5) are based on the fact that these linear recurrence systems can be represented as lower triangular banded linear systems of equations. Such linear systems can be solved using modifications of the *divide-and-conquer* algorithms proposed in [2, 4, 6]. First, several linear systems of smaller size must be solved in parallel. Then, instead of the full solutions of systems (3), (4) and (5), we need to calculate only the last two terms.

Let p be the number of available parallel processors. For the parallel version of Goertzel algorithm, we have to solve p linear recurrence systems with constant coefficients for q equations (it is assumed that $q = n/p$ is an integer) in parallel, and compute the last two terms of the final solution using a *recursive doubling* scheme [1]. For the parallel Reinsch algorithm we have to solve $p + 2$ linear recurrence systems for q equations (where $2n = pq$) in parallel, and then find the last two terms of the solution of system (4) or (5) using a similar *recursive doubling* scheme. (We solve (4) or (5) depending on the sign of the function $\cos x$.)

3. Results. The experiments were performed on a 20-processor Sequent Symmetry shared-memory parallel computer. The code was implemented using Fortran 77 and Sequent-provided parallelization primitives (DOACROSS). The speedup of the parallel algorithms was calculated against the sequential algorithms. The following table shows an example of the computed speedup for $n = 6930$.

# proc.	2	3	5	6	7	9	10	11	14	15	18
Goertzel	0.88	1.31	2.13	2.54	2.93	3.53	3.79	4.17	4.89	5.34	5.75
Reinsch	1.01	1.51	2.47	2.92	3.37	4.30	4.70	5.05	6.26	6.48	7.38

Parallel Reinsch algorithm achieves better speedup than parallel Goertzel algorithm. This is typical example of Amdahl's Effect as Reisch algorithm solves $2n$ equations whereas Goertzel algorithm solves only n equations. For both parallel algorithms the speedup increases as the size of the problem (n) increases. Numerical tests show that the rounding errors of parallel algorithms are of the same order as the rounding errors of the corresponding sequential algorithms.

Acknowledgments. Computer time grant from the Department of Computer Science Engineering of the Southern Methodist University is kindly acknowledged.

REFERENCES

[1] J. Modi, *Parallel Algorithms and Matrix Computations*, Oxford University Press, Oxford, 1988.

[2] M. Paprzycki and P. Stpiczyński, *Solving linear recurrence systems on parallel computers*, Proceedings of the Mardi Gras '94 Conference, Baton Rouge, Nova Science Publishers, New York, 1994, to appear.

[3] J. Stoer and R. Bulirsch, *Introduciton to Numerical Analysis*, Springer Verlag, New York, 1980.

[4] P. Stpiczyński, *Parallel algorithms for solving linear recurrence systems*, in Lecture Notes in Computer Science 634, L. Bougé et al. eds., Springer Verlag, Berlin, 1992, pp. 343-348.

[5] P. Stpiczyński, *Error analysis of two parallel algorithms for solving linear recurrence systems*, Parallel Comput., 19 (1993), pp. 917-923.

[6] P. Stpiczyński, M. Paprzycki and R. Daniel, *Parallel algorithm for solving linear recurrence systems with constant coefficients*, submited for publication.