Journal of Research of the National Bureau of Standards Vol. 48, No. 6, June 1952 Research Paper 2330

Gradient Methods in the Solution of Systems of Linear Equations [1]

Marvin L. Stein

The results of various experiments with iterative methods for solving systems of linear algebraic equations are discussed. Modifications of the optimum gradient method are compared, and the rather interesting self-acceleration properties of a class of methods here named "almost optimum" gradient methods are pointed out.

1. Introduction

The method of steepest descent, or the optimum gradient method, has been known to mathematicians since the time of Cauchy [1].[2] Others who have discussed this method include Curry [2], Forsythe and Motzkin [3], Householder [4], Kantorovitch [5], and Temple [6]. Its infrequent application in computational work is no doubt due to the slowness with which it converges. This slowness of convergence is unfortunately generally true of gradient methods. However, with the advent of large-scale computing machinery it has become feasible to seriously consider them in practical numerical analysis.

In a forthcoming paper, Hestenes and Stein [7] discuss a large class of gradient procedures for solving systems of linear equations. These procedures contain the optimum gradient method as a special case. The present note is mainly a report on some numerical experiments with them that were carried out on the IBM Card-Programmed Electronic Calculator at the Institute for Numerical Analysis of the National Bureau of Standards. Some attention is also given to an experiment in which the problem of solving a system of linear equations was changed to an equivalent eigenvalue problem and then solved by a modification of one of the gradient methods discussed by Hestenes and Karush [8]. The most striking result of the experiments indicates that there is a large class of gradient methods that is *self-accelerating*, that is, which irregularly shows a large increase in the rate of convergence without the introduction of any modification in the computational routine. This behavior is in sharp contrast to that of the method of steepest descent, which in the light of the present results can no longer be considered as optimum from an over-all point of view unless modified by some special accelerating routine [9].

2. Summary of the Theory

Let A denote an m-rowed and n-columned matrix, and let b and x be m-rowed and n-rowed column vectors, respectively. The vector $r(x)=b-Ax$ is then an m-rowed column vector. In the following a star (*) affixed to the symbol for any vector or matrix will indicate its conjugate transpose. Hence, if H is a positive $m\times m$ Hermitian matrix, we have $H^*=H$, $r^*Hr>0$, provided $r\neq 0$. Clearly, the nonnegative real-valued function

$$f(x)=r^*Hr \tag{1}$$

vanishes if and only if

$$Ax=b. \tag{2}$$

Therefore, the problem of minimizing (1) is equivalent to the problem of solving the system (2), provided a solution exists. If no solution exists, a vector x which minimizes $f(x)$ yields a best fit of b by Ax in a least-squares sense with the metric determined by H.

Hestenes and Stein [7] have analyzed the following algorithm for constructing the minimum of (1). Consider iterations of the type

$$x_{i+1}=x_i+\alpha_i\xi_i \quad (i=0,1,2, \ldots), \tag{3}$$

where x_0 is chosen initially and where, after x_i has been determined, the gradient vector ξ_i is defined by the rule $\xi_i=A^*Hr(x_i)$. If $\xi_i=0$, the problem is solved. If $\xi_i\neq 0$, the scalar α_i is taken to be of the form $\alpha_i=\beta_i\gamma_i$, where

$$\gamma_i=\frac{\xi_i^*\xi_i}{\xi_i^*A^*HA\xi_i}$$

and β_i is any complex number. The sequence (3) has been shown to converge to the minimum of $f(x)$ provided the coefficients β_i satisfy the conditions

$$\frac{1}{\beta_i}+\frac{1}{\bar{\beta}_i}\geq 1+\delta, \qquad |\beta_i|\geq\delta,$$

where δ is arbitrary on the range $0<\delta<1$.

Setting $\beta_i\equiv 1$ gives the optimum gradient method, which has the following geometrical interpretation. Starting at x_0, one proceeds along the normal to that member of the family of concentric ellipsoids

$$f(x)=\text{constant}, \tag{4}$$

which passes through x_0 until a point tangent to another ellipsoid of the family is reached. One then goes along the normal to this second ellipsoid until one is again at a point of tangency to a member of (4),

---

[1] The preparation of this paper was sponsored (in part) by the Office of Naval Research.

[2] Figures in brackets indicate the literature references at the end of this paper.

and so on until the common center is reached. Clearly, it would be highly desirable to land on an axis of the family. However, as simple two-dimensional examples illustrate, one usually overshoots the major axis by proceeding all the way to a point of tangency. Hence, it was conjectured by M. R. Hestenes that of the allowed values of β_i some of those chosen from the range $\beta_i<1$ would yield better convergence than $\beta_i\equiv 1$. This conjecture is the motivation for the experimental work whose results will be tabulated and discussed below.

3. Experimental Data

The algorithm described in the preceding section was carried out for the problem $Ax=b$ with

$$A=\begin{pmatrix} .06667 & .02634 & -.04640 & -.07368 & -.02131 & -.00431 \\ .02634 & .26841 & -.02243 & .15952 & -.05923 & -.12797 \\ -.04640 & -.02243 & .10932 & .05150 & -.04100 & .08558 \\ -.07368 & .15952 & .05150 & .25152 & -.01141 & -.07169 \\ -.02131 & -.05923 & -.04100 & -.01141 & .14403 & .01105 \\ -.00431 & -.12797 & .08558 & -.07169 & .01105 & .19450 \end{pmatrix} \tag{5}$$

and

$$b=\begin{pmatrix} -.008609 \\ -.014279 \\ -.000243 \\ .004576 \\ .008043 \\ -.004895 \end{pmatrix}. \tag{6}$$

Runs were made with fixed values of β_i ranging between $\beta_i\equiv .1$ and $\beta_i\equiv 1.9$. Since the matrix A and the vector b were obtained from an original matrix B and vector c as

$$A=B^*B, \qquad b=B^*c, \tag{7}$$

A is positive and symmetric. Hence, a convenient metric is

$$H=A^{-1}. \tag{8}$$

In the metric (8) the gradient becomes $\xi_i=r(x_i)$. Thus the other significant quantities defined in section 2 assume the form

$$\gamma_i=\frac{r^*(x_i)\,r(x_i)}{r^*(x_i)\,Ar(x_i)}$$

$$f(x_i)=c^*c-(x_i^*b+b^*x_i)+x_i^*Ax_i$$
$$=c^*c-x_i^*(b+r(x_i)).$$

Table 1 lists runs of $f(x_i)$ and $f(x_i)/f(x_{i-1})$ for various fixed values of β_i. In each case the initial $x=x_0$ was zero. The values of $f(x_i)$ are given here to the same number of places as in the original calculation, while the ratios $f(x_i)/f(x_{i-1})$ have been cut down from a six-place table. Table 2 gives the values of α_i corresponding to the runs listed in table 1. These numbers were originally computed to 10 places. The results listed in tables 1 and 2 are based for the most part on single runs of the Card-Programmed Calculator. The main check used in the machine computations was the monotonicity of $f(x)$. However, the run $\beta_i\equiv .9$ has been exactly duplicated on two separate occasions The results of the run $\beta_i\equiv 1$ check closely with results obtained by A. I. Forsythe in an entirely independent manner.

Table 1

i \ β	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$
	0.1		0.3		0.6		0.8	
0	333840	------	333840	------	333840	------	333840	------
1	293093	0. 8779	224451	0. 6723	153663	0. 4603	127922	0. 3832
2	257410	. 8782	152582	. 6798	98661	. 6421	93810	. 7333
3	226497	. 8799	111480	. 7306	77941	. 7900	73423	. 7827
4	199981	. 8829	87335	. 7834	67416	. 8650	65477	. 8918
5	177424	. 8872	72579	. 8310	61492	. 9121	58631	. 8954
6	158324	. 8923	63339	. 8727	57677	. 9379	55826	. 9521
7	142190	. 8981	57304	. 9047	55073	. 9548	51049	. 9144
8	128561	. 9041	52467	. 9156	52979	. 9620	49055	. 9609
9	117033	. 9103	44607	. 8502	51324	. 9688	23614	. 4814
10	107265	. 9165	39040	. 8752	49694	. 9682	22633	. 9584
11	98962	. 9226	38532	. 9870	48416	. 9743	19411	. 8576
12	91885	. 9285	36121	. 9374	46747	. 9655	18091	. 9320
13	85836	. 9342	35705	. 9885	45718	. 9780	16729	. 9247
14	80647	. 9395	34881	. 9769	41494	. 9076	16068	. 9605
15	76184	. 9447	34138	. 9787	40042	. 9650	14243	. 8864
16	72336	. 9495	33281	. 9749	39063	. 9755	13389	. 9400
17	68999	. 9539	32693	. 9823	38192	. 9777	13008	. 9715
18	66094	. 9579	31085	. 9508	37193	. 9738	12727	. 9784
19	63546	. 9614	30737	. 9888	36444	. 9799	12414	. 9754
20	61295	. 9646	29969	. 9750	35033	. 9613	12152	. 9789
21	59273	. 9670	29460	. 9830	34325	. 9798	11807	. 9716
22	57423	. 9688	27849	. 9453	18343	. 5344	11544	. 9777
23	55677	. 9696	27540	. 9889	16437	. 8961	10972	. 9504
24	53951	. 9690	27027	. 9814	16011	. 9741	10595	. 9656
25	52120	. 9661	26105	. 9659	15288	. 9548	3694	. 3486
26	49948	. 9583	25789	. 9879	14955	. 9782	2468	. 6681
27	46891	. 9388	21529	. 8348	13055	. 8729	2389	. 9680
28	42130	. 8985	21171	. 9834	12441	. 9530	1956	. 8187
29	36978	. 8777	20929	. 9886	12216	. 9819	1767	. 9033
30	31700	. 8573	20649	. 9866	11111	. 9095	1723	. 9751

i	0. 85		0.9		0.95		1.0	
0	333840	------	333840	------	333840	------	333840	------
1	124168	0. 3719	121487	0. 3639	119877	0. 3591	119341	0. 3575
2	91763	. 7390	89569	. 7373	87412	. 7292	85444	. 7160
3	71835	. 7828	70668	. 7890	70104	. 8020	70047	. 8198
4	64192	. 8936	63092	. 8928	62460	. 8910	62360	. 8902
5	57521	. 8961	57047	. 9042	57261	. 9168	57853	. 9277
6	54774	. 9522	54256	. 9511	54401	. 9500	54959	. 9500
7	49772	. 9087	50251	. 9262	51556	. 9477	52692	. 9587
8	47378	. 9519	47793	. 9511	49435	. 9589	50807	. 9642
9	27275	. 5757	38645	. 8086	46729	. 9453	49095	. 9663
10	25034	. 9178	33695	. 8719	44495	. 9522	47519	. 9679
11	23465	. 9373	31606	. 9380	40884	. 9188	46036	. 9688
12	22561	. 9615	30107	. 9526	38017	. 9299	44634	. 9695
13	21653	. 9597	27239	. 9047	28607	. 7525	43304	. 9702
14	21039	. 9716	25370	. 9314	22911	. 8009	42036	. 9707
15	20108	. 9557	24619	. 9704	20890	. 9118	40825	. 9712
16	19456	. 9676	24072	. 9778	19416	. 9294	39667	. 9716
17	16602	. 8533	23518	. 9770	18906	. 9737	38557	. 9720
18	14946	. 9002	23014	. 9786	18497	. 9784	37489	. 9723
19	14384	. 9624	22453	. 9756	18089	. 9779	36462	. 9726
20	13963	. 9707	21954	. 9778	17701	. 9785	35473	. 9729

Table 1.—Continued

i \ β	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$	$f(x_i)$	$\frac{f(x_i)}{f(x_{i-1})}$
	0.85		0.9		0.95		1.0	
21	12095	.8662	21356	.9728	17306	.9777	34518	.9731
22	10917	.9026	20842	.9759	16935	.9786.	33597	.9733
23	10509	.9626	20115	.9651	16551	.9773	32706	.9735
24	10201	.9707	19515	.9702	16190	.9782	31843	.9736
25	8748	.8576	18227	.9340	15818	.9770	31008	.9738
26	7837	.8959	17230	.9453	15469	.9779	30197	.9738
27	7570	.9659	598	.0347	15103	.9763	29413	.9740
28	7366	.9730	280	.4682	14761	.9773	28648	.9740
29	6762	.9180	273	.9750	14396	.9753	27910	.9742
30	6352	.9394	264	.9670	14059	.9766	27191	.9742
	1.1		1.3		1.6		1.9	
0	333840	----------	333840	----------	333840	----------	333840	----------
1	121486	.3639	138645	.4153	196562	.5888	293091	.8779
2	82591	.6798	84466	.6092	124599	.6339	256477	.8751
3	70272	.8508	69395	.8216	85428	.6856	219046	.8540
4	62921	.8954	62726	.9039	68391	.8006	188063	.8585
5	58869	.9356	58755	.9367	60665	.8870	162609	.8646
6	55940	.9502	56019	.9534	56682	.9343	141685	.8713
7	53827	.9622	53906	.9623	54199	.9562	124461	.8784
8	52009	.9662	52138	.9672	52332	.9655	110243	.8858
9	50457	.9701	50571	.9699	50740	.9696	98480	.8933
10	49000	.9711	49131	.9715	49290	.9714	88712	.9008
11	47669	.9728	47784	.9726	47936	.9725	80574	.9083
12	46391	.9732	46512	.9734	46660	.9734	73768	.9155
13	45190	.9741	45301	.9740	45440	.9738	68046	.9224
14	44031	.9743	44144	.9745	44277	.9744	63212	.9290
15	42929	.9750	43035	.9749	43165	.9749	59100	.9349
16	41866	.9752	41970	.9752	42097	.9752	55587	.9405
17	40851	.9757	40948	.9756	41068	.9755	52560	.9455
18	39865	.9759	39961	.9759	40077	.9759	49934	.9500
19	38917	.9762	39009	.9762	39123	.9762	47640	.9540
20	38001	.9765	38091	.9765	38200	.9764	45618	.9575
21	37114	.9766	37203	.9767	37307	.9766	43820	.9606
22	36257	.9769	36341	.9768	36444	.9769	42212	.9633
23	35426	.9771	35506	.9770	35607	.9770	40757	.9655
24	34619	.9772	34700	.9773	34795	.9772	39433	.9675
25	33835	.9773	33912	.9773	34007	.9773	38219	.9692
26	33073	.9775	33149	.9775	33239	.9774	37097	.9706
27	32333	.9776	32407	.9776	32496	.9776	36053	.9718
28	31613	.9777	31684	.9777	31770	.9776	35075	.9729
29	30913	.9778	30982	.9778	31066	.9778	34155	.9738
30	30230	.9779	30298	.9779	30380	.9779	33283	.9744

TABLE 2.

i \ β	0.1	0.3	0.6	0.8	0.85	0.9	0.95	1.0	1.1	1.3	1.6	1.9
1	0.553	1.660	3.320	4.427	4.703	4.980	5.257	5.533	6.087	7.193	8.853	10.513
2	.595	2.094	3.576	2.827	2.691	2.602	2.558	2.551	2.625	2.983	3.751	4.613
3	.635	2.377	3.491	4.642	5.158	5.543	5.657	5.442	4.437	3.242	3.335	3.844
4	.670	2.764	3.585	2.676	2.532	2.486	2.525	2.641	3.057	3.514	3.335	3.827
5	.702	3.319	3.587	6.157	7.173	7.484	6.865	5.781	4.217	3.635	3.398	3.826
6	.732	4.064	3.766	2.391	2.315	2.362	2.508	2.756	3.369	3.740	3.522	3.827
7	.762	5.168	3.714	12.626	16.049	13.847	9.251	6.267	4.197	3.822	3.684	3.829
8	.792	7.863	4.012	2.027	2.030	2.141	2.412	2.829	3.620	3.883	3.824	3.832
9	.825	22.205	3.688	125.896	108.543	50.509	12.958	6.466	4.166	3.921	3.903	3.834
10	.862	21.294	4.390	2.248	3.025	1.886	2.248	2.849	3.766	3.944	3.939	3.837
11	.902	1.519	3.336	25.889	4.738	17.968	22.431	6.513	4.112	3.956	3.955	3.841
12	.947	11.598	5.688	1.825	2.912	2.128	2.085	2.853	3.847	3.963	3.963	3.845
13	.996	1.198	2.492	15.493	6.140	29.005	79.571	6.530	4.066	3.968	3.967	3.849
14	1.052	4.095	19.081	2.087	2.634	2.525	1.958	2.856	3.895	3.971	3.970	3.855
15	1.113	3.761	1.447	27.349	9.972	5.586	27.906	6.542	4.036	3.973	3.973	3.860
16	1.182	4.595	4.218	3.301	2.206	3.407	2.685	2.858	3.926	3.975	3.975	3.866
17	1.260	2.977	3.611	4.480	41.576	5.009	4.786	6.550	4.016	3.977	3.976	3.873
18	1.349	9.809	4.832	3.474	1.799	3.207	3.565	2.859	3.947	3.978	3.978	3.880
19	1.454	1.233	2.990	4.986	9.919	5.628	4.596	6.557	4.005	3.979	3.979	3.887
20	1.581	4.668	8.239	3.103	2.218	2.954	3.488	2.860	3.961	3.981	3.980	3.895
21	1.742	2.893	1.919	6.463	41.303	6.818	4.747	6.563	3.998	3.982	3.981	3.903
22	1.958	11.156	116.743	2.567	1.799	2.657	3.398	2.861	3.970	3.983	3.982	3.911
23	2.264	1.131	1.259	13.274	10.109	9.627	4.942	6.568	3.994	3.984	3.983	3.918
24	2.735	3.300	1.595	1.962	2.207	2.332	3.295	2.862	3.976	3.984	3.984	3.926
25	3.534	6.757	9.841	198.581	44.570	20.471	5.200	6.571	3.992	3.985	3.985	3.933
26	5.083	1.714	1.759	1.628	1.793	2.015	3.179	2.863	3.981	3.986	3.986	3.940
27	8.606	35.356	30.924	2.204	8.887	326.505	5.550	6.574	3.991	3.986	3.986	3.946
28	15.945	0.755	1.344	42.957	2.314	1.817	3.049	2.863	3.984	3.987	3.987	3.952
29	20.844	0.993	2.467	1.695	24.873	3.018	6.038	6.576	3.990	3.988	3.988	3.957
30	25.971	2.110	22.538	5.189	1.884	6.291	2.907	2.864	3.986	3.988	3.988	3.962

4. Conversion to an Eigenvalue Problem

By introducing the variable x_{n+1}, an equation $Bx=c$ can be expressed in the homogeneous form $Cy=0$, where

$$C=(B \quad c), \qquad y=\begin{pmatrix} x \\ x_{n+1} \end{pmatrix}.$$

Multiplying through by C^* gives the system of order one higher $Dy=0$, where

$$D=\begin{pmatrix} B^*B & B^*c \\ c^*B & c^*c \end{pmatrix}$$

is a symmetric matrix whose least eigenvalue is zero. The nontrivial eigenvector corresponding to this zero eigenvalue yields a solution of the original system of linear equations.

As is well known, the least eigenvector of the symmetric matrix D can be found by minimizing the Rayleigh quotient

$$\mu(y)=\frac{y^*Dy}{y^*y}, \qquad y\neq 0.$$

Hestenes and Karush [8] have examined in detail the convergence of various methods for accomplishing the minimization of μ. The following algorithm for constructing a sequence $\{y_i\}$ that minimizes μ is a modification of their "optimum α" procedure. We define the gradient as $\xi_i\equiv\xi(y_i)=Dy_i-\mu(y_i)y_i$. Let

$$\gamma_i=\frac{1}{\mu(\xi_i)}, \qquad \mu(\xi_i)\neq 0,$$

and let $\alpha_i=\beta\gamma_i$, where β is a positive number less than or equal to one. Then, given y_i, we determine y_{i+1} by the formula $y_{i+1}=y_i-\alpha_i\xi_i$.

This algorithm was actually carried out with the matrix

$$D=\begin{pmatrix} A & b \\ b^* & c^*c \end{pmatrix},$$

where A is given by (5), b is given by (6) with all components multiplied by 10 and $c^*c=.333840$. In view of (7) $Dy=0$ is actually a problem of the type just described. Runs were made with various values of β ranging between .7 and 1. For purposes of comparison, the same starting point (origin) as in the computations recorded in section 3 was used and a record of

$$f(x)=\frac{y^*Dy}{|y_{n+1}|^2}$$

was kept. In table 3[3] we present the results of the run $\beta=.9$. The runs for other values of β showed substantially the same unstable behavior, while that for $\beta=1$ (optimum) showed a very stable behavior and as a result converged quite slowly. The numbers $f(x_i)$ and α_i, appearing in table 3, were originally computed to 10 places. The ratios $f(x_i)/f(x_{i-1})$ were computed from the original values of $f(x_i)$ and then cut down to the present size.

TABLE 3.

i	$f(x_i)$	$f(x_i)/f(x_{i-1})$	$0.9\gamma_i$
0	333840	------	------
1	121482	0. 3639	0. 4980
2	86700	. 7137	. 2575
3	70323	. 8111	. 7076
4	57533	. 8181	. 3951
5	60698	1. 0550	. 5478
6	53564	0. 8825	. 3968
7	53869	1. 0057	. 5494
8	49788	0. 9242	. 3479
9	46902	. 9420	. 5237
10	45704	. 9744	. 3046
11	41266	. 9029	. 5577
12	41045	. 9947	. 2761
13	36337	. 8853	. 6639
14	36011	. 9910	. 2482
15	31318	. 8697	. 9109
16	30291	. 9672	. 2189
17	24919	. 8226	1. 7145
18	22095	. 8867	0. 1909
19	7468	. 3380	5. 9172
20	6963	. 9324	0. 2758
21	5985	. 8595	. 6081
22	5541	. 9258	. 2493
23	4642	. 8378	. 8282
24	4206	. 9060	. 2201
25	3171	. 7540	1. 5481
26	2644	. 8339	0. 1912
27	264	. 0998	6. 6970
28	151	. 5733	0. 1766
29	117	. 7708	1. 4380
30	91	. 7806	0. 1908
31	11	. 1274	6. 4755
32	8	. 6933	0. 1890

5. Conclusion

The error function $f(x)$ goes monotonically to zero in each of the gradient methods. Hence the number $P_n=100f(x_n)/f(x_0)$ tells us what percentage of the distance from the starting point to zero remains to be covered at the nth step. Table 4 gives P_{30} for various values of β_i. We note

TABLE 4.

β	0. 1	0. 3	0. 6	0. 8	0. 85	0. 9
P_{30}	9. 49	6. 18	3. 33	0. 52	1. 90	0. 08
β	0. 95	1. 0	1. 1	1. 3	1. 6	1. 9
P_{30}	4. 21	8. 14	9. 05	9. 07	9. 10	9. 97

after 30 steps that, with the exception of the case $\beta_i\equiv\beta=.1$, all the gradient methods for values $\beta<1$ are converging faster than the optimum method, while all the gradient methods for values $\beta>1$ are converging at a slightly slower rate than the optimum method. For the eigenvalue method we have $P_{32}=.002$. The increased rate of convergence for this case is offset on the Card-Programmed Calculator by the greater length of time needed for each step. However, on a high-speed machine this factor would be negligible. An explanation of the speedier convergence of the eigenvalue procedure lies in the fact that the transformation of the problem to the homogeneous form has shrunk the ratio of the largest and the smallest nonzero eigenvalues. This improvement of "condition" is something that one could not generally expect to occur [10].

The ratios $f(x_i)/f(x_{i-1})$ compare the rate of convergence at each step with that of a geometric progression having the same ratio. A study of these numbers and of the corrections α_i brings into sharp focus the contrast between the instability of the gradient methods employing $\beta<1$ and the stability of those employing $\beta\geq 1$. In the method of steepest descent it is just this stability that permits acceleration. However, this acceleration must be achieved through a modification of the computational routine. On the other hand, the instability of the methods using $\beta<1$ leads to occasional accelerations without the introduction of any changes whatsoever in the computing routine.

From the point of view of using the "almost optimum" gradient method on a large scale computer, its self-acceleration property has more than theoretical interest. As is well known, the high-speed memory capacity of the computers now in existence is rather limited. Hence, the necessity of storing a special acceleration routine might prove to be a severe handicap indeed.

It is worth while to compare the values of α_i with the reciprocal eigenvalues of the matrices A and D. For A these reciprocal eigenvalues have been found to be approximately 2.0, 3.9, 5.7, 12.1, 63, and 372, while for D the finite reciprocal eigenvalues range between approximately 8.35 and 0.189.[4] We will pay particular attention to the points at which acceleration took place. One sees that preceding an acceleration there was a "smoothing run" during which the α_i's were in the range of the small reciprocal eigenvalues. On the iteration immediately before an acceleration, α_i was almost equal to the smallest reciprocal eigenvalue, while on the iteration

[3] The author thanks R. M. Hayes for furnishing him with most of the data appearing in table 3.

[4] These values were furnished by R. M. Hayes.

during which acceleration took place α_i was between the highest and next to highest reciprocal eigenvalues. It was just this technique of choosing α_i that the present author helped develop in previous experiments with a "fixed α" gradient method, which were conducted under the direction of M. R. Hestenes. In this method the operator chooses the value of α just before each iteration, and by judicious choices he can successfully accelerate the method to a considerable extent. However, this requires too many judgments of the operator to be practical for a fast machine or an inexperienced operator. Hence, it is quite hopeful to note the existence of methods suitable for high-speed machines that can duplicate the fixed α acceleration procedures without any intervention by the operator once the process has started.

6. References

[1] A. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, Compt. rend. L'Académie Sciences **25**, 536 (1847).

[2] H. B. Curry, The method of steepest descent for nonlinear minimization problems, Quart. Applied Math. **2**, 258 (1944).

[3] G. E. Forsythe and T. S. Motzkin, Asymptotic properties of the optimum gradient method, Bul. Am. Math. Soc. **57**, 183 (1951).

[4] A. S. Householder, Notes on numerical methods (1949–51) (typescript available at Oak Ridge National Laboratory).

[5] L. V. Kantorovitch, On an effective method of solving extremal problems for quadratic functionals, Compt. rend. (Doklady) de l'Académie des Sciences de l'URSS **48**, 455 (1945).

[6] G. Temple, The general theory of relaxation methods applied to linear systems, Proc. Roy. Soc. (London) **169** [A] 476 (1938–39).

[7] M. R. Hestenes and M. L. Stein, The solution of linear equations by minimization, in manuscript at the National Bureau of Standards, Los Angeles, Calif.

[8] M. R. Hestenes and W. Karush, A method of gradients for the calculation of the characteristic roots and vectors of a real symmetric matrix, J. Research NBS **47**, 45 (1951) RP2227.

[9] The problem of accelerating the optimum gradient method has recently been considered by G. E. Forsythe and T. S. Motzkin in a paper entitled, Acceleration of the optimum gradient method, presented to the American Mathematical Society (Stanford meeting) (April 28, 1951).

[10] For a discussion of the effect of this type of transformation on the condition of a matrix, see G. E. Forsythe and T. S. Motzkin, An extension of Gauss's transformation for improving the condition of systems of linear equations, typescript available at National Bureau of Standards, Los Angeles, Calif.

Los Angeles, August 13, 1951.