HAINES & CAMPBELL 80-190

UNITED STATES DEPARTMENT OF INTERIOR

GEOLOGICAL SURVEY

Texas Instruments Model 59 Hand-Calculator Program to Calculate Theoretical Wenner and Schlumberger Vertical Electric Soundings over a Structure of up to 10 Horizontal Layers

by

Donald N. Haines and David L. Campbell

Open-File Report 80-190

1980

TITLE VES: Wenner/Schlumberger PAGE 1 OF 15

TI Programmable Program Record

PROGRAMMER Donald N. Haines and David L. Campbell DATE 8/8/78

Partitioning (Op 17) 479.59 Library Module None used Printer Optional Cards 1-Wenner 1-Schl.

PROGRAM DESCRIPTION

Given an input resistivity structure of $\leq$10 horizontal "layers" ($\leq$9 true layers plus underlying half-space), this program calculates the theoretical Wenner or Schlumberger vertical electric sounding, "VES", which would be seen over the layered structure. Output consists of pairs [a, ρ_{app}(a)], where a = array spacing (= AB/3 for Wenner, = AB/2 for Schlumberger; AB is the distance between outer, usually current, electrodes), and ρ_{app}(a) = apparent resistivity at that spacing, sampled at three points per decade of a. Separate program cards are provided to handle Wenner and Schlumberger arrays.

CARD LAYOUT

1 < > 2

Wenner Vert. Electric Sounding

		a_{max}		
Initialize	ρ_i, h_i	a_{min}	VES	Manual

1 < > 2

Schlumberger Vert. Electric Sounding

		a_{max}		
Initialize	ρ_i, h_i	a_{min}	VES	Manual

USER DEFINED KEYS	DATA REGISTERS (INV)		LABELS (Op 08)
A INITIALIZE	0 Index reg	19-28 Kernal B_i	INV _ lnx _ CE _ CLR _ x:t _ x² _
B INPUT ρ, h	1 " "	29 n = # layers	√x _ 1/x _ STO ✓ RCL _ SUM ✓ yˣ _
C Input a_{min}	2 Loop "	30 X_{min}	
D VES →	3 Counter	31 factor f	
E Manual →	4 Counter	32 Scratch	
A' —	5 Convolution Sum	33-52 (ρ, h) pairs	
B' —	6 Index reg.	53 a_{max}	
C' Input a_{max}	7 Loop reg.	54 factor a/x	
D' —	8 Index reg.	55 fug factor	
E' —	9-18: Ghosh coeffs.	→	

FLAGS	0	1	2	3	4	5	6	7	8	9
	✓	✓	✓		✓					

1014966-1

TECHNIQUE: This program was written following an HP calculator program by Campbell and Watts (1978) and FORTRAN programs by Zohdy (1973, 1974a) and Zohdy and Bisdorf (1975). The technique is described by Zohdy (1974b, 1975), and involves two steps for each electrode spacing value, "a". First, the input layered structure is used to calculate a digital kernel function B(x), where x is related to the present spacing a. Second, B(x) is convolved with a set of coefficients (Ghosh, 1971), which are specific to the array in use, to find $\rho_{app}(a)$. Iteration is accomplished by incrementing a, hence x, by 1/3 of a log cycle; B(that x) becomes $B(x-\Delta x)$; all other B(x) are cascaded in memory so that $B(x-j\Delta x)$ occupies $B(x-(j-1)\Delta x)$ slot; convolution is again performed; and so on.

EQUATIONS: $\Delta x = 10^{1/3}$

Wenner	$f = 1.36$	$x = a/10^{8/3}$
Schlumberger	$f = 1.05$	$x = a/100$

1. To get B(x) = kernel function

   a. Initialize for bottom layer (half-space)

   $B_i = 1$ $\qquad R_i = \rho_n$

   b. Iterate upward through all layers i

   $R_i = B_i \cdot R_i$

   $K_i = (\rho_{i-1} - R_i)/(\rho_{i-1} + R_i)$

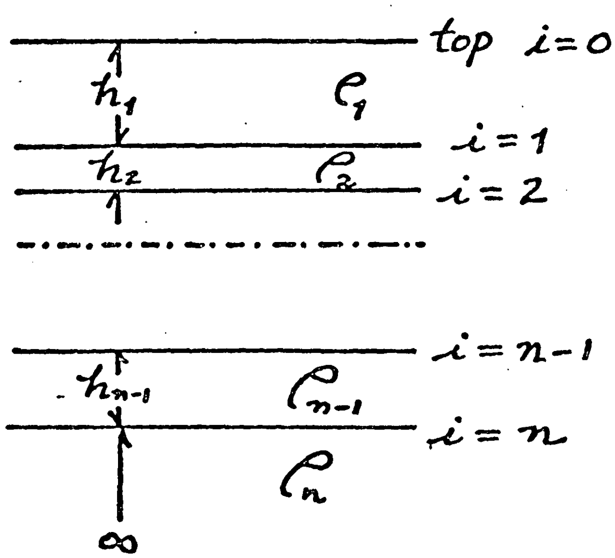
   $m_i = (-2h_{i-1})/(f \cdot x)$

   $Q_i = K_i \cdot \exp(m_i)$

   $B_{i-1} = (1-Q_i)/(1+Q_i)$

   c. After iterating through all layers

   $B(x) = B_1 \cdot \rho_1$



c. Convolve with Ghosh coefficients, G_i

$$\rho_{app}(a) = \sum_{j=0}^{9} G_j \cdot B(x - j\Delta x)$$

OPERATING LIMITS AND WARNINGS

1. The Ghosh filter has been found to have too few coefficients to adequately track a rapidly falling curve (Zohdy, 1975, p. 37; also see Anderson, 1979). Thus, curves produced by this program may be somewhat in error in cases where a very good conductor underlies a relatively resistant layer. As a rule of thumb, be wary of this program's results whenever a conducting layer has resistivity less than 1/20 that of the overlying material.

2. If an attempt is made at step 3 to enter more than 10 layers, the program will stop with a flashing "11" in display. (Press "CLR" to clear the flashing.) All further calculations will assume the 10th layer represents the bottom half-space, and will ignore the deeper, invalid layer(s).

REFERENCES

Anderson, W. L., 1979, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, v. 44, p. 1287-1305.

Campbell, D. L., and Watts, R. D., 1978, Exploration geophysics calculator programs for use on Hewlett-Packard models 67 and 97 programmable calculators: U.S. Geological Survey Open-File Report 78-815, p. 33-50.

Ghosh, D. P., 1971, Inverse filter coefficients for the computation of apparent resistivity standard curves for a horizontally stratified earth: Geophysical Prospecting [Netherlands], v. 19, no. 4, p. 769-775.

REFERENCES (continued)

Zohdy, A. A. R., 1973, A computer program for the automatic interpretation of Schlumberger sounding curves over horizontally stratified media: U.S. Geological Survey Report USGS-GD-74-017, 25 p.; available only from U.S. Department of Commerce National Technical Information Service, Springfield, Virginia 22161 as PB-232 703.

_____1974a, A computer program for the calculation of Schlumberger sounding curves by convolution: U.S.Geological Survey Report USGS-GD-74-010, 11 p.; available only from U.S. Department of Commerce National Technical Information Service, Springfield, Virginia 22161 as PB-232 053.

_____1974b, Use of Dar Zarrouk curves in the interpretation of vertical electrical sounding data: U.S. Geological Survey Bulletin 1313-D, 41 p.

Zohdy, A. A. R., 1975, Automatic interpretation of Schlumberger sounding curves, using modified Dar Zarrouk functions: U.S. Geological Survey Bulletin 1313-E, 39 p.

Zohdy, A. A. R., and Bisdorf, R. J., 1975, Computer programs for the forward calculation and automatic inversion of Wenner sounding curves: available only from U.S. Department of Commerce National Technical Information Service, Springfield, Virginia 22161 as PB-247 265.

TITLE VES: Wenner/Schlumberger PAGE 5 OF 15 — TI Programmable **Program Record**

PROGRAMMER Donald N. Haines and David L. Campbell DATE 8/8/78

Partitioning (Op 17) 479.59 (Standard) Library Module None used. Printer Optional Cards 1-Wenner 1-Schlumb.

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Load sides 1 and 2 of appropriate (Wenner or Schlumberger) card.			
2	Initialize program...		RST A	0.
3	Input layer structure			
	a. Resistivity of layer i...	ρ_i	$x \leftrightarrow t$	0.
	b. Thickness of layer i...	h_i	B (or R/S)	i
	Repeat step 3 for all layers, from shallowest ($i=1$) to deepest ($i \leq 10$). Program takes final input layer to be an infinite half-space ($h=\infty$), no matter what value h_i you Key for it.			
4	(Optional) Input a_{max}...	a_{max}	2nd C	a_{max}
	Perform this step only if VES printed via PC-100 print cradle is desired. For manual output, skip to Step 5.			
5	Input starting spacing...	a_{min}	C	0
6	Start calculation of VES...		D	
	There will now be a several minute wait for output.			
	Printer output is automatic from this point, and consists of a set of pairs $[a, \rho_{app}(a)]$, sampled at 3 points per log cycle of a, from $a = a_{min}$ (exactly) to $a \sim a_{max}$ (approximately — output pairs continue until spacing "a" exceeds the value of a_{max} entered at step 4.)			
	For manual output (i.e., no value of a_{max} was entered at step 4, or a_{max} was set less than a_{min}), the program stops with $\rho_{app}(a_{min})$ in display. (To recall a_{min}, press $x \leftrightarrow t$.) Program is now ready to calculate a VES sampled 3 points per cycle. For each successive pair, press...		E $x \leftrightarrow t$	$\rho_{app}(a)$ a
	Note that apparent resistivities may be calculated manually (but slowly!) for arbitrary spacings a' via the above steps 5 and 6 by taking $a_{min} = a'$ for each value a' of interest.			
7	(Optional) To change a_{max}...	New a_{max}	2nd C E	New a_{max} Output as above.
8	(Optional) To change a_{min}...	New a_{min}	RST C	" " "
	Note in this case execution begins immediately on pressing C.			
9	For a new case, go to step 2.			
	UNITS: Units must be compatable; that is, if ρ's are in ohm-meters, say, then h's and a's must be in meters.			

TI Programmable Coding Form

PROGRAMMER ______ DATE ______

LOC	CODE	KEY	COMMENTS
000	76	LBL	Fcn A:
001	11	A	
002	47	CMS	Clear Mems.
003	29	CP	
004	71	SBR	Store Ghosh
005	42	STO	coeffs.
006	05	5	
007	02	2	Set storage
008	42	STO	reg. j.
009	08	08	
010	03	3	↑
011	05	5	
012	05	5	
013	07	7	Print
014	02	2	"R,H"
015	03	3	
016	69	OP	
017	01	01	
018	69	OP	↓
019	05	05	
020	76	LBL	Iff: Entry
021	87	IFF	for new
022	01	1	a_{min}.
023	09	9	
024	42	STO	
025	00	00	
026	09	9	Set
027	42	STO	initial
028	01	01	values
029	01	1	for
030	00	0	all
031	42	STO	index
032	02	02	regs.
033	42	STO	
034	03	03	
035	86	STF	
036	04	04	
037	00	0	
038	42	STO	
039	04	04	
040	42	STO	
041	05	05	
042	42	STO	
043	06	06	
044	92	RTN	
045	76	LBL	Fcn B:
046	12	B	
047	72	ST*	
048	08	08	Print &
049	69	OP	store
050	38	38	(ρ, h)
051	32	X⇄T	pairs.
052	72	ST*	
053	08	08	
054	99	PRT	

LOC	CODE	KEY	COMMENTS
055	01	1	
056	01	1	
057	32	X⇄T	
058	99	PRT	
059	69	OP	Count #
060	38	38	of pairs
061	69	OP	input:
062	27	27	if too
063	43	RCL	many,
064	07	07	go to
065	77	GE	GE.
066	77	GE	
067	29	CP	
068	98	ADV	
069	91	R/S	
070	61	GTO	
071	12	B	
072	76	LBL	GE:
073	77	GE	
074	69	OP	Display
075	37	37	flashing
076	33	X²	"11"
077	94	+/-	
078	34	√X	
079	92	RTN	
080	76	LBL	Fcn C':
081	18	C'	
082	42	STO	Store
083	53	53	a_{max}
084	91	R/S	
085	76	LBL	Fcn C:
086	13	C	
087	55	÷	
088	43	RCL	Get & store
089	54	54	X_{min}.
090	95	=	
091	42	STO	
092	30	30	
093	87	IFF	
094	04	04	Steps to
095	01	01	get going
096	01	01	again
097	71	SBR	after
098	87	IFF	new a_{min}
099	61	GTO	is input.
100	79	x̄	
101	01	1	↑
102	03	3	
103	05	5	
104	07	7	Print
105	03	3	
106	05	5	"A,RAPP
107	01	1	(A)"
108	03	3	
109	69	OP	

LOC	CODE	KEY	COMMENTS
110	01	01	
111	03	3	
112	03	3	
113	03	3	
114	03	3	(These
115	05	5	steps may
116	05	5	be dropped
117	01	1	if you
118	03	3	have no
119	05	5	print
120	06	6	cradle.)
121	69	OP	
122	02	02	
123	69	OP	
124	05	05	↓
125	69	OP	A stop
126	00	00	to allow
127	25	CLR	setting display
128	91	R/S	formats.
129	76	LBL	Fcn D:
130	14	D	
131	43	RCL	Save n
132	07	07	(= # layers)
133	42	STO	in a protected
134	29	29	register.
135	76	LBL	
136	89	π	π: Entry
137	69	OP	from Fcn E.
138	37	37	
139	01	1	
140	42	STO	initialize
141	32	32	B_i reg.
142	69	OP	
143	28	28	
144	76	LBL	Lbl:
145	76	LBL	
146	73	RC*	Begin loop
147	08	08	for Kernel
148	65	×	calc.
149	43	RCL	
150	32	32	
151	95	=	
152	42	STO	$R = \rho_i \cdot B_i$
153	32	32	
154	02	2	
155	44	SUM	
156	08	08	
157	73	RC*	
158	08	08	
159	75	-	

MERGED CODES

62	72 (STO)	83 (GTO)
63	73 (RCL)	84
64	74 (SUM)	92 (INV) (SBR)

TITLE **VES: Wenner/Schlumberger** PAGE **7** OF **15**

TI Programmable Coding Form

PROGRAMMER ______________ DATE ______________

LOC	CODE	KEY	COMMENTS
160	43	RCL	
161	32	32	
162	95	=	
163	55	÷	
164	53	(	$K_i = \frac{e_{i-1}-R}{e_{i+1}+R}$
165	73	RC*	
166	08	08	
167	85	+	
168	43	RCL	
169	32	32	
170	54	)	
171	95	=	
172	42	STO	
173	32	32	
174	69	OP	
175	28	28	
176	02	2	
177	94	+/-	
178	65	×	
179	73	RC*	
180	08	08	
181	55	÷	$Q = K_i e^m$
182	43	RCL	
183	30	30	
184	55	÷	where
185	43	RCL	
186	31	31	$m = \frac{-2h_{i-}}{f \cdot x}$
187	95	=	
188	22	INV	
189	23	LNX	
190	65	×	
191	43	RCL	
192	32	32	
193	95	=	
194	24	CE	
195	42	STO	
196	32	32	
197	94	+/-	
198	85	+	
199	01	1	
200	95	=	
201	55	÷	
202	53	(	
203	01	1	
204	85	+	$B_{i+1} = \frac{1-Q}{1+Q}$
205	43	RCL	
206	32	32	
207	54	)	
208	95	=	
209	42	STO	
210	32	32	
211	69	OP	
212	38	38	
213	97	DSZ	End Loop
214	07	07	for Kernel

LOC	CODE	KEY	COMMENTS
215	76	LBL	calculatn.
216	73	RC*	
217	08	08	Normalize:
218	49	PRD	New Kernel
219	32	32	element
220	43	RCL	$B(x) = B_n(x)$
221	32	32	e_i.
222	71	SBR	New Kernel
223	48	EXC	to convolve
224	22	INV	Test if
225	87	IFF	convolutn
226	02	02	has begun.
227	15	E	
228	99	PRT	
229	32	X⇌T	Print
230	99	PRT	$a, \rho_{app}(a)$
231	98	ADV	
232	48	EXC	
233	53	53	Test:
234	77	GE	Is $a > a_{mx}$
235	98	ADV	?
236	48	EXC	
237	53	53	If so, stop
238	91	R/S	
239	76	LBL	
240	98	ADV	If not,
241	48	EXC	go on.
242	53	53	
243	76	LBL	Fcn E:
244	15	E	
245	03	3	
246	35	1/X	Increment
247	22	INV	x
248	28	LOG	
249	65	×	
250	43	RCL	
251	30	30	
252	95	=	
253	42	STO	
254	30	30	
255	76	LBL	$\bar{x}$: Entry
256	79	$\bar{x}$	from Fcn C.
257	43	RCL	
258	29	29	
259	42	STO	Initialize
260	07	07	loop &
261	65	×	index
262	02	2	registers.
263	75	-	
264	01	1	
265	95	=	
266	22	INV	
267	44	SUM	
268	08	08	
269	61	GTO	

LOC	CODE	KEY	COMMENTS
270	89	π	
271	76	LBL	Exc:
272	48	EXC	
273	32	X⇌T	Enter
274	22	INV	new B_i
275	87	IFF	and
276	02	02	convolve.
277	78	Σ+	
278	69	OP	
279	32	32	
280	43	RCL	
281	00	00	
282	42	STO	
283	06	06	
284	69	OP	
285	20	20	
286	76	LBL	Dsz:
287	97	DSZ	
288	73	RC*	
289	00	00	Cascade
290	72	ST*	Kernel
291	06	06	elements
292	69	OP	B_i in
293	20	20	memory
294	69	OP	
295	26	26	
296	97	DSZ	
297	02	02	
298	97	DSZ	
299	43	RCL	
300	03	03	Prepare
301	42	STO	for next
302	02	02	B_i
303	69	OP	
304	30	30	
305	76	LBL	Σ+:
306	78	Σ+	
307	32	X⇌T	Store
308	72	ST*	newest B_i
309	00	00	
310	69	OP	
311	24	24	Check
312	87	IFF	whether
313	01	01	there
314	44	SUM	are now
315	69	OP	enough
316	20	20	B_i's to
317	69	OP	convolve
318	24	24	
319	43	RCL	

MERGED CODES
62 63 64 72 (STO) 73 (RCL) 74 (SUM) 83 (GTO) 84 92 (INV) (SBR)

TITLE VES: Wenner/Schlumberger PAGE 8 OF 15 TI Programmable Coding Form

PROGRAMMER ______ DATE ______

LOC	CODE	KEY	COMMENTS
320	02	02	Check whether can convl after B_i is input.
321	32	X:T	
322	43	RCL	
323	04	04	
324	67	EQ	
325	86	STF	
326	69	OP	Return to calc. next B_i.
327	34	34	
328	43	RCL	
329	04	04	
330	92	RTN	
331	76	LBL	Sum:
332	44	SUM	
333	73	RC*	
334	01	01	
335	65	×	
336	73	RC*	
337	00	00	
338	95	=	
339	44	SUM	Convolve
340	05	05	
341	69	OP	
342	21	21	
343	69	OP	
344	30	30	
345	97	DSZ	
346	02	02	
347	44	SUM	
348	69	OP	Restore loop reg index value.
349	20	20	
350	43	RCL	
351	03	03	
352	42	STO	
353	02	02	
354	94	+/-	Init. index reg.
355	44	SUM	
356	01	01	
357	86	STF	Zero the Σ (convolve) accumulator.
358	02	02	
359	43	RCL	
360	05	05	
361	94	+/-	
362	44	SUM	
363	05	05	
364	94	+/-	
365	32	X:T	Convert x back to spacing "a".
366	43	RCL	
367	30	30	
368	55	÷	
369	43	RCL	
370	55	55	
371	95	=	
372	92	RTN	
373	76	LBL	Stf:
374	86	STF	
375	86	STF	Set flag 1 to mean "Ready to convolve"
376	01	01	
377	69	OP	
378	34	34	
379	29	CP	
380	92	RTN	

Both Wenner and Schlumberger procedures have all the same steps to this point. From here on (locations 381 through 479) the steps differ, for Wenner and Schlumberger cards have separate "STO" subroutines. These two versions of subroutine "STO" are given on pages 9 and 10.

MERGED CODES
62 | 72 (STO) | 83 (GTO)
63 | 73 (RCL) | 84
64 | 74 (SUM) | 92 (INV) (SBR)

TEXAS INSTRUMENTS INCORPORATED

TI-24181

TITLE VES: Wenner PAGE 9 OF 15 **TI Programmable Coding Form**

PROGRAMMER ____ DATE ____

LOC	CODE	KEY	COMMENTS
381	76	LBL	Subr STO:
382	42	STO	
383	93	.	
384	00	0	
385	02	2	
386	08	8	
387	04	4	
388	42	STO	
389	09	09	
390	93	.	
391	04	4	
392	05	5	
393	08	8	
394	02	2	
395	42	STO	
396	10	10	
397	01	1	
398	93	.	
399	05	5	
400	06	6	
401	06	6	
402	02	2	
403	42	STO	Store
404	11	11	Ghosh
405	01	1	coeffs.
406	93	.	for
407	03	3	Wenner
408	03	3	
409	04	4	
410	01	1	
411	94	+/-	
412	42	STO	
413	12	12	
414	93	.	
415	03	3	
416	04	4	
417	07	7	
418	03	3	
419	42	STO	
420	13	13	
421	93	.	
422	00	0	
423	09	9	
424	03	3	
425	05	5	
426	94	+/-	
427	42	STO	
428	14	14	
429	93	.	

LOC	CODE	KEY	COMMENTS
430	00	0	
431	04	4	
432	01	1	
433	06	6	
434	42	STO	
435	15	15	
436	93	.	
437	00	0	
438	02	2	
439	05	5	
440	03	3	
441	94	+/-	
442	42	STO	
443	16	16	
444	93	.	
445	00	0	
446	01	1	
447	07	7	
448	09	9	
449	42	STO	
450	17	17	
451	93	.	
452	00	0	
453	00	0	
454	06	6	
455	07	7	
456	94	+/-	
457	42	STO	
458	18	18	
459	03	3	
460	35	1/X	Store lag
461	22	INV	factor for
462	28	LOG	convolution
463	42	STO	process.
464	55	55	
465	45	Y^X	Store
466	08	8	factor
467	95	=	relating
468	42	STO	a to x
469	54	54	
470	01	1	
471	93	.	Store
472	03	3	factor
473	06	6	f.
474	42	STO	
475	31	31	
476	92	RTN	
477	00	0	
478	00	0	
479	00	0	

MERGED CODES

62 ▭ ▭ 72 (STO) ▭ 83 (GTO) ▭
63 ▭ ▭ 73 (RCL) ▭ 84 ▭ ▭
64 ▭ ▭ 74 (SUM) ▭ 92 (INV) (SBR)

TEXAS INSTRUMENTS INCORPORATED

TI-34181

TITLE VES: Schlumberger PAGE 10 OF 15 **TI Programmable Coding Form**

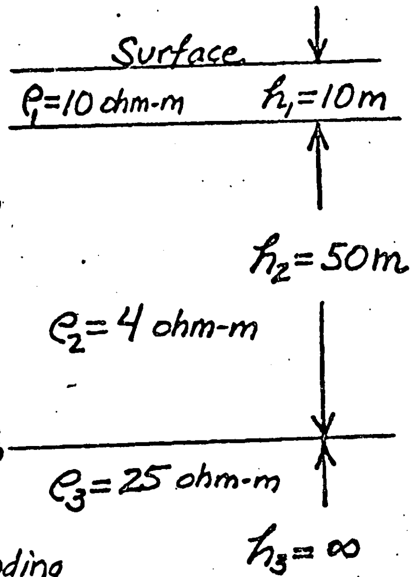
PROGRAMMER ________ DATE ________

LOC	CODE	KEY	COMMENTS
381	76	LBL	Subr STO:
382	42	STO	
383	93	.	
384	00	0	
385	02	2	
386	02	2,	
387	05	5	
388	42	STO	
389	09	09	
390	93	.	
391	00	0	
392	04	4	
393	09	9	
394	09	9	
395	94	+/-	
396	42	STO	
397	10	10	
398	93	.	
399	01	1	
400	00	0	
401	06	6	
402	04	4	
403	42	STO	Store
404	11	11	Ghosh
405	93	.	coeffs.
406	01	1	for
407	08	8	Schlumb.
408	05	5	
409	04	4	
410	42	STO	
411	12	12	
412	01	1	
413	93	.	
414	09	9	
415	07	7	
416	02	2	
417	42	STO	
418	13	13	
419	01	1	
420	93	.	
421	05	5	
422	07	7	
423	01	1	
424	06	6	
425	94	+/-	
426	42	STO	
427	14	14	
428	93	.	
429	04	4	

LOC	CODE	KEY	COMMENTS
430	00	0	
431	01	1	
432	08	8	
433	42	STO	
434	15	15	
435	93	.	
436	00	0	
437	08	8	
438	01	1	
439	04	4	
440	94	+/-	
441	42	STO	
442	16	16	
442	93	.	
444	00	0	
445	01	1	
446	04	4	
447	08	8	
448	42	STO	
449	17	17	
450	00	0	
451	42	STO	
452	18	18	
453	01	1	
454	00	0	Store lag
455	42	STO	factor for
456	55	55	convolution
457	01	1	Store
458	00	0	factor
459	00	0	relating
460	42	STO	a to x
461	54	54	
462	01	1	
463	93	.	Store
464	00	0	factor
465	05	5	f.
466	42	STO	
467	31	31	
468	92	RTN	
469	00	0	
470	00	0	
471	00	0	
472	00	0	
473	00	0	
474	00	0	
475	00	0	
476	00	0	
477	00	0	
478	00	0	
479	00	0	

MERGED CODES
62 63 64 72 STO 73 RCL 74 SUM 83 GTO 84 92 INV SBR

EXAMPLE: Calculate VES curve for this structure.



SOLUTION: Suppose 6 points per log cycle are adequate for our purposes, for a-values from 1m to 1000m.

Since program calculates 3 points/cycle, we'll make 2 passes — the first to get values at $a = 10^{0/6} = 1$, $10^{2/6} = 2.15$, $10^{4/6} = 4.64$, etc.; the second to give values at $a = 10^{1/6} = 1.47$, $10^{3/6} = 3.16$, $10^{5/6} = 6.82$, etc.

Values given in chart below are for Wenner calculation. Corresponding values on printer tape for Schlumberger case are on page 14.

STEP (p.5)	VARIABLE	KEY	DISPLAY
2.		RST A	0.
3.	$\rho_1 = 10$	x⇆t	0.
	$h_1 = 10$	B	i = 1.
	$\rho_2 = 4$	x⇆t	0.
	$h_2 = 50$	B	i = 2.
	$\rho_3 = 25$	x⇆t	0.
	$h_3 = 100000$	B	i = 3.
	(Because this is the deepest layer, any value of h_3 may be input here.)		

The procedures for manual and printer output are slightly different from this point on. Instructions for printer output follow on pages 12, 13 and 14. For manual output instructions, skip to page 15.

For printer output:

STEP	VARIABLE	KEY	DISPLAY	PRINTOUT
• Echo print of input layer structure, as shown at right, will have resulted from steps 2 and 3. (Note printer gives "R" instead of Greek letter "ρ".)				R, H 10. 10. 4. 50. 25. 100000.
4.	$a_{max} = 100$	2nd C	100.	
Note a_{max} is purposely chosen too small in this example so as to demonstrate the "change a_{max}" option later on...				
5.	$a_{min} = 1$	C	0.	A, RAPP(A) 1. 9.999734309
Printer responds with heading "A, RAPP(A)"; again, "R" in place of Greek "ρ".				2.15443469 9.985672905
6.		D	...12.59139515	4.641588834 9.77223771
After a few minute's wait, printout of $[a, \rho_{app}(a)]$ pairs proceeds. The first number of each pair is "a"; the second is $\rho_{app}(a)$.				10. 8.539385065
Note calculation actually proceeds one pair past input value of a_{max} ($\cong 100$, here).				21.5443469 5.992546854
Suppose at this point we wish to change display format to suppress insignificant decimals in output. Key...				46.41588834 5.221504638
		2nd Fix 2	12.59	100. 7.742246853
(This step may actually be done at any time after step 5... if performed before step 5 the alphanumeric headings will be scrambled.)				215.443469 12.59139515
7.	New $a_{max} = 1000$	2nd C	1000.00	464.16 18.00
		E	... 24.10	
Printout again proceeds. As before, calculations continue one point past nominal a_{max}.				1000.00 22.09 2154.43 24.10

Printer output (continued):

STEP	VARIABLE	KEY	DISPLAY	PRINT-OUT
Now we're ready for 2nd pass through the program, interpolating e_{app} values between those calculated during the first pass. (This procedure also demonstrates how to change a_{min}, step 8 on page 5.)				1.47 10.00 3.16 9.94 6.81 9.36
First calculate new $a_{min} = 10^{1/6} = 1.47$:				
	6	1/x	.17	14.68 7.29
		Inv 2nd log	1.47	
8.	New $a_{min} = 1.47$	Rst C	... 23.33	31.62 5.20
(Printout of interpolated $[a, \rho_{app}(a)]$ pairs follows ...)				68.13 6.10 146.78 9.98 316.23 15.35 681.29 20.30 1467.80 23.33

Printer tapes for Schlumberger case:

STEP	PRINTOUT
2.	R, H
	10.
3.	10.
	4.
	50.
	25.
	100000.
5.	A, RAPP(A)
6.	1.
	9.979896606
	2.15443469
	9.977938627
	4.641588834
	9.907182152
	10.
	9.319470423
	21.5443469
	7.047480762
	46.41588834
	5.094844712
	100.
	6.350793289
	215.443469
	10.50266499
7.	464.16
	15.96
	1000.00
	20.81
	2154.43
	23.60

STEP	PRINTOUT
8.	1.47
	9.98
	3.16
	9.97
	6.81
	9.74
	14.68
	8.40
	31.62
	5.76
	68.13
	5.29
	146.78
	8.15
	316.23
	13.18
	681.29
	18.58
	1467.80
	22.48

For manual output:

STEP	VARIABLE	KEY	DISPLAY
5.	$a_{min} = 1$	C	
6.		D	$(e_a =)$ 9.999734309
		x⇌t	$(a =)$ 1.
		E	$(e_a =)$ 9.985672905
		x⇌t	$(a =)$ 2.15443469
		E	$(e_a =)$ 9.77223771
		x⇌t	$(a =)$ 4.641588834
:		and so on...	

Now we're ready for 2nd pass through the program, interpolating e_{app} values between those calculated during first pass! (This procedure also demonstrates how to change a_{min} — step 8 on page 5.)

8.	6	1/x	.1666666667
		Inv 2nd log	1.467799268
	New a_{min} = 1.467799268	Rst C	$(e_a =)$ 9.997507923
		x⇌t	$(a =)$ 1.467799268
		E	$(e_a =)$ 9.935229902
		x⇌t	$(a =)$ 3.16227766

and so on...

(Check manual output values by comparing with printer tapes shown on pages 12-14.)

Note display format may be changed at any point after step 5 to supress insignificant decimals in display; that is,

	2nd Fix 2 ...	$(e_a =)$ 10.00
...	x⇌t	$(a =)$ 1.47
	E	$(e_a =)$ 9.94
	x⇌t	$(a =)$ 3.16

and so on, as on page 13.