Numerical Algorithms 2 (1992) 133–136

Addendum to "Avoiding breakdown and near-breakdown in Lanczos type algorithms"

C. Brezinski [1], M. Redivo Zaglia [2] and H. Sadok [1]

[1] *Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, UFR IEEA, Bât M3, 59655 Villeneuve d'Ascq Cedex, France*

[2] *Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, via Gradenigo 6 / a, 35131 -Padova, Italy*

Received 24 October 1991

Some new numerical results for the near-breakdown free version of Lanczos method are reported.

Subject classifications: AMS (MOS) 65F10, 65F25

Keywords: Lanczos method

In [1] we gave an algorithm, called the BSMRZ, for avoiding near-breakdown in Lanczos type algorithms. Since the propagation of rounding errors influences very much the occurence of near-breakdowns and their treatment then, obviously, the programming of the algorithm also does.

The results given in [1] were obtained with a subroutine which was modified afterwards and since they no more exactly correspond to the software which can be found in netlib, the aim of this note is to report on the results obtained by this actual software. Additional results will be also presented. The data are the same as in [1].

The subroutine BSMRZ now gives the following results with $x_0 = 0$ and $y = (1, \ldots, 1)^T$. n_k is the dimension of the corresponding Krylov subspace (without rounding errors we should have $n_k \leq n$). Thus, the solution is now obtained in cases where it was not.

	$\varepsilon=10^{-8}$ $\varepsilon_1=10^{-14}$		$\varepsilon=10^{-1}$ $\varepsilon_1=10^{-11}$		$\varepsilon=1$ $\varepsilon_1=10^{-11}$	
n	$\Vert r_k \Vert$	n_k	$\Vert r_k \Vert$	n_k	$\Vert r_k \Vert$	n_k
4	$1.83\cdot 10^{-15}$	4	$1.83\cdot 10^{-15}$	4	$1.83\cdot 10^{-15}$	4
5	$1.65\cdot 10^{-13}$	5	$1.65\cdot 10^{-13}$	5	$1.65\cdot 10^{-13}$	5
6	$1.47\cdot 10^{-13}$	6	$1.47\cdot 10^{-13}$	6	$1.14\cdot 10^{-13}$	6
7	$8.34\cdot 10^{-14}$	11	$2.13\cdot 10^{-13}$	7	$3.45\cdot 10^{-13}$	7
8	$4.59\cdot 10^{-14}$	13	$2.09\cdot 10^{-12}$	8	$1.96\cdot 10^{-12}$	8
9	$1.72\cdot 10^{-14}$	15	$2.51\cdot 10^{-12}$	9	$2.23\cdot 10^{-12}$	9
10	$5.07\cdot 10^{-14}$	17	$2.02\cdot 10^{-12}$	10	$3.29\cdot 10^{-12}$	10
11	$3.85\cdot 10^{-14}$	21	$5.55\cdot 10^{-12}$	11	$3.86\cdot 10^{-12}$	11
12			$2.67\cdot 10^{-12}$	12	$1.68\cdot 10^{-12}$	12

With $x_0 = 0$ and $y = r_0$, we have

	$\varepsilon=10^{-8}$ $\varepsilon_1=10^{-14}$		$\varepsilon=10^{-1}$ $\varepsilon=10^{-11}$		$\varepsilon=1$ $\varepsilon_1=10^{-11}$	
n	$\Vert r_k \Vert$	n_k	$\Vert r_k \Vert$	n_k	$\Vert r_k \Vert$	n_k
5	$4.27\cdot 10^{-13}$	5	$2.56\cdot 10^{-13}$	5	$2.56\cdot 10^{-13}$	5
6	$1.09\cdot 10^{-10}$	6	$1.09\cdot 10^{-10}$	6	$2.22\cdot 10^{-13}$	6
7	$4.00\cdot 10^{-12}$	7	$2.08\cdot 10^{-12}$	7	$2.08\cdot 10^{-12}$	7
8	$3.39\cdot 10^{-12}$	13	$3.66\cdot 10^{-13}$	13	$1.21\cdot 10^{-12}$	8
9	$2.77\cdot 10^{-12}$	15			$1.87\cdot 10^{-12}$	9
10	$4.72\cdot 10^{-12}$	17			$1.74\cdot 10^{-12}$	10
11	$4.63\cdot 10^{-12}$	19			$5.88\cdot 10^{-12}$	11
12	$2.11\cdot 10^{-12}$	21	$2.07\cdot 10^{-10}$	12	$3.73\cdot 10^{-12}$	12

To show the improvement brought by our algorithm let us give the number k of iterations needed and the dimension n_k of the corresponding Krylov subspace to obtain a residual vector with a norm smaller than 10^{-11} when using the BSMRZ with $\varepsilon = 10^{-8}$ and $\varepsilon_1 = 10^{-14}$ and with $\varepsilon = \varepsilon_1 = 10^{-30}$ (in that case we never pass through the rules for near-breakdown). We have

	$\varepsilon=10^{-8}$ $\varepsilon_1=10^{-14}$ $y=(1,\ldots,1)^T$		$\varepsilon=10^{-30}$ $\varepsilon_1=10^{-30}$ $y=(1,\ldots,1)^T$		$\varepsilon=10^{-8}$ $\varepsilon_1=10^{-14}$ $y=r_0$		$\varepsilon=10^{-30}$ $\varepsilon_1=10^{-30}$ $y=r_0$	
n	k	n_k	k	n_k	k	n_k	k	n_k
7	11	11	11	11	7	7	7	7
8	12	13	13	13	13	13	13	13
9	13	15	15	15	14	15	15	15
10	14	17	17	17	15	17	17	17
11	15	21	19	19	16	19	19	19
12			21	21	17	21	21	21

Now, when the system has dimension $n = 50$ and for $y = r_0$, $\varepsilon_1 = 10^{-15}$, the norms of the residual vector when $n_k = 50$ are the following on a VAX computer

ε	k	$n.j.$	$\Vert r_k \Vert$	m	n_m
1	12	2	$1.22 \cdot 10^4$	57	95
10^{-1}	7	2	$5.46 \cdot 10^2$	54	97
10^{-2}	15	2	$4.78 \cdot 10^5$	57	92
10^{-3}	11	2	$3.17 \cdot 10^4$	54	93
10^{-4}	9	2	$1.33 \cdot 10^4$	55	96
10^{-5}	8	1	$4.92 \cdot 10^2$	55	97
10^{-6}	8	1	$4.92 \cdot 10^2$	55	97
10^{-7}	8	1	$4.92 \cdot 10^2$	55	97
10^{-8}	8	1	$4.92 \cdot 10^2$	55	97
10^{-9}	8	1	$4.92 \cdot 10^2$	55	97
10^{-10}	8	1	$4.92 \cdot 10^2$	57	99
10^{-11}	30	1	$2.81 \cdot 10^5$	92	112
10^{-12}	30	2	$5.68 \cdot 10^5$		
10^{-13}	47	2	$2.39 \cdot 10^6$		
10^{-14}	49	1	$6.16 \cdot 10^7$		
10^{-15}	50	0	$1.87 \cdot 10^8$		

$n.j.$ represents the number of jumps, m the iteration such that $\Vert r_m \Vert \le \varepsilon$ and n_m the dimension of the corresponding Krylov subspace (with an upper bound of 150).

In the numerical results given in [1] it can be seen that, sometimes, the MRZ (which is only valid when an exact breakdown occurs) gives better values that the BSMRZ (which works for breakdowns and near-breakdowns). The reason seems to be that, due to the computer's arithmetic, the occurence of a breakdown is never detected by a quantity strictly equal to zero but only by a quantity close to it. Using the MRZ, instead of the BSMRZ, when there is an exact breakdown only detected by a quantity close to zero, corrects, in fact, the computer's arithmetic by imposing this quantity (and possibly some others) to be exactly zero and, thus, the results are improved.

In fact the programming of the BSMRZ needs the introduction of four independent ε's

1 – an ε for testing if $|c^{(1)}(\xi^i P_k^{(1)})| \le \varepsilon$

2 – an ε for testing the pivots in Gaussian elimination

3 – an ε for testing if $\Vert r_k \Vert \le \varepsilon$

4 – an ε for testing if $|c(\xi^{n_k} P_k)| \le \varepsilon$ because the algorithm cannot be used if this quantity is zero.

We are now conducting many more numerical experiments for trying to fully understand the behaviour of the algorithm.

The subroutines corresponding to this work can be found in the NETLIB library NUMERALGO. They can be obtained by the command

send na1 from numeralgo

With this command, all the subroutines are automatically obtained (in order to avoid name conflicts, a single subroutine can never be obtained alone).

Netlib can be reached by electronic mail using the Internet address

netlib@research.att.com

The list of the software contained in NUMERALGO can be obtained by the command

send index from numeralgo

Reference

[1] C. Brezinski, M. Redivo Zaglia and H. Sadok, Avoiding breakdown and near-breakdown in Lanczos type algorithms, Numerical Algorithms 1 (1991) 261–284.