THE EFFECT OF TEMPERATURE AND PRESSURE ON LABORATORY OXIDIZED ASPHALT FILMS WITH COMPARISON TO FIELD AGING

Volume II

A Dissertation
by
KEVIN MICHAEL LUNSFORD

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 1994

Major Subject: Chemical Engineering

THE EFFECT OF TEMPERATURE AND PRESSURE ON LABORATORY OXIDIZED ASPHALT FILMS WITH COMPARISON TO FIELD AGING

Volume II

A Dissertation

by

KEVIN MICHAEL LUNSFORD

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Charles J. Glover

Charles J. Glover
(Chair of Committee)

Richard R. Davison

Richard R. Davison
(Member)

Alan Letton
(Member)

Jerry A. Bullin

Jerry A. Bullin
(Member)

Kenneth R. Hall
(Interim Department Head)

May 1994

Major Subject: Chemical Engineering

TABLE OF CONTENTS

Volume I

Volume II

APPENDIX

LIST OF TABLES

Volume I

LIST OF FIGURES

Volume I

Figure Page

Figure Page

APPENDIX A

NEAT DATA

Table A-1. Properties of All Neat Asphalts Studied[a]

Reference	Asphalt	CA	η_o^*[b] P	$(1\,/\,J'')$[c] dyne / cm^2	MW	$\%A$ wt/wt
Lau *et al.*, (1992)						
	Ampet AC-20	0.452	2690	25400	–	16.37
	Coastal AC-20	0.461	2570	24300	–	23.80
	Cosden AC-20	0.462	2080	20400	–	18.81
	Exxon AC-20	0.450	2700	26700	–	14.69
	Texaco AC-20	0.438	2150	20400	–	23.49
SHRP (1990)						
	AAA-1	0.497	915	9890	–	21.70
	AAC-1	0.431	943	12800	–	15.58
	AAD-1	0.734	1332	16000	–	26.32
	AAG-1	0.642	1925	23000	–	10.34
	AAK-2	0.585	1420	13500	–	23.25
	AAM-1	0.434	2230	–	–	7.26
Jemison *et al.*, (1992)						
	Coastal AC-20	0.479	2350	–	–	23.34
	Fina AC-20	0.458	2570	–	–	18.51
	Texaco AC-20	0.452	2780	–	–	22.86
Adams and Holgreen (1986)						
Dickens						
	Cosden AC-10	0.405	950	9450	1523	–
	Cosden AC-20	0.474	1650	16140	1597	–
	D.S.[c] AC-20	0.421	2700	24900	3677	–
	Dorchester AC-20	0.396	1850	17100	2069	–
	Exxon AC-20	0.499	2200	21700	1669	–
	MacMillan AC-20	0.498	2200	19500	2501	–
Pineland						
	Cosden AC-20	0.493	1860	18200	1503	–
	Dorchester AC-20	0.512	1360	13200	1776	–
	Exxon AC-20	0.488	2470	25000	1538	–
	MacMillan AC-20	0.460	1700	16300	2452	–
	Texaco AC-20	0.519	1460	14200	1892	–
Davison *et al.*, (1989)						
Bryan						
	Exxon AC-20 #8	0.460	2130	21200	1621	–
	Exxon AC-20 #15	0.447	2130	21200	–	–
	Exxon AC-20 #16	0.470	1860	18400	–	–
	Exxon AC-20 #18	0.452	1950	19400	–	–
	Exxon AC-20 #19	0.450	2020	20200	–	–
	Exxon AC-20 #1B	0.469	2560	25600	–	–

[a] – Signifies the values were not determined.

[b] At 333.3 K

[c] At 333.3 K and 10 rad/s

[d] D.S. represents Diamond Shamrock

APPENDIX B

CARBONYL FORMATION DATA

Table B-1. Ampet AC-20 Kinetic Data[a]

	0.2 atm[b]			2 atm[b]			20 atm[c]		
t days	333.3[d] *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*
2	–	–	0.637	–	–	–	–	–	0.890
4	–	–	0.679	–	0.745	0.917	–	–	–
5	–	–	–	0.764	–	–	–	–	1.046
6	–	0.681	0.698	–	–	–	–	0.941	–
7	–	–	–	–	–	–	0.752	–	–
8	0.712	0.705	0.797	–	–	–	–	–	–
9	–	–	–	–	0.772	1.014	–	–	–
10	–	–	0.777	–	–	1.063	–	–	–
12	–	0.730	–	0.793	0.882	–	–	–	–
13	–	–	–	–	–	–	–	1.247	1.496
14	–	–	–	–	–	1.276	–	–	–
17	–	–	–	–	0.917	–	–	–	–
20	–	0.814	0.975	–	0.927	–	1.017	1.424	1.995
21	–	–	–	0.902	–	1.468	–	–	–
24	0.728	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–	2.583
28	–	0.821	–	0.925	1.056	–	–	–	–
30	–	–	–	–	–	–	1.060	1.697	–
32	0.753	0.882	–	–	1.212	–	–	–	–
36	–	–	–	–	1.151	–	–	–	–
40	–	–	–	–	1.262	–	–	–	–
41	0.815	–	–	–	–	–	–	–	–
44	–	–	–	0.993	–	–	–	2.078	–
50	–	–	–	–	–	–	1.310	–	–
57	0.865	–	–	–	–	–	–	–	–
58	–	–	–	–	–	–	–	2.436	–
68	–	–	–	1.237	–	–	–	–	–
70	–	–	–	–	–	–	–	2.567	–
72	0.885	–	–	–	–	–	–	–	–
76	–	–	–	1.218	–	–	–	–	–
80	0.938	–	–	–	–	–	–	–	–

---

[a] – Signifies the values were not determined.

[b] Exposed surface analysis

[c] Bulk analysis

[d] Units of K

Table B-2. Coastal AC-20 Kinetic Data[a]

	0.2 atm[b]			2 atm[b]			20 atm[c]		
t days	333.3[d] *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*
2	–	–	0.595	–	–	0.845	–	–	1.037
4	–	–	0.710	–	0.793	–	–	–	–
5	–	–	–	0.823	–	0.894	–	–	1.224
6	–	–	0.652	–	–	–	–	1.138	–
7	–	–	–	–	–	–	0.992	–	–
8	0.595	0.720	0.756	–	–	–	–	–	–
9	–	–	–	–	0.846	–	–	–	–
10	–	–	0.835	–	–	–	–	–	–
12	–	0.750	–	0.885	0.992	–	–	–	–
13	–	–	–	–	–	–	–	1.386	1.754
14	–	–	–	–	–	1.250	–	–	–
15	–	–	0.982	–	–	–	–	–	–
16	–	0.791	0.773	–	–	–	–	–	–
17	–	–	–	–	1.104	–	–	–	–
18	–	–	0.986	–	–	–	–	–	–
20	–	0.790	0.918	–	–	–	1.235	1.682	2.464
21	–	–	–	0.890	–	1.570	–	–	–
23	–	–	1.140	–	–	1.591	–	–	–
24	0.710	0.862	–	–	–	–	–	–	–
28	–	0.874	–	0.951	–	–	–	–	–
30	–	–	–	–	–	–	1.326	1.934	–
32	0.815	0.943	–	–	1.306	–	–	–	–
36	–	0.974	–	1.049	1.253	–	–	–	–
40	–	–	–	–	1.398	–	–	–	–
41	0.791	0.998	–	–	–	–	–	–	–
44	–	–	–	1.044	–	–	–	2.493	–
48	0.797	–	–	–	–	–	–	–	–
50	–	–	–	–	–	–	1.517	–	–
52	–	–	–	1.180	–	–	–	–	–
57	0.795	–	–	–	–	–	–	–	–
58	–	–	–	–	–	–	–	2.845	–
60	–	–	–	1.285	–	–	–	–	–
64	0.856	–	–	–	–	–	–	–	–
68	–	–	–	1.161	–	–	–	–	–
70	–	–	–	–	–	–	1.753	3.100	–
72	0.873	–	–	–	–	–	–	–	–
76	–	–	–	1.237	–	–	–	–	–

---

[a] – Signifies the values were not determined.

[b] Exposed surface analysis

[c] Bulk analysis

[d] Units of K

Table B-3. Cosden AC-20 Kinetic Data[a]

	0.2 atm[b]			2 atm[b]			20 atm[c]		
t days	333.3[d] *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*
2	–	–	0.646	–	–	0.946	–	–	1.290
4	–	–	0.767	–	0.833	–	–	–	–
5	–	–	–	0.856	–	–	–	–	1.688
6	–	–	0.762	–	–	1.006	–	–	–
7	–	–	–	–	–	–	1.052	1.468	–
8	0.694	0.763	0.912	–	–	–	–	–	–
9	–	–	–	–	0.943	1.209	–	–	–
12	–	0.778	–	0.945	–	–	–	–	–
13	–	–	–	–	–	–	–	–	2.385
14	–	–	–	–	–	1.375	–	2.020	–
16	0.739	–	1.041	–	–	–	–	–	–
17	–	0.788	–	–	1.088	–	–	–	–
18	–	–	1.069	–	–	1.426	–	–	–
20	–	0.822	1.095	–	1.153	–	1.598	–	3.026
21	–	–	–	1.053	–	1.769	–	–	–
24	0.831	0.951	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	2.270	–
27	–	–	–	–	–	–	–	–	3.497
30	–	–	–	–	–	–	1.904	–	–
32	0.765	1.015	–	–	1.393	–	–	–	–
35	–	–	–	–	–	–	–	2.465	–
36	–	–	–	–	–	–	–	–	–
44	–	–	–	1.249	–	–	–	–	–
48	0.929	–	–	–	–	–	–	–	–
50	–	–	–	–	–	–	2.166	–	–
52	–	–	–	1.318	–	–	–	–	–
57	0.917	–	–	–	–	–	–	–	–
60	–	–	–	1.430	–	–	–	–	–
64	0.975	–	–	–	–	–	–	–	–
70	–	–	–	–	–	–	2.292	–	–
72	1.031	–	–	–	–	–	–	–	–
76	–	–	–	1.487	–	–	–	–	–
80	1.027	–	–	–	–	–	–	–	–

---

[a] – Signifies the values were not determined.

[b] Exposed surface analysis

[c] Bulk analysis

[d] Units of K

Table B-4. Exxon AC-20 Carbonyl Areas[a]

	0.2 atm[b]			2 atm[b]			20 atm[c]		
t days	333.3[d] *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*
2	–	–	0.740	–	–	0.871	–	–	0.953
4	–	0.745	0.853	–	0.813	–	–	–	–
5	–	–	–	0.875	–	–	–	–	1.471
6	–	–	–	–	0.918	0.955	–	–	–
7	–	–	–	–	–	–	0.949	1.268	–
8	–	0.759	0.966	–	–	–	–	–	–
9	–	–	–	–	–	1.146	–	–	–
10	–	–	1.033	–	–	–	–	–	–
12	–	0.826	–	0.988	–	–	–	–	–
13	–	–	–	–	–	–	–	–	1.727
14	–	–	–	–	–	1.267	–	1.465	–
16	0.744	0.894	1.089	–	–	1.283	–	–	–
17	–	–	–	–	1.037	–	–	–	–
18	–	–	–	–	–	1.400	–	–	–
20	–	0.866	1.149	–	1.051	–	1.315	–	2.074
21	–	–	–	1.009	–	–	–	–	–
23	–	–	1.200	–	–	–	–	–	–
24	0.819	0.922	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	1.872	–
27	–	–	–	–	–	–	–	–	2.651
28	–	0.889	–	1.032	1.192	–	–	–	–
30	–	–	–	–	–	–	1.547	–	–
32	–	1.033	–	–	–	–	–	–	–
35	–	–	–	–	–	–	–	2.071	–
36	–	0.958	–	1.226	–	–	–	–	–
40	–	–	–	–	1.319	–	–	–	–
41	–	1.056	–	–	–	–	–	–	–
44	–	–	–	1.218	–	–	–	–	–
48	0.909	–	–	–	–	–	–	–	–
50	–	–	–	–	–	–	1.707	–	–
52	–	–	–	1.314	–	–	–	–	–
60	–	–	–	1.412	–	–	–	–	–
68	–	–	–	1.386	–	–	–	–	–
70	–	–	–	–	–	–	1.775	–	–
76	–	–	–	1.426	–	–	–	–	–
80	1.027	–	–	–	–	–	–	–	–

[a] – Signifies the values were not determined.

[b] Exposed surface analysis

[c] Bulk analysis

[d] Units of K

Table B-5. Texaco AC-20 Kinetic Data[a]

	0.2 atm[b]			2 atm[b]			20 atm[c]		
t days	333.3[d] *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*	333.3 *CA*	344.4 *CA*	355.5 *CA*
2	–	–	0.594	–	–	0.747	–	0.731	0.946
4	–	0.617	0.623	–	0.699	–	–	–	–
5	–	–	–	0.729	–	–	–	–	1.130
6	–	–	0.656	–	–	0.927	–	0.973	–
7	–	–	–	–	–	–	0.782	–	–
8	–	0.644	0.722	–	–	–	–	–	–
9	–	–	–	–	0.809	–	–	–	–
10	–	–	0.772	–	–	–	–	–	–
12	–	–	–	0.765	0.856	–	–	–	–
13	–	–	–	–	–	–	–	1.247	1.620
14	–	–	–	–	–	1.182	–	–	–
15	–	–	0.853	–	–	–	–	–	–
16	0.560	0.726	0.848	–	–	–	–	–	–
17	–	–	–	–	0.918	–	–	–	–
18	–	–	–		–	1.172	–	–	–
20	–	0.714	0.918	–	–	–	1.048	1.374	1.985
21	–	–	–	0.819	–	1.353	–	–	–
23	–	–	–	–	–	1.354	–	–	–
24	0.667	0.726	–	–	1.045	–	–	–	–
27	–	–	–	–	–	–	–	–	2.363
28	–	0.793	–	0.850	–	–	–	–	–
30	–	–	–	–	–	–	1.195	1.730	–
32	0.622	0.851	–	–	1.115	–	–	–	–
36	–	0.896	–	0.912	–	–	–	–	–
41	0.716	0.878	–	–	–	–	–	–	–
44	–	–	–	0.971	–	–	–	2.002	–
48	0.763	–	–	–	–	–	–	–	–
50	–	–	–	–	–	–	1.364	–	–
52	–	–	–	1.100	–	–	–	–	–
58	–	–	–	–	–	–	–	2.315	–
60	–	–	–	1.011	–	–	–	–	–
64	0.766	–	–	–	–	–	–	–	–
68	–	–	–	1.120	–	–	–	–	–
70	–	–	–	–	–	–	1.505	2.687	–
72	0.803	–	–	–	–	–	–	–	–
76	–	–	–	1.132	–	–	–	–	–
80	0.815	–	–	–	–	–	–	–	–

---

[a] – Signifies the values were not determined.

[b] Exposed surface analysis

[c] Bulk analysis

[d] Units of K

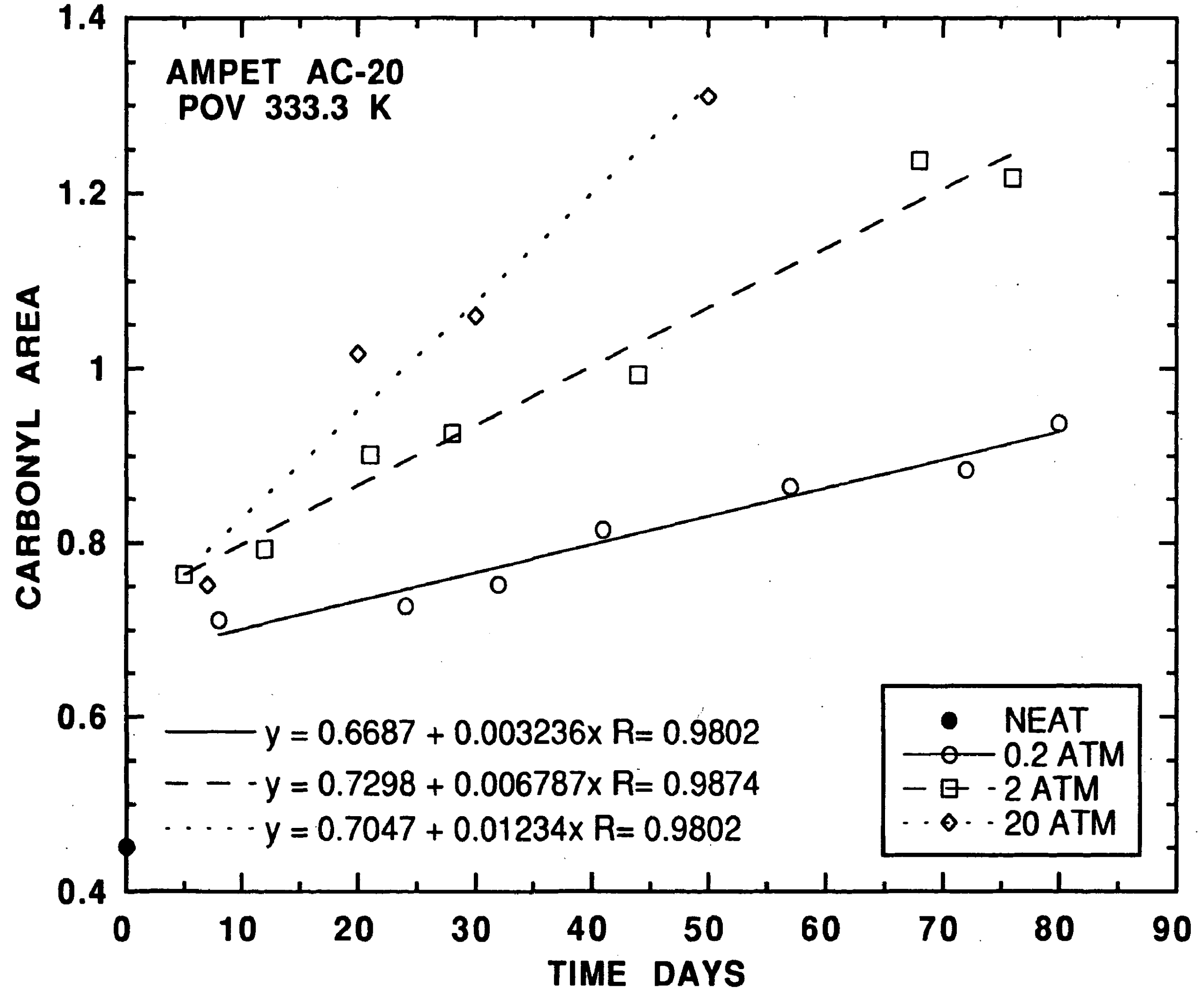

Figure B-1. *CA*s of neat and POV-aged Ampet AC-20 at 333.3 K and 0.2, 2, and 20 atm.

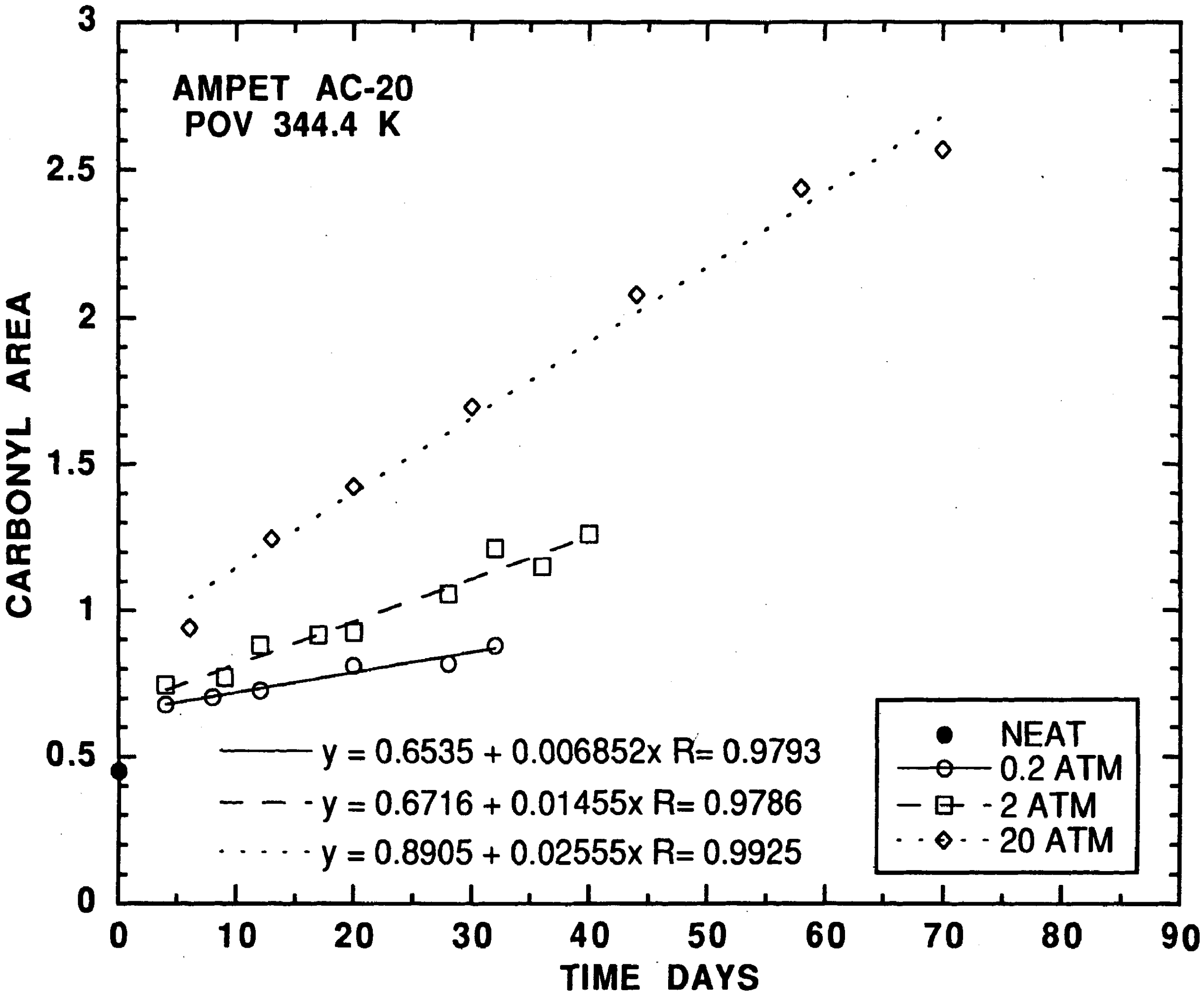

Figure B-2. *CA*s of neat and POV-aged Ampet AC-20 at 344.4 K and 0.2, 2, and 20 atm.

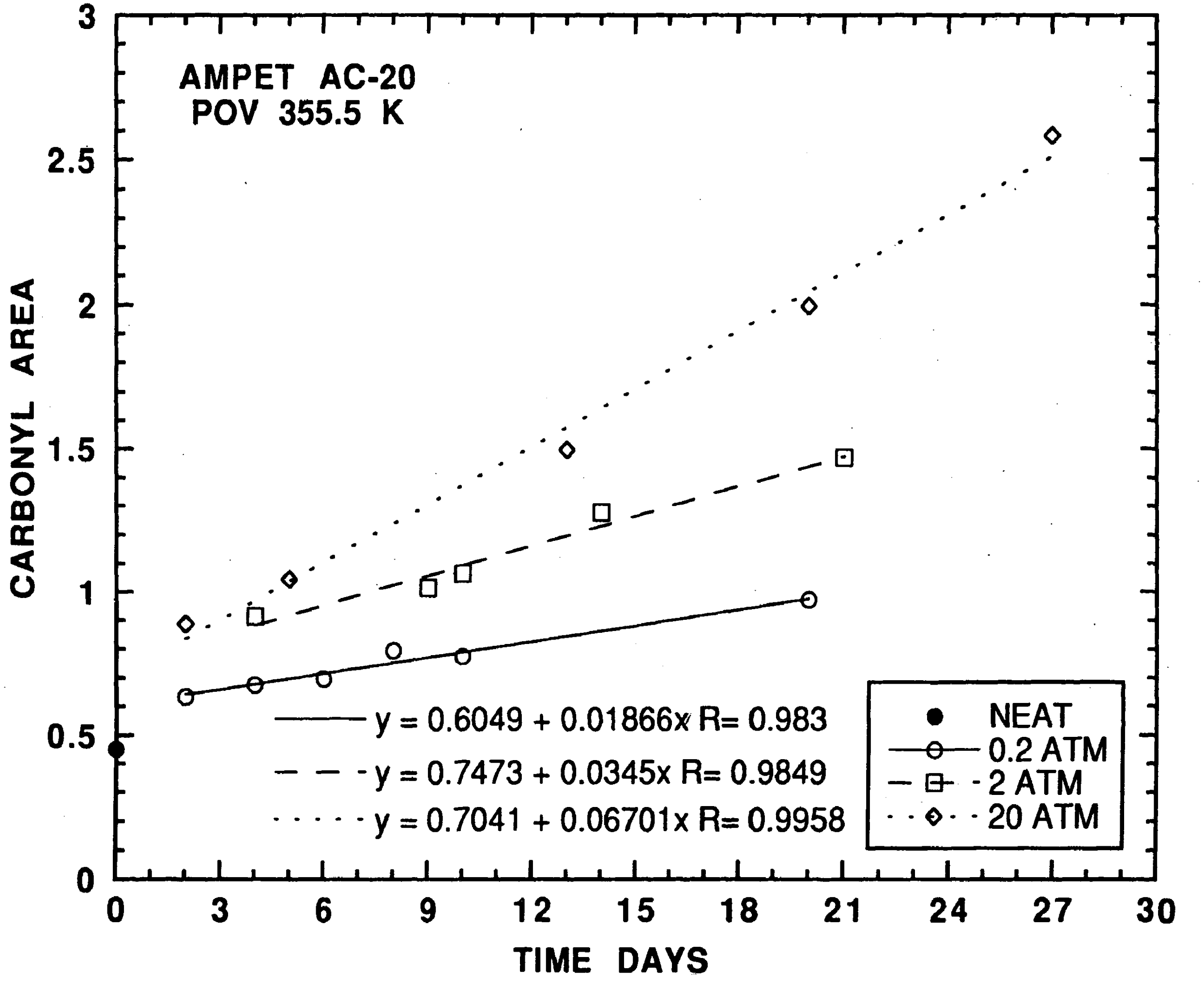

Figure B-3. *CA*s of neat and POV-aged Ampet AC-20 at 355.5 K and 0.2, 2, and 20 atm.

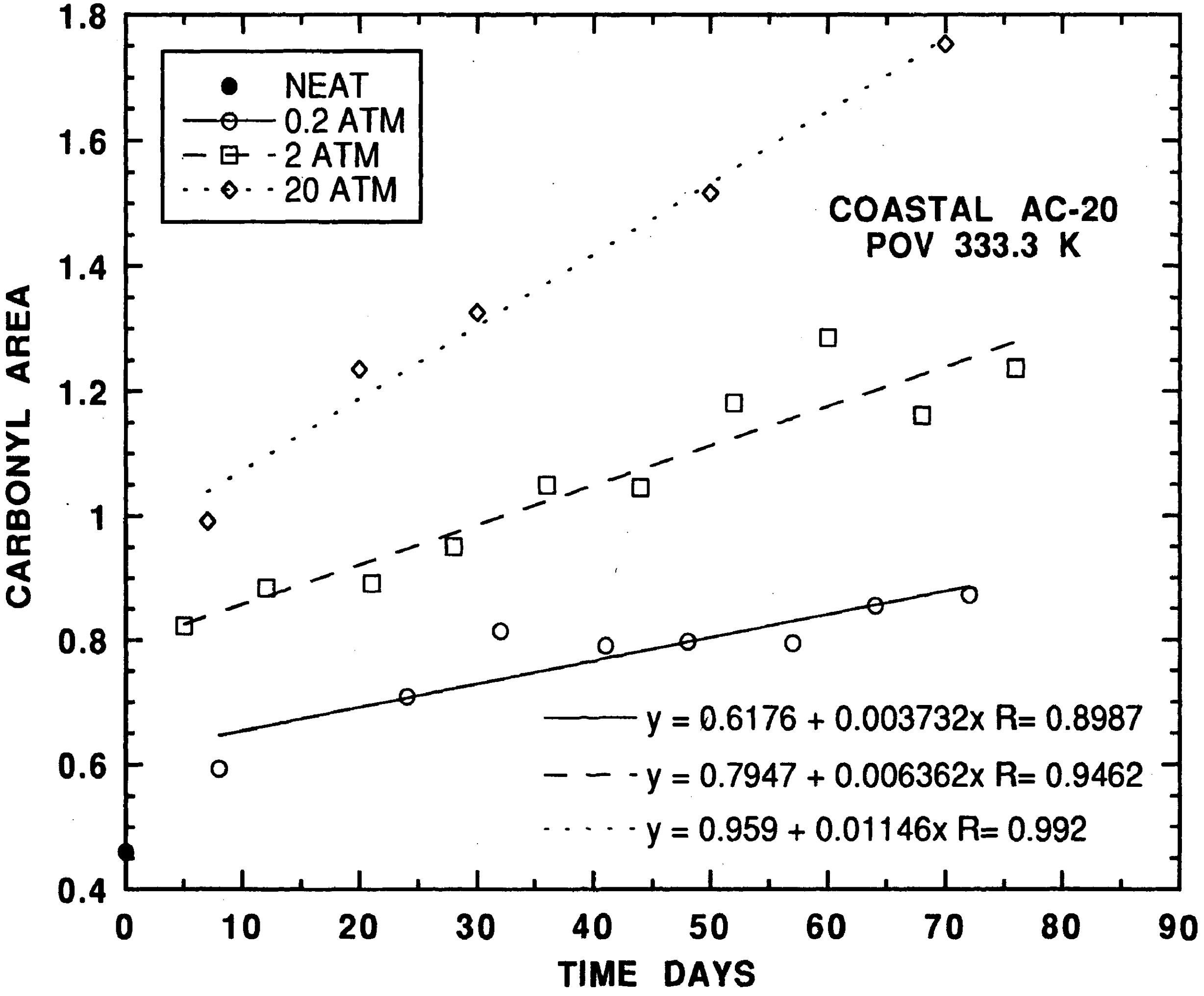

Figure B-4. *CA*s of neat and POV-aged Coastal AC-20 at 333.3 K and 0.2, 2, and 20 atm.

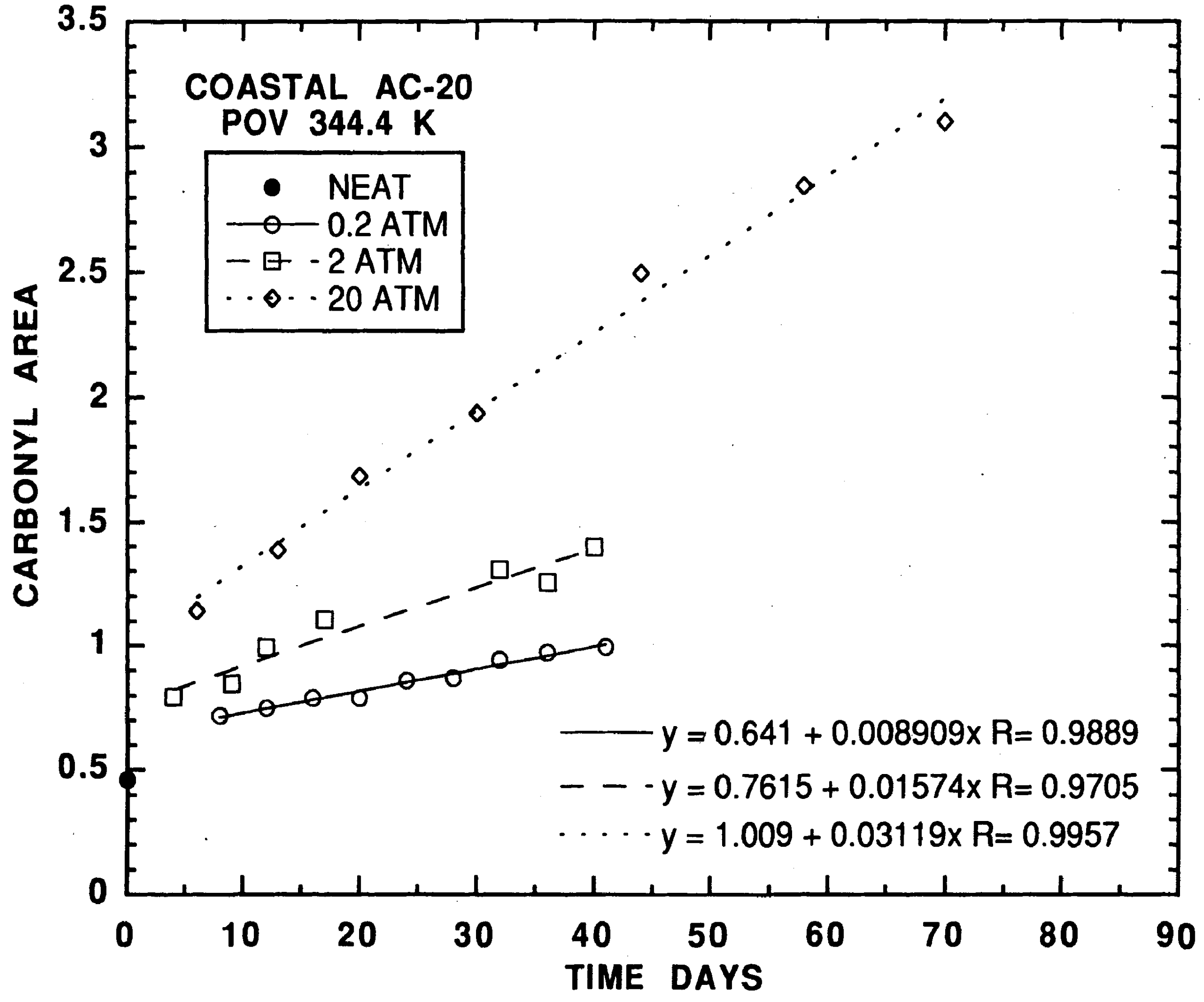

Figure B-5. *CA*s of neat and POV-aged Coastal AC-20 at 344.4 K and 0.2, 2, and 20 atm.

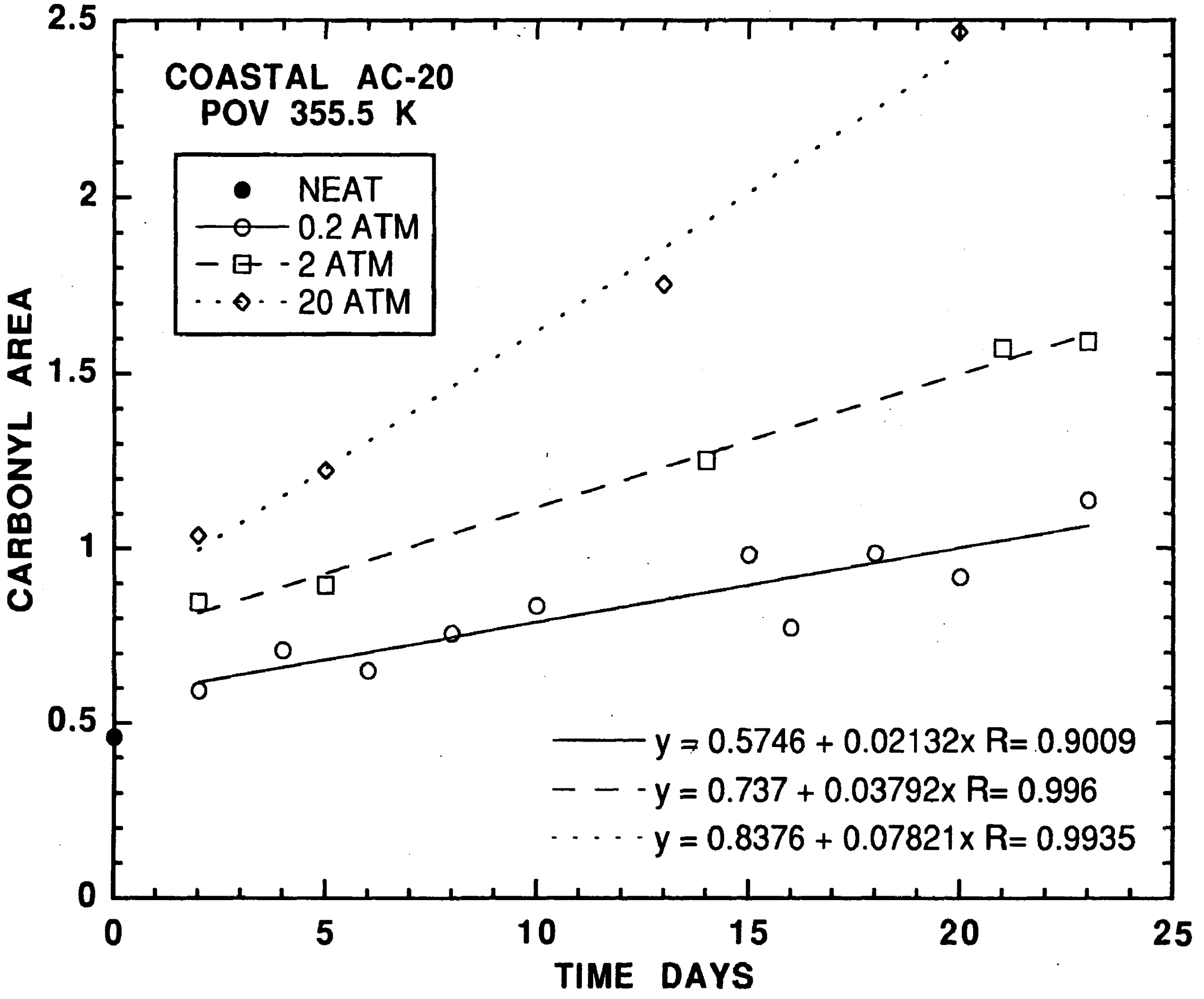

Figure B-6. *CA*s of neat and POV-aged Coastal AC-20 at 355.5 K and 0.2, 2, and 20 atm.

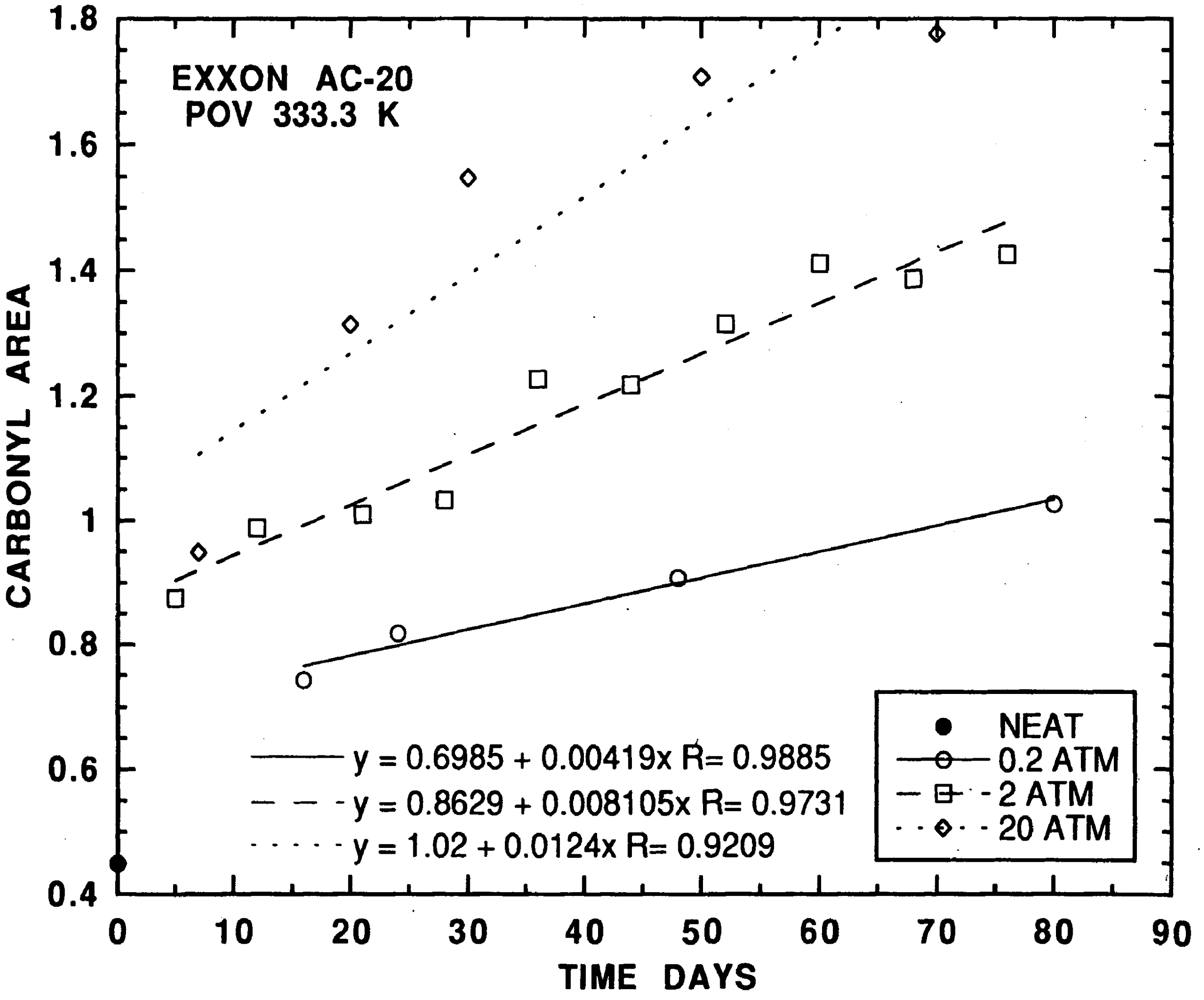

Figure B-7. *CA*s of neat and POV-aged Exxon AC-20 at 333.3 K and 0.2, 2, and 20 atm.

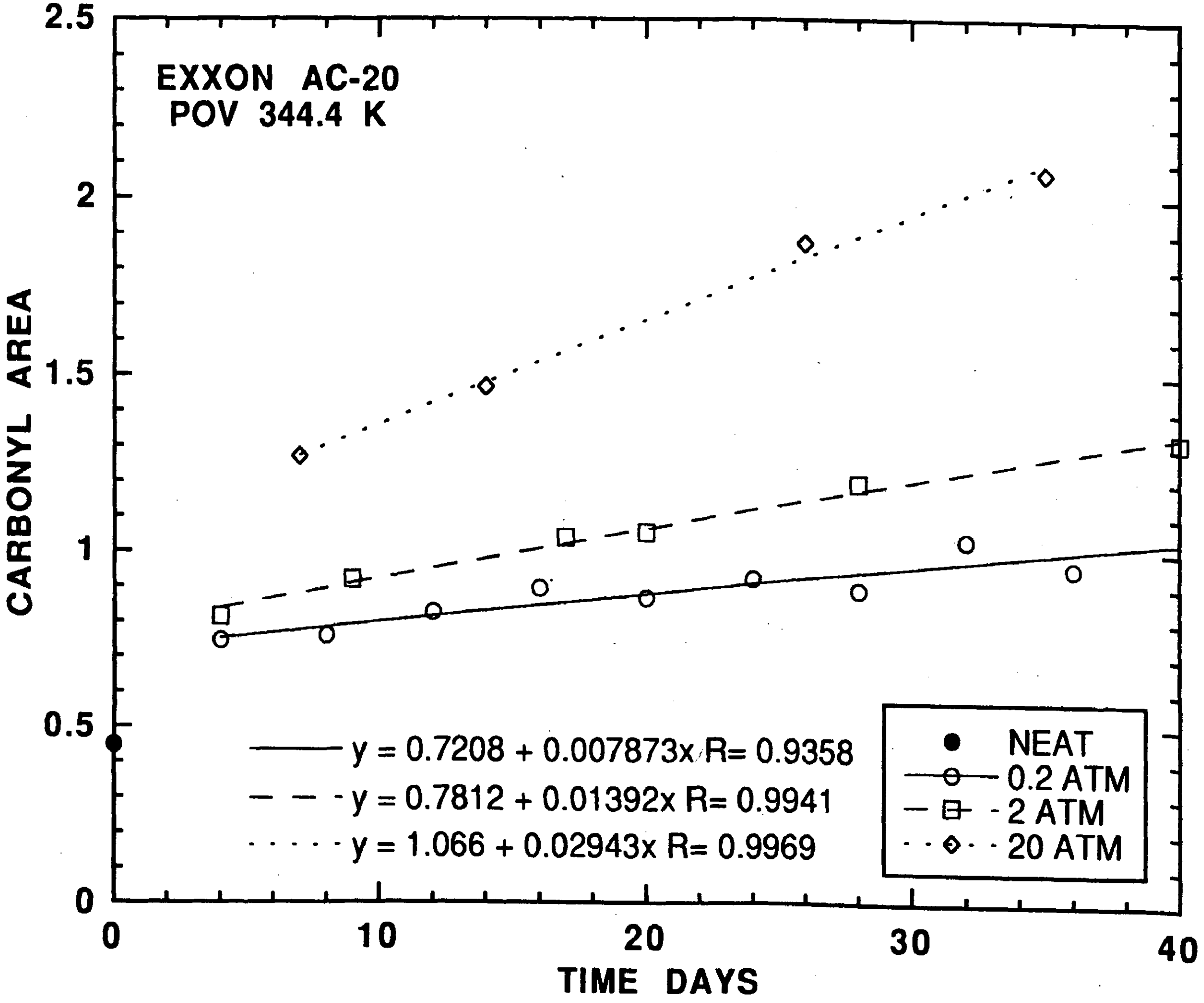

Figure B-8. *CA*s of neat and POV-aged Exxon AC-20 at 344.4 K and 0.2, 2, and 20 atm.

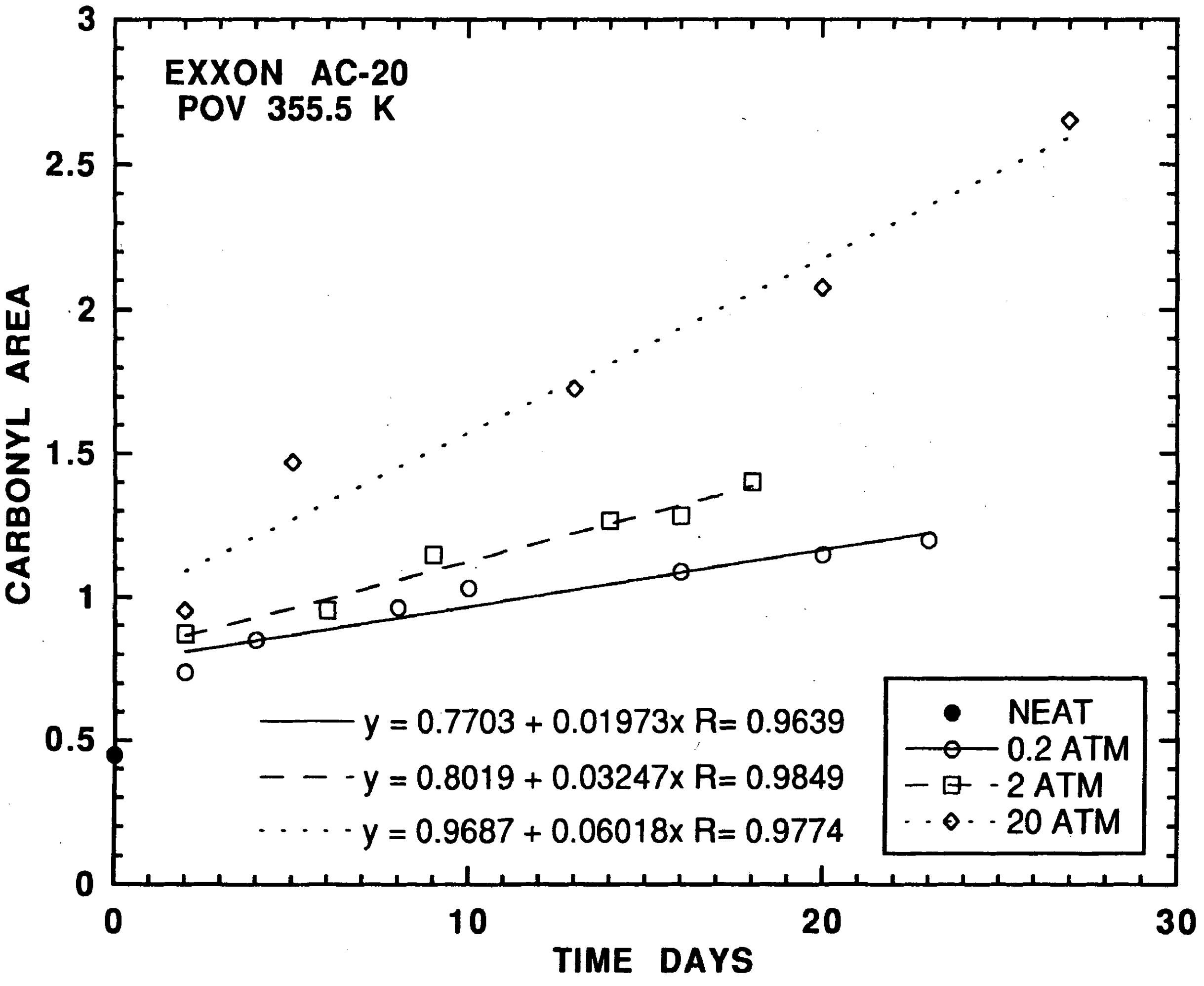

Figure B-9. *CA*s of neat and POV-aged Exxon AC-20 at 355.5 K and 0.2, 2, and 20 atm.

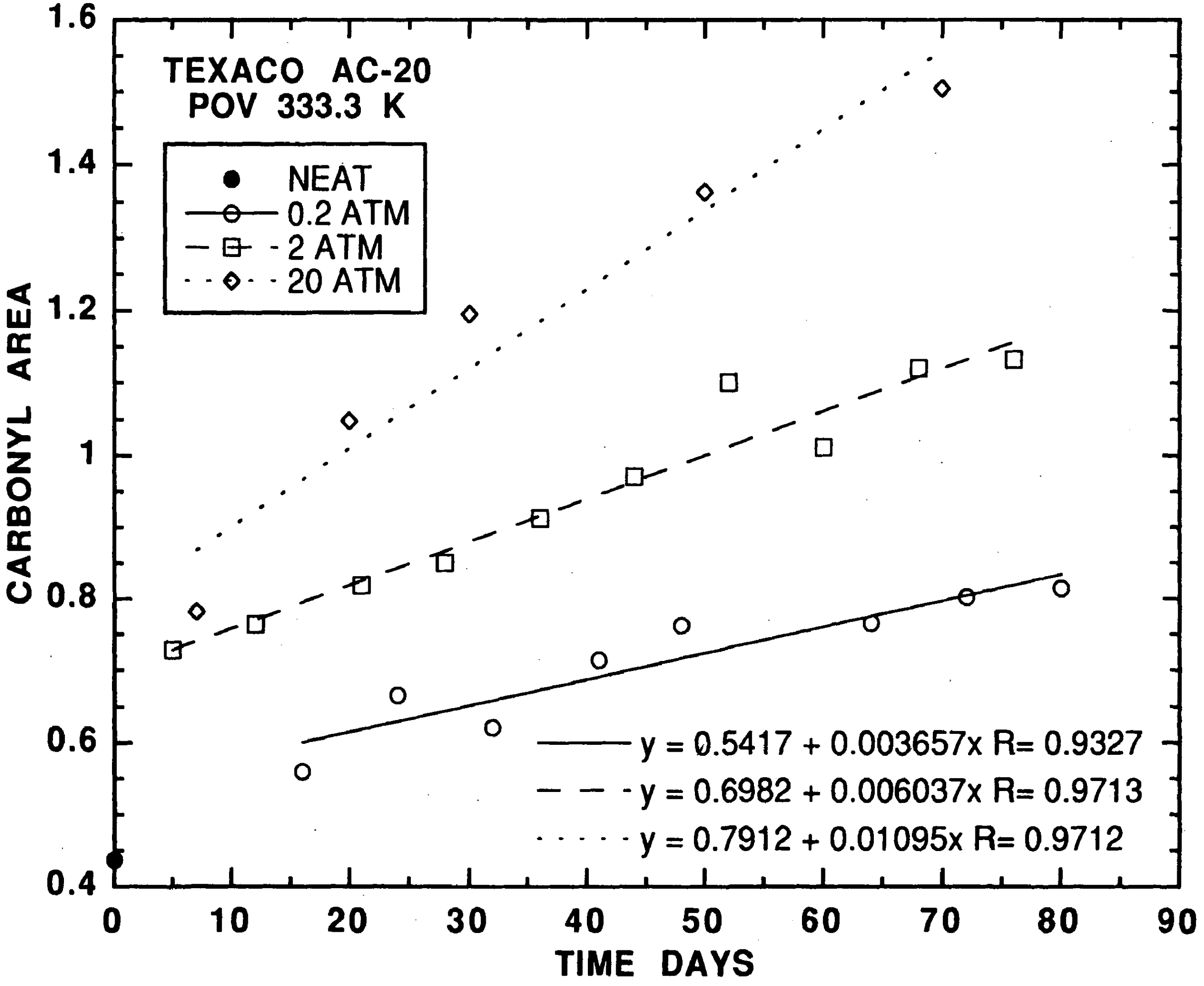

Figure B-10. *CA*s of neat and POV-aged Texaco AC-20 at 333.3 K and 0.2, 2, and 20 atm.

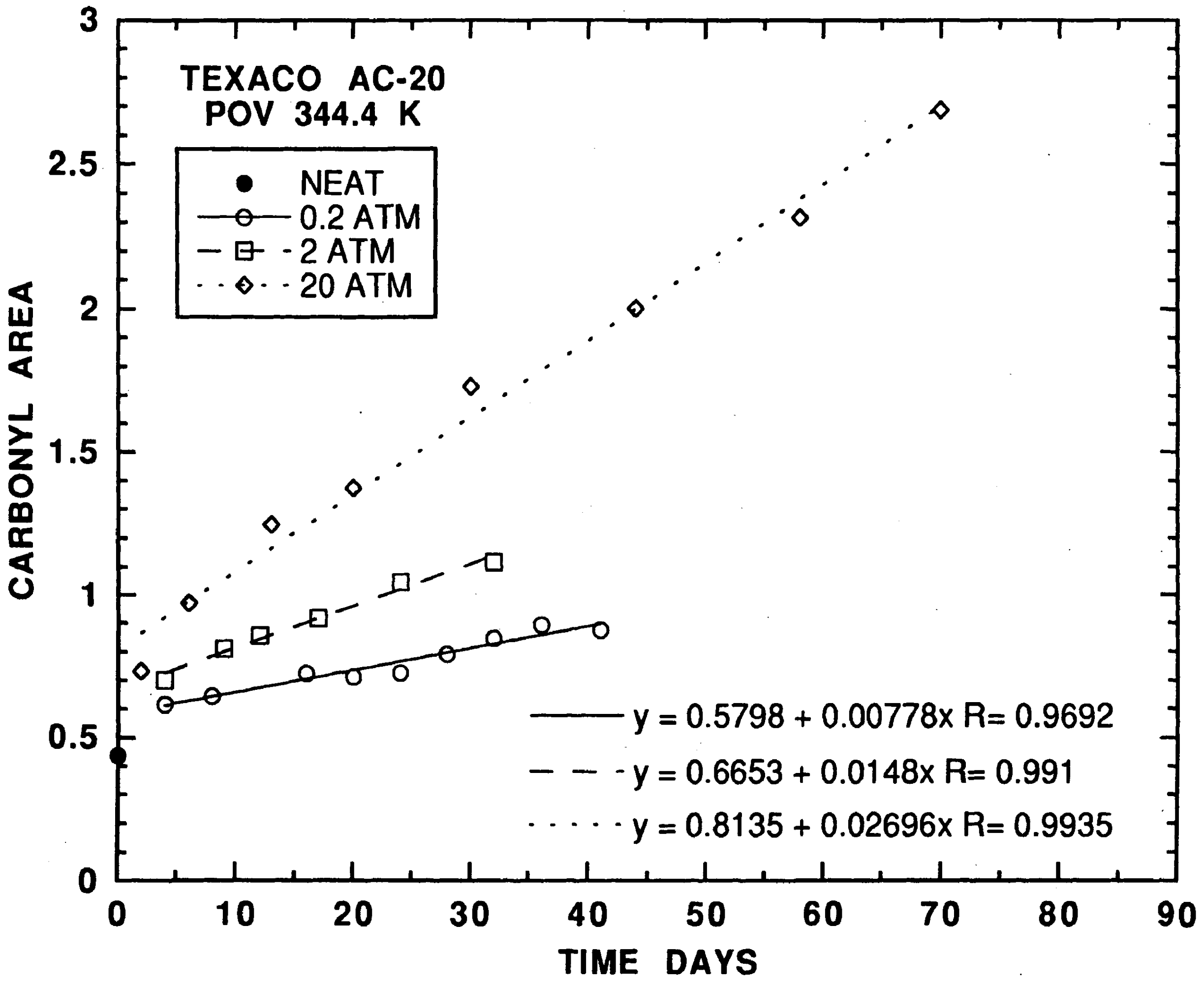

Figure B-11. *CA*s of neat and POV-aged Texaco AC-20 at 344.4 K and 0.2, 2, and 20 atm.

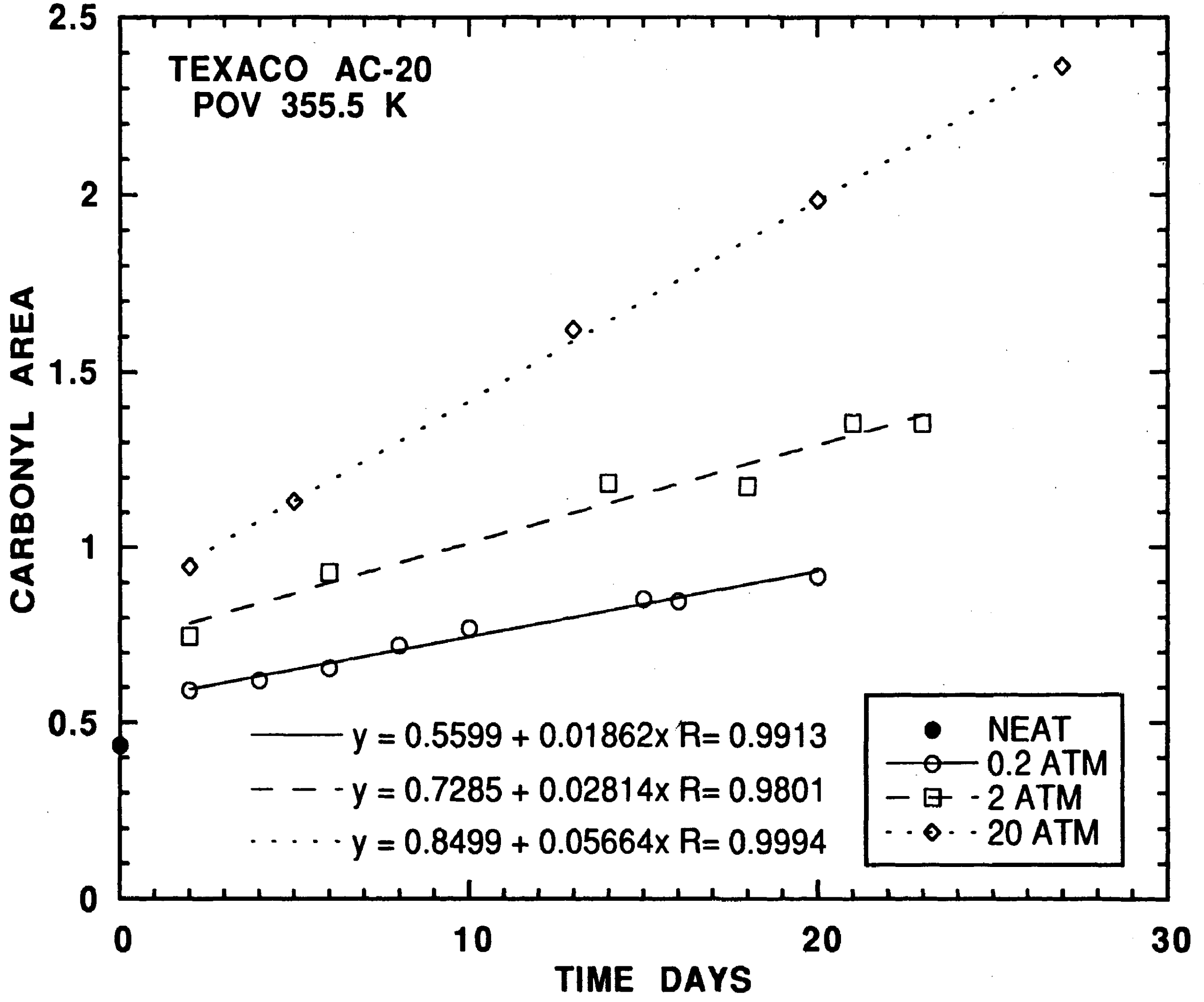

Figure B-12. *CA*s of neat and POV-aged Texaco AC-20 at 355.5 K and 0.2, 2, and 20 atm.

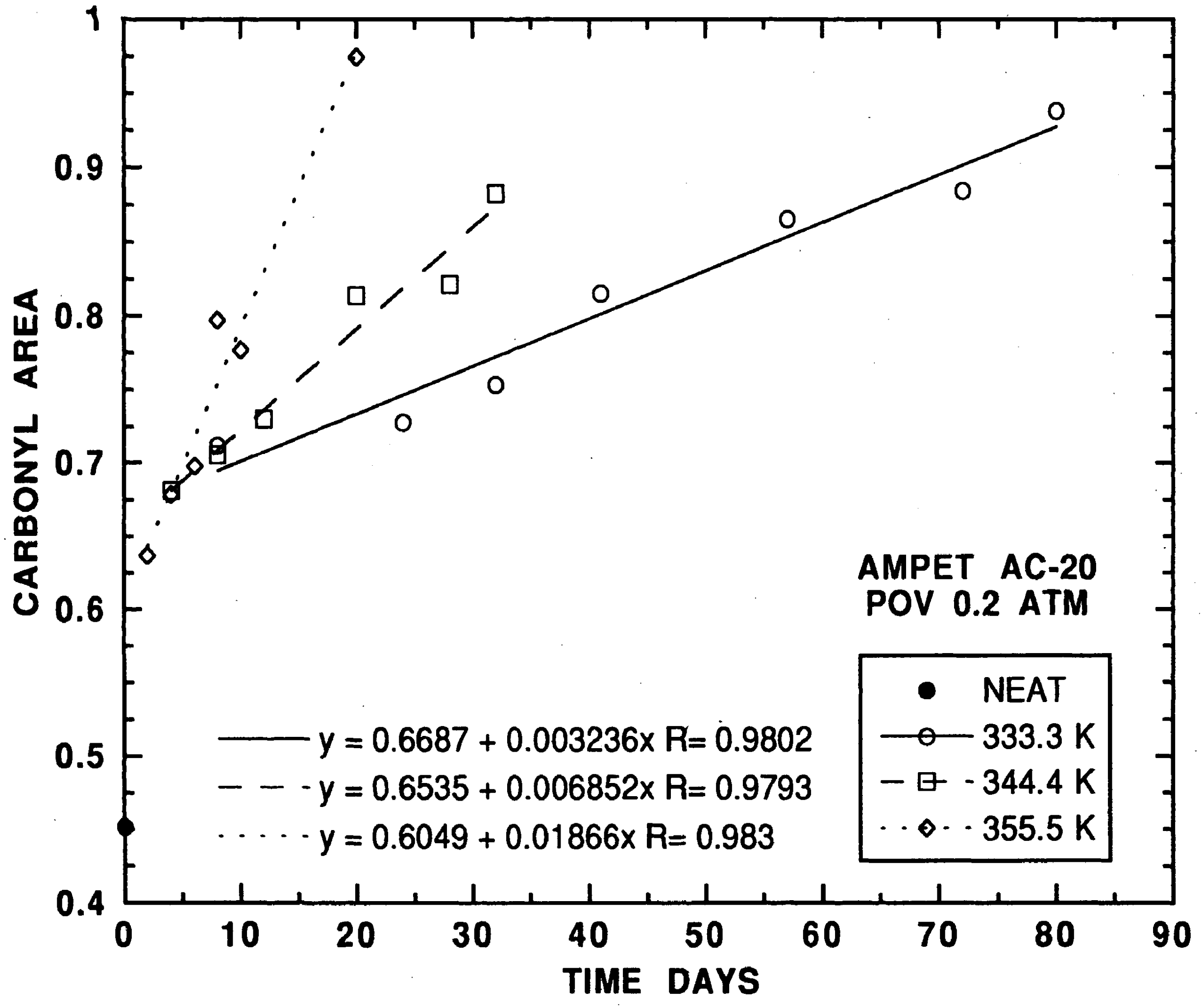

Figure B-13. *CA*s of neat and POV-aged Ampet AC-20 at 0.2 atm and 333.3, 344.4, and 355.5 K.

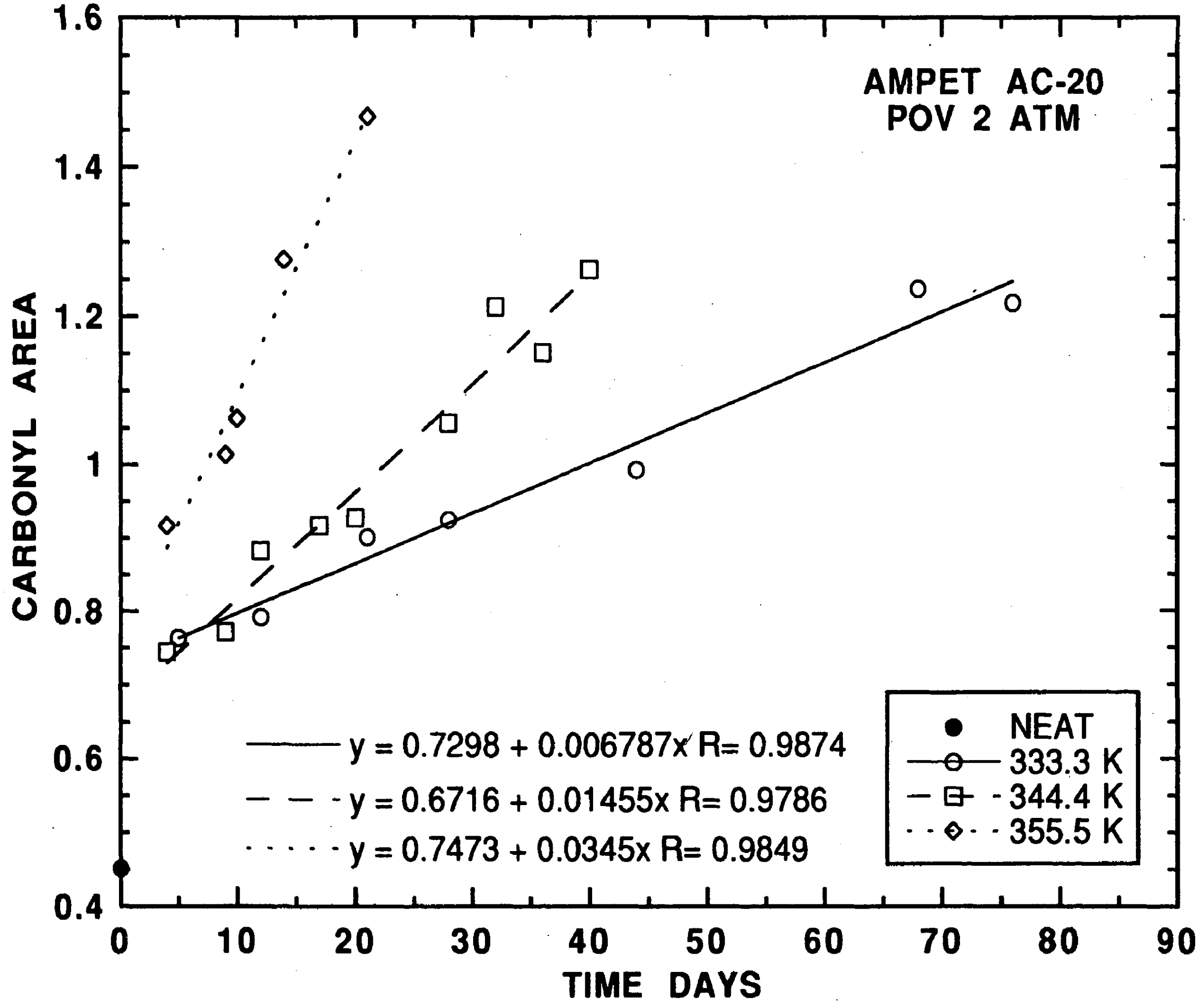

Figure B-14. *CA*s of neat and POV-aged Ampet AC-20 at 2 atm and 333.3, 344.4, and 355.5 K.

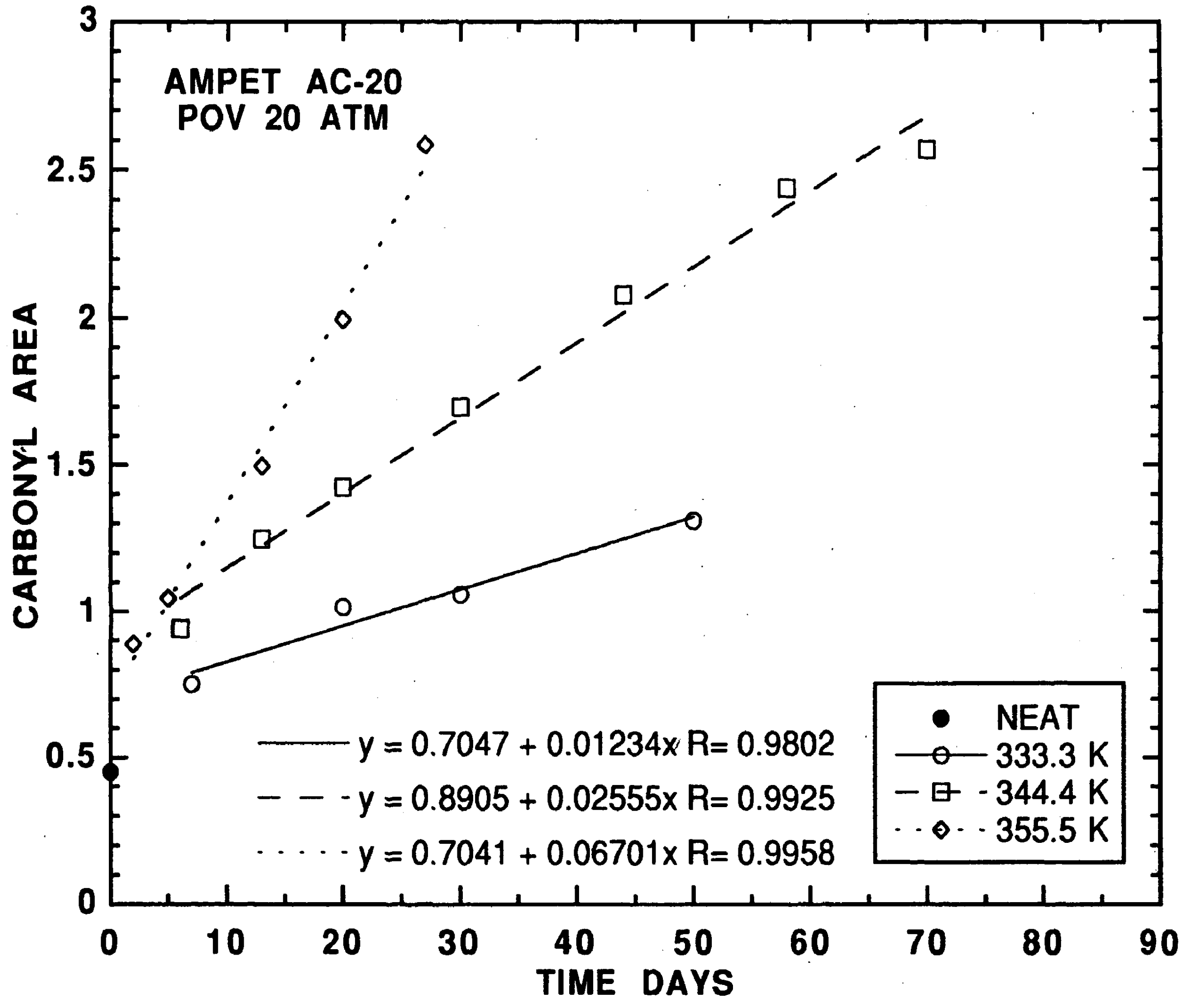

Figure B-15. *CA*s of neat and POV-aged Ampet AC-20 at 20 atm and 333.3, 344.4, and 355.5 K.

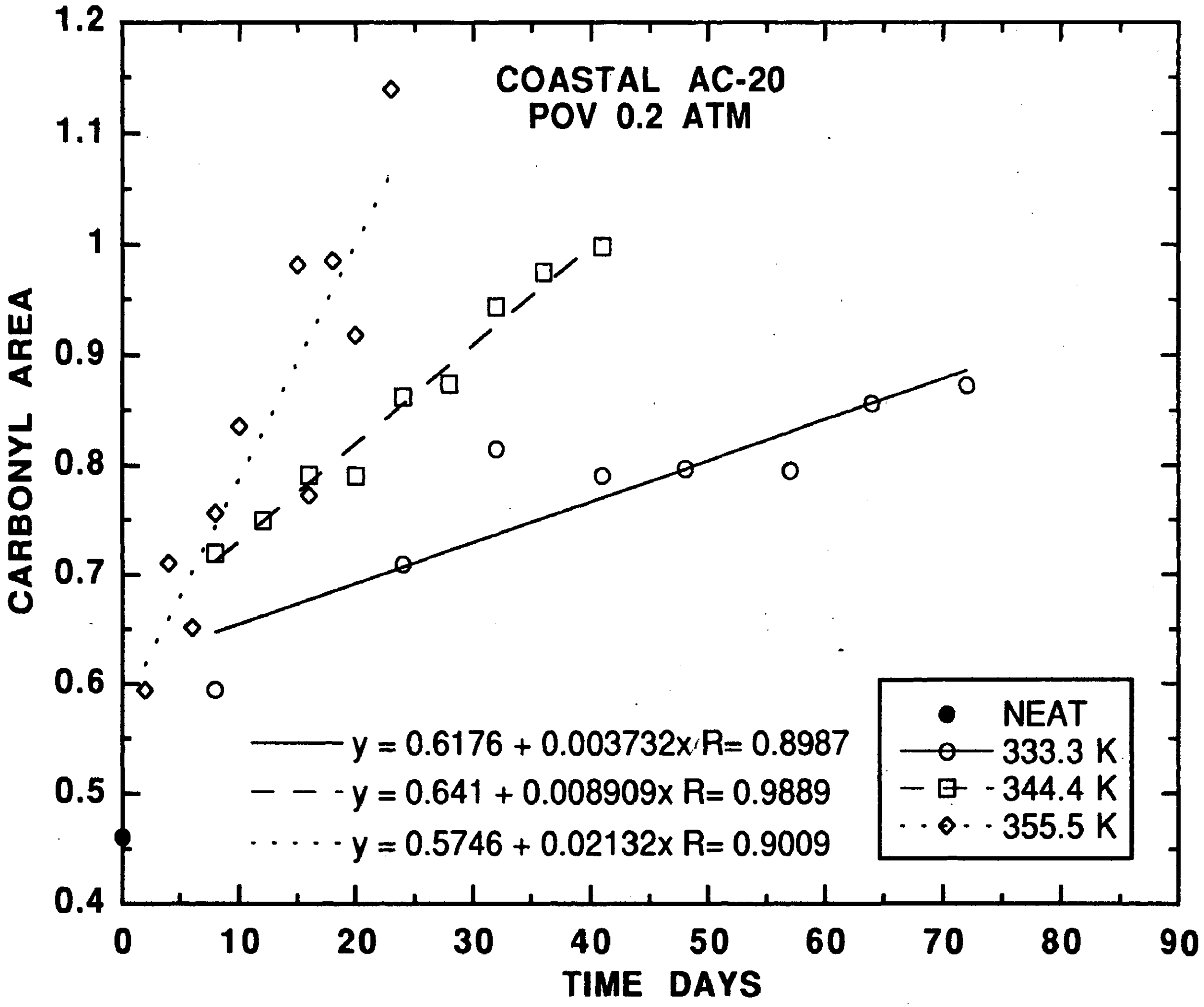

Figure B-16. *CA*s of neat and POV-aged Coastal AC-20 at 0.2 atm and 333.3, 344.4, and 355.5 K.

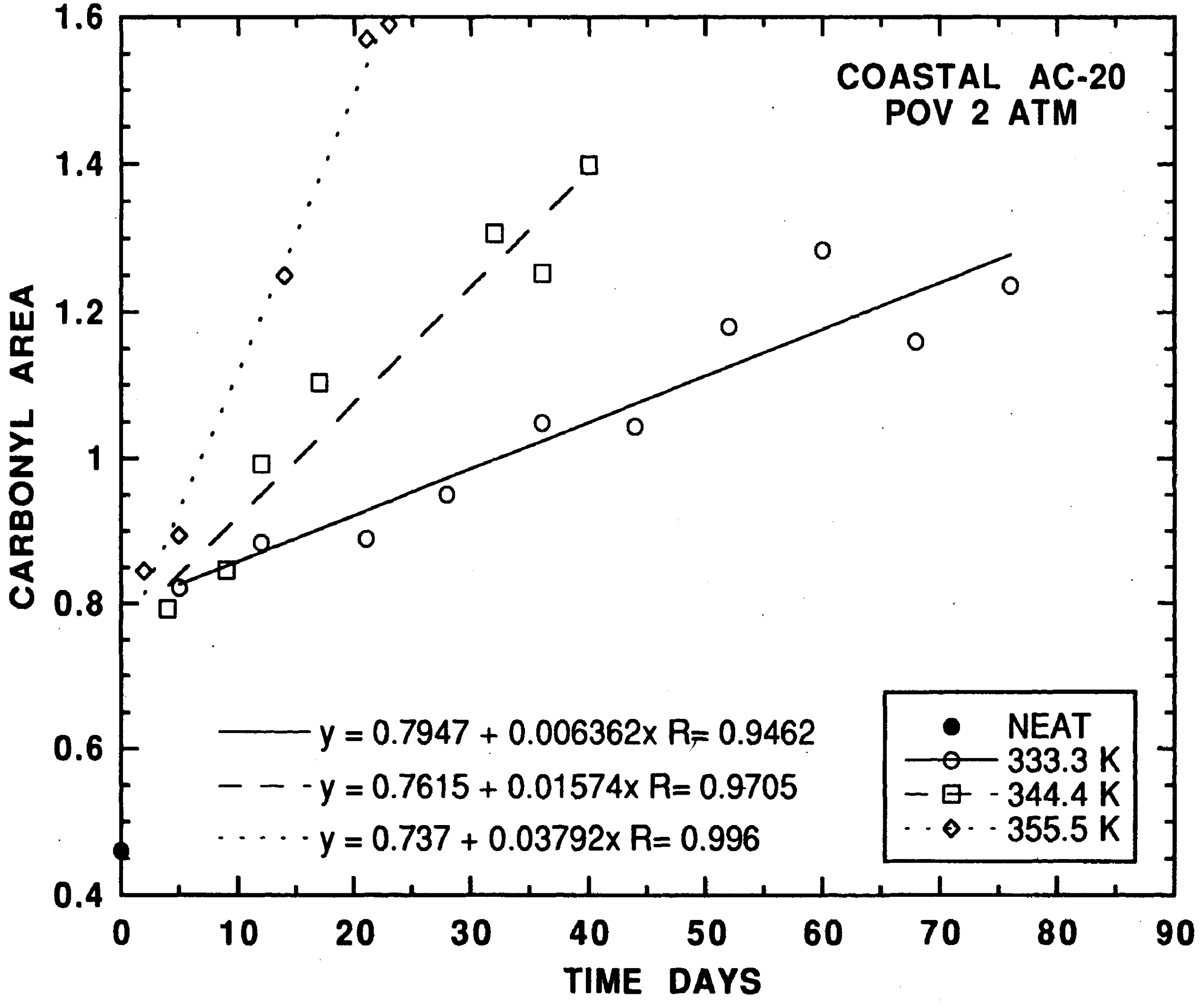

Figure B-17. *CA*s of neat and POV-aged Coastal AC-20 at 2 atm and 333.3, 344.4, and 355.5 K.

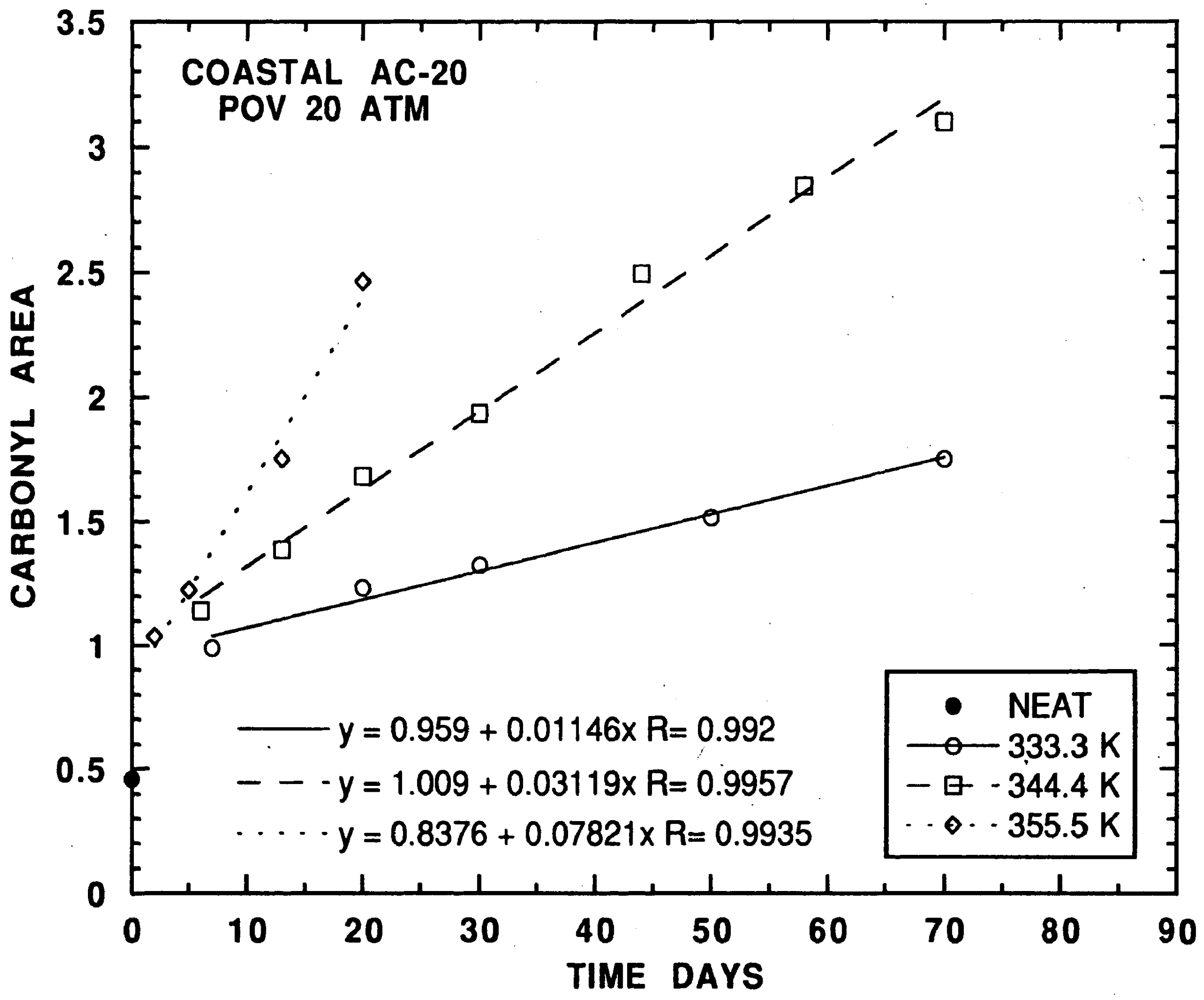

Figure B-18. *CA*s of neat and POV-aged Coastal AC-20 at 20 atm and 333.3, 344.4, and 355.5 K.

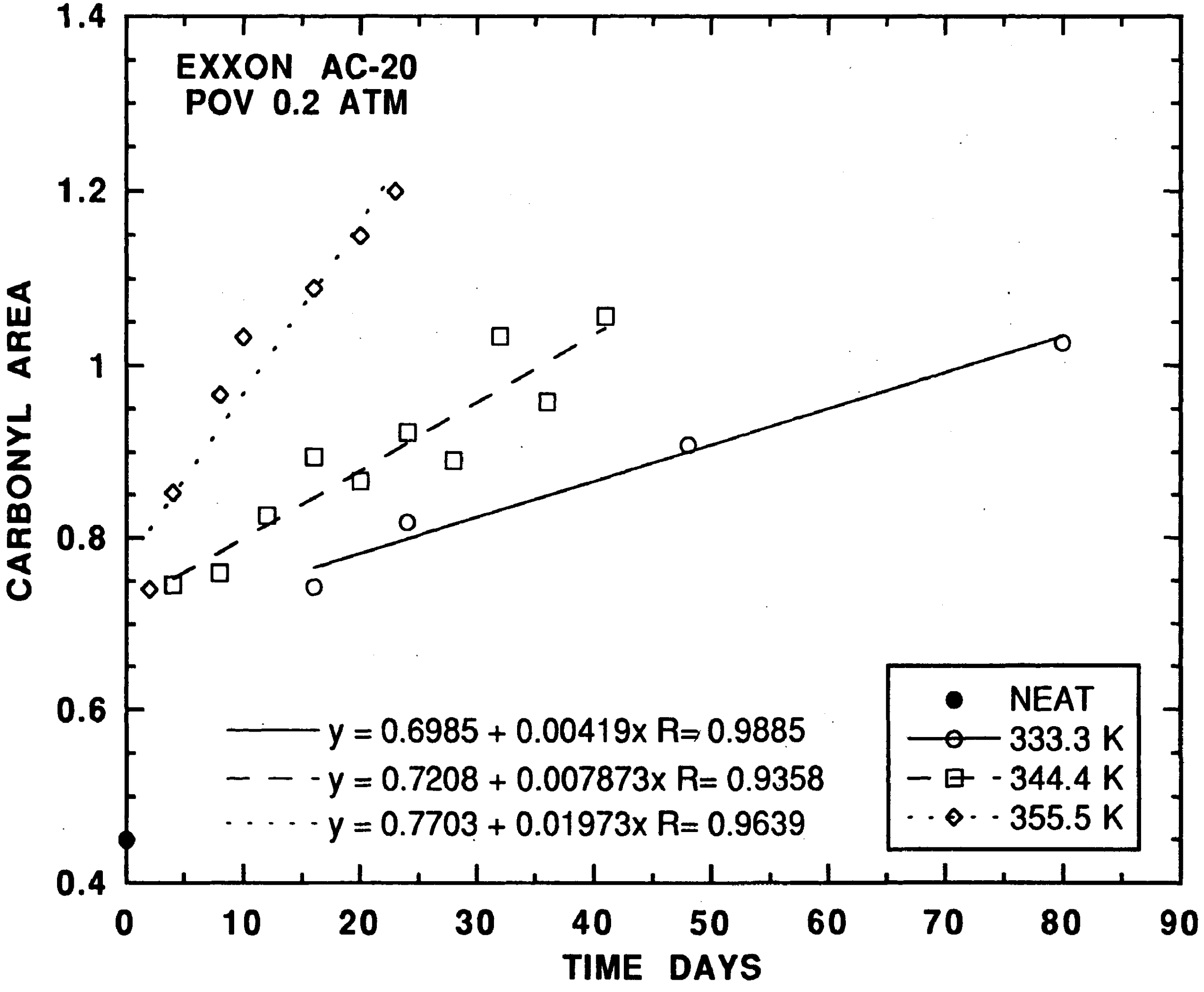

Figure B-19. *CA*s of neat and POV-aged Exxon AC-20 at 0.2 atm and 333.3, 344.4, and 355.5 K.

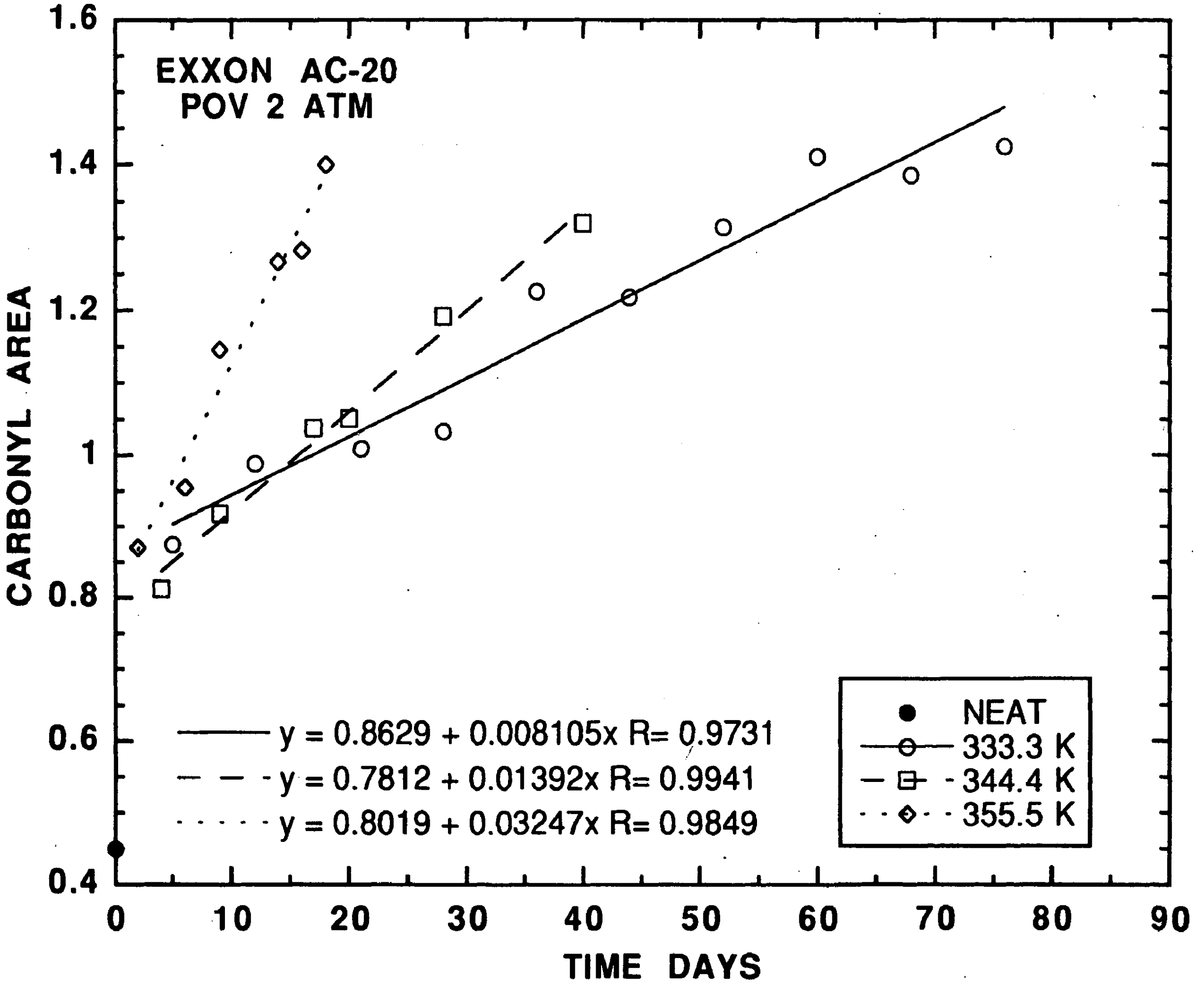

Figure B-20. *CA*s of neat and POV-aged Exxon AC-20 at 2 atm and 333.3, 344.4, and 355.5 K.

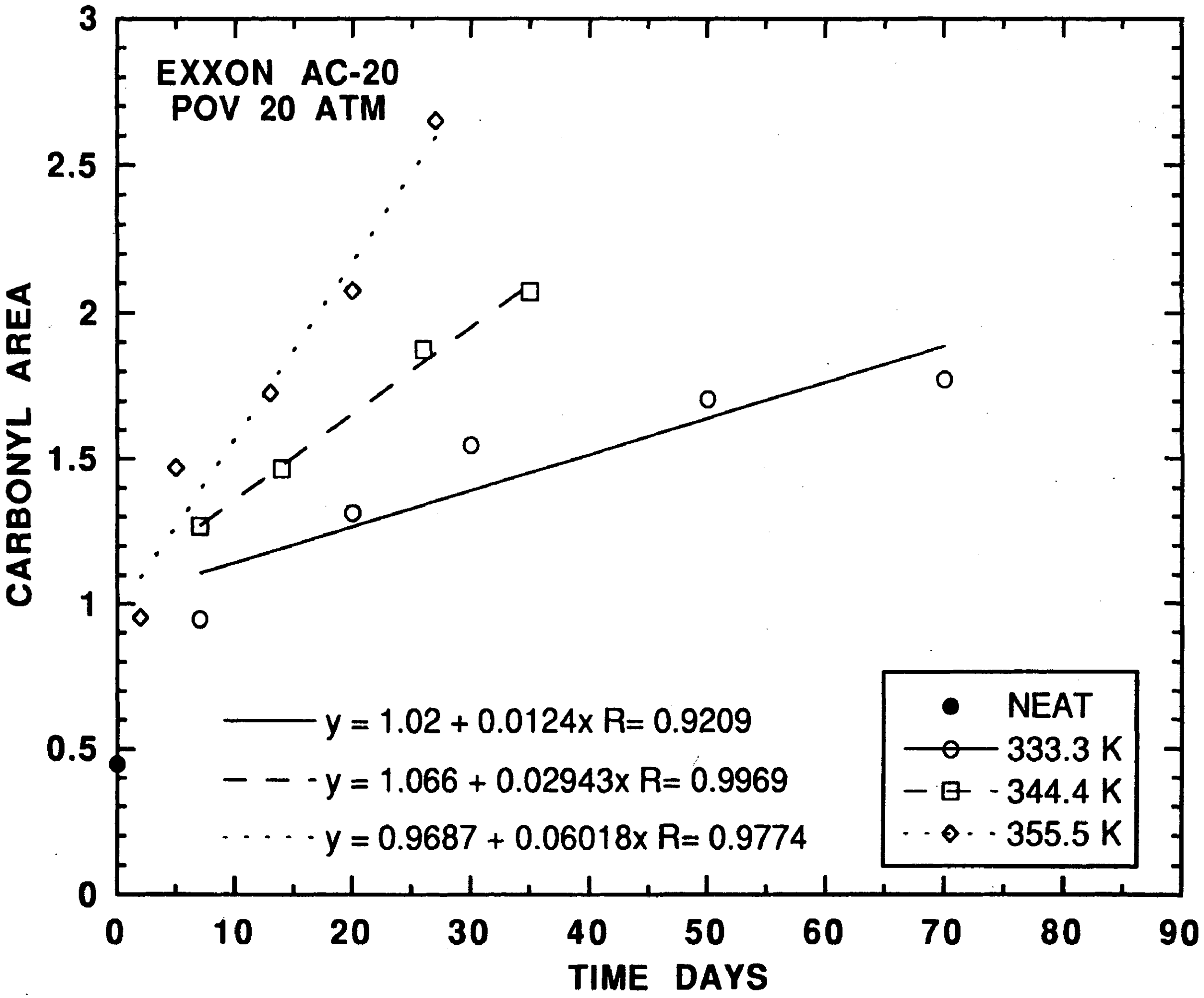

Figure B-21. *CA*s of neat and POV-aged Exxon AC-20 at 20 atm and 333.3, 344.4, and 355.5 K.

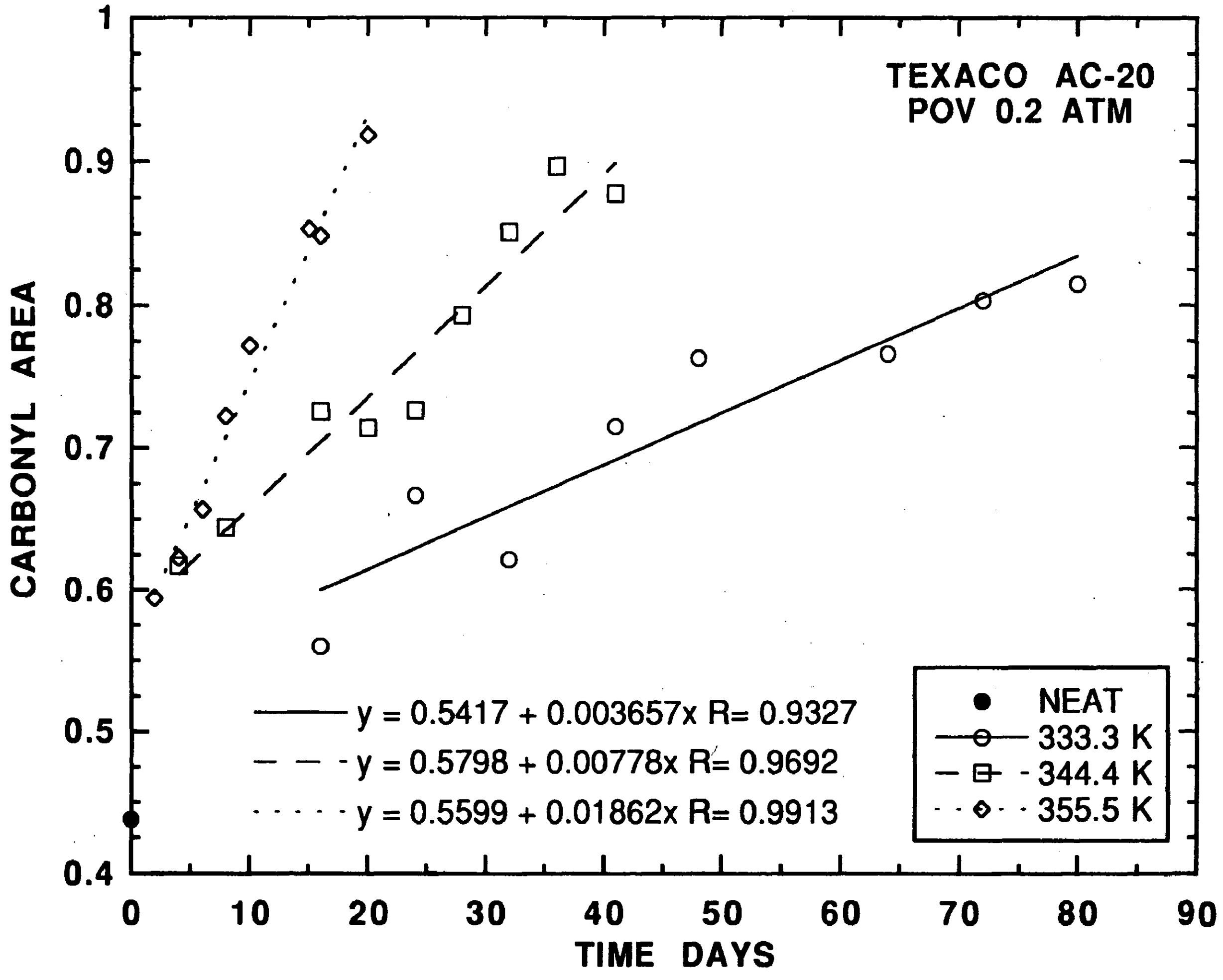

Figure B-22. *CA*s of neat and POV-aged Texaco AC-20 at 0.2 atm and 333.3, 344.4, and 355.5 K.

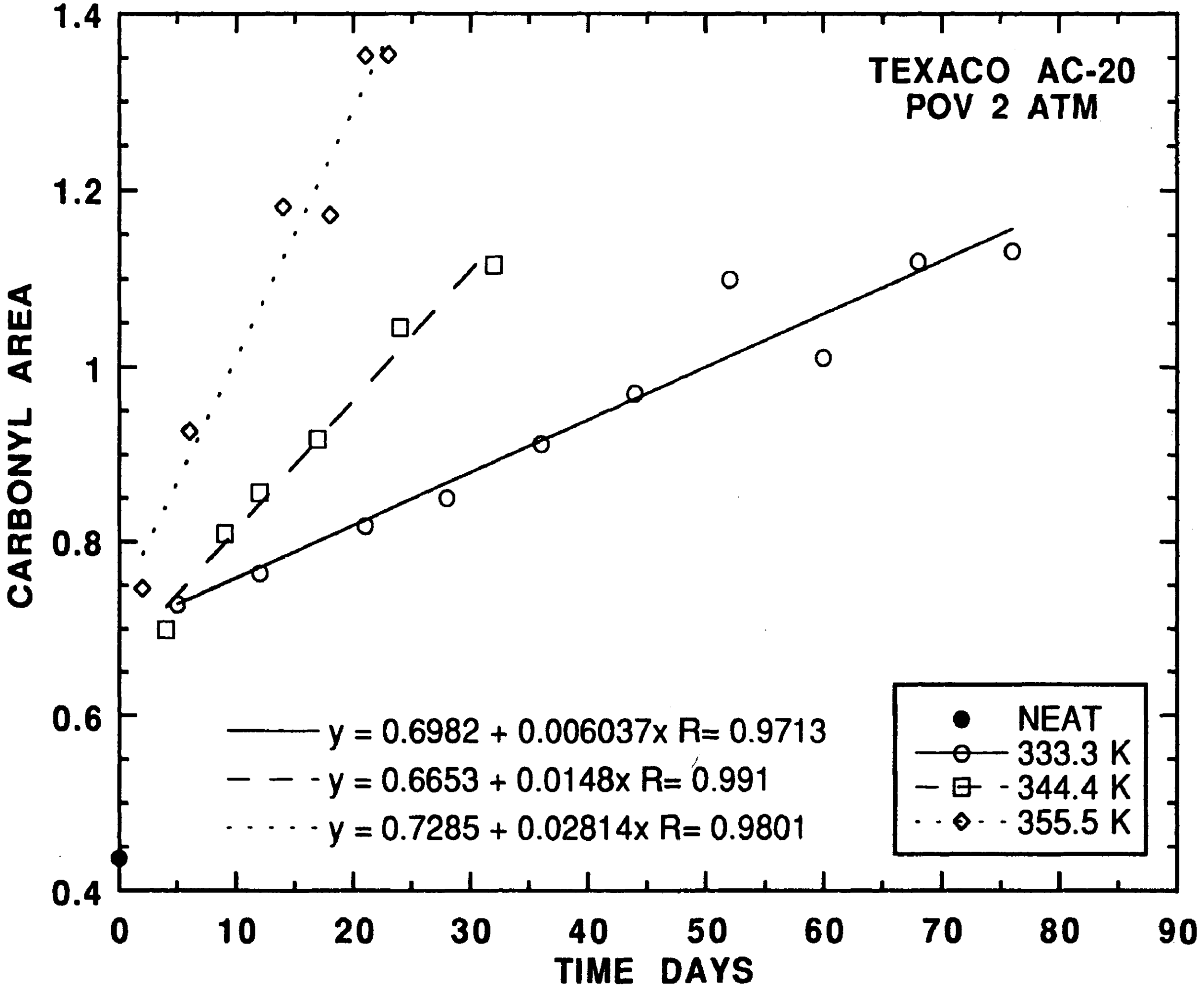

Figure B-23. *CA*s of neat and POV-aged Texaco AC-20 at 2 atm and 333.3, 344.4, and 355.5 K.

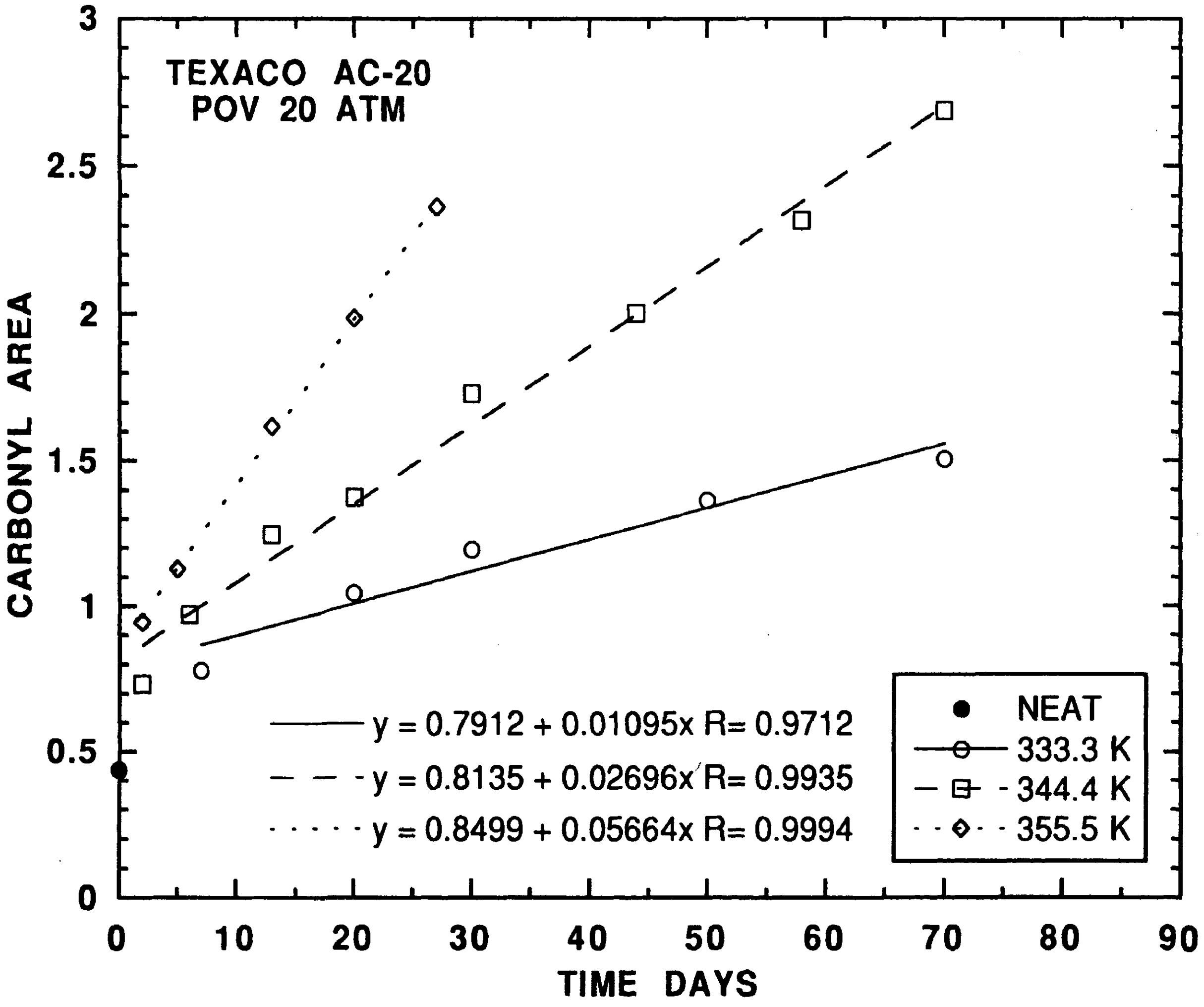

Figure B-24. *CA*s of neat and POV-aged Texaco AC-20 at 20 atm and 333.3, 344.4, and 355.5 K.

Table B-6. Kinetic Data for Asphalts Aged at 322.2 K and 20 atm[a]

	AC-20 Asphalts				
t days	Ampet CA	Coastal CA	Cosden CA	Exxon CA	Texaco CA
7	0.752	0.746	0.882	0.832	0.710
21	0.876	0.887	1.042	0.973	0.785
35	0.965	0.940	1.178	1.110	0.855
54	1.081	1.018	1.282	1.199	0.935
72	1.110	1.058	1.311	1.268	–
82	1.144	1.114	1.367	1.292	–

[a] – Signifies the values were not determined.
Bulk analysis

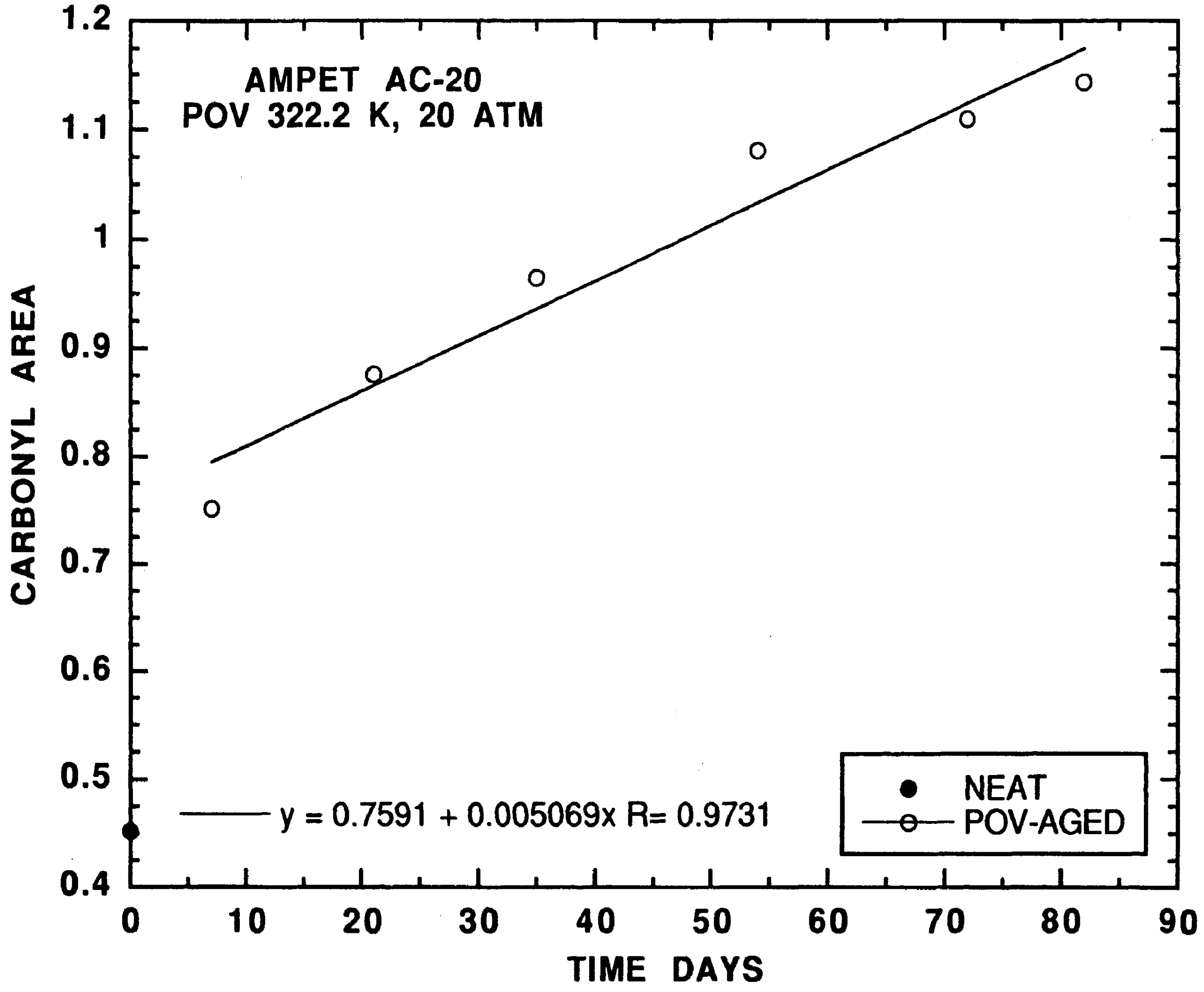

Figure B-25. *CA*s of neat and POV-aged Ampet AC-20 at 20 atm and 322.2 K.

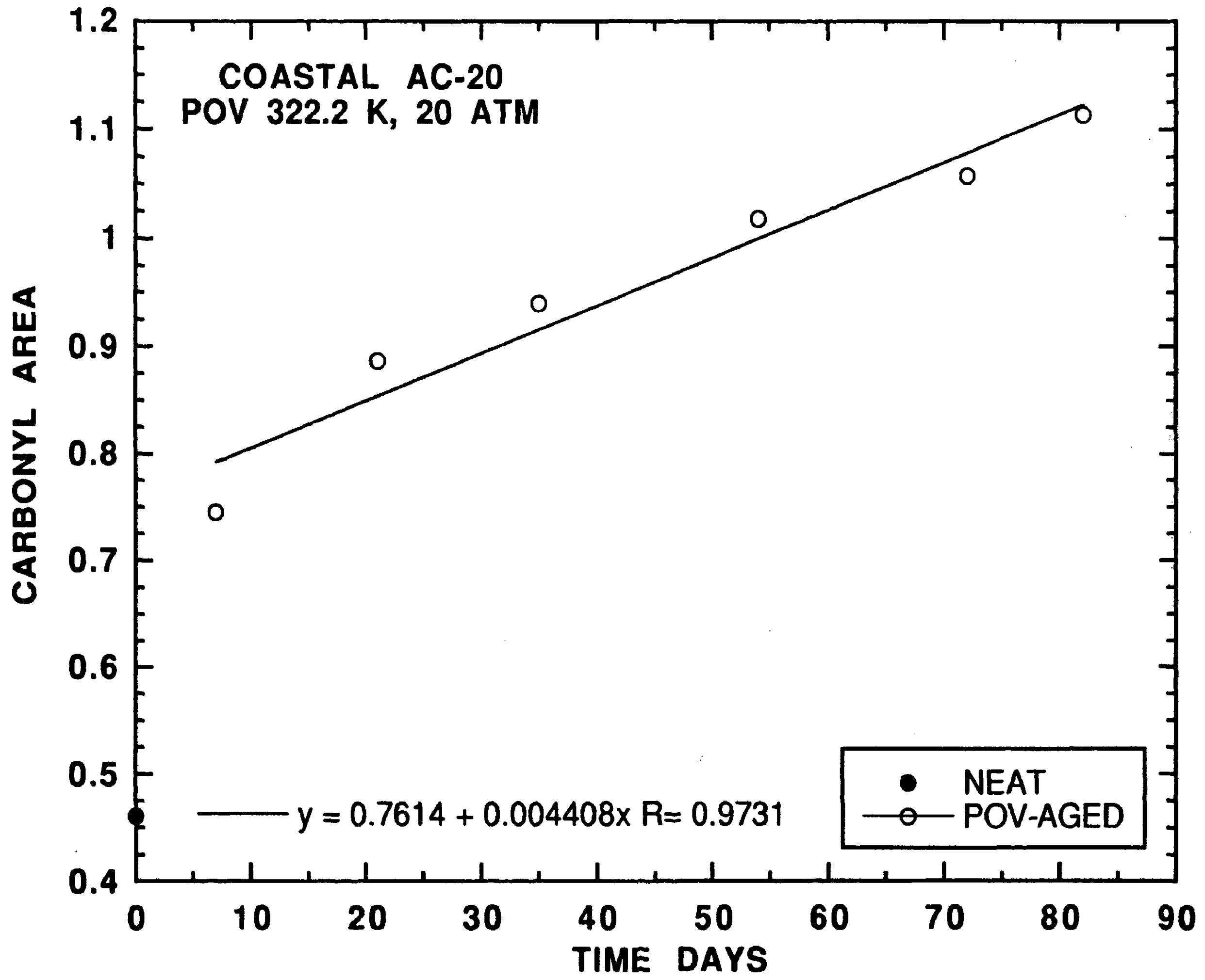

Figure B-26. *CA*s of neat and POV-aged Coastal AC-20 at 20 atm and 322.2 K.

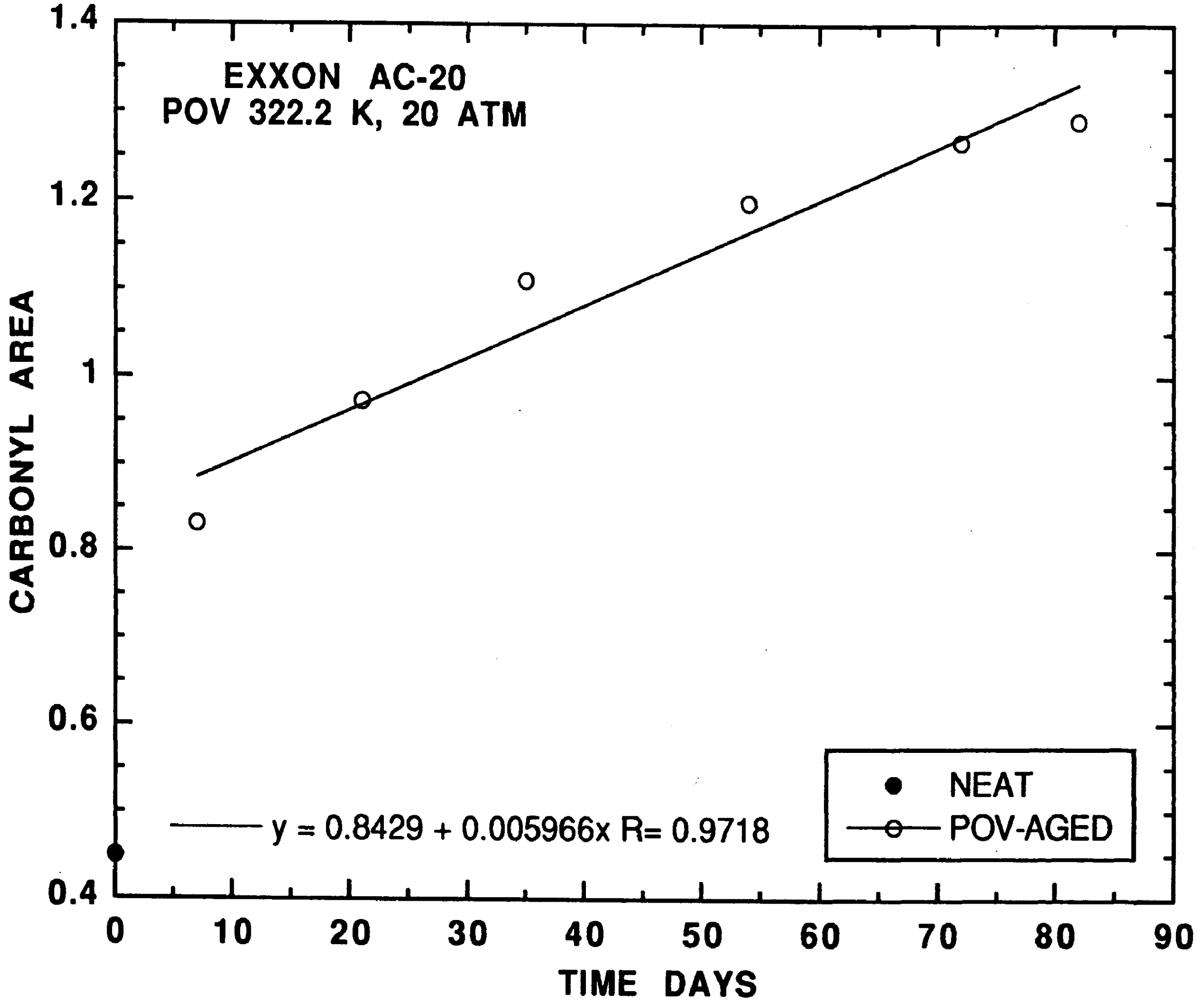

Figure B-27. *CA*s of neat and POV-aged Exxon AC-20 at 20 atm and 322.2 K.

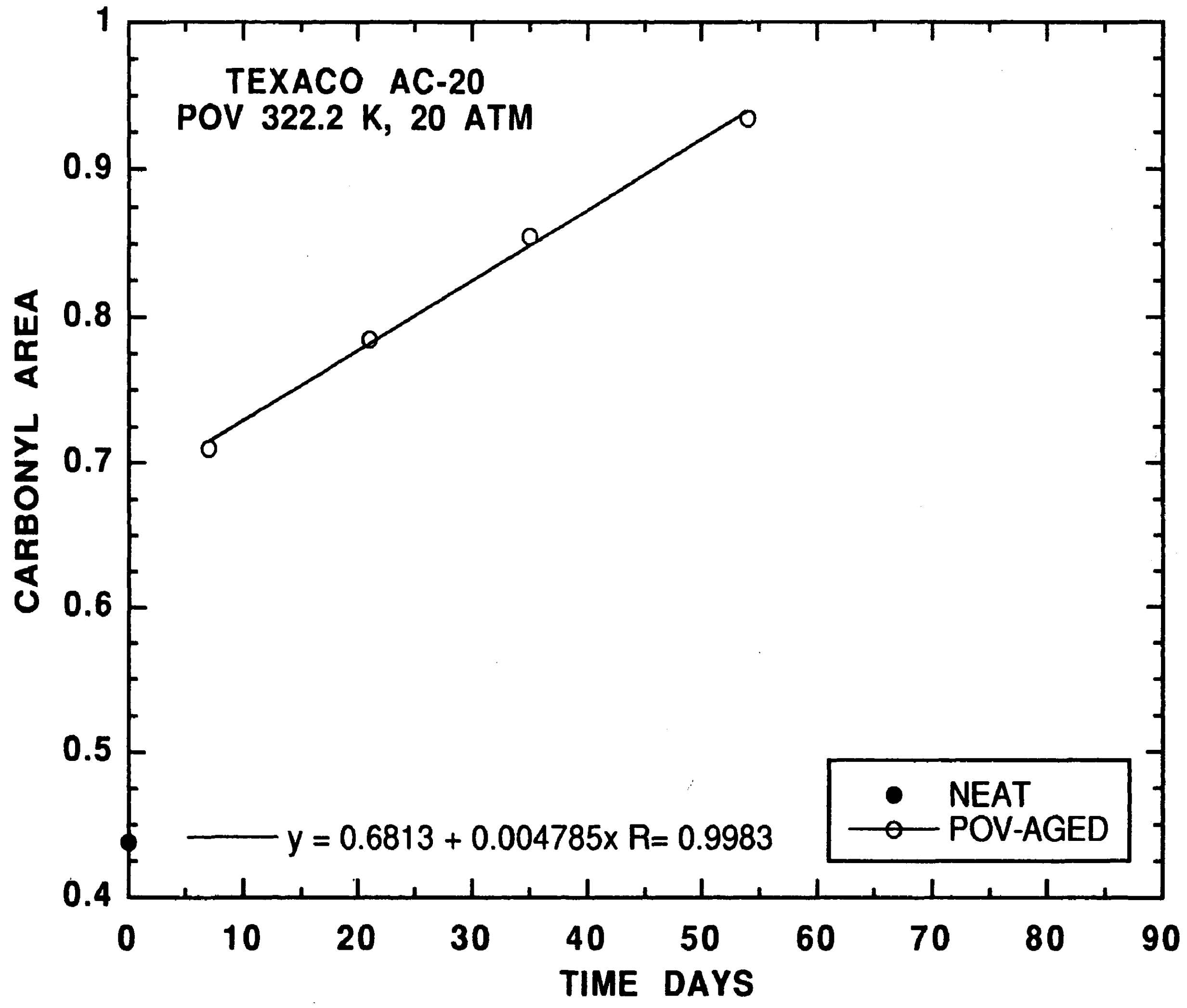

Figure B-28. *CA*s of neat and POV-aged Texaco AC-20 at 20 atm and 322.2 K.

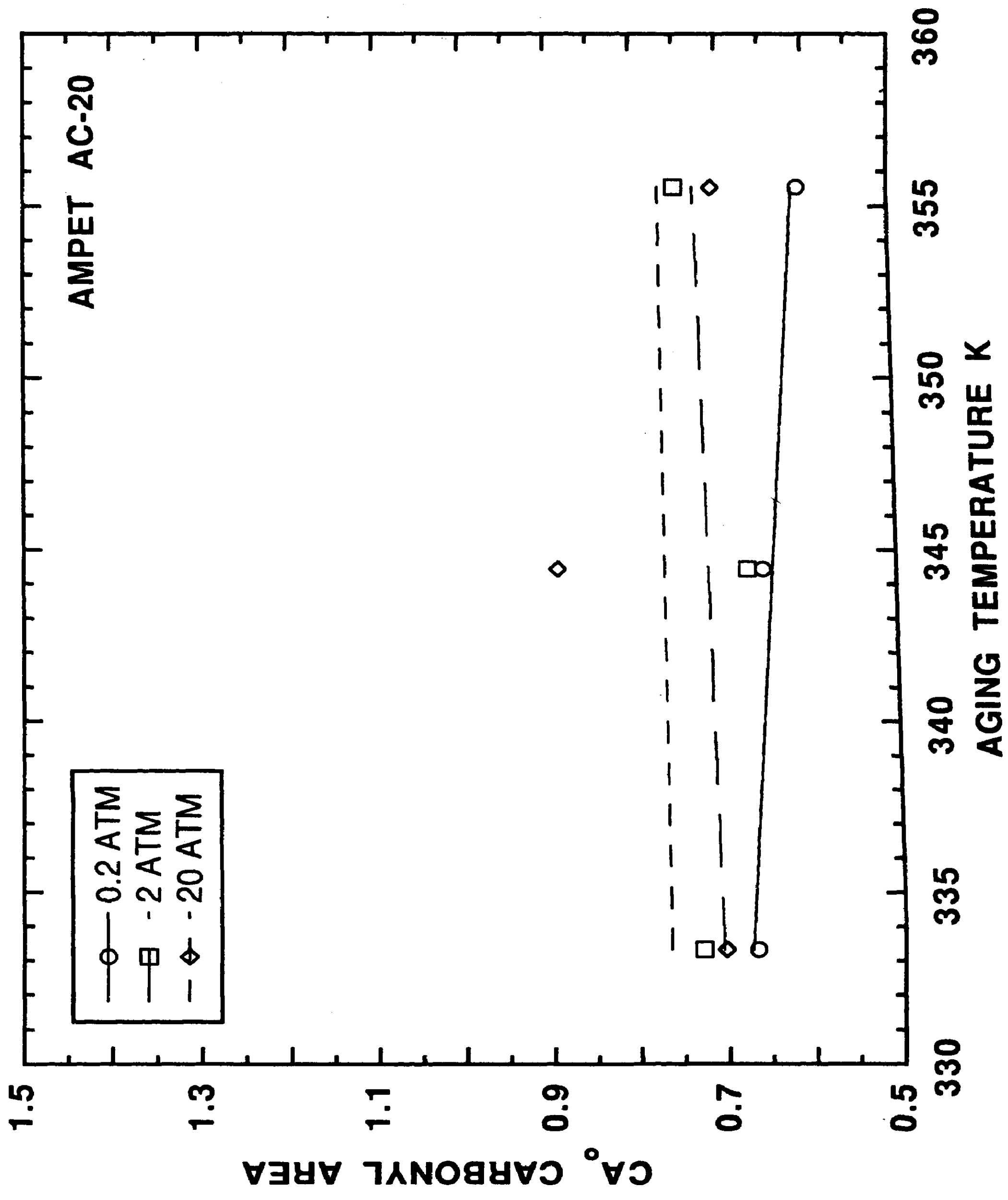

Figure B-29. CA_0 versus POV aging temperature for Ampet AC-20.

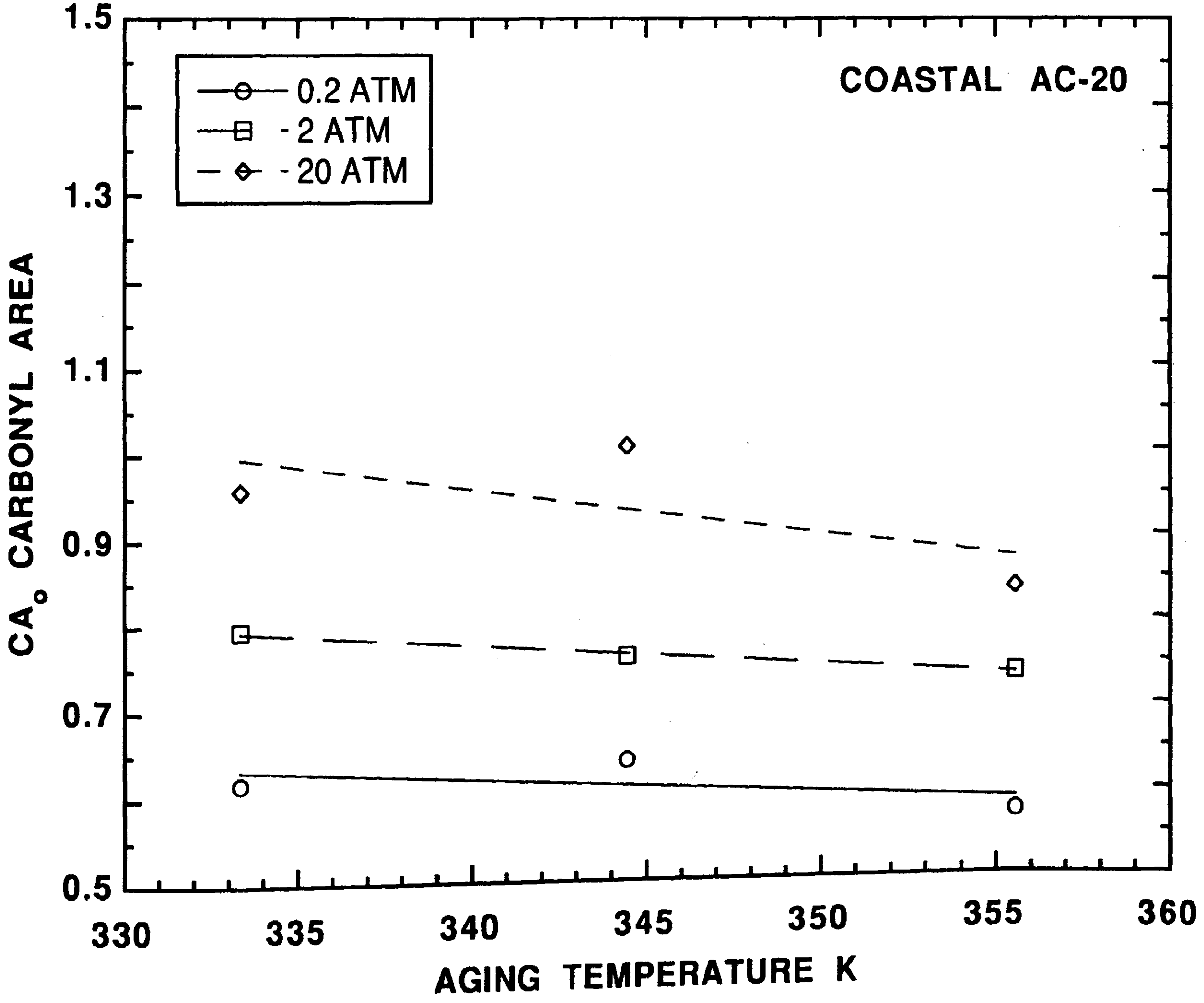

Figure B-30. CA_0 versus POV aging temperature for Coastal AC-20.

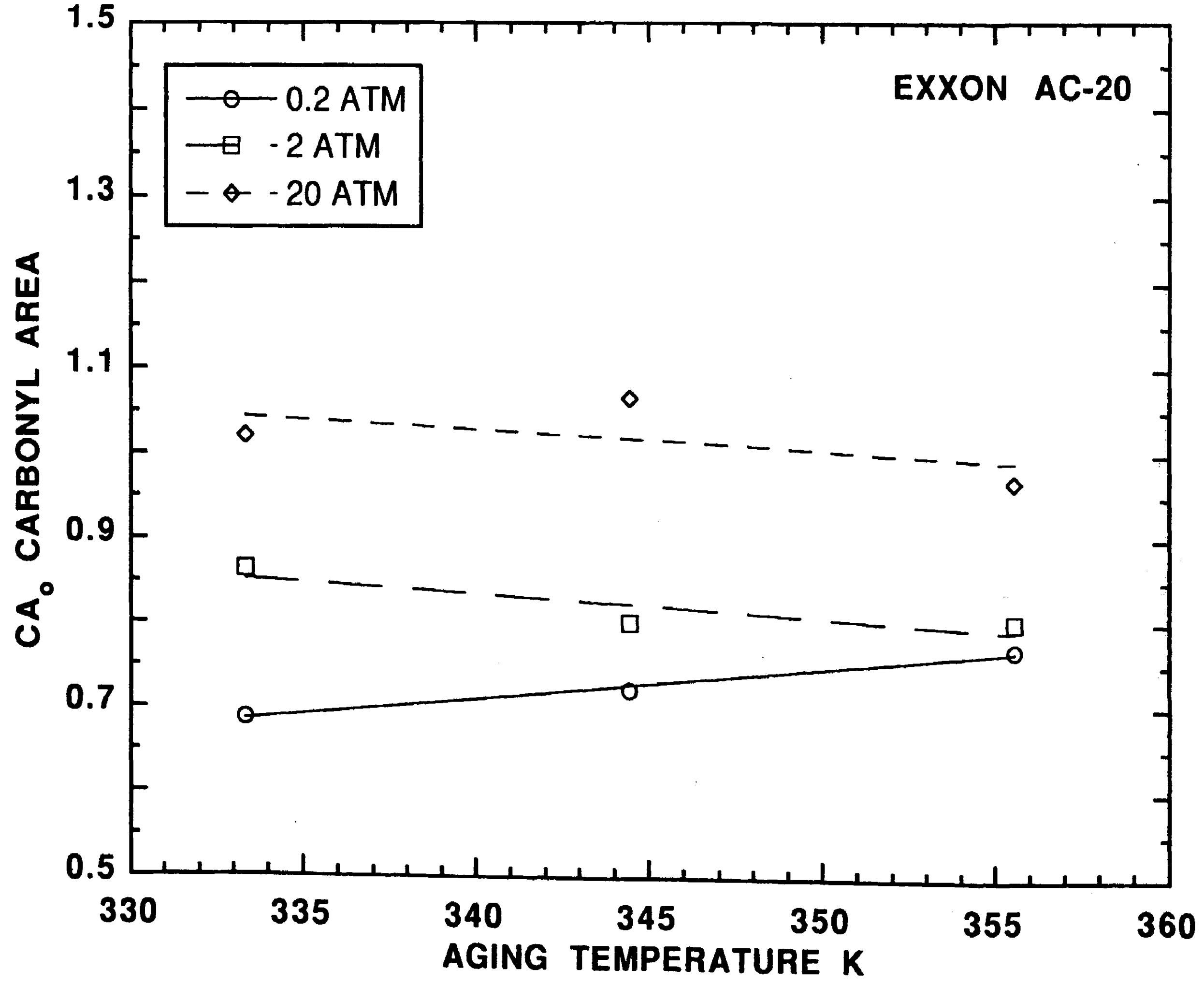

Figure B-31. CA_o versus POV aging temperature for Exxon AC-20.

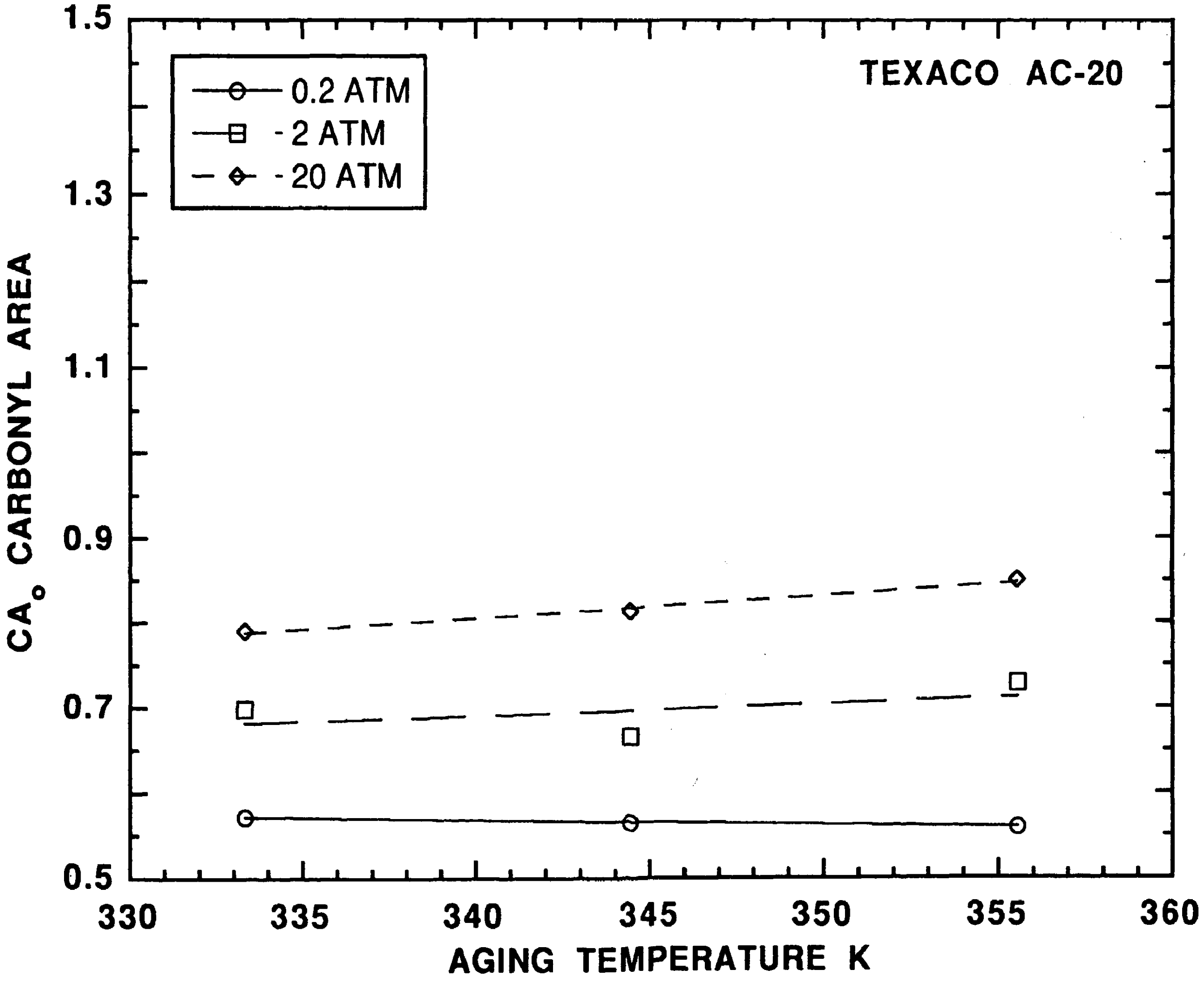

Figure B-32. CA_o versus POV aging temperature for Texaco AC-20.

APPENDIX C

OXYGEN DIFFUSION DATA

Table C-1. *CA* and a_T Neat and Aged Asphalts[a]

Asphalt	Aging	CA	Shift Factor			
			$a_T(273.2)$	$a_T(283.2)$ $\times 10^2$	$a_T(298.2)$ $\times 10^4$	$a_T(313.2)$ $\times 10^5$
Ampet AC-20	Neat	0.452	1.0	5.11	13.7	3.74
	RTFO	0.581	1.0	6.09	10.8	3.04
	2 d	0.890	1.0	4.28	7.76	1.69
	5 d	1.046	1.0	5.22	9.26	1.90
	13 d	1.496	1.0	4.10	6.65	1.21
Coastal AC-20	Neat	0.461	1.0	5.34	12.3	3.96
	RTFO	0.663	1.0	6.23	11.3	3.42
	2 d	1.037	1.0	3.29	5.22	1.23
	5 d	1.224	1.0	5.00	8.29	1.90
	13 d	1.754	1.0	5.28	7.11	1.32
Cosden AC-20	Neat	0.462	1.0	5.83	11.0	2.87
	RTFO	0.637	1.0	5.11	8.96	2.09
	2 d	1.290	1.0	9.68	15.7	2.87
	5 d	1.688	1.0	6.09	8.11	1.26
	13 d	2.385	1.0	0.73	2.82	0.36
Exxon AC-20	Neat	0.450	1.0	6.95	16.1	5.64
	RTFO	0.523	1.0	6.36	12.9	3.74
	2 d	0.953	1.0	5.11	9.06	2.27
	5 d	1.471	1.0	5.11	8.86	1.79
	13 d	1.727	1.0	5.11	8.86	1.64
Texaco AC-20	Neat	0.438	1.0	7.26	20.0	7.68
	RTFO	0.595	1.0	6.09	14.1	4.87
	2 d	0.946	1.0	4.68	8.67	2.63
	5 d	1.130	1.0	5.70	9.26	2.29
	13 d	1.620	1.0	5.83	6.95	1.42

[a] POV conditions 355.5 K, 20 atm

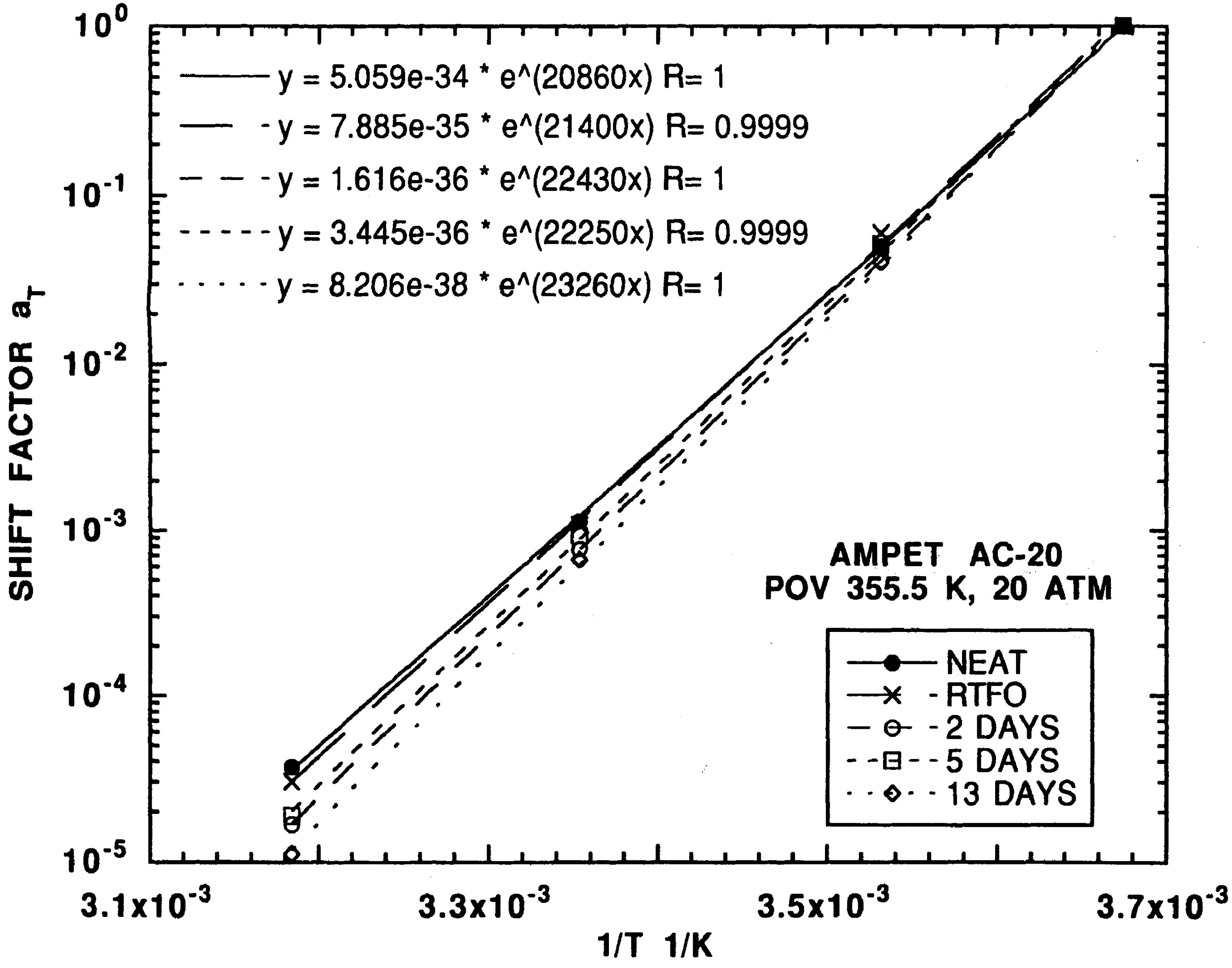

Figure C-1. a_T versus $(1 / T)$ for neat, RTFO, and POV-aged Ampet AC-20 with reference temperature of 273.2 K.

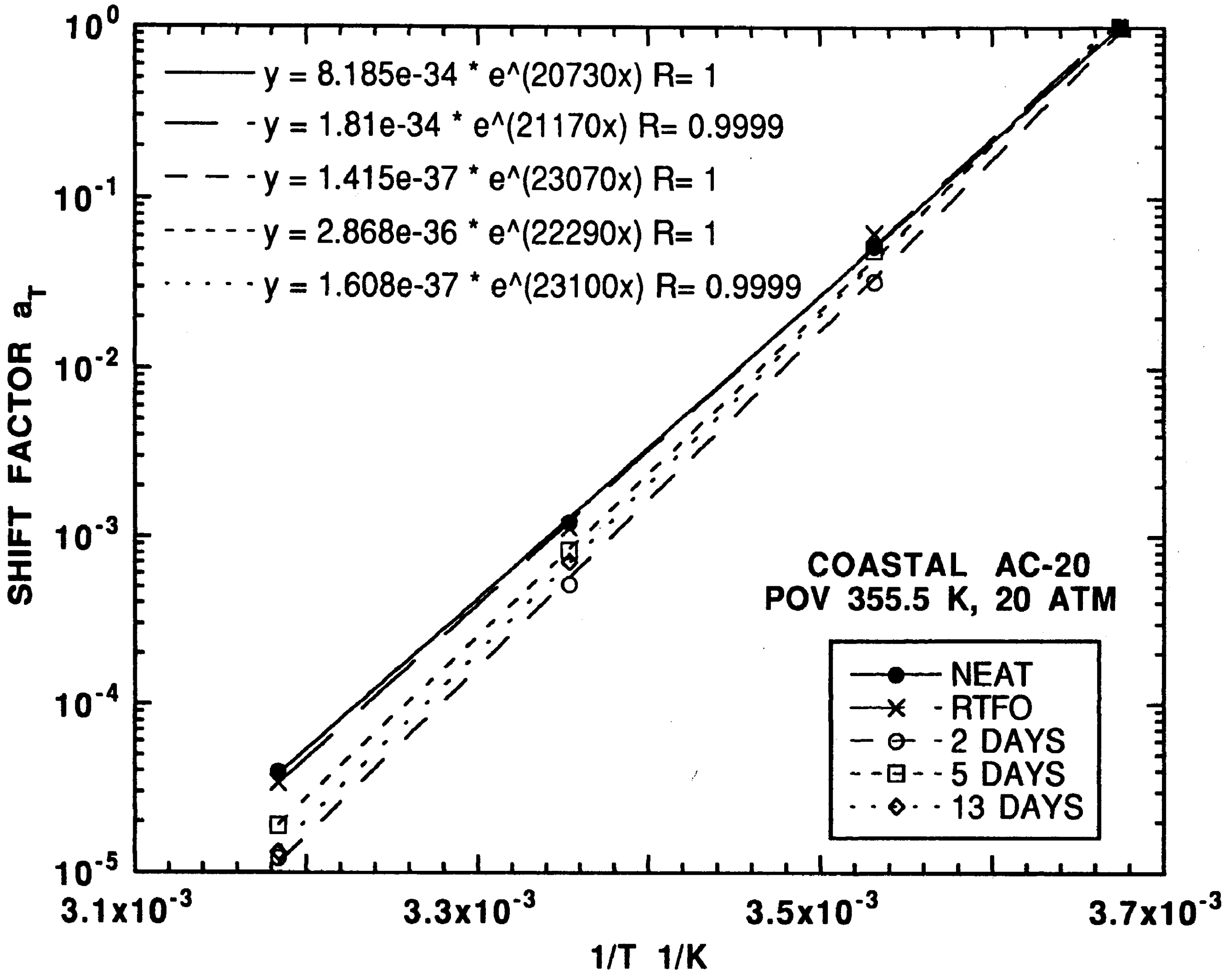

Figure C-2. a_T versus $(1 / T)$ for neat, RTFO, and POV-aged Coastal AC-20 with reference temperature of 273.2 K.

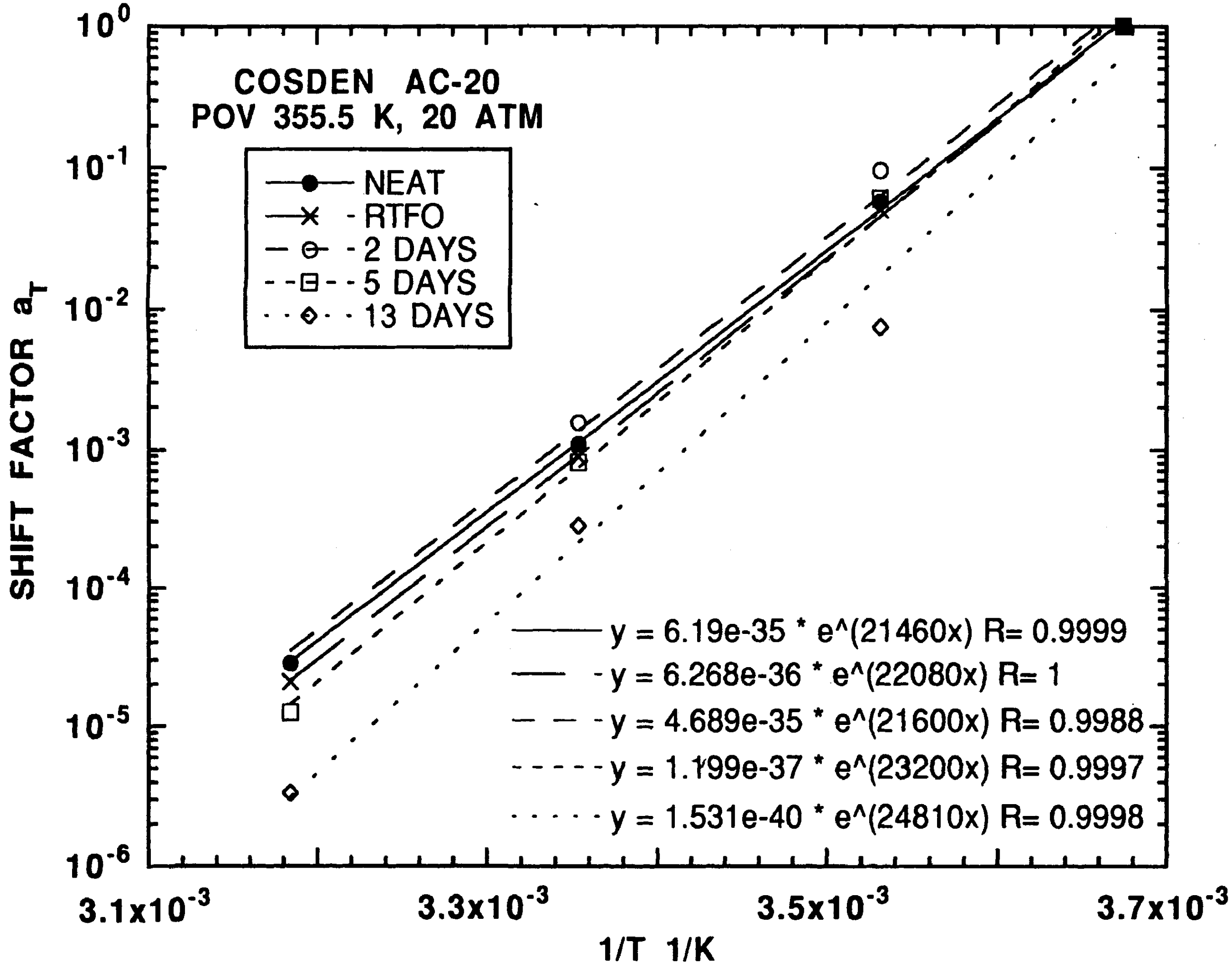

Figure C-3. a_T versus $(1 / T)$ for neat, RTFO, and POV-aged Cosden AC-20 with reference temperature of 273.2 K.

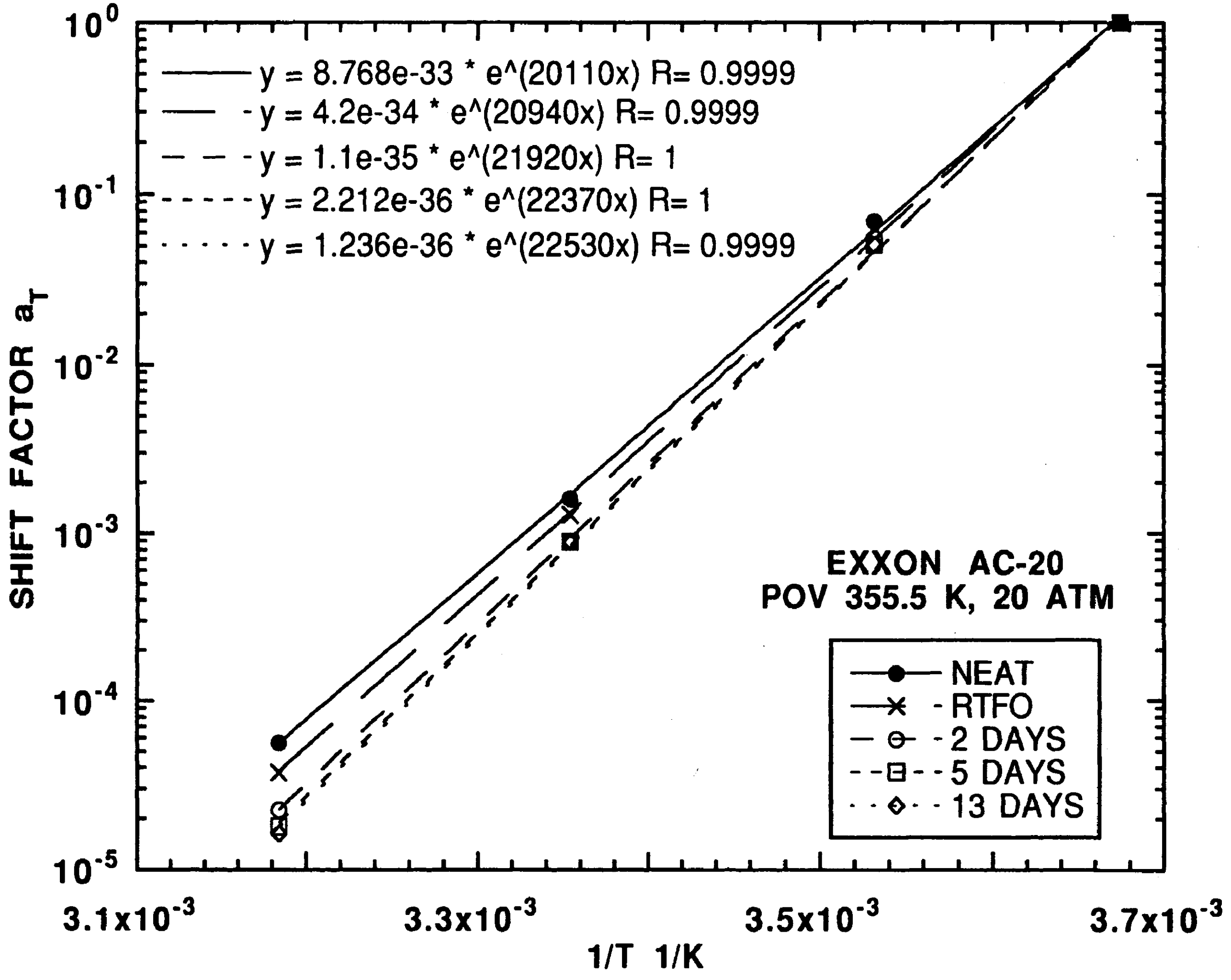

Figure C-4. a_T versus $(1 / T)$ for neat, RTFO, and POV-aged Exxon AC-20 with reference temperature of 273.2 K.

Table C-2. *CA* at the *ES* and *SI* for P_{ES} of 0.2 atm, 1 mm Thick Ampet AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.637	–
4	–	–	–	0.601	0.679	0.653
6	–	–	0.681	–	0.698	0.625
8	0.712	0.520	0.705	–	0.797	0.678
10	–	–	–	–	0.777	0.752
12	–	–	0.730	–	–	0.793
16	–	–	–	0.693	–	–
20	–	–	0.814	0.682	0.975	–
24	0.728	–	–	0.730	–	–
28	–	–	0.821	0.726	–	–
32	0.753	0.627	0.822	0.797	–	–
36	–	–	–	0.764	–	–
41	0.815	0.610	–	0.850	–	–
48	–	0.605	–	–	–	–
57	0.865	–	–	–	–	–
64	–	0.671	–	–	–	–
72	0.885	–	–	–	–	–
80	0.938	–	–	–	–	–

[a] – Signifies the values were not determined

Table C-3. *CA* at the *ES* and *SI* for P_{ES} of 2 atm, 1 mm Thick Ampet AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	–	0.729
4	–	–	0.745	0.707	0.917	0.749
5	0.764	–	–	–	–	–
8	–	–	–	–	1.014	0.927
9	0.772	–	0.772	–	1.063	–
10	–	–	–	–	–	–
12	0.882	–	0.882	0.846	–	–
14	–	–	–	–	1.276	1.082
17	–	–	0.917	–	–	–
20	–	–	0.927	0.904	–	–
21	0.902	–	–	–	1.468	–
23	–	–	–	–	–	1.418
24	–	–	–	1.003	–	–
28	0.925	0.878	1.056	0.993	–	–
32	–	–	1.212	1.101	–	–
36	–	–	1.151	1.046	–	–
40	–	–	1.262	1.155	–	–
44	0.993	0.948	–	–	–	–
68	1.237	–	–	–	–	–
76	1.218	–	–	–	–	–

[a] – Signifies the values were not determined

Table C-4. *CA* at the *ES* and *SI* for P_{ES} of 0.2 atm, 1 mm Thick Coastal AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.595	0.534
4	–	–	–	0.632	–	–
6	–	–	–	–	0.652	0.589
8	0.595	–	0.720	0.645	0.756	0.660
10	–	–	–	–	0.835	–
12	–	–	0.750	0.655	–	–
15	–	–	–	–	0.982	–
16	–	–	0.791	0.714	–	0.834
18	–	–	0.791	0.714	0.986	–
20	–	–	0.790	0.719	–	0.888
23	–	–	–	–	1.140	–
24	0.710	0.629	0.862	0.710	–	–
28	–	–	0.874	0.791	–	–
32	0.815	0.660	0.943	0.778	–	–
36	–	–	0.974	0.801	–	–
41	0.791	0.706	0.998	0.828	–	–
48	0.797	0.744	–	–	–	–
57	0.795	–	–	–	–	–
64	0.856	0.749	–	–	–	–
72	0.873	0.803	–	–	–	–

[a] – Signifies the values were not determined

Table C-5. ***CA*** **at the** ***ES*** **and** ***SI*** **for** P_{ES} **of 2 atm, 1 mm Thick Coastal AC-20**[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.845	0.689
4	–	–	0.793	0.665	0.894	0.773
5	0.823	–	–	–	–	–
6	–	–	–	–	–	0.831
9	–	–	0.846	–	–	–
10	–	–	–	–	–	0.900
12	0.885	–	0.992	–	–	–
14	–	–	–	–	1.250	0.967
17	–	–	1.104	–	–	–
18	–	–	1.104	–	1.325	1.252
20	–	–	–	0.936	–	–
21	0.890	–	–	–	1.570	–
23	–	–	–	–	1.591	–
24	–	–	–	0.910	–	–
28	0.951	0.900	–	–	–	–
32	–	–	1.306	–	–	–
36	1.049	0.939	1.253	0.992	–	–
40	–	–	1.398	–	–	–
44	1.044	0.966	–	–	–	–
52	1.180	1.057	–	–	–	–
60	1.285	1.047	–	–	–	–
68	1.161	–	–	–	–	–
76	1.237	–	–	–	–	–

[a] – Signifies the values were not determined

Table C-6. *CA* at the *ES* and *SI* for P_{ES} of 0.2 atm, 1 mm Thick Cosden AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.646	0.457
4	–	–	–	0.540	0.767	0.569
6	–	–	–	–	0.762	–
8	0.694	–	0.763	0.604	0.912	–
10	–	–	–	–	–	–
12	–	–	0.778	0.672	–	–
16	0.739	0.637	–	–	1.041	–
17	–	–	0.788	–	–	–
18	–	–	–	–	1.069	–
20	–	–	0.822	–	1.095	0.988
24	0.831	0.693	0.951	–	–	–
28	–	–	–	0.765	–	–
32	0.765	–	1.015	–	–	–
36	–	–	–	0.901	–	–
41	–	0.717	–	0.936	–	–
48	0.929	–	–	–	–	–
57	0.917	–	–	–	–	–
64	0.975	–	–	–	–	–
72	1.031	–	–	–	–	–
80	1.027	–	–	–	–	–

[a] – Signifies the values were not determined

Table C-7. *CA* at the *ES* and *SI* for P_{ES} of 2 atm, 1 mm Thick Cosden AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.946	–
4	–	–	0.833	0.802	–	0.834
5	0.856	–	–	–	–	–
6	–	–	–	–	1.006	–
9	–	–	0.943	–	1.209	1.083
12	0.945	–	–	–	–	–
14	–	–	–	–	1.375	–
16	–	–	–	–	–	1.286
17	–	–	1.088	–	–	–
18	–	–	–	–	1.425	–
20	–	–	1.153	0.982	–	–
21	1.053	0.982	–	–	1.769	–
28	–	1.052	–	1.192	–	–
32	–	–	1.393	–	–	–
36	–	–	–	1.161	–	–
44	1.249	1.157	–	–	–	–
52	1.318	1.214	–	–	–	–
60	1.430	–	–	–	–	–
76	1.487	1.400	–	–	–	–

[a] – Signifies the values were not determined

Table C-8. *CA* at the *ES* and *SI* for P_{ES} of 0.2 atm, 1 mm Thick Exxon AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.740	–
4	–	–	0.745	0.622	0.853	–
8	–	0.577	0.759	0.650	0.966	0.850
10	–	–	–	–	1.033	0.897
12	–	–	0.826	0.667	–	–
16	0.744	0.607	0.894	0.749	1.089	0.977
17	–	–	–	0.747	–	–
20	–	–	0.866	–	1.149	–
23	–	–	–	–	1.200	–
24	0.819	0.640	0.922	0.783	–	–
28	–	–	0.889	0.744	–	–
32	–	–	1.033	–	–	–
36	–	–	0.958	–	–	–
41	–	–	1.056	0.885	–	–
48	0.909	0.738	–	–	–	–
80	1.027	–	–	–	–	–

---

[a] – Signifies the values were not determined

Table C-9. *CA* at the *ES* and *SI* for P_{ES} of 2 atm, 1 mm Thick Exxon AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.871	0.729
4	–	–	0.813	–	–	–
5	0.875	0.765	–	–	–	–
6	–	–	–	–	0.955	0.889
9	–	–	0.918	0.891	1.146	0.922
10	–	–	–	–	–	1.002
12	0.988	0.872	–	0.973	–	–
14	–	–	–	–	1.267	1.105
16	–	–	–	–	1.283	1.134
17	–	–	1.037	–	–	–
18	–	–	–	–	1.400	1.287
20	–	–	1.051	0.959	–	–
21	1.009	0.948	–	–	–	–
28	1.032	–	1.192	–	–	–
32	–	–	–	1.203	–	–
36	1.226	1.038	–	1.214	–	–
40	–	–	1.319	–	–	–
44	1.218	1.100	–	–	–	–
52	1.314	1.186	–	–	–	–
60	1.412	1.208	–	–	–	–
68	1.387	1.202	–	–	–	–
76	1.426	1.287	–	–	–	–

[a] – Signifies the values were not determined

Table C-10. *CA* at the *ES* and *SI* for P_{ES} of 0.2 atm, 1 mm Thick Texaco AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*	*ES* *CA*	*SI* *CA*
2	–	–	–	–	0.594	–
4	–	–	0.617	–	0.623	0.574
6	–	–	–	–	0.656	0.569
8	–	0.522	0.644	0.576	0.722	0.635
12	–	–	–	0.584	–	–
15	–	–	–	–	0.853	0.756
16	0.560	0.549	0.726	–	0.848	–
17	–	–	–	0.583	–	–
20	–	–	0.714	0.648	0.918	–
24	0.667	0.566	0.726	–	–	–
28	–	–	0.793	0.709	–	–
32	0.622	0.597	0.851	–	–	–
36	–	–	0.896	–	–	–
41	0.716	0.611	0.878	0.782	–	–
48	0.763	0.667	–	–	–	–
57	–	0.671	–	–	–	–
64	0.766	0.701	–	–	–	–
72	0.803	–	–	–	–	–
80	0.815	–	–	–	–	–

[a] – Signifies the values were not determined

Table C-11. *CA* at the *ES* and *SI* for P_{ES} of 2 atm, 1 mm Thick Texaco AC-20[a]

	333.3 K		344.4 K		355.5 K	
t days	*ES CA*	*SI CA*	*ES CA*	*SI CA*	*ES CA*	*SI CA*
2	–	–	–	–	0.747	0.665
4	–	–	0.699	0.693	–	0.669
5	0.729	0.619	–	–	–	–
6	–	–	–	–	0.927	0.759
9	–	–	0.809	0.782	–	–
10	–	–	–	–	–	0.801
12	0.765	0.717	0.856	–	–	–
17	–	–	0.918	–	–	–
18	–	–	–	–	1.172	0.962
20	–	–	–	0.845	–	–
21	0.819	0.777	–	–	1.353	–
23	–	–	–	–	1.353	1.222
24	–	–	1.045	–	–	–
28	0.850	0.770	–	1.020	–	–
32	–	–	1.115	1.045	–	–
36	0.912	–	–	–	–	–
40	–	–	–	1.042	–	–
44	0.971	–	–	–	–	–
52	1.100	–	–	–	–	–
60	1.011	0.919	–	–	–	–
68	1.120	–	–	–	–	–
76	1.132	1.062	–	–	–	–

[a] – Signifies the values were not determined

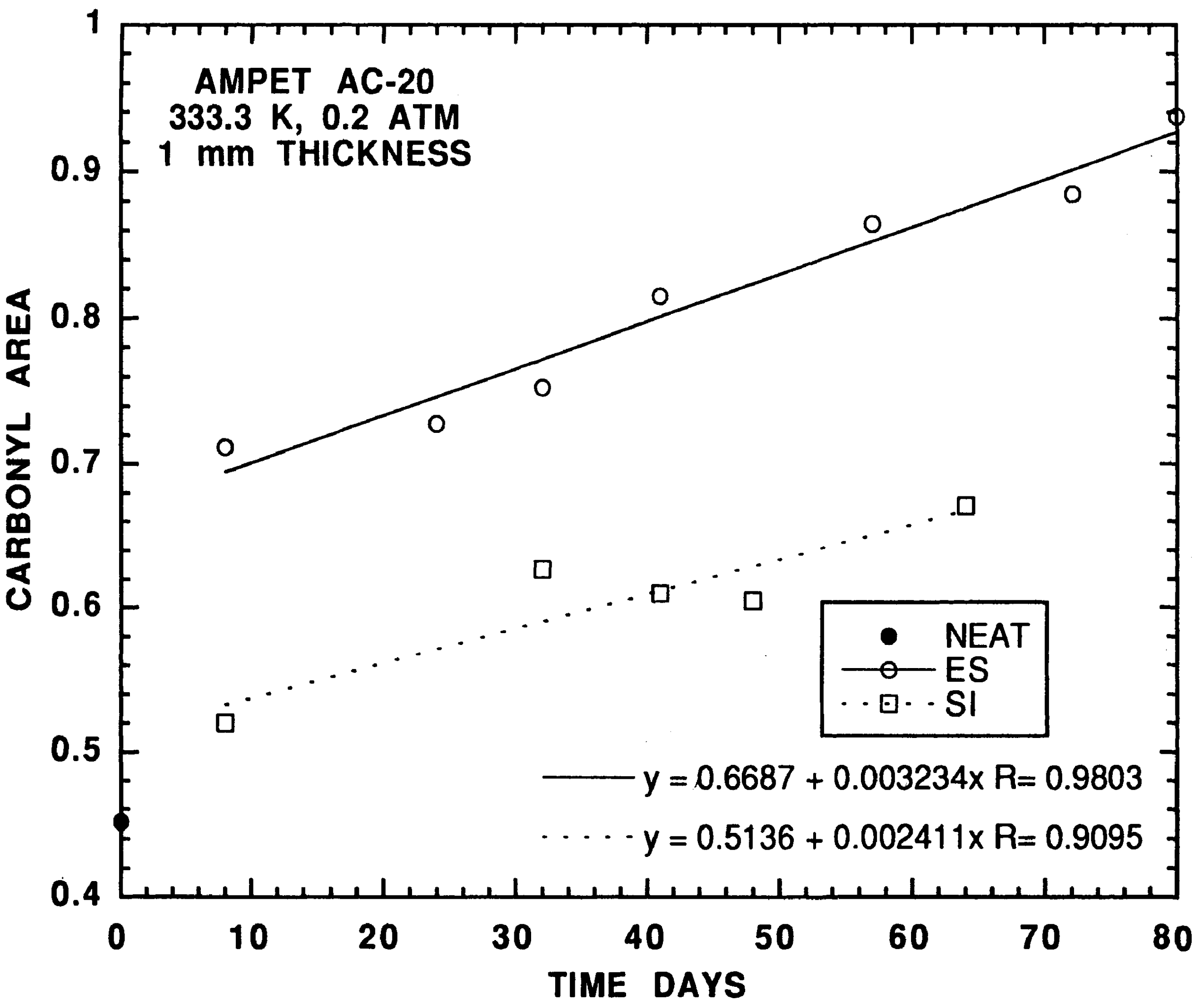

Figure C-5. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 333.3 K and P_{ES} of 0.2 atm.

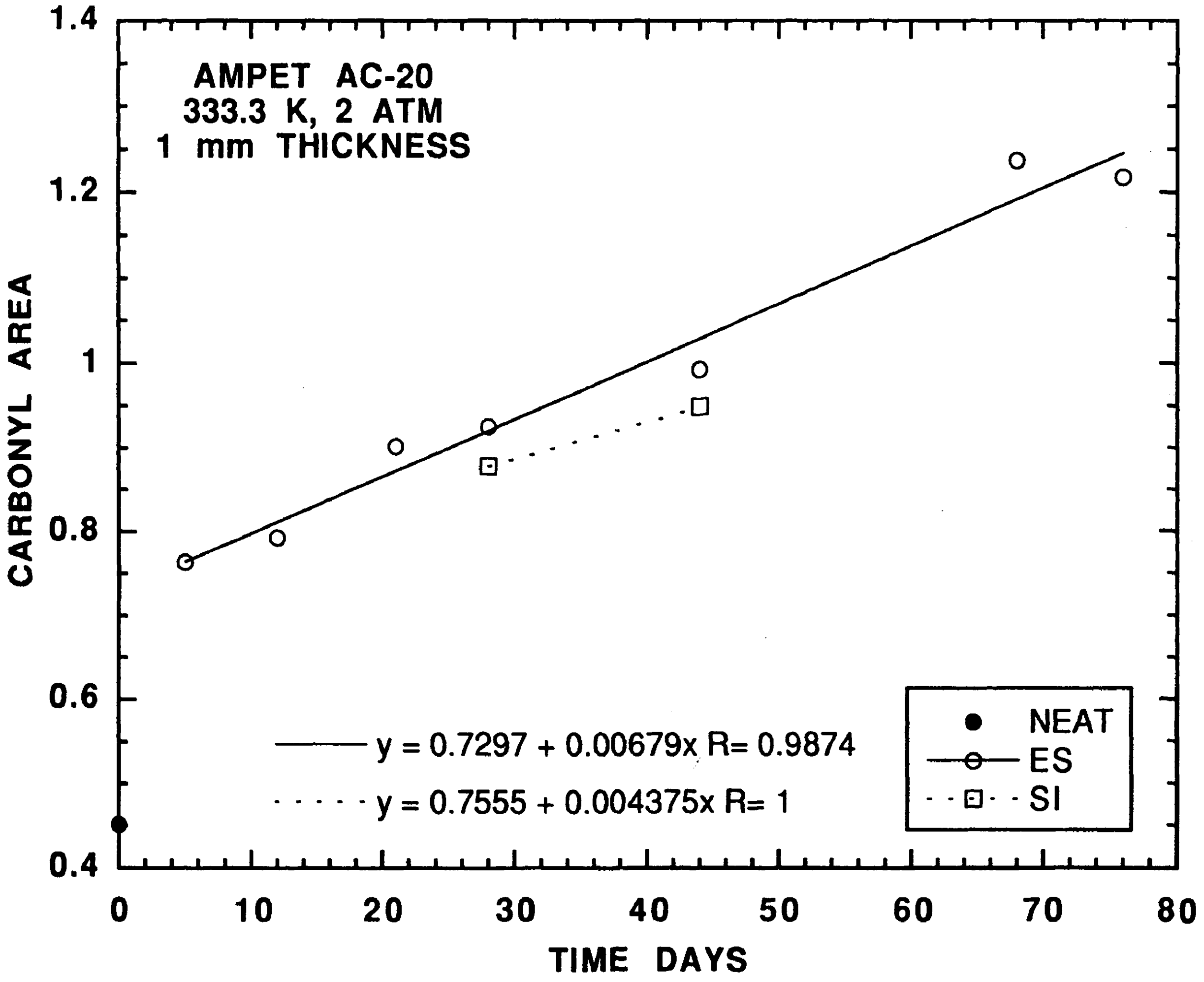

Figure C-6. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 333.3 K and P_{ES} of 2 atm.

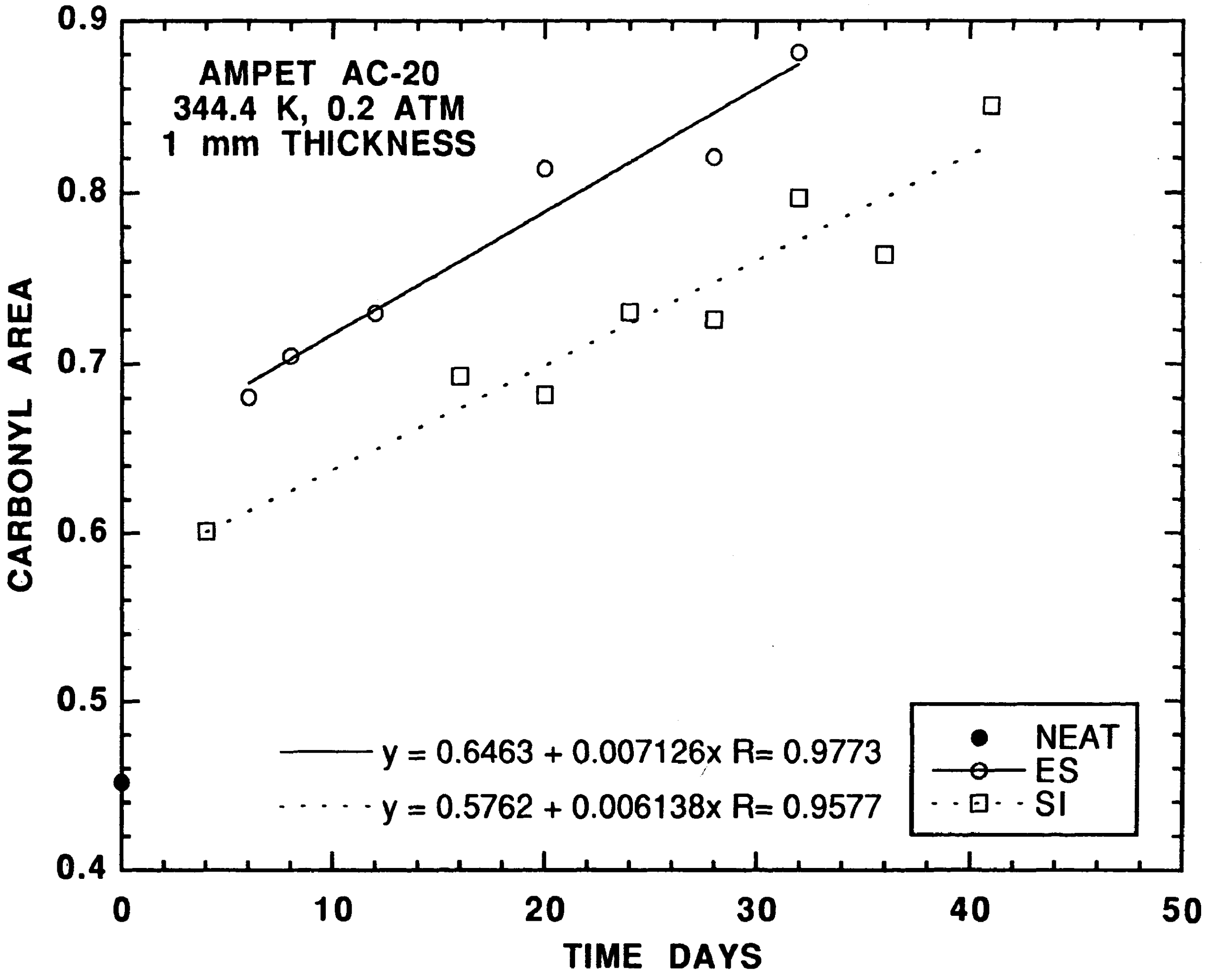

Figure C-7. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 344.4 K and P_{ES} of 0.2 atm.

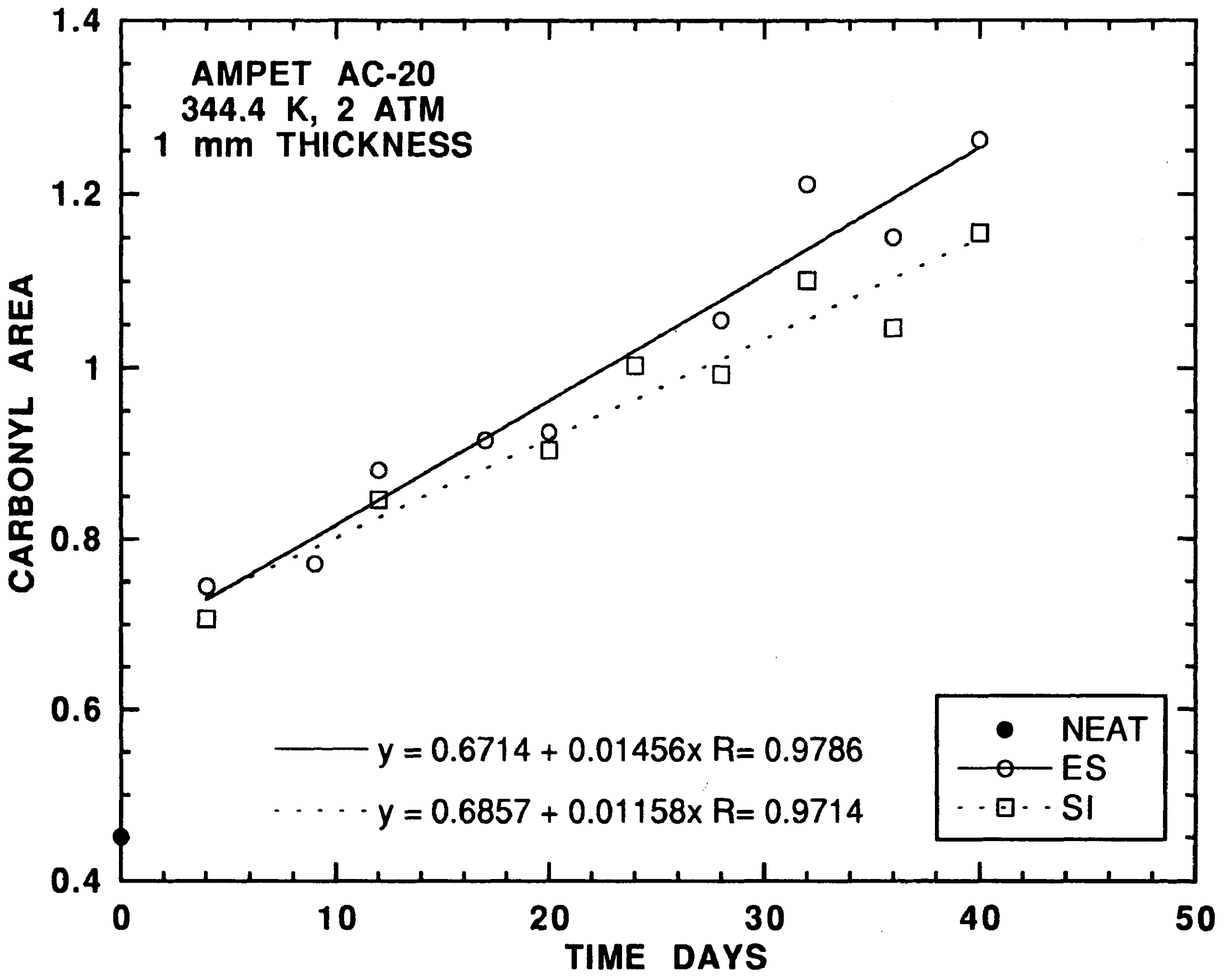

Figure C-8. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 344.4 K and P_{ES} of 2 atm.

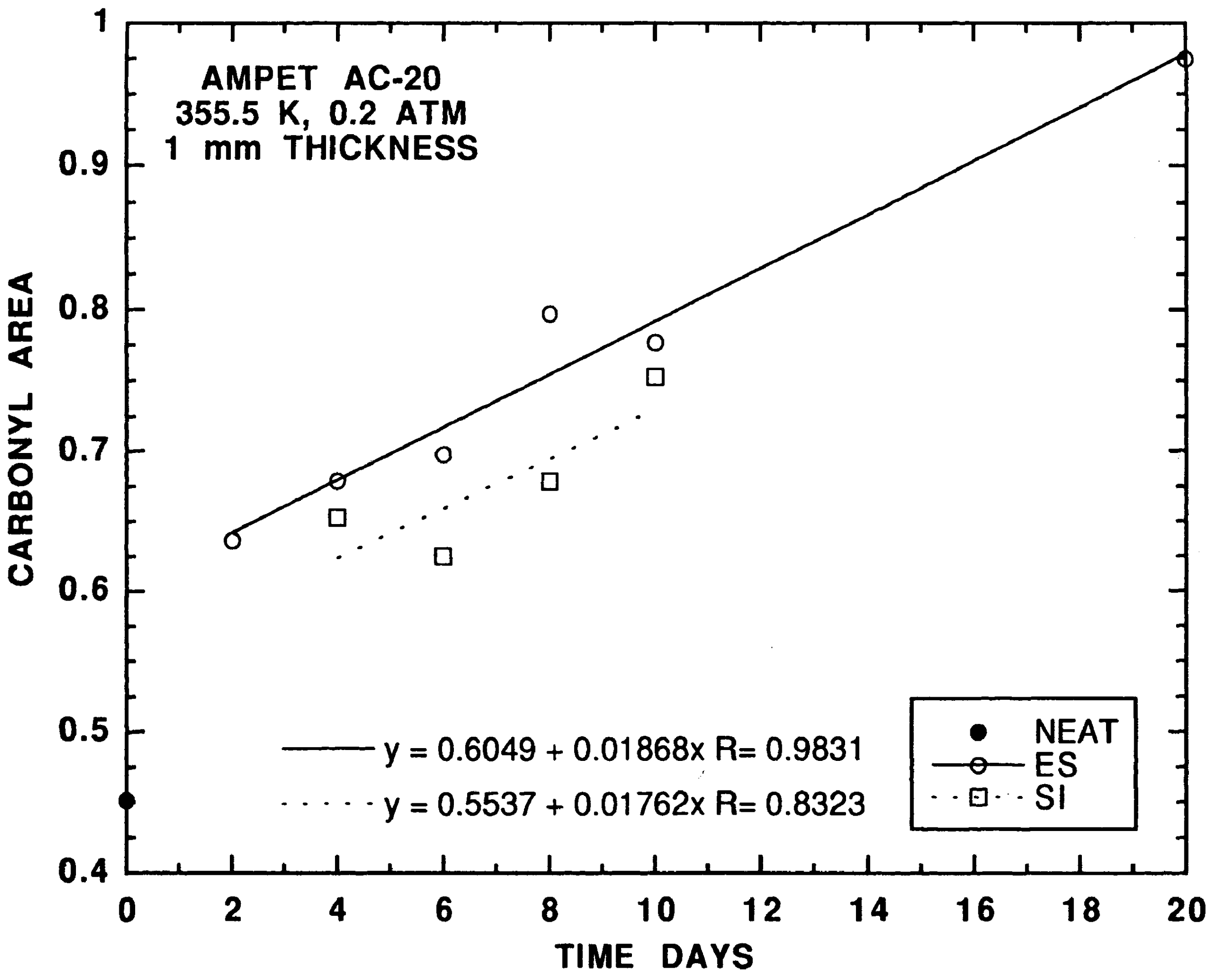

Figure C-9. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 355.5 K and P_{ES} of 0.2 atm.

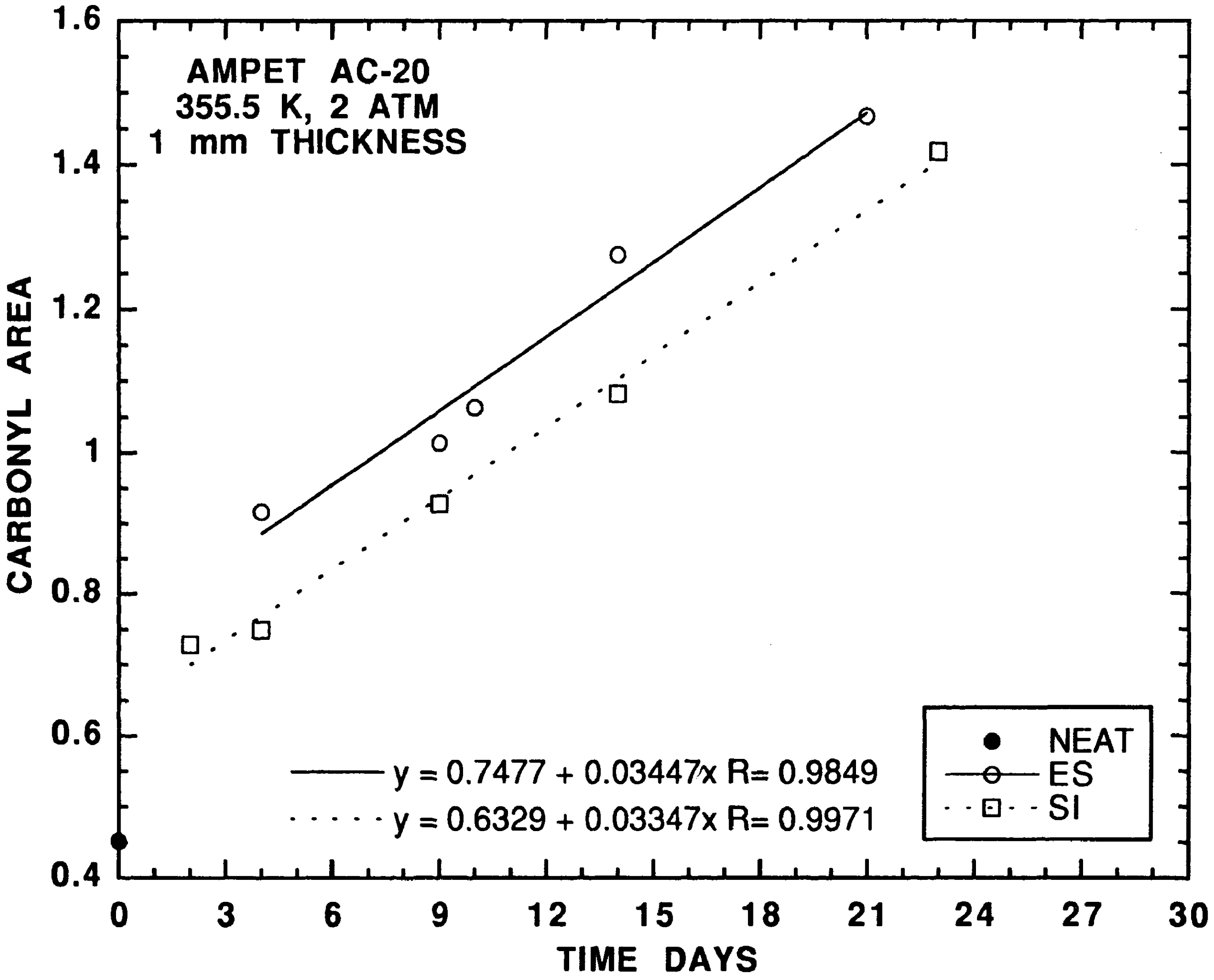

Figure C-10. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Ampet AC-20 at 355.5 K and P_{ES} of 2 atm.

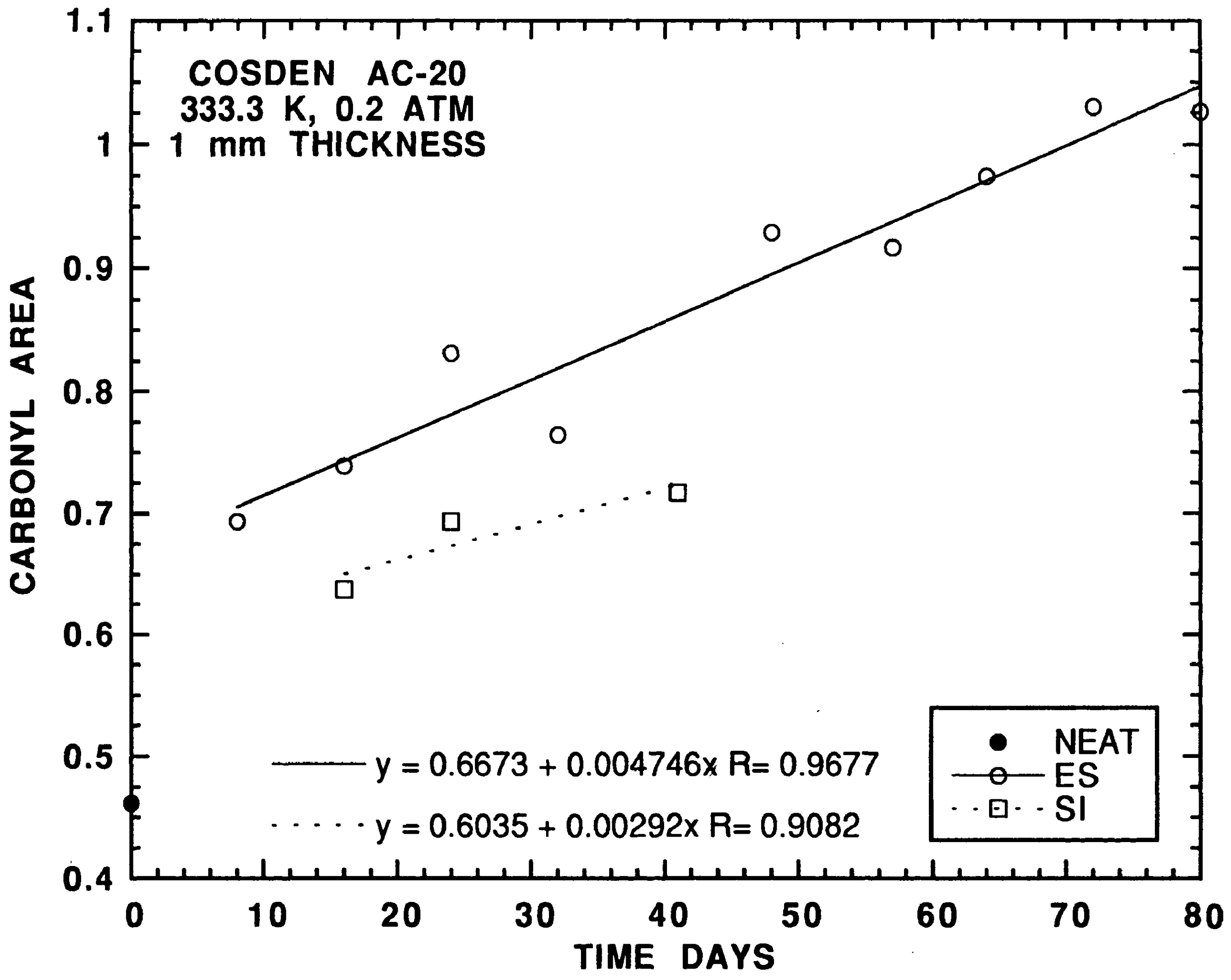

Figure C-11. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 333.3 K and P_{ES} of 0.2 atm.

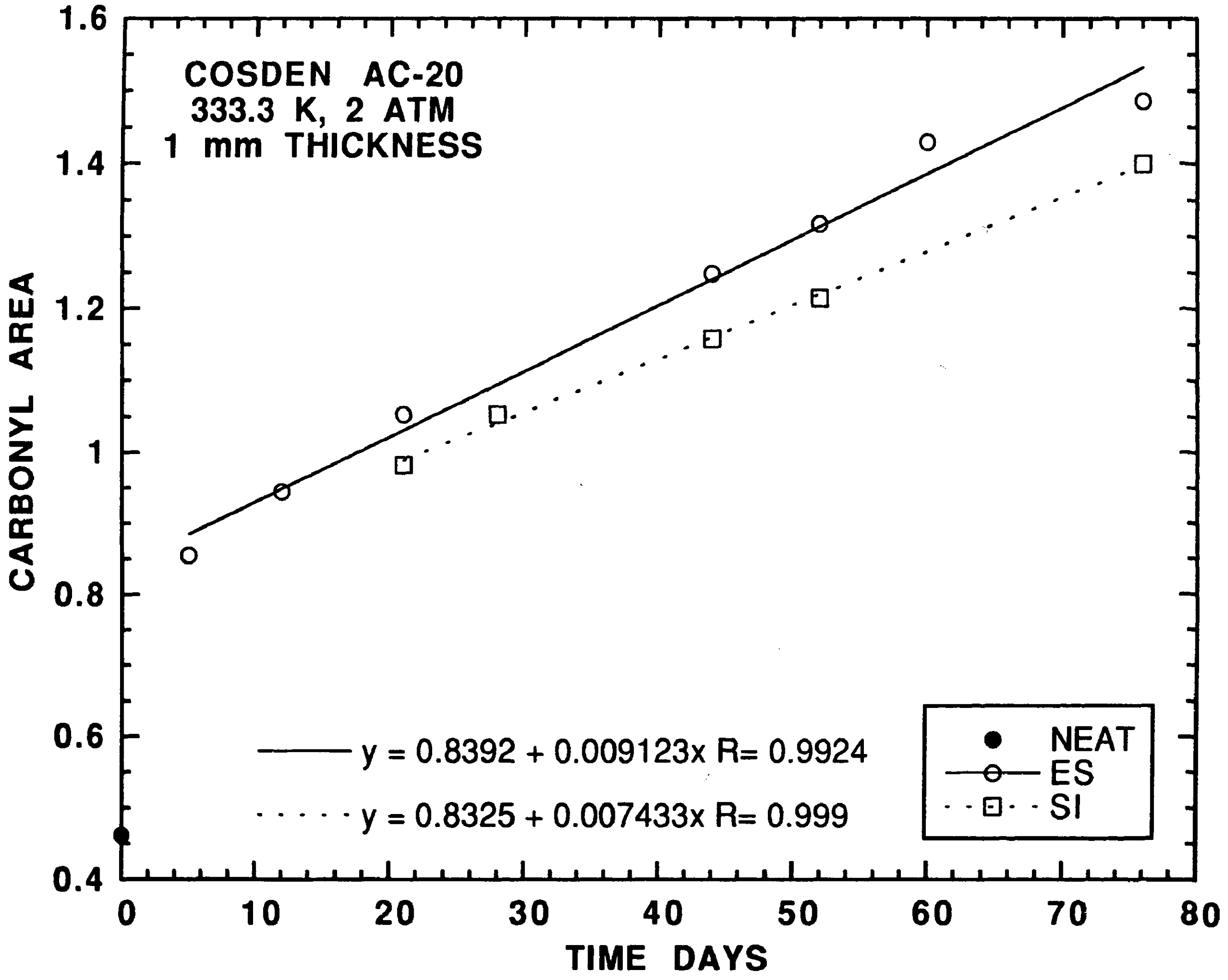

Figure C-12. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 333.3 K and P_{ES} of 2 atm.

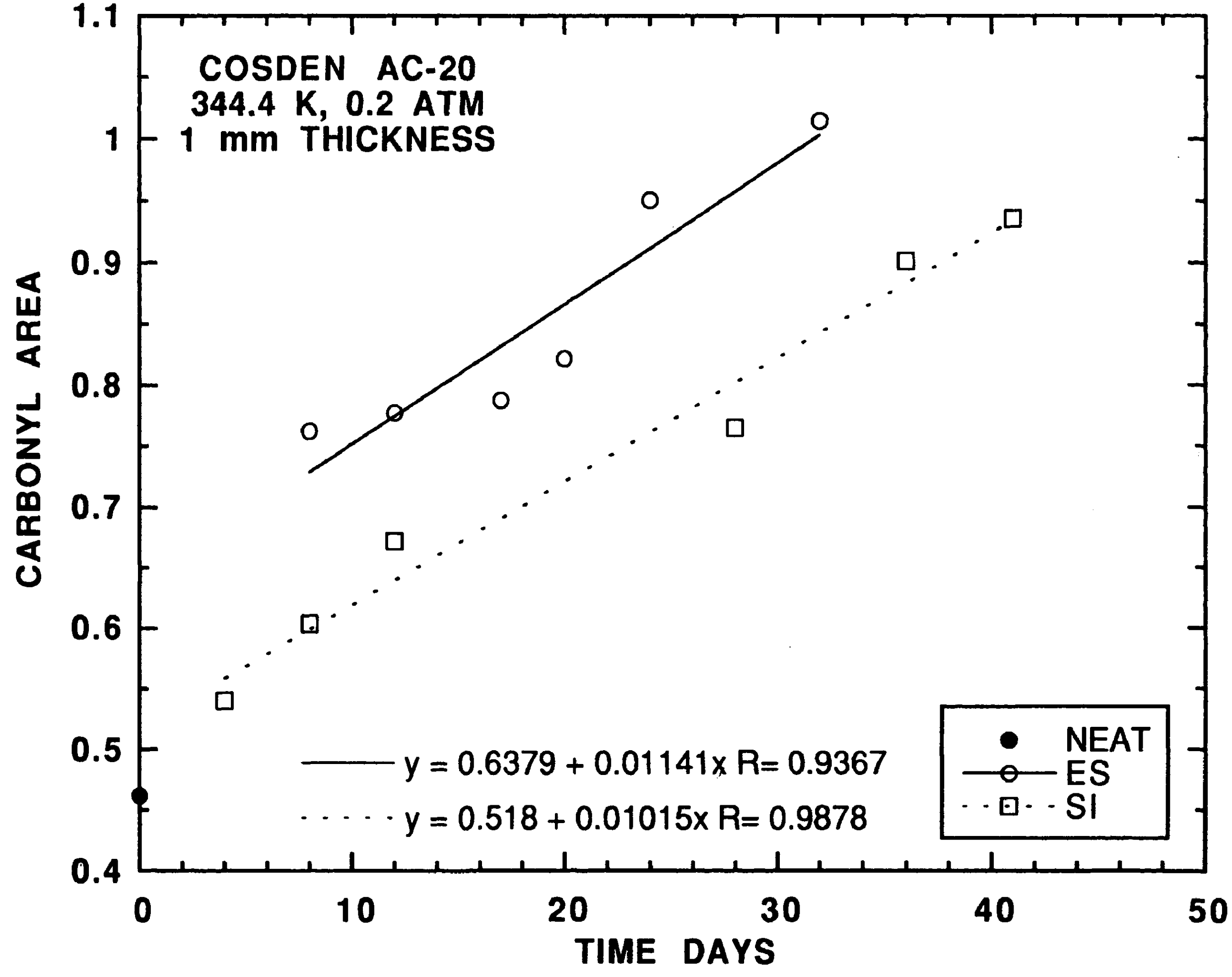

Figure C-13. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 344.4 K and P_{ES} of 0.2 atm.

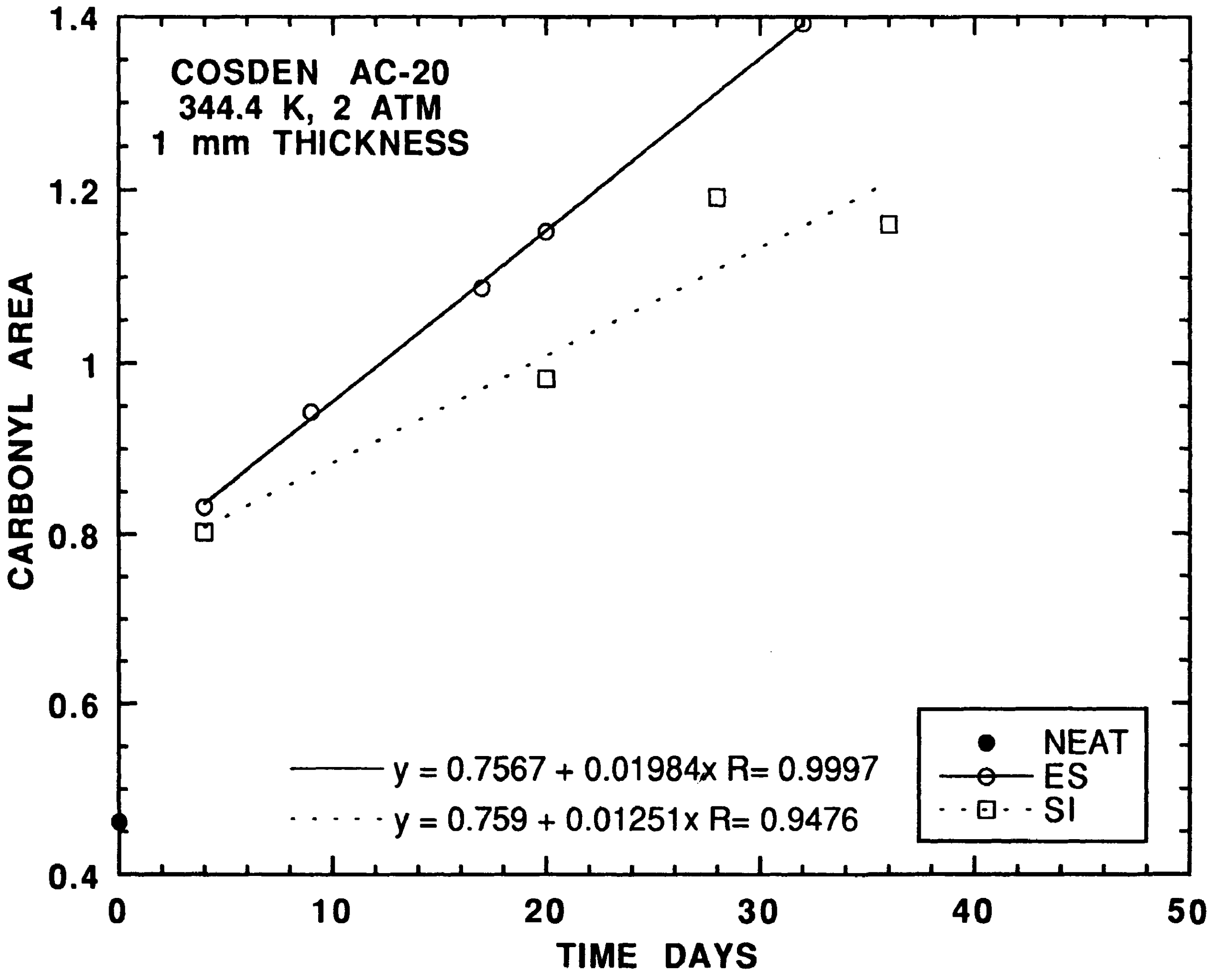

Figure C-14. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 344.4 K and P_{ES} of 2 atm.

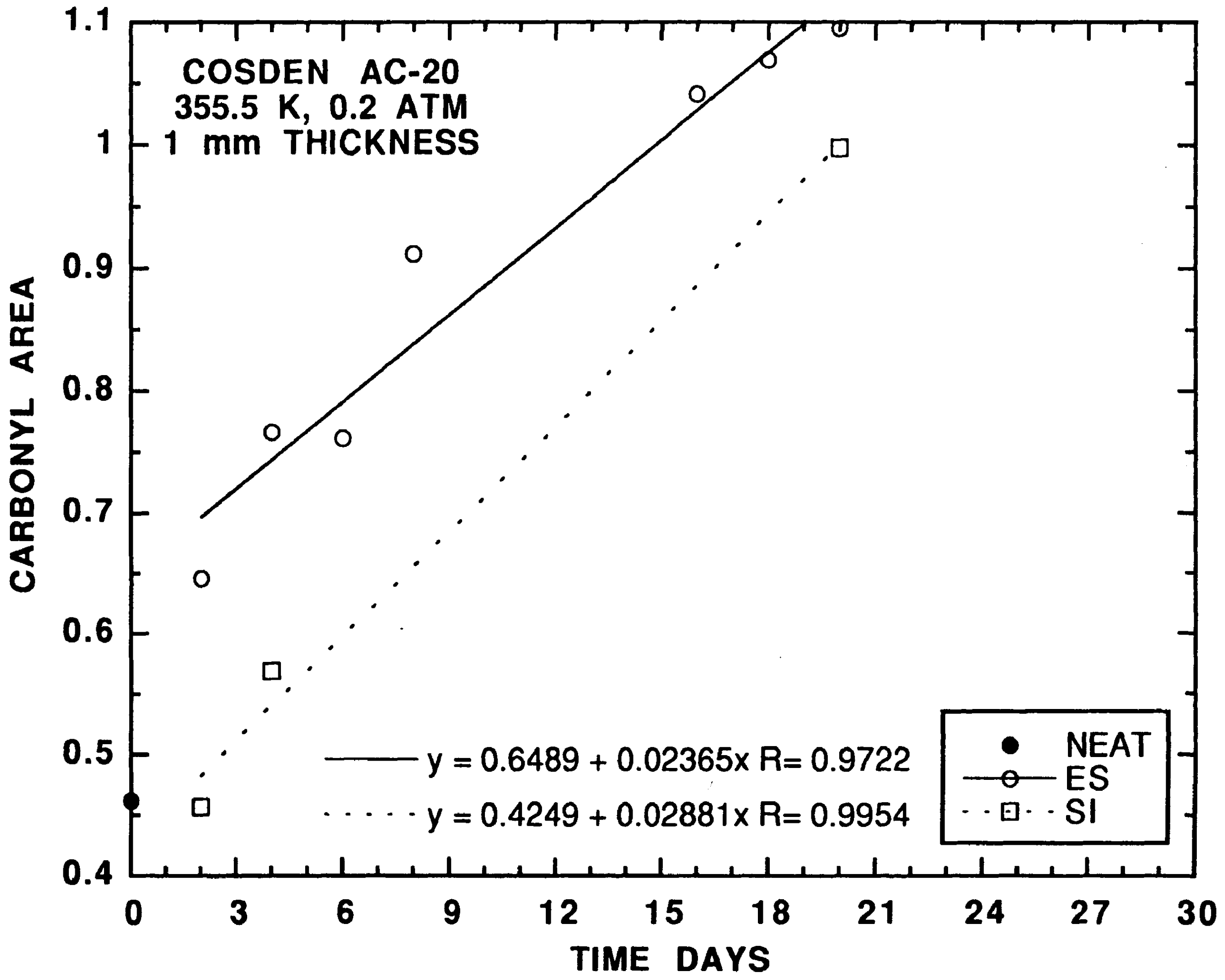

Figure C-15. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 355.5 K and P_{ES} of 0.2 atm.

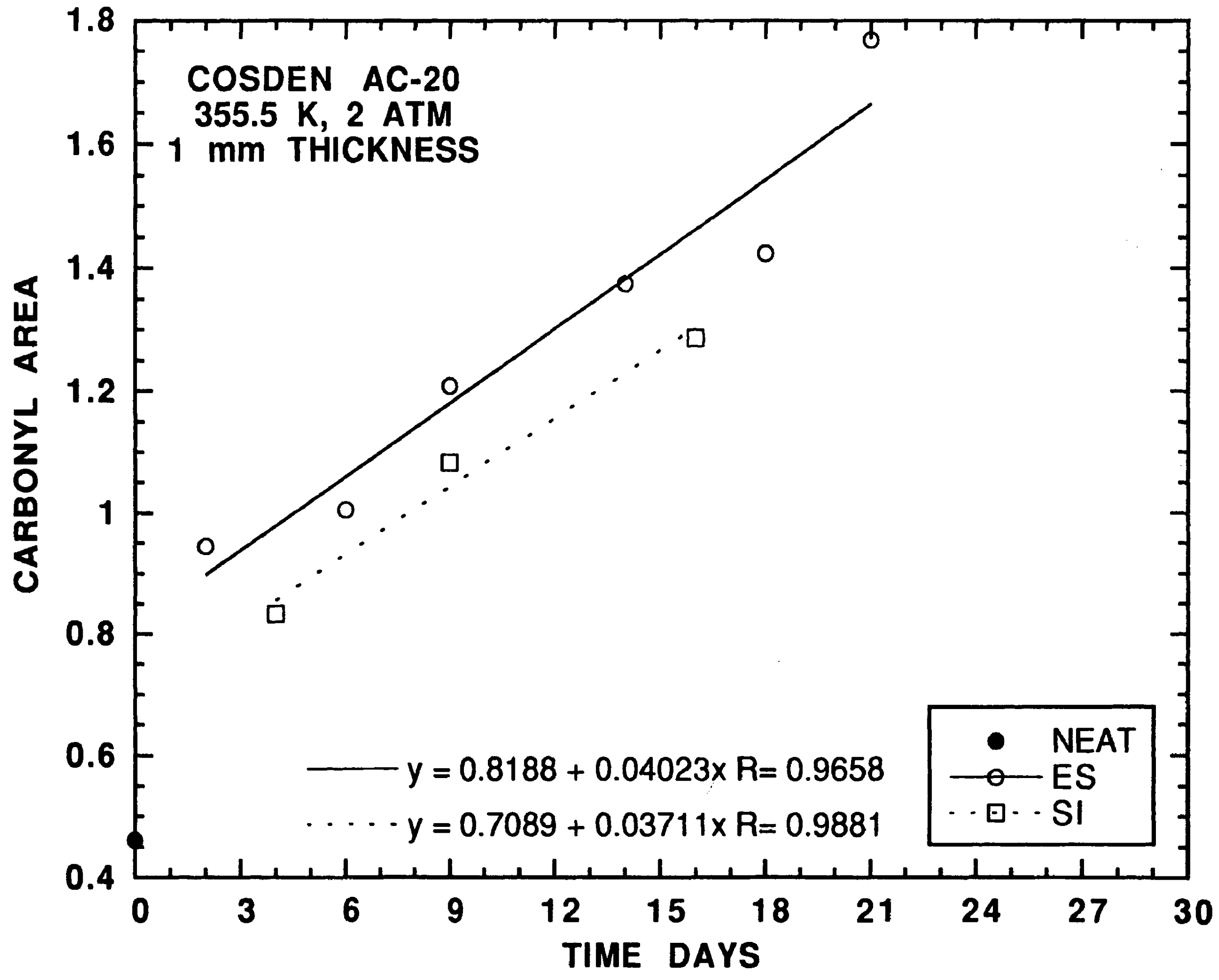

Figure C-16. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Cosden AC-20 at 355.5 K and P_{ES} of 2 atm.

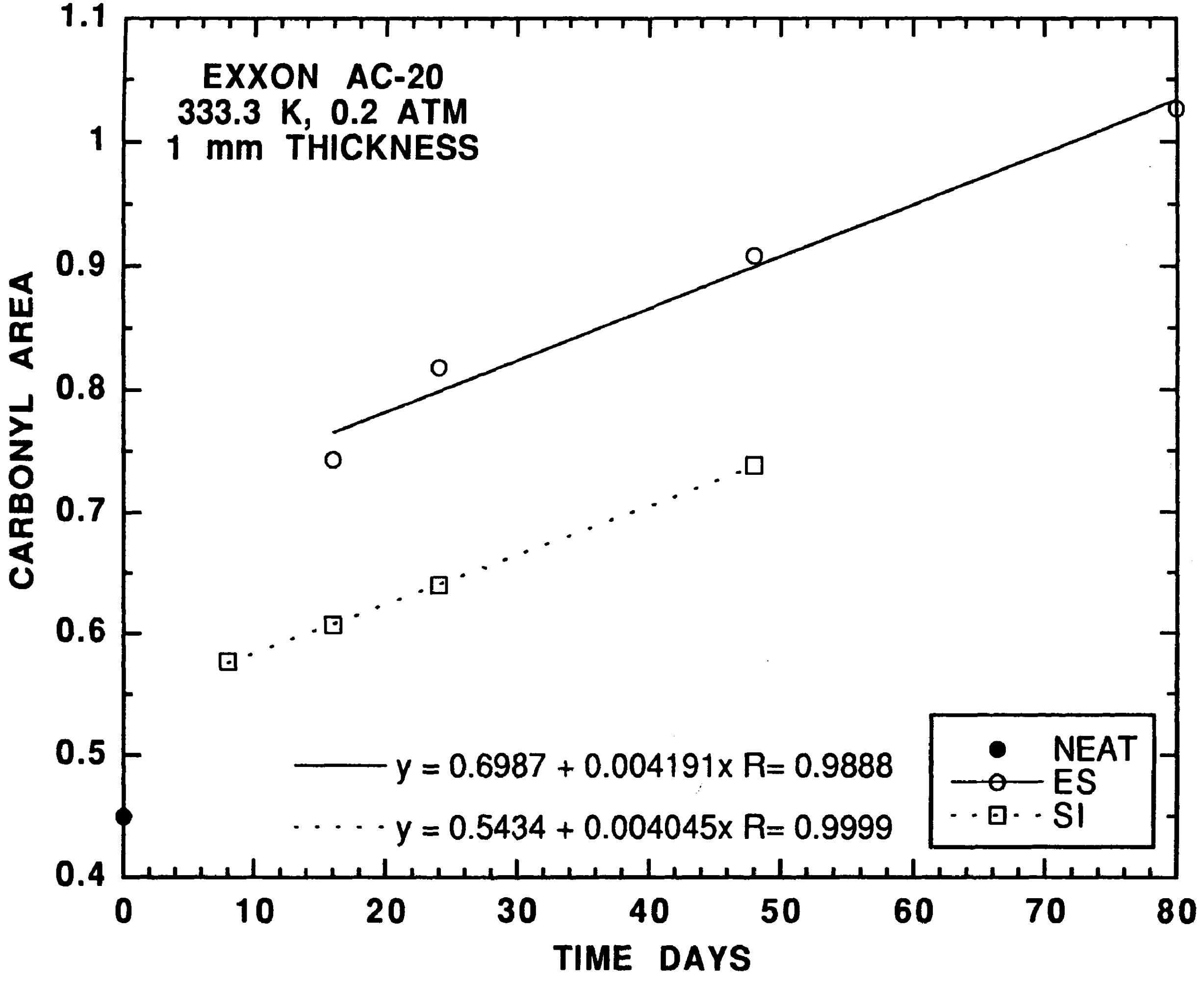

Figure C-17. *CAs* of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 333.3 K and P_{ES} of 0.2 atm.

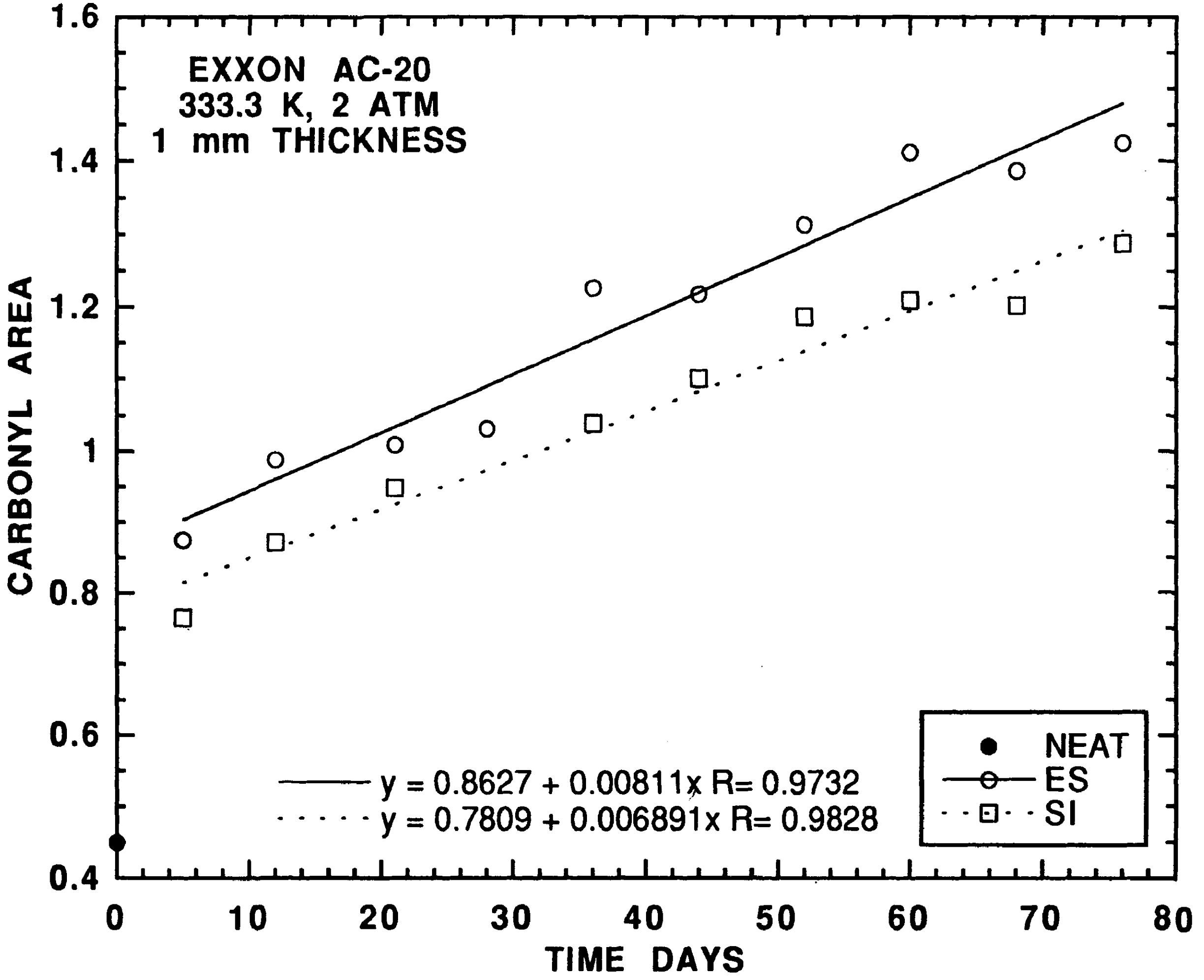

Figure C-18. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 333.3 K and P_{ES} of 2 atm.

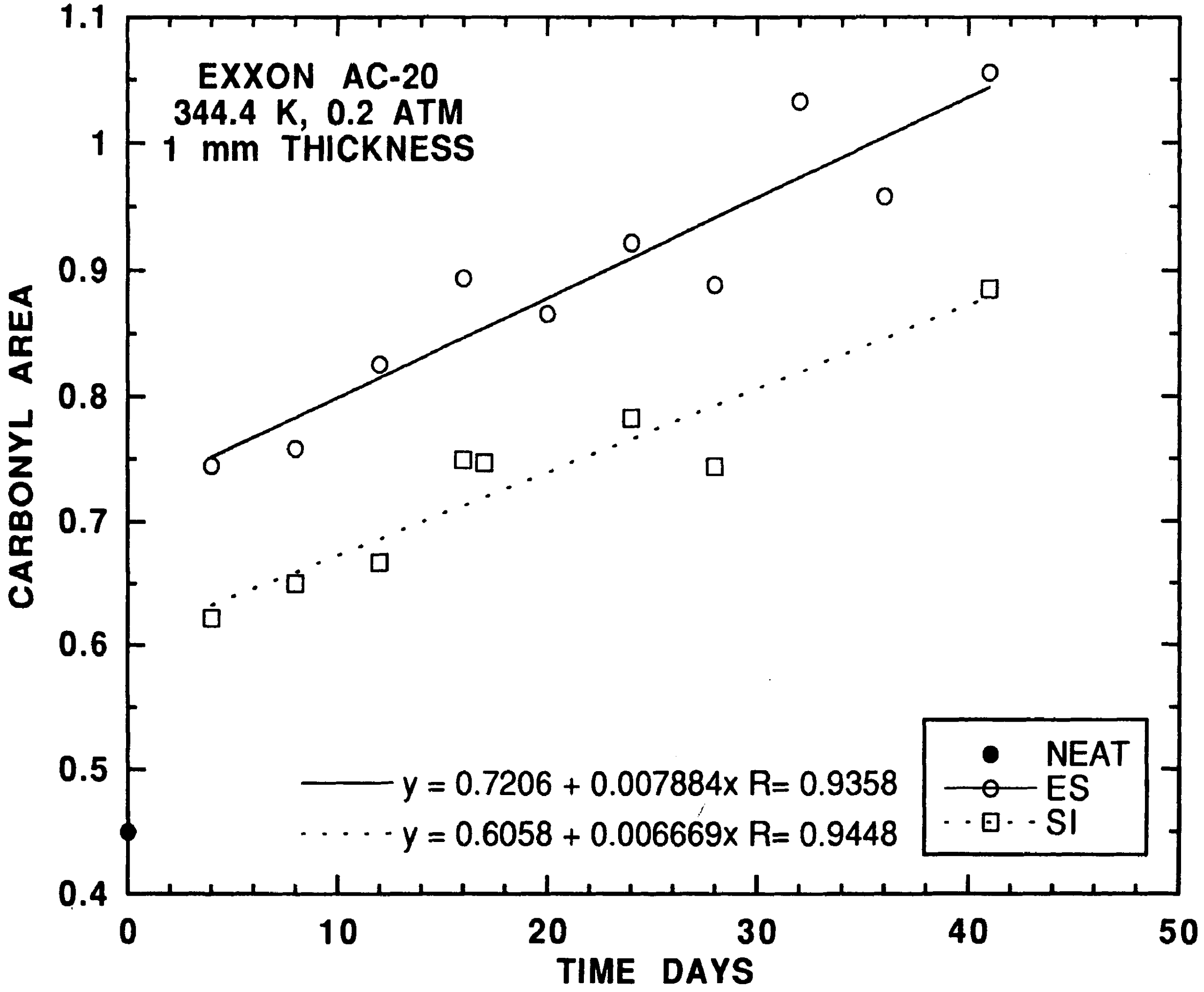

Figure C-19. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 344.4 K and P_{ES} of 0.2 atm.

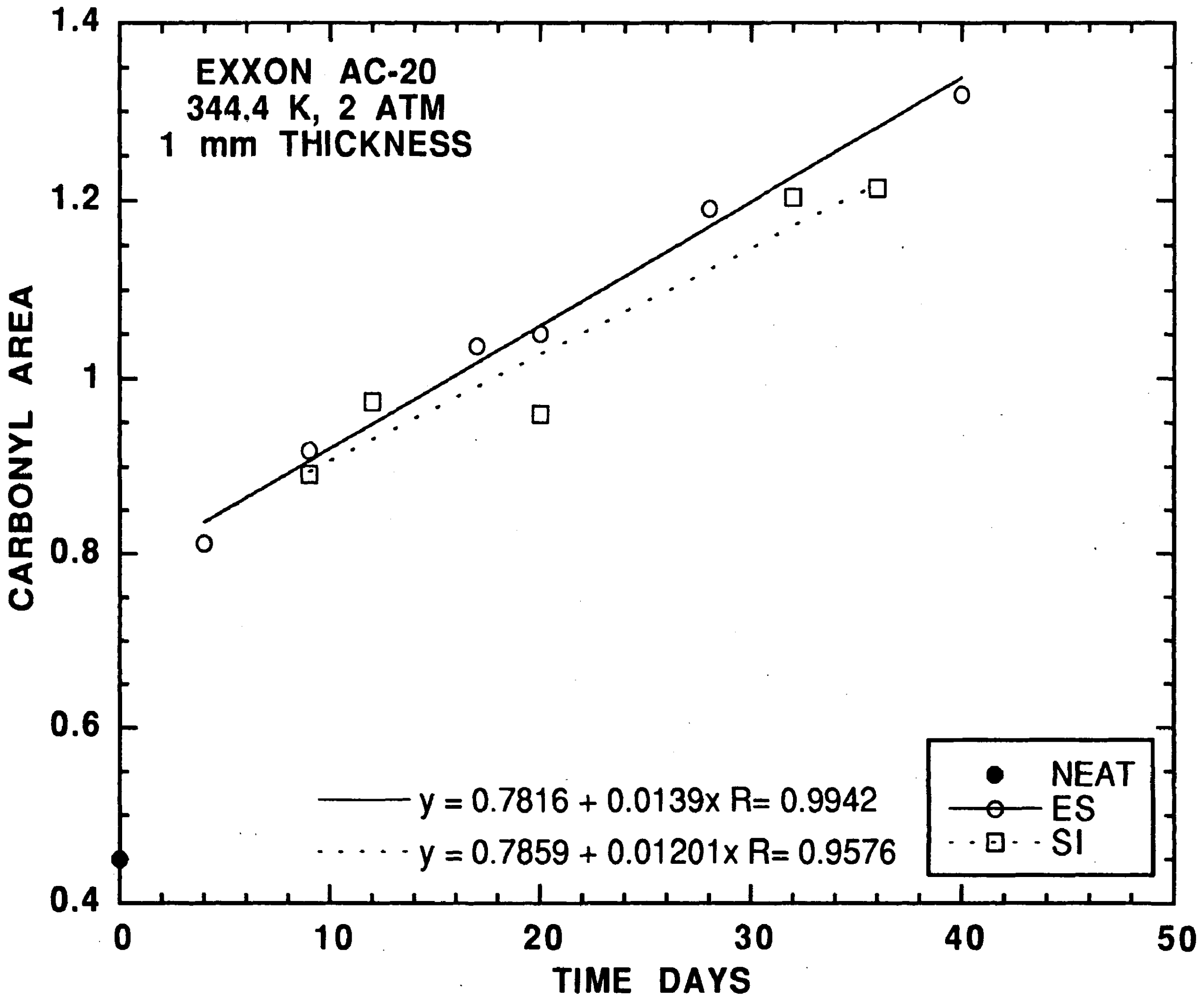

Figure C-20. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 344.4 K and P_{ES} of 2 atm.

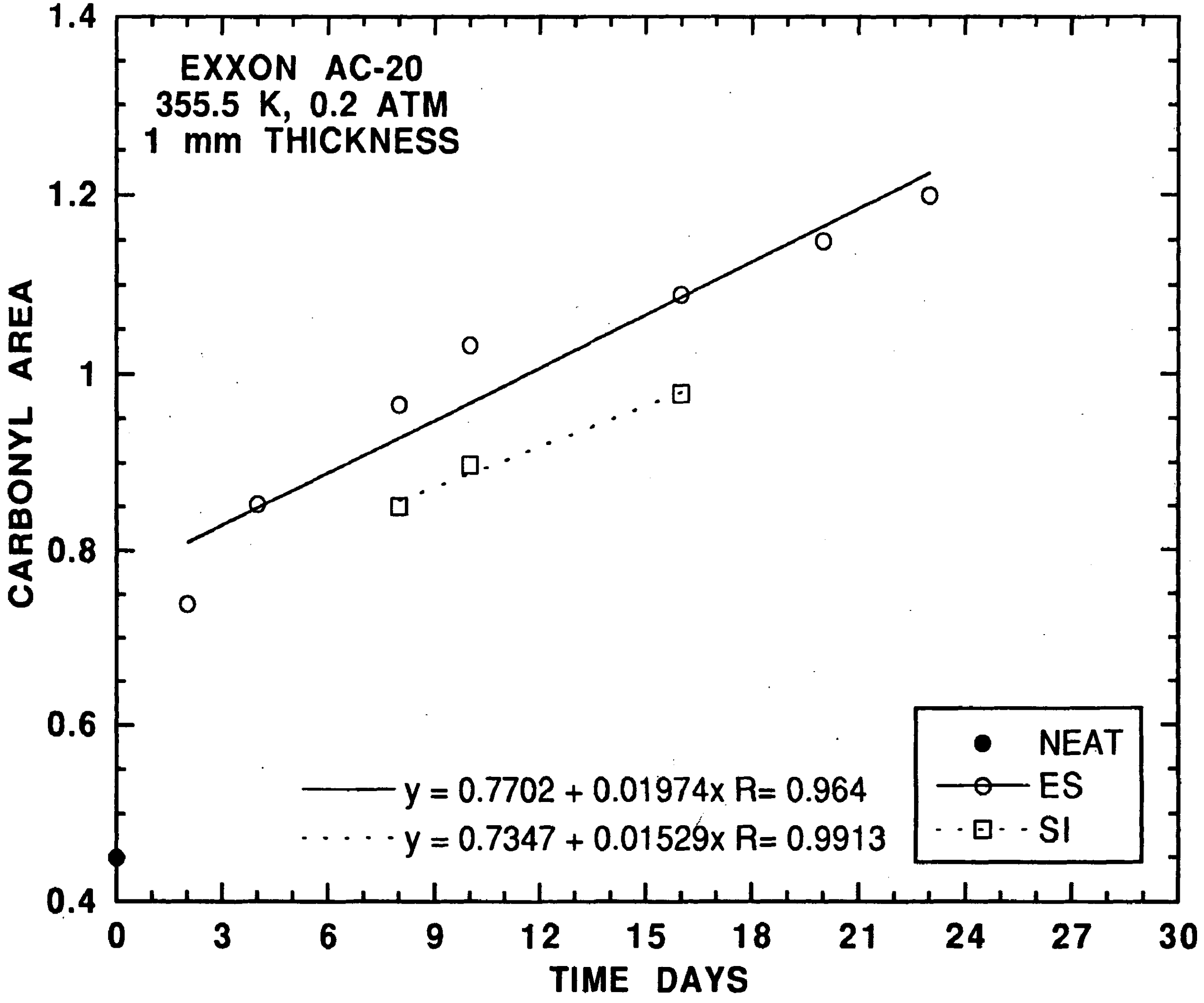

Figure C-21. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 355.5 K and P_{ES} of 0.2 atm.

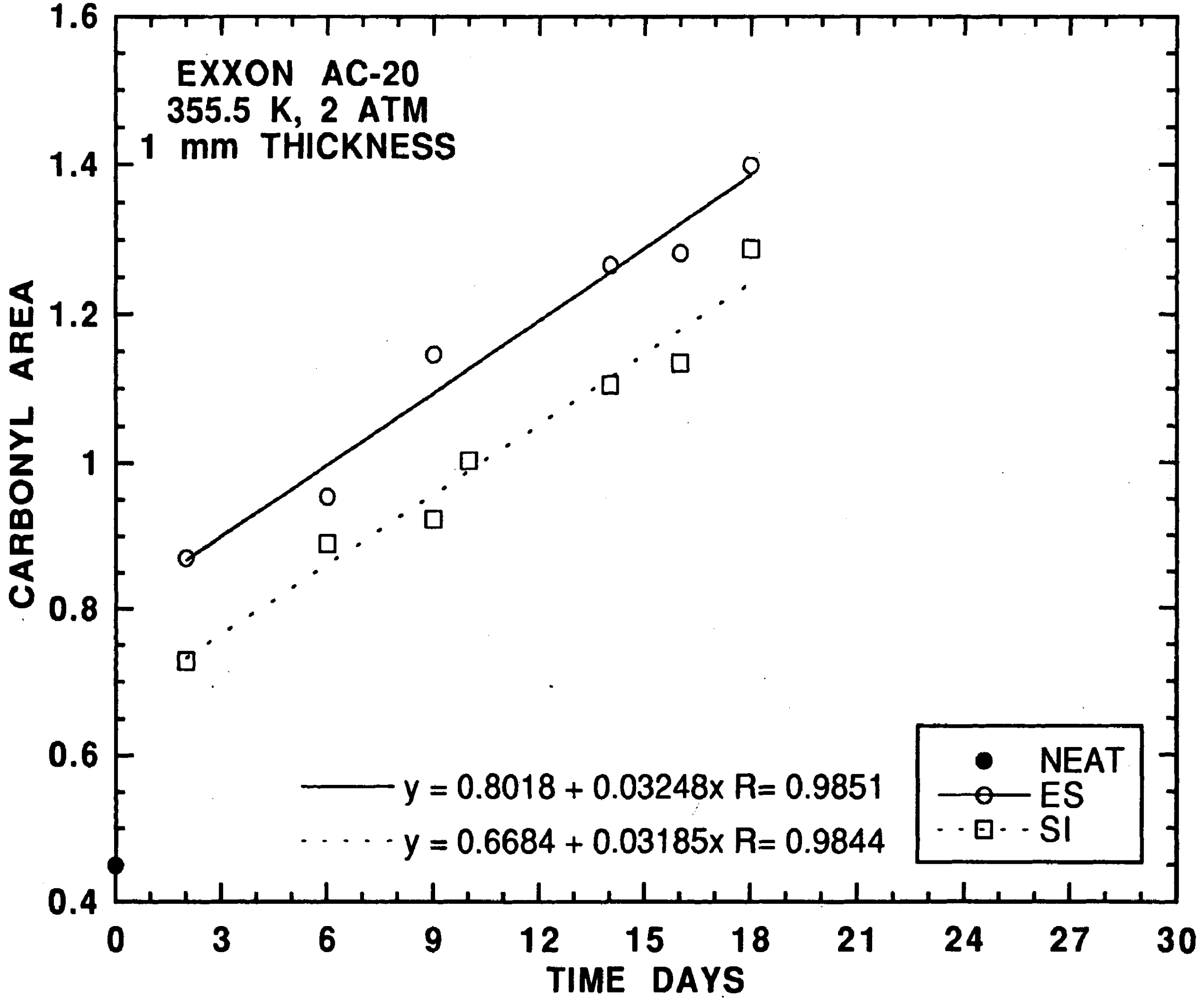

Figure C-22. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Exxon AC-20 at 355.5 K and P_{ES} of 2 atm.

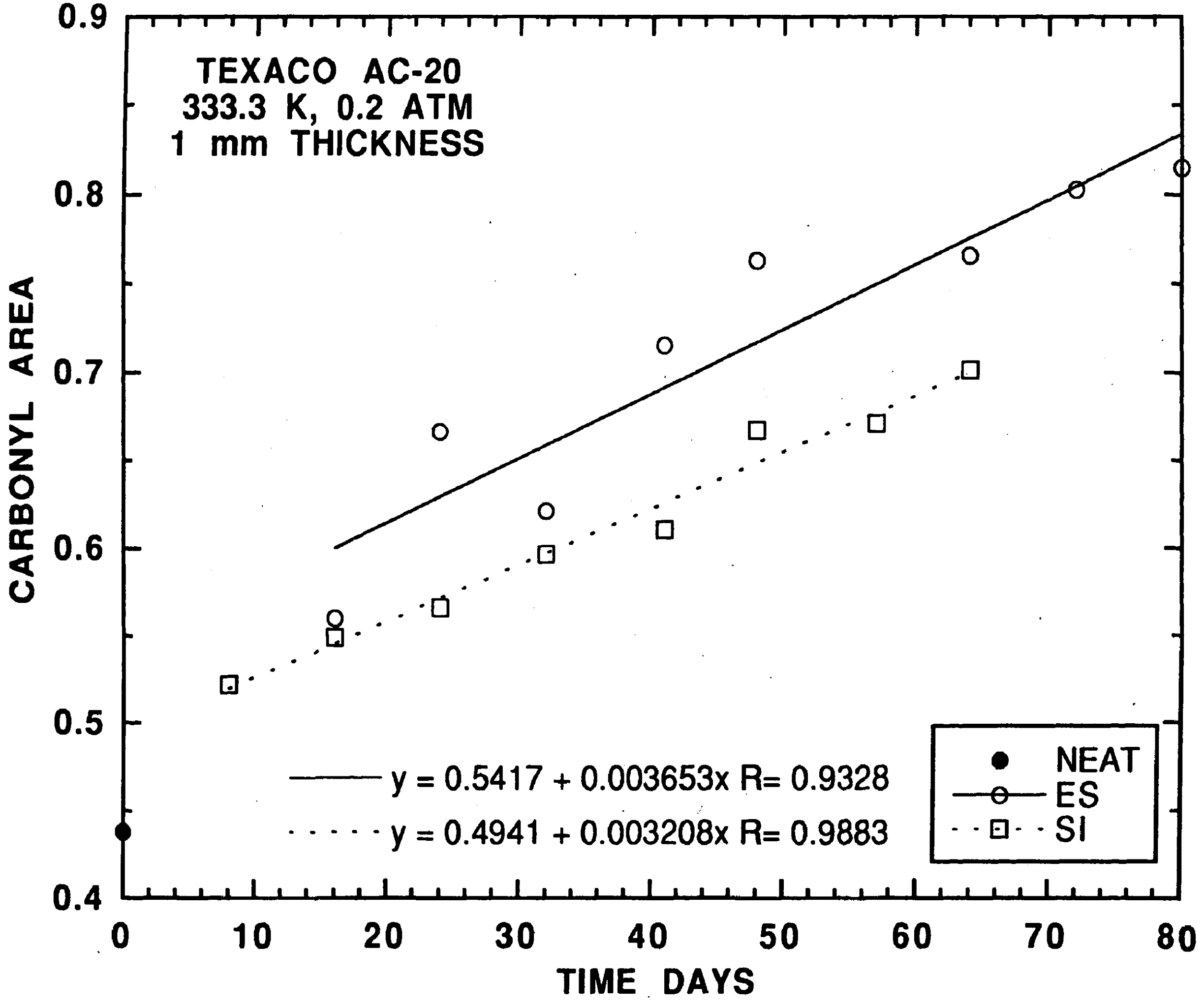

Figure C-23. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 333.3 K and P_{ES} of 0.2 atm.

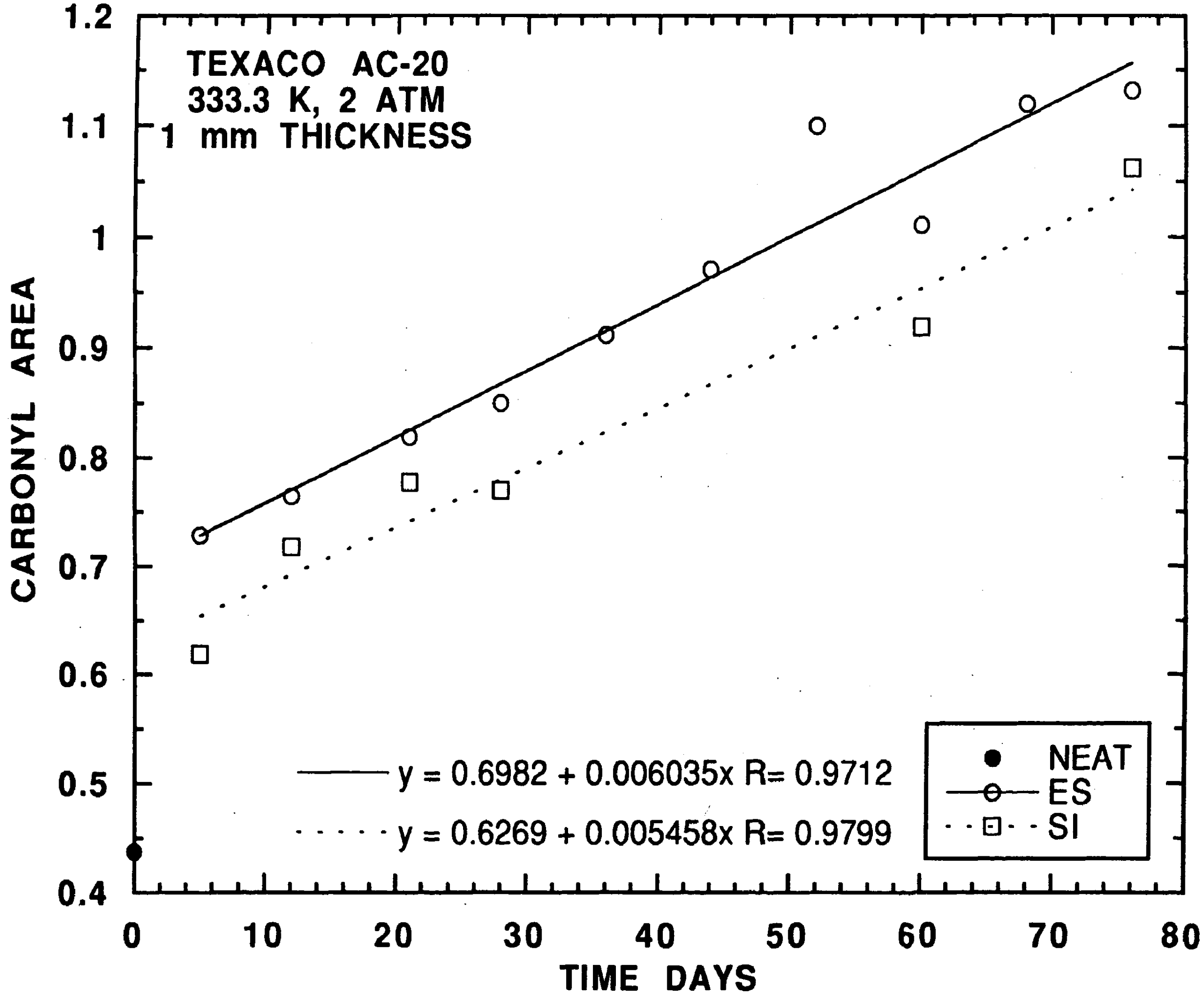

Figure C-24. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 333.3 K and P_{ES} of 2 atm.

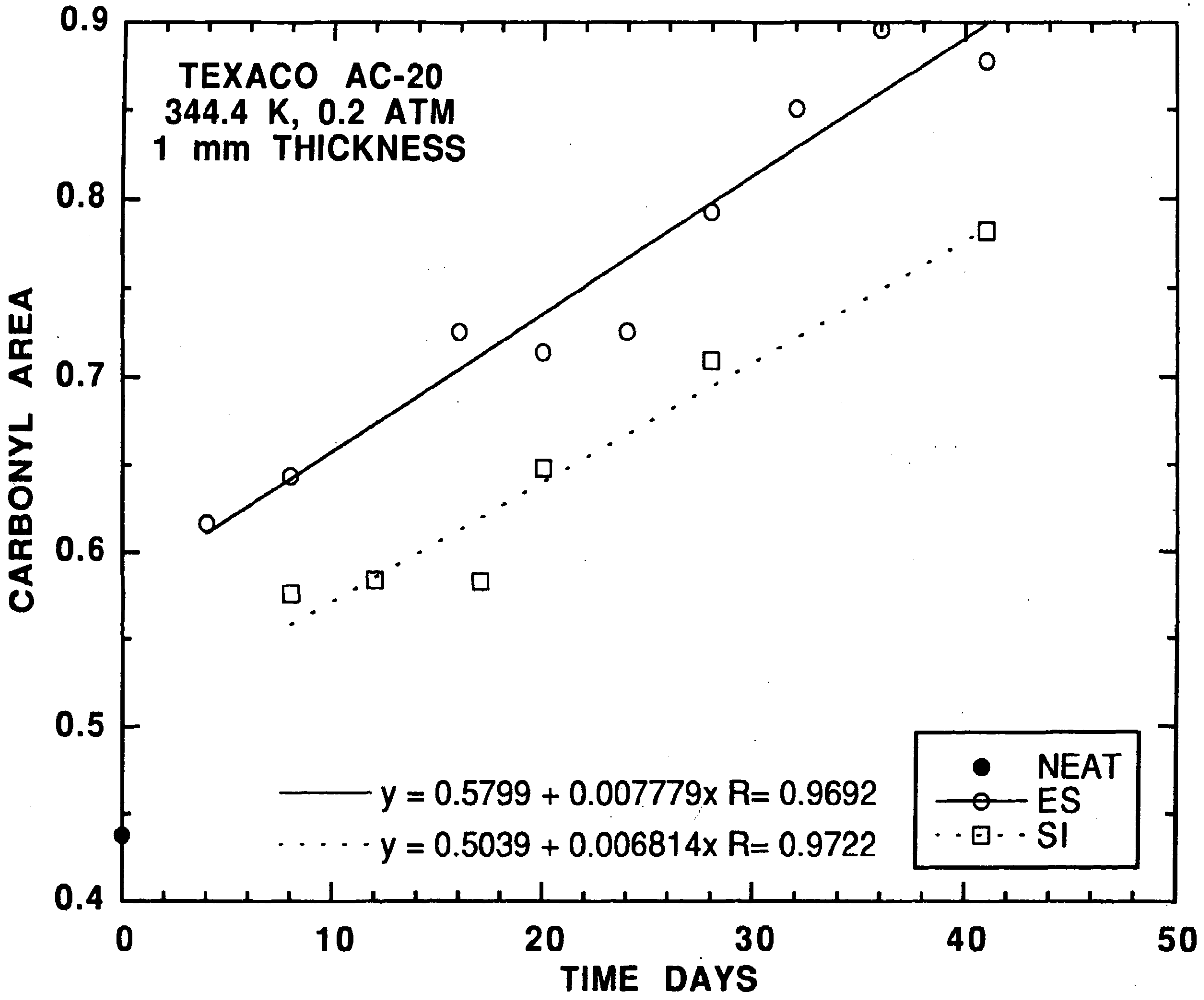

Figure C-25. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 344.4 K and P_{ES} of 0.2 atm.

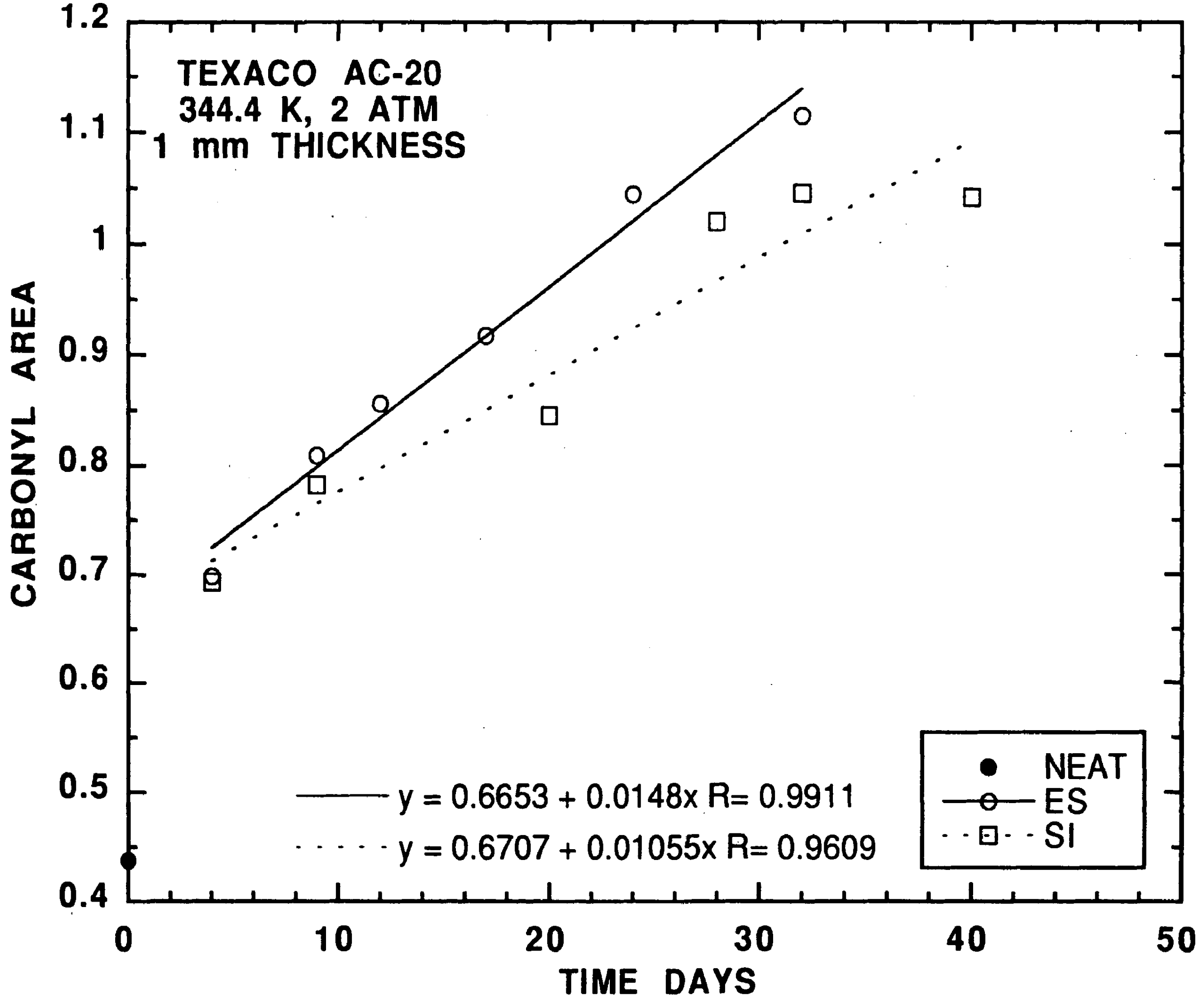

Figure C-26. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 344.4 K and P_{ES} of 2 atm.

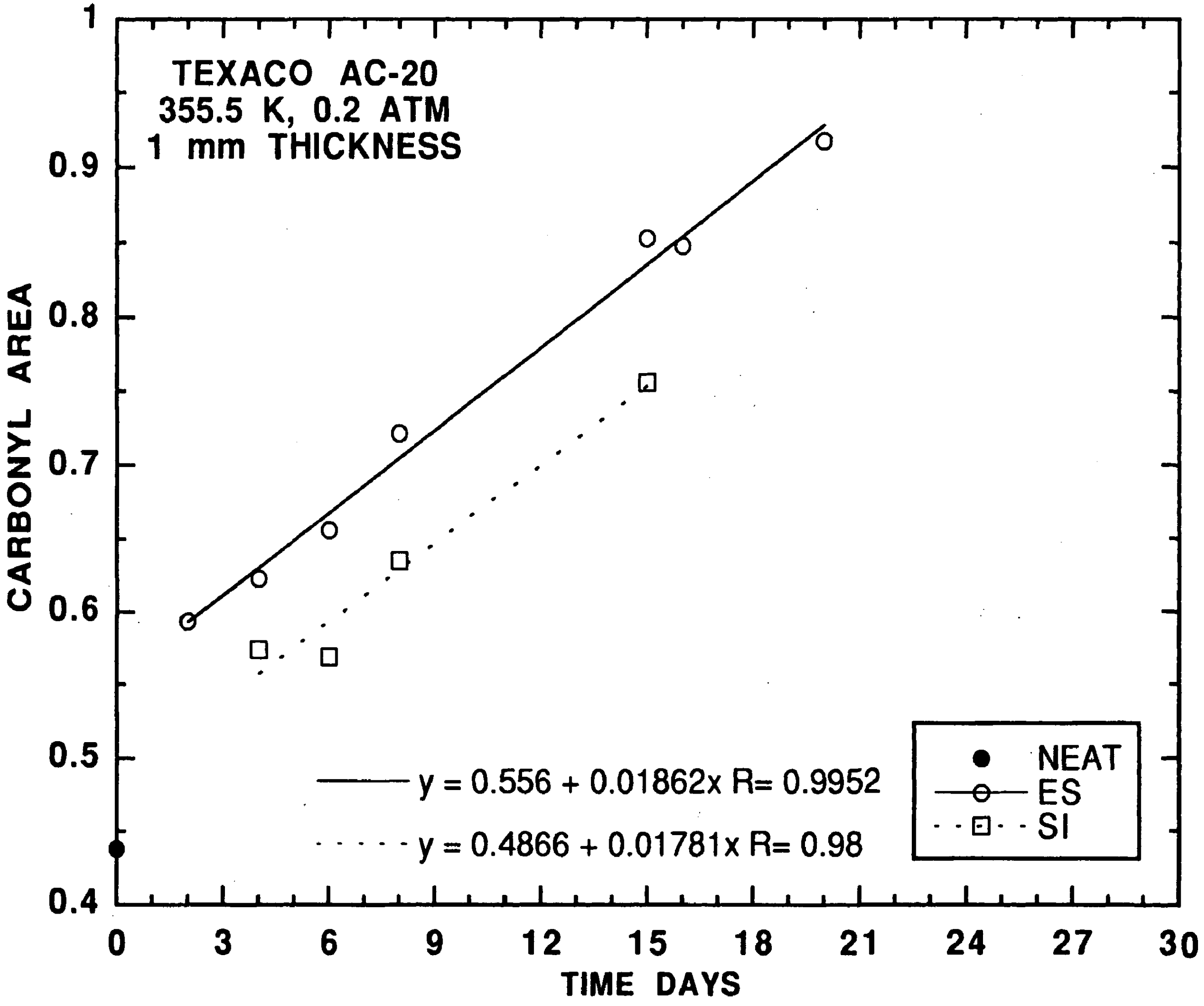

Figure C-27. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 355.5 K and P_{ES} of 0.2 atm.

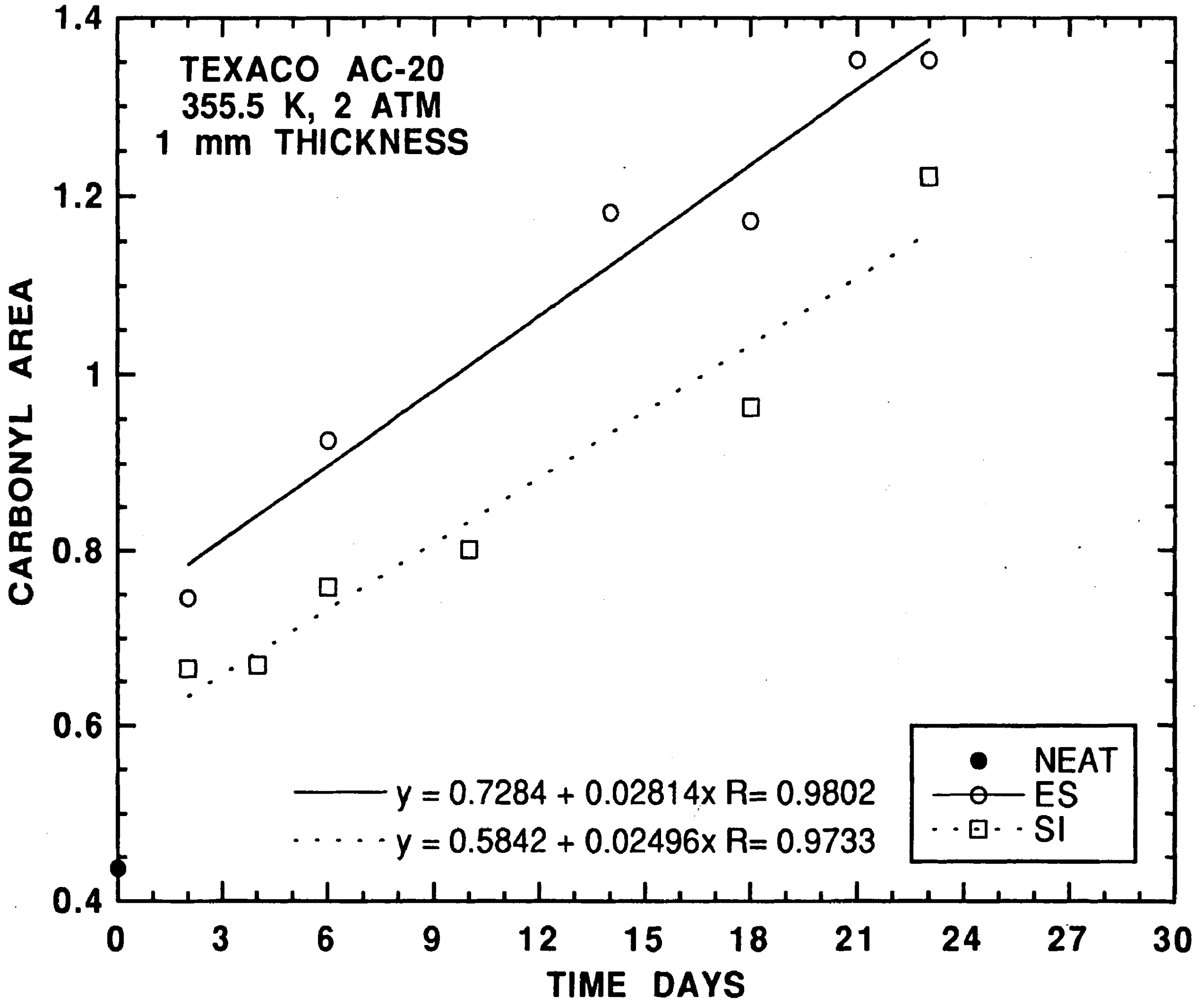

Figure C-28. *CA*s of neat and at the *ES* and *SI* of 1 mm thick film POV-aged Texaco AC-20 at 355.5 K and P_{ES} of 2 atm.

Table C-12. Measured and Estimated P and CA_o for All POV-Aging Conditions Studied[a]

	Ampet AC-20		Coastal AC-20		Cosden AC-20		Exxon AC-20		Texaco AC-20	
	P atm	CA_0 CA	P atm	CA_0 CA	P atm	CA_0 CA	P atm	CA_0 CA	P atm	CA_0 CA
Measured *ES*										
	0.2	0.643	0.2	0.611	0.2	0.651	0.2	0.726	0.2	0.560
	2	0.716	2	0.764	2	0.805	2	0.815	2	0.697
	20	0.766	20	0.935	20	1.242	20	1.018	20	0.818
Estimated *SI*										
	0.0673	0.514	0.149	0.555	0.0331	0.603	0.175	0.543	0.123	0.494
	0.115	0.576	0.0345*	0.605*	0.013	0.518	0.107	0.606	0.122	0.504
	0.161	0.554	0.0789	0.485	–	–	0.0776*	0.735*	0.170	0.486
	0.393*	0.756*	0.925	0.754	0.937	0.832	1.090	0.781	1.380	0.627
	0.857	0.686	0.407	0.661	0.363	0.759	1.160	0.786	0.570*	0.671*
	1.800	0.633	1.105*	0.626*	1.480	0.709	1.860	0.668	1.280	0.584

[a] – Signifies values were not determined

* Data removed from parameter estimation in Figure V-11

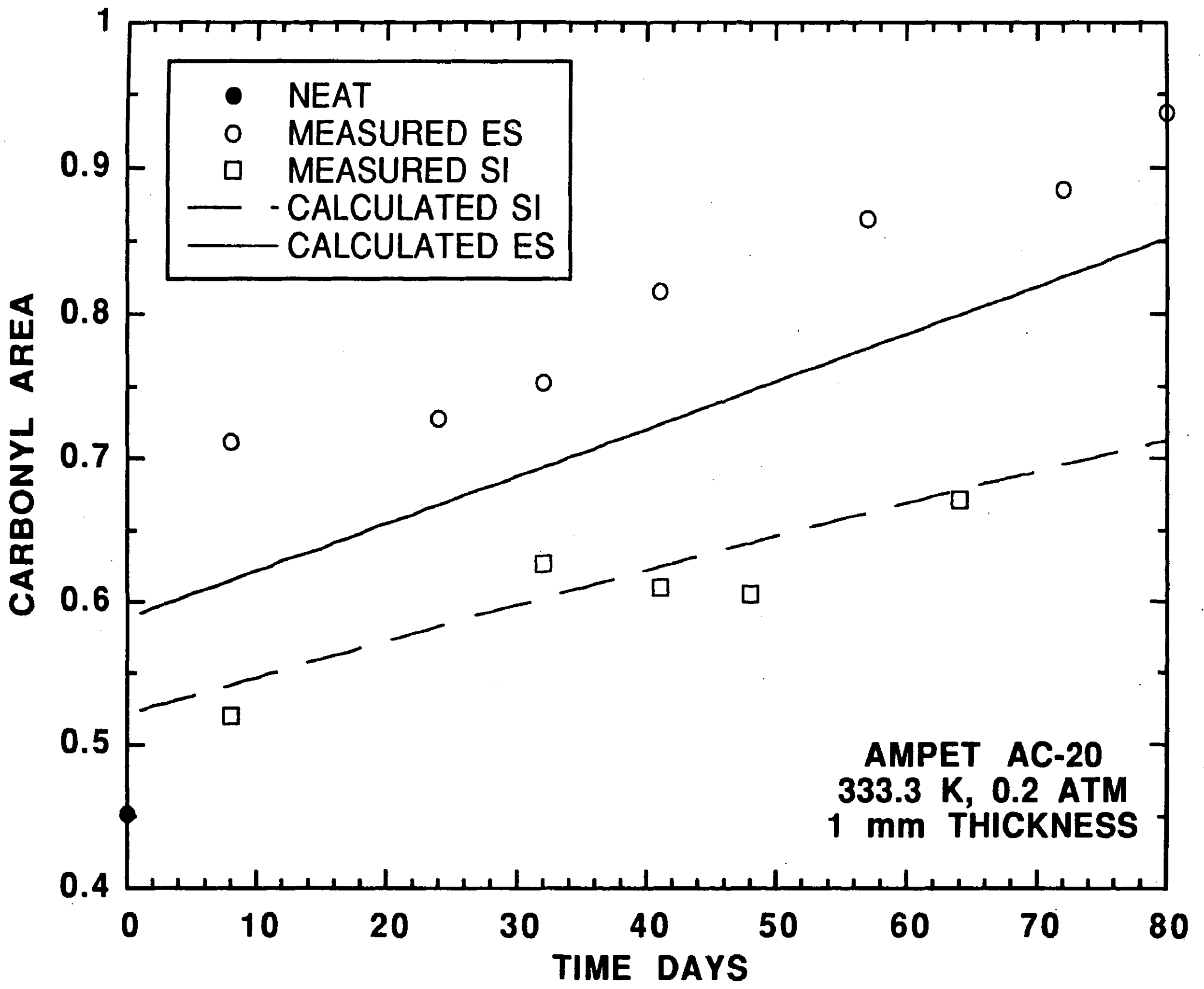

Figure C-29. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 333.3 K and P_{ES} of 0.2 atm.

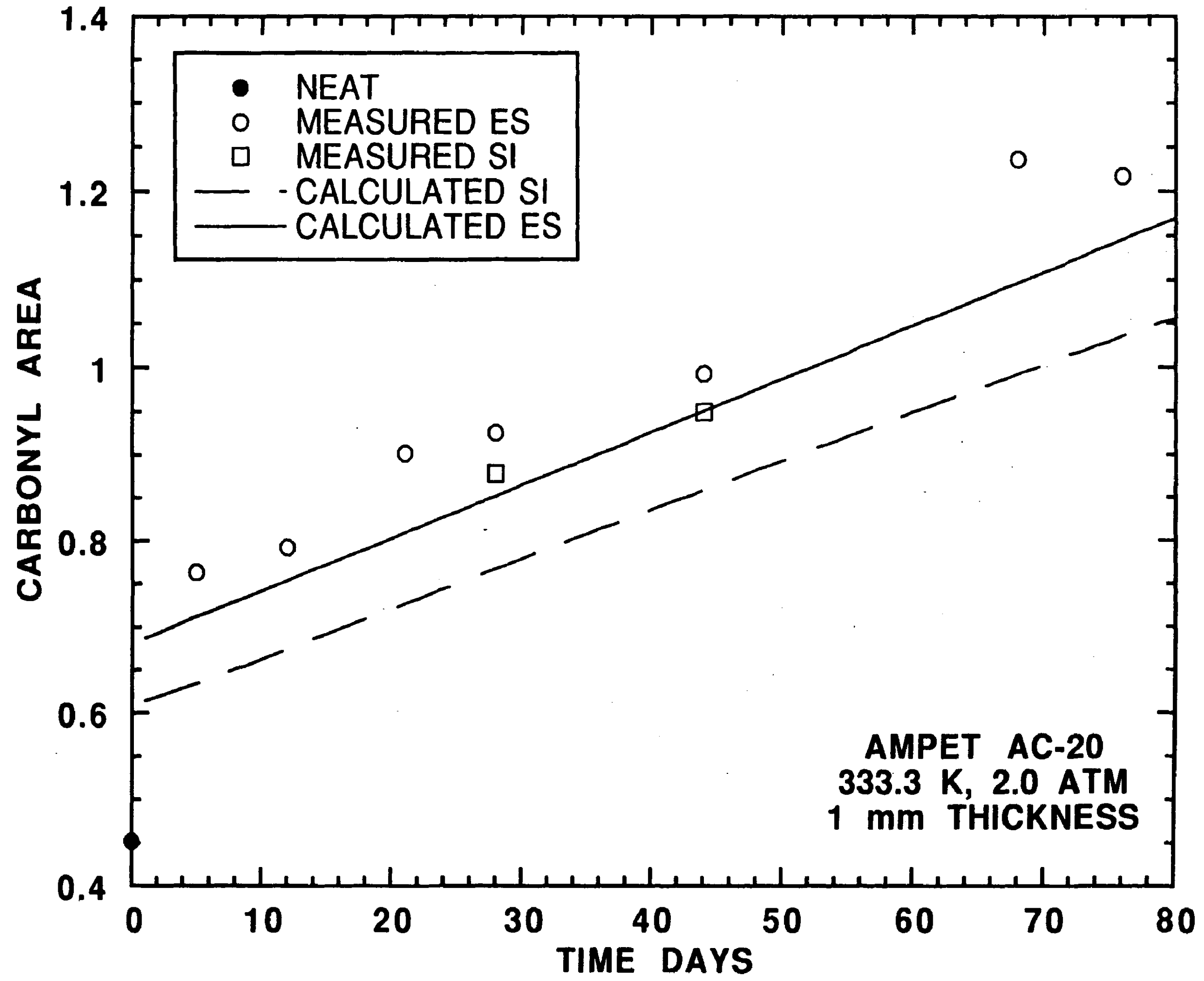

Figure C-30. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 333.3 K and P_{ES} of 2 atm.

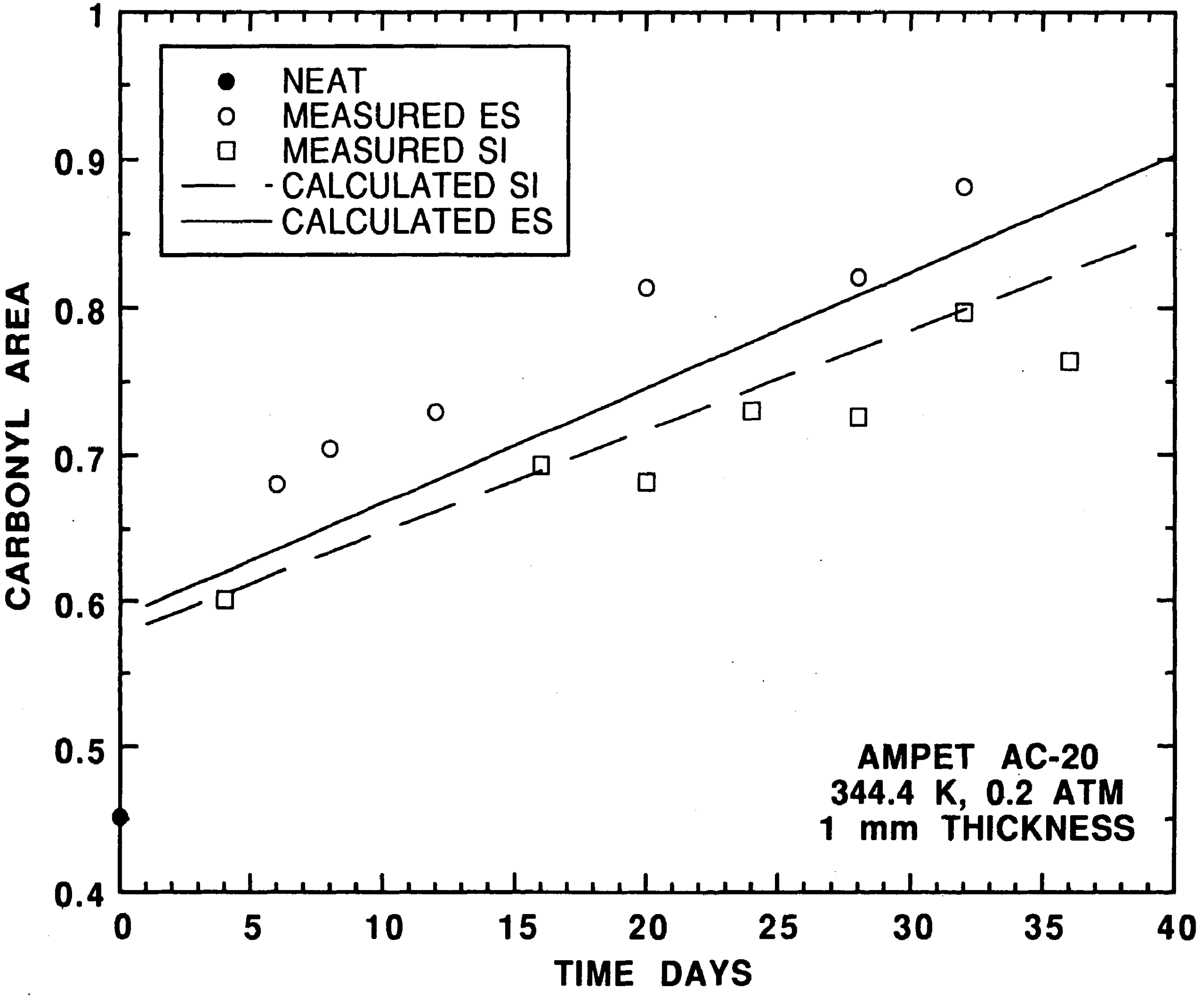

Figure C-31. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 344.4 K and P_{ES} of 0.2 atm.

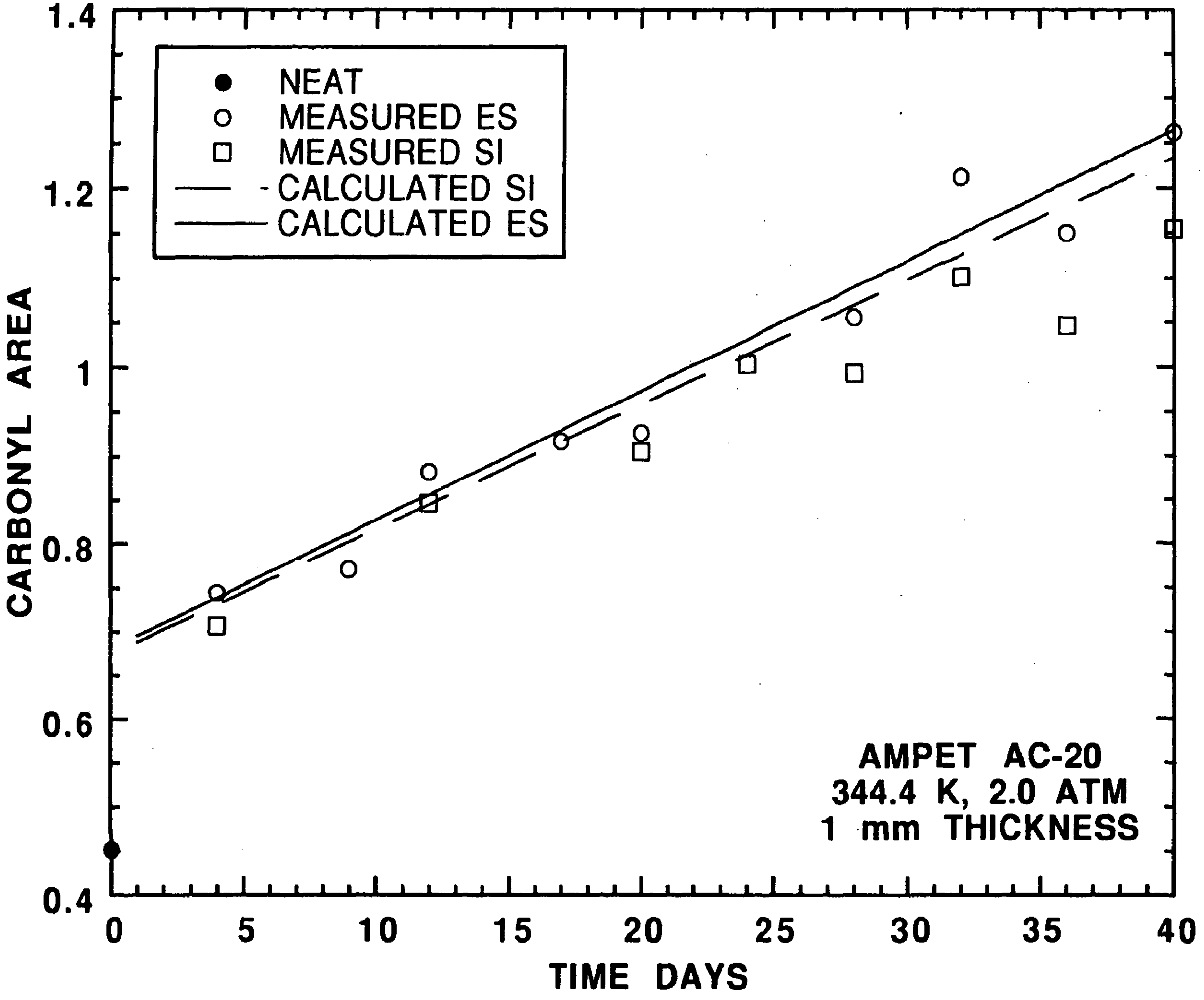

Figure C-32. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 344.4 K and P_{ES} of 2 atm.

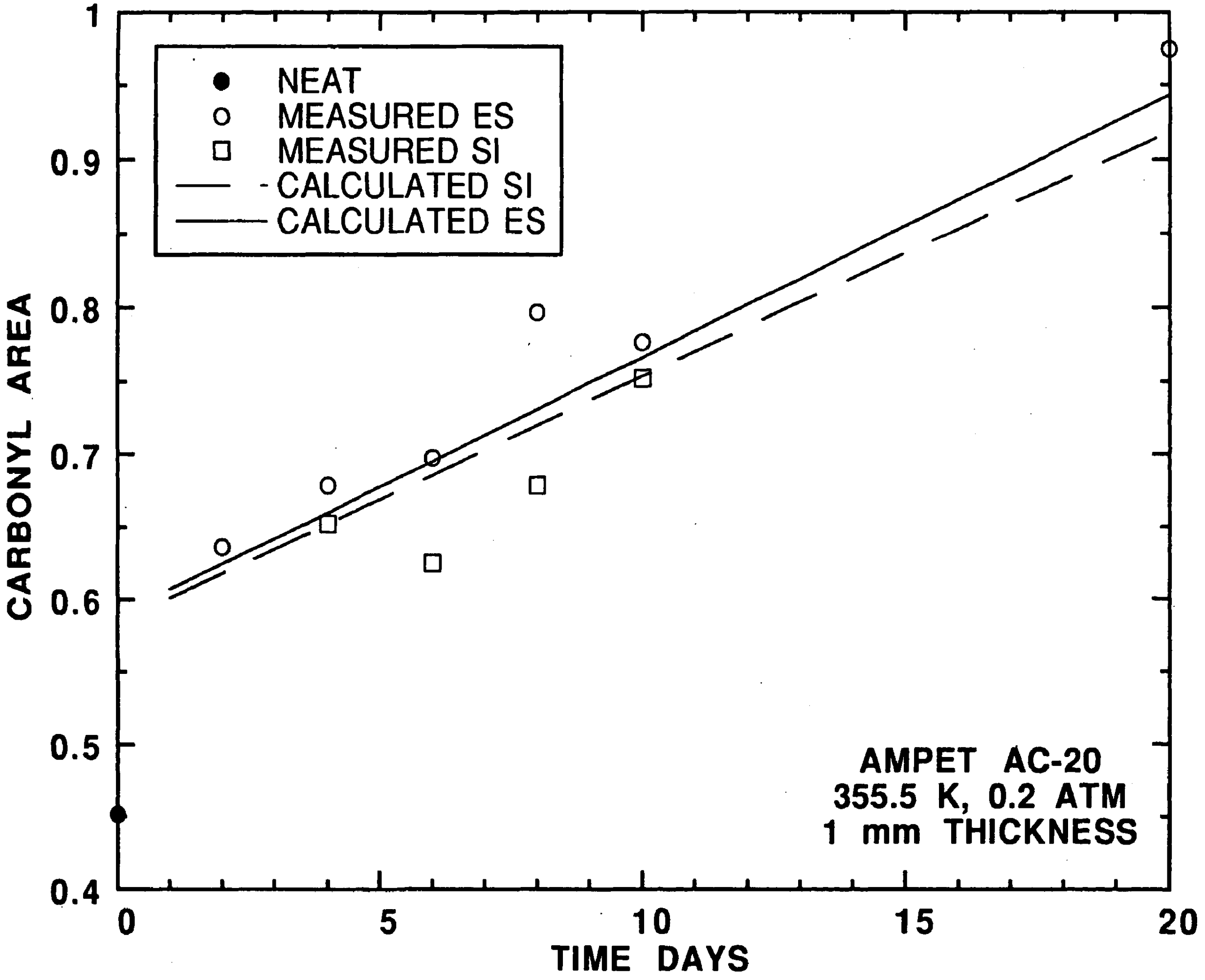

Figure C-33. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 355.5 K and P_{ES} of 0.2 atm.

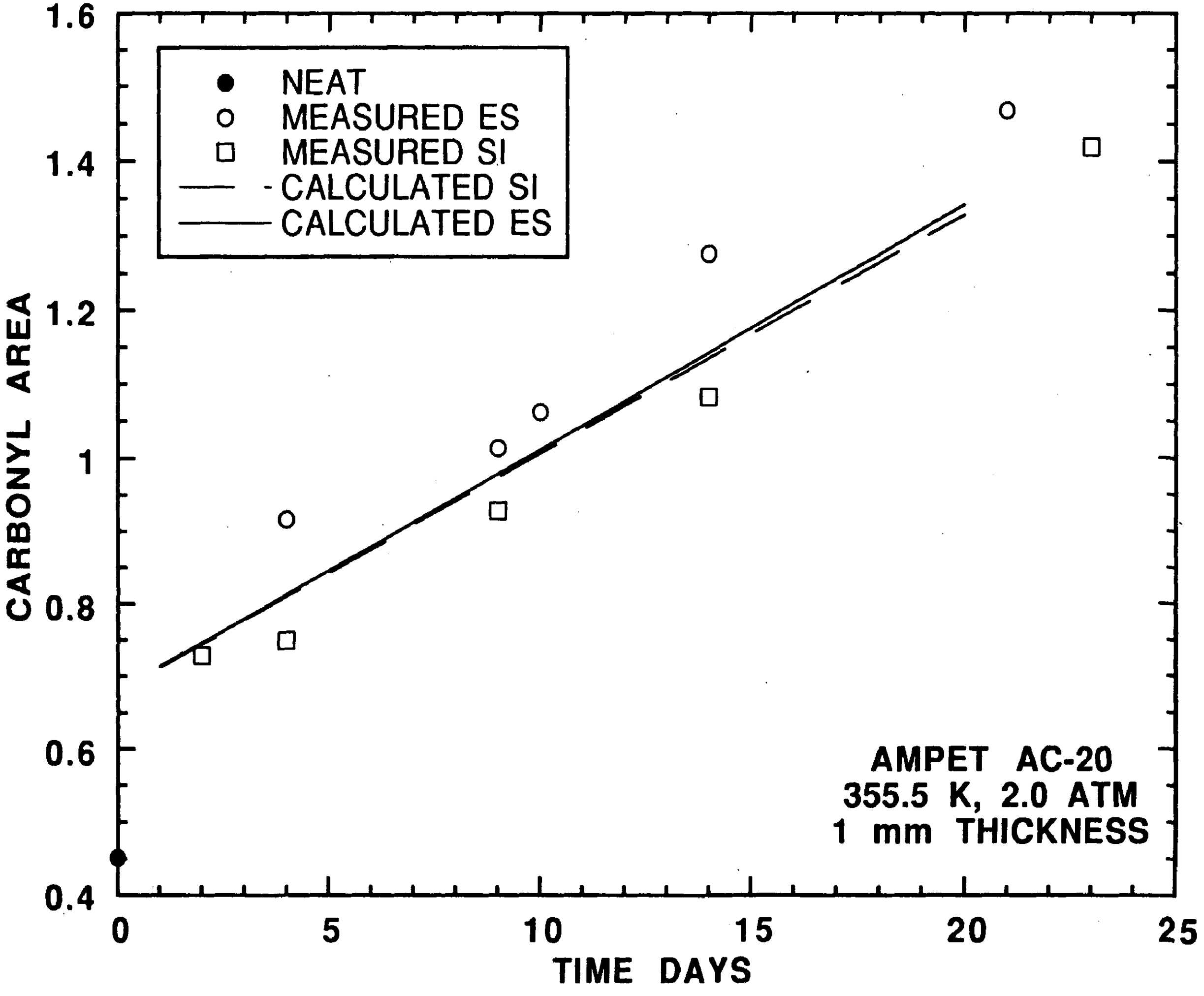

Figure C-34. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Ampet AC-20 at 355.5 K and P_{ES} of 2 atm.

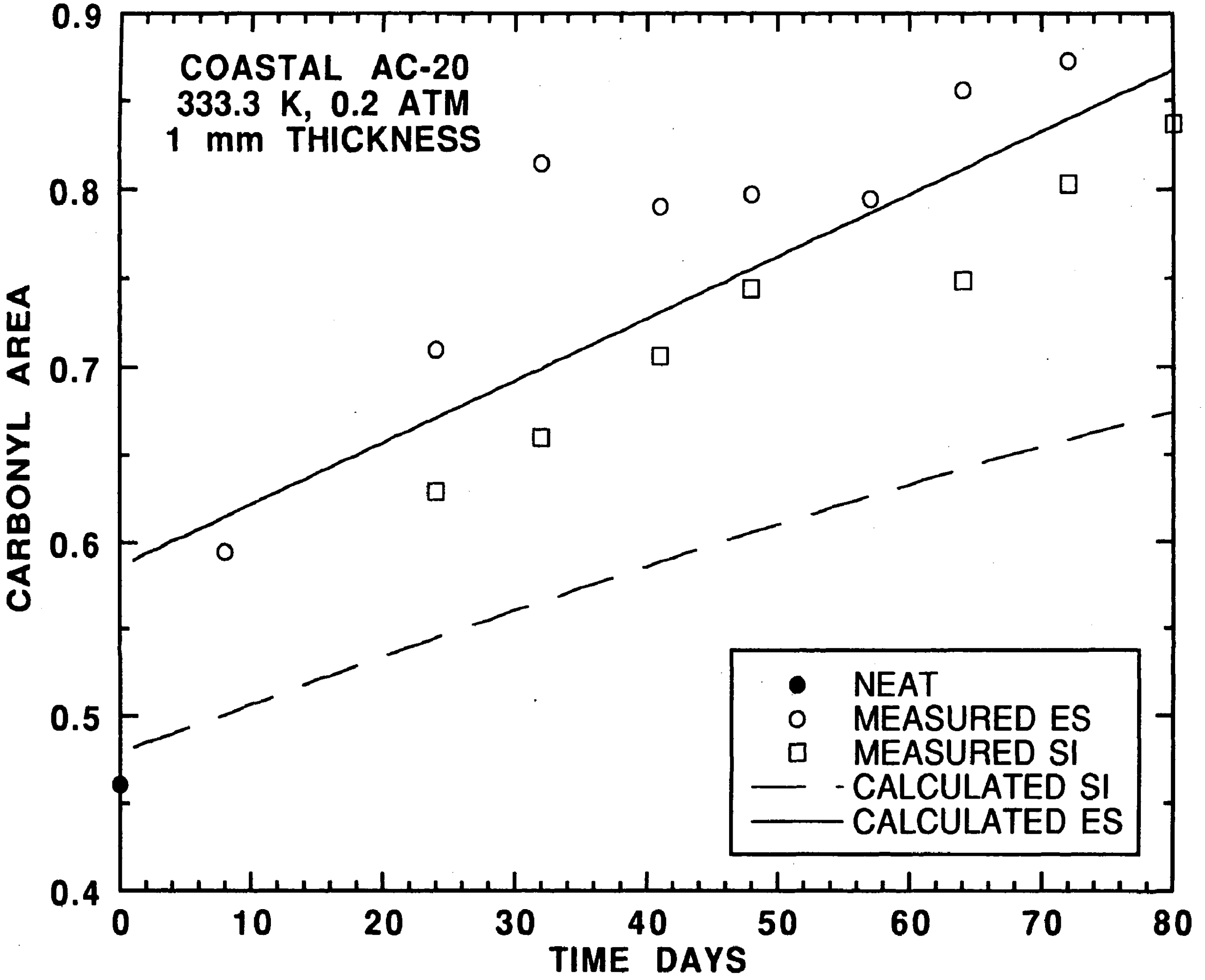

Figure C-35. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Coastal AC-20 at 333.3 K and P_{ES} of 0.2 atm.

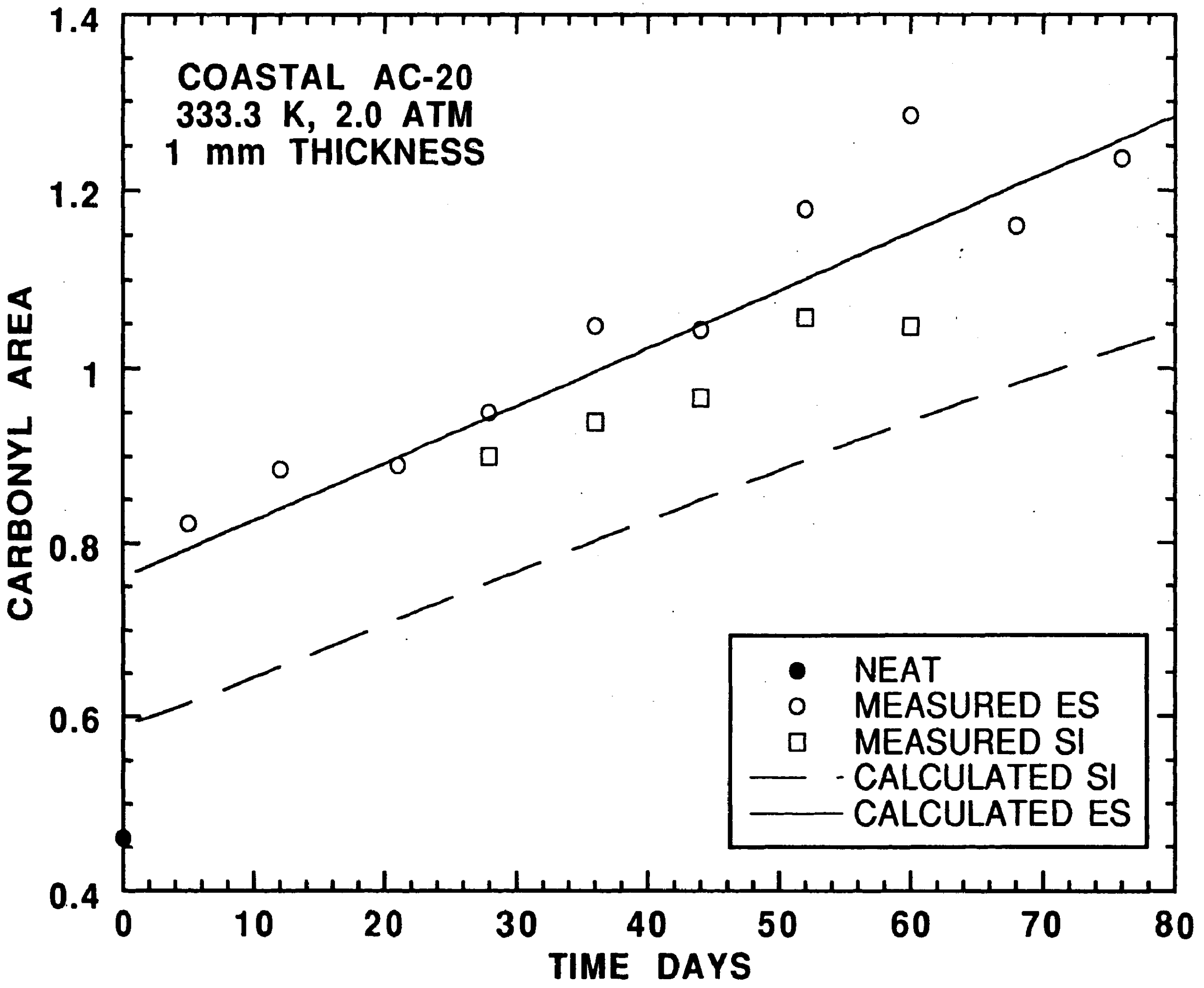

Figure C-36. Comparisons between measured and calculated CA at the ES and SI of 1 mm thick POV-aged Coastal AC-20 at 333.3 K and P_{ES} of 2 atm.

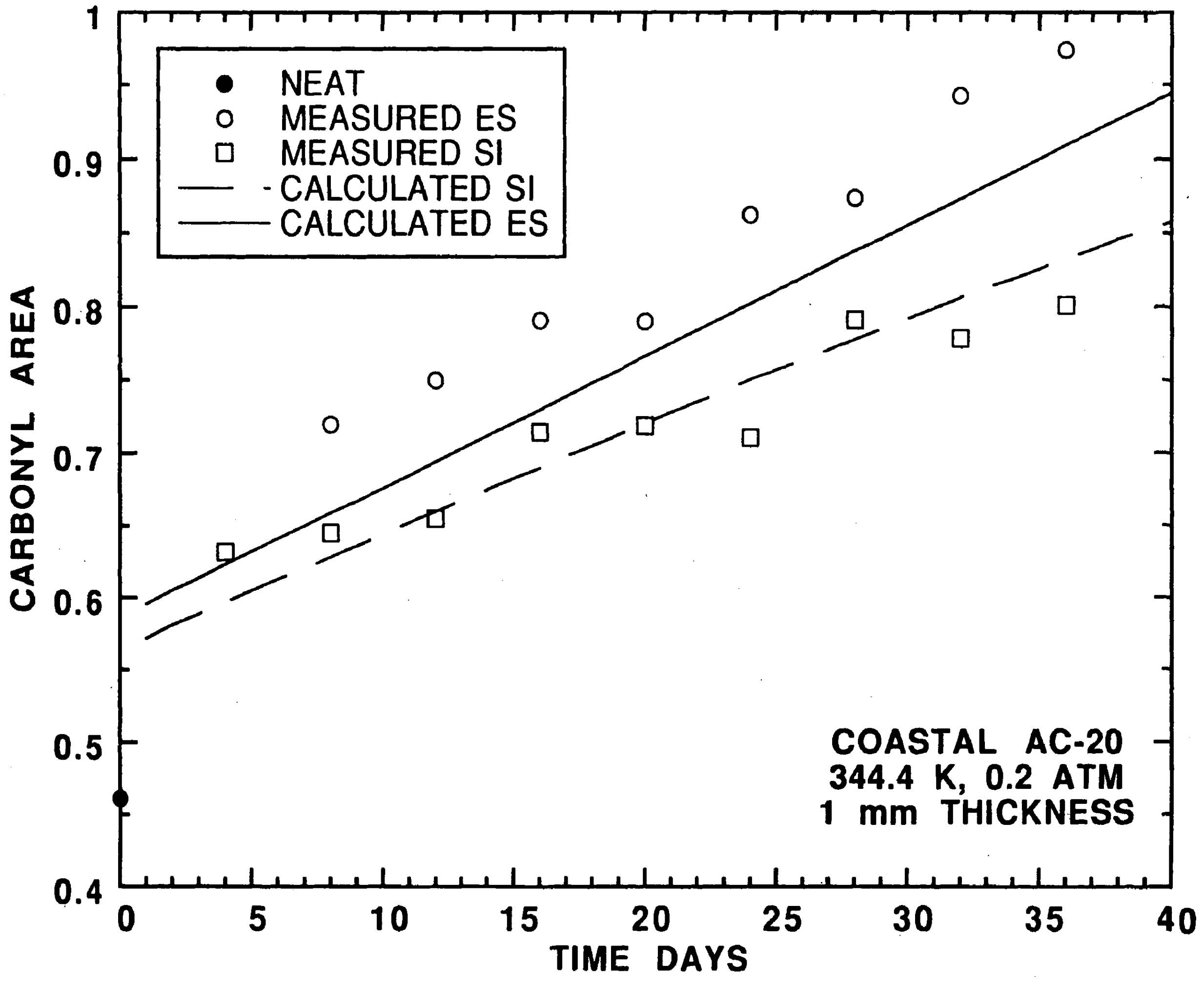

Figure C-37. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Coastal AC-20 at 344.4 K and P_{ES} of 0.2 atm.

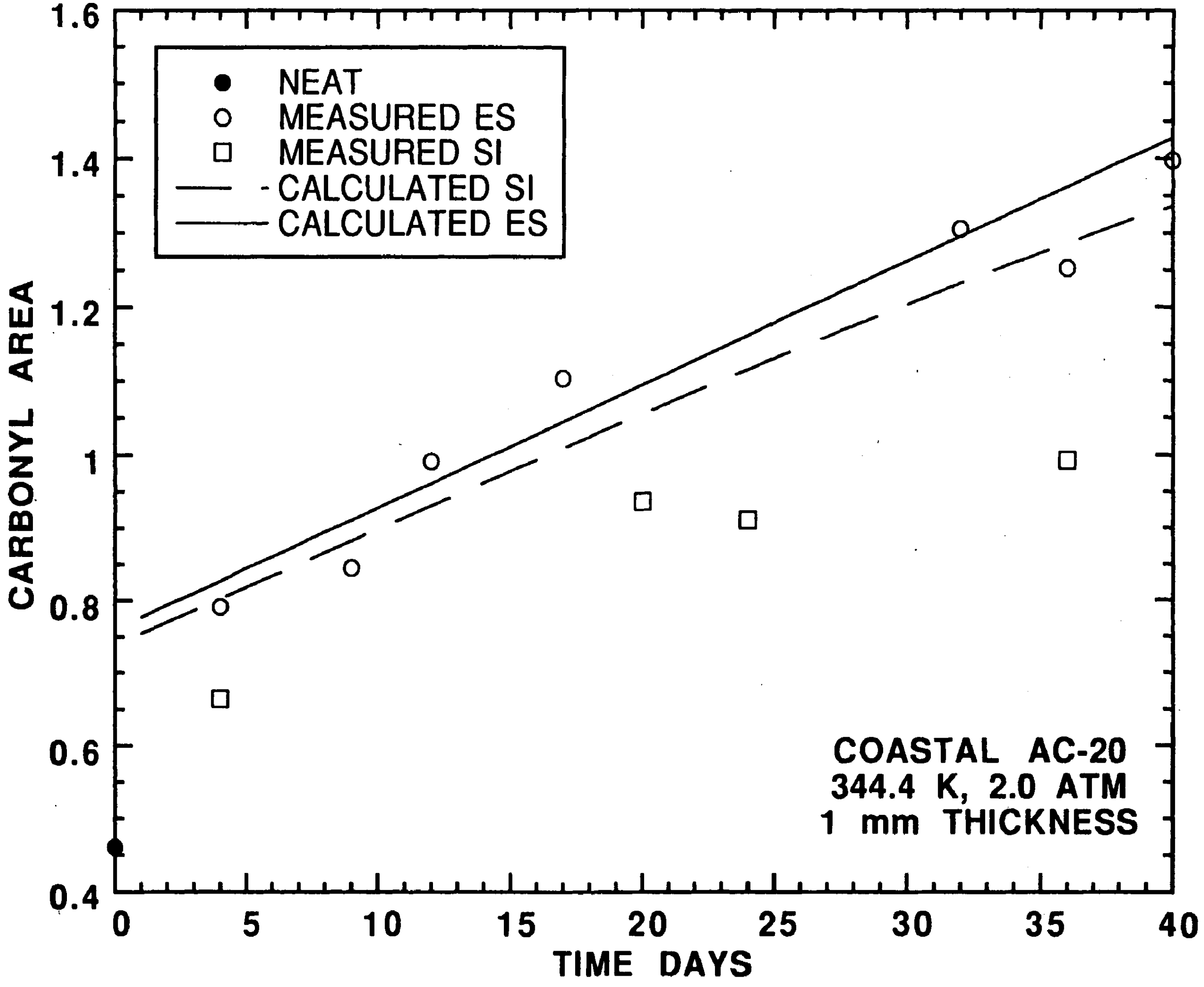

Figure C-38. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Coastal AC-20 at 344.4 K and P_{ES} of 2 atm.

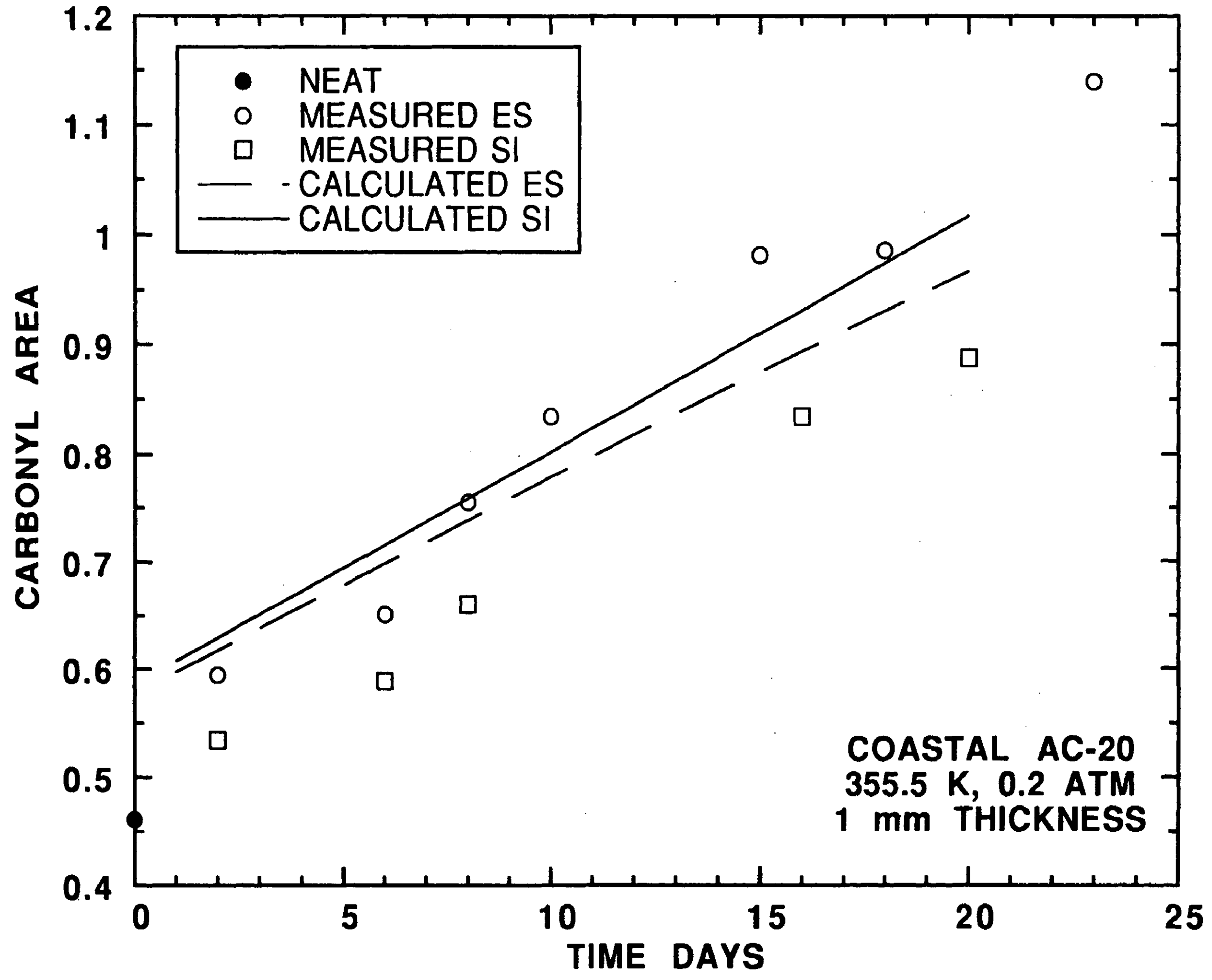

Figure C-39. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Coastal AC-20 at 355.5 K and P_{ES} of 0.2 atm.

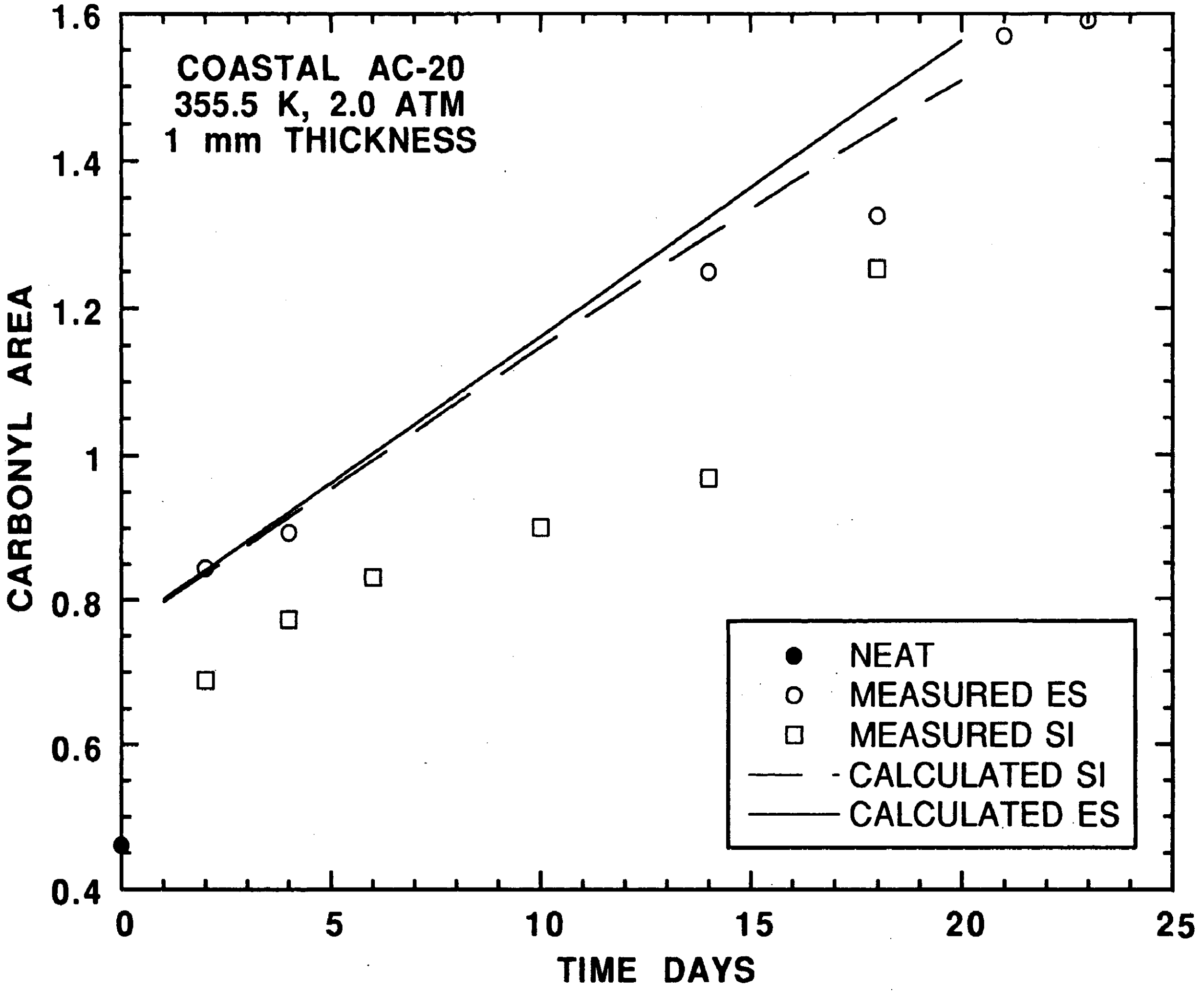

Figure C-40. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Coastal AC-20 at 355.5 K and P_{ES} of 2 atm.

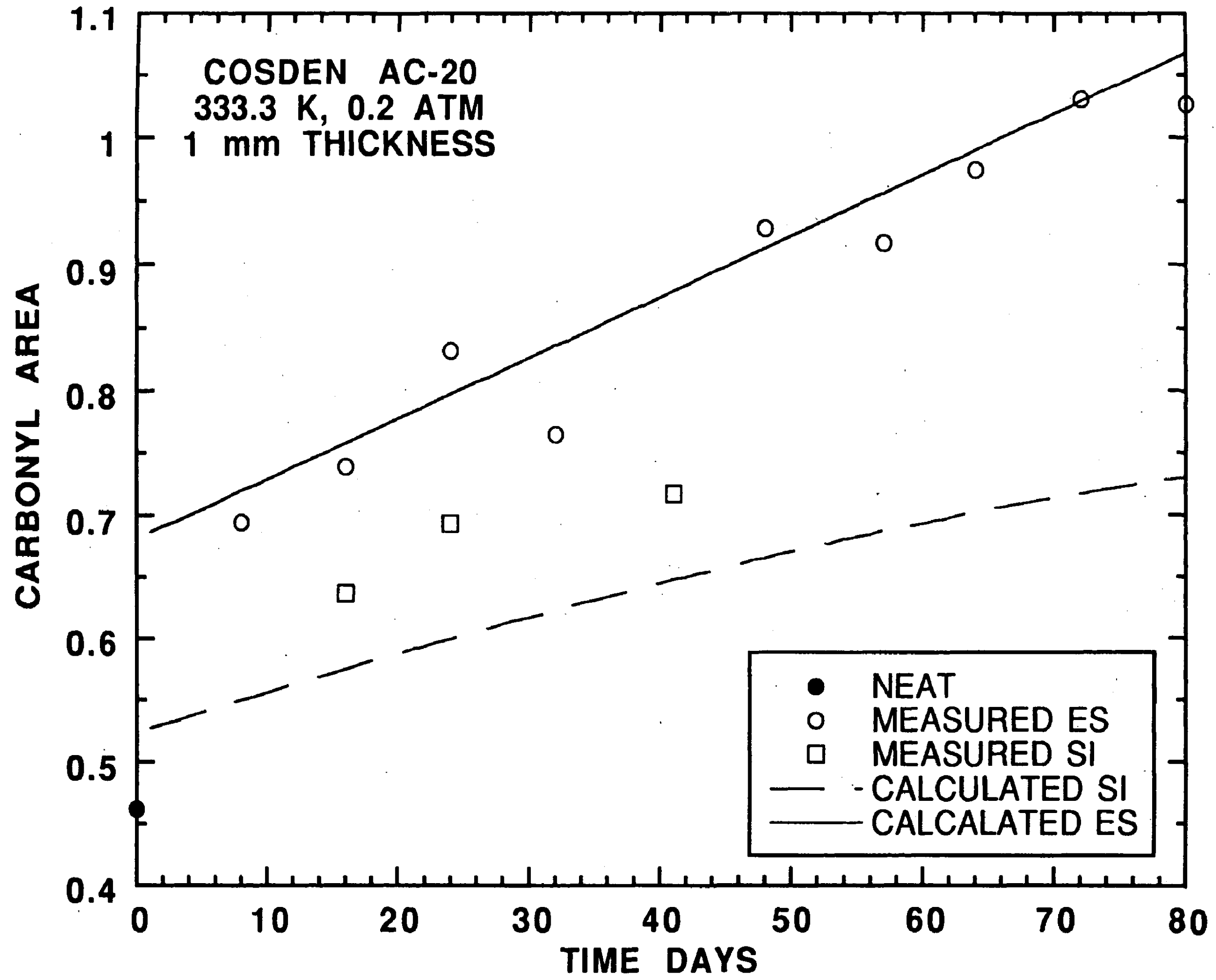

Figure C-41. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 333.3 K and P_{ES} of 0.2 atm.

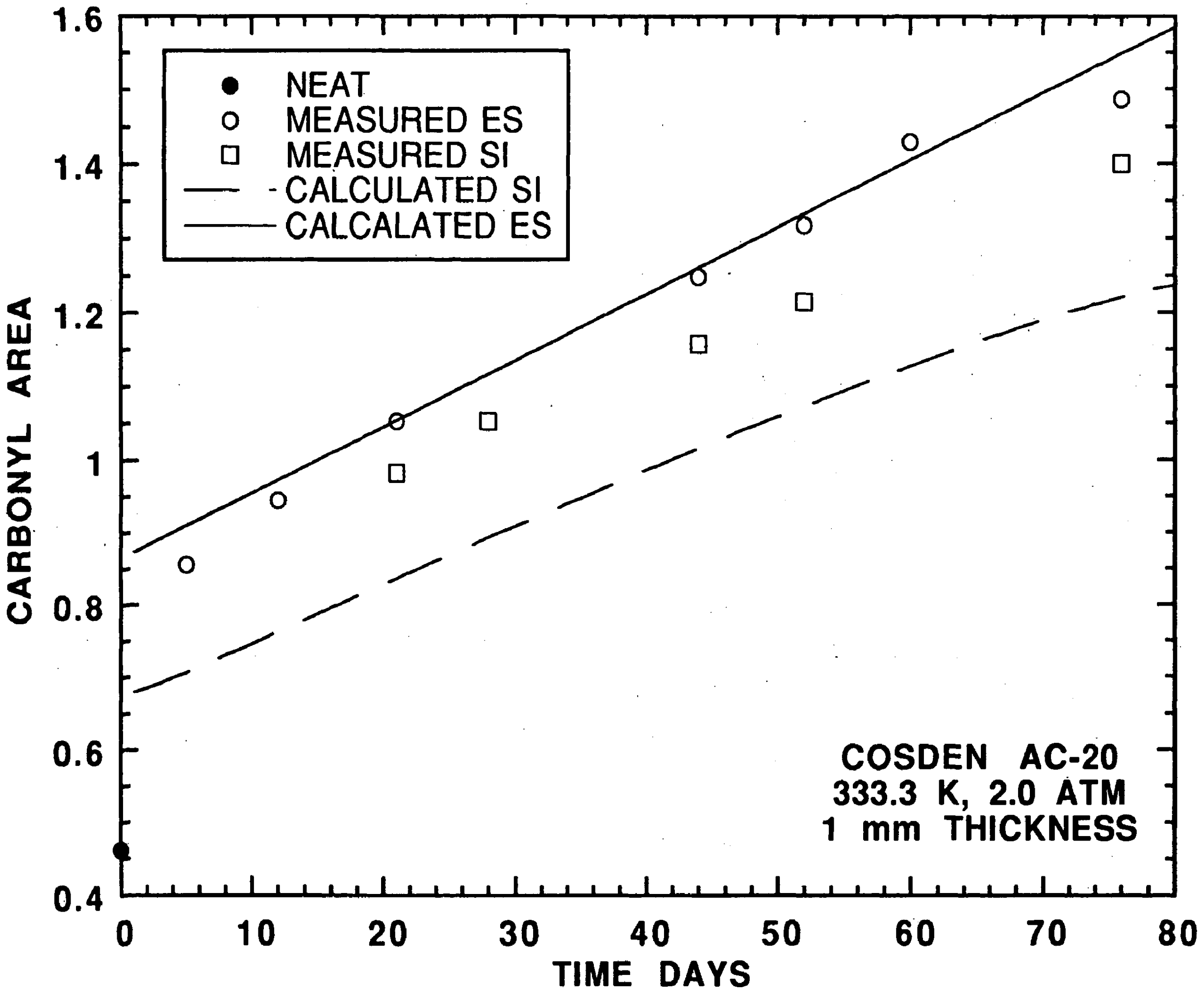

Figure C-42. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 333.3 K and P_{ES} of 2 atm.

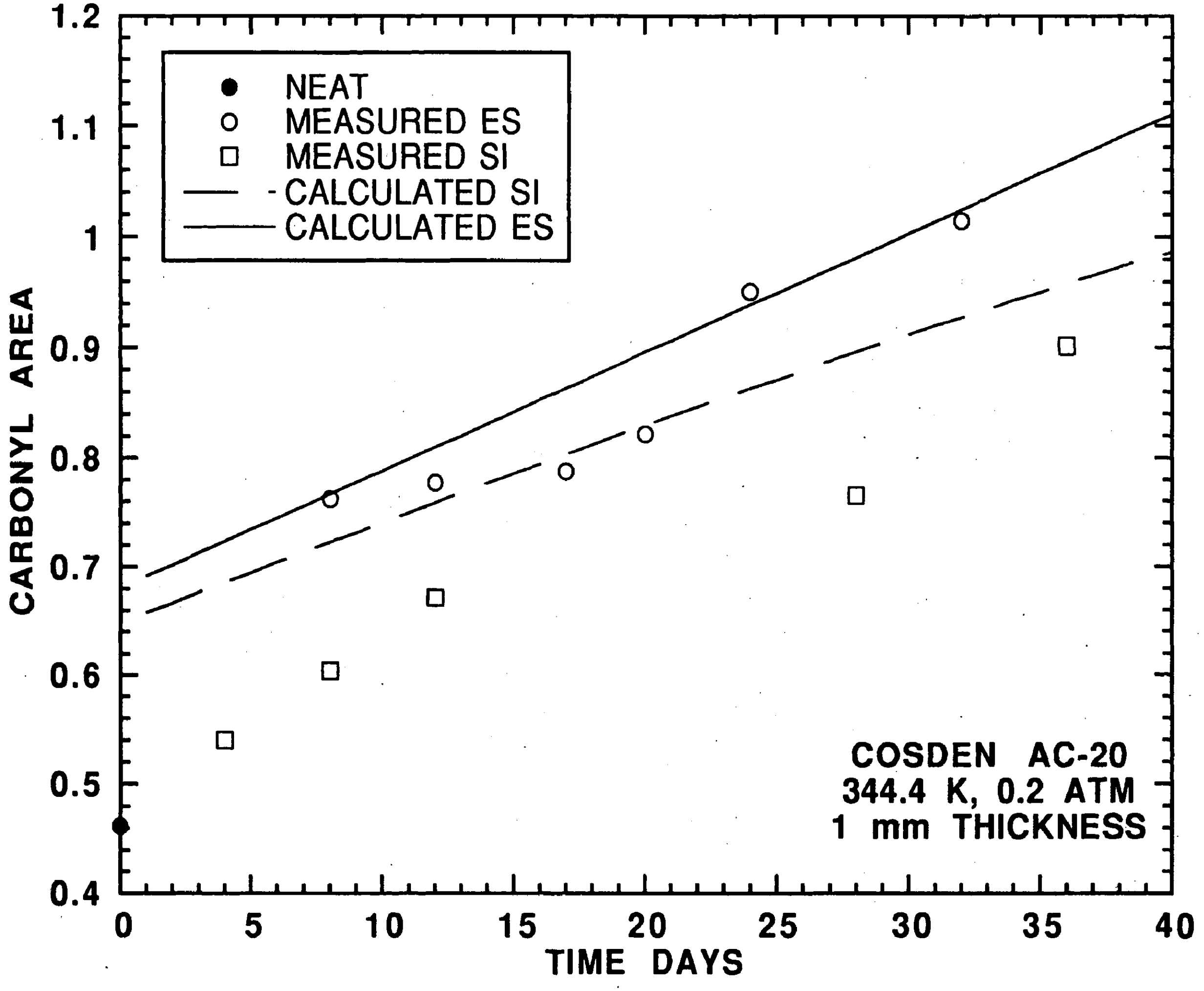

Figure C-43. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 344.4 K and P_{ES} of 0.2 atm.

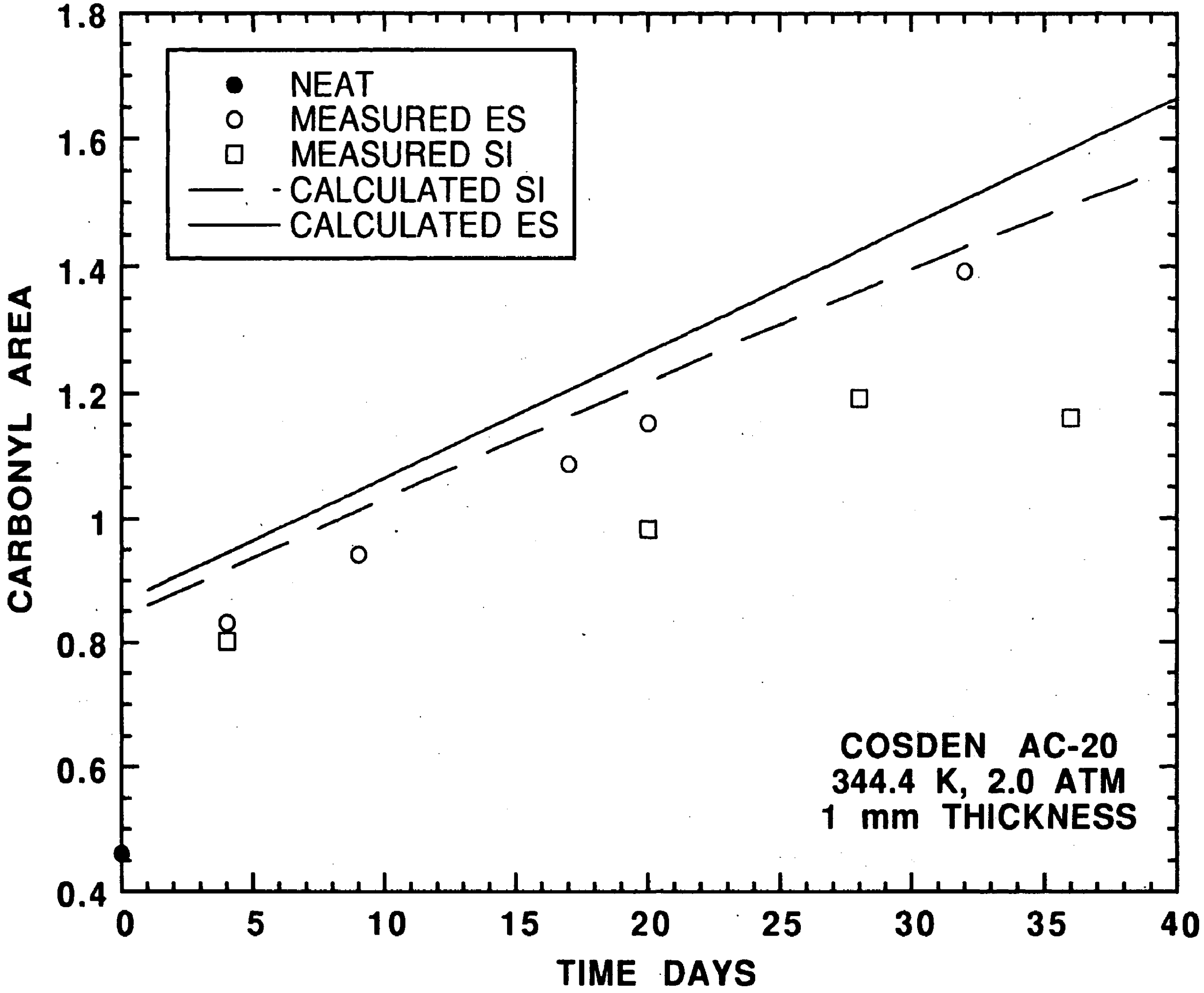

Figure C-44. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 344.4 K and P_{ES} of 2 atm.

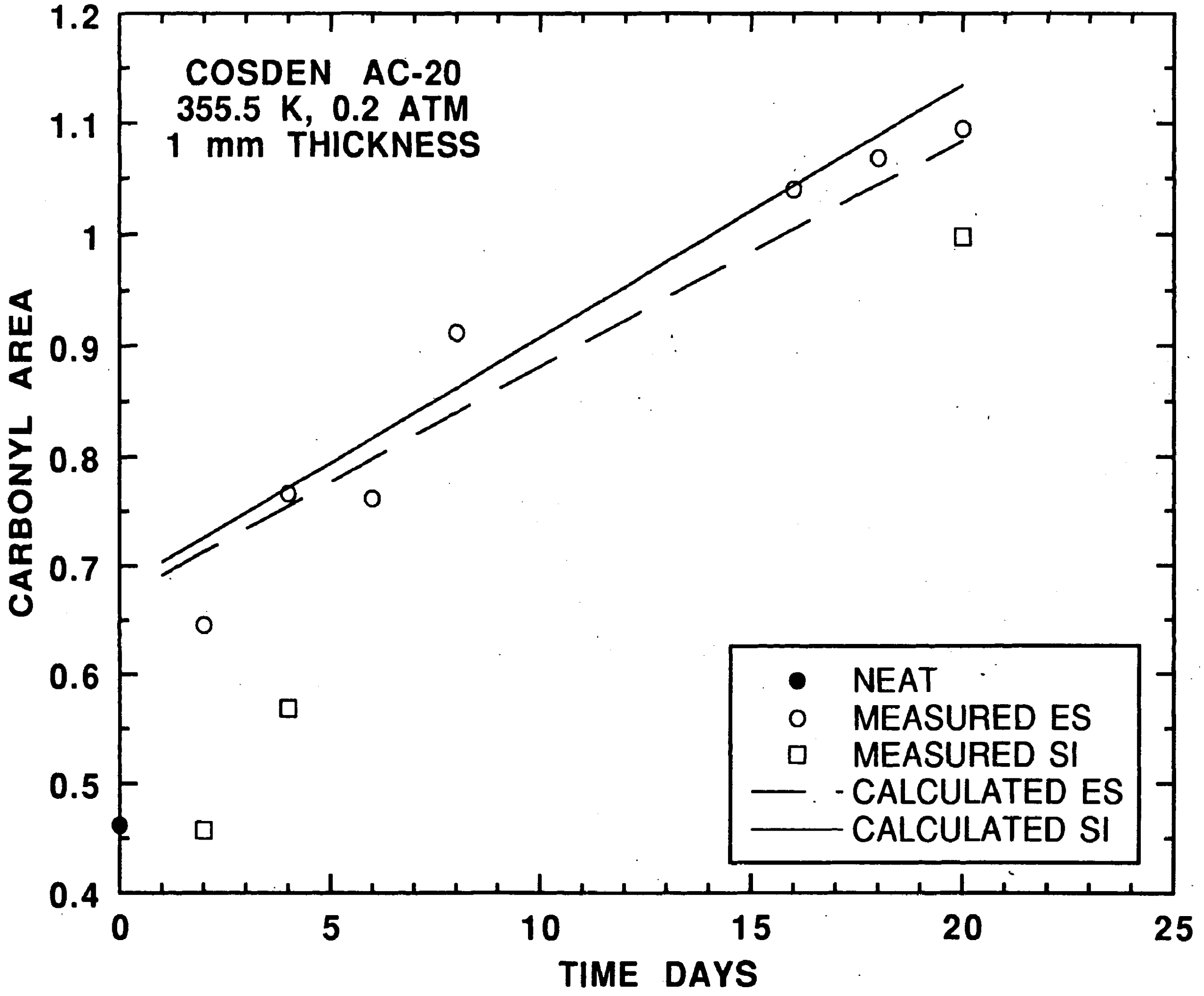

Figure C-45. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 355.5 K and P_{ES} of 0.2 atm.

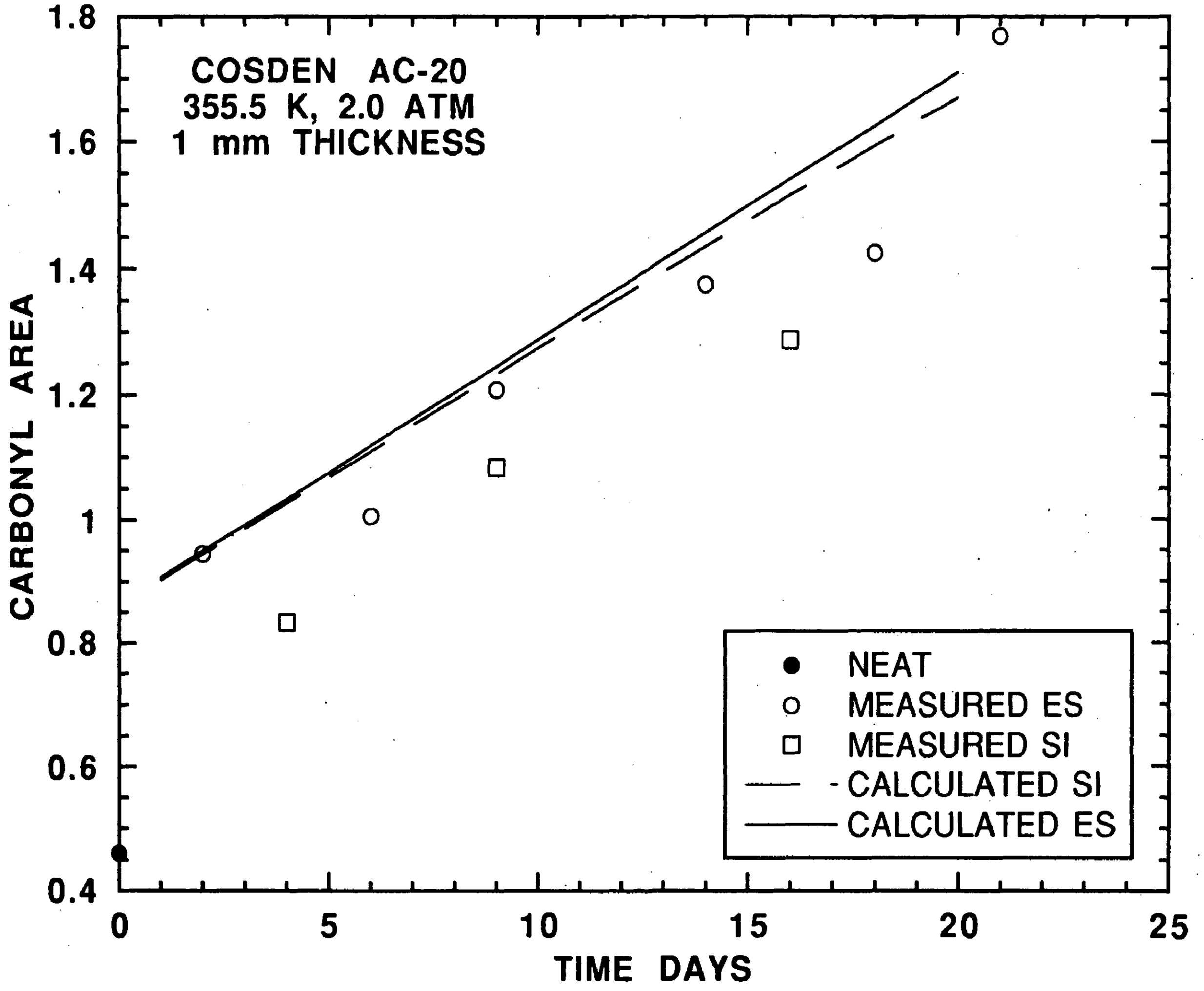

Figure C-46. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Cosden AC-20 at 355.5 K and P_{ES} of 2 atm.

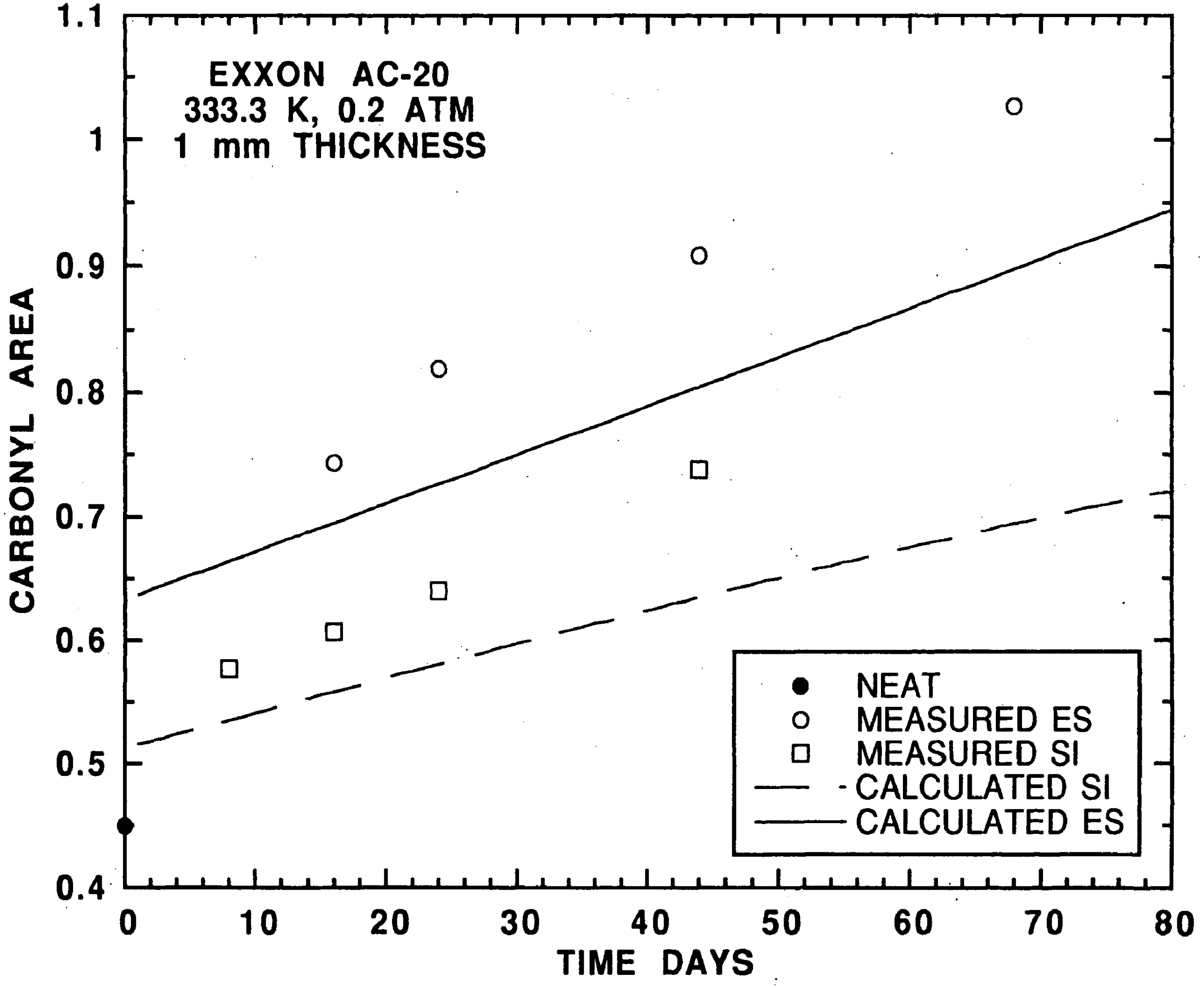

Figure C-47. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 333.3 K and P_{ES} of 0.2 atm.

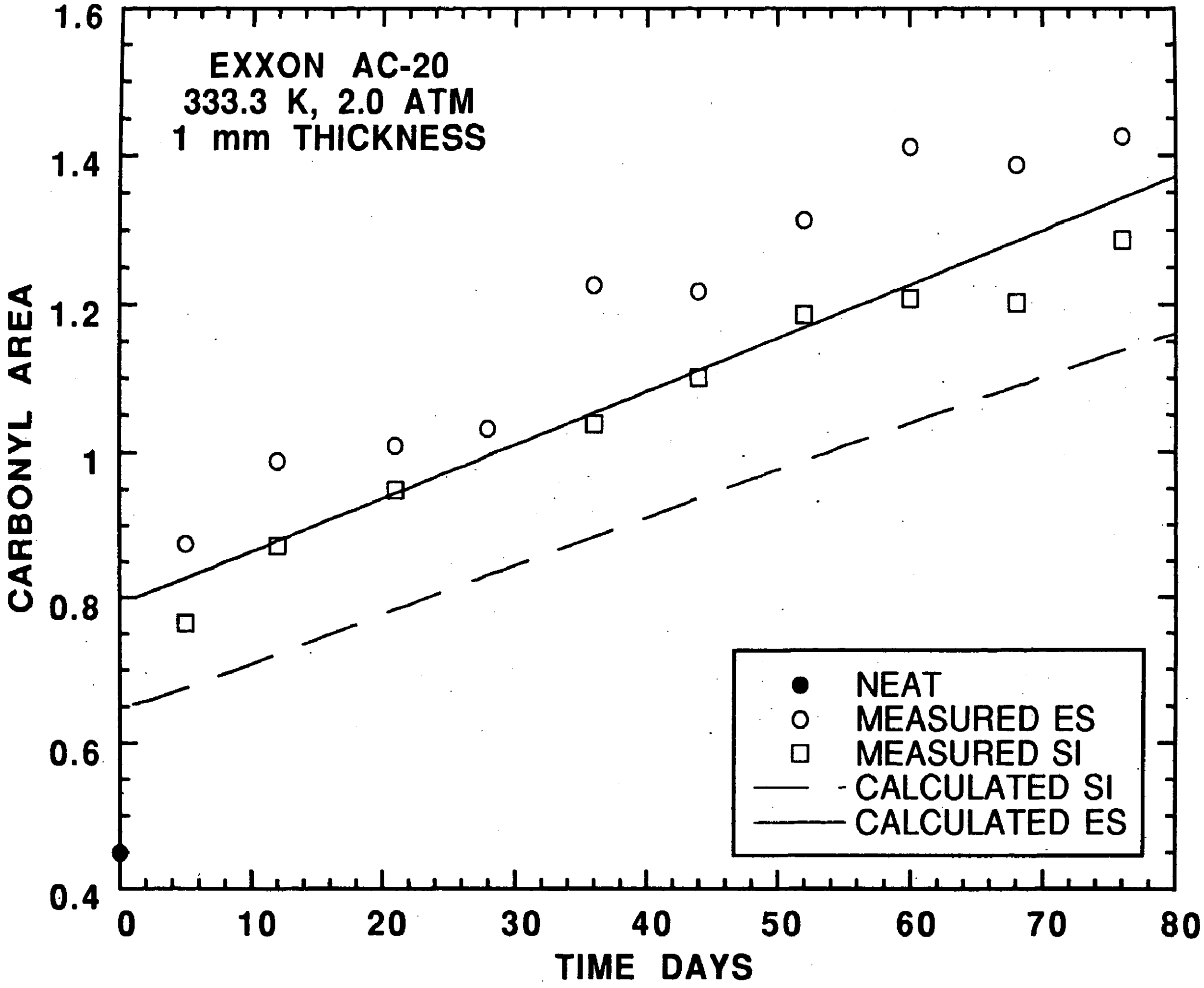

Figure C-48. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 333.3 K and P_{ES} of 2 atm.

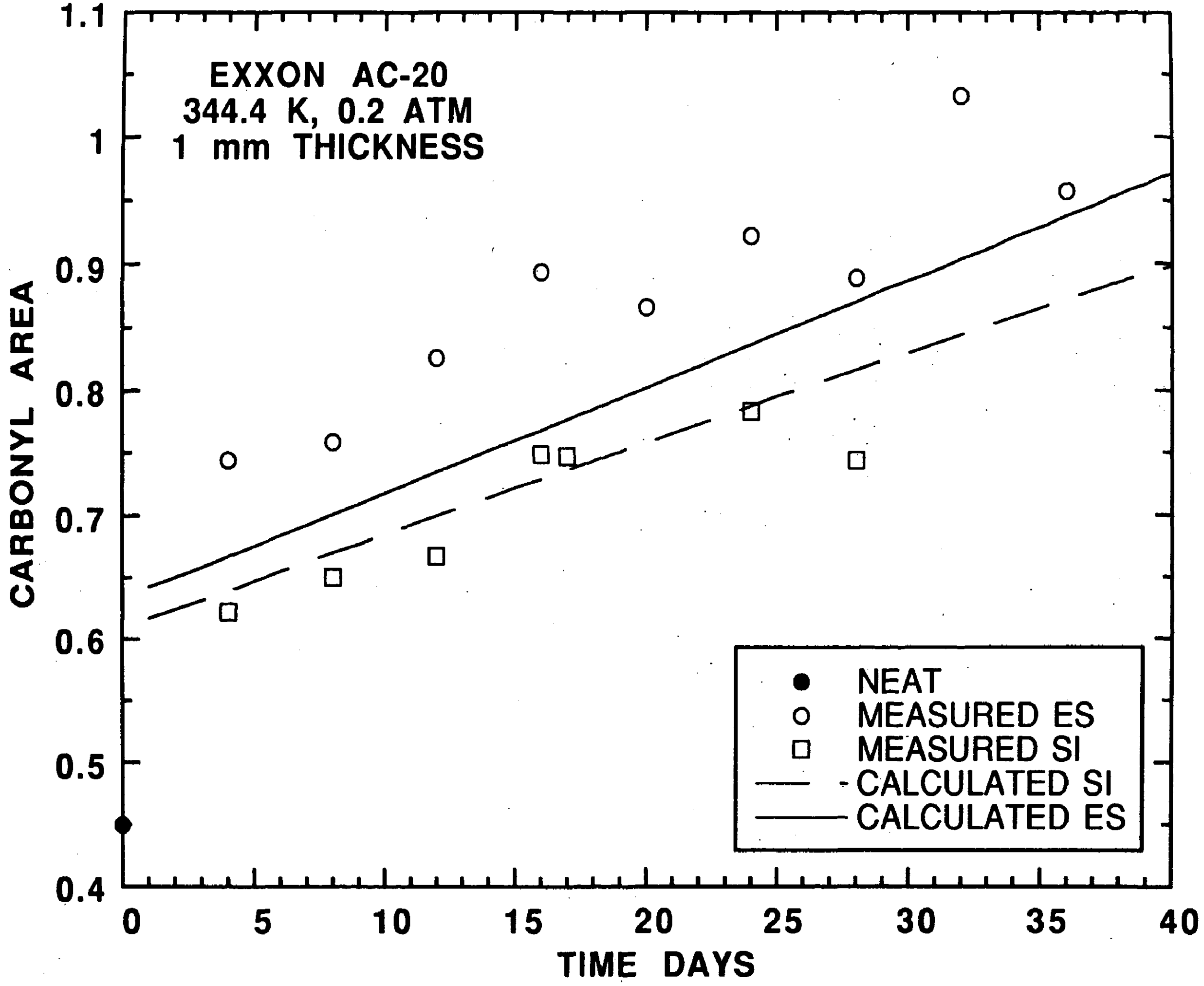

Figure C-49. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 344.4 K and P_{ES} of 0.2 atm.

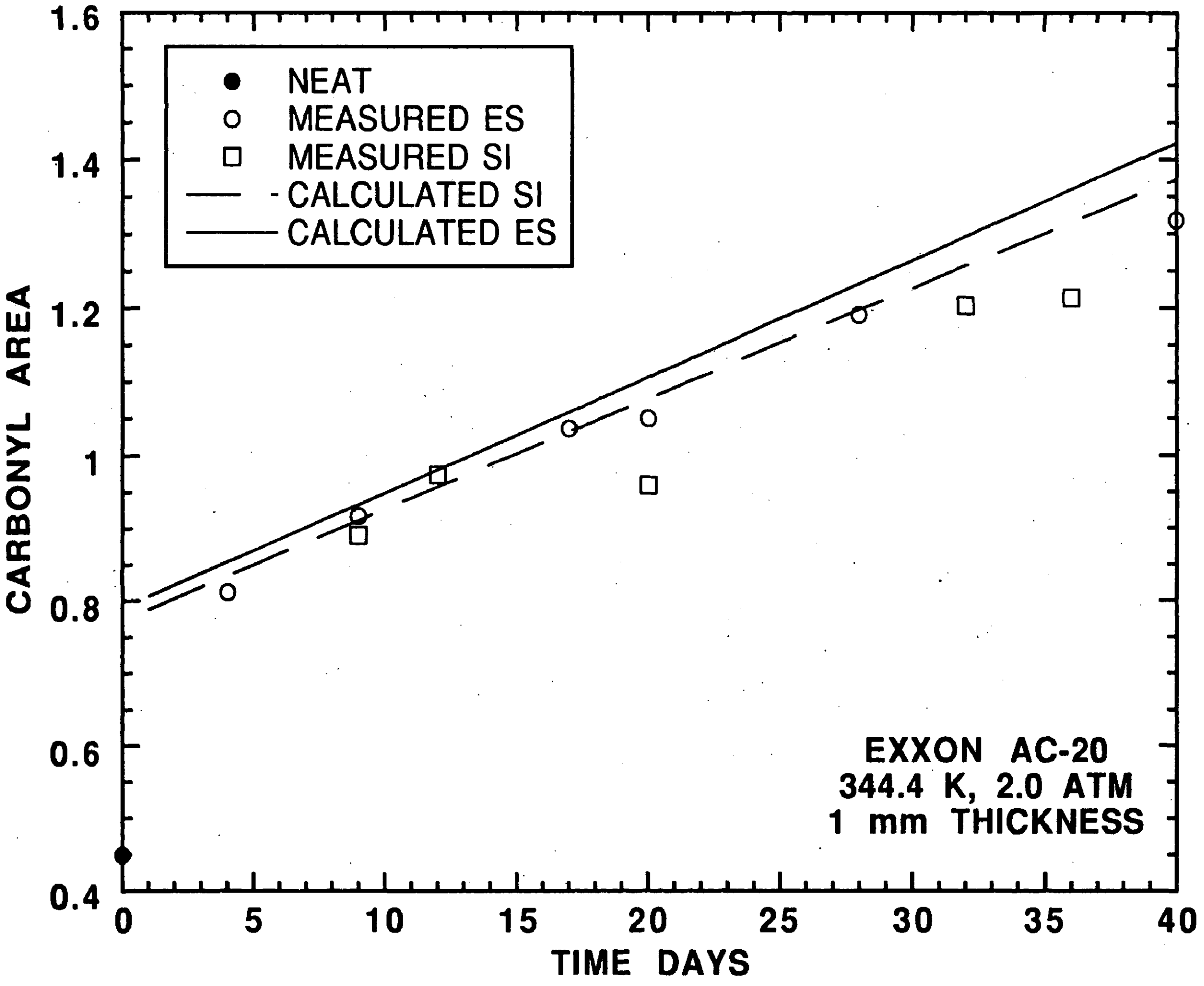

Figure C-50. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 344.4 K and P_{ES} of 2 atm.

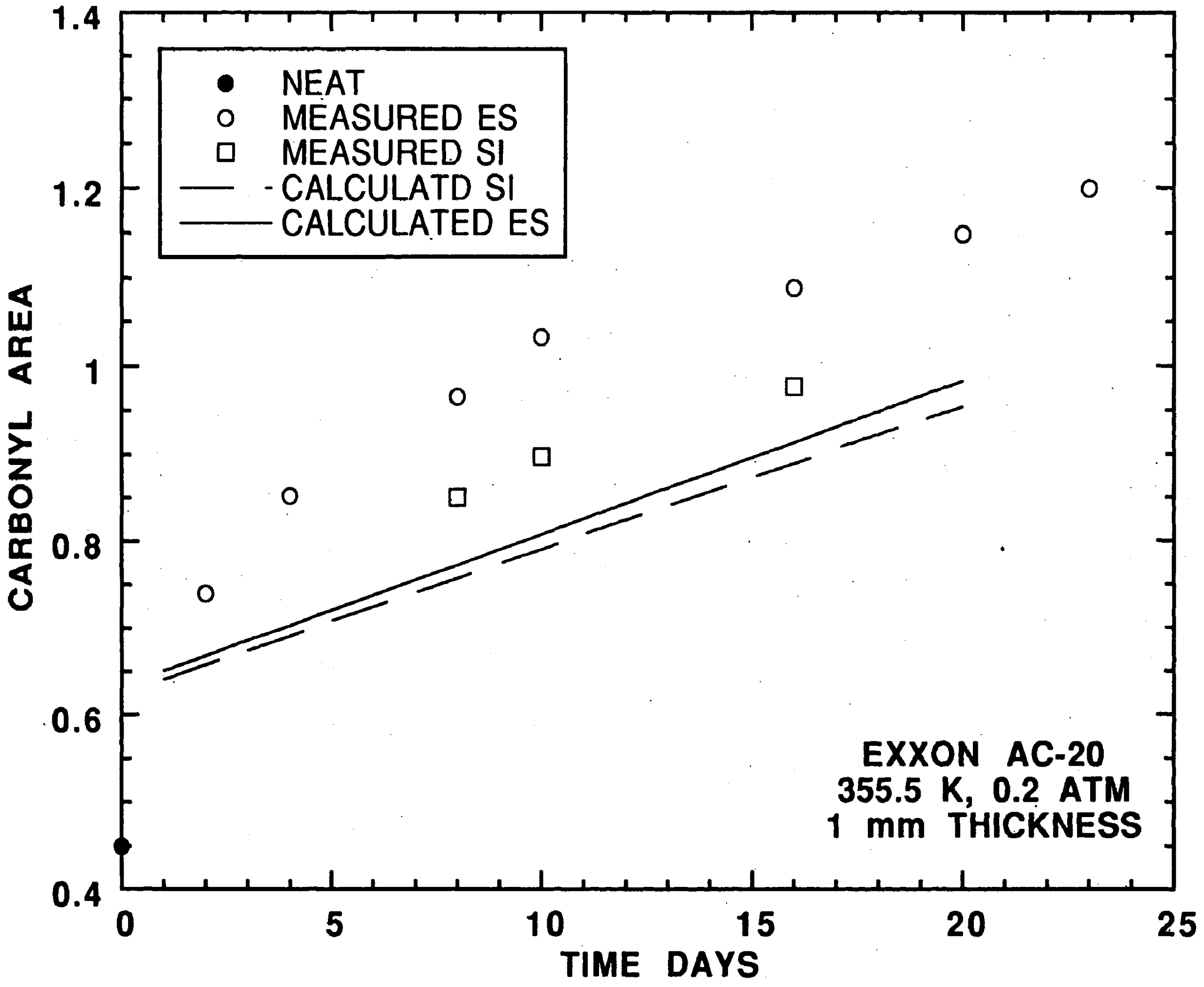

Figure C-51. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 355.5 K and P_{ES} of 0.2 atm.

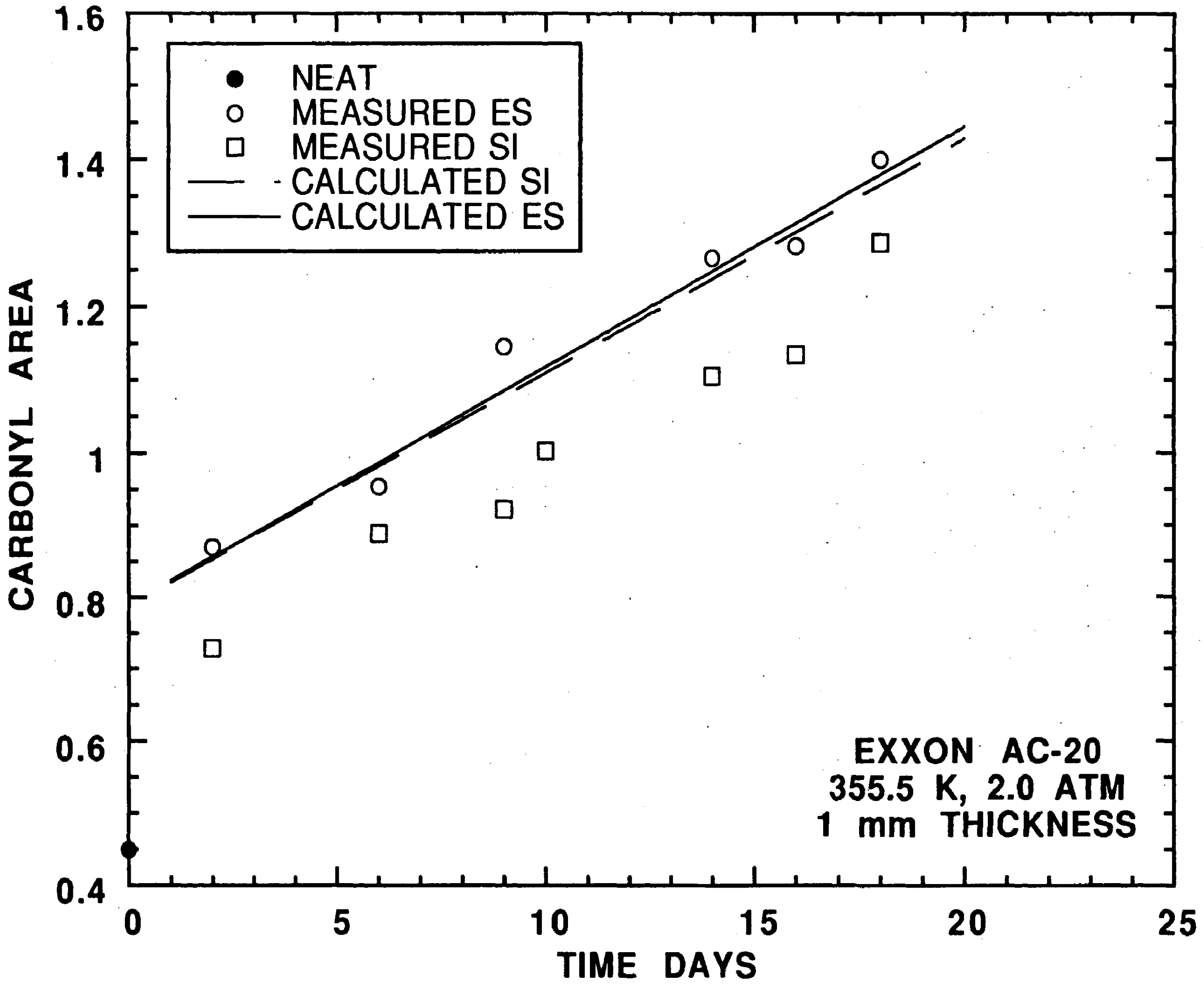

Figure C-52. Comparisons between measured and calculated *CA* at the *ES* and *SI* of 1 mm thick POV-aged Exxon AC-20 at 355.5 K and P_{ES} of 2 atm.

APPENDIX D

OXYGEN DIFFUSION PROGRAMS

```
c*****************************************************************************
c
c       This program estimates the diffusivity for 1-dimensional
c             steady-state constant diffusivity oxygen diffusion
c                          and reaction.
c
c*****************************************************************************
c
      integer n
      parameter (n=2)
c
      integer num, i, kount, kmax
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, deldiff, ratio, b, analdif, slpm
      real*4 f1, f2, f3, diff1, diff2, diff3, eps, p_es
      real*4 stepp, p_si, alpha, c, diff, difgues, temp
      character*8 asphalt
      character*12 outputfile, datafile, sumfile
      common /blocka/ conv, alpha, diff
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter you summary file name'
      read(*,'(a)') sumfile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 50
c
      read(8,*) alpha
      read(8,*) deldiff
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
c
      write(10,1031) alpha
      read(8,*) num
c
      do 20 i = 1, num, 1
          read(8,'(a)') asphalt
          read(8,*) p_si, p_es
          read(8,*) kp, temp
          read(8,*) slpm
```

```
           conv = c*r1*temp*kp*(1./24.)*(1./3600.)
c
c   These lines have been added to study the sensitivity of
c        diffusivity with respect to changes in p_si
c
           p_si = p_si*(1.0 + 0.40)
           ratio = (p_es/p_si)
           b = (1.0/length)*log(ratio + sqrt((ratio**2) - 1.0))
           analdif = (conv/(b**2))
c
c   Guess a diffusivity
c
           difgues = 1.0e-4
           diff = difgues
           write(9,1000) asphalt, p_si, p_es, kp, temp, diff
           write(9,1001) conv, ratio, b, analdif, slpm
           kount = 0
           call search (difgues, deldiff, diff1, diff2, f1,
     +              f2, kount, n)
c
c    The root is bounded
c
           call regfal (diff1, diff2, f1, f2, diff3, f3,
     +              eps, kount, kmax, n)
c
c    The root is found
c
           write(9,1020) kount, diff3, analdif, f3
           write(10,1030) asphalt, p_es, temp, diff3
c
   20 continue
c
c*****************************************************************************
c
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_si = ',f10.3,' atm',
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'guess diff = ',1pe10.4,' m^2/s')
 1001 format(' ',4x,'conv = ',1pe10.4,' atm^(alpha-1)/s',
     +     /,5x,'ratio = ',e10.4,' dimensionless',
     +     /,5x,'b = ',e10.4,' m^-1',
     +     /,5x,'caldif =',e10.4,' m^2/s',
     +     /,5x,'slpm =',0pf10.4,' atm/m',/)
 1020 format(' ',4x,'Number of integrations = ',i3,
     +      /,5x,'diff = ',1pe10.4,' m^2/s',
     +      /,5x,'analdiff = ',1pe10.4,' m^2/s',
     +      /,5x,'f3 = ',1pe11.4,' m',/)
 1031 format(' ',2x,'ASPHALT',7x,'Pressure',2x,'Temperature',
     +       2x,'Diffusivity',/,
```

```
     +       20x,'alpha = ',f10.4,/)
 1030 format(' ',5x,a8,5x,f5.2,5x,f5.1,5x,1pe10.4)
c
      stop
      end
c
c**************************************************************************
c
      subroutine search (difgues, deldiff, diff1, diff2, f1,
     +              f2, kount, n)
      integer iflag, kount, n
      real*4 t, y(2), x_f, p_f, z(2), deldiff, diff1
      real*4 diff2, f1, f2, length
      real*4 conv, alpha, diff
      real*4 step, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, diff
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      diff = difgues
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f1 = -(length - z(1))
           goto 55
      endif
      f1 = p_es - y(1)
c
   55 continue
      kount = kount + 1
      diff1 = diff
   50 diff = diff/deldiff
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
```

```
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f2 = -(length - z(1))
           goto 65
      endif
      f2 = p_es - y(1)
c
  65  kount = kount + 1
      diff2 = diff
c
c     check to see that the root is bounded
c
      if((f1*f2).gt.0.0) then
           diff1=diff2
           diff=diff1
           f1=f2
           goto 50
      endif
      return
      end
c
c*****************************************************************************
c
      subroutine regfal (diff1, diff2, f1, f2, diff3, f3,
     +           eps, kount, kmax, n)
c
      integer iflag, kount, kmax, n
      real*4 diff1, diff2, diff3, f1, f2, f3, eps, diff
      real*4 y(2), t, z(2), p, tend, pend, p_f, x_f
      real*4 conv, alpha, step, length, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, diff
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
  60  diff3 = (diff1*f2-diff2*f1)/(f2-f1)
      diff = diff3
c
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
```

```
          p_f = y(1)
          p = p_f
          z(1) = x_f
          z(2) = 1/y(2)
          call kevin2 (p, pend, n, z, iflag)
          f3 = -(length - z(1))
          p_cal = pend
          goto 65
      endif
c
      f3 = p_es - y(1)
c
   65 kount = kount+1
c
      if(abs(f3).ge.eps) then
          if(kount.ge.kmax) then
              goto 70
          endif
      if((f1*f3).le.0.0) then
          diff2=diff3
          f2=f3
          goto 60
      endif
          diff1=diff3
          f1=f3
          goto 60
      endif
c
   70 continue
c
c   The root has been found
c
      return
      end
c
c************************************************************************
c
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
      ido = 1
      istep = 0
c
   10 continue
      istep = istep+1
```

```
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
c
      if(tend.lt.corr) then
           if(y(2).gt.slpm) then
               iflag = 1
               istep = istep+1
               tend = (float(istep)/step)*length
               call ivprk(ido, n, equation1, t, tend, tol, param, y)
               thold = tend
               y1hold = y(1)
               y2hold = y(2)
               ido = 3
               tend = (float(istep)/step)*length
               call ivprk(ido, n, equation1, t, tend, tol, param, y)
               tend = thold
               y(1) = y1hold
               y(2) = y2hold
               return
           endif
           goto 10
      endif
c
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
      return
      end
c
c****************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 phold, z1hold, z2hold
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      external sset, ivprk, equation2
```

```
      call sset (50, 0.0, param, 1)
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
c
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
c
      if(pend.lt.corrp) then
           goto 10
      endif
c
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
      return
      end
c
c***********************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, diff, term2
      common /blocka/ conv, alpha, diff
c
      yprime(1) = y(2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = (1.0/diff)*(term2)
      return
      end
c
c***********************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
```

```
      real*4 p, z(n), zprime(n)
      real*4 conv, alpha, diff, term2
      common /blocka/ conv, alpha, diff
c
      zprime(1) = z(2)
      term2 = conv*(p**alpha)
      zprime(2) = -(term2/diff)*(z(2)**3)
      return
      end
```

```
c*************************************************************************
c
c          This program calculates the oxygen pressure profile
c                 for steady-state constant diffusivity
c                    oxygen diffusion and reaction
c
c*************************************************************************
c
      integer n
      parameter (n=2)
c
      integer num, i, kount, kmax
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, slpm
      real*4 eps, p_es
      real*4 stepp, p_si, alpha, c, diff, temp
      character*8 asphalt
      character*12 outputfile, datafile
      common /blocka/ conv, alpha, diff
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 50
c
      read(8,*) alpha
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
c
      write(9,1031) alpha, eps, tol, length, step, stepp
      read(8,*) num
c
      do 20 i = 1, num, 1
           read(8,'(a)') asphalt
           read(8,*) p_si, p_es
           read(8,*) kp, temp
           read(8,*) slpm
           read(8,*) diff
           conv = c*r1*temp*kp*(1./24.)*(1./3600.)
           write(9,1000) asphalt, p_si, p_es, kp, temp, diff
           write(9,1001) conv, slpm
```

```
          call kevin (i,n)
   20 continue
c
c*****************************************************************************
c
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_si = ',f10.3,' atm',
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'guess diff = ',1pe10.4,' m^2/s')
 1001 format(' ',4x,'conv = ',1pe10.4,' atm^(alpha-1)/s',
     +     /,5x,'slpm =',0pf10.4,' atm/m',/)
 1031 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +      /,5x,'tol = ',1pe10.4,' dimensionless',
     +      /,5x,'eps = ',e10.4,' dimensionless',
     +      /,5x,'length = ',e10.4,' m',
     +      /,5x,'step = ',0pf5.1,' dimensionless',
     +      /,5x,'stepp = ',f5.1,' dimensionless',/)
c
      stop
      end
c
c*****************************************************************************
c
      subroutine kevin (i,n)
c
      integer iflag, n, i
      real*4 t, y(2), x_f, p_f, z(2)
      real*4 length
      real*4 conv, alpha, diff
      real*4 step, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, diff
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      write(9,1800) i
 1800 format(' ',5x,i3,/)

      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
```

```
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
      endif
c
      return
      end
c
c*****************************************************************************
c
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
      ido = 1
      istep = 0
c
      write(9,2000) t, y(1)
c
   10 continue
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      write(9,2000) tend, y(1)
      if(tend.lt.corr) then
           if(y(2).gt.slpm) then
                iflag = 1
                istep = istep+1
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                thold = tend
                y1hold = y(1)
                y2hold = y(2)
                ido = 3
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                tend = thold
                y(1) = y1hold
                y(2) = y2hold
                return
           endif
           goto 10
      endif
c
      istep = istep+1
```

```
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      write(9,2000) tend, y(2)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
c
 2000 format(' ',5x,1pe10.4,' , ',e10.4)
c
      return
      end
c
c***************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 phold, z1hold, z2hold
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      external sset, ivprk, equation2
      call sset (50, 0.0, param, 1)
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
      write(9,3000) z(1), p_f
c
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
      write(9,3000) z(1) , pend
      if(pend.lt.corrp) then
           goto 10
      endif
c
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
```

```
      write(9,3000) z(1), pend
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
c
 3000 format(' ',5x,1pe10.4,' , 'e10.4)
c
      return
      end
c
c*****************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, diff, term2
      common /blocka/ conv, alpha, diff
c
      yprime(1) = y(2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = (1.0/diff)*(term2)
      return
      end
c
c*****************************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
      real*4 p, z(n), zprime(n)
      real*4 conv, alpha, diff, term2
      common /blocka/ conv, alpha, diff
c
      zprime(1) = z(2)
      term2 = conv*(p**alpha)
      zprime(2) = -(term2/diff)*(z(2)**3)
      return
      end
```

```
c***************************************************************************
c
c           This calculates the optimum "a" for the model
c
c                           D = an^-B
c
c       for steady-state variable diffusivity oxygen diffusion
c              and reaction.  The constant "B" is fixed.
c
c***************************************************************************
c
      integer n
      parameter (n=2)
c
      integer numa, numt, i, j, kount, kmax
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, dela, slpm, time
      real*4 f1, f2, f3, adif1, adif2, adif3, eps, p_es
      real*4 stepp, p_si, alpha, c, adif, adifgues, temp
      real*4 hs, m, s, beta
      character*8 asphalt
      character*12 outputfile, datafile, sumfile
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter your summary file name'
      read(*,'(a)') sumfile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 20
      read(8,*) alpha
      read(8,*) dela
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
      read(8,*) numa, numt
      read(8,*) beta, s
      read(8,*) bdif
      write(9,900) alpha, dela, eps, tol, length, step, stepp
      write(9,950) numa, numt, beta, s
```

```
c
      do 20 i = 1, numa, 1
           read(8,'(a)') asphalt
           read(8,*) p_si, p_es
           read(8,*) kp, temp
           read(8,*) hs, m
           m = log(m)
           conv = c*r1*temp*kp*(1./24.)*(1./3600.)
           write(9,1000) asphalt, p_si, p_es, kp, temp, hs, m, conv
c
           do 30 j = 1, numt, 1
                read(8,*) time, slpm
                write(9,1001) time, slpm
c
c     guess an adif of 1.0e-4
c
                adifgues = 1.0e-4
                adif = adifgues
                kount = 0
                call search (adifgues, dela, adif1, adif2, f1,
     +             f2, kount, n)
c
c    The root is bounded
c
                call regfal (adif1, adif2, f1, f2, adif3, f3,
     +             eps, kount, kmax, n)
c
c    The root is found
c
                write(9,1020) kount, time, adif3, f3
                write(10,1030) asphalt, time, adif3, kount
   30      continue
   20 continue
c
c*****************************************************************************
c
  900 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'dela = ',f10.4,' dimensionless',
     +     /,5x,'eps = ',1pe10.4,' dimensionless',
     +     /,5x,'tol = ',e10.4,' dimensionless',
     +     /,5x,'length = ',e10.4,' m',
     +     /,5x,'step = ',0pf10.4,' dimensionless',
     +     /,5x,'stepp = ',f10.4,' dimensionless',/)
  950 format(' ',4x,'numa = ',i3,' dimensionless',
     +     /,5x,'numt = ',i3,' dimensionless',
     +     /,5x,'beta = ',1pe10.4,' dimensionless',
     +     /,5x,'s =',e10.4,' ca/atm^beta',/)
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_si = ',f10.3,' atm',
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
```

```
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'hs = ',f10.4,' log(vis)/CA',
     +      /,5x,'m = ',f10.4,' log(vis)',
     +      /,5x,'conv = ',1pe10.4,' atm^(alpha-1)/s',/)
 1001 format(' ',4x,'time = ',f10.4,' days',
     +     /,5x,'slpm =',f10.4,' atm/m',/)
 1020 format(' ',4x,'Number of integrations = ',i3,
     +      /,5x,'time = ',f10.4,' days',
     +      /,5x,'adif = ',1pe10.4,' m^2/s',
     +      /,5x,'f3 = ',e11.4,' m',/)
 1030 format(' ',4x,a8,3x,f5.1,7x,1pe10.4,5x,i3)
c
      stop
      end
c
c**************************************************************************
c
      subroutine search (adifgues, dela, adif1, adif2, f1,
     +              f2, kount, n)
c
      integer iflag, kount, n
      real*4 t, y(2), x_f, p_f, z(2), dela, adif1
      real*4 adif2, f1, f2, length
      real*4 conv, alpha, adif
      real*4 step, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      adif = adifgues
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f1 = -(length - z(1))
           goto 55
      endif
      f1 = p_es - y(1)
c
```

```
   55 continue
      kount = kount + 1
      adif1 = adif
c
   50 adif = adif/dela
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f2 = -(length - z(1))
           goto 65
      endif
c
      f2 = p_es - y(1)
  65  kount = kount + 1
c
      adif2 = adif
c
c     check to see that the root is bounded
c
      if((f1*f2).gt.0.0) then
           adif1=adif2
           adif=adif1
           f1=f2
           goto 50
      endif
      return
      end
c
c*****************************************************************************
c
      subroutine regfal (adif1, adif2, f1, f2, adif3, f3,
     +             eps, kount, kmax, n)
c
      integer iflag, kount, kmax, n
      real*4 adif1, adif2, adif3, f1, f2, f3, eps, adif
      real*4 y(2), t, z(2), p, tend, pend, p_f, x_f
      real*4 conv, alpha, step, length, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
```

```
c
  60  adif3 = (adif1*f2-adif2*f1)/(f2-f1)
      adif = adif3
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f3 = -(length - z(1))
           goto 65
      endif
      f3 = p_es - y(1)
c
   65 kount = kount+1
      if(abs(f3).ge.eps) then
           if(kount.ge.kmax) then
                goto 70
           endif
           if((f1*f3).le.0.0) then
                adif2 = adif3
                f2 = f3
           goto 60
           endif
           adif1 = adif3
           f1=f3
           goto 60
      endif
c
   70 continue
c
c        The root has been found
c
      return
      end
c
c*****************************************************************************
c
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
```

```
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
      ido = 1
      istep = 0
c
   10 continue
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
c
      if(tend.lt.corr) then
          if(y(2).gt.slpm) then
                iflag = 1
                istep = istep+1
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                thold = tend
                y1hold = y(1)
                y2hold = y(2)
                ido = 3
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                tend = thold
                y(1) = y1hold
                y(2) = y2hold
                return
          endif
      goto 10
      endif
c
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk (ido, n, equation1, t, tend, tol, param, y)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
      return
      end
c
c*****************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
```

```
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 phold, z1hold, z2hold
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      external sset, ivprk, equation2
      call sset (50, 0.0, param, 1)
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
c
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
      if(pend.lt.corrp) then
           goto 10
      endif
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
      return
      end
c
c**************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, adif, term2, diff
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term1
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(y(1)**alpha)*time
```

```
      ca2 = s*(y(1)**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/y(1)
      diffp = bdif*hs*diff*cap
c
      yprime(1) = y(2)
      term1 = diffp*(y(2)**2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = -(1.0/diff)*(term1 - term2)
      return
      end
c
c****************************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
      real*4 p, z(n), zprime(n), diff
      real*4 conv, alpha, adif, term2
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term3
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(p**alpha)*time
      ca2 = s*(p**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/p
      diffp = bdif*hs*diff*cap
c
      zprime(1) = z(2)
      term3 = conv*(p**alpha)
      zprime(2) = (1/diff)*(diffp*z(2) - term3*(z(2)**3))
      return
      end
```

```
c*****************************************************************************
c
c          Calculates the oxygen pressure profile in the asphalt
c               film based on the optimum parameter "a"
c                          in the model
c
c                            D = an^-B
c
c           for steady-state variable diffusivity oxygen
c                     diffusion and reaction
c
c*****************************************************************************
c
      integer n
      parameter (n=2)
c
      integer numa, numt, i, j
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, slpm, time
      real*4 p_es, eps
      real*4 stepp, p_si, alpha, c, adif, temp
      real*4 hs, m, s, beta
      character*8 asphalt
      character*12 outputfile, datafile
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 50
      read(8,*) alpha
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
      read(8,*) numa, numt
      read(8,*) beta, s
      read(8,*) bdif
      write(9,900) alpha, eps, tol, length, step, stepp
      write(9,950) numa, numt, beta, s, bdif
c
      do 20 i = 1, numa, 1
```

```
          read(8,'(a)') asphalt
          read(8,*) p_si, p_es
          read(8,*) kp, temp
          read(8,*) hs, m
          m = log(m)
          conv = c*r1*temp*kp*(1./24.)*(1./3600.)
          write(9,1000) asphalt, p_si, p_es, kp, temp, hs, m, conv
c
          do 30 j = 1, numt, 1
               read(8,*) time, slpm, adif
               write(9,1001) time, slpm, adif
               call kevin (j, n)
   30     continue
   20 continue
c
c****************************************************************************
c
  900 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'eps = ',1pe10.4,' dimensionless',
     +     /,5x,'tol = ',e10.4,' dimensionless',
     +     /,5x,'length = ',e10.4,' m',
     +     /,5x,'step =',0pf10.4,' dimensionless',
     +     /,5x,'stepp =',f10.4,' dimensionless',/)
  950 format(' ',4x,'numa = ',i3,' dimensionless',
     +     /,5x,'numt = ',i3,' dimensionless',
     +     /,5x,'beta = ',1pe10.4,' dimensionless',
     +     /,5x,'s = ',e10.4,' ca/atm^beta',
     +     /,5x,'bdif = ',e11.3,' dimensionless',/)
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_si = ',f10.3,' atm',
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'hs = ',f10.4,' log(vis)/CA',
     +      /,5x,'m = ',f10.4,' log(vis)',
     +      /,5x,'conv = ',1pe10.4,' atm^(alpha-1)/s',/)
 1001 format(' ',4x,'time = ',f10.4,' days',
     +     /,5x,'slpm = ',f10.4,' atm/m',
     +     /,5x,'adif = ',1pe10.4,' m^2/s',/)
c
      stop
      end
c
c****************************************************************************
c
      subroutine kevin (j, n)
c
      integer iflag, n, j
      real*4 t, y(2), x_f, p_f, z(2)
      real*4 length, adif
      real*4 conv, alpha
```

```
      real*4 step, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      write (9,1100) j
 1100 format (' '/,4x,i3,/)
c
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           goto 55
      endif
c
   55 continue
      return
      end
c
c*****************************************************************************
c
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
c
      ido = 1
      istep = 0
      write(9,2000) t, y(1)
c
   10 continue
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
```

```
      write(9,2000) tend, y(1)
c
      if(tend.lt.corr) then
           if(y(2).gt.slpm) then
                iflag = 1
                istep = istep+1
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                thold = tend
                y1hold = y(1)
                y2hold = y(2)
                ido = 3
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                tend = thold
                y(1) = y1hold
                y(2) = y2hold
                return
           endif
           goto 10
      endif
c
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      write(9,2000) tend, y(1)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
c
 2000 format(' ',5x,1pe10.4,' , ',1pe10.4)
      return
      end
c
c****************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 phold, z1hold, z2hold
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
```

```
c
      external sset, ivprk, equation2
      call sset (50, 0.0, param, 1)
c
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
      write(9,3000) z(1), p
c
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
      write(9,3000) z(1), pend
      if(pend.lt.corrp) then
           goto 10
      endif
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
c
      write(9,3000) z(1), pend
 3000 format(' ',5x,1pe10.4,' , ',1pe10.4)
c
      return
      end
c
c*****************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, adif, term2, diff
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term1
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
```

```
      ca1 = kp*(y(1)**alpha)*time
      ca2 = s*(y(1)**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/y(1)
      diffp = bdif*hs*diff*cap
c
      yprime(1) = y(2)
      term1 = diffp*(y(2)**2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = -(1.0/diff)*(term1 - term2)
      return
      end
c
c***********************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
      real*4 p, z(n), zprime(n), diff
      real*4 conv, alpha, adif, term2
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term3
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(p**alpha)*time
      ca2 = s*(p**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/p
      diffp = bdif*hs*diff*cap
c
      zprime(1) = z(2)
      term3 = conv*(p**alpha)
      zprime(2) = (1/diff)*(diffp*z(2) - term3*(z(2)**3))
      return
      end
```

```
c*****************************************************************************
c
c        For the known model parameters "a" and "B" in the model
c
c                                 D = an^-B
c
c     this program calculates P_SI for given P_ES for steady-state
c          variable diffusivity oxygen diffusion and reaction
c
c*****************************************************************************
c
      integer n
      parameter (n=2)
c
      integer numa, numt, i, j, kount, kmax
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, delp, slpm, time
      real*4 f1, f2, f3, p_si1, p_si2, p_si3, eps, p_es
      real*4 stepp, p_si, alpha, c, adif, temp
      real*4 hs, m, s, beta, p_sim, ca_si, ca_es
      character*8 asphalt
      character*12 outputfile, datafile, sumfile
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter your summary file name'
      read(*,'(a)') sumfile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 20
      read(8,*) alpha
      read(8,*) delp
      read(8,*) adif, bdif
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
      read(8,*) numa, numt
      read(8,*) beta, s
      write(9,900) alpha, delp, eps, tol, length, step, stepp,
     +             adif, bdif
```

```
      write(9,950) numa, numt, beta, s
c
      do 20 i = 1, numa, 1
           read(8,'(a)') asphalt
           read(8,*) p_sim, p_es
           read(8,*) kp, temp
           read(8,*) hs, m
           m = log(m)
           conv = c*r1*temp*kp*(1./24.)*(1./3600.)
           write(9,1000) asphalt, p_sim, p_es, kp, temp, hs, m, conv
           write(10,1000) asphalt, p_sim, p_es, kp, temp, hs, m, conv
c
           do 30 j = 1, numt, 1
                read(8,*) time, slpm
                write(9,1001) time, slpm
c
c  guess a substrate interface pressure
c
                pgues = p_es
                p_si = pgues
                kount = 0
                call search (pgues, delp, p_si1, p_si2, f1,
     +             f2, kount, n)
c
c    The root is bounded
c
                call regfal (p_si1, p_si2, f1, f2, p_si3, f3,
     +             eps, kount, kmax, n)
c
c    The root is found
c
                write(9,1020) kount, time, p_si3, p_sim, f3
                ca_si = kp*(p_si3)**(alpha)*time + s*(p_si3)**(beta)
                ca_es = kp*(p_es)**(alpha)*time + s*(p_es)**(beta)
                write(10,1030) time, p_si3, ca_si, ca_es
c
   30      continue
   20 continue
c
c***************************************************************************
c
  900 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'dela = ',f10.4,' dimensionless',
     +     /,5x,'eps = ',1pe10.4,' dimensionless',
     +     /,5x,'tol = ',e10.4,' dimensionless',
     +     /,5x,'length = ',e10.4,' m',
     +     /,5x,'step = ',0pf10.4,' dimensionless',
     +     /,5x,'stepp = ',f10.4,' dimensionless',
     +     /,5x,'adif = ',1pe11.4,' m^2/s',
     +     /,5x,'bdif = ',0pf10.4,' dimensionless',/)
  950 format(' ',4x,'numa = ',i3,' dimensionless',
```

```
     +     /,5x,'numt = ',i3,' dimensionless',
     +     /,5x,'beta = ',1pe10.4,' dimensionless',
     +     /,5x,'s =',e10.4,' ca/atm^beta',/)
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_si = ',f10.3,' atm',
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'hs = ',f10.4,' log(vis)/CA',
     +      /,5x,'m = ',f10.4,' log(vis)',
     +      /,5x,'conv = ',1pe10.4,' atm^(alpha-1)/s',/)
 1001 format(' ',4x,'time = ',f10.4,' days',
     +     /,5x,'slpm =',f10.4,' atm/m',/)
 1020 format(' ',4x,'Number of integrations = ',i3,
     +      /,5x,'time = ',f10.4,' days',
     +      /,5x,'p_sical = ',1pe10.4,' atm',
     +      /,5x,'p_sim = ',e10.4,' atm',
     +      /,5x,'f3 = ',e11.4,' m',/)
 1030 format(' ',2x,f5.1,' , ',1pe10.4,' , ',e10.4,' , ',e10.4)
c
      stop
      end
c
c***************************************************************************
c
      subroutine search (pgues, delp, p_si1, p_si2, f1,
     +             f2, kount, n)
      integer iflag, kount, n
      real*4 t, y(2), x_f, p_f, z(2), delp, p_si1
      real*4 p_si2, f1, f2, length
      real*4 conv, alpha, adif
      real*4 step, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      p_si = pgues
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
```

```
            z(2) = 1/y(2)
            call kevin2 (p, pend, n, z, iflag)
            f1 = -(length - z(1))
            goto 55
         endif
         f1 = p_es - y(1)
c
   55 continue
         kount = kount + 1
         p_si1 = p_si
c
   50 p_si = p_si/delp
         iflag = 0
         t = 0.0
         y(1) = p_si
         y(2) = 0.0
         call kevin1 (t, tend, n, y, iflag)
c
         if(iflag.eq.1) then
            x_f = tend
            p_f = y(1)
            p = p_f
            z(1) = x_f
            z(2) = 1/y(2)
            call kevin2 (p, pend, n, z, iflag)
            f2 = -(length - z(1))
            goto 65
         endif
         f2 = p_es - y(1)
c
  65    kount = kount + 1
         p_si2 = p_si
c
c        check to see that the root is bounded
c
         if((f1*f2).gt.0.0) then
            p_si1 = p_si2
            p_si = p_si1
            f1 = f2
            goto 50
         endif
         return
         end
c
c***************************************************************************
c
         subroutine regfal (p_si1, p_si2, f1, f2, p_si3, f3,
        +            eps, kount, kmax, n)
c
         integer iflag, kount, kmax, n
         real*4 p_si1, p_si2, p_si3, f1, f2, f3, eps, adif
```

```
      real*4 y(2), t, z(2), p, tend, pend, p_f, x_f
      real*4 conv, alpha, step, length, corr, tol, slpm
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
   60 p_si3 = (p_si1*f2-p_si2*f1)/(f2-f1)
      p_si = p_si3
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
           x_f = tend
           p_f = y(1)
           p = p_f
           z(1) = x_f
           z(2) = 1/y(2)
           call kevin2 (p, pend, n, z, iflag)
           f3 = -(length - z(1))
           goto 65
      endif
      f3 = p_es - y(1)
c
   65 kount = kount+1
      if(abs(f3).ge.eps) then
           if(kount.ge.kmax) then
                goto 70
           endif
           if((f1*f3).le.0.0) then
                p_si2 = p_si3
                f2 = f3
                goto 60
           endif
                p_si1 = p_si3
                f1=f3
                goto 60
      endif
c
   70 continue
c
c        The root has been found
c
      return
      end
c
c*****************************************************************************
c
```

```
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
c
      ido = 1
      istep = 0
   10 continue
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
c
      if(tend.lt.corr) then
           if(y(2).gt.slpm) then
                iflag = 1
                istep = istep+1
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                thold = tend
                y1hold = y(1)
                y2hold = y(2)
                ido = 3
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                tend = thold
                y(1) = y1hold
                y(2) = y2hold
                return
           endif
      goto 10
      endif
c
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk (ido, n, equation1, t, tend, tol, param, y)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
      return
      end
```

```
c
c***************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 phold, z1hold, z2hold
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
      external sset, ivprk, equation2
      call sset (50, 0.0, param, 1)
c
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
c
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
      if(pend.lt.corrp) then
           goto 10
      endif
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
      return
      end
c
c***************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, adif, term2, diff
```

```
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term1
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(y(1)**alpha)*time
      ca2 = s*(y(1)**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/y(1)
      diffp = bdif*hs*diff*cap
c
      yprime(1) = y(2)
      term1 = diffp*(y(2)**2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = -(1.0/diff)*(term1 - term2)
      return
      end
c
c*************************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
      real*4 p, z(n), zprime(n), diff
      real*4 conv, alpha, adif, term2
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term3
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(p**alpha)*time
      ca2 = s*(p**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/p
      diffp = bdif*hs*diff*cap
c
      zprime(1) = z(2)
      term3 = conv*(p**alpha)
      zprime(2) = (1/diff)*(diffp*z(2) - term3*(z(2)**3))
      return
      end
```

```
c*****************************************************************************
c
c          Calculates the oxygen pressure profile in the asphalt
c                film based on the estimated "p_si" for
c               known "P_ES", "a" and "B" for the model
c
c                              D = an^-B
c
c                 for steady-state variable diffusivity
c                    oxygen diffusion and reaction
c
c*****************************************************************************
c
      integer n
      parameter (n=2)
c
      integer numa, numt, i, j
      real*4 t, tend, tol, y(n), kp, length, r1, step
      real*4 conv, corr, slpm, time
      real*4 p_es, eps
      real*4 stepp, p_si, alpha, c, adif, temp
      real*4 hs, m, s, beta
      character*8 asphalt
      character*12 outputfile, datafile
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
c
      open(8,file=datafile ,status='old')
      open(9,file=outputfile ,status='unknown')
c
      c =  6.56e-4
      r1 = 82.057
      kmax = 50
      read(8,*) alpha
      read(8,*) eps, tol
      read(8,*) length, step, stepp
      corr = length*(1.0 - (1.0/step))
      read(8,*) numa, numt
      read(8,*) beta, s
      read(8,*) bdif, adif
      write(9,900) alpha, eps, tol, length, step, stepp
      write(9,950) numa, numt, beta, s, bdif
c
      do 20 i = 1, numa, 1
```

```
            read(8,'(a)') asphalt
            read(8,*) p_es
            read(8,*) kp, temp
            read(8,*) hs, m
            m = log(m)
            conv = c*r1*temp*kp*(1./24.)*(1./3600.)
            write(9,1000) asphalt, p_es, kp, temp, hs, m, conv
c
            do 30 j = 1, numt, 1
                 read(8,*) time, slpm, p_si
                 write(9,1001) time, slpm, p_si
                 call kevin (j, n)
   30       continue
   20 continue
c
c****************************************************************************
c
  900 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'eps = ',1pe10.4,' dimensionless',
     +     /,5x,'tol = ',e10.4,' dimensionless',
     +     /,5x,'length = ',e10.4,' m',
     +     /,5x,'step =',0pf10.4,' dimensionless',
     +     /,5x,'stepp =',f10.4,' dimensionless',/)
  950 format(' ',4x,'numa = ',i3,' dimensionless',
     +     /,5x,'numt = ',i3,' dimensionless',
     +     /,5x,'beta = ',1pe10.4,' dimensionless',
     +     /,5x,'s = ',e10.4,' ca/atm^beta',
     +     /,5x,'bdif = ',e11.3,' dimensionless',/)
 1000 format(' ',4x,'file = ',a8,
     +      /,5x,'P_es = ',f10.3,' atm',
     +      /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +      /,5x,'temp = ',0pf10.1,' K',
     +      /,5x,'hs = ',f10.4,' log(vis)/CA',
     +      /,5x,'m = ',f10.4,' log(vis)',
     +      /,5x,'conv = ',1pe10.4,' atm^(alpha-1)/s',/)
 1001 format(' ',4x,'time = ',f10.4,' days',
     +     /,5x,'slpm = ',f10.4,' atm/m',
     +     /,5x,'p_si = ',f10.4,' atm',/)
c
      stop
      end
c
c****************************************************************************
c
      subroutine kevin (j, n)
c
      integer iflag, n, j
      real*4 t, y(2), x_f, p_f, z(2)
      real*4 length, adif
      real*4 conv, alpha
      real*4 step, corr, tol, slpm
```

```
      real*4 p_es, stepp, p_si
      common /blocka/ conv, alpha, adif
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
c     set the initial conditions
c
      write (9,1100) j
 1100 format (' '/,4x,i3,/)
c
      iflag = 0
      t = 0.0
      y(1) = p_si
      y(2) = 0.0
      call kevin1 (t, tend, n, y, iflag)
c
      if(iflag.eq.1) then
          x_f = tend
          p_f = y(1)
          p = p_f
          z(1) = x_f
          z(2) = 1/y(2)
          call kevin2 (p, pend, n, z, iflag)
          goto 55
      endif
c
   55 continue
      return
      end
c
c****************************************************************************
c
      subroutine kevin1 (t, tend, n, y, iflag)
c
      integer n, ido, istep, iflag
      real*4 t, y(n), param(50), tend, step, corr
      real*4 tol, length, slpm, thold, y1hold, y2hold
      common /blockb/ step, length, corr, tol, slpm
c
      external sset, ivprk, equation1
      call sset (50, 0.0, param, 1)
c
      ido = 1
      istep = 0
      write(9,2000) t, y(1)
c
   10 continue
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      write(9,2000) tend, y(1)
```

```
c
      if(tend.lt.corr) then
           if(y(2).gt.slpm) then
                iflag = 1
                istep = istep+1
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                thold = tend
                y1hold = y(1)
                y2hold = y(2)
                ido = 3
                tend = (float(istep)/step)*length
                call ivprk(ido, n, equation1, t, tend, tol, param, y)
                tend = thold
                y(1) = y1hold
                y(2) = y2hold
                return
           endif
      goto 10
      endif
c
      istep = istep+1
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      write(9,2000) tend, y(1)
      thold = tend
      y1hold = y(1)
      y2hold = y(2)
      ido = 3
      tend = (float(istep)/step)*length
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      tend = thold
      y(1) = y1hold
      y(2) = y2hold
c
 2000 format(' ',5x,1pe10.4,' , ',1pe10.4)
      return
      end
c
c************************************************************************
c
      subroutine kevin2 (p, pend, n, z, iflag)
c
      integer n, ido, istep, iflag
      real*4 p, z(n), param(50), pend, stepp, corrp
      real*4 tol, slpm, pstar, p_f, p_es, corr
      real*4 phold, z1hold, z2hold
      real*4 length, step, p_si
      common /blockb/ step, length, corr, tol, slpm
      common /blockc/ p_es, stepp, p_si
c
```

```
      external sset, ivprk, equation2
      call sset (50, 0.0, param, 1)
c
      p_f = p
      ido = 1
      istep = 0
      pstar = (p_es - p_f)
      corrp = pstar*(1.0 - (1.0/stepp)) + p_f
      write(9,3000) z(1), p
   10 continue
      istep = istep+1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk(ido, n, equation2, p, pend, tol, param, z)
      write(9,3000) z(1), pend
      if(pend.lt.corrp) then
           goto 10
      endif
      istep = istep + 1
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      phold = pend
      z1hold = z(1)
      z2hold = z(2)
      ido=3
      pend = (float(istep)/stepp)*pstar + p_f
      call ivprk (ido, n, equation2, p, pend, tol, param, z)
      pend = phold
      z(1) = z1hold
      z(2) = z2hold
      write(9,3000) z(1), pend
c
 3000 format(' ',5x,1pe10.4,' , ',1pe10.4)
      return
      end
c
c***********************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 conv, alpha, adif, term2, diff
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term1
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(y(1)**alpha)*time
      ca2 = s*(y(1)**beta)
      ca = ca1 + ca2
```

```
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/y(1)
      diffp = bdif*hs*diff*cap
c
      yprime(1) = y(2)
      term1 = diffp*(y(2)**2)
      term2 = conv*(y(1)**alpha)
      yprime(2) = -(1.0/diff)*(term1 - term2)
      return
      end
c
c*****************************************************************************
c
      subroutine equation2 (n, p, z, zprime)
c
      integer n
      real*4 p, z(n), zprime(n), diff
      real*4 conv, alpha, adif, term2
      real*4 s, beta, m, hs, bdif, time, kp
      real*4 ca1, ca2, ca, eta, cap, diffp
      real*4 term3
      common /blocka/ conv, alpha, adif
      common /blocke/ s, beta, m, hs, bdif, time, kp
c
      ca1 = kp*(p**alpha)*time
      ca2 = s*(p**beta)
      ca = ca1 + ca2
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      cap = ((alpha*ca1)+(beta*ca2))/p
      diffp = bdif*hs*diff*cap
c
      zprime(1) = z(2)
      term3 = conv*(p**alpha)
      zprime(2) = (1/diff)*(diffp*z(2) - term3*(z(2)**3))
      return
      end
```

```
c*****************************************************************************
c
c            For the given parameters "a" and "B", this program
c      solves the partial differential equation for unsteady-state
c      variable diffusivity oxygen diffusion and reaction.
c
c            The spatial variable is written as finite
c      difference equations. Futhermore, for each spatial node, the
c      rate of carbonyl formation is also expressed.  This yields a
c      total of 2n first order ODEs.
c
c            These equations are integrated forward in
c      time by using the IVPRK method.  The initial condition for all
c      temperatures and pressure is zero oxygen pressure in the film.  The
c      boundary conditions are specified such that the oxygen pressure at
c      the surface in known and the flux and the impermeable substrate is 0.
c
c*****************************************************************************
c
      integer nmax
      parameter (nmax = 100, nmax2 = 200)
c
      integer i, j, n, numtemp, numas, numpres
      integer ii, iii, inc, nn
      real*4 y(nmax), z(nmax2), length, kp, conv
      real*4 p_es, tol, delx, diff, fact
      real*4 alpha, a, e_a, temp, r1, r2, c
      real*4 adif, bdif, hs, m, s, beta
      real*4 m_o, hs_o, temp_o
      character*8 asphalt
      character*12 outputfile, datafile, sumfile
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
      common /blockc/ tol
      common /blockd/ inc, nn
      common /blocke/ delx, delxsq
      common /blocki/ p_es, fact_i
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter your summary file name'
      read(*,'(a)') sumfile
c
      open(unit=8,file=datafile ,status='old')
      open(unit=9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
c
      read(8,*) numas , numtemp, numpres
      read(8,*) tol, length, n
```

```
      read(8,*) alpha, adif, bdif, temp_o
      adif = adif*(3600.0)*(24.0)
      c = 6.56e-4
      r1 = 82.057
      r2 = 8.314
      n2 = 2*n
      delx = length/float(n)
      delxsq = delx**2
      inc = n/5
      nn = n - (inc/2)
      iflag = 0
c
      do 20 i = 1, numas , 1
           read(8,'(a)') asphalt
           read(8,*) a, e_a
           read(8,*) s, beta, hs_o, m_o, gamma, delta
           m_o = log(m_o)
c
           do 30 ii = 1, numtemp, 1
                read(8,*) temp, timemax
                kp = a*exp(-e_a/(r2*temp))
                conv = c*r1*temp
                hs = hs_o + gamma*((1.0/temp) - (1.0/temp_o))
                m = m_o + delta*((1.0/temp) - (1.0/temp_o))
c
                do 40 iii = 1, numpres, 1
                     read(8,*) p_es
                     write(9,800) asphalt
                     write(9,1200) tol, length, p_es, temp, delx
                     write(9,1300) alpha, a, e_a, c, r1, r2, kp, conv
                     write(9,1400) adif, bdif, hs_o, m_o, temp_o,
     +                    gamma, delta, beta, s
                     call propinit (length, diff_i)
                     fact_i = (diff_i/delxsq)
                     write(9,1500) temp, hs, m, diff_i, fact_i
                     write(10, 1600) asphalt, temp, p_es, diff_i
                     call kevin (n, y, n2, z, timemax, iflag, p_es)
                     iflag = 1
c
   40           continue
   30      continue
   20 continue
c
c***************************************************************************
c
  800 format(' ',/,5x,'asphalt = ',a8,/)
 1200 format(' ',4x,'tol = ',1pe11.4,' dimensionless',
     +     /,5x,'length = ',0pf10.4,' m',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'temp = ',f10.4,' K',
     +     /,5x,'delx = '1pe10.4,' m',/)
```

```
 1300 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'a = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'e_a = ',0pf10.1,' J/mol',
     +     /,5x,'c  = ',1pe10.4,' mol/cm^3 CA',
     +     /,5x,'r1 = ',0pf10.4,' cm^3 atm/mol K',
     +     /,5x,'r2 = ',f10.4,' J/mol K',
     +     /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'conv = ',e10.4,' atm/CA',/)
 1400 format(' ',4x,'adif = ',1pe10.4,' m^2/day',
     +     /,5x,'bdif = ',0pf10.4,' dimensionless',
     +     /,5x,'hs_o = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m_o  = ',f10.4,' ln(vis)',
     +     /,5x,'temp_o = ',f10.1,' K',
     +     /,5x,'gamma = ',f10.1,' K/CA',
     +     /,5x,'delta = ',f10.1,' K',
     +     /,5x,'beta = ',f10.4,' dimensionless',
     +     /,5x,'s = ',f10.4,' CA/atm^beta',/)
 1500 format(' ',4x,'temp = ',f10.1,' K',
     +     /,5x,'hs = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m  = ',f10.4,' ln(vis)',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',
     +     /,5x,'fact_i = ',1pe10.4,' 1/day',/)
 1600 format(' ',4x,'asphalt = ',a8,
     +     /,5x,'temp = ',f10.1,' K',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',/)
c
      stop
      end
c
c**************************************************************************
c
      subroutine propinit (length, diff_i)
c
      real*4 length, m, p_es, fact_i
      real*4 hs, adif, bdif, s, beta
      real*4 p_i, ca_i, eta_i, diff_i
      common /blocki/ p_es, fact_i
      common /blocka/ hs, m, adif, bdif, s, beta
c
      p_i = (p_es)*0.5
      ca_i = s*(p_i**beta)
      eta_i = exp(hs*ca_i + m)
      diff_i = adif*(eta_i**bdif)
      return
      end
c
c**************************************************************************
c
      subroutine kevin (n, y, n2, z, timemax, iflag, p_es)
c
```

```
      integer n, n2, ido, istep, iflag, nflag
      real*4 t, y(n), z(n2), param(50), tend, timemax
      real*4 tol
      common /blockc/ tol
c
      external sset, ivprk, equation1, equation2
      call sset (50, 0.0, param, 1)
      param(4) = 20000
      timei = 1.0
c
      do 100 j = 1, n, 1
           y(j) = 0.0
  100 continue
      ido = 1
      istep = 0
      nflag = 0
   10 continue
      istep = istep+1
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      call output (y, n, t, iflag, timei, nflag)
      if(nflag.eq.0) then
           goto 10
      endif
      ido = 3
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      ido = 1
      istep = 1
      t = 1.0
      do 200 j = 1, n, 1
           z(j) = y(j)
           p = y(j)
           call carbonyl(p, t, ca_i)
           z(n+j) = ca_i
  200 continue
c
      nflag = 0
      call output2 (z, n2, t, timemax, nflag)
   20 continue
      istep = istep+1
      tend = float(istep)
      call ivprk(ido, n2, equation2, t, tend, tol, param, z)
      call output2 (z, n2, t, timemax, nflag)
      if(nflag.eq.0) then
           goto 20
      endif
      ido = 3
      tend = float(istep)
      call ivprk(ido, n2, equation2, t, tend, tol, param, z)
      return
```

```
      end
c
c*******************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n, nm1, j, jp1, jm1
      real*4 t, y(n), yprime(n), term2, yima, p_es0
      real*4 p_es, fact_i, car, p, ocr
      common /blocki/ p_es, fact_i
c
      do 20 j = 1, n, 1
           if(y(j).lt.0.0) then
                y(j) = 0.0
           endif
   20 continue
c
      yima = y(2)
c
c       Calculate the 0 flux boundary condition
c

      p = y(1)
      call rates(p, ocr, car)
      term2 = (fact_i)*(y(2) - 2.0*y(1) + yima)
      yprime(1) = term2 - ocr
c
c          Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = y(j)
           call rates(p, ocr, car)
           term2 = (fact_i)*(y(jp1) - 2.0*y(j) + y(jm1))
           yprime(j) = term2 - ocr
   30 continue
c
c           Calculate the known pressure at the free surface
c
      if(t.eq.0.0) then
           p_es0 =(p_es-y(1))/2.0
           p = y(n)
           call rates(p, ocr, car)
           term2 = (fact_i)*(p_es0 - 2.0*y(n) + y(nm1))
           yprime(n) = term2 - ocr
           goto 100
      endif
c
      p = y(n)
```

```
      call rates(p, ocr, car)
      term2 = (fact_i)*(p_es - 2.0*y(n) + y(nm1))
      yprime(n) = term2 - ocr
c
  100 continue
      return
      end
c
c*************************************************************************************
c
      subroutine equation2 (n2, t, z, zprime)
c
      integer n, nm1, n2, j, jp1, jm1
      real*4 t, z(n2), zprime(n2), term2, yima
      real*4 delx, delxsq, term1, ca, diff, diffp, p, car
      real*4 p_es
      common /blocki/ p_es, fact_i
      common /blocke/ delx, delxsq
c
      n = (n2/2)
      do 20 j = 1, n, 1
           if(z(j).lt.0.0) then
                z(j) = 0.0
           endif
   20 continue
c
      zima = z(2)
c
c             Calculate the 0 flux boundary condition
c
      p = z(1)
      ca = z(n+1)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
c
      fact = (diff/delxsq)
      term2 = fact*(z(2) - 2*z(1) + zima)
      zprime(1) = term2 - ocr
      zprime(n+1) = car
c
c           Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = z(j)
           ca = z(n+j)
           call rates (p, ocr, car)
           call diffus (p, ca, diff, diffp)
           fact = (diff/delxsq)
```

```
           term1 = (diffp/(4*delxsq))*((z(jp1) - z(jm1))**2)
           term2 = fact*(z(jp1) - 2.0*z(j) + z(jm1))
           zprime(j) = term1 + term2 - ocr
           zprime(n+j) = car
   30 continue
c
c           Calculate the known pressure at the free surface
c
      p = z(n)
      ca = z(n+n)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
      fact = (diff/delxsq)
      term1 = (diffp/(4.0*delxsq))*((p_es - z(nm1))**2)
      term2 = fact*(p_es - 2.0*z(n) + z(nm1))
      zprime(n) = term1 + term2 - ocr
      zprime(n+n) = car
      return
      end
c
c*****************************************************************************
c
      subroutine diffus (p, ca, diff, diffp)
c
      real*4 p, ca
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
c
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      if(p.eq.0) then
           cap = 0.0
           goto 100
      endif
      cap = alpha*ca/p
c
      if(cap.gt.1.0) then
           cap = 1.0
           goto 100
      endif
c
  100 continue
      diffp = bdif*hs*diff*cap
      return
      end
c
c*****************************************************************************
c
      subroutine rates (p, ocr, car)
```

```
c
      real*4 alpha, kp, conv, ocr, car
      common /blockb/ alpha, kp, conv
c
      car = kp*(p**alpha)
      ocr = conv*car
      return
      end
c
c*****************************************************************************
c
      subroutine carbonyl (p, t, ca_i)
c
      real*4 p, t, ca_i
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
c
      ca_i = t*kp*(p**alpha) + s*(p**beta)
      return
      end
c
c*****************************************************************************
c
      subroutine output (y, n, t, iflag, timei, nflag)
c
      integer n, ido, istep, iflag, nflag
      integer inc,  nn
      real*4 t, y(n), tend, timei, tm1
      real*4 p_es, fact_i
      common /blocki/ p_es, fact_i
      common /blockd/ inc, nn
c
      do 10 j = 1, n, 1
           if(y(j).lt.0.0) then
                y(j) = 0.0
           endif
   10 continue
c
      if(iflag.eq.0) then
          if(t.ge.timei) then
               nflag = 1
          endif
      return
      endif
c
      tm1 = t - 1
      if(tm1.eq.0.0) then
           return
      endif
c
```

```
      if(tm1.ge.timei) then
           nflag = 1
      endif
c
 2000 format(' ',2x,f6.0,' , ',5(f6.3,' , '),f6.3)
      return
      end
c
c*******************************************************************************
c
      subroutine output2 (z, n2, t, timemax, nflag)
c
      integer n, nflag, inc, nn, n2
      real*4 t, z(n2), tend, timemax
      real*4 p_es, fact_i
      common /blocki/ p_es, fact_i
      common /blockd/ inc, nn
c
      n = (n2/2)
      do 10 j = 1, n, 1
           if(z(j).lt.0.0) then
                z(j) = 0.0
           endif
   10 continue
c
      call carbonyl(p_es, t, ca_es)
      write(9,2000) t, (z(j), j=1,nn,inc),  p_es
      write(10,2000) t, (z(j+n), j = 1,nn,inc), ca_es
      if(t.ge.timemax) then
           nflag = 1
      endif
c
 2000 format(' ',2x,f6.0,' , ',5(f6.3,' , '),f6.3)
      return
      end
```

```
c******************************************************************************
c
c                              Aging test
c
c       This program calculates the time to reach a pre-defined failure
c                  criteria based on viscosity at 333.3 K
c
c       Two indendent calculations are performed.  The first assumes
c           no oxygen diffusion resistance.  The second assumes the
c                       models previously discussed.
c
c******************************************************************************
c
      integer nmax
      parameter (nmax = 100, nmax2 = 200)
c
      integer i, j, n, numtemp, numas, numpres
      integer ii, iii, inc, nn
      real*4 y(nmax), z(nmax2), length, kp, conv
      real*4 p_es, tol, delx, diff, fact
      real*4 alpha, a, e_a, temp, r1, r2, c
      real*4 adif, bdif, hs, m, s, beta, ca_i
      real*4 m_o, hs_o, temp_o, vis_fail
      character*8 asphalt
      character*12 outputfile, datafile, sumfile
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
      common /blockc/ tol
      common /blockd/ inc, nn
      common /blocke/ delx, delxsq
      common /blockt/ length
      common /blocki/ p_es, fact_i
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter your summary file name'
      read(*,'(a)') sumfile
c
      open(unit=8,file=datafile ,status='old')
      open(unit=9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
c
      read(8,*) numas , numtemp
      read(8,*) vis_fail , p_es
      read(8,*) tol, length, n
      read(8,*) alpha, adif, bdif, temp_o
      adif = adif*(3600.0)*(24.0)
      c = 6.56e-4
      r1 = 82.057
```

```
      r2 = 8.314
      n2 = 2*n
      delx = length/float(n)
      delxsq = delx**2
      inc = n/5
      nn = n - (inc/2)
      iflag = 0
c
      do 20 i = 1, numas , 1
           read(8,'(a)') asphalt
           read(8,*) a, e_a
           read(8,*) s, beta, hs_o, m_o, gamma, delta
           m_o = log(m_o)
c
           do 30 ii = 1, numtemp, 1
                read(8,*) temp
                kp = a*exp(-e_a/(r2*temp))
                conv = c*r1*temp
                hs = hs_o + gamma*((1.0/temp) - (1.0/temp_o))
                m = m_o + delta*((1.0/temp) - (1.0/temp_o))
                write(9,800) asphalt
                write(9,1200) tol, length, p_es, temp, delx
                write(9,1300) alpha, a, e_a, c, r1, r2, kp, conv
                write(9,1400) adif, bdif, hs_o, m_o, temp_o,
     +                   gamma, delta, beta, s
                ca_fail = (log(vis_fail) - m_o)/hs_o
c
c                        No Diffusion Problems
c
                ca_o = s*(p_es**beta)
                ca_dif = ca_fail - ca_o
                rate = kp*(p_es**alpha)
                time_fail = (ca_dif/rate)
c
                write(9,1450) vis_fail, ca_fail, ca_o, ca_dif,
     +                  rate, time_fail
c
                time_fail = (time_fail/365.0)
                write(10., 1475) asphalt, temp, time_fail
                call propinit (length, diff_i, ca_i)
                fact_i = (diff_i/delxsq)
                write(9,1500) temp, hs, m, diff_i, fact_i
                write(10,1600) asphalt, temp, p_es, diff_i
                ca_dif = ca_fail - ca_i
                call kevin (n, y, n2, z, ca_dif, iflag, p_es,
     +                       time_fail)
                iflag = 1
                write(9, 1650) time_fail
                time_fail = time_fail/365.0
                write(10,1575) asphalt, temp, time_fail
 30        continue
```

```
   20 continue
c
c*****************************************************************************
c
  800 format(' ',/,5x,'asphalt = ',a8,/)
 1200 format(' ',4x,'tol = ',1pe11.4,' dimensionless',
     +     /,5x,'length = ',0pf10.4,' m',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'temp = ',f10.4,' K',
     +     /,5x,'delx = '1pe10.4,' m',/)
 1300 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'a = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'e_a = ',0pf10.1,' J/mol',
     +     /,5x,'c  = ',1pe10.4,' mol/cm^3 CA',
     +     /,5x,'r1 = ',0pf10.4,' cm^3 atm/mol K',
     +     /,5x,'r2 = ',f10.4,' J/mol K',
     +     /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'conv = ',e10.4,' atm/CA',/)
 1400 format(' ',4x,'adif = ',1pe10.4,' m^2/day',
     +     /,5x,'bdif = ',0pf10.4,' dimensionless',
     +     /,5x,'hs_o = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m_o  = ',f10.4,' ln(vis)',
     +     /,5x,'temp_o = ',f10.1,' K',
     +     /,5x,'gamma = ',f10.1,' K/CA',
     +     /,5x,'delta = ',f10.1,' K',
     +     /,5x,'beta = ',f10.4,' dimensionless',
     +     /,5x,'s = ',f10.4,' CA/atm^beta',/)
 1450 format(' ',4x,'vis_fail = ',1pe10.4,' P',
     +     /,5x,'ca_fail = ',0pf10.4,' CA',
     +     /,5x,'ca_o = ',f10.4,' CA',
     +     /,5x,'ca_dif  = ',f10.4,' CA',
     +     /,5x,'rate = ',1pe10.4,' CA/day',
     +     /,5x,'time_fail = ',0pf10.1,' days',/)
 1475 format(' ',4x,'No Diffusion Resistance',
     +       /,5x,'asphalt =',a8,
     +       /,5x,'temp =',f10.1,' K',
     +       /,5x,'time_fail =',f10.2,' years',/)
 1500 format(' ',4x,'temp = ',f10.1,' K',
     +     /,5x,'hs = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m  = ',f10.4,' ln(vis)',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',
     +     /,5x,'fact_i = ',1pe10.4,' 1/day',/)
 1575 format(' ',4x,'With Diffusion Resistance',
     +       /,5x,'asphalt =',a8,
     +       /,5x,'temp =',f10.1,' K',
     +       /,5x,'time_fail =',f10.2,' years',/)
 1600 format(' ',4x,'asphalt = ',a8,
     +     /,5x,'temp = ',f10.1,' K',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',/)
 1650 format(' ',4x,'time_fail = ',f10.1,' days',/)
```

```
c
      stop
      end
c
c****************************************************************************
c
      subroutine propinit (length, diff_i, ca_i)
c
      real*4 length, m, p_es, fact_i
      real*4 hs, adif, bdif, s, beta
      real*4 p_i, ca_i, eta_i, diff_i
      common /blocki/ p_es, fact_i
      common /blocka/ hs, m, adif, bdif, s, beta
c
      p_i = (p_es)*0.5
      ca_i = s*(p_i**beta)
      eta_i = exp(hs*ca_i + m)
      diff_i = adif*(eta_i**bdif)
      return
      end
c
c****************************************************************************
c
      subroutine kevin (n, y, n2, z, ca_dif, iflag, p_es,
     +                  time_fail)
c
      integer n, n2, ido, istep, iflag, nflag
      real*4 t, y(n), z(n2), param(50), tend, timemax
      real*4 tol, ca_fail
      common /blockc/ tol
c
      external sset, ivprk, equation1, equation2
      call sset(50, 0.0, param, 1)
      param(4) = 20000
      timei = 1.0
      write(10,*) ca_dif
c
      do 100 j = 1, n, 1
           y(j) = 0.0
  100 continue
      ido = 1
      istep = 0
      nflag = 0
c
   10 continue
      istep = istep+1
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      call output(y, n, t, iflag, timei, nflag)
      if(nflag.eq.0) then
           goto 10
```

```
      endif
      ido = 3
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      ido = 1
      istep = 1
      t = 1.0
      do 200 j = 1, n, 1
           z(j) = y(j)
           p = y(j)
           call carbonyl(p, t, ca_i)
           z(n+j) = ca_i
  200 continue
      nflag = 0
      tm1 = t - 1
      call carbonyl(p_es, tm1, ca_es)
      call trap(z, ca_es, n2, n, ca_avg)
      if(ca_avg.ge.ca_dif) then
           goto 4000
      endif
      call output2 (z, n2, t)
c
   20 continue
      if(t.lt.10.0) then
           istep = istep+1
           goto 3000
      endif
      istep = istep + 20
 3000 continue
      tend = float(istep)
      call ivprk(ido, n2, equation2, t, tend, tol, param, z)
      call carbonyl(p_es, t, ca_es)
      call trap(z, ca_es, n2, n, ca_avg)
c
      if(ca_avg.ge.ca_dif) then
           nflag = 1
      endif
      call output2 (z, n2, t)
      if(nflag.eq.0) then
           goto 20
      endif
c
 4000 continue
      ido = 3
      tend = float(istep)
      call ivprk(ido, n2, equation2, t, tend, tol, param, z)
      time_fail = t
      return
      end
c
c****************************************************************************
```

```
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n, nm1, j, jp1, jm1
      real*4 t, y(n), yprime(n), term2, yima, p_es0
      real*4 p_es, fact_i, car, p, ocr
      common /blocki/ p_es, fact_i
c
      do 20 j = 1, n, 1
           if(y(j).lt.0.0) then
                 y(j) = 0.0
           endif
   20 continue
      yima = y(2)
c
c                 Calculate the 0 flux boundary condition
c

      p = y(1)
      call rates(p, ocr, car)
      term2 = (fact_i)*(y(2) - 2.0*y(1) + yima)
      yprime(1) = term2 - ocr
c
c          Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = y(j)
           call rates(p, ocr, car)
           term2 = (fact_i)*(y(jp1) - 2.0*y(j) + y(jm1))
           yprime(j) = term2 - ocr
   30 continue
c
c           Calculate the known pressure at the free surface
c
      if(t.eq.0.0) then
           p_es0 =(p_es-y(1))/2.0
           p = y(n)
           call rates(p, ocr, car)
           term2 = (fact_i)*(p_es0 - 2.0*y(n) + y(nm1))
           yprime(n) = term2 - ocr
           goto 100
      endif
c
      p = y(n)
      call rates(p, ocr, car)
      term2 = (fact_i)*(p_es - 2.0*y(n) + y(nm1))
      yprime(n) = term2 - ocr
c
```

```
  100 continue
      return
      end
c
c*****************************************************************************
c
      subroutine equation2 (n2, t, z, zprime)
c
      integer n, nm1, n2, j, jp1, jm1
      real*4 t, z(n2), zprime(n2), term2, yima
      real*4 delx, delxsq, term1, ca, diff, diffp, p, car
      real*4 p_es
      common /blocki/ p_es, fact_i
      common /blocke/ delx, delxsq
c
      n = (n2/2)
      do 20 j = 1, n, 1
           if(z(j).lt.0.0) then
                z(j) = 0.0
           endif
   20 continue
      zima = z(2)
c
c                 Calculate the 0 flux boundary condition
c
      p = z(1)
      ca = z(n+1)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
      fact = (diff/delxsq)
      term2 = fact*(z(2) - 2*z(1) + zima)
      zprime(1) = term2 - ocr
      zprime(n+1) = car
c
c          Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = z(j)
           ca = z(n+j)
           call rates (p, ocr, car)
           call diffus (p, ca, diff, diffp)
           fact = (diff/delxsq)
           term1 = (diffp/(4*delxsq))*((z(jp1) - z(jm1))**2)
           term2 = fact*(z(jp1) - 2.0*z(j) + z(jm1))
           zprime(j) = term1 + term2 - ocr
           zprime(n+j) = car
   30 continue
c
```

```
c           Calculate the known pressure at the free surface
c
      p = z(n)
      ca = z(n+n)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
      fact = (diff/delxsq)
      term1 = (diffp/(4.0*delxsq))*((p_es - z(nm1))**2)
      term2 = fact*(p_es - 2.0*z(n) + z(nm1))
      zprime(n) = term1 + term2 - ocr
      zprime(n+n) = car
      return
      end
c
c*******************************************************************************
c
      subroutine diffus (p, ca, diff, diffp)
c
      real*4 p, ca
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
c
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      if(p.eq.0) then
           cap = 0.0
           goto 100
      endif
      cap = alpha*ca/p
c
      if(cap.gt.1.0) then
           cap = 1.0
           goto 100
      endif
c
  100 continue
      diffp = bdif*hs*diff*cap
      return
      end
c
c*******************************************************************************
c
      subroutine rates (p, ocr, car)
c
      real*4 alpha, kp, conv, ocr, car
      common /blockb/ alpha, kp, conv
c
      car = kp*(p**alpha)
      ocr = conv*car
```

```
      return
      end
c
c************************************************************************
c
      subroutine carbonyl (p, t, ca_i)
c
      real*4 p, t, ca_i
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /blocka/ hs, m, adif, bdif, s, beta
      common /blockb/ alpha, kp, conv
c
      ca_i = t*kp*(p**alpha) + s*(p**beta)
      return
      end
c
c************************************************************************
c
      subroutine trap(z, ca_es, n2, n, ca_avg)
c
      integer n2, n
      real*4 z(n2), ca_es, ca_avg, length
      common /blocke/ delx, delxsq
      common /blockt/ length
c
      sum = 0.0
      do 10 j = n+2, n2, 1
          sum = sum + z(j)
   10 continue
c
      area = (delx/2.0)*(z(n+1) + 2.0*sum + ca_es)
      ca_avg = area/length
      return
      end
c
c************************************************************************
c
      subroutine output (y, n, t, iflag, timei, nflag)
c
      integer n, ido, istep, iflag, nflag
      integer inc,  nn
      real*4 t, y(n), tend, timei, tm1
      real*4 p_es, fact_i
      common /blocki/ p_es, fact_i
      common /blockd/ inc, nn
c
      do 10 j = 1, n, 1
          if(y(j).lt.0.0) then
              y(j) = 0.0
          endif
   10 continue
```

```
c
      if(iflag.eq.0) then
           if(t.ge.timei) then
                nflag = 1
           endif
           return
      endif
c
      tm1 = t - 1
      if(tm1.eq.0.0) then
           return
      endif
c
      if(tm1.ge.timei) then
           nflag = 1
      endif
c
 2000 format(' ',2x,f6.0,' , ',5(f6.3,' , '),f6.3)
      return
      end
c
c****************************************************************************
c
      subroutine output2 (z, n2, t)
c
      integer n, nflag, inc, nn, n2
      real*4 t, z(n2), tend
      real*4 p_es, fact_i
      common /blocki/ p_es, fact_i
      common /blockd/ inc, nn
c
      n = (n2/2)
      do 10 j = 1, n, 1
           if(z(j).lt.0.0) then
                z(j) = 0.0
           endif
   10 continue
      call carbonyl(p_es, t, ca_es)
      write(9,2000) t, (z(j), j=1,nn,inc),  p_es
      write(10,2000) t, (z(j+n), j = 1,nn,inc), ca_es
c
 2000 format(' ',2x,f6.0,' , ',5(f6.3,' , '),f6.3)
      return
      end
```

APPENDIX E

LABORATORY AND FIELD DATA

Table E-1. Dickens Cosden AC-10 POV and Field Data[a]

Description	Time days	CA	η_0^*[b] kP	$(1/J'')^c \times 10^{-3}$ dyne / cm^2	MW
POV Aged at 355.5 K, 20 atm					
	2	1.240	22.7	182.1	–
	4	1.423	39.5	295.2	1949
	6	1.567	72.0	489.4	–
	8	1.674	137.0	780.9	2075
	10	1.810	236.0	1206.0	–
	12	2.058	350.0	1637.0	2241
	14	2.066	680.0	2552.0	–
	16	2.209	1200.0	3627.0	2266
	20	2.284	2230.0	5454.0	–
	22	2.383	4610.0	8377.0	2309
Cores from February 1993, Station #599+00 originally placed in 1982					
Auto	–	1.522	120.0	805.9	2548
Micro	–	1.590	115.0	698.9	2434
Micro	–	1.521	120.0	679.2	2561

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-2. Dickens Cosden AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 355.5 K, 20 atm					
	2	1.406	44.0	329.0	–
	4	1.530	90.0	612.0	2032
	6	1.679	145.0	923.0	–
	8	1.861	240.0	1428.0	2134
	10	1.996	414.0	1993.0	–
	12	2.175	780.0	3023.0	2242
	14	2.252	1200.0	3950.0	–
	16	2.339	2200.0	6083.0	2358
	20	2.484	3600.0	8679.0	–
	22	2.549	6300.0	10610.0	2379
Cores from February 1993, Station #550+00 originally placed in 1982					
Auto	–	1.947	1200.0	4226.0	2904
Micro A	–	1.820	441.0	1650.0	2963
Micro B	–	1.733	394.0	1481.0	2975
Micro C	–	1.846	540.0	1991.0	2938
Micro D	–	1.883	568.0	2115.0	2925

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-3. Dickens Diamond Shamrock AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_0^* [b] kP	$(1/J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 355.5 K, 20 atm					
	2	1.050	29.0	187.0	-
	4	1.256	41.0	254.0	4576
	6	1.422	82.0	412.0	-
	8	1.571	113.0	517.0	4730
	10	1.625	125.0	637.0	-
	12	1.656	165.0	762.0	4904
	14	1.747	286.0	1068.0	-
	16	1.816	460.0	1232.0	5163
	20	2.041	750.0	1761.0	-
	22	2.116	960.0	1974.0	5299
Cores from February 1993, Station #458+00 originally placed in 1982					
Auto	-	2.046	6660.0	6720.0	3743
Micro A	-	1.670	714.0	1718.0	3775
Micro B	-	1.714	890.0	1858.0	3730
Micro C	-	1.728	830.0	1904.0	3724
Micro D	-	1.779	770.0	1752.0	3665

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-4. Dickens Dorchester AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_0^*[b] kP	$(1/J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 355.5 K, 20 atm					
	2	0.884	41.0	232.0	–
	4	0.988	75.0	386.0	2692
	6	1.123	178.0	679.0	–
	8	1.286	231.0	908.0	2862
	10	1.476	600.0	1394.0	–
	12	1.561	1200.0	2114.0	3164
	14	1.637	2900.0	2918.0	–
	16	1.718	3000.0	3465.0	3087
	20	1.856	8600.0	5462.0	–
	22	1.977	10000.0	6395.0	3259
Cores from February 1993, Station #391+00 originally placed in 1982					
Auto	–	1.875	2600.0	4437.0	3586
Micro A	–	1.881	596.0	1847.0	3264
Micro B	–	1.907	784.0	2172.0	3415
Micro C	–	1.978	750.0	2145.0	3249
Micro D	–	1.955	860.0	2437.0	3222

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-5. Dickens Exxon AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_0^{*}[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 355.5 K, 20 atm					
	2	1.272	51.0	367.0	-
	4	1.556	97.0	597.0	1993
	6	1.369	107.0	504.0	-
	8	1.899	288.0	1284.0	2147
	10	1.960	483.0	1923.0	-
	12	2.005	650.0	2256.0	2198
	14	2.132	980.0	2963.0	-
	16	2.197	1300.0	3443.0	2350
	20	2.314	2800.0	5530.0	-
	22	2.479	4200.0	7078.0	2482
Cores from February 1993, Station #322+00 originally placed in 1982					
Auto	-	1.943	470.0	2038.0	2637
Micro A	-	1.935	220.0	1111.0	2640
Micro B	-	1.918	185.0	1041.0	2739
Micro C	-	1.835	200.0	1047.0	2645
Micro D	-	1.835	220.0	1109.0	2671

[a] - Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-6. Dickens MacMillan AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 355.5 K, 20 atm					
	2	0.989	21.0	131.0	-
	4	1.119	37.0	210.0	3185
	6	1.186	70.0	342.0	-
	8	1.269	85.0	408.0	3451
	10	1.370	114.0	538.0	-
	12	1.419	140.0	621.0	3638
	14	1.472	250.0	791.0	-
	16	1.544	360.0	968.0	3761
	20	1.642	590.0	1235.0	-
	22	1.726	580.0	1289.0	3923
Cores from February 1993, Station #270+00 originally placed in 1982					
Auto	-	1.518	120.0	681.0	3827
Micro A	-	1.378	120.0	521.0	3751
Micro B	-	1.441	150.0	526.0	3816
Micro C	-	1.459	120.0	617.0	3761
Micro D	-	1.430	170.0	549.0	3839

[a] - Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-7. Pineland Cosden AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 333.3 K, 20 atm					
	1	0.825	14.0	112.0	-
	2	0.803	12.0	109.0	1728
	3	0.837	14.7	133.0	-
	5	0.964	17.2	149.0	1725
	8	1.066	25.3	211.0	-
	10	1.103	30.3	249.0	1806
	14	1.128	38.0	307.0	-
	16	1.243	40.6	320.0	1805
	19	1.292	53.0	399.0	-
	22	1.292	56.4	427.0	1882
Cores from March 1993, Station #570+00 originally placed in 1983					
Auto	-	0.710	5.0	45.6	1803
Micro A	-	0.706	5.8	55.3	1805
Micro B	-	0.734	7.6	71.1	1801
Micro C	-	0.718	6.1	57.3	1794
Micro D	-	0.740	6.0	55.8	1792

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-8. Pineland Dorchester AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 333.3 K, 20 atm					
	1	0.769	4.28	39.8	–
	2	0.820	5.44	49.0	1989
	3	0.883	6.50	58.5	–
	5	0.914	7.41	65.7	2020
	8	0.979	11.2	94.5	–
	10	1.010	11.2	95.5	2071
	14	1.069	15.7	129.0	–
	16	1.114	15.0	123.0	2122
	19	1.156	19.9	169.0	–
	22	1.173	22.0	156.0	2151
Cores from March 1993, Station #510+00 originally placed in 1983					
Auto	–	0.858	10.6	85.2	2306
Micro B	–	0.954	14.5	108.0	2383
Micro C	–	0.949	16.0	119.0	2422
Micro D	–	0.861	10.6	84.3	2340

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-9. Pineland Exxon AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_0^*[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 333.3 K, 20 atm					
	1	0.776	6.70	63.7	-
	2	0.837	8.00	75.1	1695
	3	0.939	9.80	94.8	-
	5	1.088	12.0	114.0	1717
	8	1.096	14.8	137.0	-
	10	1.146	17.2	157.0	1752
	14	1.305	20.0	183.0	-
	16	1.300	21.7	197.0	1795
	19	1.322	27.0	236.0	-
	22	1.352	27.0	236.0	1817
Cores from March 1993, Station #640+00 originally placed in 1983					
Auto	-	0.928	6.8	65.5	1844
Micro A	-	0.889	7.3	71.0	1842
Micro B	-	0.900	7.0	68.4	1840
Micro C	-	0.922	6.5	61.0	1829
Micro D	-	0.877	6.2	59.8	1837

[a] - Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-10. Pineland MacMillan AC-20 POV and Field Data[a]

Description	Time days	*CA*	η_o^*[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 333.3 K, 20 atm					
	1	0.753	6.20	52.8	–
	2	0.800	8.25	67.6	2687
	3	0.831	10.4	80.8	–
	5	0.906	15.0	113.0	2769
	8	0.953	21.0	146.0	–
	10	1.014	26.0	174.0	2835
	14	1.072	32.8	211.0	–
	16	1.063	39.0	243.0	2853
	19	1.140	45.0	270.0	–
	22	1.161	47.4	286.0	2908
Cores from March 1993, Station #557+00 originally placed in 1983					
Auto	–	0.760	4.9	41.0	2835
Micro A	–	0.964	12.0	91.1	3086
Micro B	–	0.990	13.0	96.2	3106
Micro C	–	1.002	17.0	118.0	3154
Micro D	–	0.988	13.5	99.6	3148

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-11. Pineland Texaco AC-20 POV and Field Data[a]

Description	Time days	*CA*	$\eta_o^{*\,b}$ kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
POV Aged at 333. K, 20 atm					
	1	0.654	3.90	36.6	-
	2	0.777	5.00	46.4	2021
	3	0.774	5.80	52.8	-
	5	0.868	7.70	69.5	2092
	8	0.884	9.74	84.8	-
	10	0.972	11.8	101.0	2117
	14	1.060	13.1	112.0	-
	16	1.081	15.8	130.0	2182
	19	1.129	19.2	155.0	-
	22	1.196	20.0	161.0	2236
Cores from March 1993, Station #285+00 originally placed in 1983					
Auto	-	0.731	5.60	53.2	2005
Micro A	-	0.789	5.44	50.1	2129
Micro B	-	0.755	4.32	41.0	2052
Micro C	-	0.736	4.80	45.0	2071
Micro D	-	0.723	4.30	39.8	2103

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-12. Bryan Exxon AC-20 #8 Aging Data[a]

Description	Time days	*CA*	$\eta_o^{*\,b}$ kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	*MW*
RTFO	0	0.588	5.550	53.7	1827
POV Aged at 344.4 K, 20 atm					
	4	1.095	18.7	171.0	–
	8	1.261	29.6	246.0	1856
	12	1.373	46.9	369.0	–
	17	1.632	83.0	597.0	2008
	24	1.814	144.0	945.0	–
	28	1.855	165.0	1081.0	2091
	33	1.985	330.0	1792.0	–
	36	2.075	397.0	2031.0	2175
	44	2.235	706.0	3029.0	–
POV Aged at 355.5 K, 20 atm					
	2	0.995	16.9	152.0	–
	4	1.223	31.2	263.0	–
	6	1.407	44.5	363.0	–
	8	1.430	58.5	455.0	–
	10	1.646	–	–	–
	12	1.647	129.0	872.0	–
	14	1.807	134.0	908.0	–
	16	1.855	262.0	1504.0	–
	21	2.090	–	3740.0	–
	22	2.113	–	2690.0	–

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-13. Bryan Exxon AC-20 #15 Aging Data[a]

Description	Time days	*CA*	η_o^{*}[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
RTFO	0	0.617	5.470	53.8
POV Aged at 344.4 K, 20 atm				
	4	1.112	16.8	152.0
	8	1.236	24.8	225.0
	12	1.444	37.0	324.0
	17	1.657	71.0	581.0
	24	1.848	98.0	735.0
	28	1.921	112.0	918.0
	33	1.984	207.0	1400.0
	36	2.117	265.0	1630.0
	40	2.103	345.0	1960.0
	44	2.143	440.0	2380.0

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-14. Bryan Exxon AC-20 #16 Aging Data[a]

Description	Time days	*CA*	η_o^*[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
POV Aged at 344.4 K, 20 atm				
	4	1.096	16.4	153.0
	8	1.215	24.4	219.0
	12	1.395	35.5	308.0
	17	1.640	64.1	514.0
	24	1.839	100.0	723.0
	28	1.822	113.0	857.0
	33	1.974	185.0	1270.0
	36	2.038	250.0	1630.0
	40	2.086	321.0	1930.0
	44	2.235	420.0	2310.0

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-15. Bryan Exxon AC-20 #18 Aging Data[a]

Description	Time days	*CA*	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
RTFO	0	0.642	4.84	47.5
POV Aged at 344.4 K, 20 atm				
	4	1.076	16.3	150.0
	8	1.234	23.8	213.0
	12	1.342	31.7	276.0
	17	1.625	60.0	497.0
	24	1.719	88.0	683.0
	28	1.849	100.0	781.0
	33	1.964	173.0	1820.0
	36	2.086	240.0	1550.0
	40	2.173	298.0	1830.0
	44	2.258	423.0	2270.0
POV Aged at 355.5 K, 20 atm				
	2	1.026	14.5	136.0
	4	1.173	24.3	216.0
	6	1.398	34.9	300.0
	8	1.435	43.1	368.0
	10	1.587	64.8	533.0
	12	1.694	92.0	701.0
	14	1.733	91.0	678.0
	16	2.026	167.0	1130.0
	21	1.946	300.0	1780.0
	22	2.104	–	1940.0

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-16. Bryan Exxon AC-20 #19 Aging Data[a]

Description	Time days	CA	η_0^*[b] kP	$(1 / J'')$[c] $\times 10^{-3}$ dyne / cm^2
RTFO	0	0.604	5.60	54.9
POV Aged at 344.4 K, 20 atm				
	4	1.093	16.8	155.0
	8	1.306	25.0	220.0
	12	1.378	34.0	293.0
	17	1.612	68.0	535.0
	24	1.667	91.8	699.0
	28	1.785	117.0	882.0
	33	1.900	188.0	1280.0
	36	1.979	276.0	1700.0
	40	2.051	323.0	1870.0
	44	2.251	453.0	2430.0

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-17. Bryan Exxon AC-20 #1B Aging Data[a]

Description	Time days	CA	η_0^* [b] kP	$(1/J'')^c \times 10^{-3}$ dyne / cm^2
RTFO	0	0.575	5.32	52.1
POV Aged at 344.4 K, 20 atm				
	4	1.033	18.0	157.0
	8	1.217	29.3	246.0
	12	1.309	34.7	283.0
	17	1.515	78.8	573.0
	24	1.644	102.0	703.0
	28	1.887	140.0	940.0
	33	1.820	224.0	1340.0
	36	1.974	369.0	1890.0
	40	2.003	390.0	1970.0
	44	2.083	600.0	2600.0
POV Aged at 355.5 K, 20 atm				
	2	1.039	16.4	143.0
	4	1.243	28.0	239.0
	6	1.367	42.0	337.0
	8	1.450	46.0	379.0
	10	1.552	82.0	594.0
	12	1.714	121.0	808.0
	14	1.744	140.0	960.0
	16	1.800	183.0	1170.0
	21	2.016	–	2270.0
	22	2.014	–	2250.0

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table E-18. Bryan Texas Field Aging[a]

Description	*CA*	η_0^*[b] kP	$(1 / J'')$[c] × 10^{-3} dyne / cm^2	*MW*
Hot Mix from Asphalt originally placed in 1987				
# 1518r	0.656	–	–	1693
# 1518	0.639	–	–	1766
# 1517	0.749	–	–	1774
# 1413	0.659	–	–	1804
# 1394	0.706	–	–	1738
# 1385	0.742	–	–	1799
Cores from 1989 that were placed in 1987				
# 1394 t	0.880	–	–	1922
# 1394 m	0.939	–	–	1844
# 1394 b	1.113	–	–	1951
# 1277 b2	0.876	8.50	82.4	1788
# 1277 m3	0.988	8.20	80.6	1841
# 1277 t3	0.940	11.4	110.0	1752
# 1277 b1	0.888	7.00	68.9	1783
Cores from 1992 that were placed in 1987				
# 1518 m2	1.007	15.0	142.0	1951
# 1483m b3	1.230	24.3	226.0	1908
# 1483m t1	1.118	13.0	123.0	1742
# 1483r m2	1.094	15.5	148.0	1868
# 1465 b1	1.256	45.3	386.0	2066
# 1465 m3	1.297	22.5	210.0	1949
# 1394 b1	1.249	30.0	273.0	1906
# 1394 m1	1.078	15.8	151.0	1855
# 1394 t3	1.044	20.8	190.0	1915
# 1295 b3	1.033	14.0	135.0	1809
# 1295 t3	1.212	30.2	268.0	1883
# 1277 m4	1.079	18.2	170.0	1941

---

[a] – Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

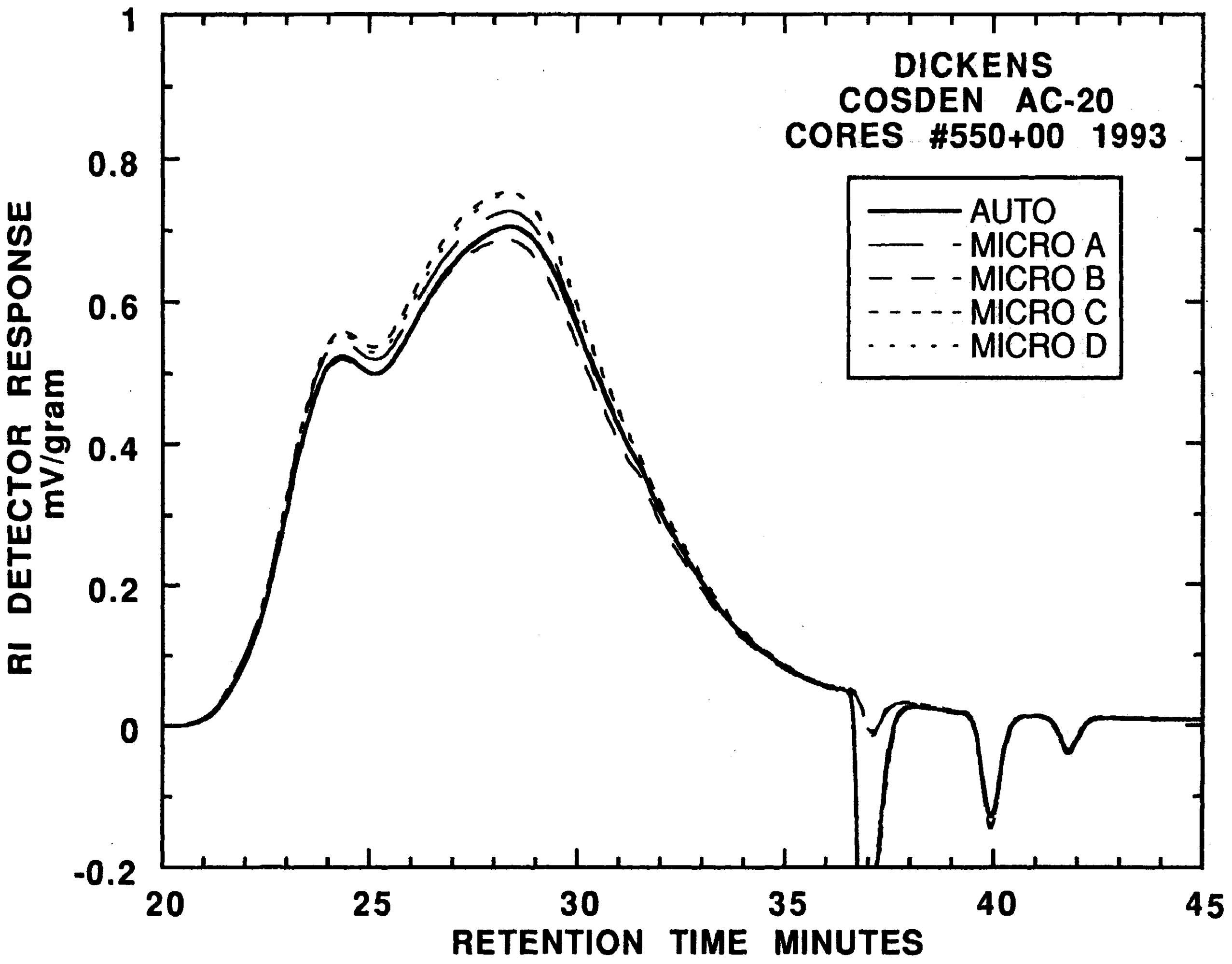

Figure E-1. GPCs of field-aged Dickens Cosden AC-20 extracted asphalt from #550+00. Cored February, 1993.

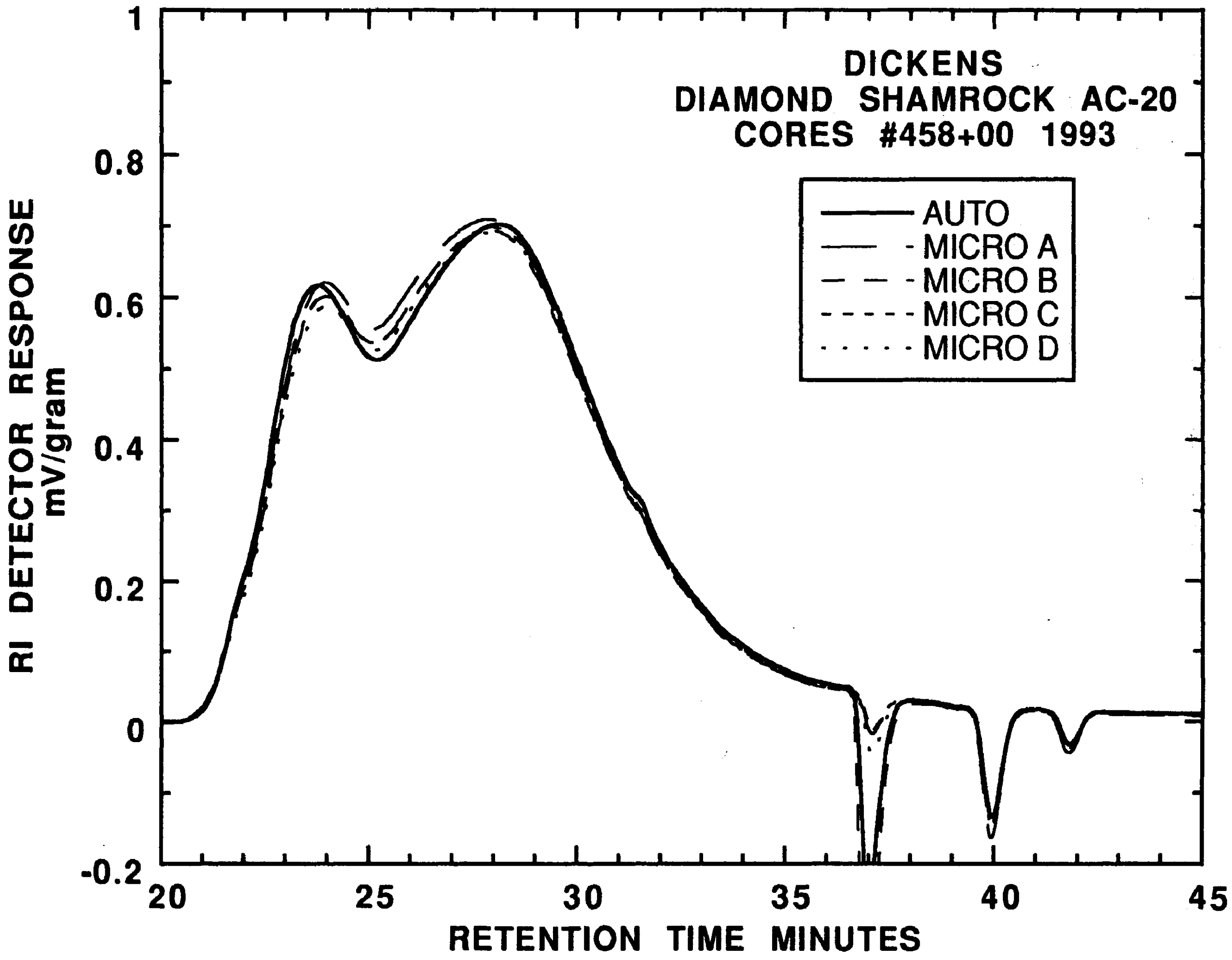

Figure E-2. GPCs of field-aged Dickens Diamond Shamrock AC-20 extracted asphalt from #458+00. Cored February, 1993.

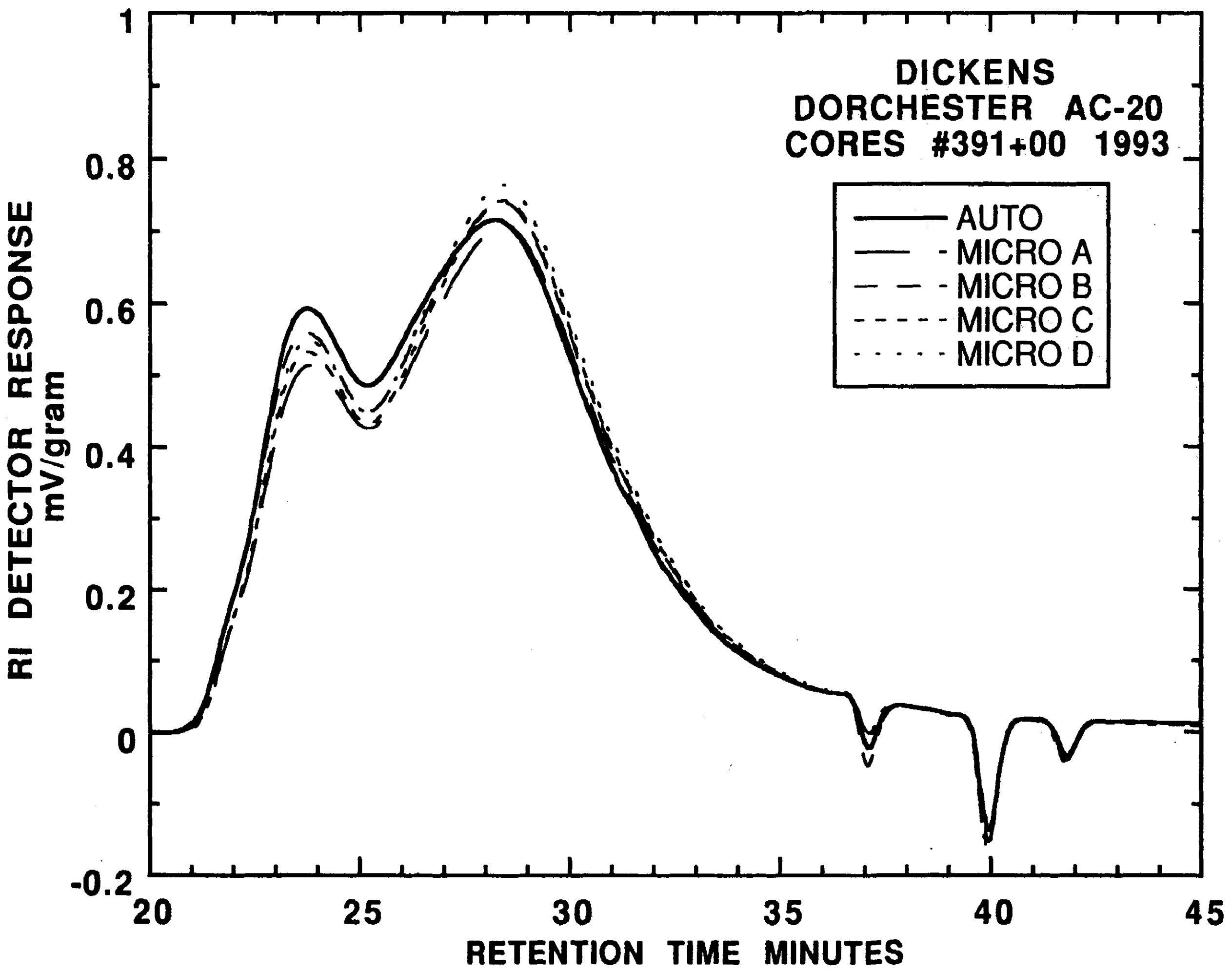

Figure E-3. GPCs of field-aged Dickens Dorchester AC-20 extracted asphalt from #391+00. Cored February, 1993.

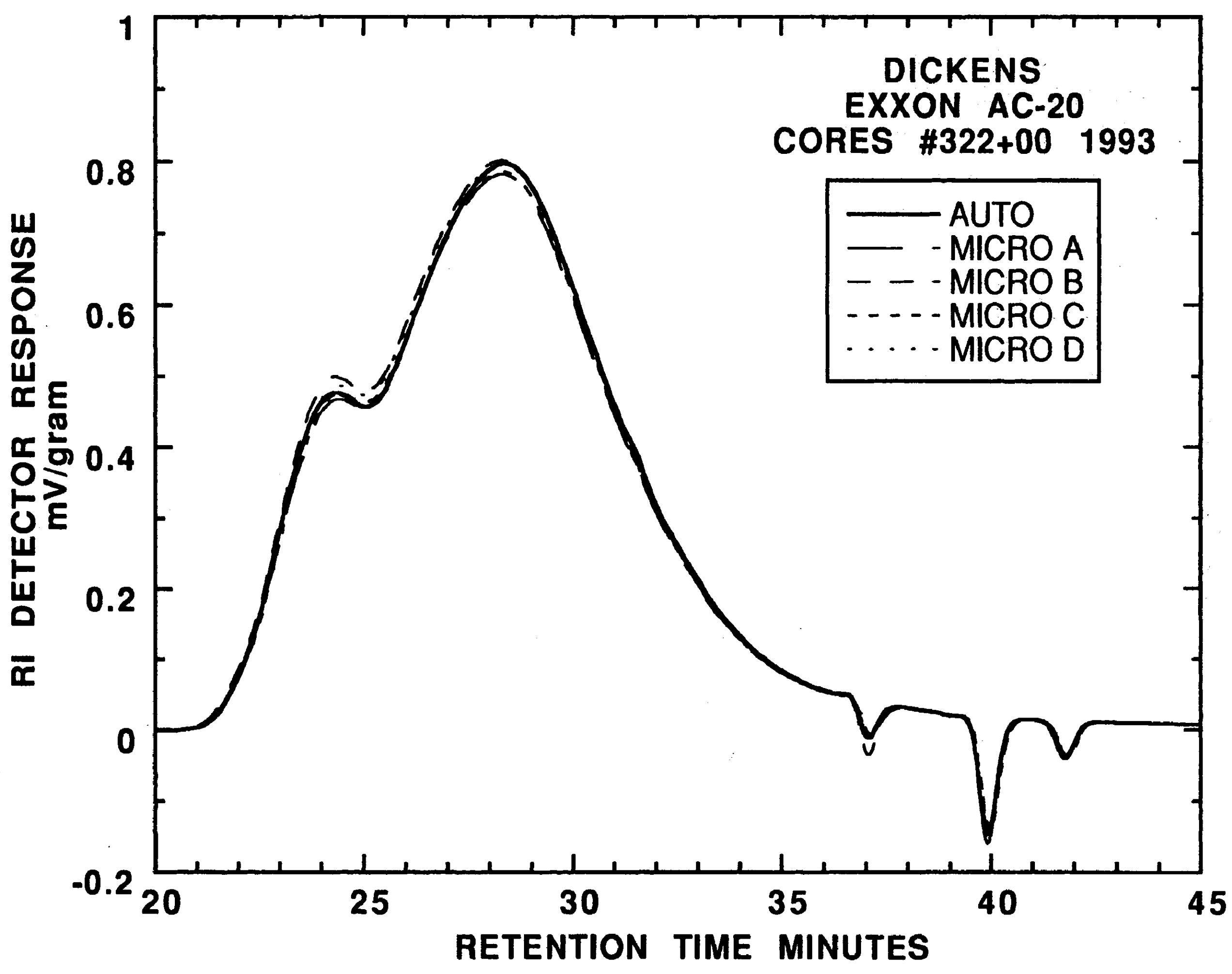

Figure E-4. GPCs of field-aged Dickens Exxon AC-20 extracted asphalt from #322+00. Cored February, 1993.

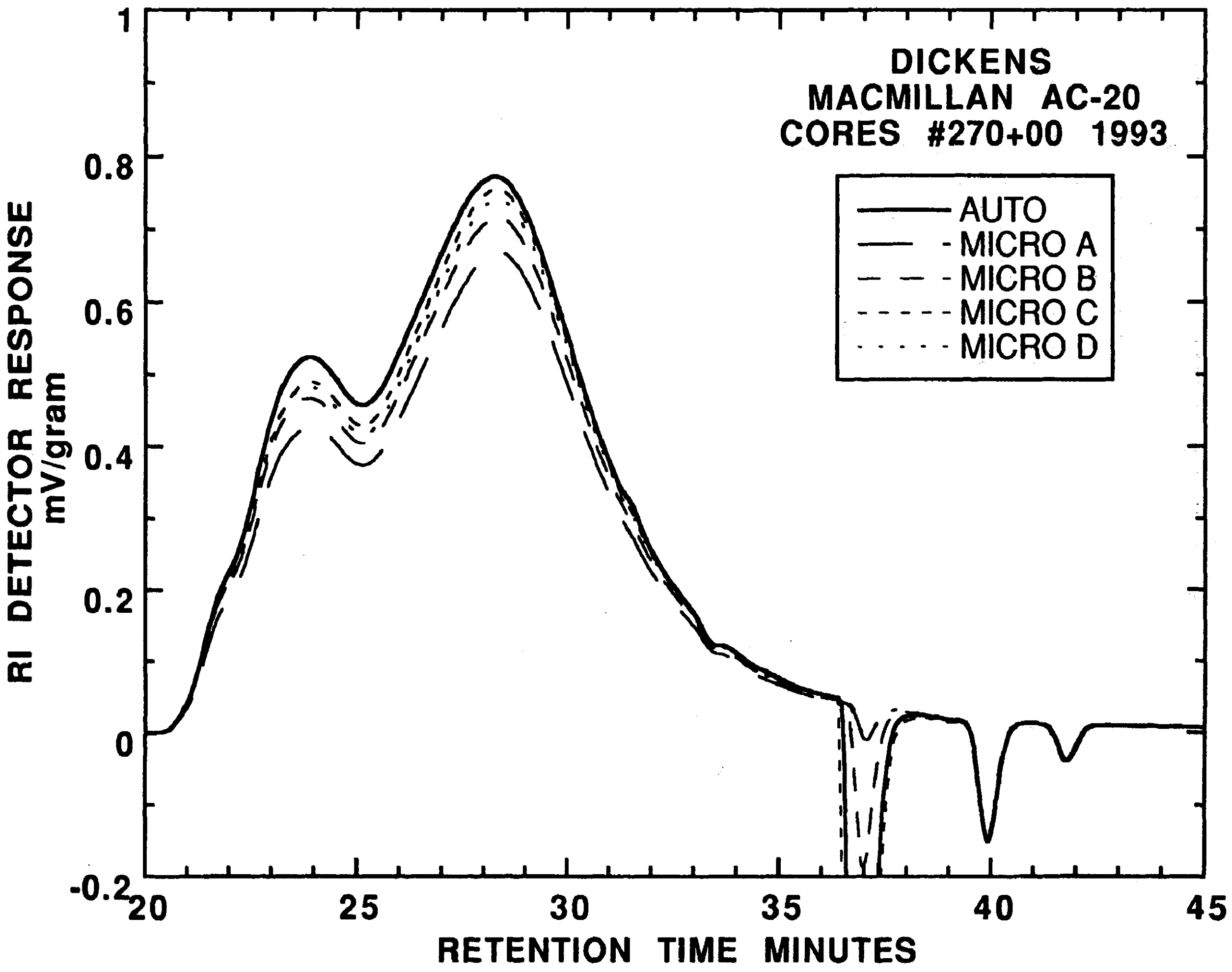

Figure E-5. GPCs of field-aged Dickens MacMillan AC-20 extracted asphalt from #270+00. Cored February, 1993.

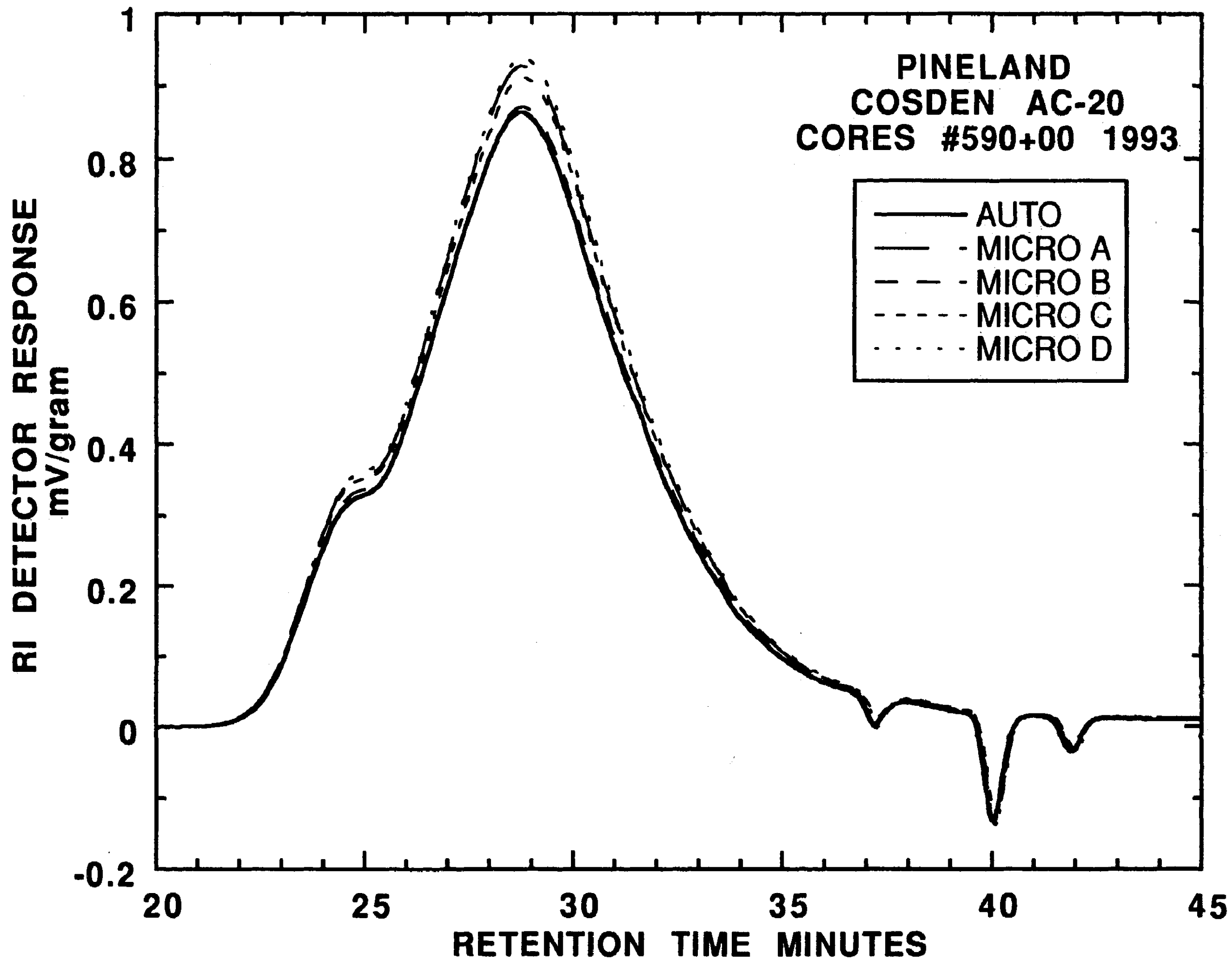

Figure E-6. GPCs of field-aged Pineland Cosden AC-20 extracted asphalt from #590+00. Cored March, 1993.

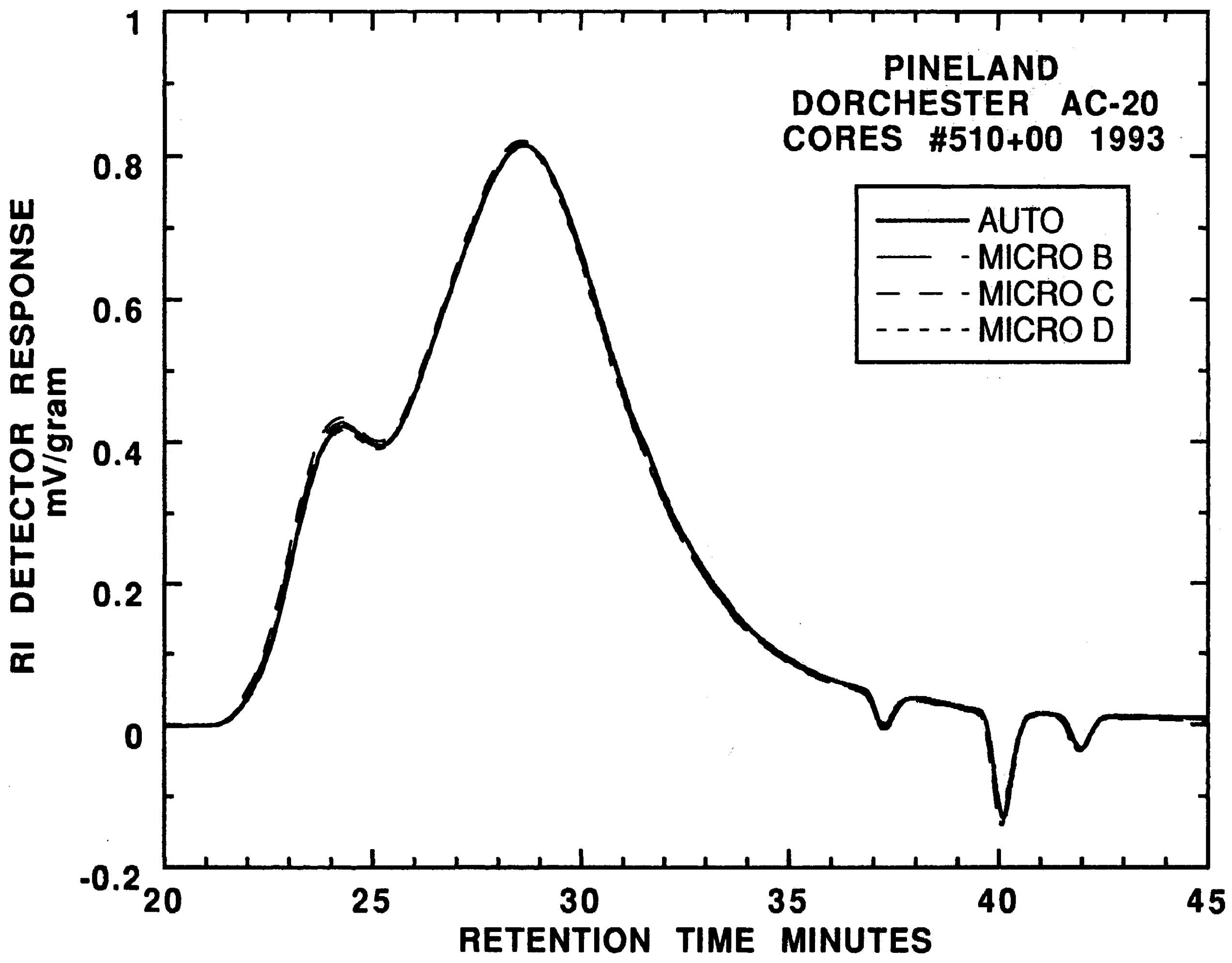

Figure E-7. GPCs of field-aged Pineland Dorchester AC-20 extracted asphalt from #510+00. Cored March, 1993.

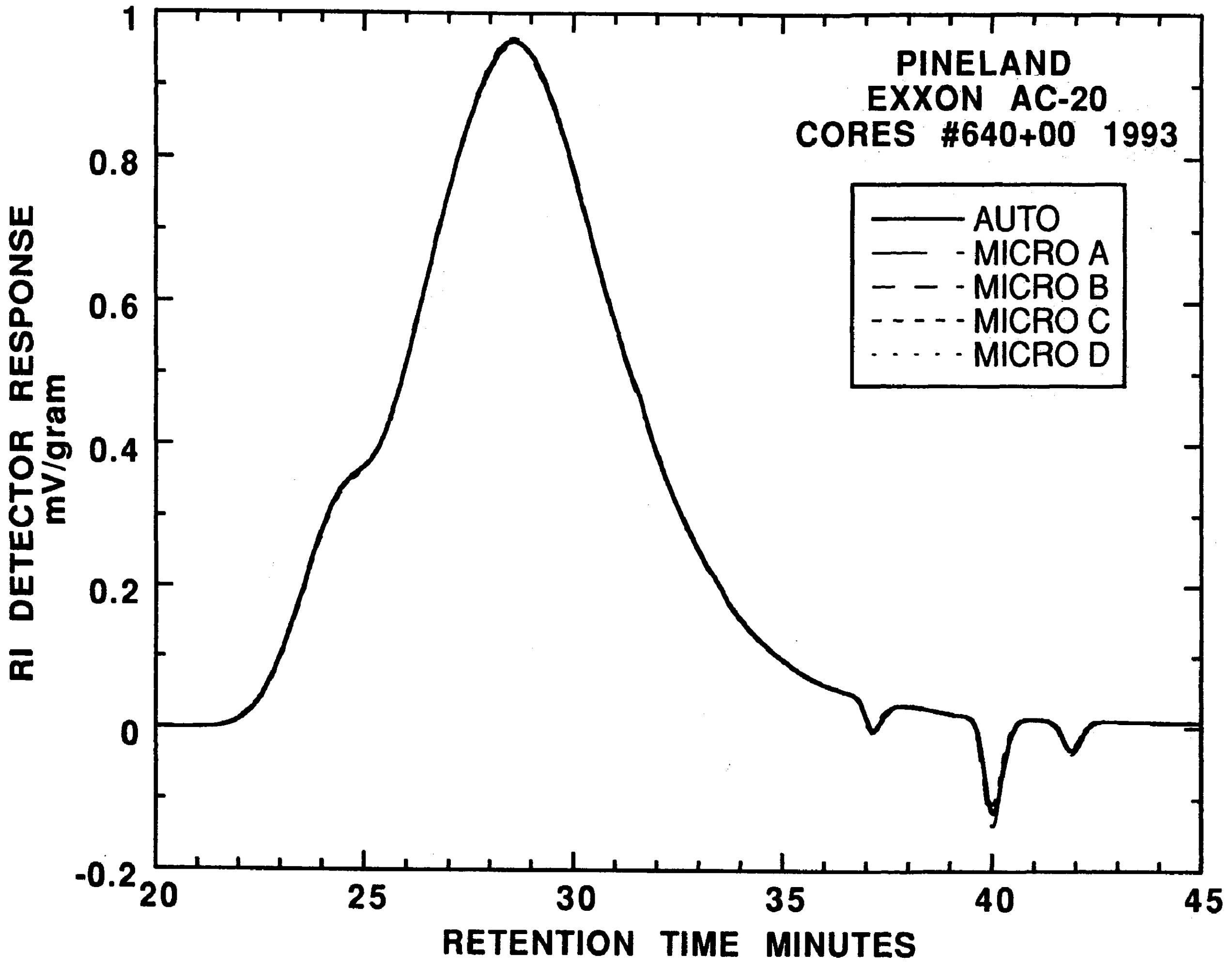

Figure E-8. GPCs of field-aged Pineland Exxon AC-20 extracted asphalt from #640+00. Cored March, 1993.

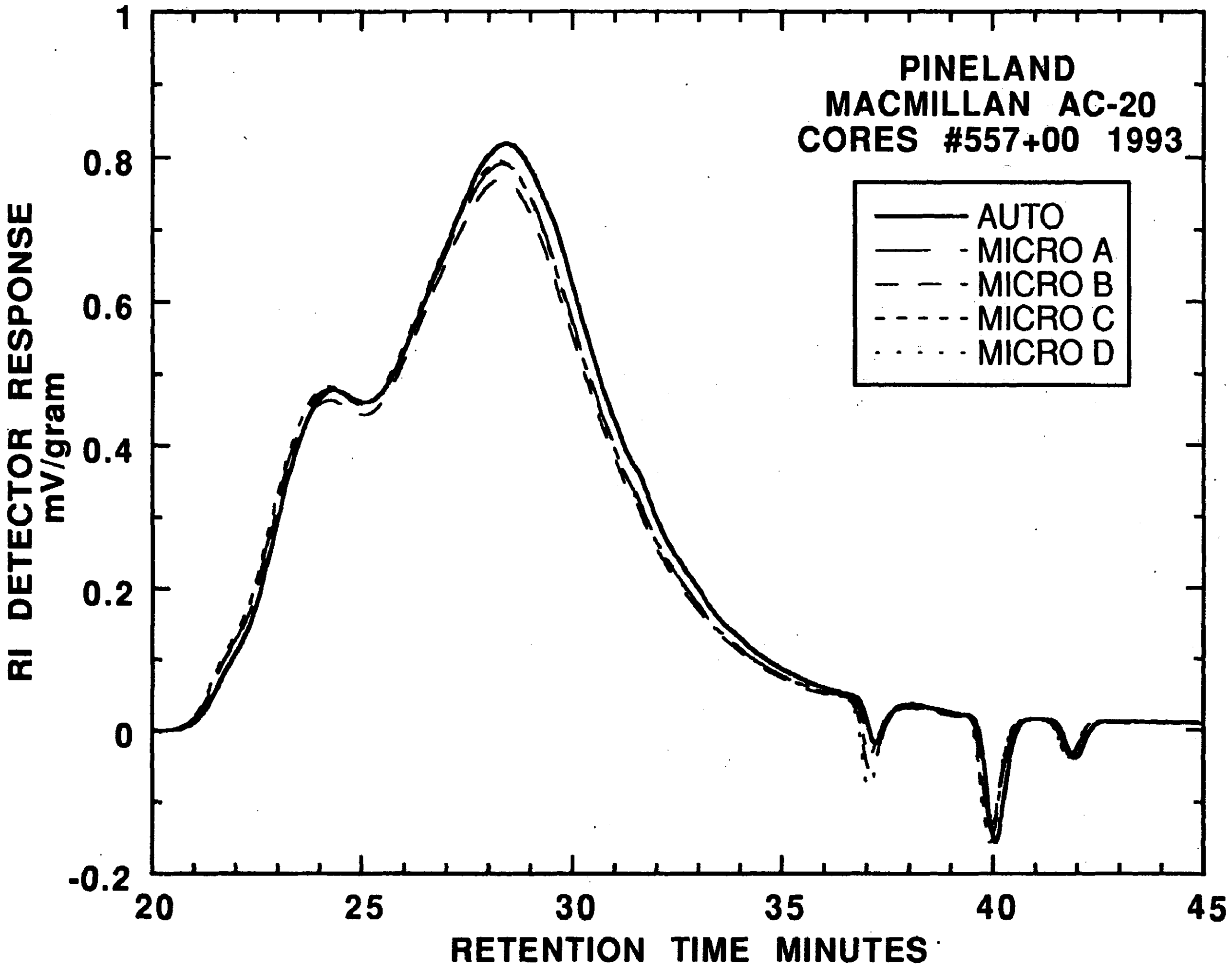

Figure E-9. GPCs of field-aged Pineland MacMillan AC-20 extracted asphalt from #557+00. Cored March, 1993.

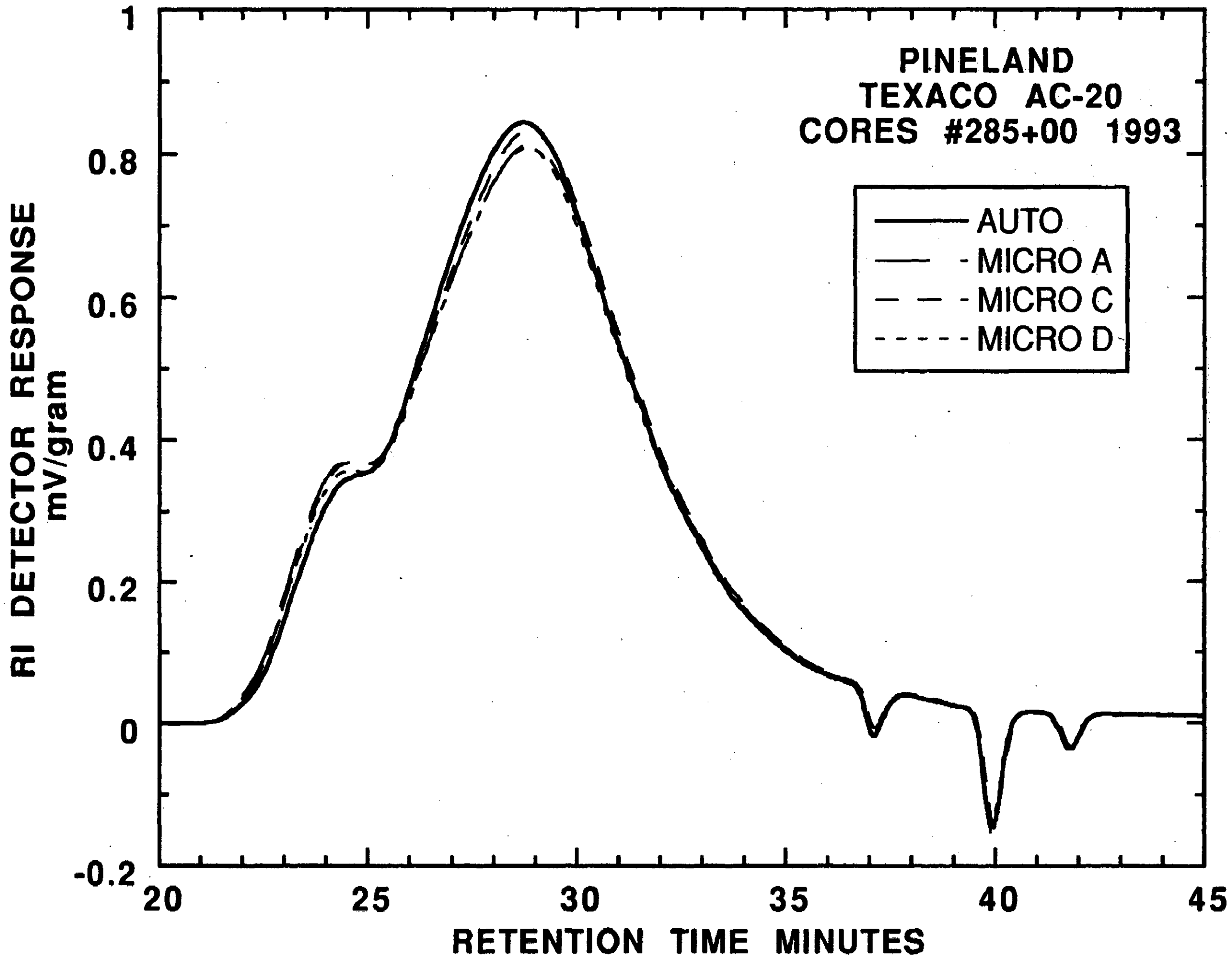

Figure E-10. GPCs of field-aged Pineland Texaco AC-20 extracted asphalt from #285+00. Cored March, 1993.

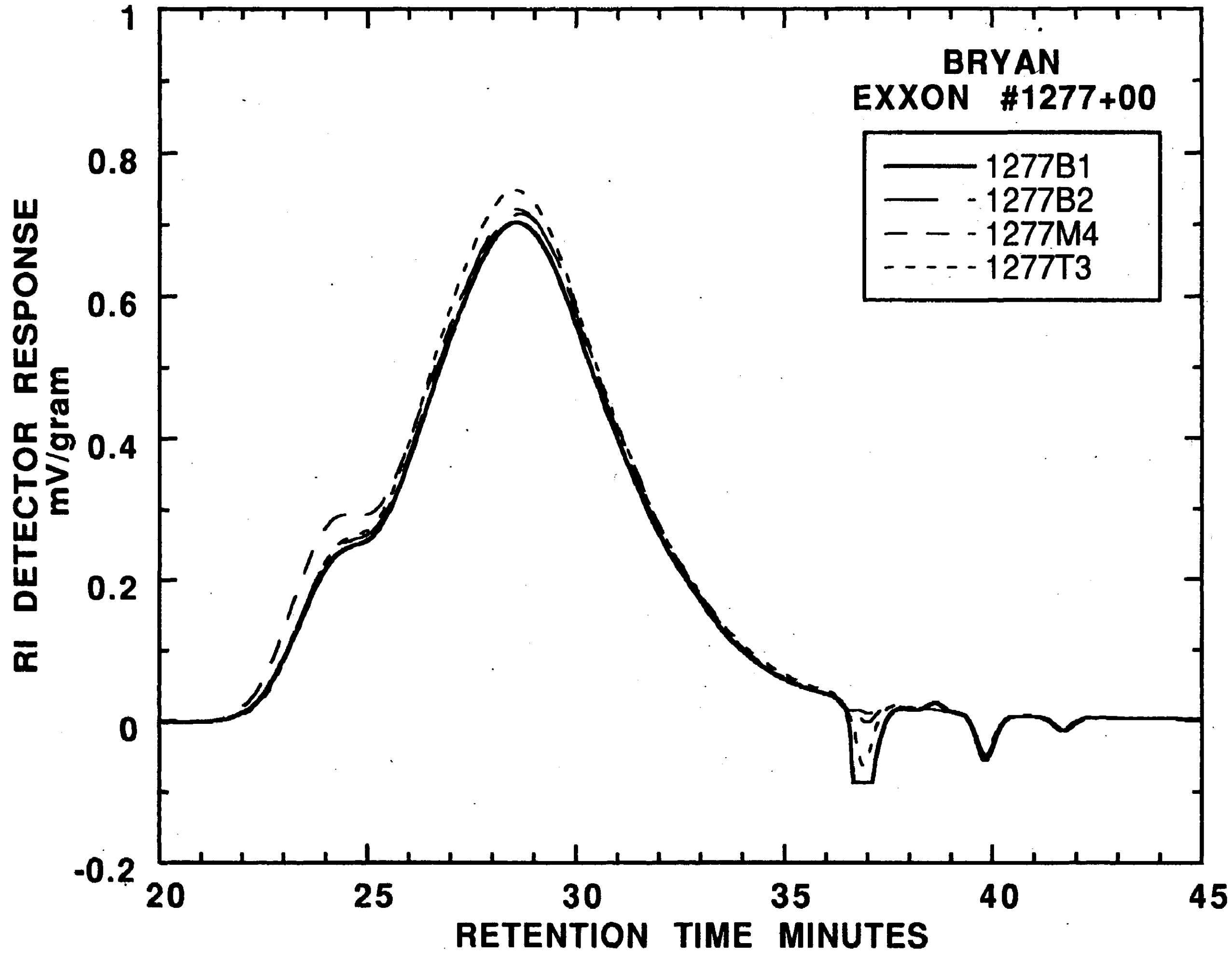

Figure E-11. GPCs of field-aged Bryan Exxon AC-20 extracted asphalt from #1277. Cored November, 1992.

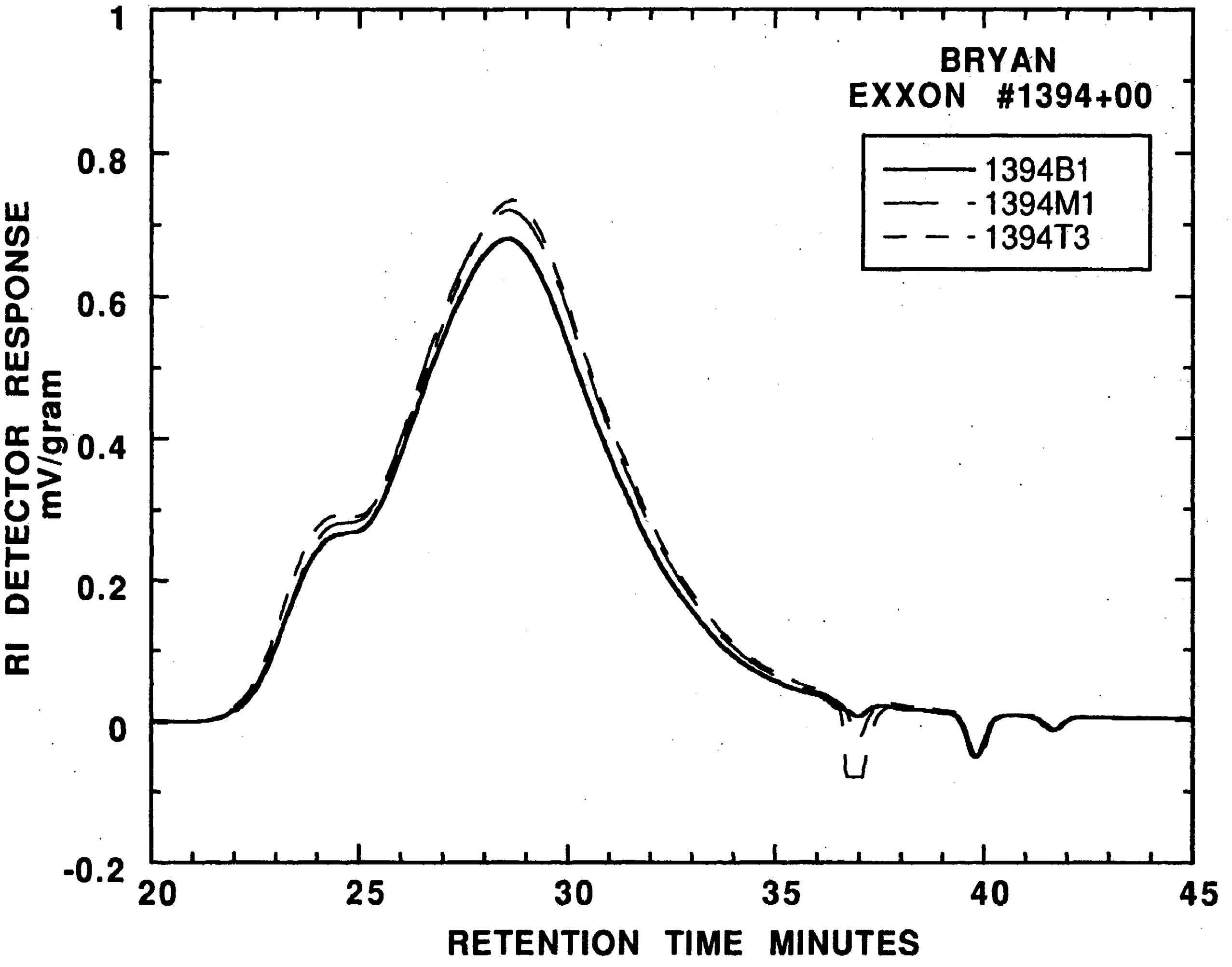

Figure E-12. GPCs of field-aged Bryan Exxon AC-20 extracted asphalt from #1394. Cored November, 1992.

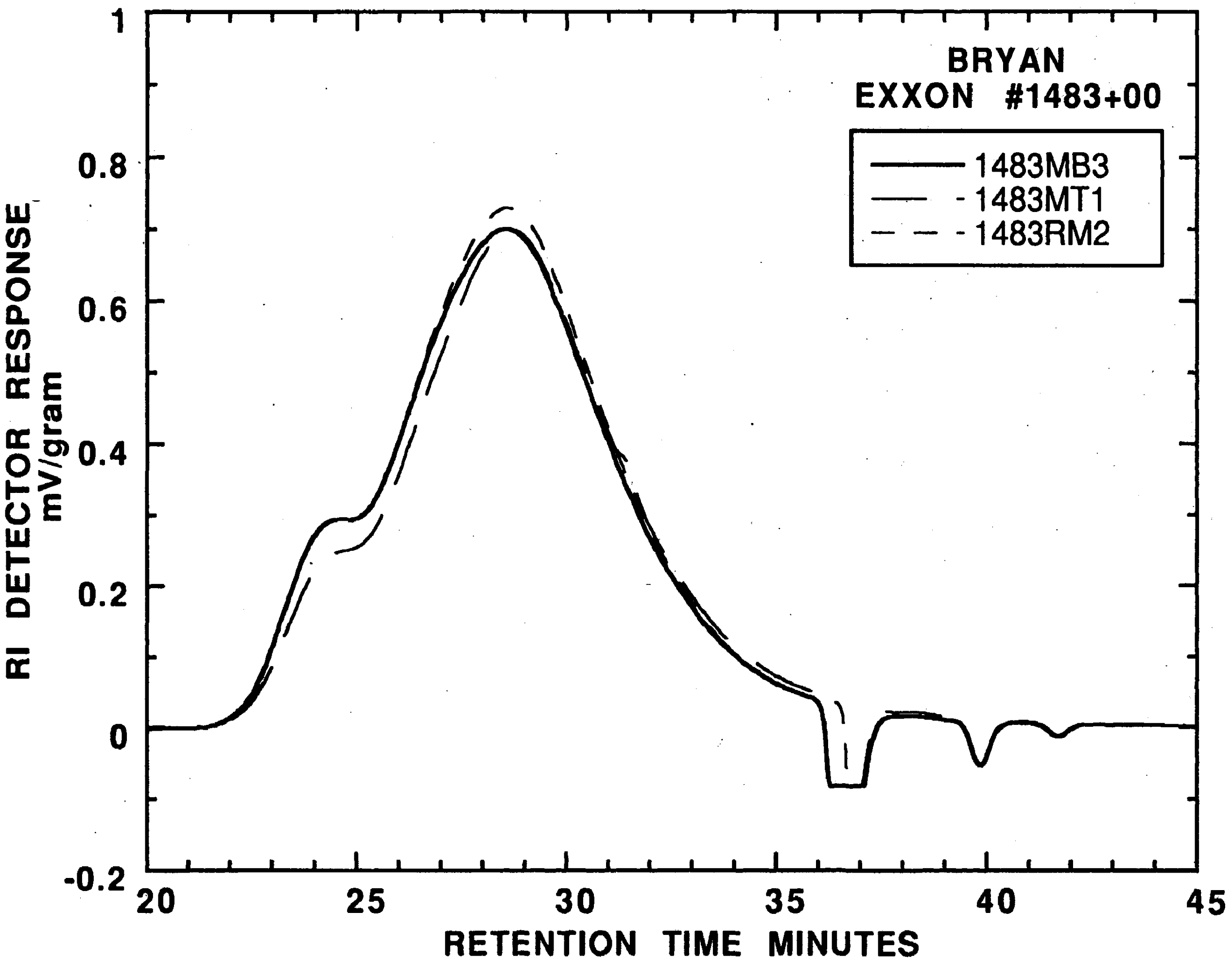

Figure E-13. GPCs of field-aged Bryan Exxon AC-20 extracted asphalt from #1483. Cored November, 1992.

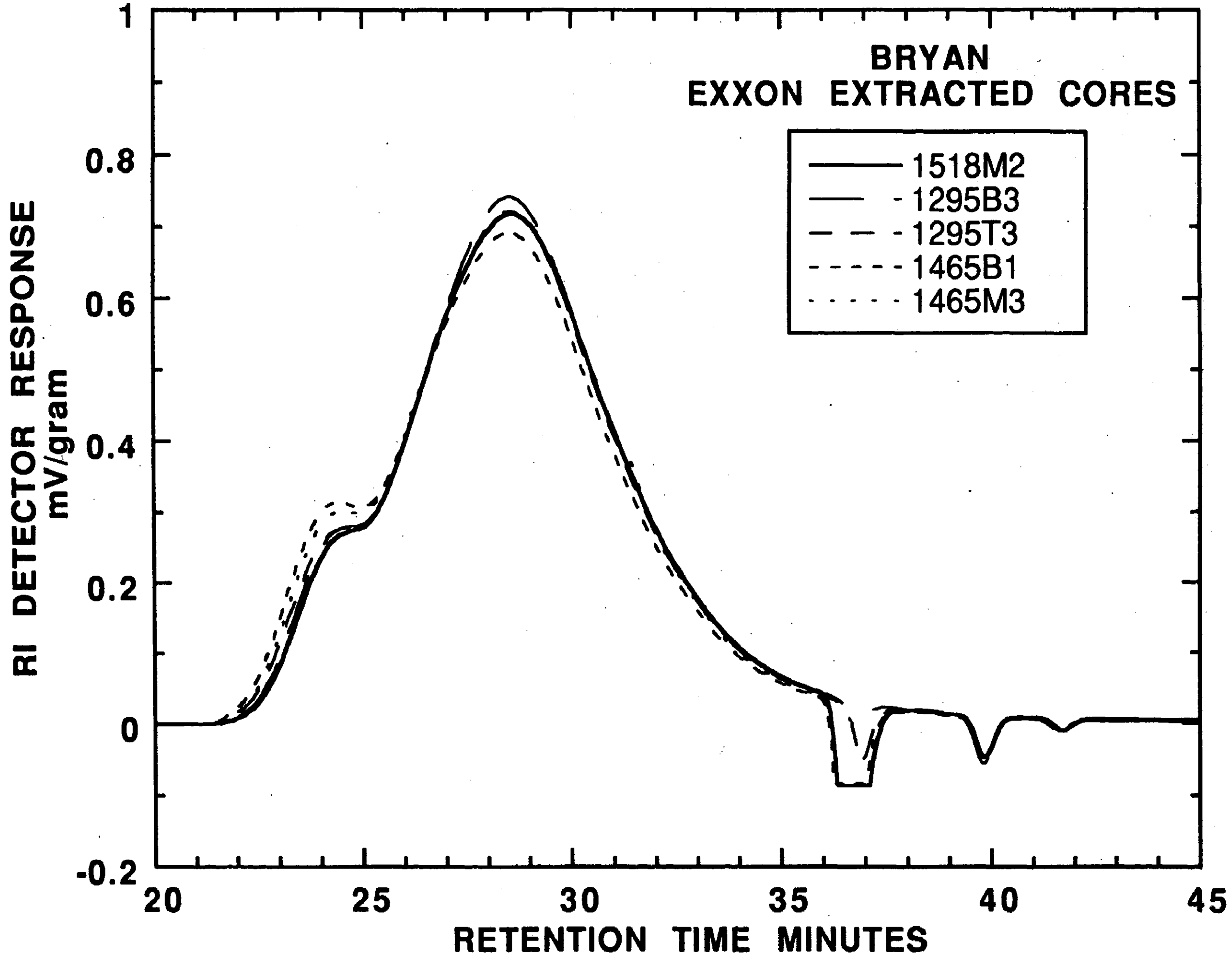

Figure E-14. GPCs of field-aged Bryan Exxon AC-20 extracted asphalt from #1518, #1295, and #1465. Cored November, 1992.

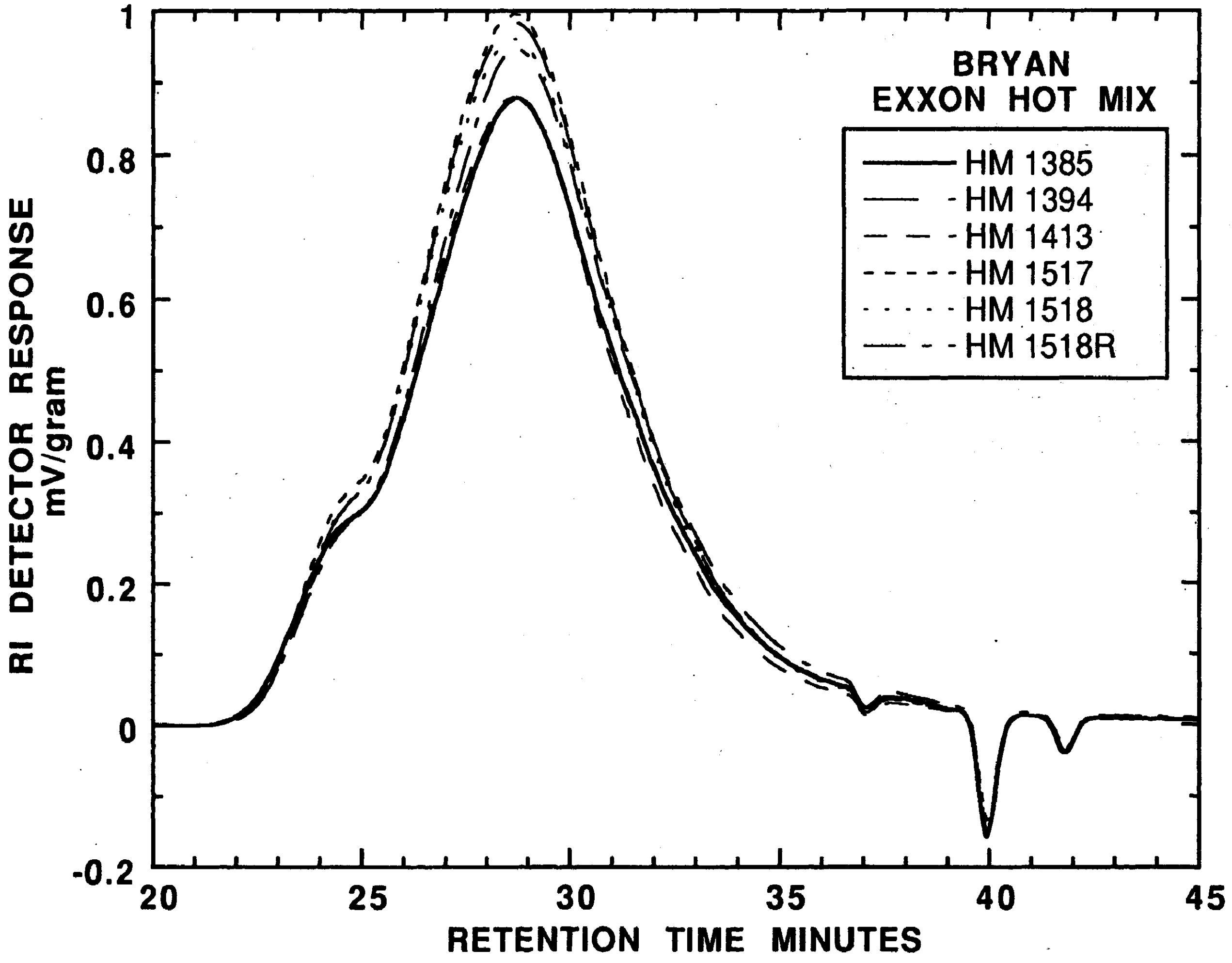

Figure E-15. GPCs of Bryan Exxon AC-20 extracted hot-mix from #1385, #1394, #1413, #1517, and #1518. Obtained 1987.

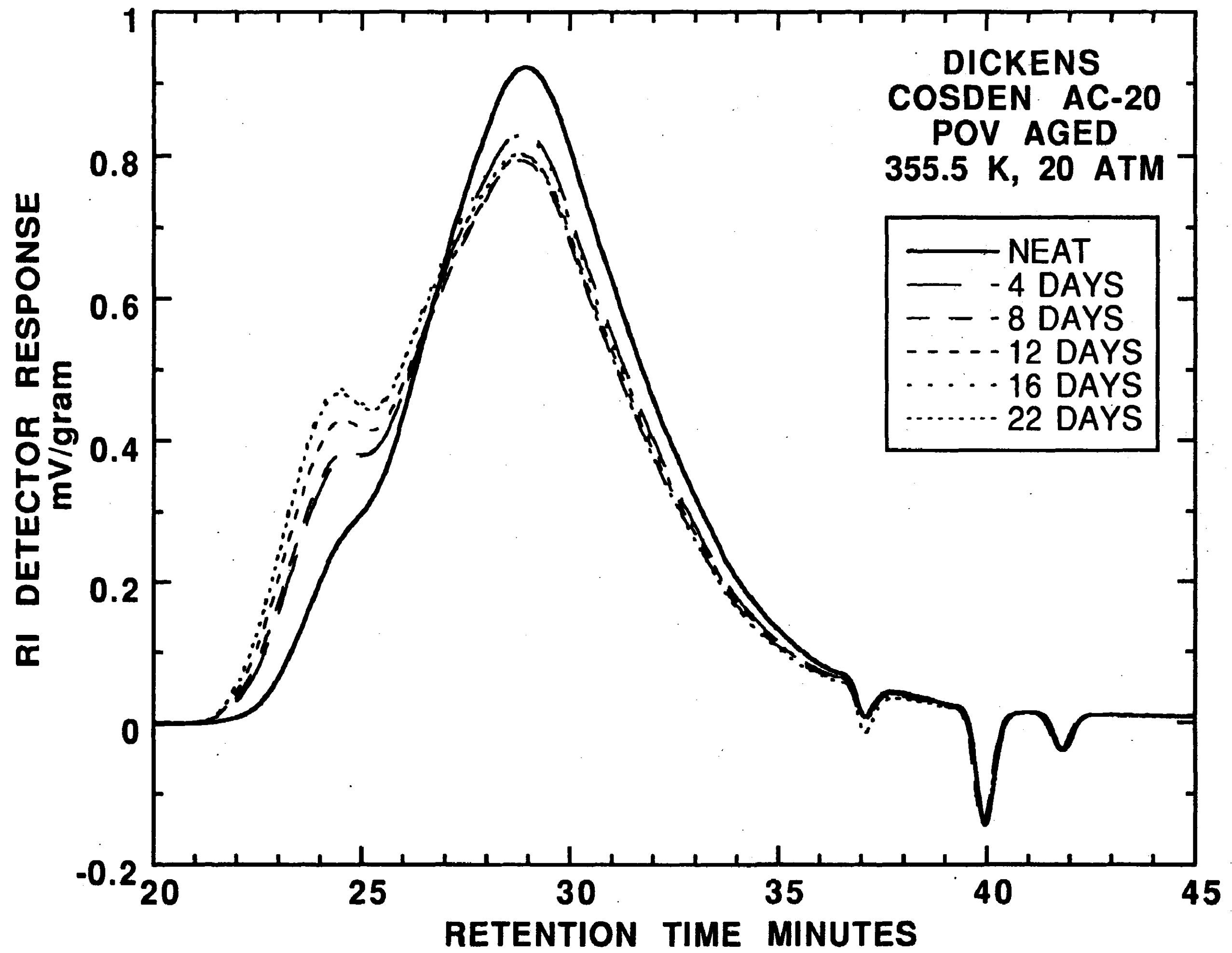

Figure E-16. GPCs of neat and POV-aged Dickens Cosden AC-20 at 355.5 K and 20 atm.

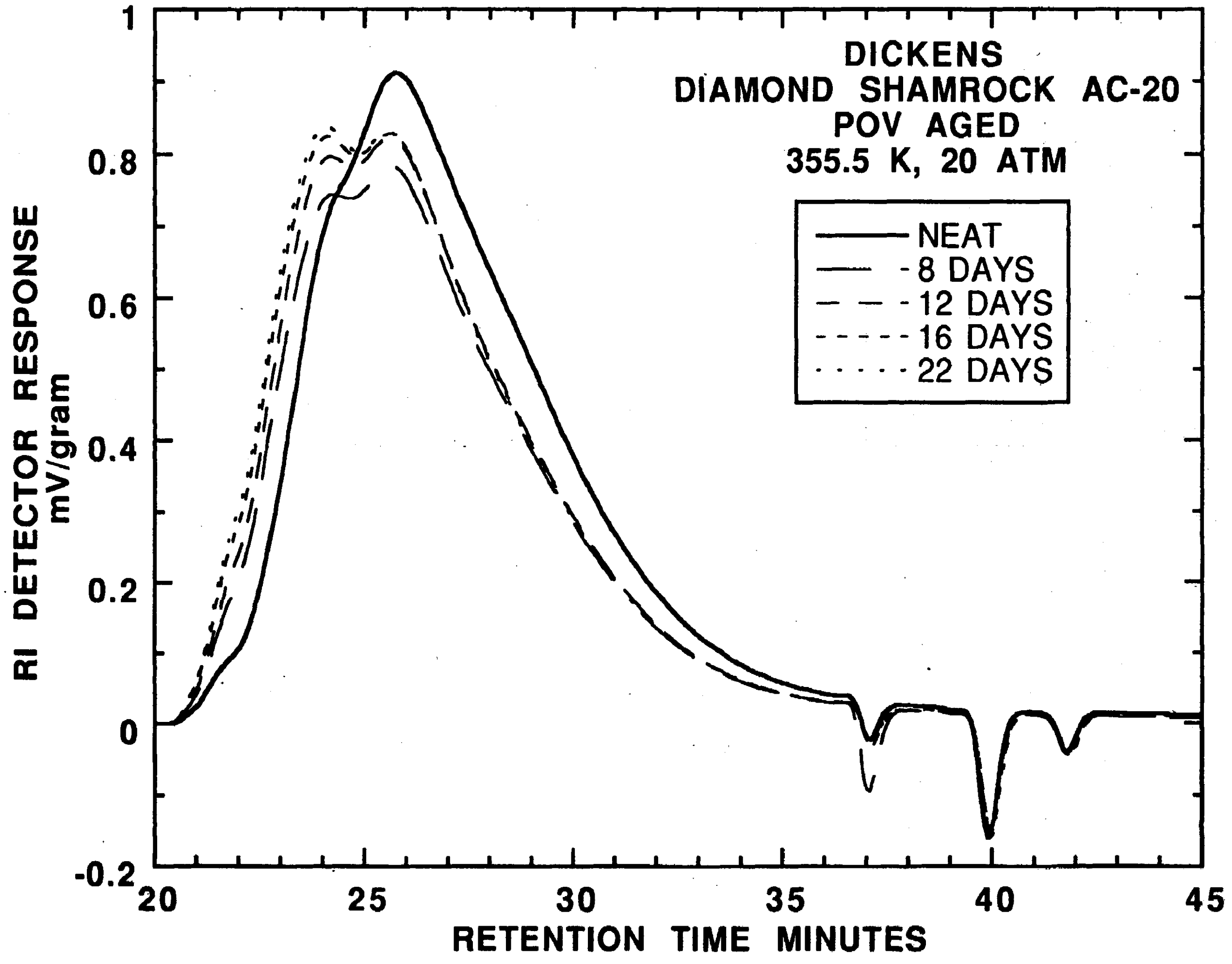

Figure E-17. GPCs of neat and POV-aged Dickens Diamond Shamrock AC-20 at 355.5 K and 20 atm.

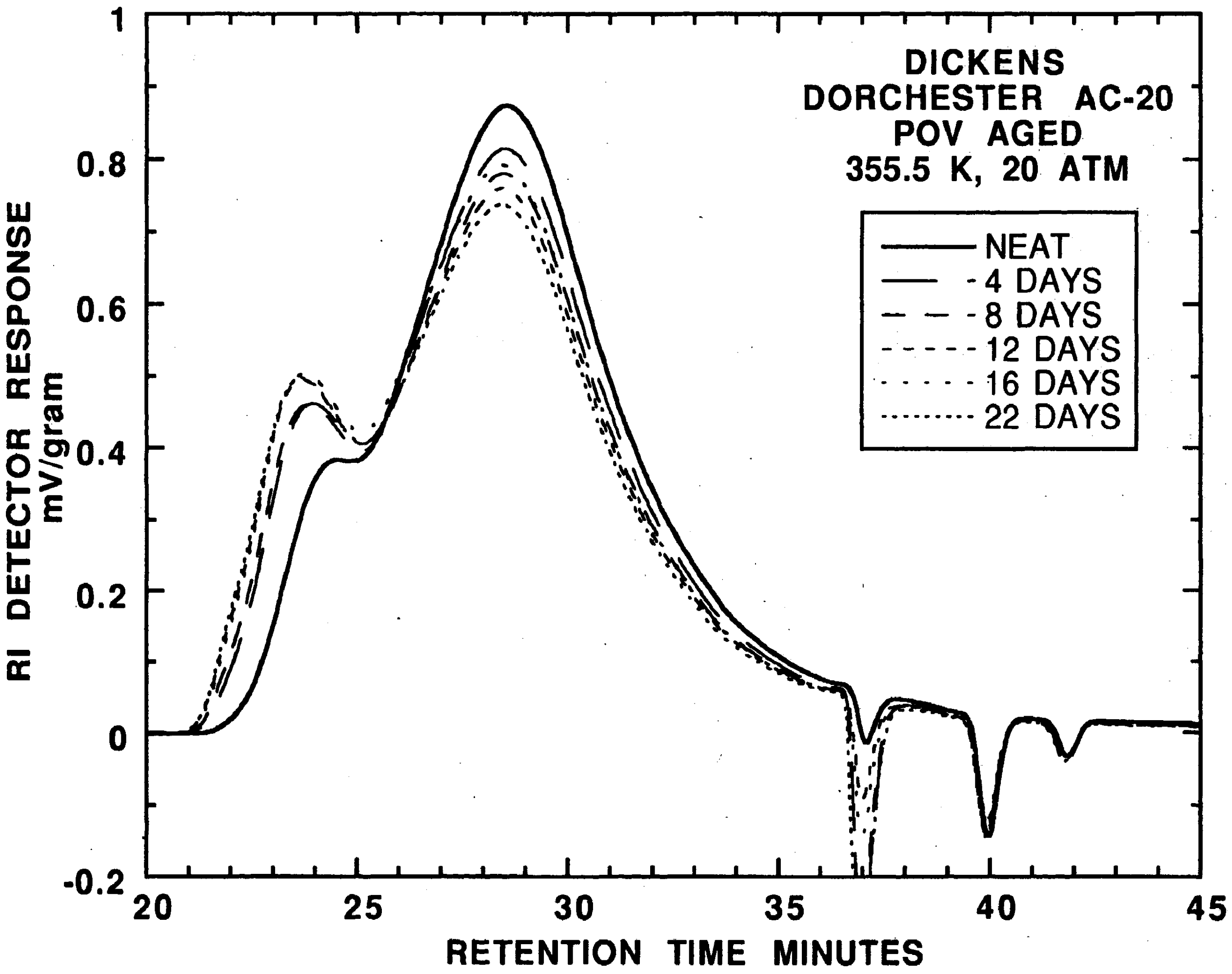

Figure E-18. GPCs of neat and POV-aged Dickens Dorchester AC-20 at 355.5 K and 20 atm.

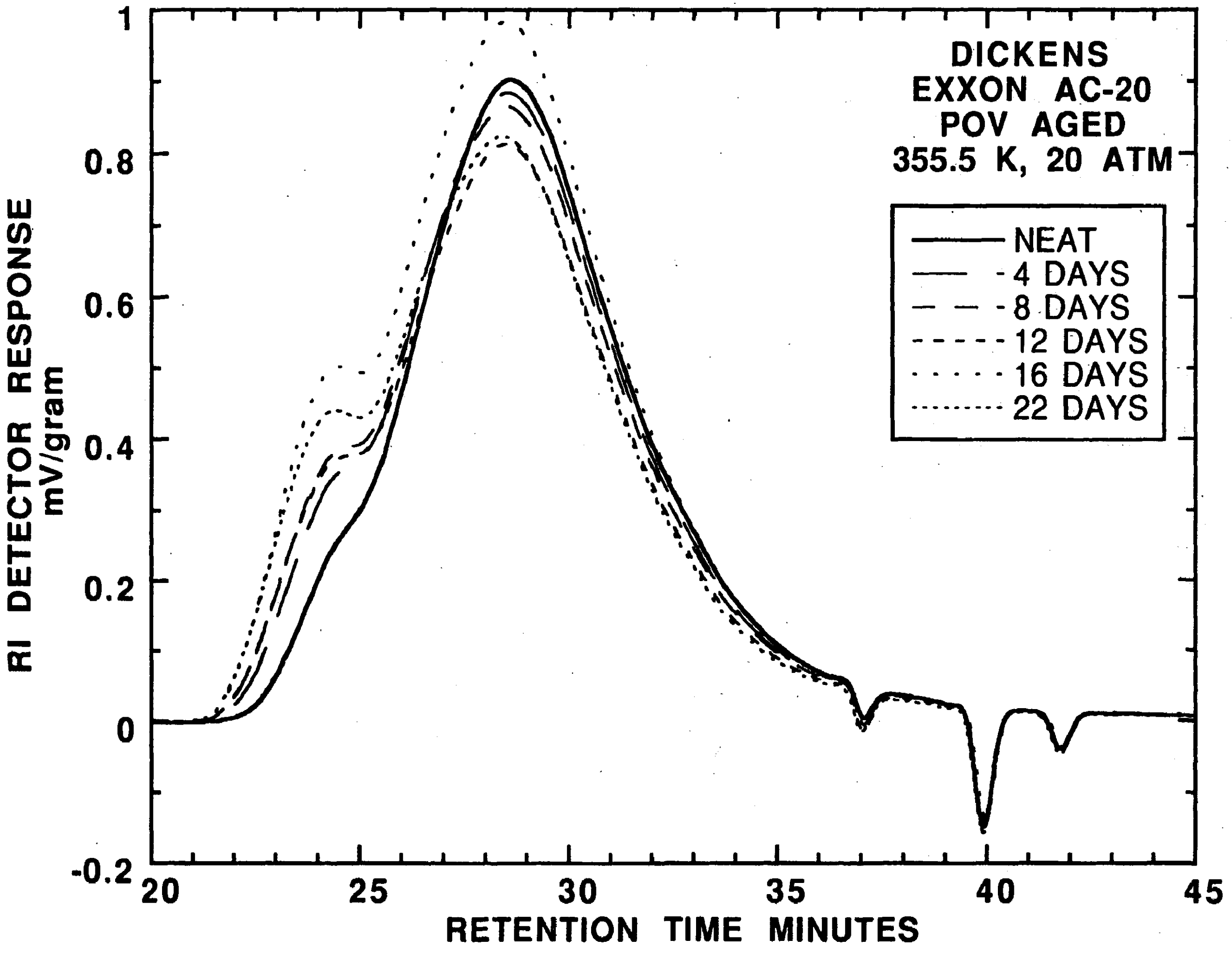

Figure E-19. GPCs of neat and POV-aged Dickens Exxon AC-20 at 355.5 K and 20 atm.

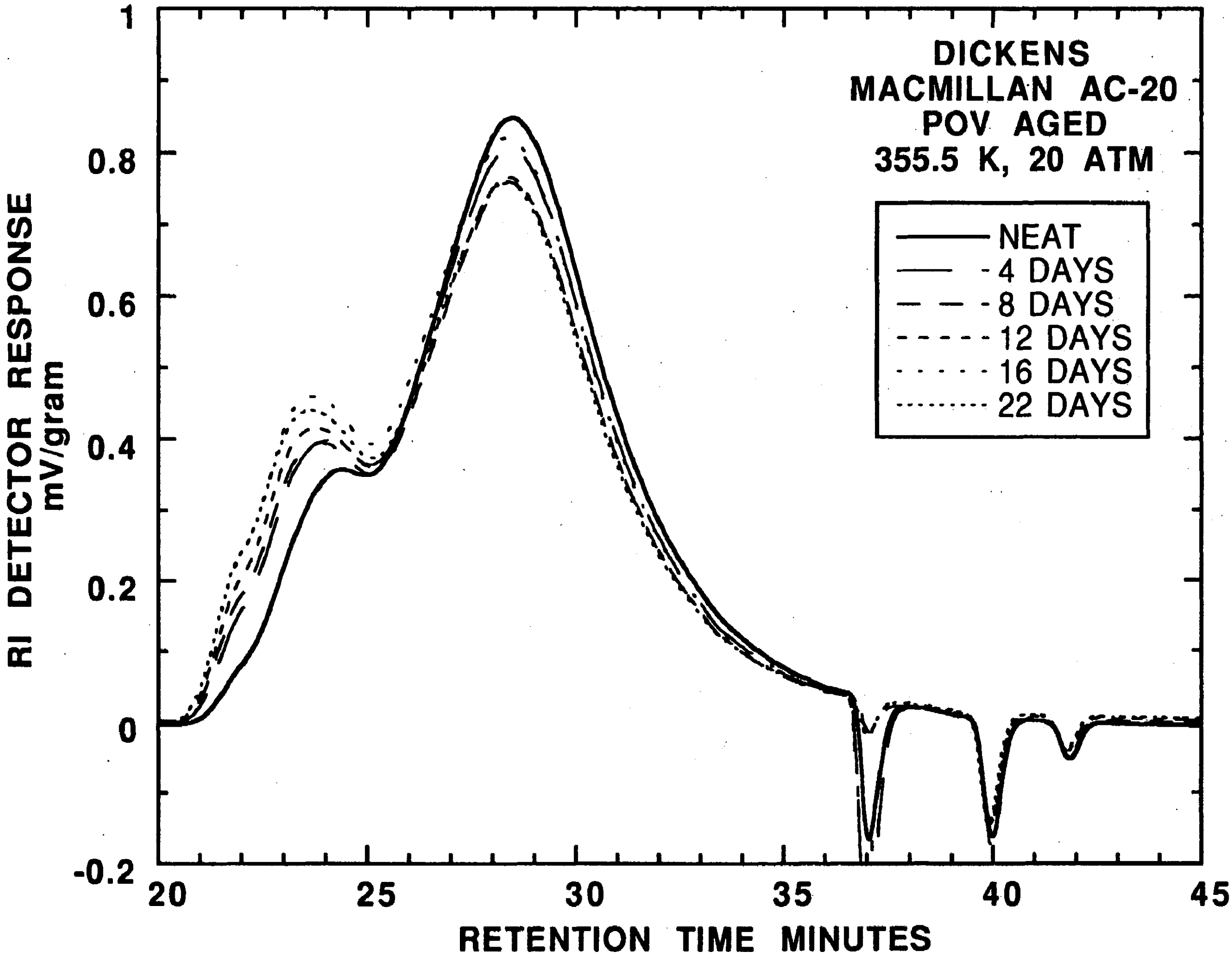

Figure E-20. GPCs of neat and POV-aged Dickens MacMillan AC-20 at 355.5 K and 20 atm.

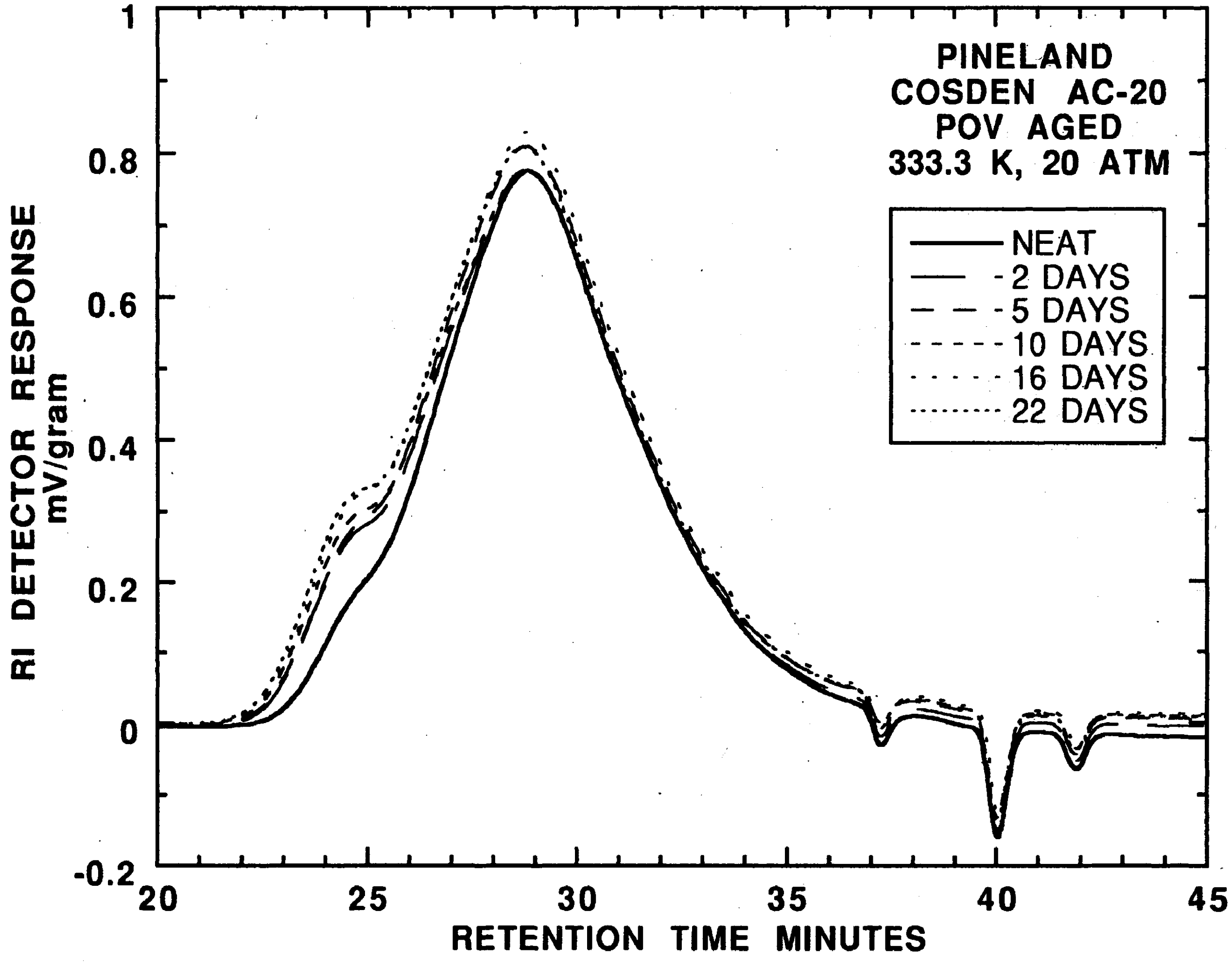

Figure E-21. GPCs of neat and POV-aged Pineland Cosden AC-20 at 333.3 K and 20 atm.

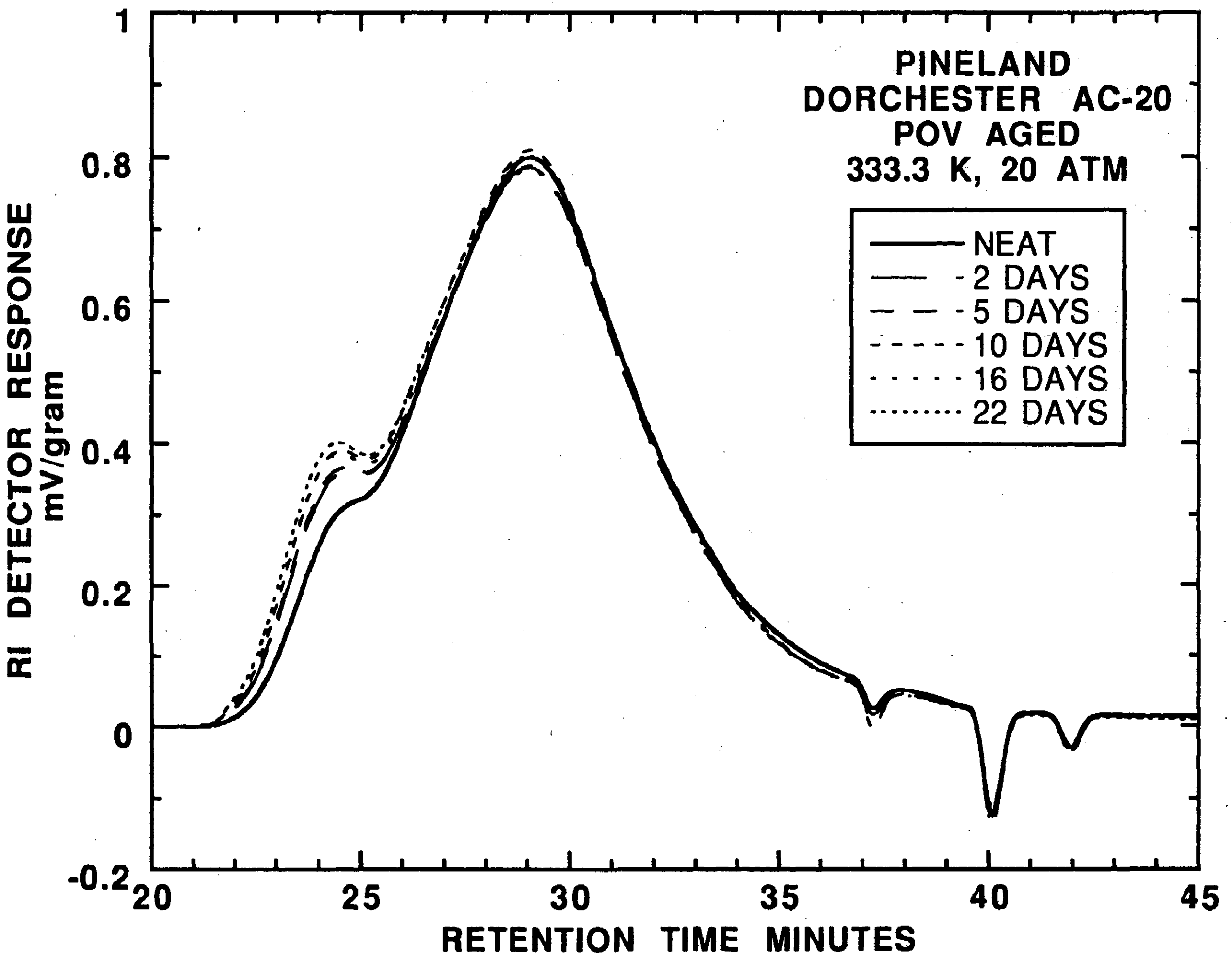

Figure E-22. GPCs of neat and POV-aged Pineland Dorchester AC-20 at 333.3 K and 20 atm.

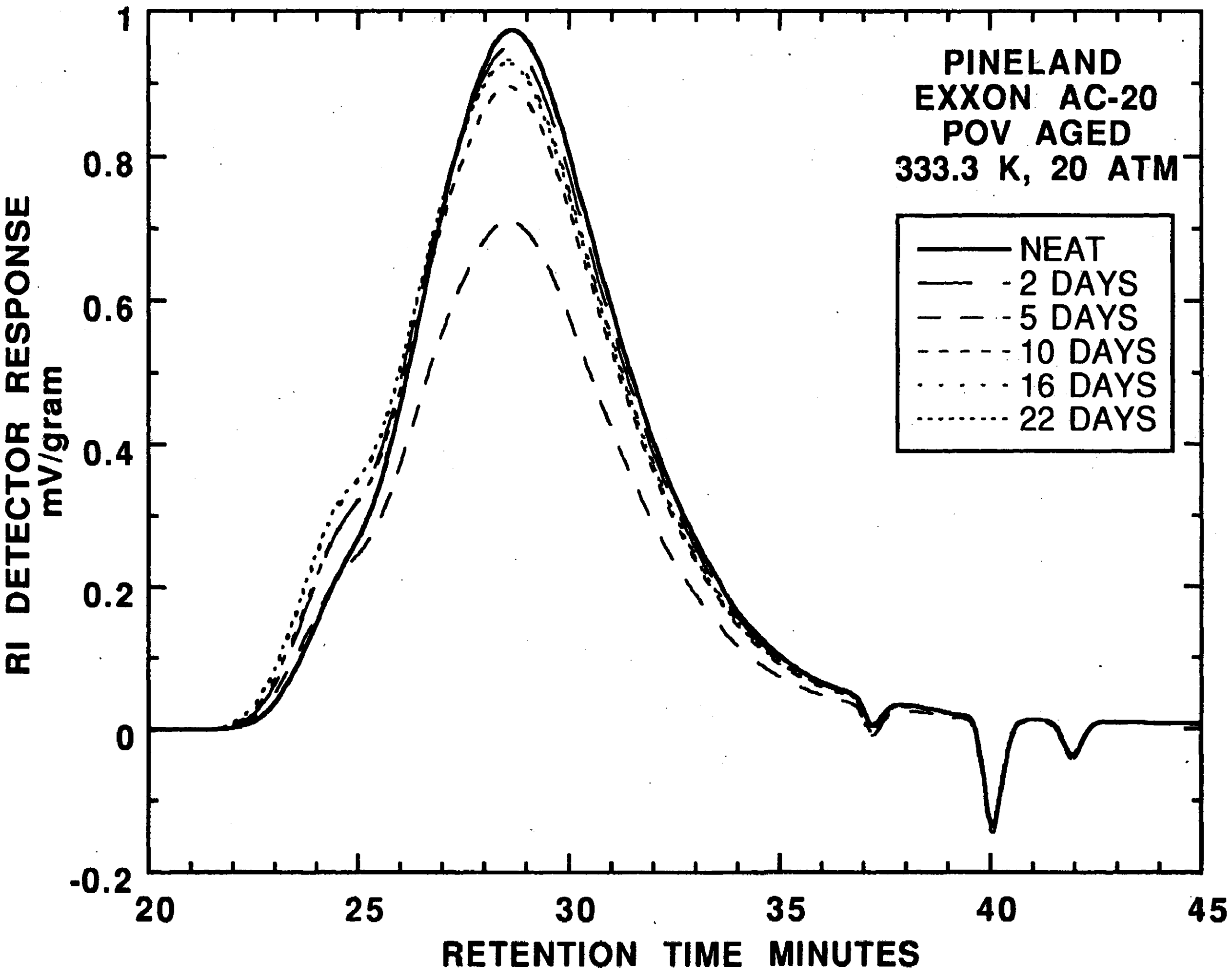

Figure E-23. GPCs of neat and POV-aged Pineland Exxon AC-20 at 333.3 K and 20 atm.

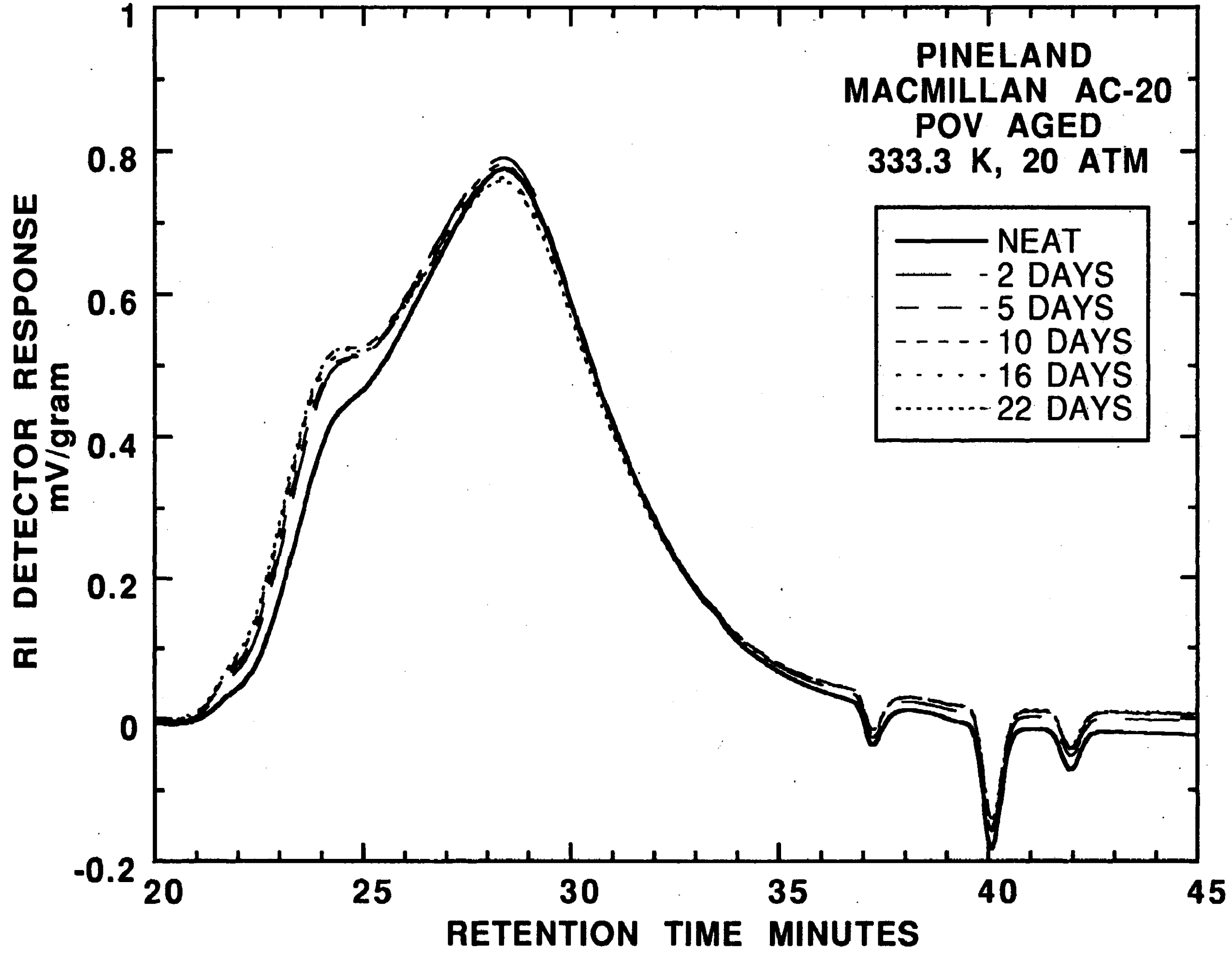

Figure E-24. GPCs of neat and POV-aged Pineland MacMillan AC-20 at 333.3 K and 20 atm.

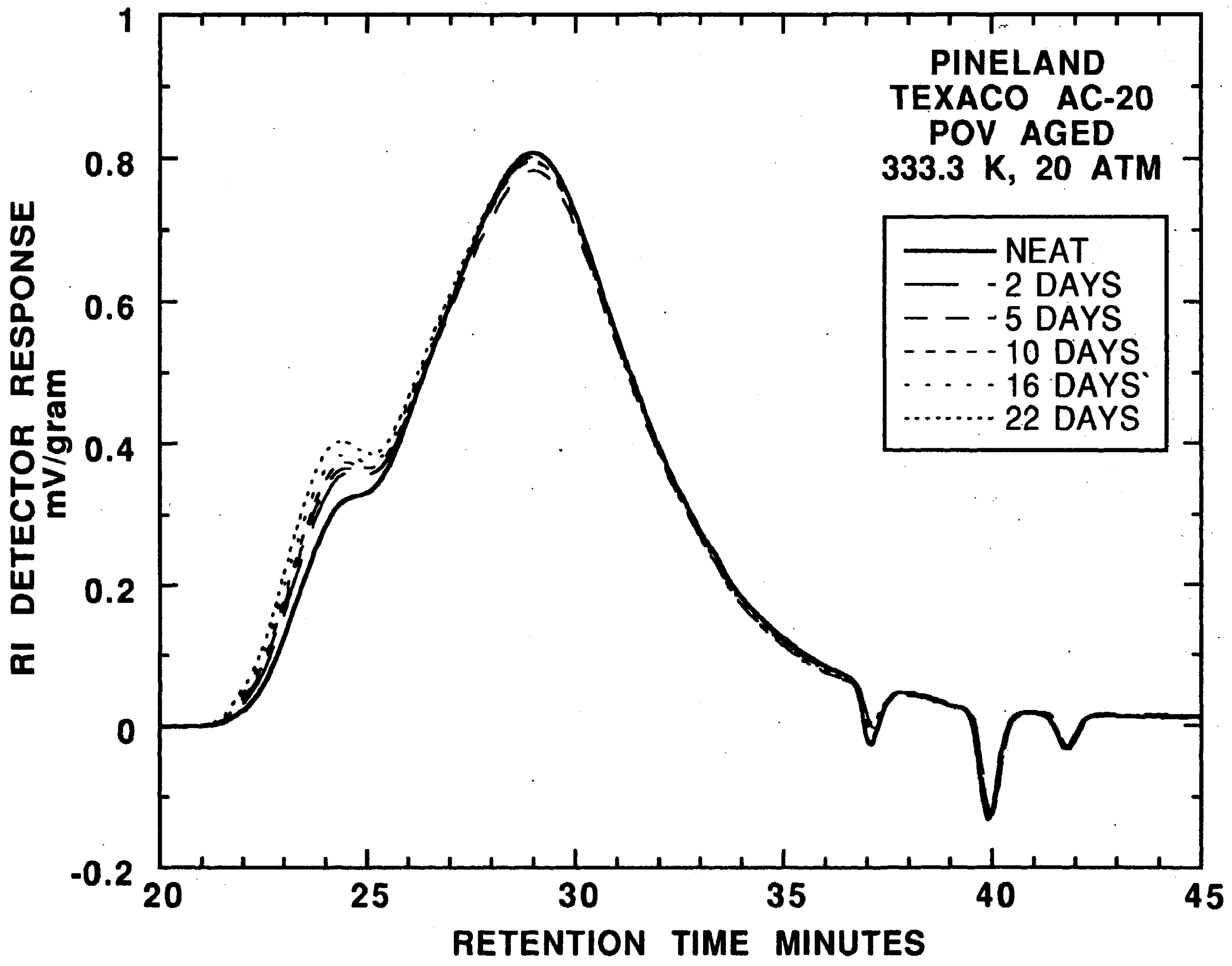

Figure E-25. GPCs of neat and POV-aged Pineland Texaco AC-20 at 333.3 K and 20 atm.

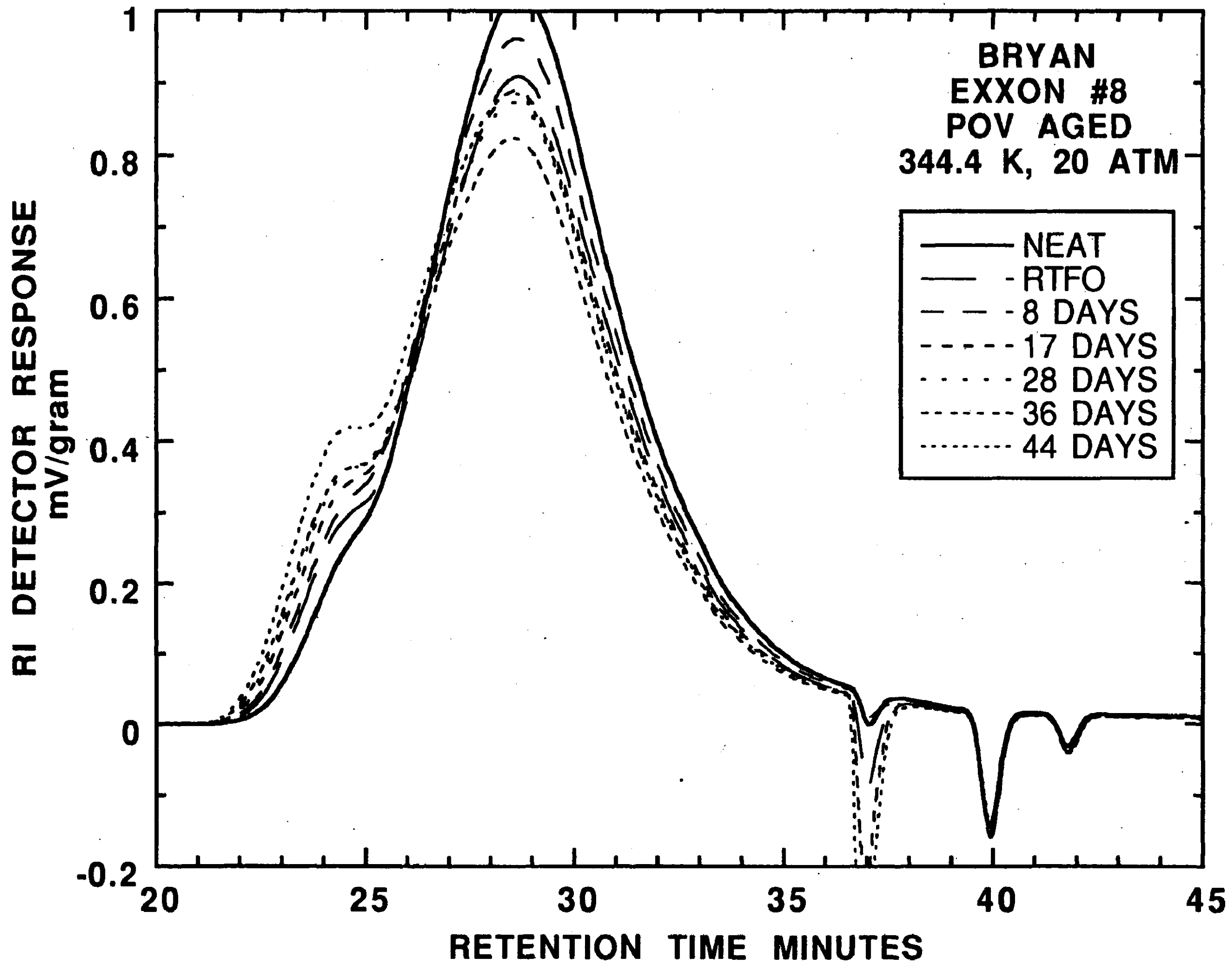

Figure E-26. GPCs of neat and POV-aged Bryan Exxon #8 AC-20 at 344.4 K and 20 atm.

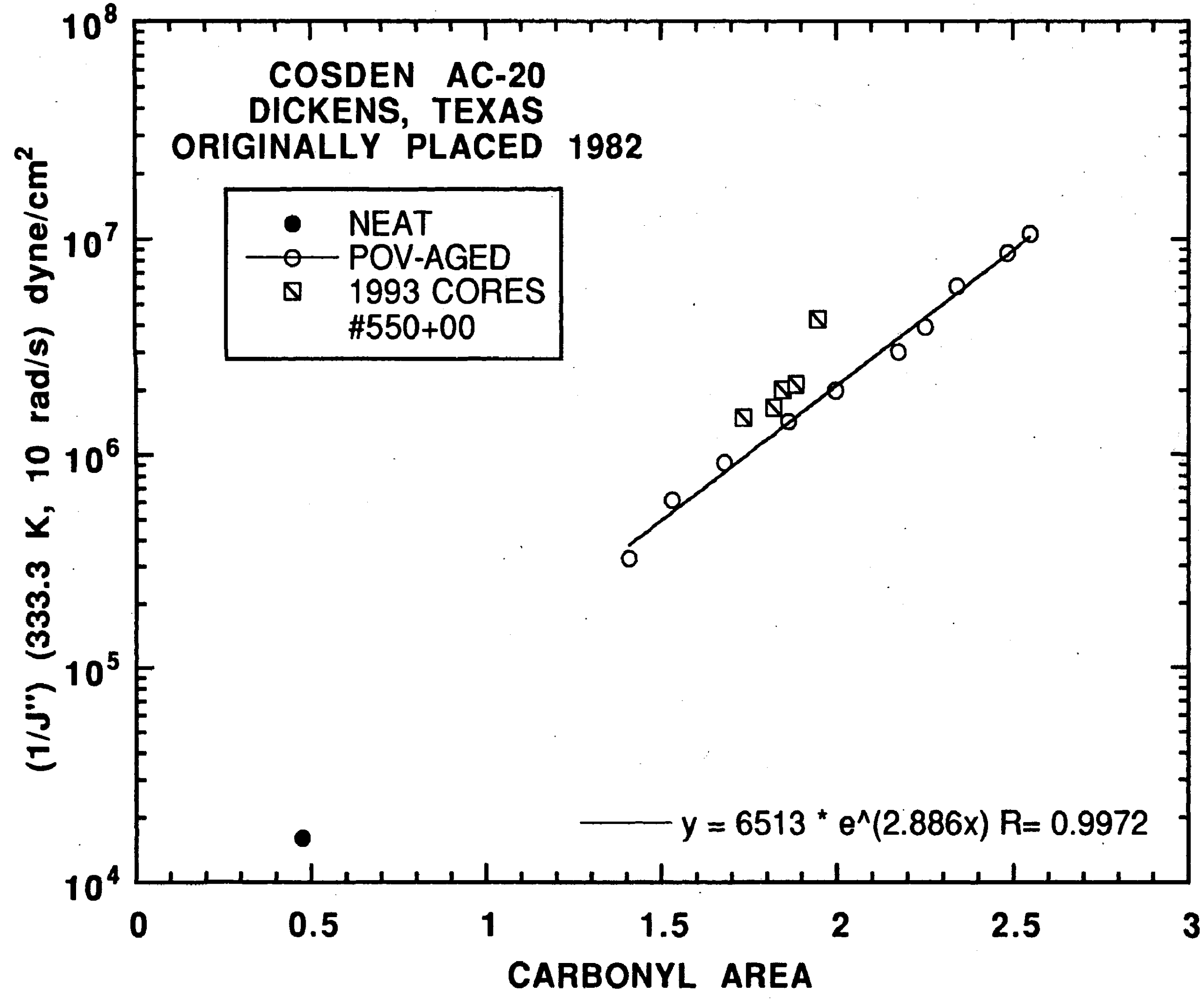

Figure E-27. Comparisons between ($1 / J''$) at 333.3 K, 10 rad/s and *CA* of neat, POV- (355.5 K, 20 atm), and field-aged (#550+00, February 1993) Dickens Cosden AC-20.

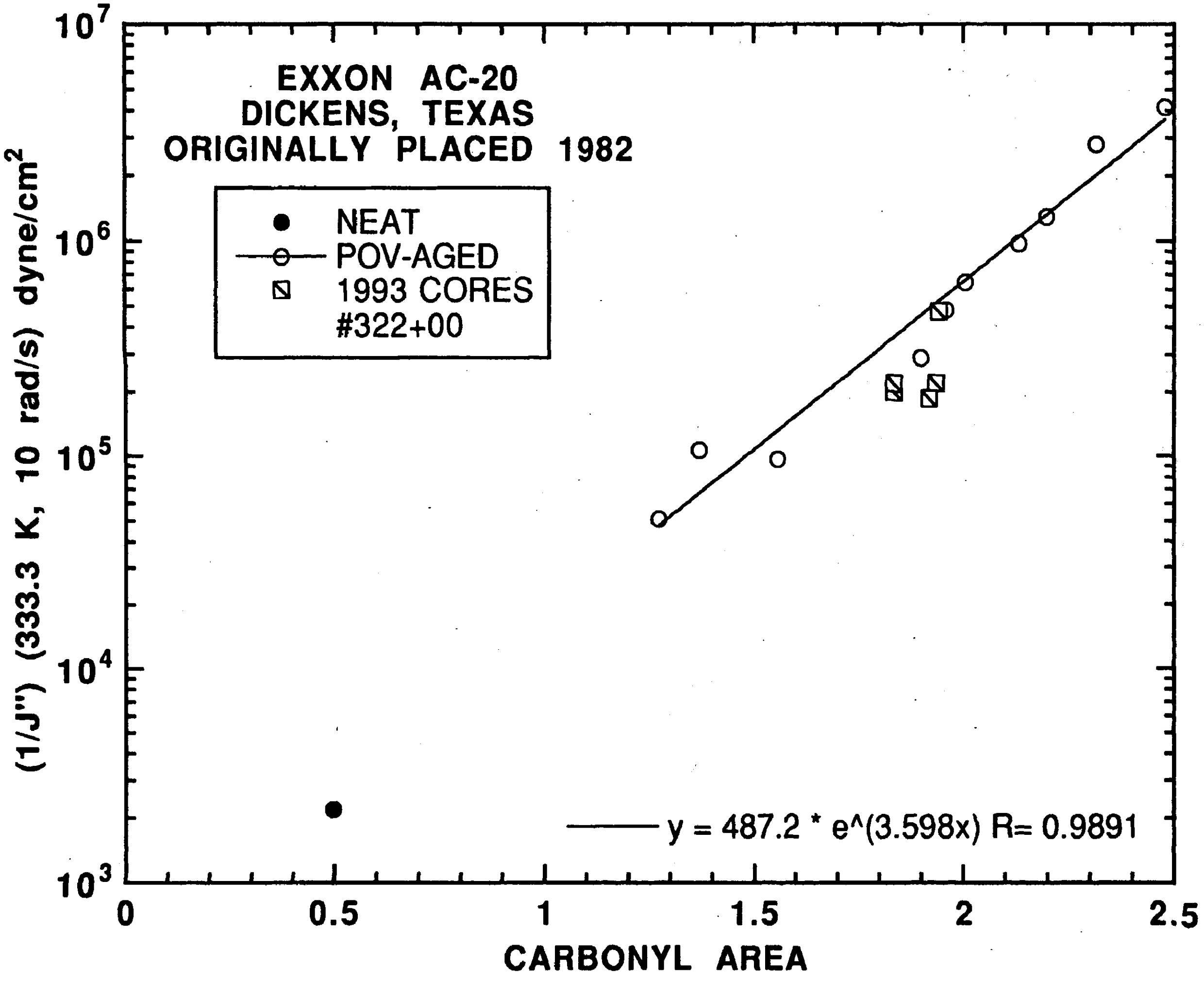

Figure E-28. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (355.5 K, 20 atm), and field-aged (#322+00, February 1993) Dickens Exxon AC-20.

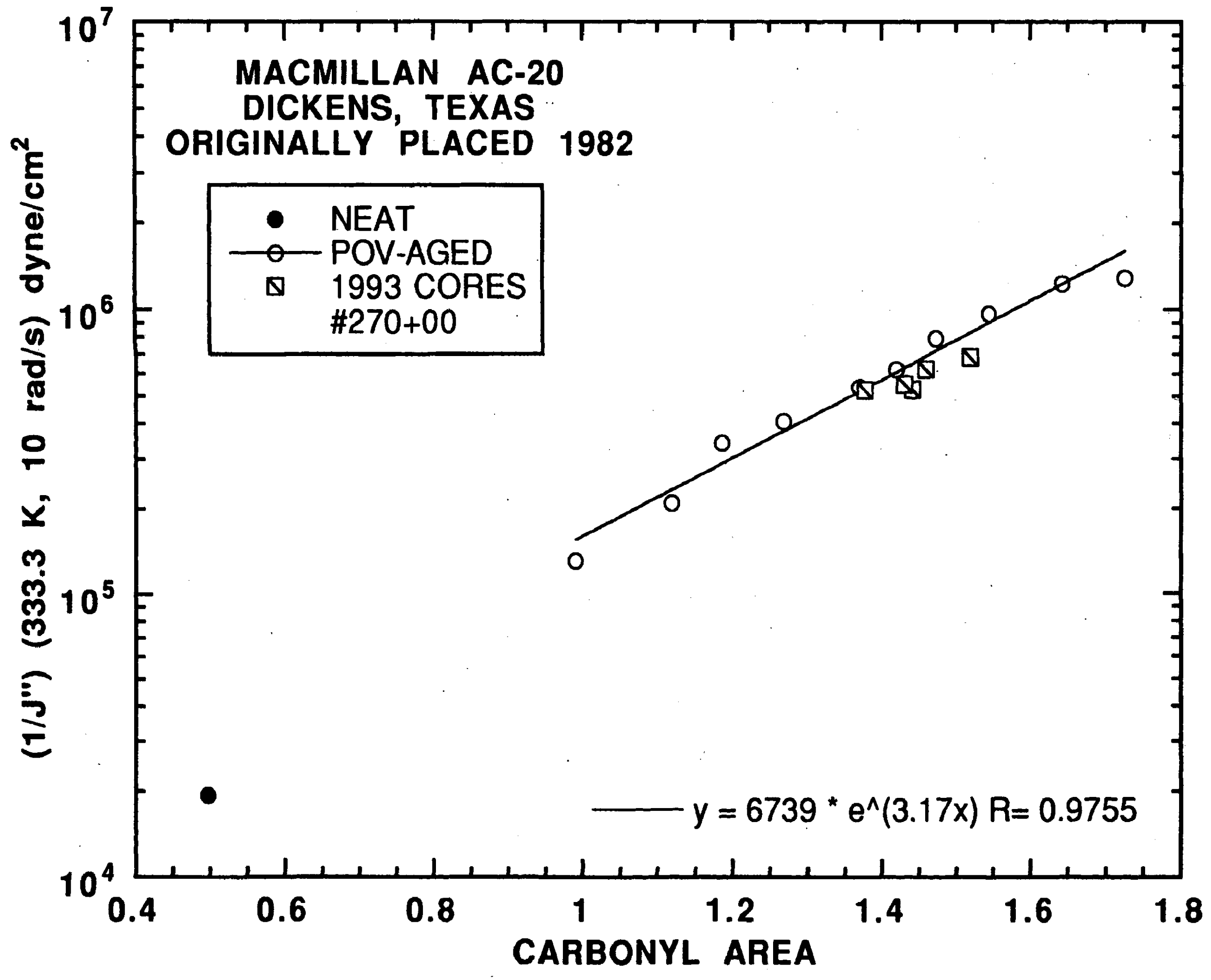

Figure E-29. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (355.5 K, 20 atm), and field-aged (#270+00, February 1993) Dickens MacMillan AC-20.

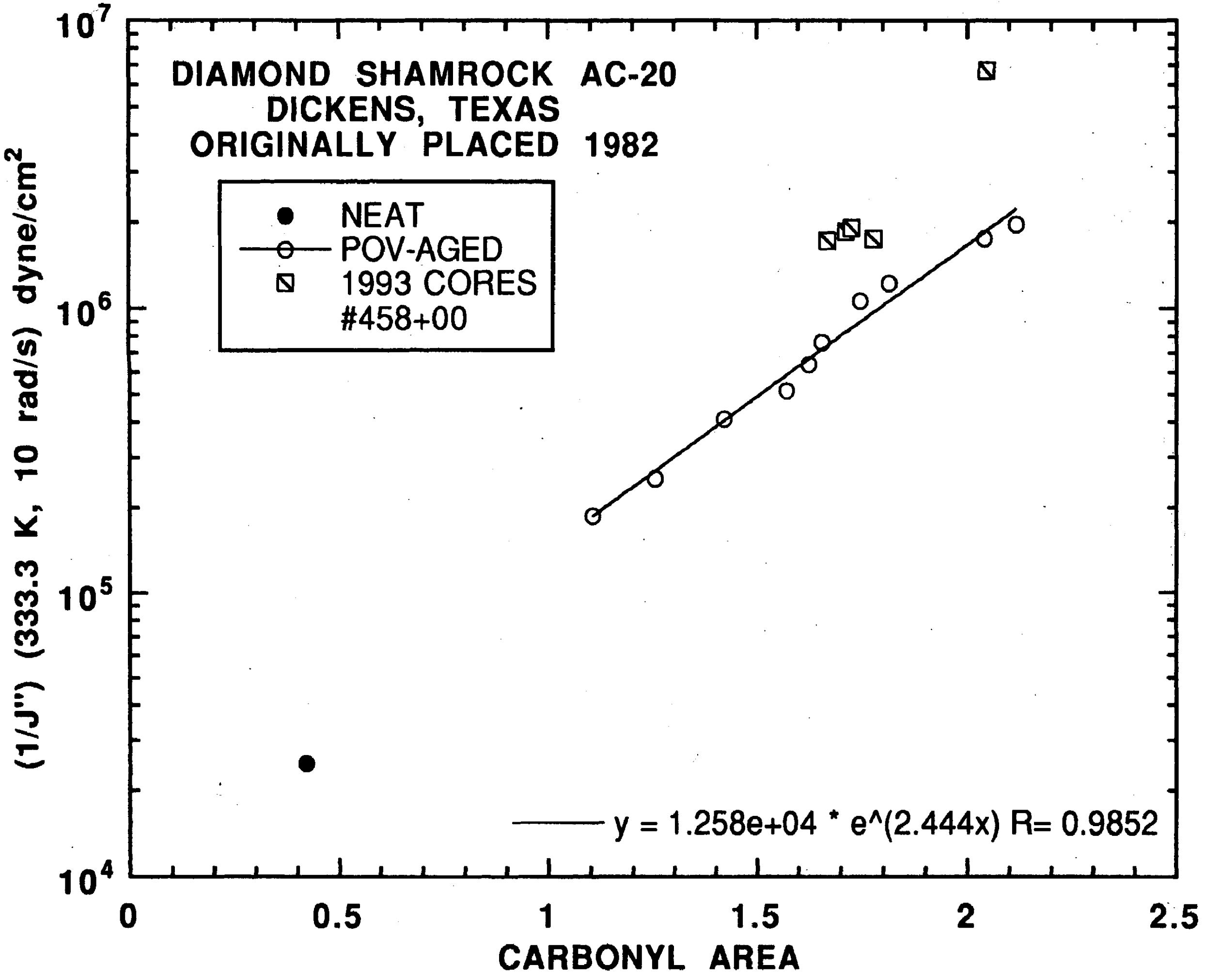

Figure E-30. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (355.5 K, 20 atm), and field-aged (#458+00, February 1993) Dickens Diamond Shamrock AC-20.

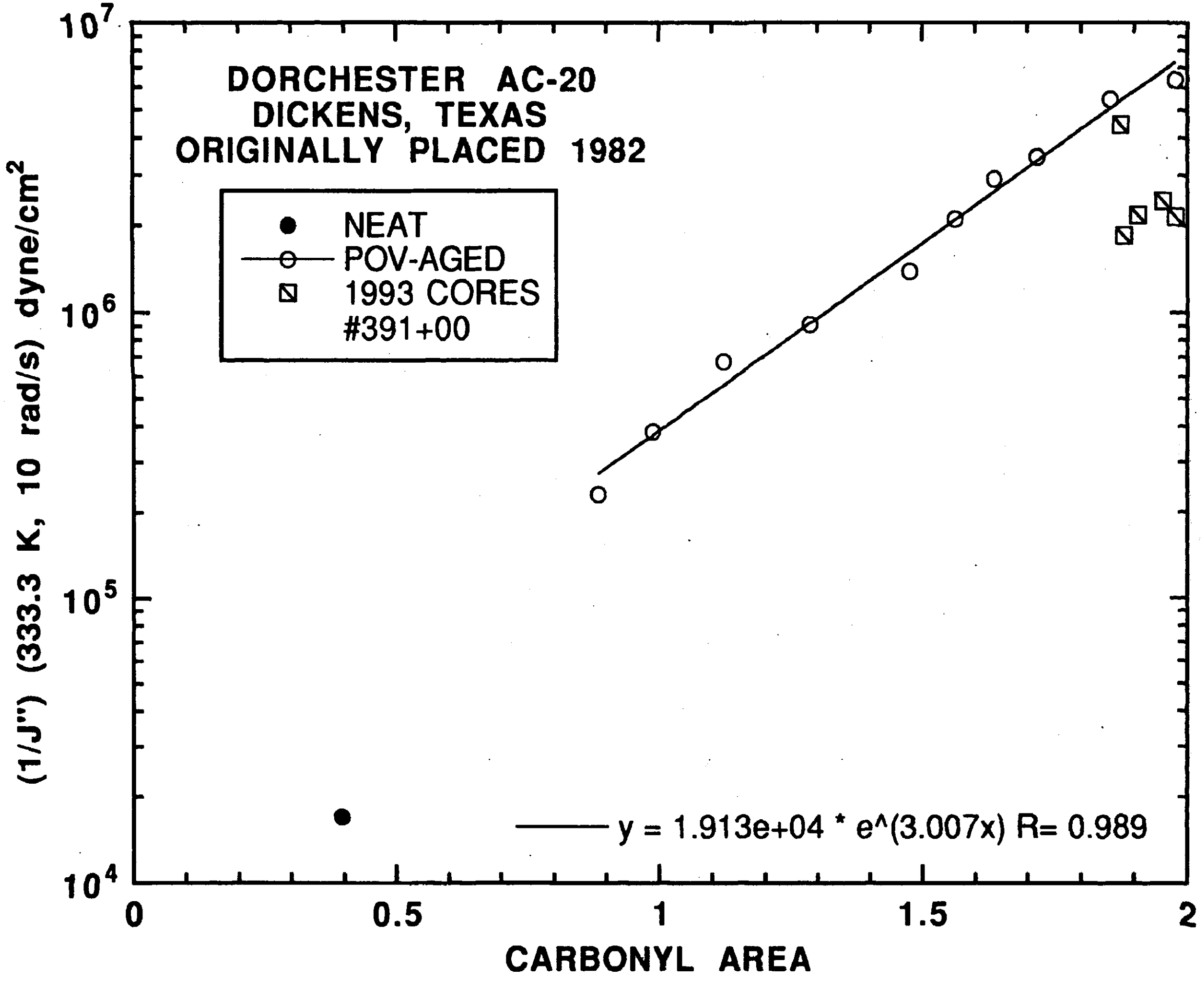

Figure E-31. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (355.5 K, 20 atm), and field-aged (#391+00, February 1993) Dickens Dorchester AC-20.

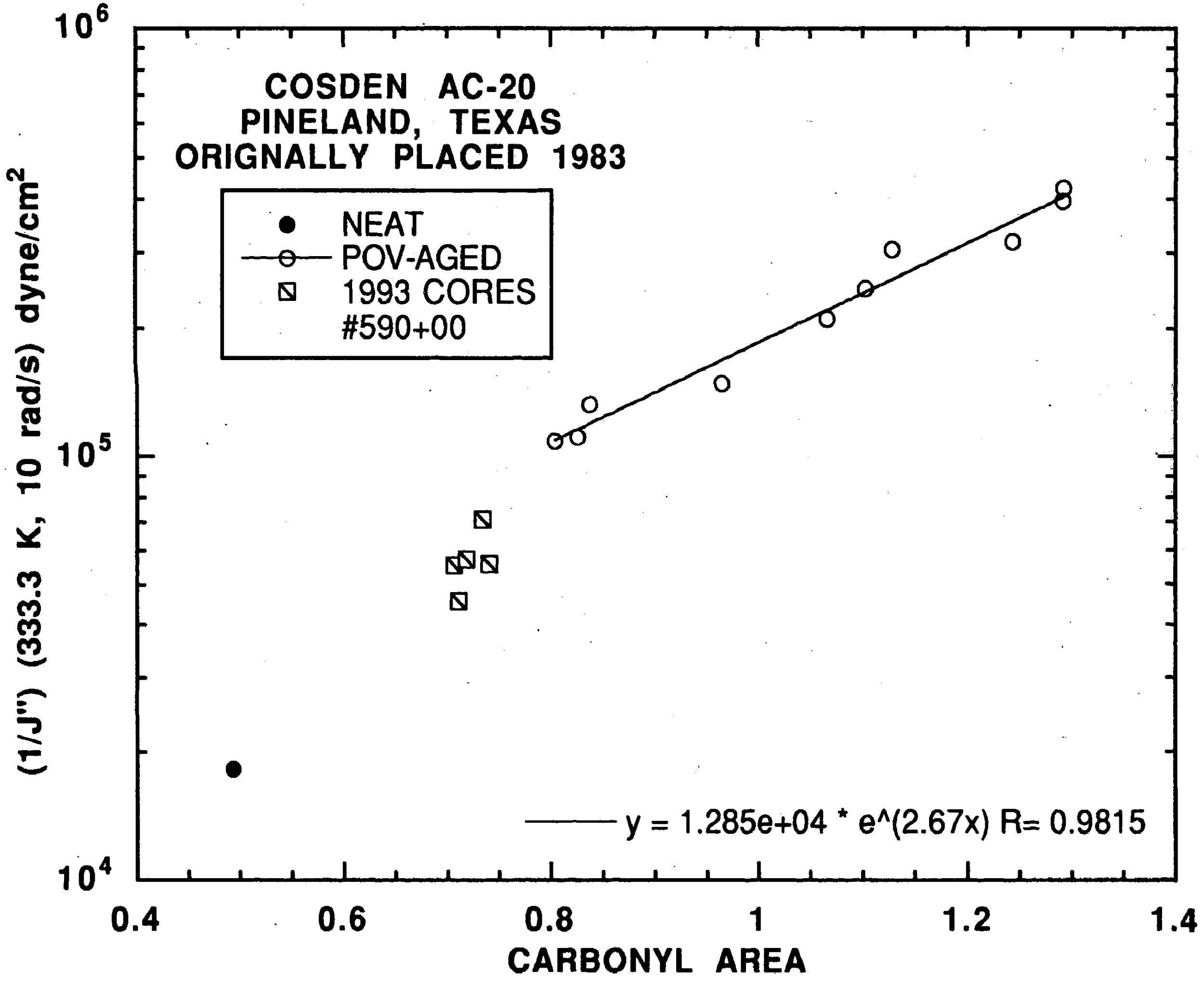

Figure E-32. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (333.3 K, 20 atm), and field-aged (#590+00, March 1993) Pineland Cosden AC-20.

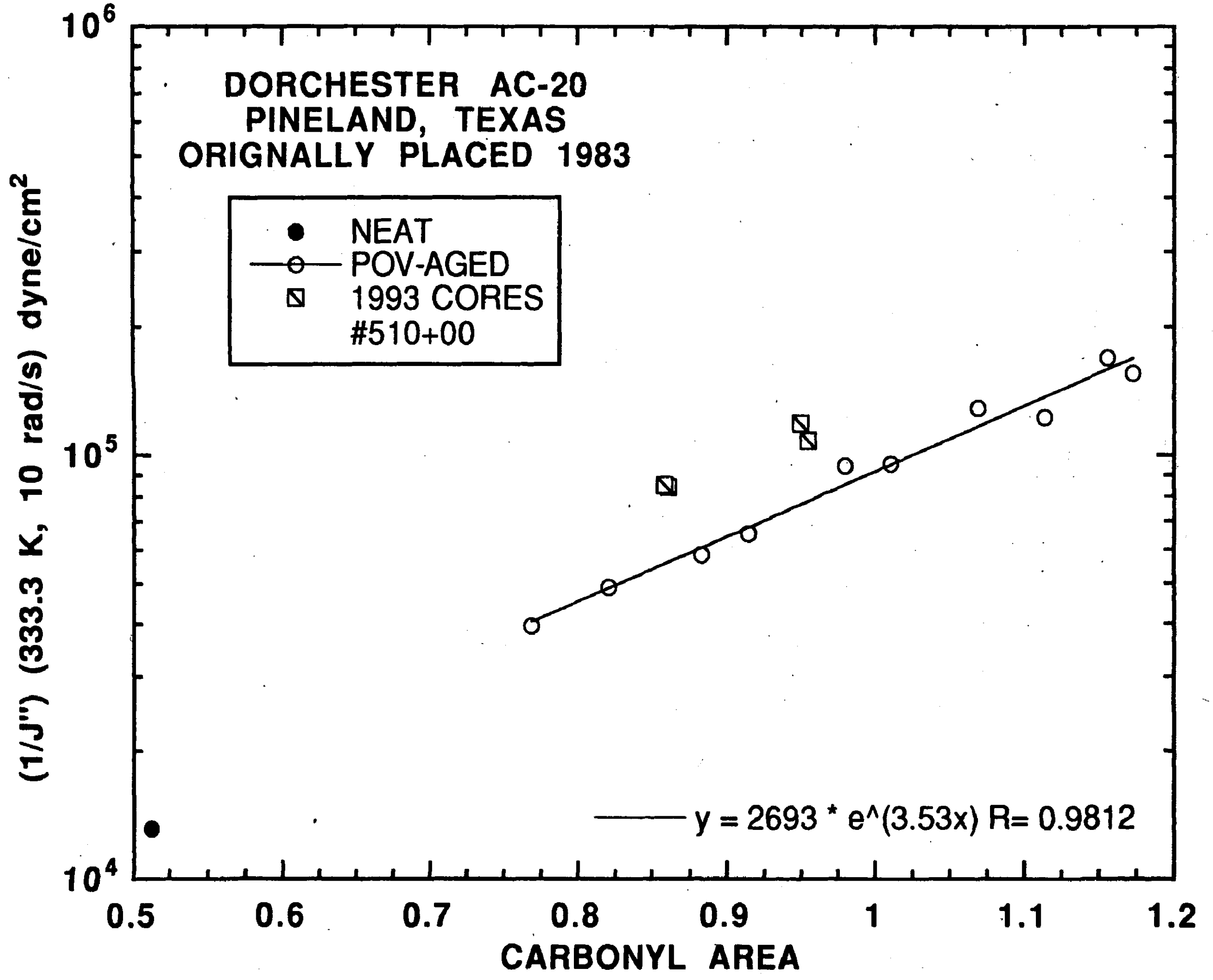

Figure E-33. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (333.3 K, 20 atm), and field-aged (#510+00, March 1993) Pineland Dorchester AC-20.

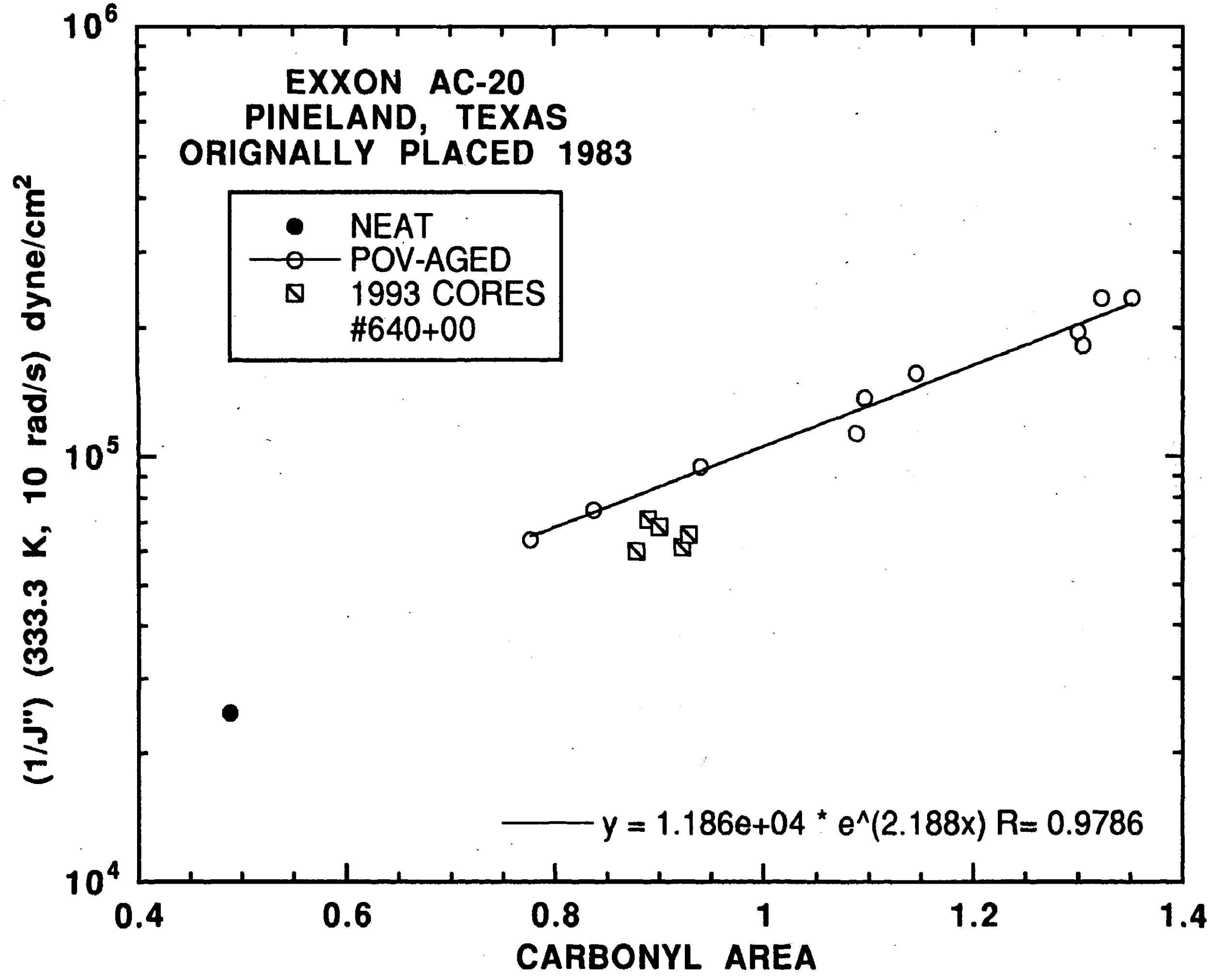

Figure E-34. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (333.3 K, 20 atm), and field-aged (#640+00, March 1993) Pineland Exxon AC-20.

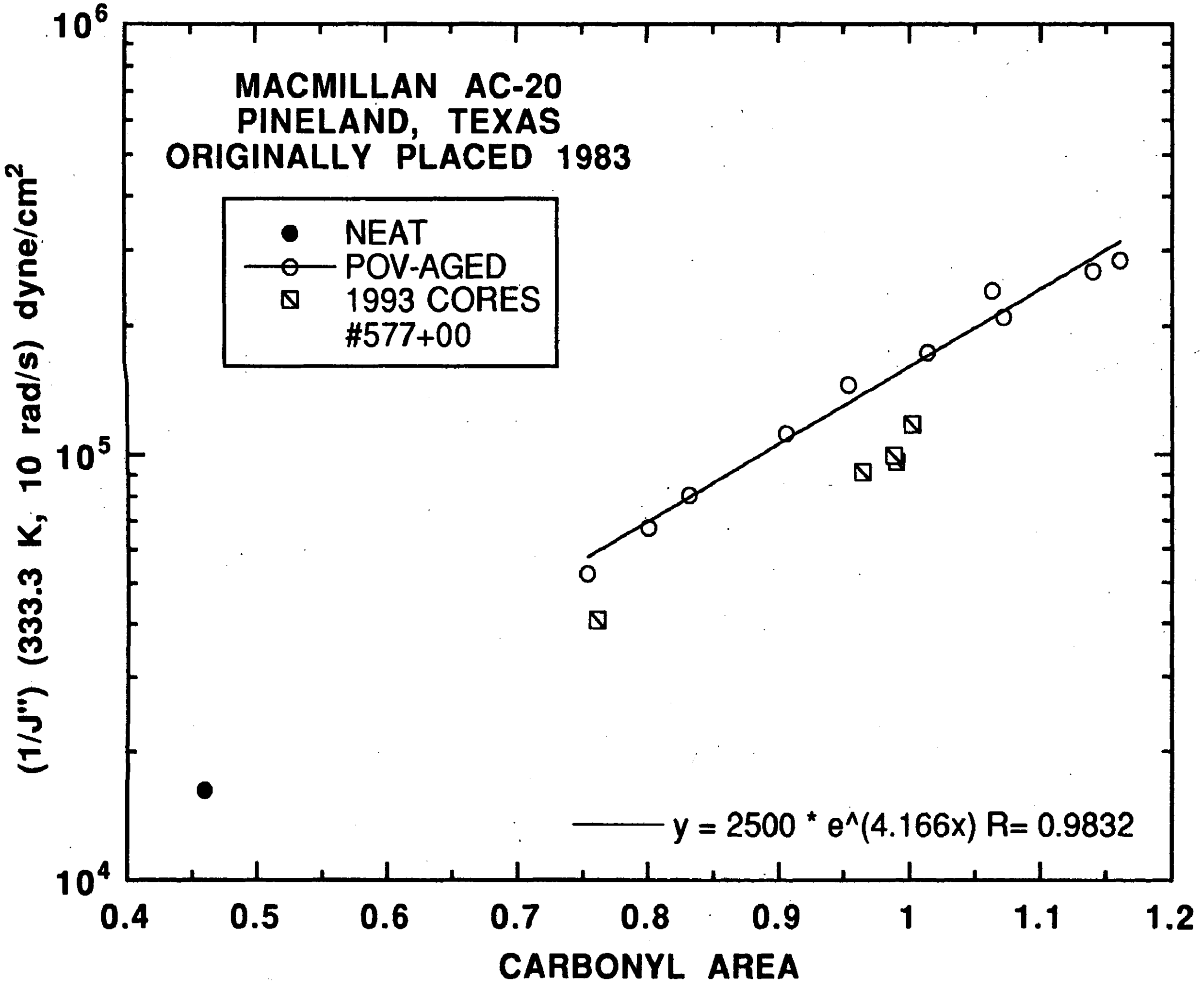

Figure E-35. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (333.3 K, 20 atm), and field-aged (#557+00, March 1993) Pineland MacMillan AC-20.

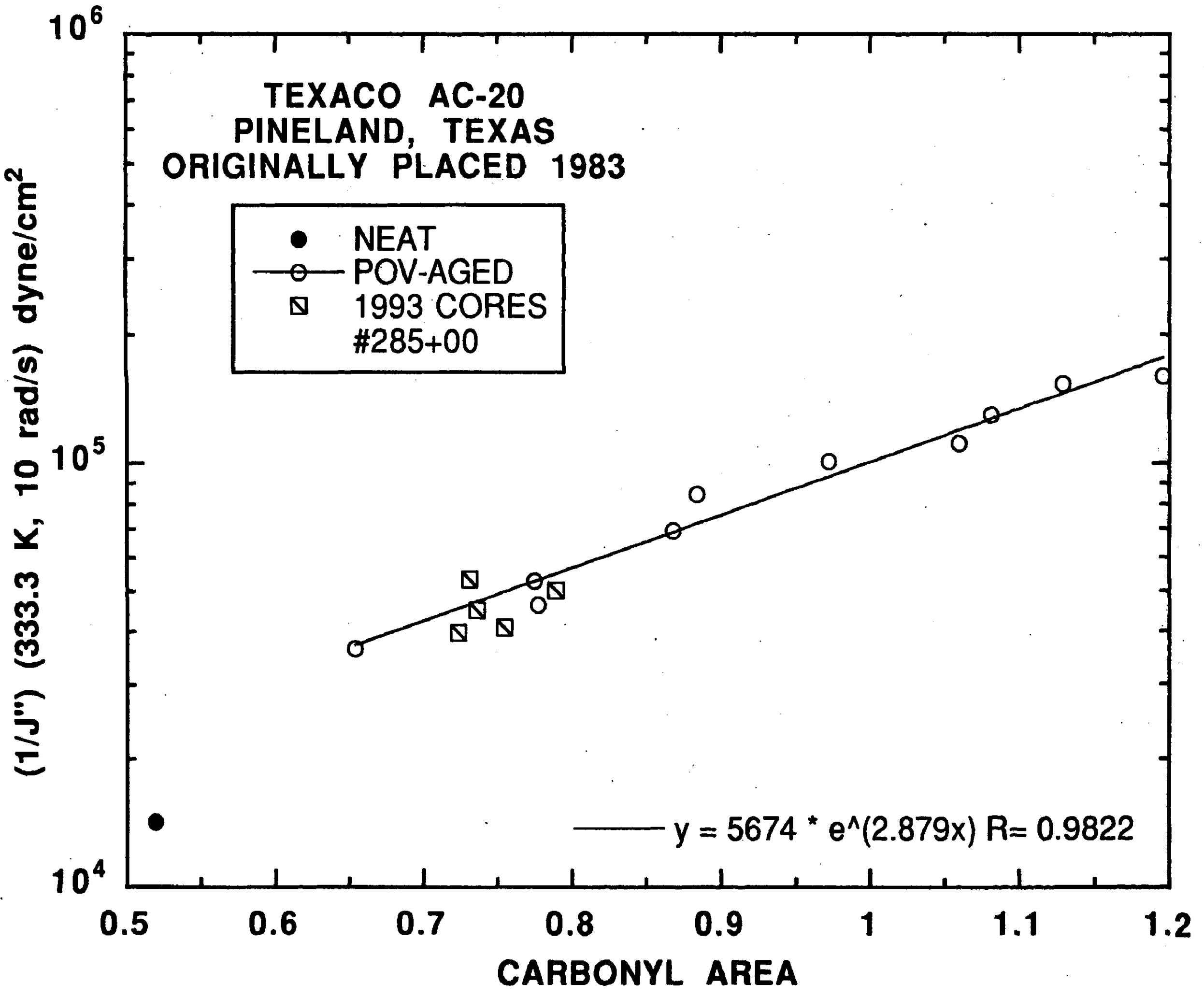

Figure E-36. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, POV- (333.3 K, 20 atm), and field-aged (#285+00, March 1993) Pineland Texaco AC-20.

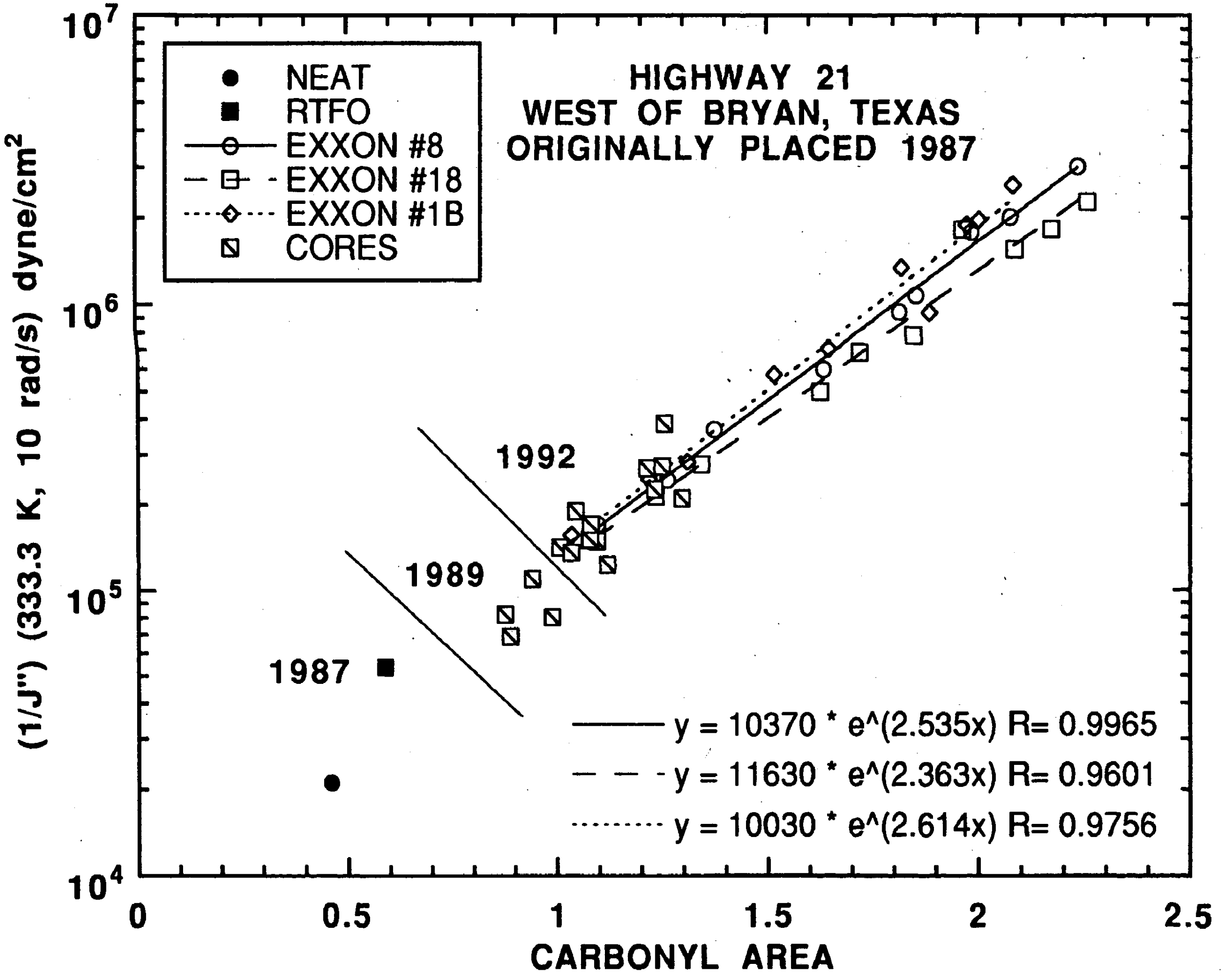

Figure E-37. Comparisons between $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of neat, RTFO, POV- (344.4 K, 20 atm) and field-aged Bryan Exxon AC-20.

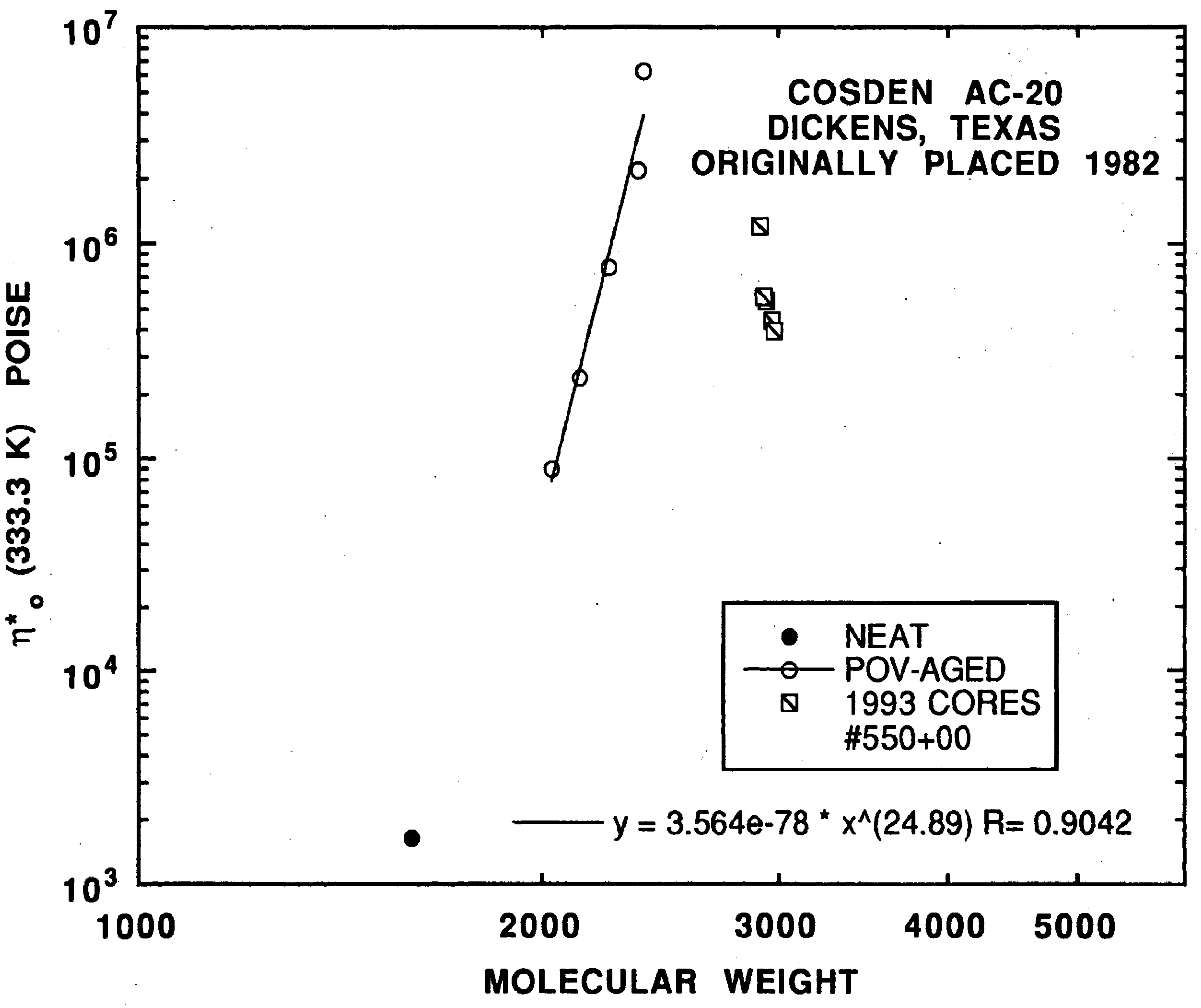

Figure E-38. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (355.5 K, 20 atm), and field-aged (#550+00, February 1993) Dickens Cosden AC-20.

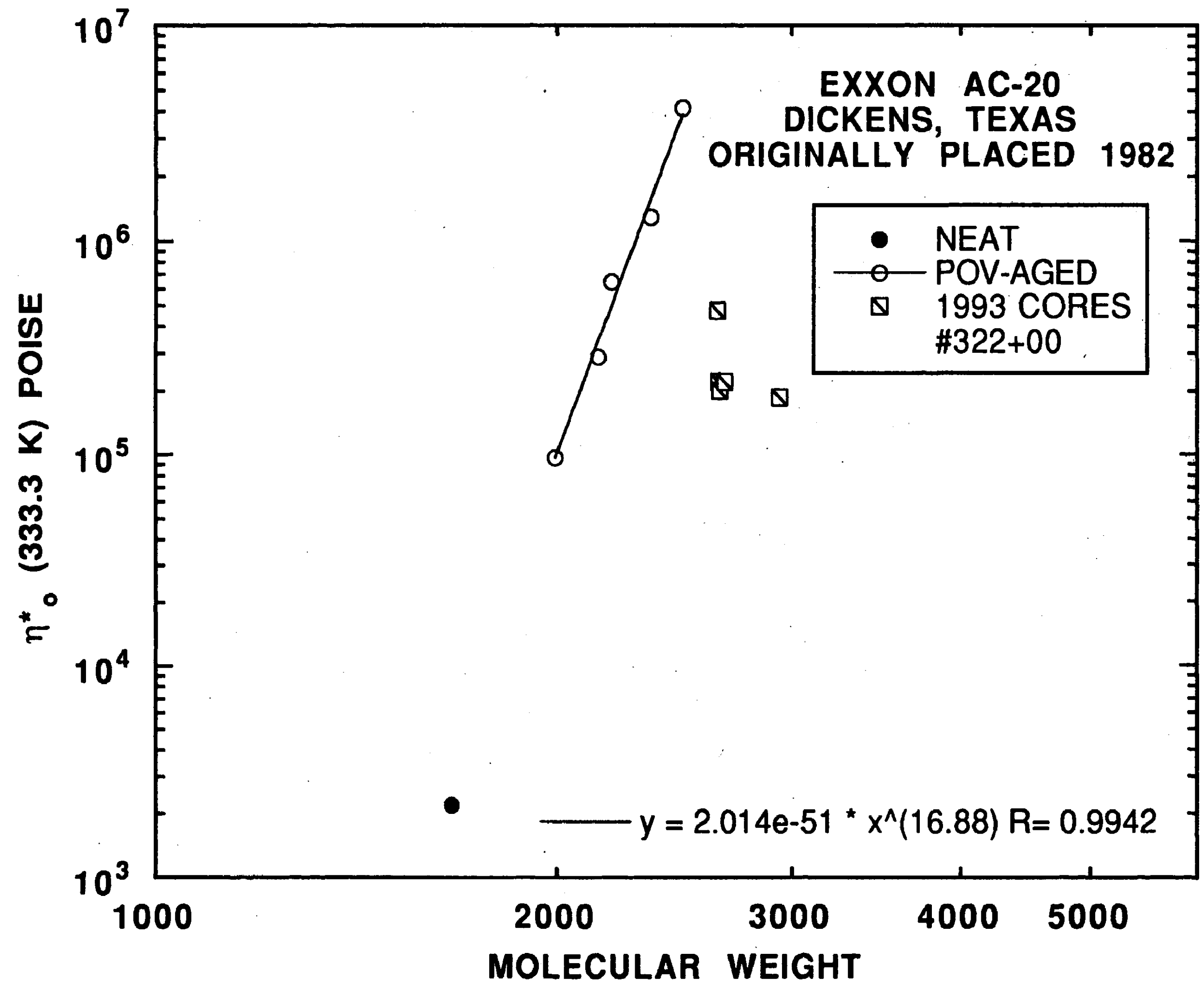

Figure E-39. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (355.5 K, 20 atm), and field-aged (#322+00, February 1993) Dickens Exxon AC-20.

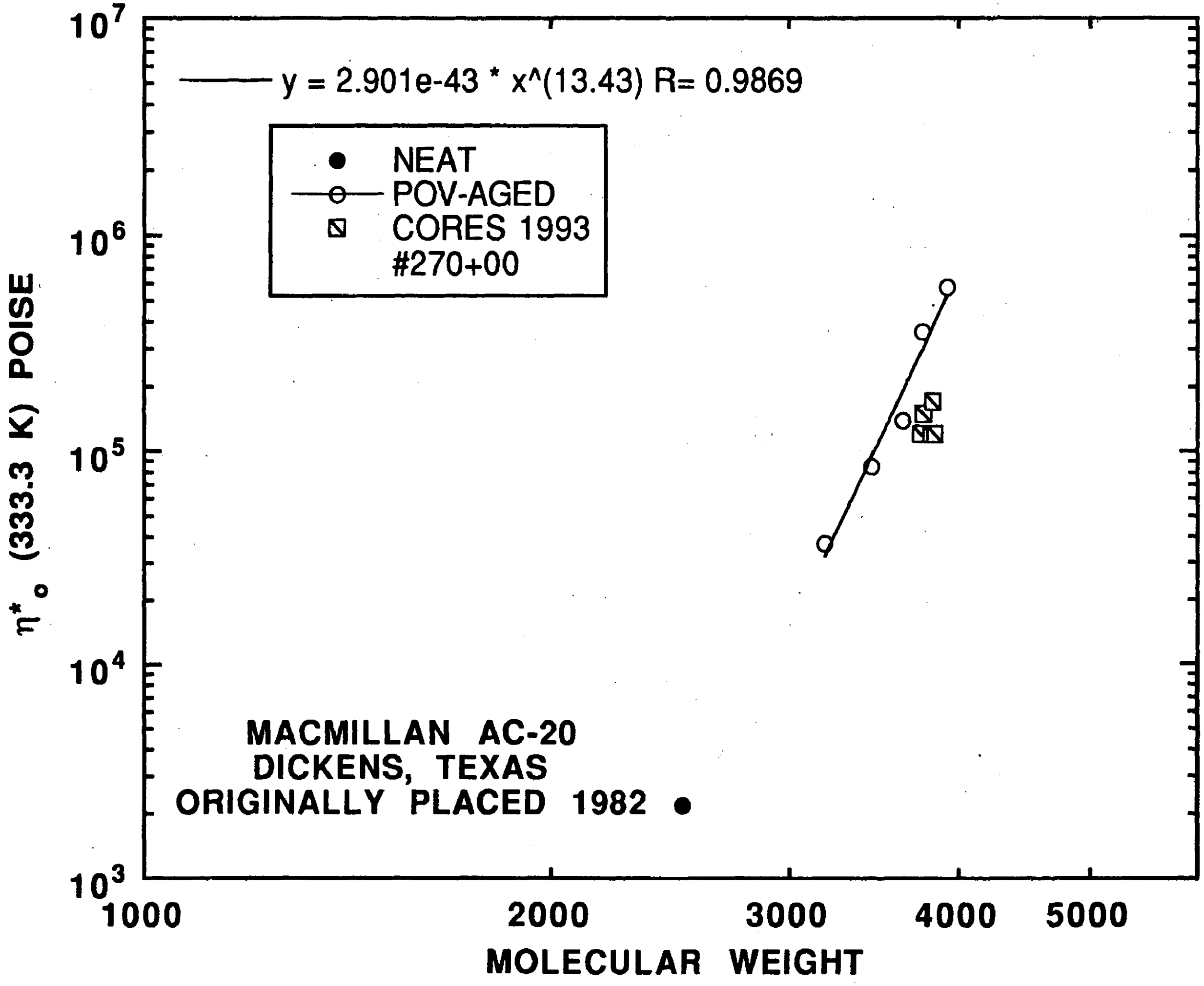

Figure E-40. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (355.5 K, 20 atm), and field-aged (#270+00, February 1993) Dickens MacMillan AC-20.

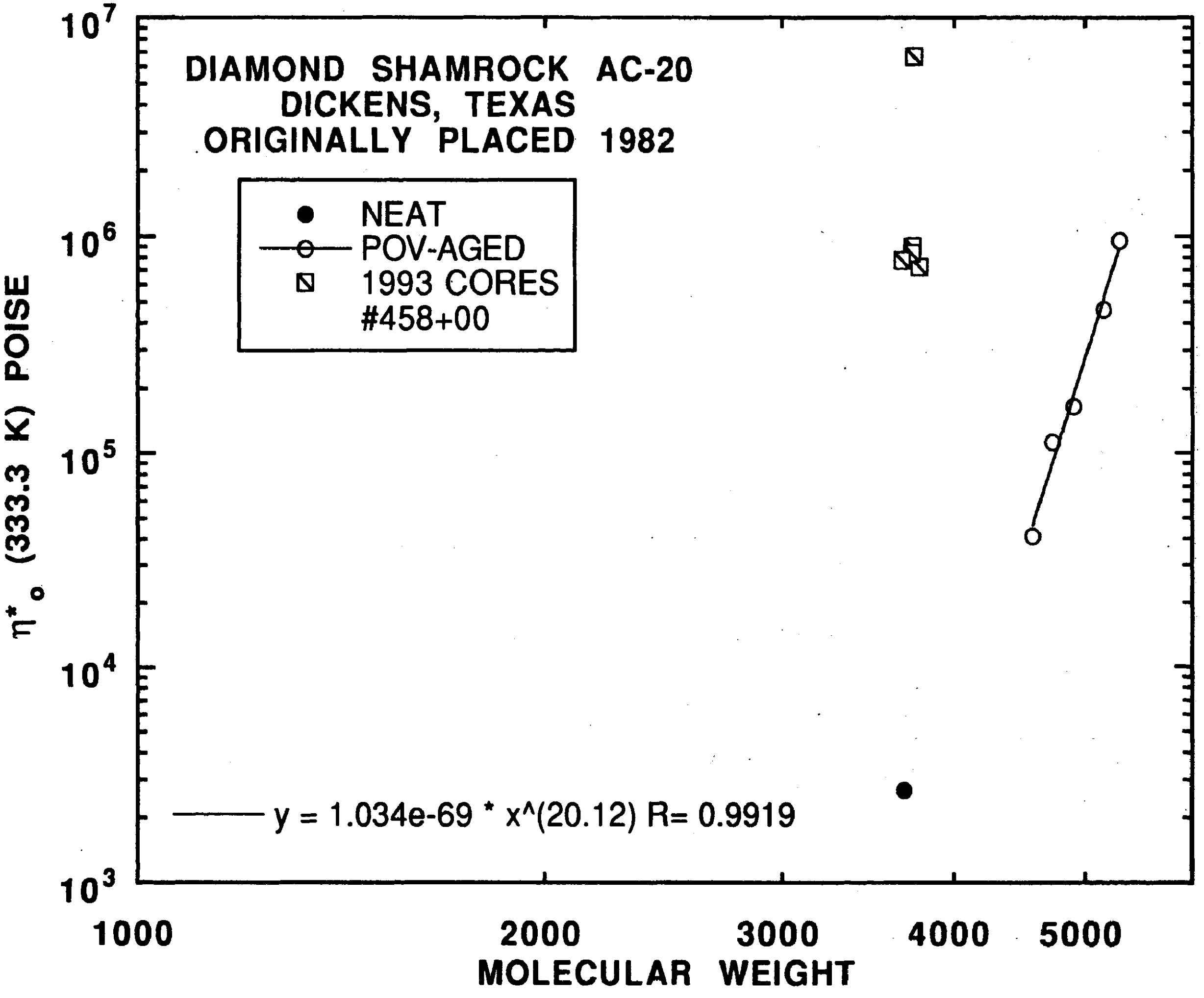

Figure E-41. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (355.5 K, 20 atm), and field-aged (#458+00, February 1993) Dickens Diamond Shamrock AC-20.

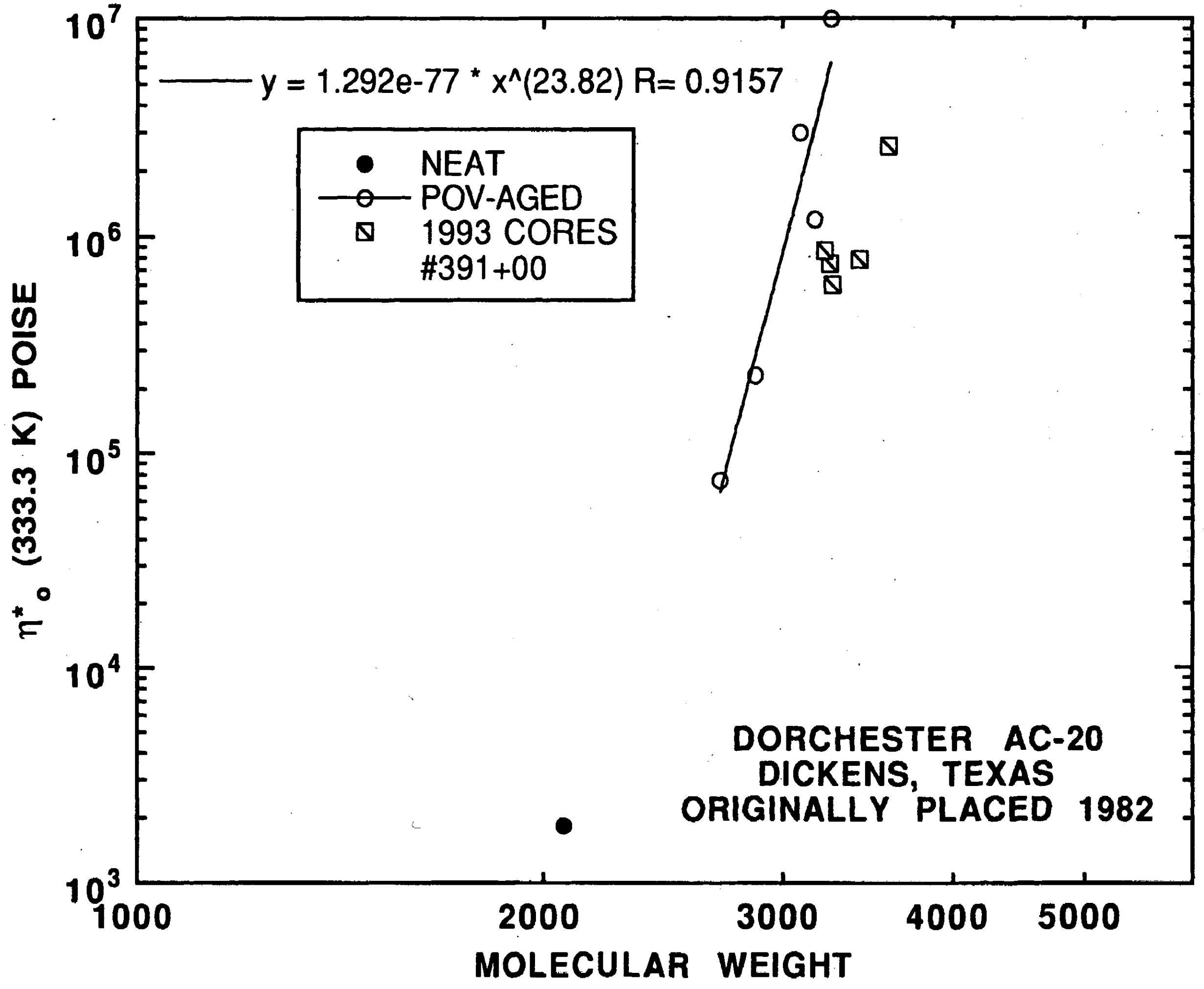

Figure E-42. Comparisons between η_o^* at 333.3 K and *MW* of neat, POV- (355.5 K, 20 atm), and field-aged (#391+00, February 1993) Dickens Dorchester AC-20.

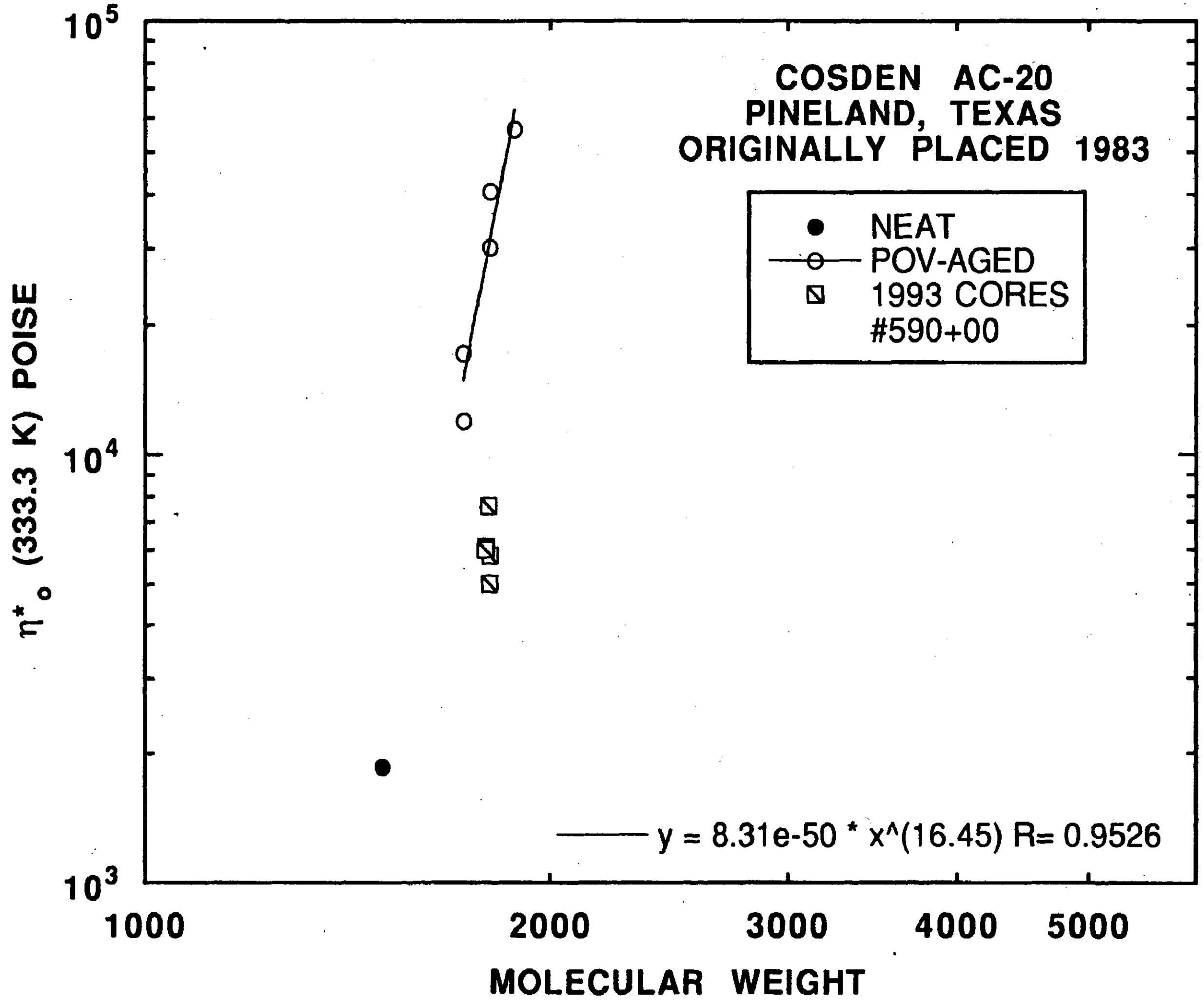

Figure E-43. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (333.3 K, 20 atm), and field-aged (#590+00, March 1993) Pineland Cosden AC-20.

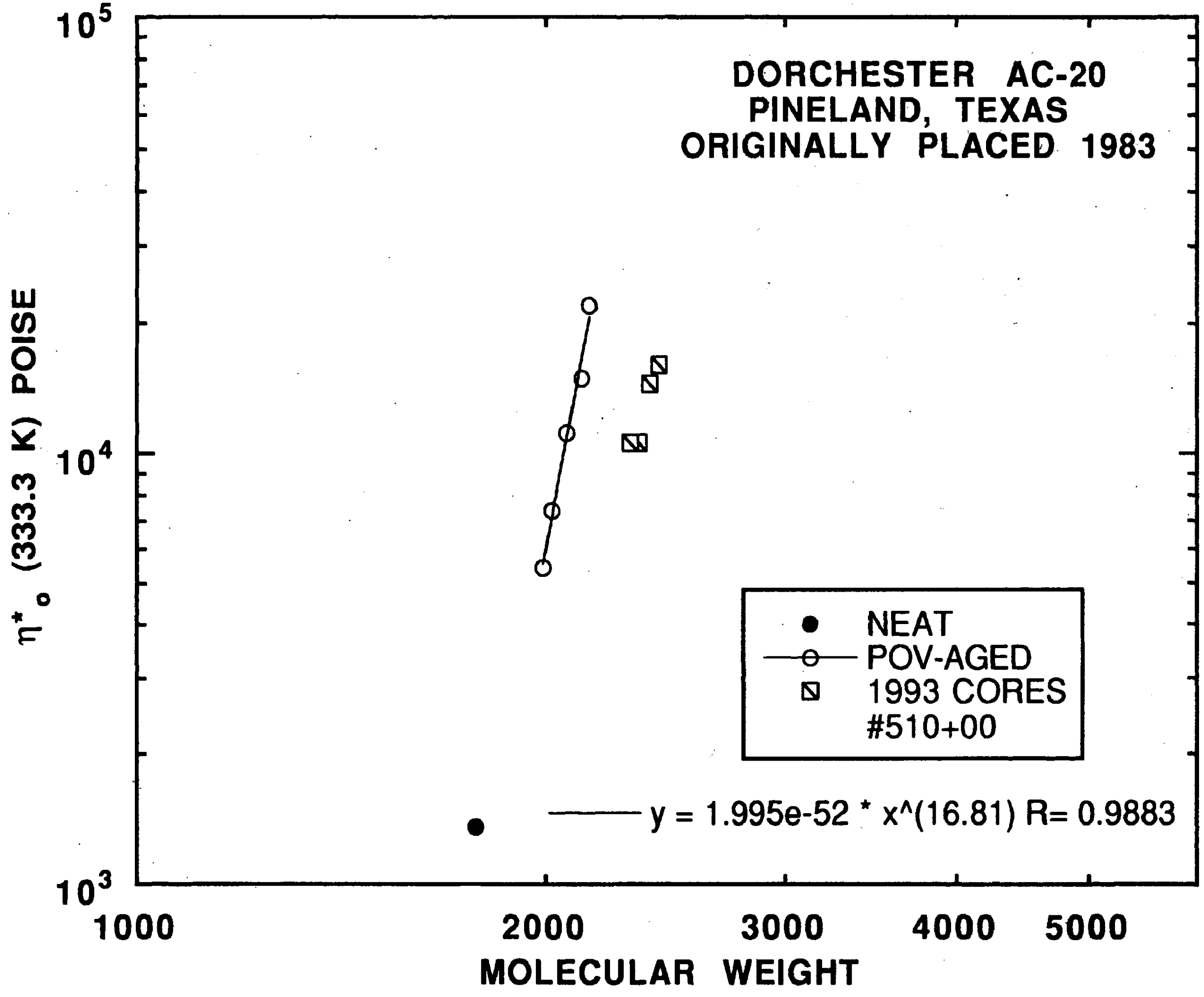

Figure E-44. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (333.3 K, 20 atm), and field-aged (#510+00, March 1993) Pineland Dorchester AC-20.

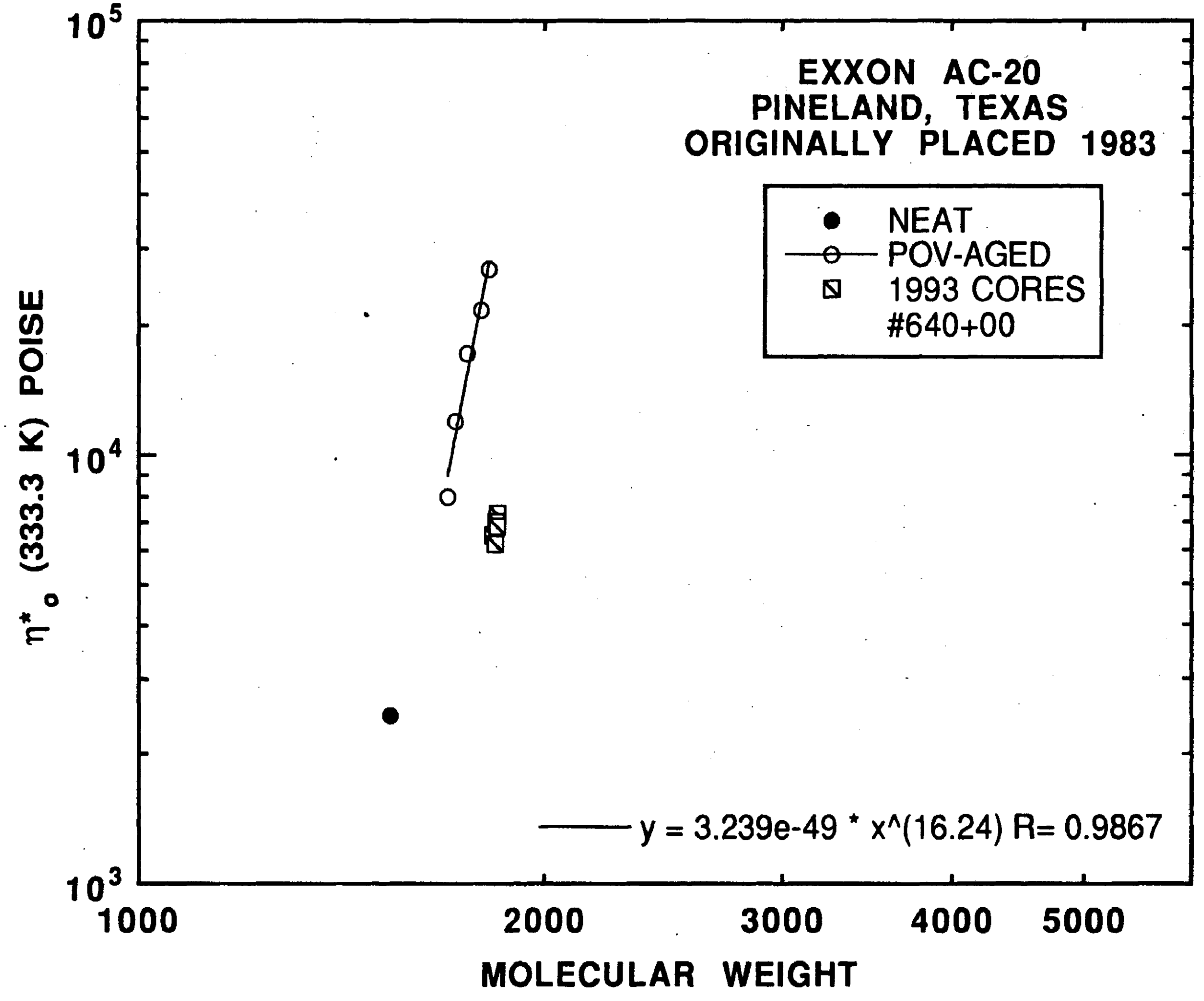

Figure E-45. Comparisons between η_o^* at 333.3 K and *MW* of neat, POV- (333.3 K, 20 atm), and field-aged (#640+00, March 1993) Pineland Exxon AC-20.

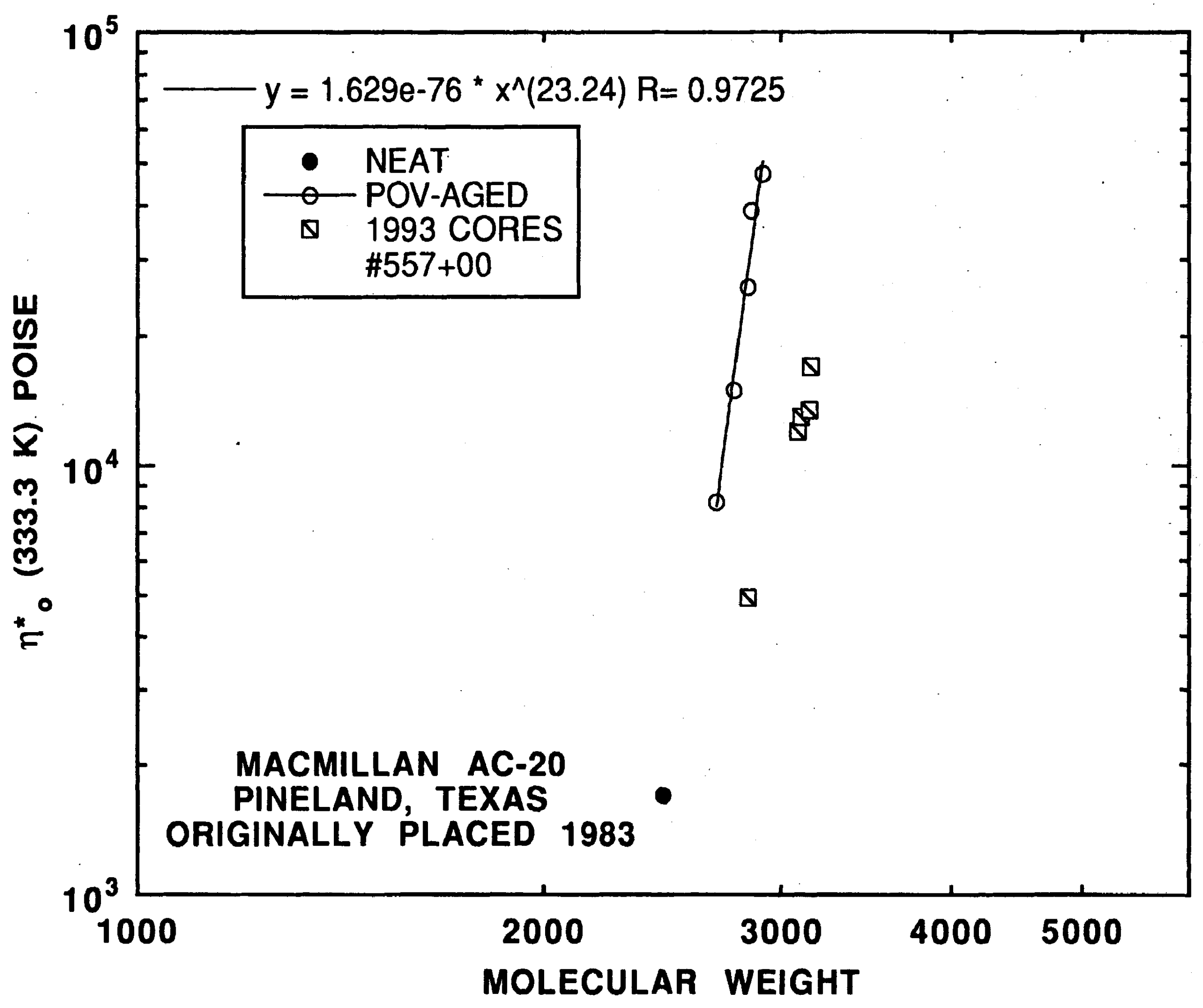

Figure E-46. Comparisons between η_0^* at 333.3 K and *MW* of neat, POV- (333.3 K, 20 atm), and field-aged (#557+00, March 1993) Pineland MacMillan AC-20.

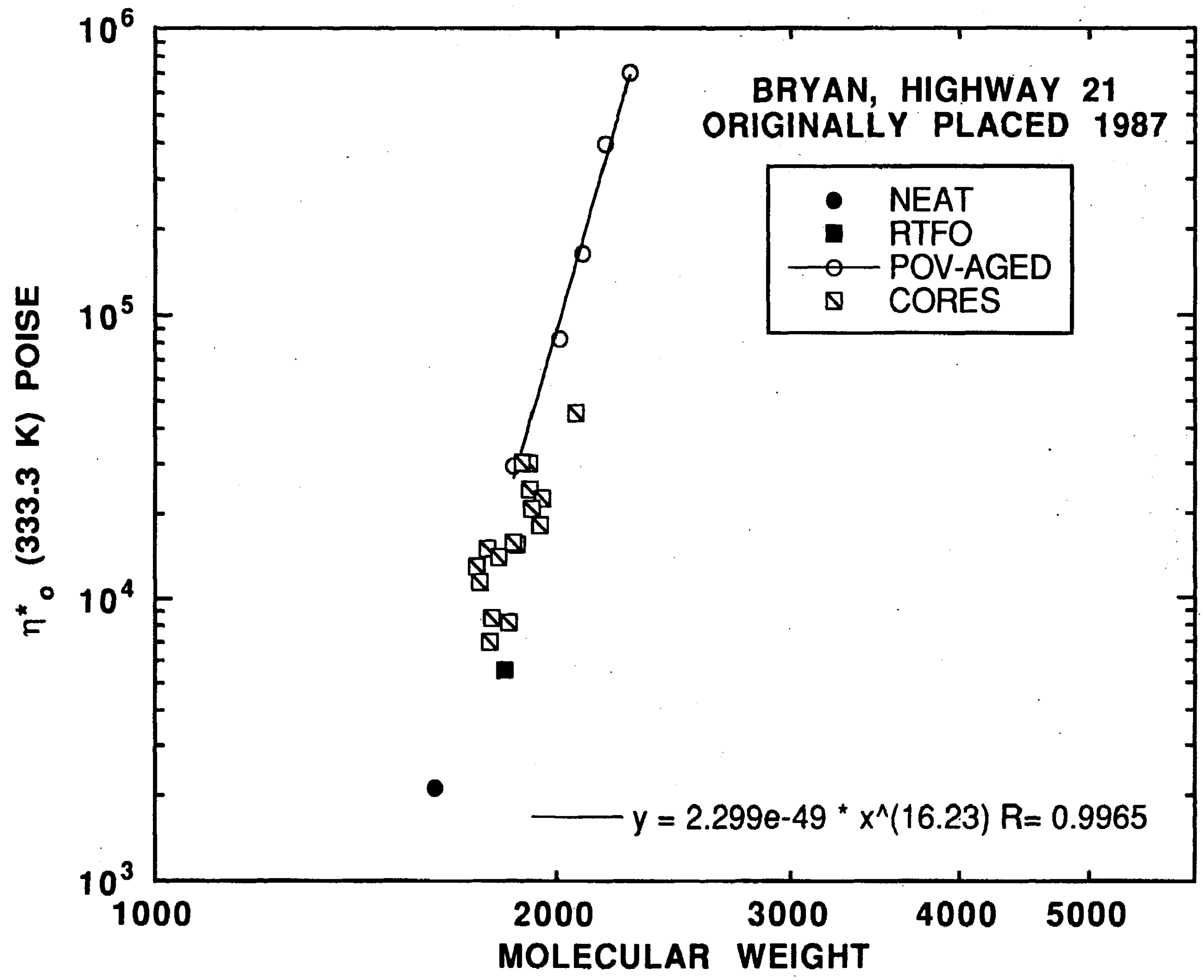

Figure E-47. Comparisons between η_0^* at 333.3 K and *MW* of neat, RTFO, POV-(344.4 K, 20 atm), and field-aged Bryan Exxon AC-20.

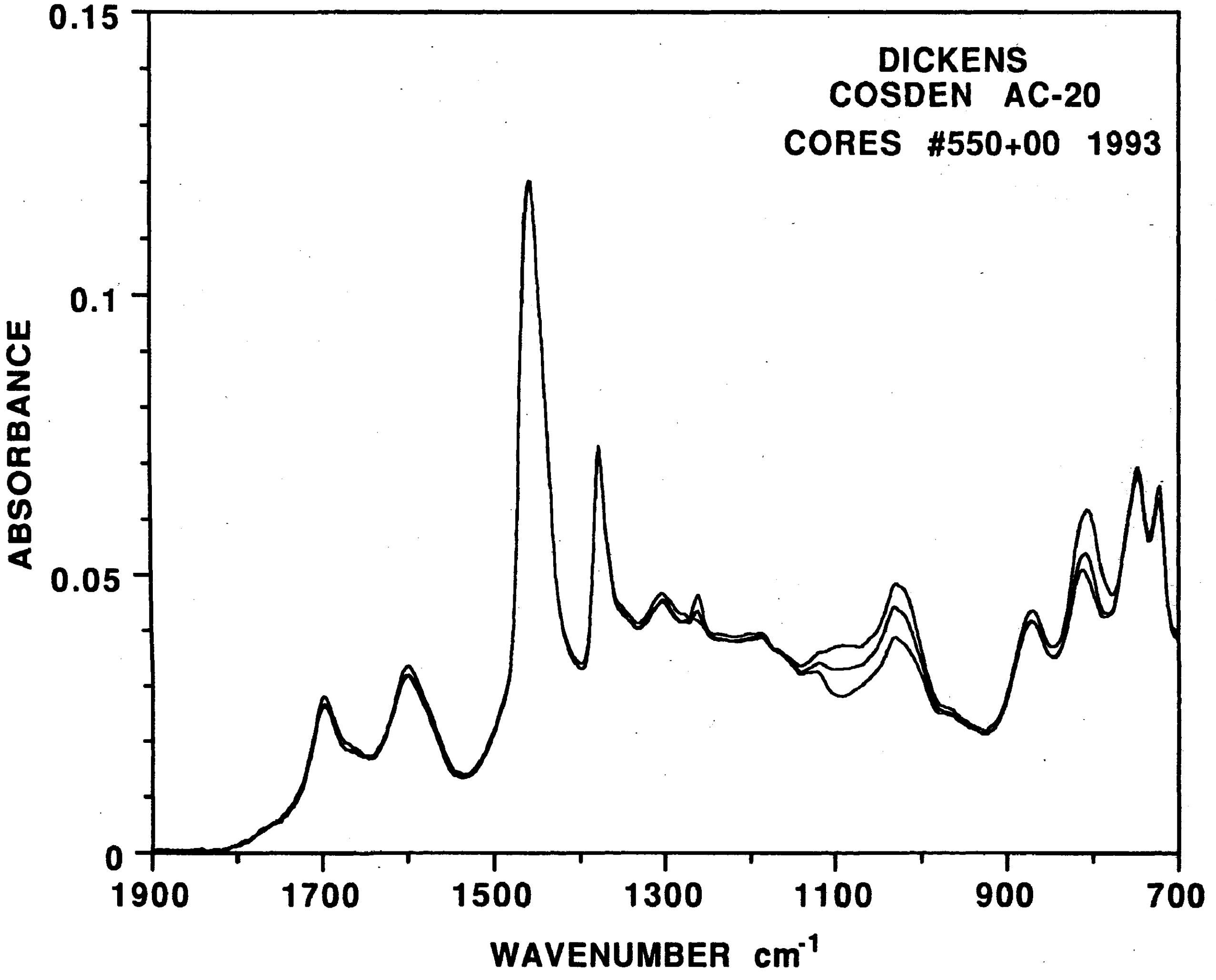

Figure E-48. IR spectra of field-aged Dickens Cosden AC-20 extracted asphalt from #550+00. Cored February, 1993.

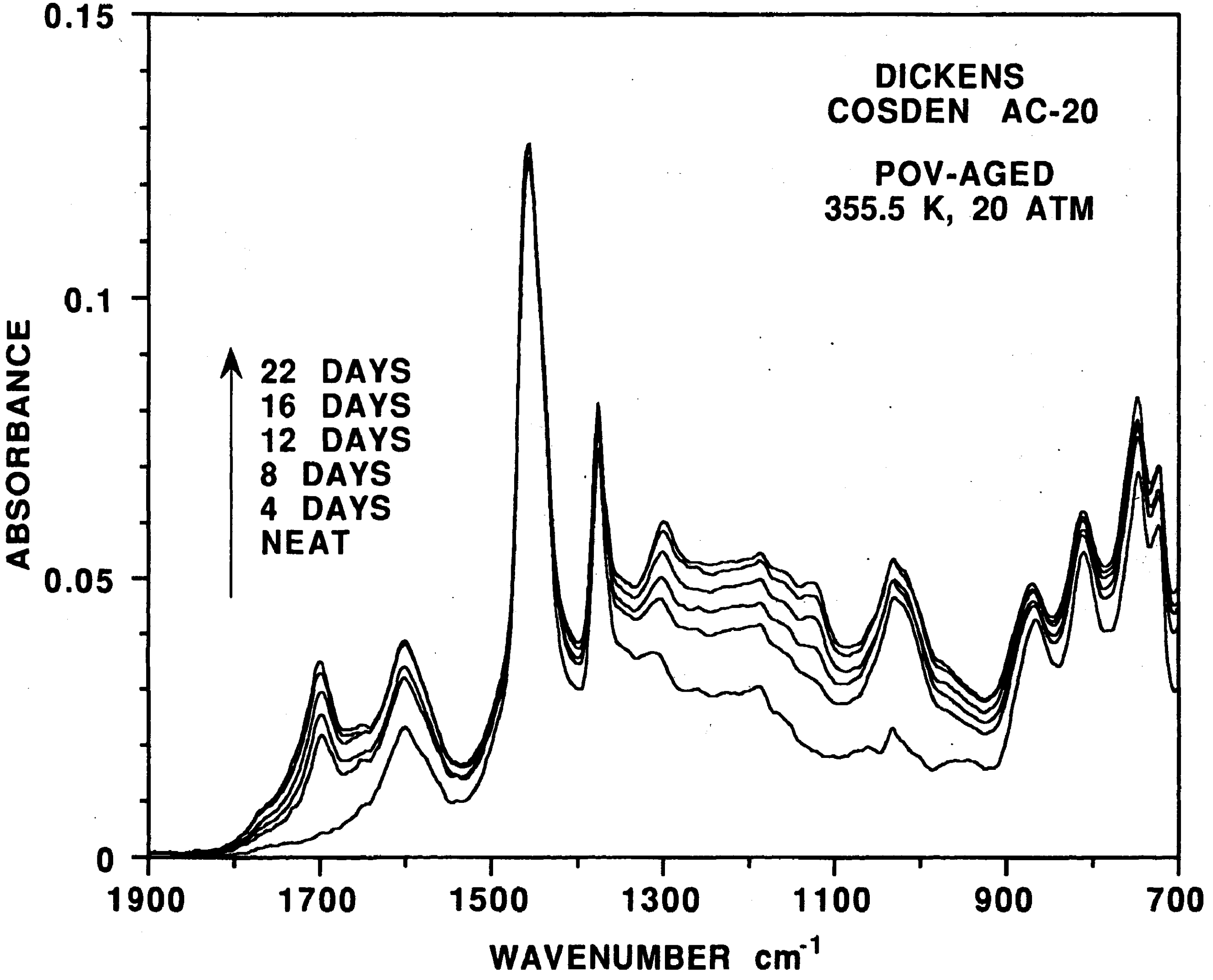

Figure E-49. IR spectra of neat and POV-aged Dickens Cosden AC-20 at 355.5 K and 20 atm.

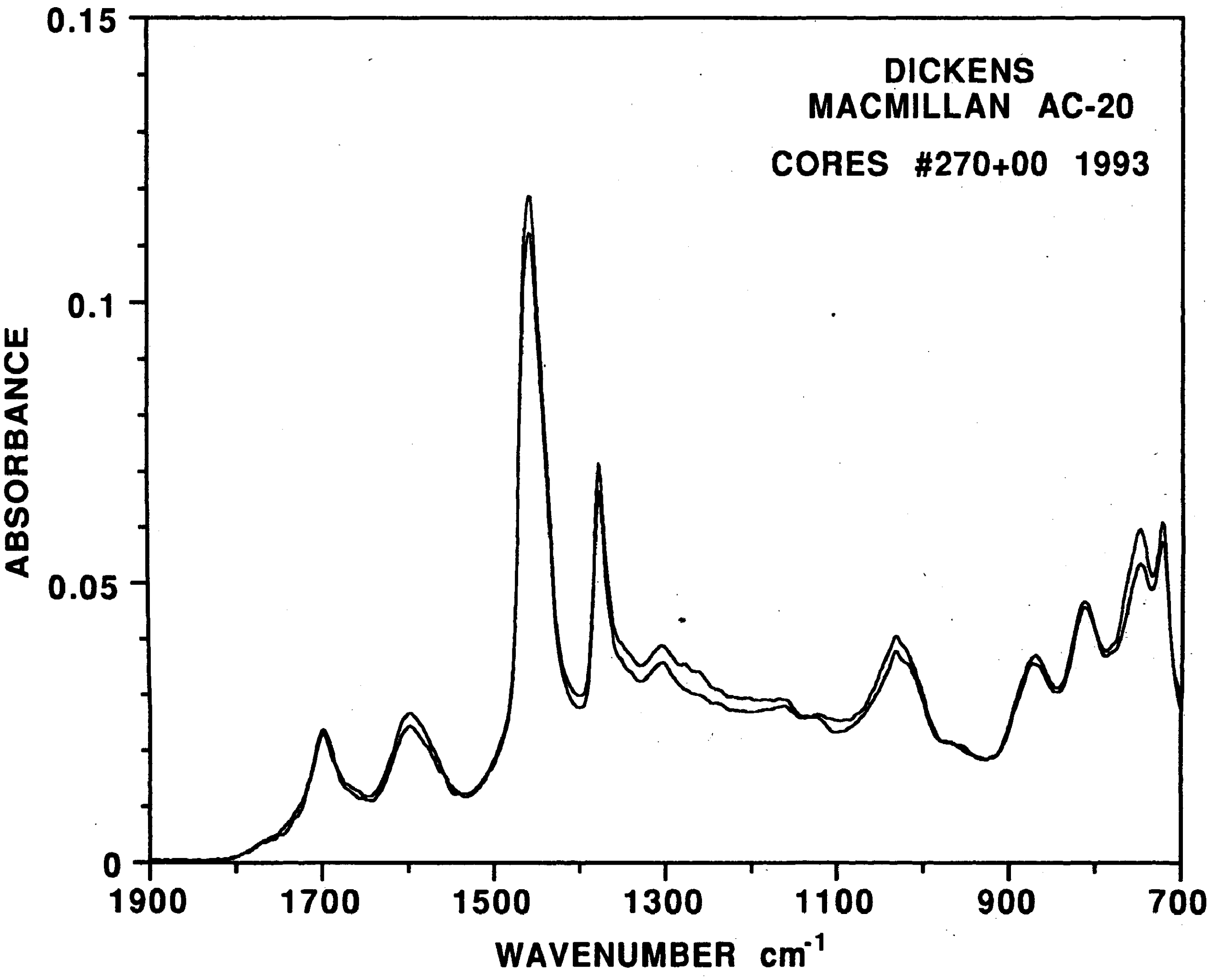

Figure E-50. IR spectra of field-aged Dickens MacMillan AC-20 extracted asphalt from #270+00. Cored February, 1993.

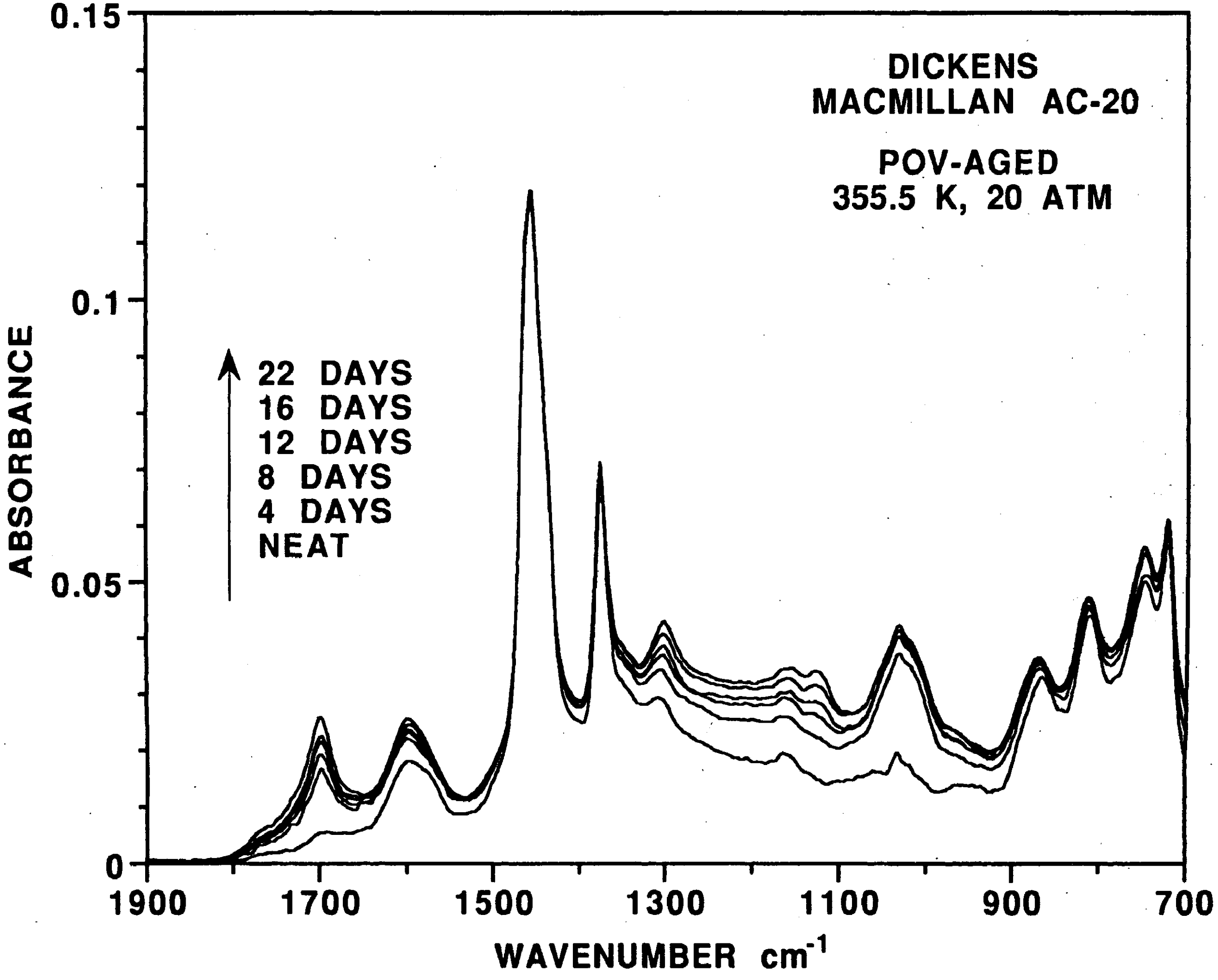

Figure E-51. IR spectra of neat and POV-aged Dickens MacMillan AC-20 at 355.5 K and 20 atm.

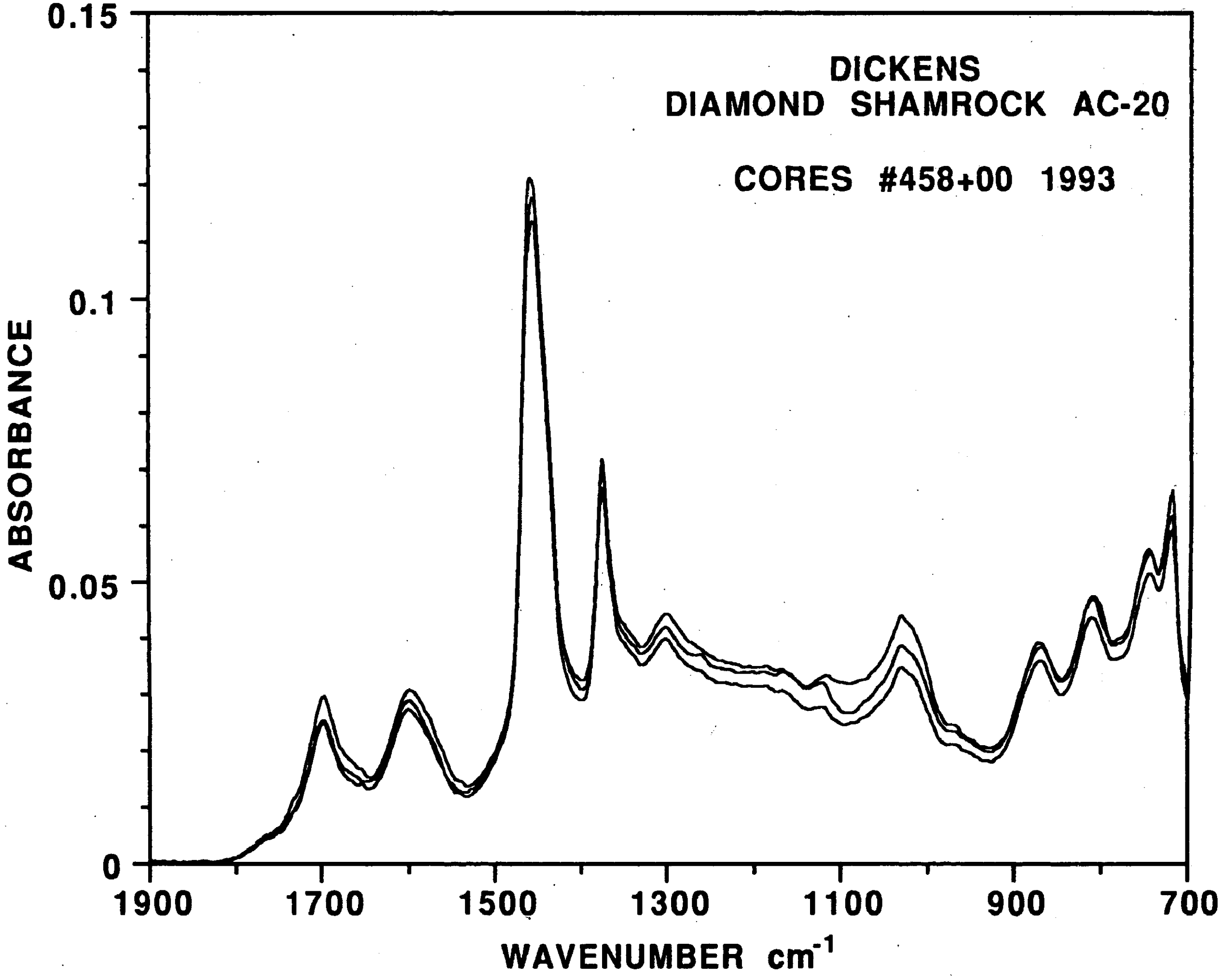

Figure E-52. IR spectra of field-aged Dickens Diamond Shamrock AC-20 extracted asphalt from #458+00. Cored February, 1993.

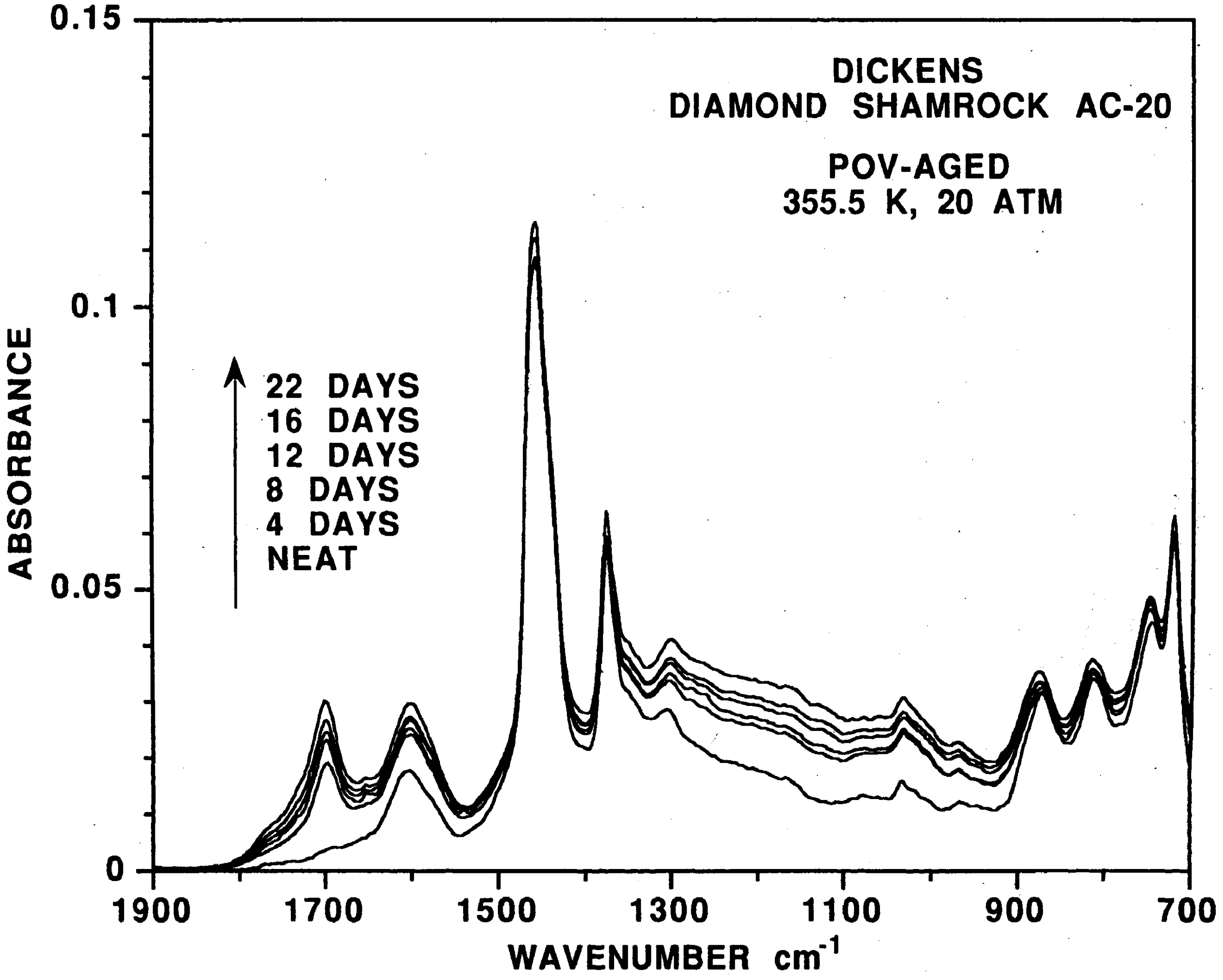

Figure E-53. IR spectra of neat and POV-aged Dickens Diamond Shamrock AC-20 at 355.5 K and 20 atm.

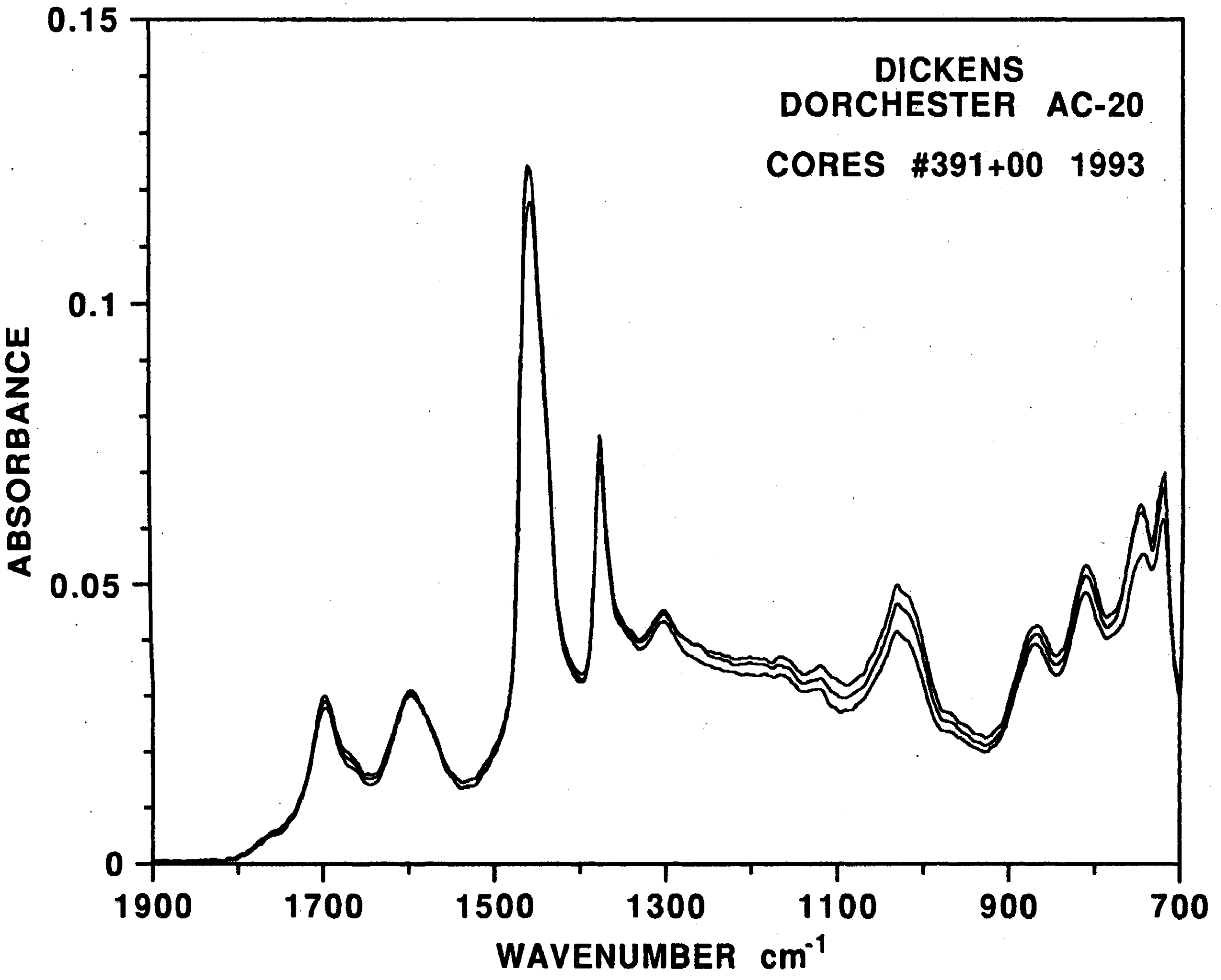

Figure E-54. IR spectra of field-aged Dickens Dorchester AC-20 extracted asphalt from #391+00. Cored February, 1993.

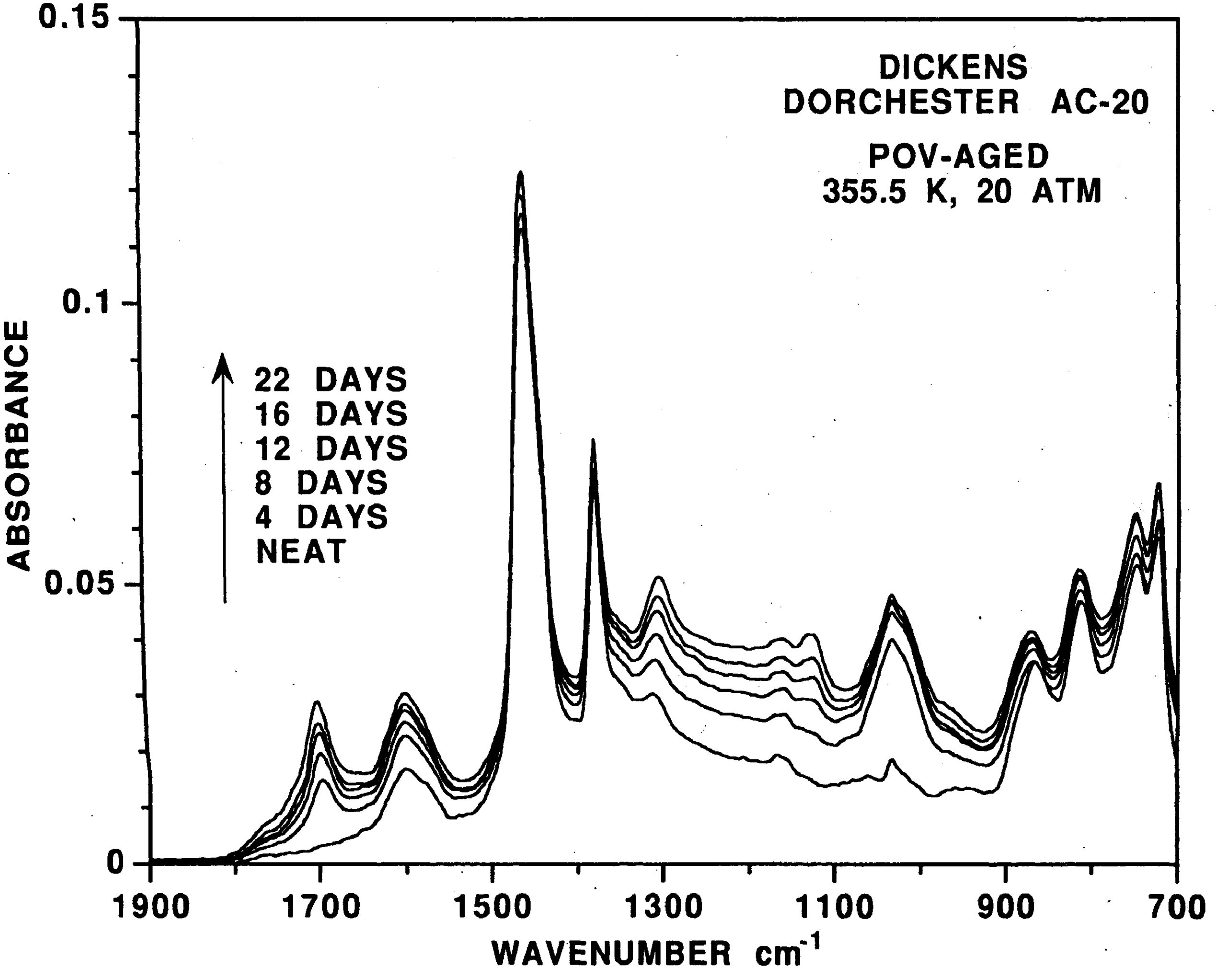

Figure E-55. IR spectra of neat and POV-aged Dickens Dorchester AC-20 at 355.5 K and 20 atm.

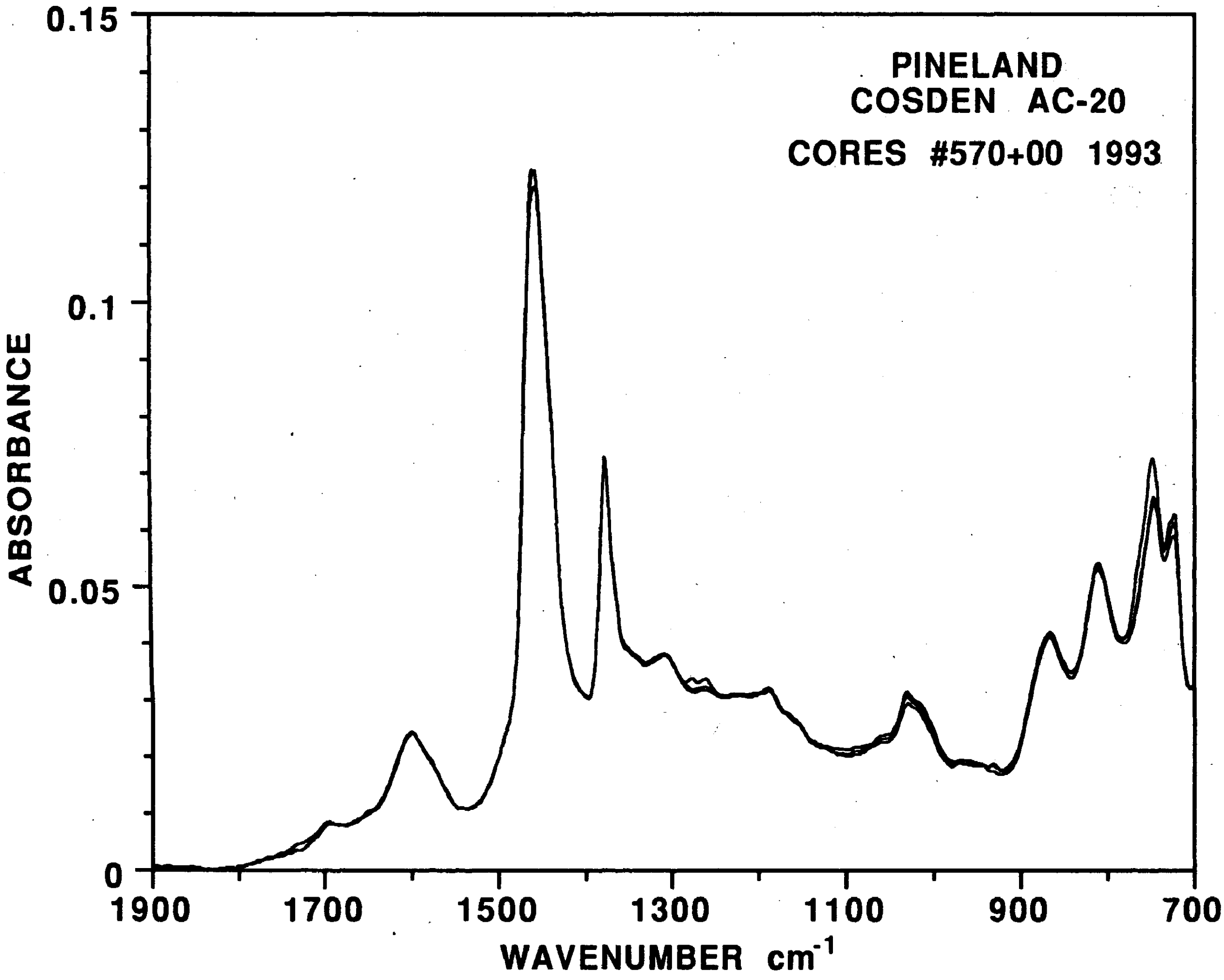

Figure E-56. IR spectra of field-aged Pineland Cosden AC-20 extracted asphalt from #590+00. Cored March, 1993.

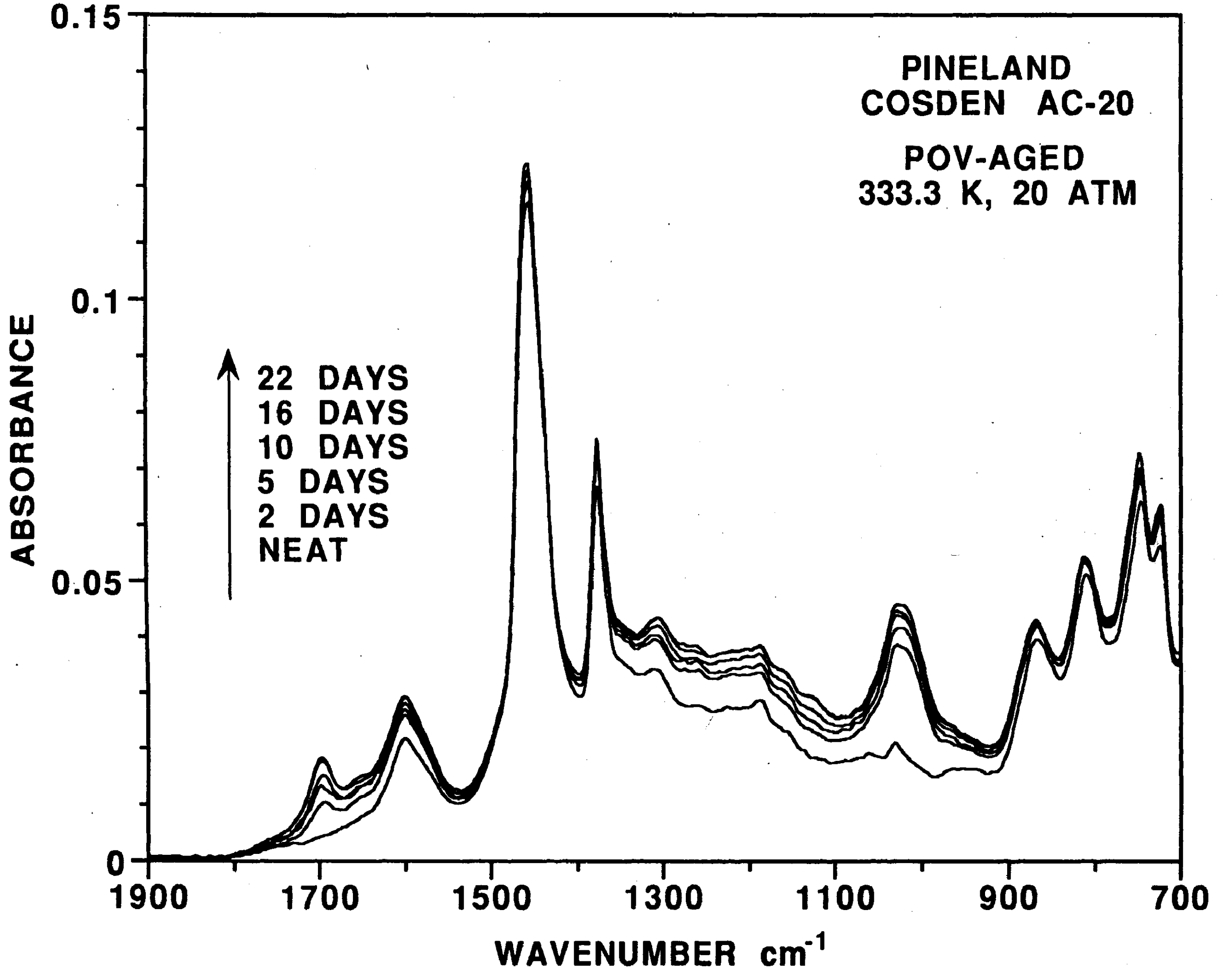

Figure E-57. IR spectra of neat and POV-aged Pineland Cosden AC-20 at 333.3 K and 20 atm.

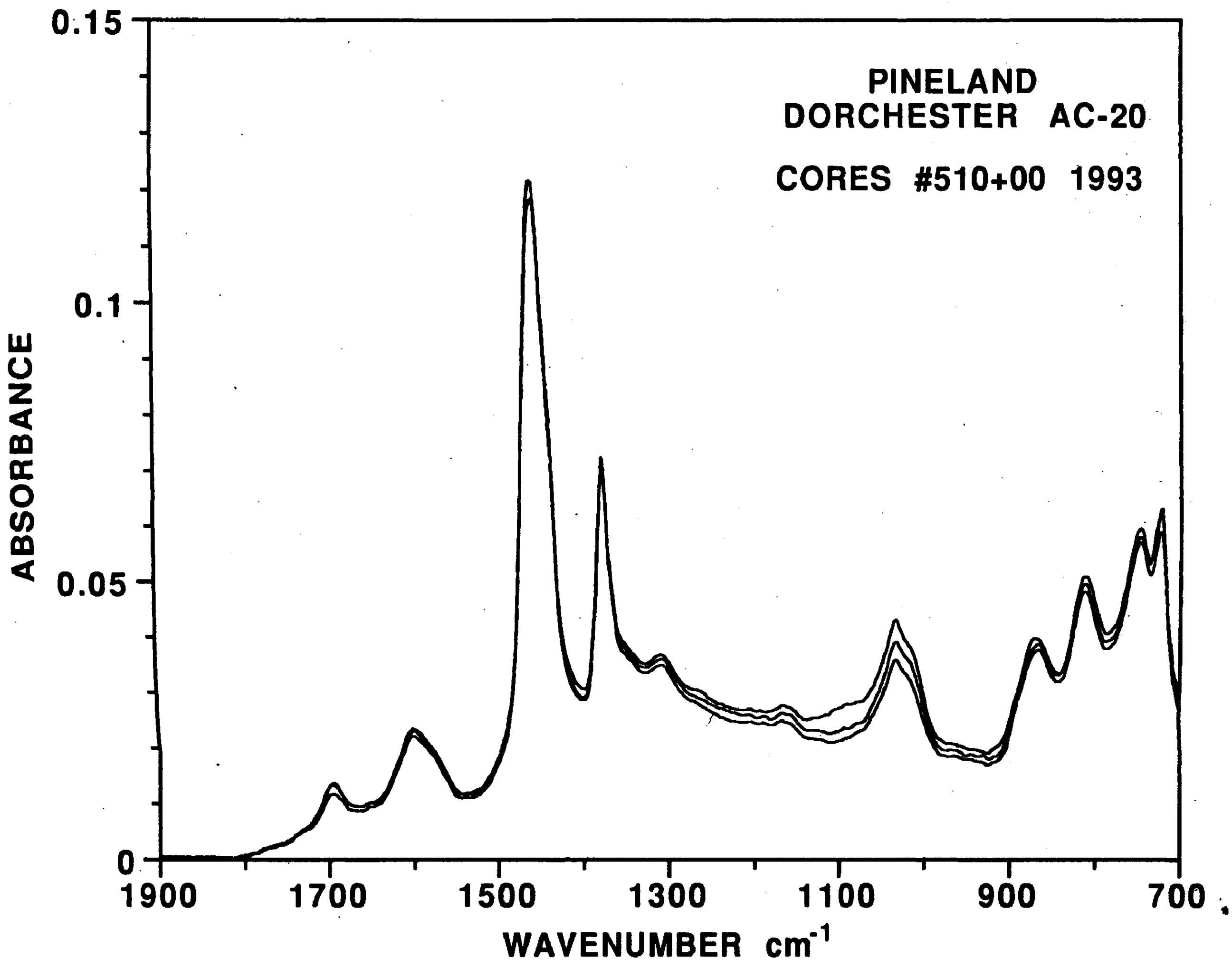

Figure E-58. IR spectra of field-aged Pineland Dorchester AC-20 extracted asphalt from #510+00. Cored March, 1993.

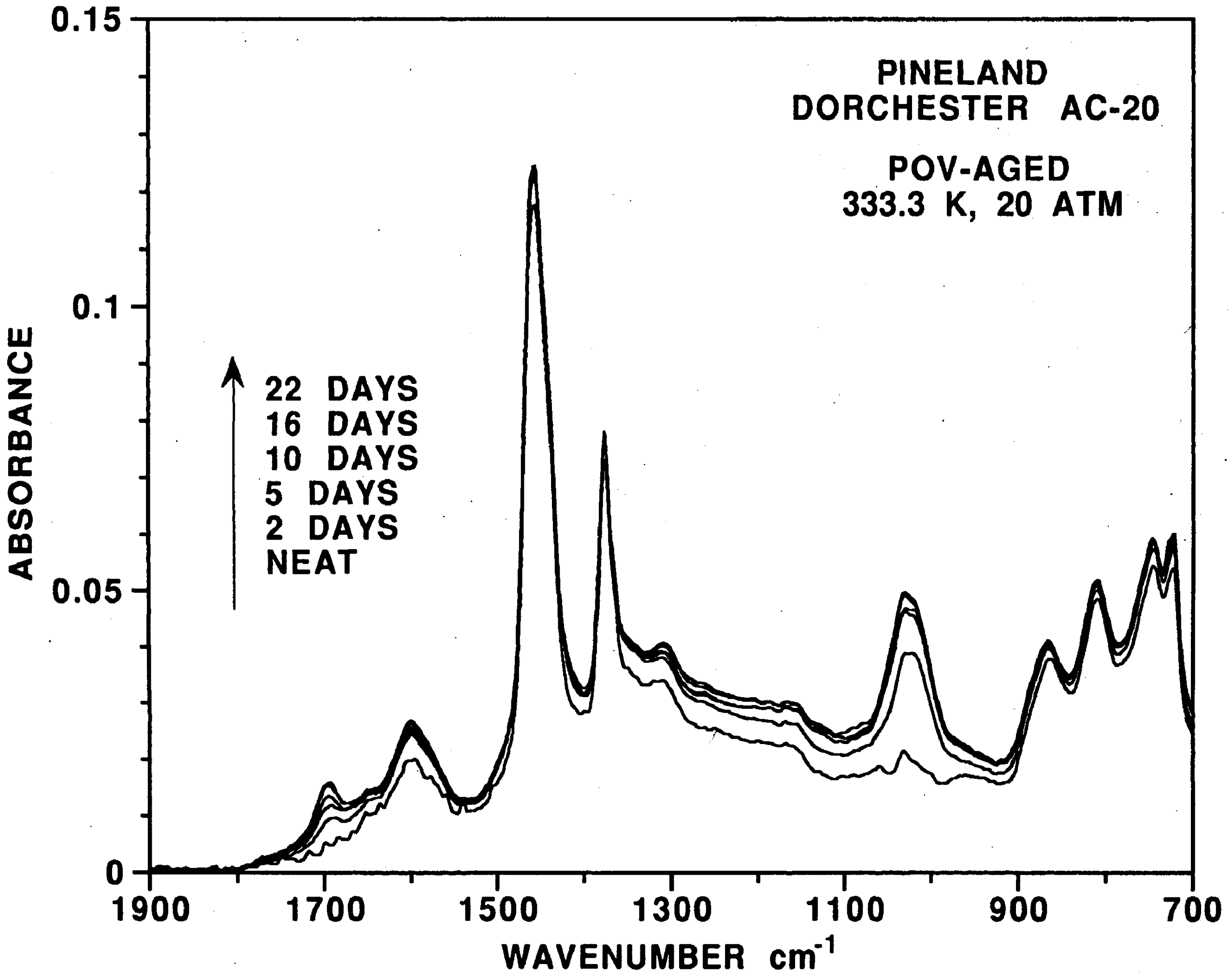

Figure E-59. IR spectra of neat and POV-aged Pineland Dorchester AC-20 at 333.3 K and 20 atm.

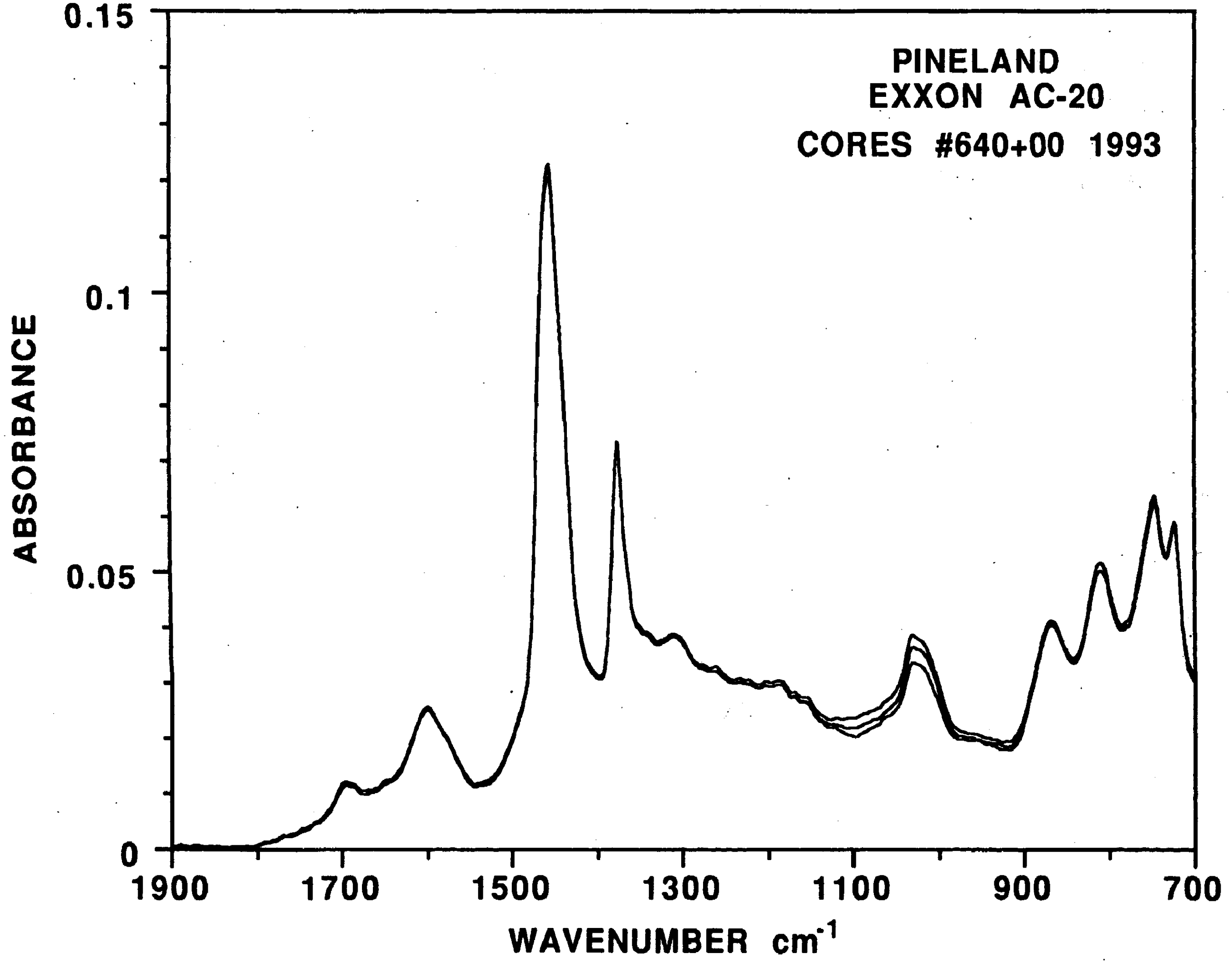

Figure E-60. IR spectra of field-aged Pineland Exxon AC-20 extracted asphalt from #640+00. Cored March, 1993.

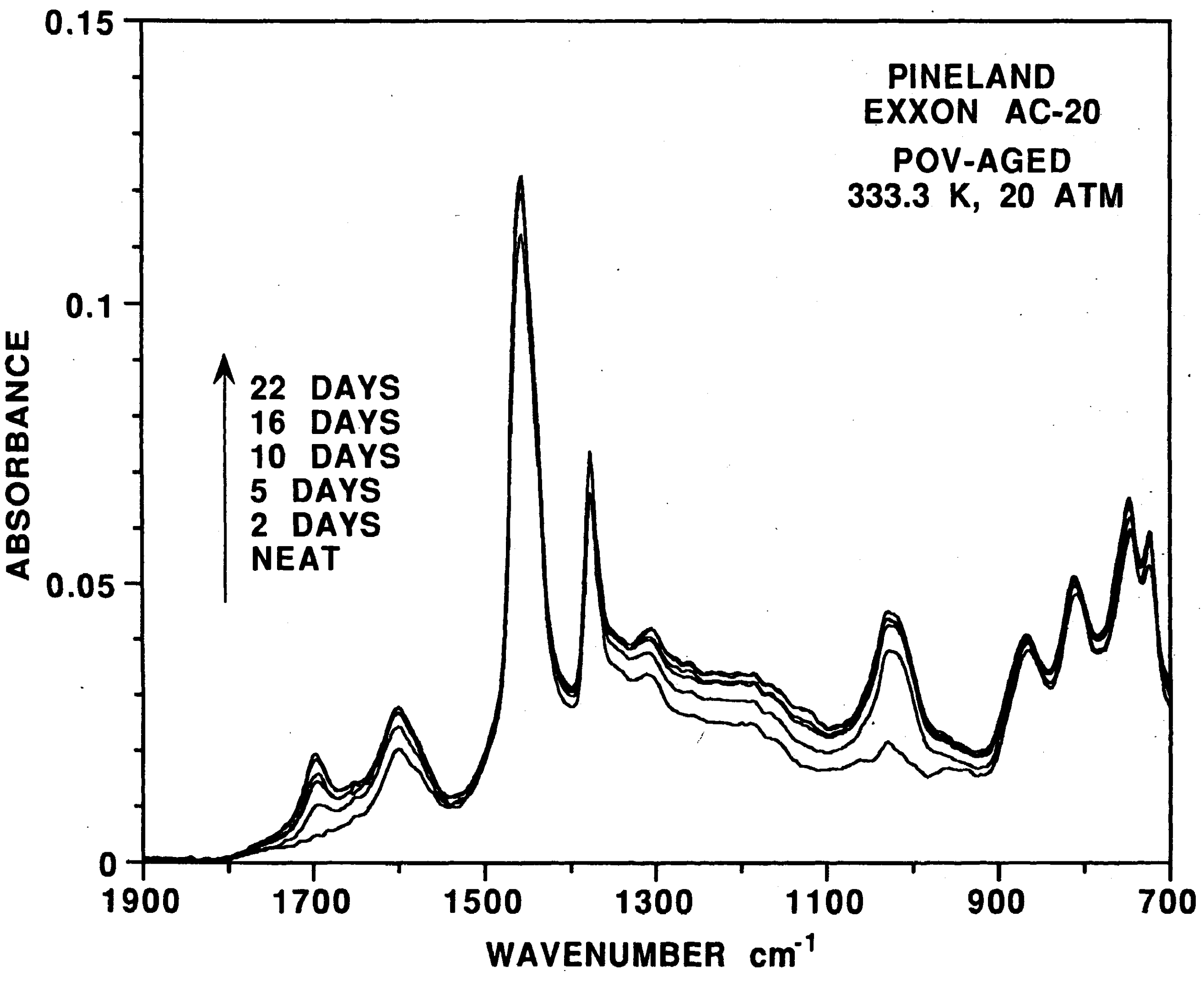

Figure E-61. IR spectra of neat and POV-aged Pineland Exxon AC-20 at 333.3 K and 20 atm.

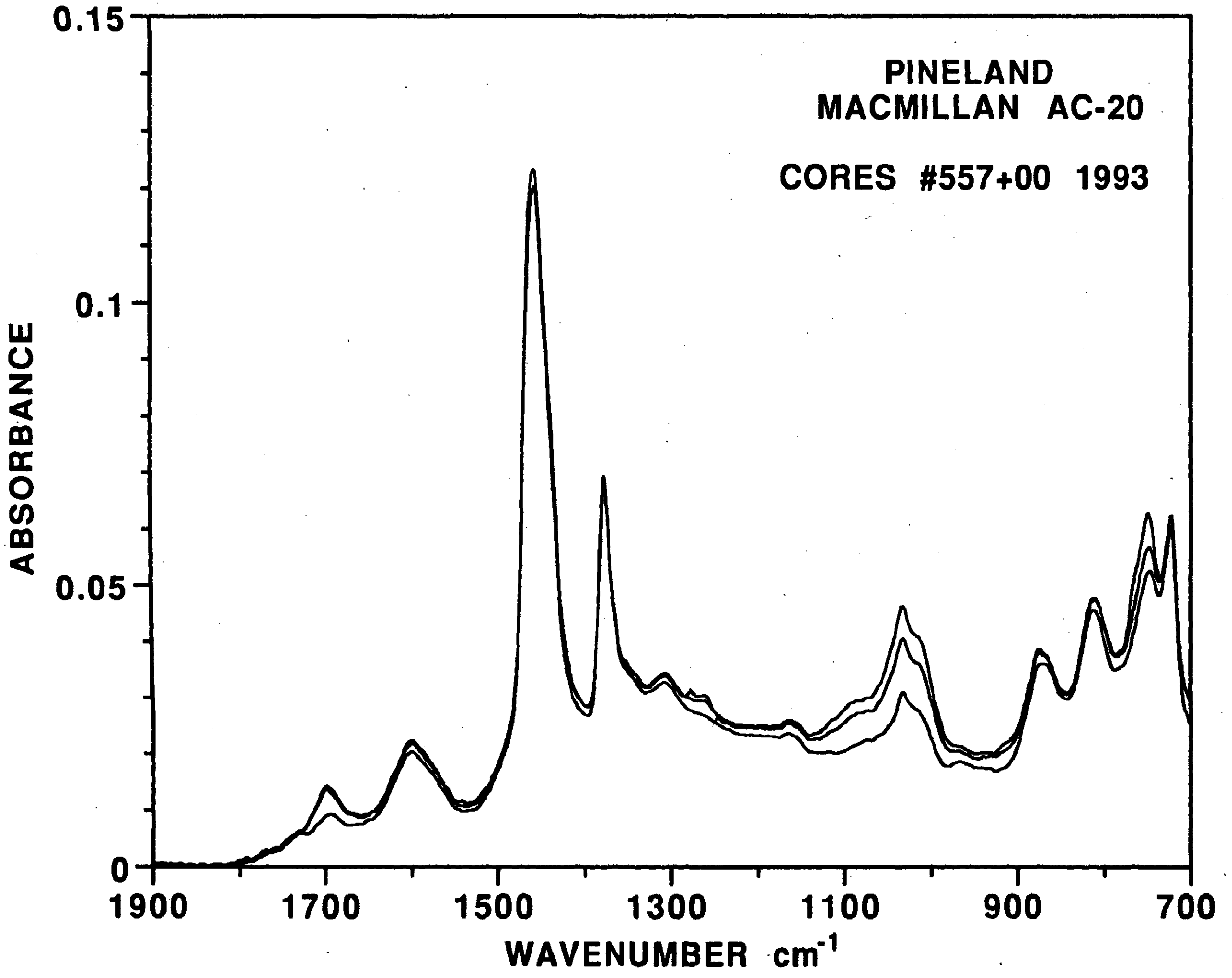

Figure E-62. IR spectra of field-aged Pineland MacMillan AC-20 extracted asphalt from #557+00. Cored March, 1993.

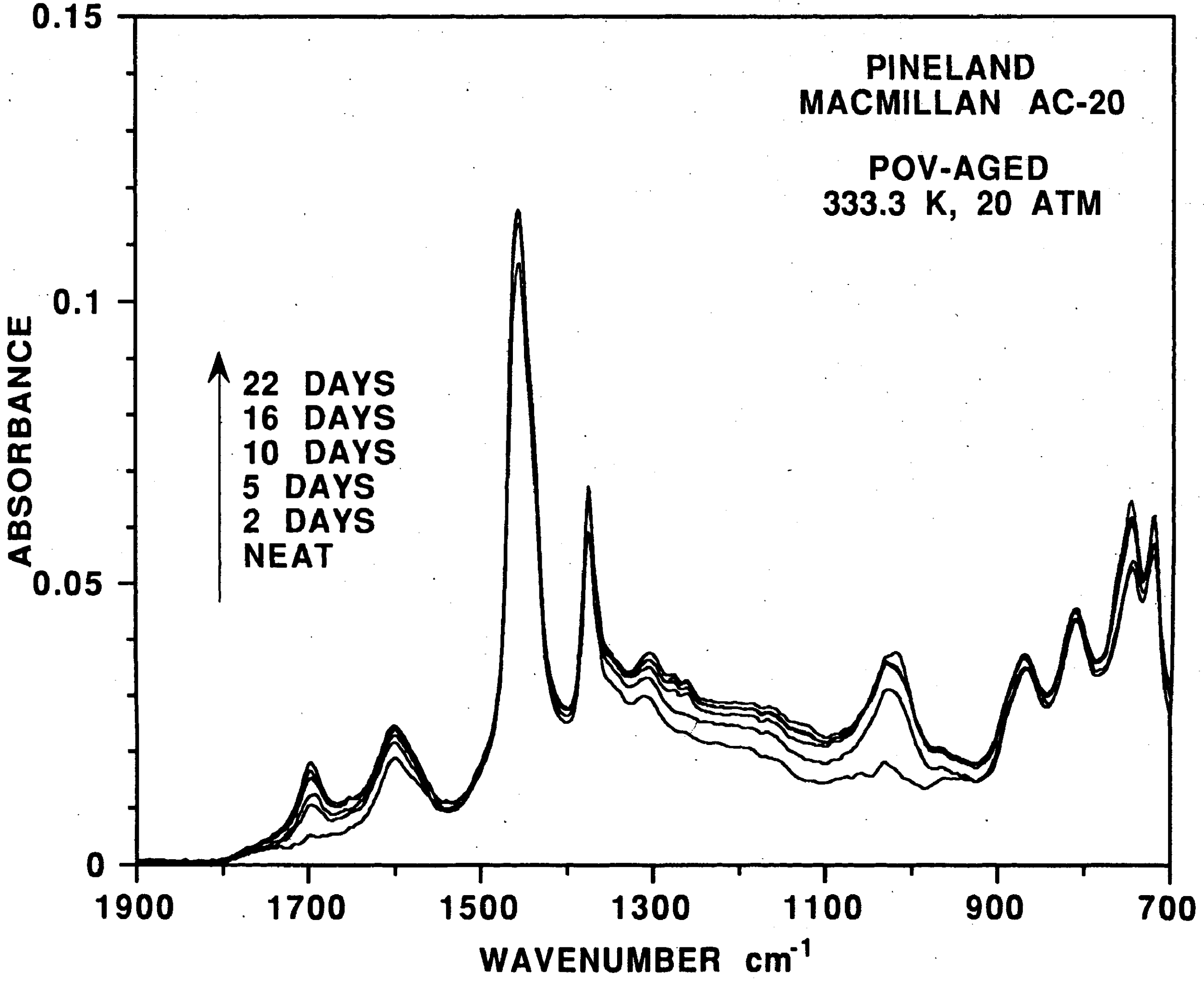

Figure E-63. IR spectra of neat and POV-aged Pineland MacMillan AC-20 at 333.3 K and 20 atm.

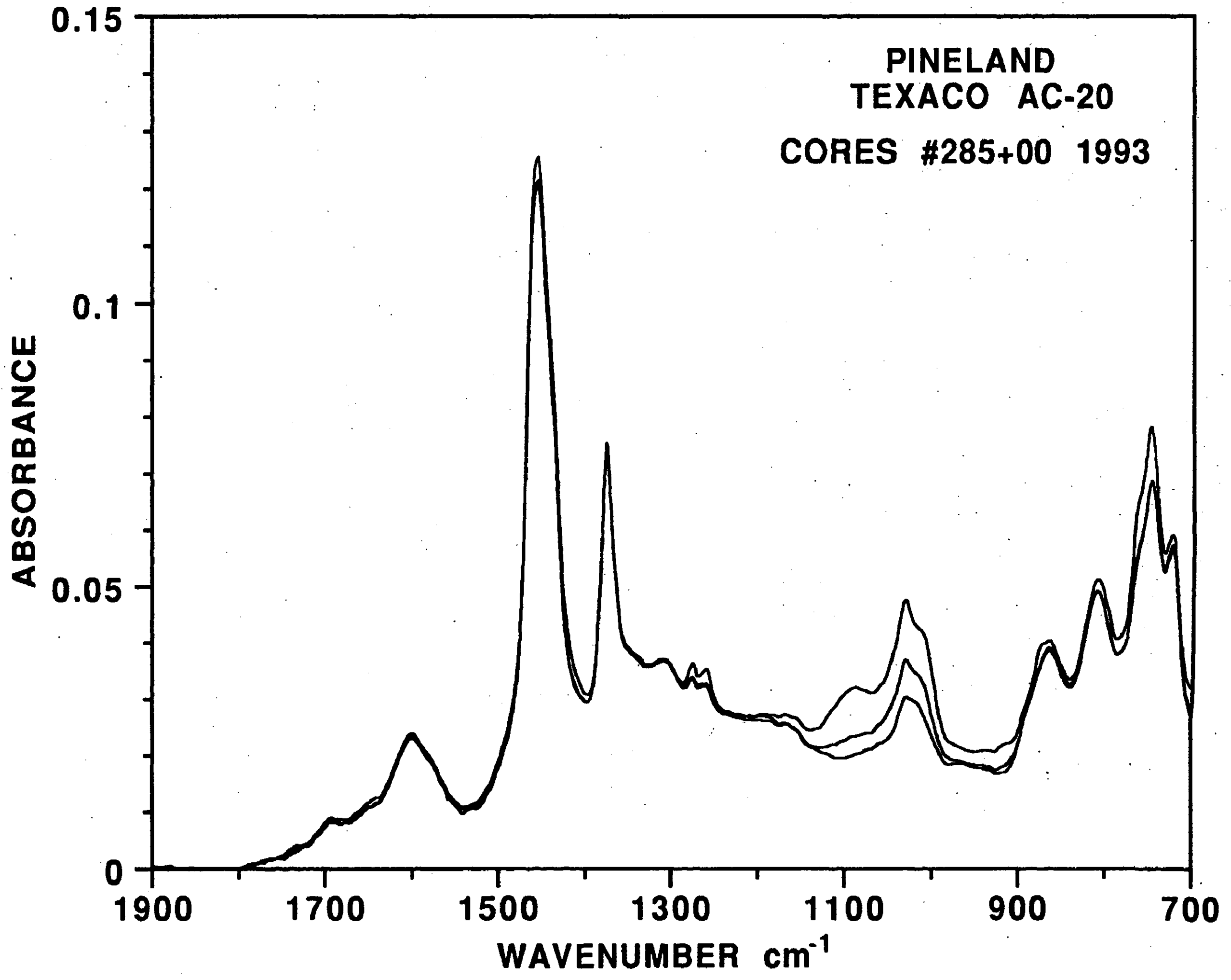

Figure E-64. IR spectra of field-aged Pineland Texaco AC-20 extracted asphalt from #285+00. Cored March, 1993.

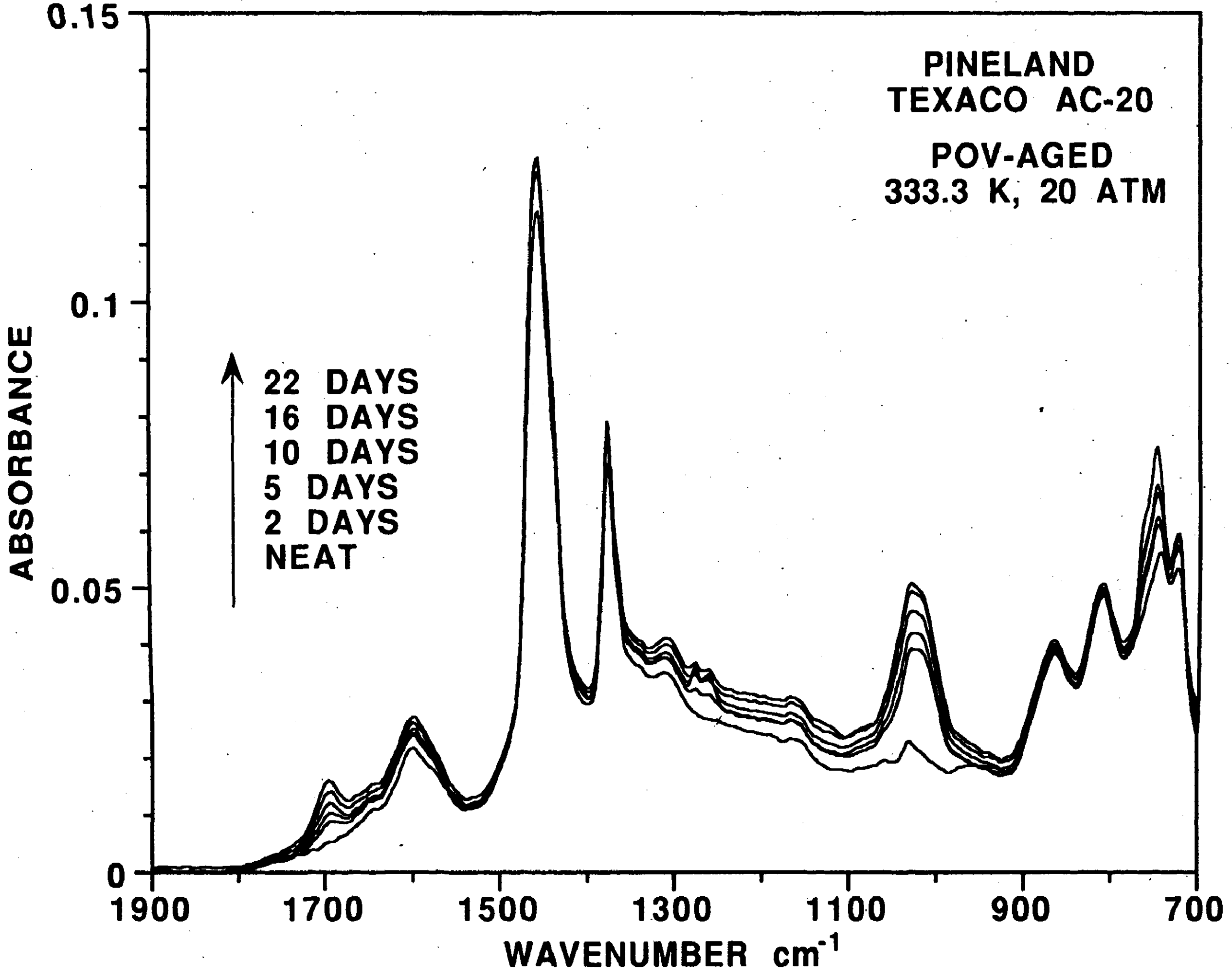

Figure E-65. IR spectra of neat and POV-aged Pineland Texaco AC-20 at 333.3 K and 20 atm.

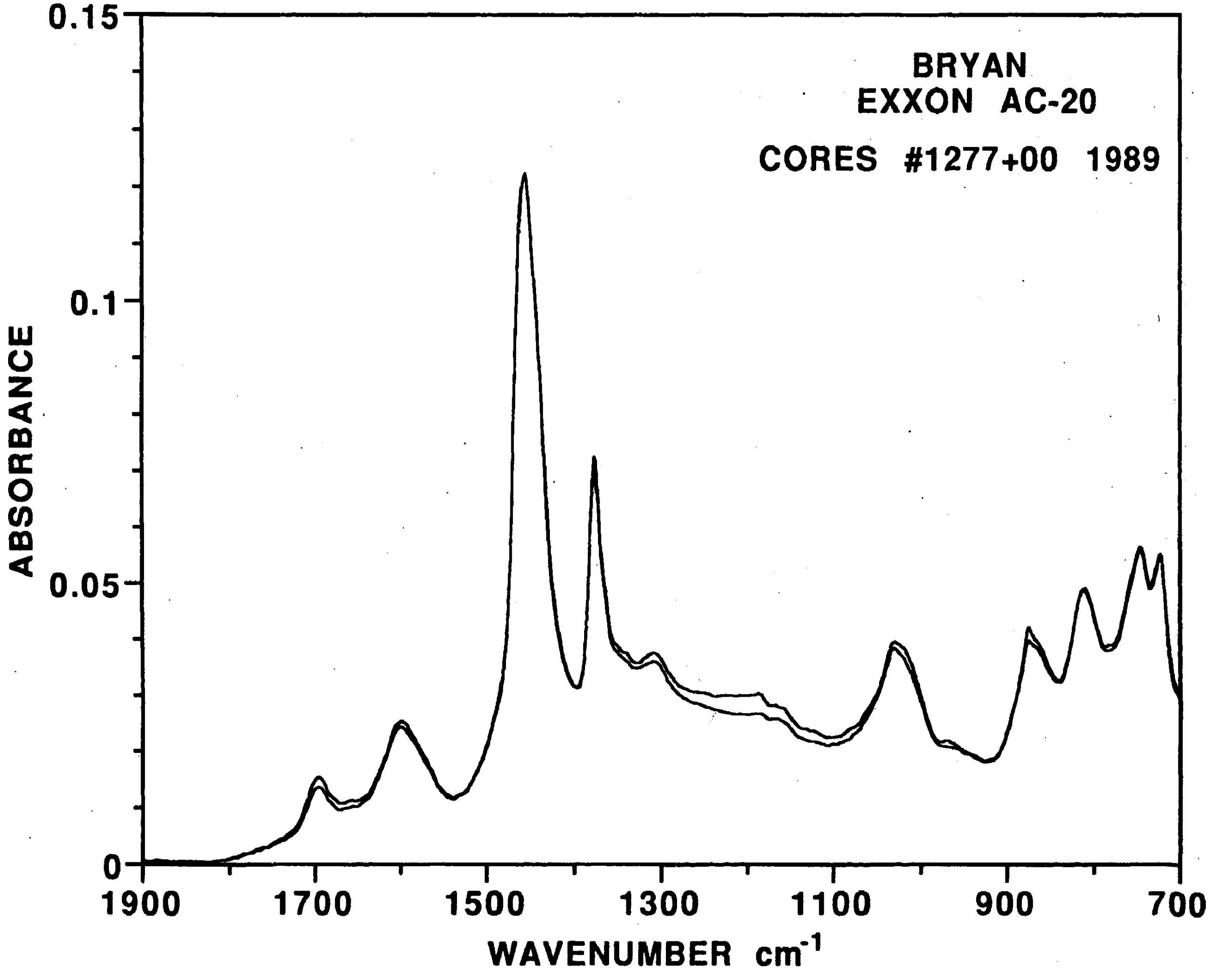

Figure E-66. IR spectra of field-aged Bryan Exxon AC-20 extracted asphalt from #1277. Cored November, 1992.

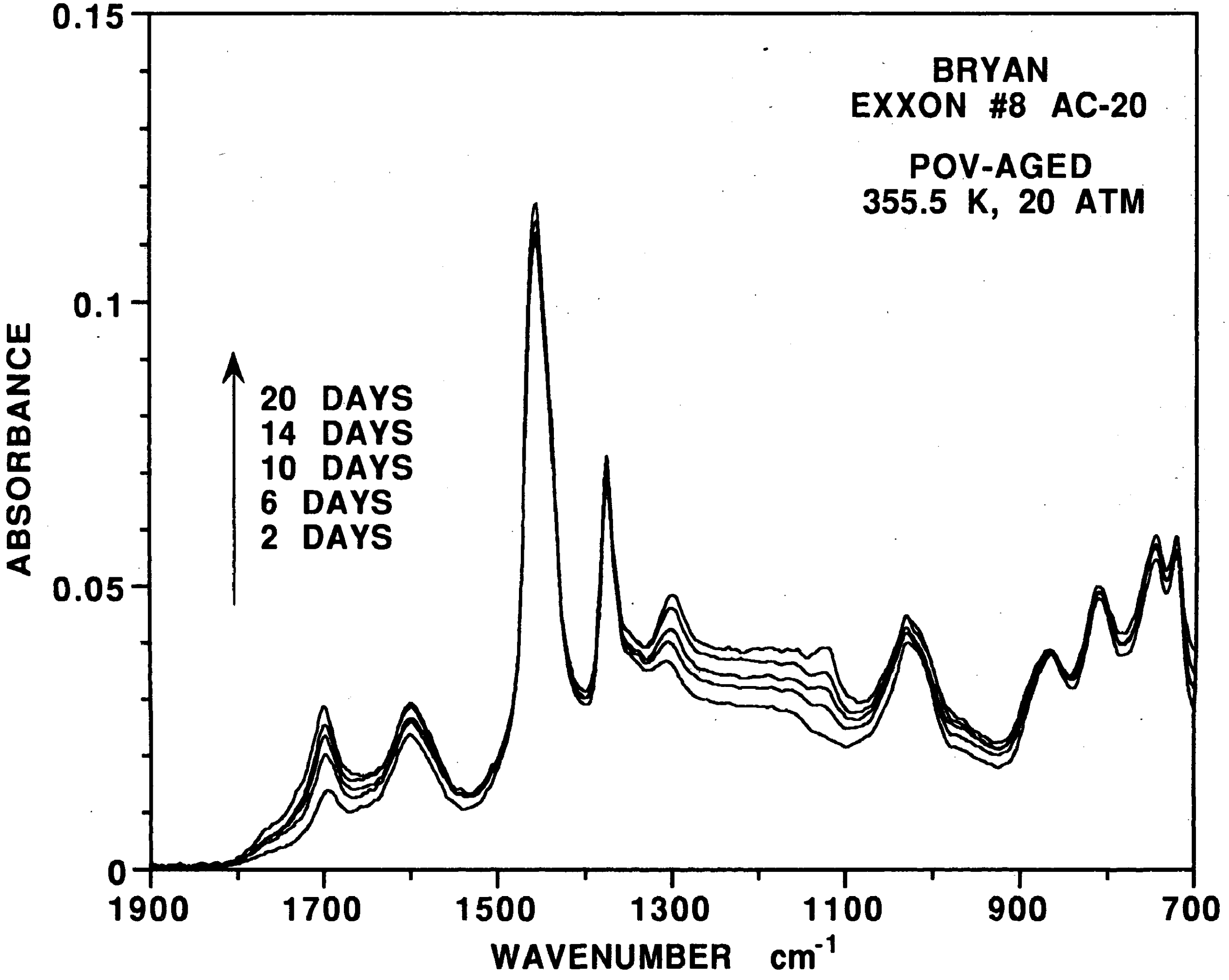

Figure E-67. IR spectra of neat and POV-aged Bryan Exxon #8 AC-20 at 344.4 K and 20 atm.

Table E-19. Kinetic Data for Dickens Cosden AC-10[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.675	–	–
2	–	–	1.240
4	0.846	0.973	1.423
6	0.899	–	1.567
8	–	1.124	1.674
9	1.017	–	–
10	–	–	1.810
12	–	1.300	2.058
14	1.112	–	2.066
16	–	1.385	2.209
20	–	1.439	2.284
22	–	–	2.383
24	–	1.614	–
28	1.334	–	–
35	1.379	–	–

[a] POV aged at 20 atm.

– Signifies values were not determined

Table E-20. Kinetic Data for Dickens Cosden AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.729	–	–
2	–	–	1.131
4	0.910	1.045	1.348
6	0.984	–	1.781
8	–	1.265	1.606
9	1.087	–	–
10	–	–	1.748
12	–	1.388	1.900
14	1.189	–	1.976
16	–	1.487	2.061
20	–	1.580	2.449
22	–	–	2.644
24	–	1.776	–
28	1.366	–	–
35	1.481	–	–

[a] POV aged at 20 atm.

– Signifies values were not determined

Table E-21. Kinetic Data for Dickens Diamond Shamrock AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.638	–	–
2	–	–	1.105
4	0.910	1.073	1.256
6	1.016	–	1.422
8	–	1.212	1.571
9	1.121	–	–
10	–	–	1.624
12	–	1.360	1.656
14	1.219	–	1.747
16	–	1.421	1.816
20	–	1.503	2.041
22	–	–	2.116
24	–	1.593	–
28	1.414	–	–
35	1.427	–	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-22. Kinetic Data for Dickens Dorchester AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.522	–	–
2	–	–	0.884
4	0.679	0.753	0.988
6	0.732	–	1.122
8	–	0.954	1.286
9	0.812	–	–
10	–	–	1.476
12	–	0.990	1.561
14	0.920	–	1.637
16	–	1.096	1.718
20	–	1.216	1.856
22	–	–	1.977
24	–	1.276	–
28	1.039	–	–
35	1.151	–	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-23. Kinetic Data for Dickens Exxon AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.764	–	–
2	–	–	1.272
4	1.002	1.112	1.556
6	1.042	–	1.368
8	–	1.294	1.899
9	1.145	–	–
10	–	–	1.960
12	–	1.477	2.005
14	1.282	–	2.132
16	–	1.557	2.197
20	–	1.657	2.314
22	–	–	2.479
24	–	1.791	–
28	1.517	–	–
35	1.575	–	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-24. Kinetic Data for Dickens MacMillan AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.621	–	–
2	–	–	0.989
4	0.722	0.842	1.119
6	0.785	–	1.188
8	–	0.957	1.269
9	0.830	–	–
10	–	–	1.370
12	–	1.025	1.452
14	0.890	–	1.472
16	–	1.110	1.544
20	–	1.180	1.642
22	–	–	1.726
24	–	1.234	–
28	1.040	–	–
35	1.124	–	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-25. Kinetic Data for Pineland Cosden AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.815	–	–
2	0.807	–	1.025
3	0.834	–	–
4	–	1.004	1.224
5	0.964	–	–
6	–	–	1.471
8	1.066	1.224	1.558
10	1.102	–	1.687
12	–	1.257	1.810
14	1.128	–	–
16	1.243	1.493	–
19	1.294	–	–
20	–	1.612	–
22	1.292	–	–
24	–	1.661	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-26. Kinetic Data for Pineland Dorchester AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.769	–	–
2	0.820	–	0.981
3	0.878	–	–
4	–	0.915	1.112
5	0.964	–	–
6	–	–	1.257
8	0.980	1.098	1.436
10	1.010	–	1.562
12	–	1.188	1.664
14	1.069		–
16	1.114	1.265	–
19	1.156	–	–
20	–	1.436	–
22	1.173	–	–
24	–	1.492	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-27. Kinetic Data for Pineland Exxon AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.776	–	–
2	0.839	–	1.105
3	0.943	–	–
4	–	1.067	1.213
5	1.088	–	–
6	–	–	1.348
8	1.096	1.287	1.555
10	1.146	–	1.696
12	–	1.356	1.802
14	1.305	–	–
16	1.300	1.563	–
19	1.322	–	–
20	–	1.671	–
22	1.352	–	–
24	–	1.722	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-28. Kinetic Data for Pineland MacMillan AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.750	–	–
2	0.816	–	0.996
3	0.849	–	–
4	–	0.909	1.064
5	0.905	–	–
6	–	–	1.195
8	0.953	1.091	1.339
10	1.014	–	1.497
12	–	1.194	1.562
14	1.046	–	–
16	1.063	1.279	–
19	1.140	–	–
20	–	1.357	–
22	1.161	–	–
24	–	1.418	–

[a] POV aged at 20 atm.
– Signifies values were not determined

Table E-29. Kinetic Data for Pineland Texaco AC-20[a]

Time	333.3 K *CA*	344.4 K *CA*	355.5 K *CA*
1	0.682	–	–
2	0.777	–	0.969
3	0.774	–	–
4	–	0.930	1.102
5	0.868	–	–
6	–	–	1.268
8	0.884	1.103	1.414
10	0.972	–	1.611
12	–	1.220	1.697
14	1.060	–	–
16	1.081	1.320	–
19	1.148	–	–
20	–	1.426	–
22	1.196	–	–
24	–	1.468	–

[a] POV aged at 20 atm.

– Signifies values were not determined

Table E-30. r_{CA} and CA_o of Dickens and Pineland Asphalts[a]

Location	Asphalts	$r_{CA} \times 10^2$			CA_o		
		333.3 CA/day	344.4 CA/day	355.5 CA/day	333.3 CA	344.4 CA	355.5 CA
Dickens							
	Cosden AC-10	1.426	3.024	6.348	0.904	0.882	1.173
	Cosden AC-20	1.457	3.358	6.672	0.968	0.953	1.301
	D.S.[b] AC-20	1.213	2.525	5.192	1.035	1.007	1.064
	Dorchester AC-20	1.197	2.505	6.068	0.723	0.697	0.783
	Exxon AC-20	1.644	3.261	6.378	1.026	1.025	1.217
	MacMillan AC-20	1.118	1.934	3.866	0.731	0.787	0.952
Pineland							
	Cosden AC-20	1.811	3.348	7.712	0.919	0.906	0.923
	Dorchester AC-20	1.457	2.842	7.065	0.867	0.835	0.841
	Exxon AC-20	1.847	3.308	7.340	0.980	0.981	0.939
	MacMillan AC-20	1.433	2.449	6.106	0.850	0.865	0.848
	Texaco AC-20	1.859	3.016	7.592	0.790	0.838	0.812

[a] POV aged at 20 atm.

[b] D.S. represents Diamond Shamrock

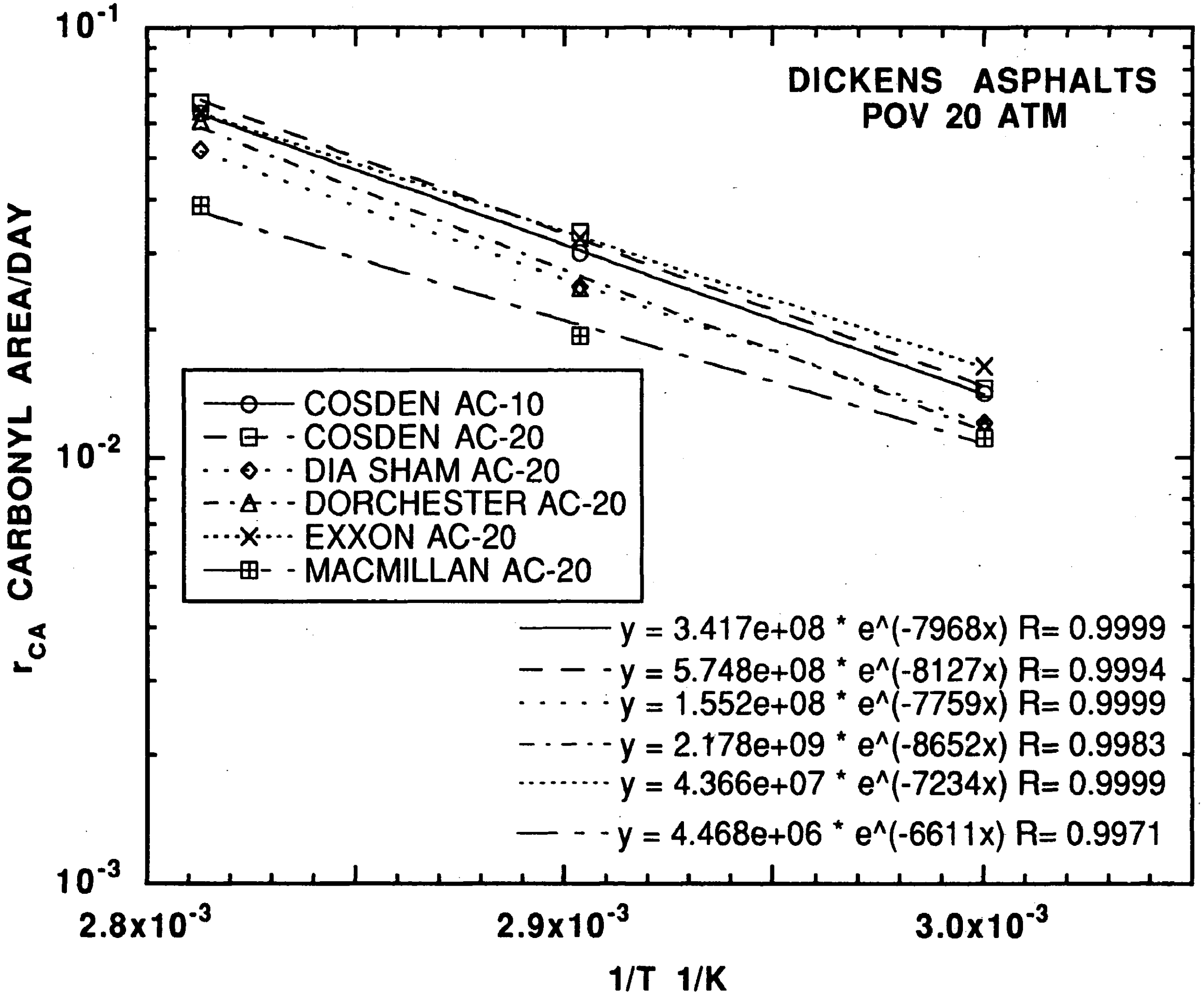

Figure E-68. r_{CA} versus $(1 / T)$ at 20 atm for all Dickens asphalts studied.

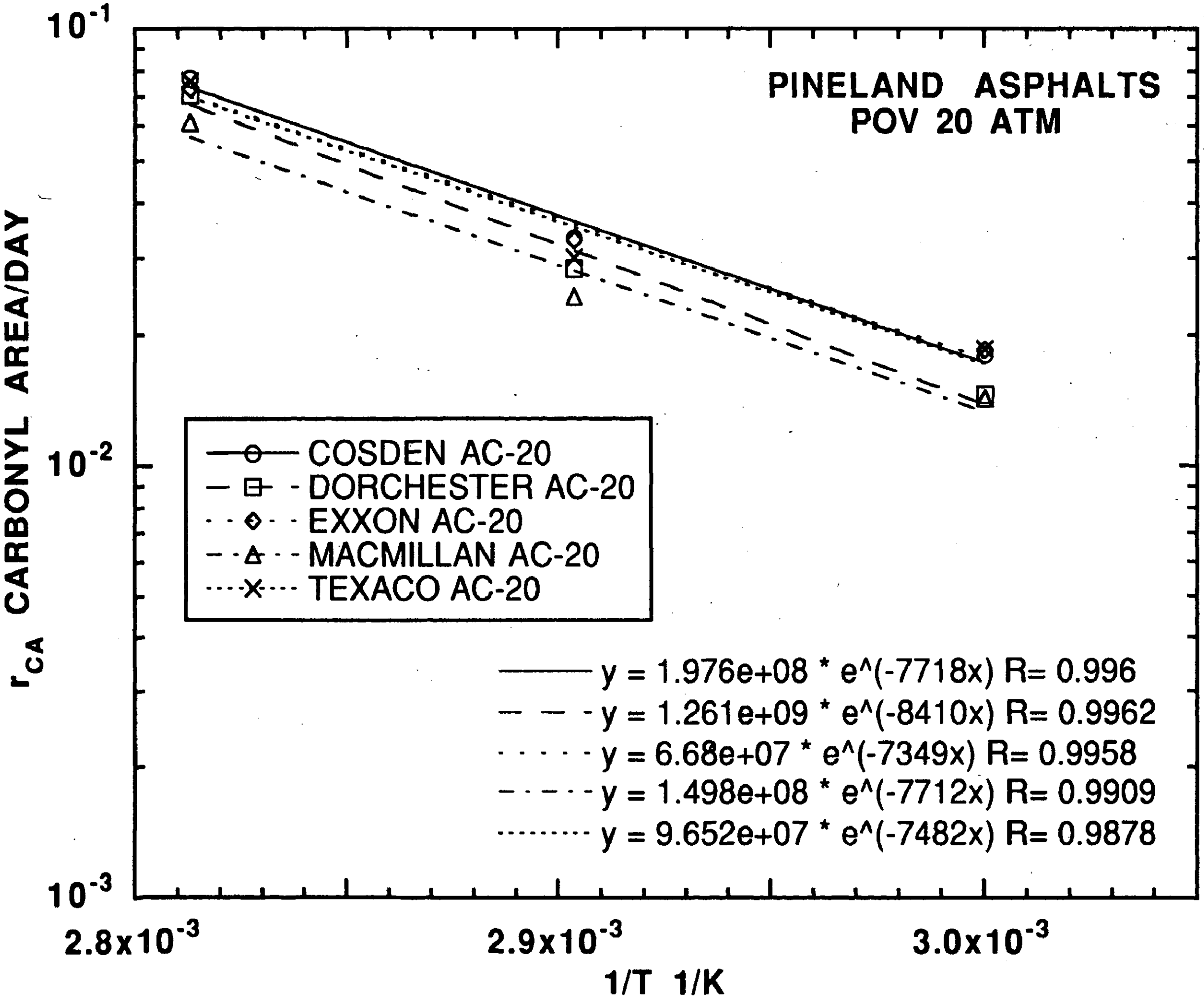

Figure E-69. r_{CA} versus $(1 / T)$ at 20 atm for all Pineland asphalts studied.

APPENDIX F

LABORATORY AND FIELD COMPARISON PROGRAMS

```
c*****************************************************************************
c
c                       Minimum Theoretical Time
c
c          This program determines the minimum theorectical time to reach
c     a measured core carbonyl content.  No diffusion resistances are
c     included.  The user must supply the regions climatic data.
c
c*****************************************************************************
c
      integer nmax
      parameter (nmax = 100, nmax2 = 200)
c
      integer i, j, n, numtemp, numas, numpres
      integer ii, iii, inc, nn, n2p1
      real*4 y(nmax), kp
      real*4 p_es, tol
      real*4 alpha, a, e_a, r2
      real*4 s, beta
      character*8 asphalt
      character*12 outputfile, datafile
      common /temper/ temp_mean, amp, omega
      common /kev/ a, e_a, r2
      common /tempdep/ kp
      common /constants/ alpha, s, beta
      common /blockc/ tol
      common /blocki/ p_es
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
c
      open(unit=8,file=datafile ,status='old')
      open(unit=9,file=outputfile ,status='unknown')
c
      read(8,*) numas
      read(8,*) p_es , tol, alpha
      read(8,*) temp_min, temp_max
      temp_mean = (temp_min + temp_max)/2.0
      amp = temp_mean - temp_min
      pi = 3.1415
      omega = 2.0*pi/365.0
      n = 1
      r2 = 8.314
c
      do 20 i = 1, numas , 1
           read(8,'(a)') asphalt
           read(8,*) a, e_a
           read(8,*) s, beta
           read(8,*) ca_fail
```

```
          call kevin (n, y, ca_fail, p_es, time_fail)
          write(9,1575) asphalt, ca_fail, time_fail
   20 continue
c
c***************************************************************************
c
 1575 format(' ',4x,'No Diffusion Resistance',
     +       /,5x,'asphalt =',a8,
     +       /,5x,'ca_fail =',f10.4,' K',
     +       /,5x,'time_fail =',f10.2,' days',/)
c
      stop
      end
c
c***************************************************************************
c
      subroutine kevin (n, y, ca_fail, p_es, time_fail)
c
      integer n, ido, istep, nflag
      real*4 t, y(n), param(50), tend
      real*4 tol, ca_fail, alpha, s, beta, time_fail
      common /blockc/ tol
      common /constants/ s, beta, alpha
c
      external sset, ivprk, equation1
      call sset(50, 0.0, param, 1)
      param(4) = 20000
c
      y(1) = s*(p_es**beta)
      ido = 1
      istep = 0
      nflag = 0
      t = 0.0
   20 continue
      if(t.lt.10.0) then
           istep = istep+1
           goto 3000
      endif
      istep = istep + 20
c
 3000 continue
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      call tempcal (t, temp)
      if(y(1).ge.ca_fail) then
          nflag = 1
      endif
      if(nflag.eq.0) then
           goto 20
      endif
c
```

```
 4000 continue
      ido = 3
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      time_fail = t
      return
      end
c
c*****************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n
      real*4 t, y(n), yprime(n)
      real*4 p_es, car
      common /blocki/ p_es
c
      call tempcal(t, temp)
      call rates(p_es, car)
      yprime(1) = car
      return
      end
c
c****************************************************************************
c
      subroutine rates (p_es, car)
c
      real*4 alpha, kp, car, t
      real*4 s, beta
      common /tempdep/ kp
      common /constants/ s, beta, alpha
c
      car = kp*(p_es**alpha)
      return
      end
c
c****************************************************************************
c
      subroutine tempcal (t, temp)
c
      real*4 t, temp, amp, temp_mean
      real*4 a, e_a, r2, c, r1, hs_o, m_o, gamma, delta
      real*4 temp_o, kp, conv, hs, m
      common /temper/ temp_mean, amp, omega
      common /kev/ a, e_a, r2
      common /tempdep/ kp
c
      temp = temp_mean + amp*cos(omega*t)
      kp = a*exp(-e_a/(r2*temp))
      return
      end
```

```
c*****************************************************************************
c
c                 Estimation of "L_ef" from field data
c
c
c          This program estimates "L_ef" by comparing values of average
c     carbonyl content with measured carbonyl from field aging.
c
c*****************************************************************************
c
      integer nmax
      parameter (nmax = 100, nmax2 = 200)
c
      integer i, j, n, numtemp, numas, numpres
      integer ii, iii, inc, nn, n2p1
      real*4 y(nmax), z(nmax2), length, kp, conv
      real*4 p_es, tol, delx, diff, fact
      real*4 alpha, a, e_a, temp, r1, r2, c
      real*4 adif, bdif, hs, m, s, beta, ca_i
      real*4 m_o, hs_o, temp_o, vis_fail
      character*8 asphalt
      character*12 outputfile, datafile, sumfile, avgfile
      common /temper/ temp_mean, amp, omega
      common /kev/ a, e_a, r2, c, r1, hs_o, m_o, gamma,
     +               delta, temp_o
      common /tempdep/ kp, conv, hs, m
      common /constants/ alpha, adif, bdif, s, beta
      common /blockc/ tol
      common /blockd/ inc, nn
      common /blockt/ length, delx
      common /blocki/ p_es, delxsq, fact_i
      common /pinel/ decay
c
      write(*,*) 'Enter your output file name'
      read(*,'(a)') outputfile
      write(*,*) 'Enter your data file name'
      read(*,'(a)') datafile
      write(*,*) 'Enter your summary file name'
      read(*,'(a)') sumfile
      write(*,*) 'Enter your average file name'
      read(*,'(a)') avgfile
c
      open(unit=8,file=datafile ,status='old')
      open(unit=9,file=outputfile ,status='unknown')
      open(unit=10,file=sumfile ,status='unknown')
      open(unit=11,file=avgfile , status='unknown')
c
      read(8,*) numas, p_es
      read(8,*) tol, n
      read(8,*) alpha, adif, bdif, temp_o
      read(8,*) temp_min, temp_max, time_max
```

```
c
      temp_mean = (temp_min + temp_max)/2.0
      amp = temp_mean - temp_min
      pi = 3.1415
      omega = 2.0*pi/365.0
      adif = adif*(3600.0)*(24.0)
      c = 6.56e-4
      r1 = 82.057
      r2 = 8.314
      n2p1 = (2*n) + 1
      iflag = 0
c
      do 20 i = 1, numas , 1
           read(8,'(a)') asphalt
           read(8,*) a, e_a
           read(8,*) s, beta, hs_o, m_o, gamma, delta
           read(8,*) length, decay
           delx = length/float(n)
           delxsq = delx**2
           inc = n/5
           nn = n - (inc/2)
           m_o = log(m_o)
           t = 0.0
           call propinit (length, t, diff_i, ca_i, temp)
           fact_i = (diff_i/delxsq)
           write(9,800) asphalt
           write(9,1200) tol, length, p_es, temp, delx
           write(9,1300) alpha, a, e_a, c, r1, r2, kp, conv
           write(9,1400) adif, bdif, hs_o, m_o, temp_o,
     +                gamma, delta, beta, s
           write(9,1500) temp, hs, m, diff_i, fact_i
           write(10,1600) asphalt, temp, p_es, diff_i
           ca_dif = ca_fail - ca_i
           call kevin (n, y, n2p1, z, ca_dif, iflag, p_es,
     +                  time_max)
           iflag = 1
   20 continue
c
c*****************************************************************************
c
  800 format(' ',/,5x,'asphalt = ',a8,/)
 1200 format(' ',4x,'tol = ',1pe11.4,' dimensionless',
     +     /,5x,'length = ',0pf10.4,' m',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'temp = ',f10.4,' K',
     +     /,5x,'delx = '1pe10.4,' m',/)
 1300 format(' ',4x,'alpha = ',f10.4,' dimensionless',
     +     /,5x,'a = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'e_a = ',0pf10.1,' J/mol',
     +     /,5x,'c  = ',1pe10.4,' mol/cm^3 CA',
     +     /,5x,'r1 = ',0pf10.4,' cm^3 atm/mol K',
```

```
     +     /,5x,'r2 = ',f10.4,' J/mol K',
     +     /,5x,'kp = ',1pe10.4,' CA/day atm^alpha',
     +     /,5x,'conv = ',e10.4,' atm/CA',/)
 1400 format(' ',4x,'adif = ',1pe10.4,' m^2/day',
     +     /,5x,'bdif = ',0pf10.4,' dimensionless',
     +     /,5x,'hs_o = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m_o  = ',f10.4,' ln(vis)',
     +     /,5x,'temp_o = ',f10.1,' K',
     +     /,5x,'gamma = ',f10.1,' K/CA',
     +     /,5x,'delta = ',f10.1,' K',
     +     /,5x,'beta = ',f10.4,' dimensionless',
     +     /,5x,'s = ',f10.4,' CA/atm^beta',/)
 1450 format(' ',4x,'vis_fail = ',1pe10.4,' P',
     +     /,5x,'ca_fail = ',0pf10.4,' CA',
     +     /,5x,'ca_o = ',f10.4,' CA',
     +     /,5x,'ca_dif  = ',f10.4,' CA',
     +     /,5x,'rate = ',1pe10.4,' CA/day',
     +     /,5x,'time_fail = ',0pf10.1,' days',/)
 1475 format(' ',4x,'No Diffusion Resistance',
     +        /,5x,'asphalt =',a8,
     +        /,5x,'temp =',f10.1,' K',
     +        /,5x,'time_fail =',f10.2,' years',/)
 1500 format(' ',4x,'temp = ',f10.1,' K',
     +     /,5x,'hs = ',f10.4,' ln(vis)/CA',
     +     /,5x,'m  = ',f10.4,' ln(vis)',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',
     +     /,5x,'fact_i = ',1pe10.4,' 1/day',/)
 1575 format(' ',4x,'With Diffusion Resistance',
     +        /,5x,'asphalt =',a8,
     +        /,5x,'temp =',f10.1,' K',
     +        /,5x,'time_fail =',f10.2,' years',/)
 1600 format(' ',4x,'asphalt = ',a8,
     +     /,5x,'temp = ',f10.1,' K',
     +     /,5x,'p_es = ',f10.4,' atm',
     +     /,5x,'diff_i = ',1pe10.4,' m^2/day',/)
c
      stop
      end
c
c*************************************************************************
c
      subroutine propinit (length, t, diff_i, ca_i, temp)
c
      real*4 length, m, p_es, fact_i, t, temp
      real*4 kp, conv, alpha, hs, adif, bdif, s, beta
      real*4 p_i, ca_i, eta_i, diff_i, delxsq
      common /tempdep/ kp, conv, hs, m
      common /constants/ alpha, adif, bdif, s, beta
      common /blocki/ p_es, delxsq, fact_i
c
      call tempcal (t, temp)
```

```
      p_i = (p_es)*0.5
      ca_i = s*(p_i**beta)
      eta_i = exp(hs*ca_i + m)
      diff_i = adif*(eta_i**bdif)
      return
      end
c
c***************************************************************************
c
      subroutine kevin (n, y, n2p1, z, ca_dif, iflag, p_es,
     +                  time_max)
c
      integer n, n2, ido, istep, iflag, nflag, n2p1
      real*4 t, y(n), z(n2p1), param(50), tend
      real*4 tol, ca_dif, time_max
      common /blockc/ tol
c
      external sset, ivprk, equation1, equation2
      call sset(50, 0.0, param, 1)
      param(4) = 20000
      timei = 1.0
c
      do 100 j = 1, n, 1
           y(j) = 0.0
  100 continue
      ido = 1
      istep = 0
      nflag = 0
c
   10 continue
      istep = istep+1
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      call output(y, n, t, iflag, timei, nflag)
      if(nflag.eq.0) then
           goto 10
      endif
      ido = 3
      tend = float(istep)
      call ivprk(ido, n, equation1, t, tend, tol, param, y)
      ido = 1
      istep = 1
      nflag = 0
      t = 1.0
      call tempcal (t, temp)
      call carbonyl(p_es, t, ca_es)
      do 200 j = 1, n, 1
           z(j) = y(j)
           z(n+j) = ca_es
  200 continue
      z(n2p1) = ca_es
```

```
      call trap(z, n2p1, n, ca_avg)
      call output2(z, n2p1, t, temp, ca_avg)
c
   20 continue
      if(t.lt.10.0) then
           istep = istep+1
           goto 3000
      endif
      istep = istep + 20
 3000 continue
      tend = float(istep)
      call ivprk(ido, n2p1, equation2, t, tend, tol, param, z)
      call trap(z, n2p1, n, ca_avg)
      call tempcal(t, temp)
c
      if(t.ge.time_max)then
           nflag = 1
      endif
      call output2 (z, n2p1, t, temp, ca_avg)
      if(nflag.eq.0) then
           goto 20
      endif
      ido = 3
      tend = float(istep)
      call ivprk(ido, n2p1, equation2, t, tend, tol, param, z)
      return
      end
c
c*****************************************************************************
c
      subroutine equation1 (n, t, y, yprime)
c
      integer n, nm1, j, jp1, jm1
      real*4 t, y(n), yprime(n), term2, yima, p_es0
      real*4 p_es, delxsq, car, p, ocr, fact_i
      common /blocki/ p_es, delxsq, fact_i
c
      call tempcal (t, temp)
      do 20 j = 1, n, 1
           if(y(j).lt.0.0) then
           y(j) = 0.0
      endif
   20 continue
      yima = y(2)
c
c                 Calculate the 0 flux boundary condition
c

      p = y(1)
      call rates(p, ocr, car)
      term2 = (fact_i)*(y(2) - 2.0*y(1) + yima)
```

```
      yprime(1) = term2 - ocr
c
c           Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = y(j)
           call rates(p, ocr, car)
           term2 = (fact_i)*(y(jp1) - 2.0*y(j) + y(jm1))
           yprime(j) = term2 - ocr
   30 continue
c
c      Calculate the known pressure at the free surface
c
      if(t.eq.0.0) then
           p_es0 =(p_es-y(1))/2.0
           p = y(n)
           call rates(p, ocr, car)
           term2 = (fact_i)*(p_es0 - 2.0*y(n) + y(nm1))
           yprime(n) = term2 - ocr
           goto 100
      endif
      p = y(n)
      call rates(p, ocr, car)
      term2 = (fact_i)*(p_es - 2.0*y(n) + y(nm1))
      yprime(n) = term2 - ocr
c
  100 continue
      return
      end
c
c***************************************************************************
c
      subroutine equation2 (n2p1, t, z, zprime)
c
      integer n, nm1, n2p1, j, jp1, jm1
      real*4 t, z(n2p1), zprime(n2p1), term2, yima
      real*4 delx, delxsq, term1, ca, diff, diffp, p, car
      real*4 p_es, fact_i
      common /blocki/ p_es, delxsq, fact_i
c
      call tempcal (t, temp)
      n = (n2p1-1)/2
      do 20 j = 1, n, 1
           if(z(j).lt.0.0) then
           z(j) = 0.0
      endif
   20 continue
      zima = z(2)
```

```
c
c                 Calculate the 0 flux boundary condition
c
      p = z(1)
      ca = z(n+1)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
c
      fact = (diff/delxsq)
      term2 = fact*(z(2) - 2*z(1) + zima)
      zprime(1) = term2 - ocr
      zprime(n+1) = car
c
c          Calculate the derivatives at all of the interior nodes
c
      nm1 = n - 1
      do 30 j = 2, nm1, 1
           jp1 = j + 1
           jm1 = j - 1
           p = z(j)
           ca = z(n+j)
           call rates(p, ocr, car)
           call diffus(p, ca, diff, diffp)
           fact = (diff/delxsq)
           term1 = (diffp/(4.0*delxsq))*((z(jp1) - z(jm1))**2)
           term2 = fact*(z(jp1) - 2.0*z(j) + z(jm1))
           zprime(j) = term1 + term2 - ocr
           zprime(n+j) = car
   30 continue
c
c           Calculate the known pressure at the free surface
c
      call press (t, p_es, p_est)
      p = z(n)
      ca = z(n+n)
      call rates(p, ocr, car)
      call diffus(p, ca, diff, diffp)
      fact = (diff/delxsq)
      term1 = (diffp/(4.0*delxsq))*((p_est - z(nm1))**2)
      term2 = fact*(p_est - 2.0*z(n) + z(nm1))
      zprime(n) = term1 + term2 - ocr
      zprime(n+n) = car
      call rates (p_est, ocr, car)
      zprime(n2p1) = car
      return
      end
c
c***************************************************************************
c
      subroutine diffus (p, ca, diff, diffp)
c
```

```
      real*4 p, ca
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /tempdep/ kp, conv, hs, m
      common /constants/ alpha, adif, bdif, s, beta
c
      eta = exp(hs*ca+m)
      diff = adif*(eta**bdif)
c
      if(p.eq.0) then
           cap = 0.0
           goto 100
      endif
      cap = alpha*ca/p
      if(cap.gt.1.0) then
           cap = 1.0
           goto 100
      endif
c
  100 continue
      diffp = bdif*hs*diff*cap
      return
      end
c
c***********************************************************************
c
      subroutine rates (p, ocr, car)
      real*4 alpha, kp, conv, ocr, car, t
      real*4 hs, m, adif, bdif, s, beta
      common /tempdep/ kp, conv, hs, m
      common /constants/ alpha, adif, bdif, s, beta
c
      car = kp*(p**alpha)
      ocr = conv*car
      return
      end
c
c***********************************************************************
c
      subroutine carbonyl (p, t, ca_i)
c
      real*4 p, t, ca_i
      real*4 hs, m, adif, bdif, s, beta, alpha, kp, conv
      common /tempdep/ kp, conv, hs, m
      common /constants/ alpha, adif, bdif, s, beta
c
      ca_i = t*kp*(p**alpha) + s*(p**beta)
      return
      end
c
c***********************************************************************
c
```

```
      subroutine trap(z, n2p1, n, ca_avg)
c
      integer n2, n, n2p1
      real*4 z(n2p1), ca_es, ca_avg, length
      common /blockt/ length, delx
c
      sum = 0.0
      do 10 j = n+2, 2*n, 1
           sum = sum + z(j)
   10 continue
      area = (delx/2.0)*(z(n+1) + 2.0*sum + z(n2p1))
      ca_avg = area/length
      return
      end
c
c***********************************************************************
c
      subroutine output (y, n, t, iflag, timei, nflag)
c
      integer n, ido, istep, iflag, nflag
      integer inc,  nn
      real*4 t, y(n), tend, timei, tm1
      real*4 p_es, delxsq, fact_i
      common /blocki/ p_es, delxsq, fact_i
      common /blockd/ inc, nn
c
      do 10 j = 1, n, 1
           if(y(j).lt.0.0) then
                y(j) = 0.0
           endif
   10 continue
c
      if(iflag.eq.0) then
           if(t.ge.timei) then
                nflag = 1
           endif
           return
      endif
      tm1 = t - 1
      if(tm1.eq.0.0) then
           return
      endif
      if(tm1.ge.timei) then
      nflag = 1
      endif
c
 2000 format(' ',2x,f6.0,' , ',5(f6.3,' , '),f6.3)
      return
      end
c
c***********************************************************************
```

```
c
      subroutine output2 (z, n2p1, t, temp, ca_avg)
c
      integer n, nflag, inc, nn, n2p1
      real*4 t, z(n2p1), tend, timemax
      real*4 p_es, delxsq, fact_i, ca_avg
      real*4 a, e_a, r2, c, r1, hs_o, m_o, gamma,
     +       delta, temp_o
      common /blocki/ p_es, delxsq, fact_i
      common /blockd/ inc, nn
      common /kev/ a, e_a, r2, c, r1, hs_o, m_o, gamma,
     +              delta, temp_o
c
      n = (n2p1 - 1)/2
      call tempcal(t, temp)
c
      do 10 j = 1, n, 1
          if(z(j).lt.0.0) then
               z(j) = 0.0
          endif
   10 continue
c
      eta_avg = exp(hs_o*ca_avg + m_o)
      write(9,2000) t, temp, (z(j), j=1,nn,inc),  p_es
      write(10,2000) t, temp, (z(j+n), j = 1,nn,inc), z(n2p1)
      write(11,2010) t, ca_avg, eta_avg
c
 2000 format(' ',2x,f6.0,' , ',f6.0,' , ',5(f6.3,' , '),f6.3)
 2010 format(' ',2x,f6.0,' , ',f6.3,' , ',1pe10.3)
      return
      end
c
c*****************************************************************************
c
      subroutine tempcal (t, temp)
c
      real*4 t, temp, amp, temp_mean
      real*4 a, e_a, r2, c, r1, hs_o, m_o, gamma, delta
      real*4 temp_o, kp, conv, hs, m
      common /temper/ temp_mean, amp, omega
      common /kev/ a, e_a, r2, c, r1, hs_o, m_o, gamma,
     +              delta, temp_o
      common /tempdep/ kp, conv, hs, m
c
      temp = temp_mean + amp*cos(omega*t)
      kp = a*exp(-e_a/(r2*temp))
      conv = c*r1*temp
      hs = hs_o + gamma*((1.0/temp) - (1.0/temp_o))
      m = m_o + delta*((1.0/temp) - (1.0/temp_o))
      return
      end
```

```
c
c****************************************************************************
c
      subroutine press (t, p_es, p_est)
c
      real*4 t, p_es, p_est
      common /pinel/ decay
c
      p_est = p_es*exp(-decay*t)
      return
      end
```

APPENDIX G

SHORT-TERM AGING DATA

Table G-1. Short-Term POV Aging Data for Lau *et al.*, (1992) Ampet AC-20[a]

T K	t days	CA	η_0^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2	% A^d
RTFO	–	0.581	6.75	59.9	22.84
322.2	1	0.590	4.07	12.5	21.04
	2	0.553	5.06	42.4	20.58
	3	0.558	5.40	44.8	21.43
	4	0.556	5.90	53.0	21.40
	5	0.529	6.30	54.6	21.61
333.3	1	0.537	5.62	49.8	20.22
	2	0.576	7.13	62.4	21.02
	3	0.680	8.50	71.7	23.44
	4	0.962	9.70	84.0	24.71
	5	0.963	9.56	81.7	24.49
344.4	1	0.725	10.4	76.2	23.08
	2	0.872	13.9	116.0	25.35
	3	1.062	17.6	135.0	27.14
	4	1.046	22.6	181.0	28.95
	5	0.915	18.8	154.0	27.10
355.5	1	0.744	13.1	109.0	25.49
	2	1.242	18.7	156.0	27.73
	3	1.240	25.5	205.0	30.44
	4	1.268	38.5	296.0	33.72
	5	1.359	39.0	297.0	32.01
366.7	1	1.282	24.0	200.0	29.69
	2	1.066	32.0	259.0	31.38
	3	2.079	627.0	2670.0	41.70
	4	2.680	17900.0	18200.0	53.82
	5	1.798	142.0	903.0	38.43

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-2. Short-Term POV Aging Data for Lau *et al.*, (1992) Coastal AC-20[a]

T K	t days	CA	$\eta_o^{*\,b}$ kP	$(1\ /\ J'')^c \times 10^{-3}$ dyne / cm^2	% A^d
RTFO	–	0.665	8.80	73.1	27.90
322.2	1	0.604	4.58	41.5	26.42
	2	0.570	5.30	47.2	27.30
	3	0.610	6.30	55.6	27.52
	4	0.680	6.75	59.7	28.12
	5	0.641	7.20	62.6	29.36
333.3	1	0.545	5.95	52.6	28.28
	2	0.586	8.55	72.8	29.83
	3	0.677	11.2	83.6	30.17
	4	0.686	12.8	102.0	30.94
	5	0.703	12.9	99.7	31.21
344.4	1	0.826	13.7	105.0	30.79
	2	0.816	21.8	160.0	32.90
	3	0.932	30.3	213.0	34.34
	4	0.987	40.0	268.0	33.48
	5	0.886	34.0	236.0	34.39
355.5	1	0.889	23.4	171.0	32.86
	2	0.938	30.0	212.0	35.70
	3	1.075	60.0	383.0	37.38
	4	1.180	96.5	529.0	37.15
	5	1.220	120.0	612.0	39.88
366.7	1	1.127	71.5	420.0	39.84
	2	1.368	110.0	544.0	39.32
	3	2.374	39400.0	14740.0	50.75
	4	3.553	–	34870000.0	58.07
	5	2.282	7680.0	7350.0	47.67

---

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-3. Short-Term POV Aging Data for Lau *et al.*, (1992) Cosden AC-20[a]

T K	t days	CA	$\eta_0^{*\,b}$ kP	$(1\,/\,J'')^c \times 10^{-3}$ dyne / cm^2	% A^d
RTFO	–	0.637	7.00	64.3	22.16
322.2	1	0.484	4.15	40.2	22.34
	2	0.682	5.80	54.0	22.91
	3	0.710	7.05	61.1	24.68
	4	0.565	8.30	74.8	24.36
	5	0.583	7.37	68.6	24.16
333.3	1	0.651	6.77	64.4	24.27
	2	0.723	8.70	80.6	25.54
	3	0.817	14.4	122.0	27.36
	4	0.826	13.6	122.0	26.70
	5	0.834	14.2	116.0	26.05
344.4	1	0.921	12.4	110.0	27.32
	2	1.040	28.0	227.0	29.15
	3	1.165	36.1	286.0	30.88
	4	1.430	51.9	390.0	30.64
	5	1.242	36.5	284.0	29.74
355.5	1	1.059	26.8	219.0	28.98
	2	1.193	40.4	312.0	30.72
	3	1.518	75.0	521.0	32.46
	4	1.568	121.0	788.0	35.49
	5	1.705	166.0	973.0	34.67
366.7	1	1.438	79.4	543.0	32.88
	2	1.659	130.0	774.0	34.28
	3	2.353	3260.0	6320.0	37.14
	4	2.675	34000.0	21100.0	56.53
	5	2.342	2700.0	6080.0	48.45

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-4. Short-Term POV Aging Data for Lau *et al.*, (1992) Exxon AC-20[a]

T K	*t* days	*CA*	η_o^*[b] kP	$(1 / J'')$[c] $\times 10^{-3}$ dyne / cm^2	% *A*[d]
RTFO	–	0.523	6.48	62.2	17.44
322.2	1	0.469	4.43	42.8	17.87
	2	0.411	5.24	50.2	17.28
	3	0.502	6.32	55.7	18.24
	4	0.541	6.90	64.6	24.70
	5	0.529	7.20	68.9	17.50
333.3	1	0.649	5.50	53.6	16.59
	2	0.688	8.19	78.1	17.96
	3	0.766	10.1	96.7	18.87
	4	0.778	11.8	107.0	19.20
	5	0.865	12.3	111.0	19.48
344.4	1	0.799	11.1	104.0	19.39
	2	0.933	17.4	154.0	20.85
	3	1.005	22.0	193.0	22.57
	4	1.190	25.0	216.0	22.59
	5	1.235	25.4	220.0	22.59
355.5	1	0.989	18.3	161.0	20.89
	2	1.044	23.0	206.0	23.14
	3	1.268	43.5	310.0	24.74
	4	1.441	56.5	428.0	25.95
	5	1.713	59.5	566.0	27.74
366.7	1	1.233	36.8	308.0	25.29
	2	1.707	57.0	442.0	21.97
	3	1.972	346.0	1790.0	34.31
	4	2.331	2040.0	5070.0	37.67
	5	1.965	55100.0	2110.0	34.38

[a] POV aged at 20 atm.
– Signifies the values were not determined
[b] Measured at 333.3 K
[c] Measured at 333.3 K and 10 rad/s
[d] Hexane

Table G-5. Short-Term POV Aging Data for Lau *et al.*, (1992) Texaco AC-20[a]

T K	t days	CA	η_o^{*}[b] kP	$(1 / J'')$[c] $\times 10^{-3}$ dyne / cm^2	% A[d]
RTFO	–	0.595	7.63	63.3	23.39
322.2	1	0.546	3.87	35.0	22.44
	2	0.500	5.13	43.3	23.06
	3	0.553	5.60	47.4	20.51
	4	0.525	6.20	52.3	23.18
	5	0.554	6.51	54.9	23.72
333.3	1	0.547	5.29	46.1	21.64
	2	0.601	7.50	63.4	24.58
	3	0.594	9.05	72.8	24.57
	4	0.604	10.9	81.3	21.37
	5	0.642	11.4	84.4	25.50
344.4	1	0.634	9.64	78.5	24.81
	2	0.753	17.5	129.0	27.71
	3	0.843	23.6	160.0	27.43
	4	0.984	29.0	194.0	28.54
	5	1.142	30.6	193.0	26.08
355.5	1	0.686	17.0	124.0	27.05
	2	0.937	23.0	164.0	28.68
	3	0.978	43.5	267.0	30.10
	4	1.062	56.5	323.0	31.26
	5	1.127	59.5	321.0	32.04
366.7	1	1.324	48.5	310.0	29.89
	2	1.284	109.0	509.0	32.14
	3	1.929	2760.0	3440.0	44.57
	4	2.426	50100.0	11430.0	52.10
	5	2.176	5260.0	4790.0	46.10

---

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-6. Short-Term POV Aging Data for Jemison *et al.*, (1992b) Coastal AC-20[a]

T K	t days	CA	η_0^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
322.2	1	0.543	5.20	43.0
	2	0.622	7.10	59.3
	3	0.606	8.50	68.2
	4	0.681	9.50	72.0
	5	0.670	11.0	82.9
333.3	1	0.568	7.18	57.3
	2	0.603	13.6	94.8
	3	0.685	18.2	122.0
	4	0.632	21.0	137.0
	5	0.710	25.6	158.0
344.4	1	0.651	18.0	116.0
	2	0.811	48.0	248.0
	3	0.900	60.0	296.0
	4	0.956	81.1	371.0
	5	0.931	84.0	380.0
355.5	1	0.918	55.0	271.0
	2	1.020	77.0	367.0
	3	1.131	165.0	629.0
	4	1.166	197.0	697.0
	5	1.277	260.0	808.0
366.7	1	1.098	64.0	656.0
	2	1.343	1020.0	1520.0
	3	1.851	37100.0	8870.0
	4	2.329	-	20300.0
	5	2.395	814000.0	21400.0

---

[a] POV aged at 20 atm.

- Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-7. Short-Term POV Aging Data for Jemison *et al.*, (1992b) Fina AC-20[a]

T K	t days	CA	η_0^*[b] kP	$(1 / J'')$[c] × 10^{-3} dyne / cm^2
322.2	1	0.627	5.44	52.0
	2	0.693	7.80	74.6
	3	0.789	9.35	84.3
	4	0.691	10.7	92.9
	5	0.684	11.5	102.0
333.3	1	0.665	7.15	68.6
	2	0.791	13.7	124.0
	3	0.907	18.5	163.0
	4	0.845	21.5	188.0
	5	0.873	24.4	209.0
344.4	1	0.766	15.2	136.0
	2	1.138	36.8	317.0
	3	1.275	48.7	401.0
	4	1.136	63.4	516.0
	5	1.479	70.0	557.0
355.5	1	1.151	43.0	359.0
	2	1.310	71.0	555.0
	3	1.448	121.0	874.0
	4	1.526	133.0	860.0
	5	1.786	210.0	1380.0
366.7	1	1.672	164.0	1110.0
	2	1.981	377.0	1950.0
	3	2.412	3410.0	8300.0
	4	2.640	4600.0	13700.0
	5	2.805	13700.0	18100.0

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-8. Short-Term POV Aging Data for Jemison *et al.*, (1992b) Texaco AC-20[a]

T K	t days	CA	η_0^{*}[b] kP	$(1\ /\ J'')^{c} \times 10^{-3}$ dyne / cm^2
322.2	1	0.519	4.93	71.2
	2	0.542	6.22	53.8
	3	0.556	6.34	59.7
	4	0.533	8.13	65.4
	5	0.552	8.72	42.7
333.3	1	0.489	5.75	50.7
	2	0.556	10.1	106.0
	3	0.590	11.9	91.2
	4	0.663	13.5	84.4
	5	0.625	15.4	119.0
344.4	1	0.569	9.96	82.5
	2	0.876	22.9	155.0
	3	0.933	29.4	199.0
	4	0.953	37.0	243.0
	5	1.036	43.5	273.0
355.5	1	0.732	28.0	189.0
	2	0.911	36.2	240.0
	3	1.071	67.0	407.0
	4	1.157	78.0	463.0
	5	1.127	74.0	451.0
366.7	1	1.139	120.0	550.0
	2	1.354	308.0	970.0
	3	1.923	3490.0	4200.0
	4	2.555	–	9940.0
	5	2.464	81000.0	19300.0

---

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-9. Short-Term POV Aging Data for Davison *et al.*, (1989) Dickens Diamond Shamrock AC-20[a]

T K	t days	CA	η_o^* [b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
RTFO	–	–	9.70	75.8
322.2	1	0.500	4.50	40.4
	2	0.501	5.90	51.1
	3	0.604	6.76	59.0
	4	0.612	7.70	64.8
	5	0.692	8.39	69.4
333.3	1	0.601	5.86	51.2
	2	0.692	9.44	74.1
	3	0.845	11.9	93.4
	4	0.868	10.9	99.0
	5	0.838	14.5	111.0
344.4	1	0.817	11.0	86.4
	2	0.941	20.7	143.0
	3	1.096	24.1	162.0
	4	1.159	33.5	209.0
	5	1.331	32.9	204.0
355.5	1	1.053	23.6	154.0
	2	1.014	29.0	186.0
	3	1.240	42.3	252.0
	4	1.342	58.0	321.0
	5	1.345	–	376.0
366.7	1	1.460	66.0	326.0
	2	1.542	136.0	–
	3	1.717	448.0	–
	4	2.027	840.0	–
	5	1.930	948.0	–

---

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-10. Short-Term POV Aging Data for SHRP (1990) AAA-1[a]

T K	t days	CA	η_0^{*} [b] kP	$(1 / J'')^{c} \times 10^{-3}$ dyne / cm^2	% A[d]
322.2	1	0.534	1.32	12.7	21.82
	2	0.593	1.66	15.9	21.75
	3	0.641	1.86	17.8	22.33
	4	0.577	1.99	19.0	22.61
	5	0.644	2.10	19.9	23.13
333.3	1	0.536	1.46	14.3	22.47
	2	0.655	2.40	22.7	23.43
	3	0.676	2.64	24.4	23.92
	4	0.658	2.50	23.0	24.54
	5	0.698	3.70	30.8	25.18
344.4	1	0.670	2.33	21.6	23.33
	2	0.816	6.46	54.4	28.69
	3	0.884	9.35	72.9	29.20
	4	0.962	8.99	69.4	29.76
	5	0.954	14.1	102.0	30.84
355.5	1	0.693	4.05	70.8	23.33
	2	0.874	10.8	72.3	28.69
	3	1.098	30.0	189.0	29.20
	4	1.274	64.0	348.0	29.76
	5	1.111	27.5	161.0	30.84
366.7	1	1.329	74.0	385.0	34.42
	2	1.809	598.0	1324.0	39.92
	3	2.361	31100	16300.0	45.72
	4	2.936	-	21080.0	50.40
	5	3.394		55320.0	50.72

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-11. Short-Term POV Aging Data for SHRP (1990) AAC-1[a]

T K	t days	CA	η_0^*[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
322.2	1	0.487	1.57	15.5
	2	0.586	2.16	21.1
	3	0.624	2.50	24.6
	4	0.674	2.83	27.5
	5	0.699	3.17	30.5
333.3	1	0.535	1.77	17.5
	2	0.730	3.25	31.3
	3	0.796	3.95	36.9
	4	0.833	4.30	40.3
	5	0.891	5.05	46.5
344.4	1	0.704	2.80	26.9
	2	1.081	6.95	61.9
	3	1.161	8.85	74.6
	4	1.114	11.0	91.8
	5	1.227	13.3	106.0
355.5	1	1.114	9.56	81.5
	2	1.081	9.40	70.7
	3	1.304	16.1	127.0
	4	1.428	29.3	205.0
	5	1.416	21.3	162.0
366.7	1	1.462	29.0	211.0
	2	1.694	66.3	360.0
	3	1.917	188.0	736.0
	4	2.051	475.0	1390.0
	5	2.186	857.0	2050.0

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-12. Short-Term POV Aging Data for SHRP (1990) AAD-1[a]

T K	t days	CA	η_0^*[b] kP	$(1 / J'')^c \times 10^{-3}$ dyne / cm^2
322.2	1	0.754	1.91	17.6
	2	0.804	2.63	23.7
	3	0.843	2.97	25.4
	4	0.909	3.26	28.8
	5	0.918	3.53	30.6
333.3	1	0.853	2.20	20.4
	2	0.951	4.50	39.0
	3	1.008	5.25	44.4
	4	0.968	6.17	42.8
	5	1.053	7.00	51.4
344.4	1	0.929	3.25	28.1
	2	1.245	11.8	81.7
	3	1.249	17.0	105.0
	4	1.425	22.5	148.0
	5	1.480	30.0	171.0
355.5	1	1.070	19.0	121.0
	2	1.321	24.0	158.0
	3	1.666	63.5	351.0
	4	1.723	105.0	555.0
	5	1.628	68.8	361.0
366.7	1	1.823	240.0	871.0
	2	2.358	2700.0	4030.0
	3	3.384	12900.0	31100.0
	4	3.957	429000	66400.0
	5	5.393		–

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

Table G-13. Short-Term POV Aging Data for SHRP (1990) AAG-1[a]

T K	t days	CA	η_0^* [b] kP	$(1 / J'')$ [c] × 10^{-3} dyne / cm^2	% A [d]
322.2	1	0.838	3.56	35.0	10.99
	2	0.932	5.64	56.1	11.83
	3	1.017	6.35	62.6	12.06
	4	1.106	7.17	72.4	12.61
	5	1.162	8.02	80.1	12.97
333.3	1	0.738	3.75	37.7	11.33
	2	1.298	7.00	70.4	12.99
	3	1.516	8.00	77.9	13.68
	4	1.569	9.10	90.2	14.15
	5	1.694	10.2	102.0	14.74
344.4	1	1.023	5.07	49.6	11.84
	2	2.140	11.4	111.0	15.49
	3	2.216	13.2	130.0	16.22
	4	2.276	14.8	141.0	17.68
	5	2.404	16.5	159.0	17.73
355.5	1	2.001	12.6	120.0	15.80
	2	2.122	13.2	129.0	16.14
	3	2.384	18.5	180.0	18.34
	4	2.772	24.5	233.0	20.10
	5	2.534	24.0	226.0	19.87
366.7	1	2.687	20.2	198.0	20.07
	2	3.012	32.0	296.0	22.43
	3	3.099	46.0	406.0	25.26
	4	3.440	73.0	621.0	27.08
	5	3.675	96.0	787.0	29.15

---

[a] POV aged at 20 atm.

– Signifies the values were not determined

[b] Measured at 333.3 K

[c] Measured at 333.3 K and 10 rad/s

[d] Hexane

Table G-14. Short-Term POV Aging Data for SHRP (1990) AAK-2[a]

T K	t days	CA	η_0^* [b] kP	$(1\,/\,J'')^c \times 10^{-3}$ dyne / cm^2
322.2	1	0.710	1.80	16.4
	2	0.719	2.28	20.3
	3	0.784	2.48	21.8
	4	0.823	2.71	23.6
	5	0.751	2.89	25.0
333.3	1	0.778	1.84	17.1
	2	0.847	3.20	28.2
	3	0.861	3.62	31.4
	4	0.873	4.20	35.1
	5	0.906	4.43	37.9
344.4	1	0.732	2.30	20.6
	2	0.996	6.90	54.2
	3	1.040	8.80	67.0
	4	1.117	10.6	79.2
	5	1.208	12.7	91.5
355.5	1	1.051	9.75	72.1
	2	1.098	13.0	–
	3	1.460	29.0	–
	4	1.619	64.1	–
	5	1.469	24.5	–
366.7	1	1.823	88.0	425.0
	2	4.025	32500.0	15800.0
	3	5.676	–	–
	4	6.924	–	–
	5	8.671	–	–

[a] POV aged at 20 atm.
– Signifies the values were not determined
[b] Measured at 333.3 K
[c] Measured at 333.3 K and 10 rad/s

Table G-15. Short-Term POV Aging Data for SHRP (1990) AAM-1[a]

T K	t days	CA	η_o^{*}[b] kP	$(1 / J'')^{c} \times 10^{-3}$ dyne / cm^2
322.2	1	0.496	3.99	36.0
	2	0.510	6.18	52.7
	3	0.605	7.33	62.7
	4	0.694	8.55	69.5
	5	0.535	9.30	76.3
333.3	1	0.520	4.75	42.7
	2	0.719	11.2	88.5
	3	0.763	14.0	105.0
	4	0.854	16.9	118.0
	5	0.840	20.2	142.0
344.4	1	0.556	6.45	55.6
	2	1.001	27.5	176.0
	3	1.070	39.4	228.0
	4	1.123	50.0	273.0
	5	1.348	51.0	290.0
355.5	1	1.025	38.0	223.0
	2	1.195	50.4	271.0
	3	1.309	86.0	406.0
	4	1.370	80.0	387.0
	5	1.442	90.0	467.0
366.7	1	1.557	125.0	498.0
	2	1.780	417.0	1000.0
	3	1.931	828.0	1460.0
	4	2.127	1770.0	2260.0
	5	2.436	12400.0	6040.0

[a] POV aged at 20 atm.
– Signifies the values were not determined
[b] Measured at 333.3 K
[c] Measured at 333.3 K and 10 rad/s

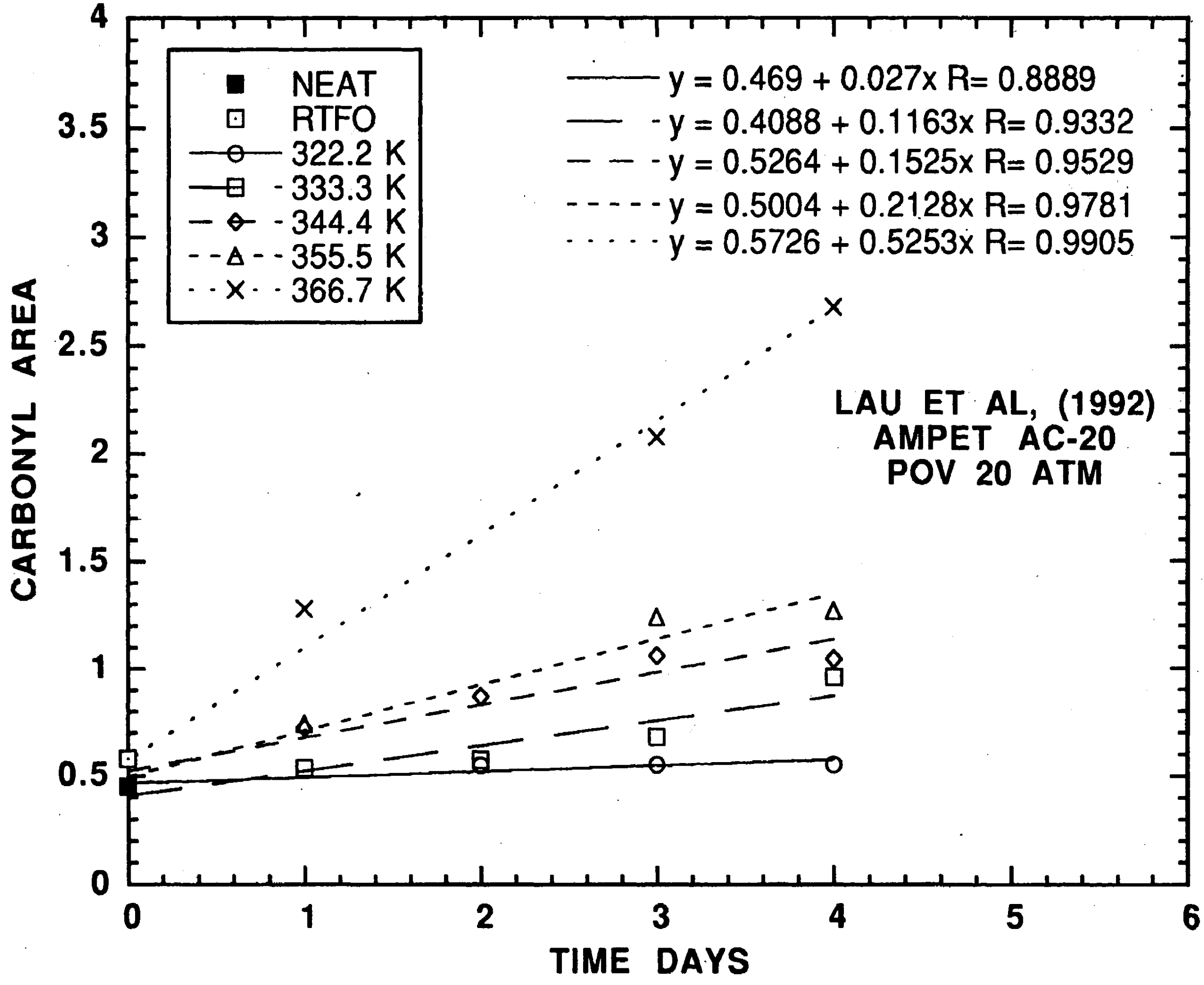

Figure G-1. *CA*s of neat, RTFO, and POV-aged Lau *et al.*, (1992) Ampet AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

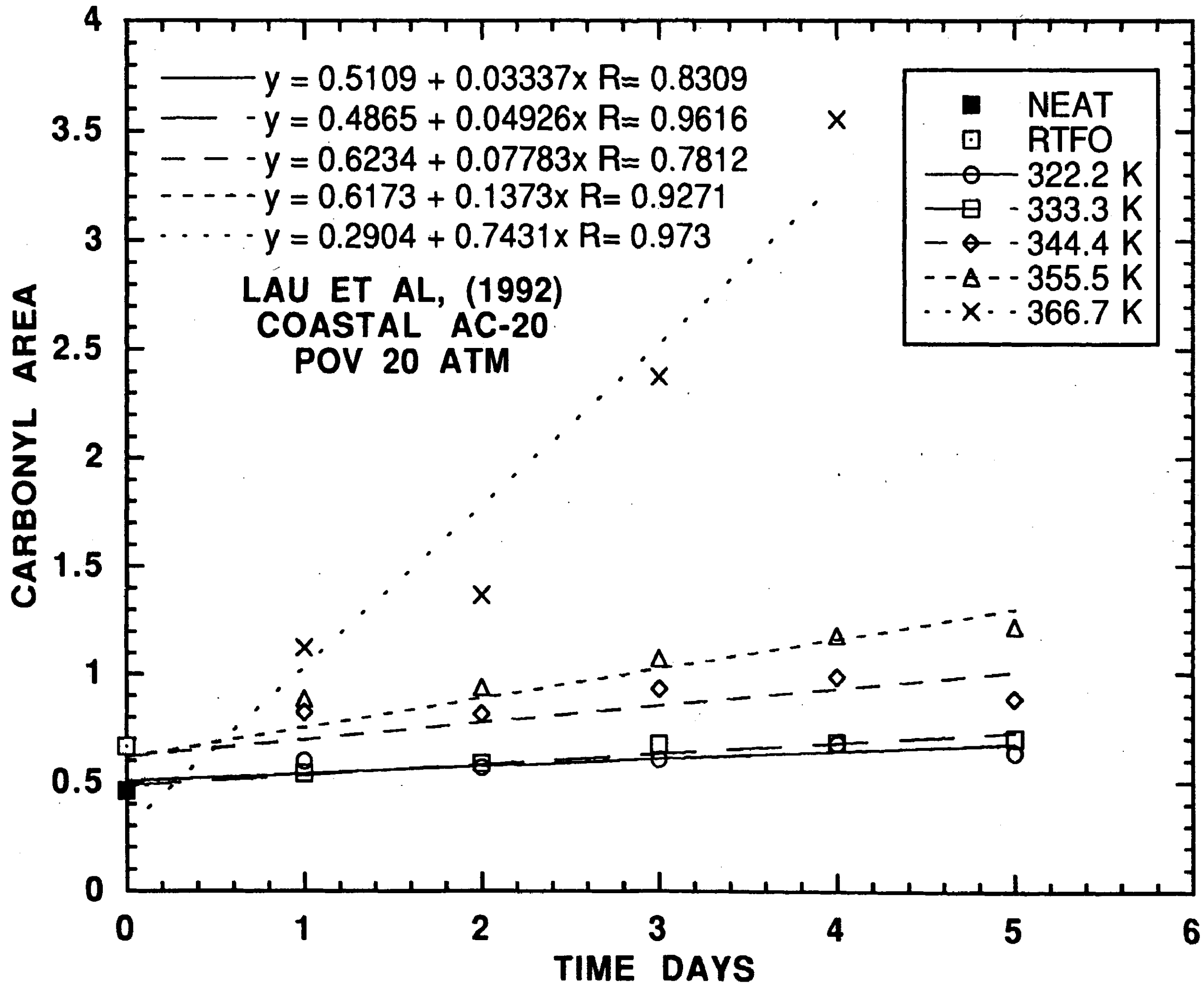

Figure G-2. *CA*s of neat, RTFO, and POV-aged Lau *et al.*, (1992) Coastal AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

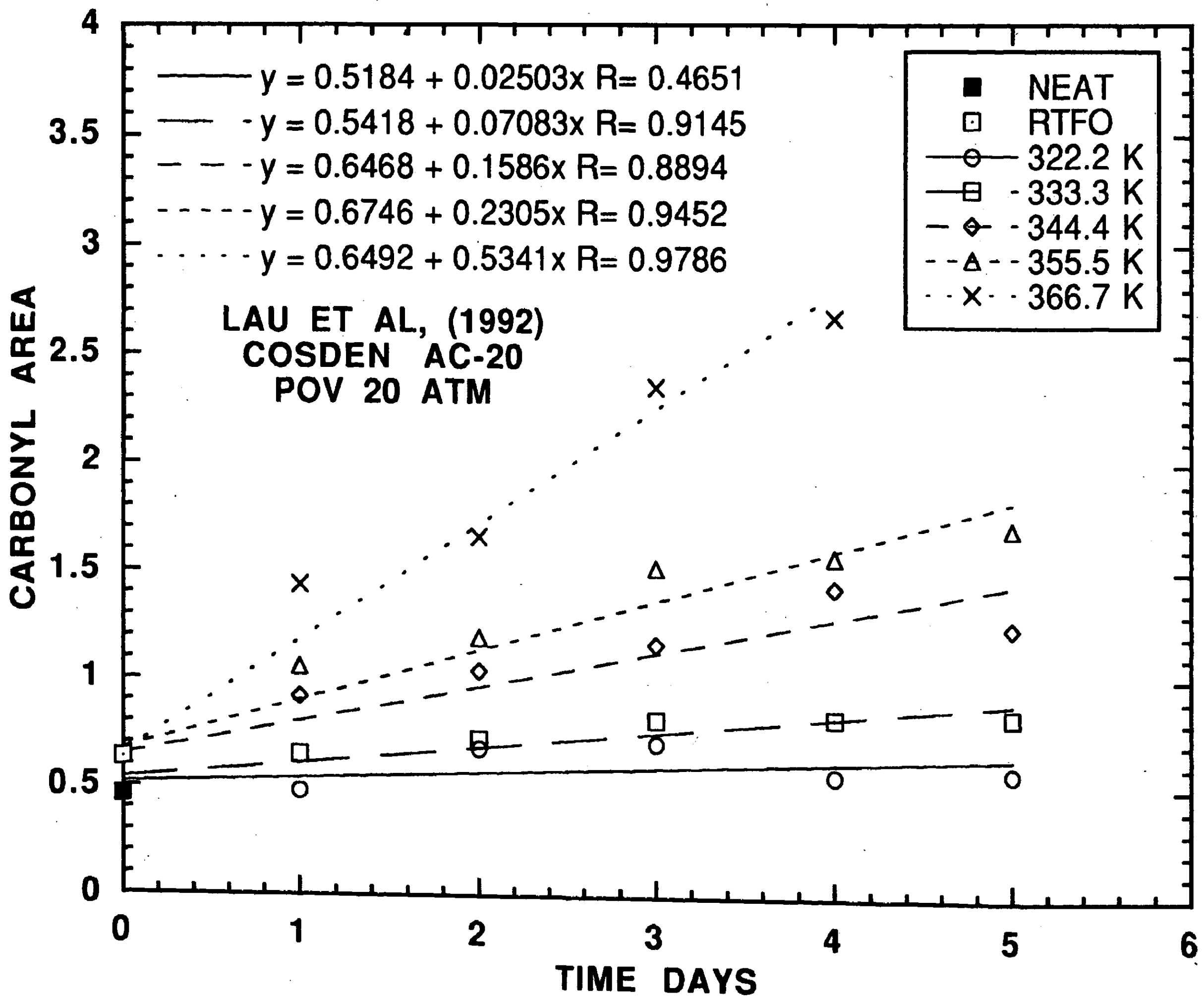

Figure G-3. *CAs* of neat, RTFO, and POV-aged Lau *et al.*, (1992) Cosden AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

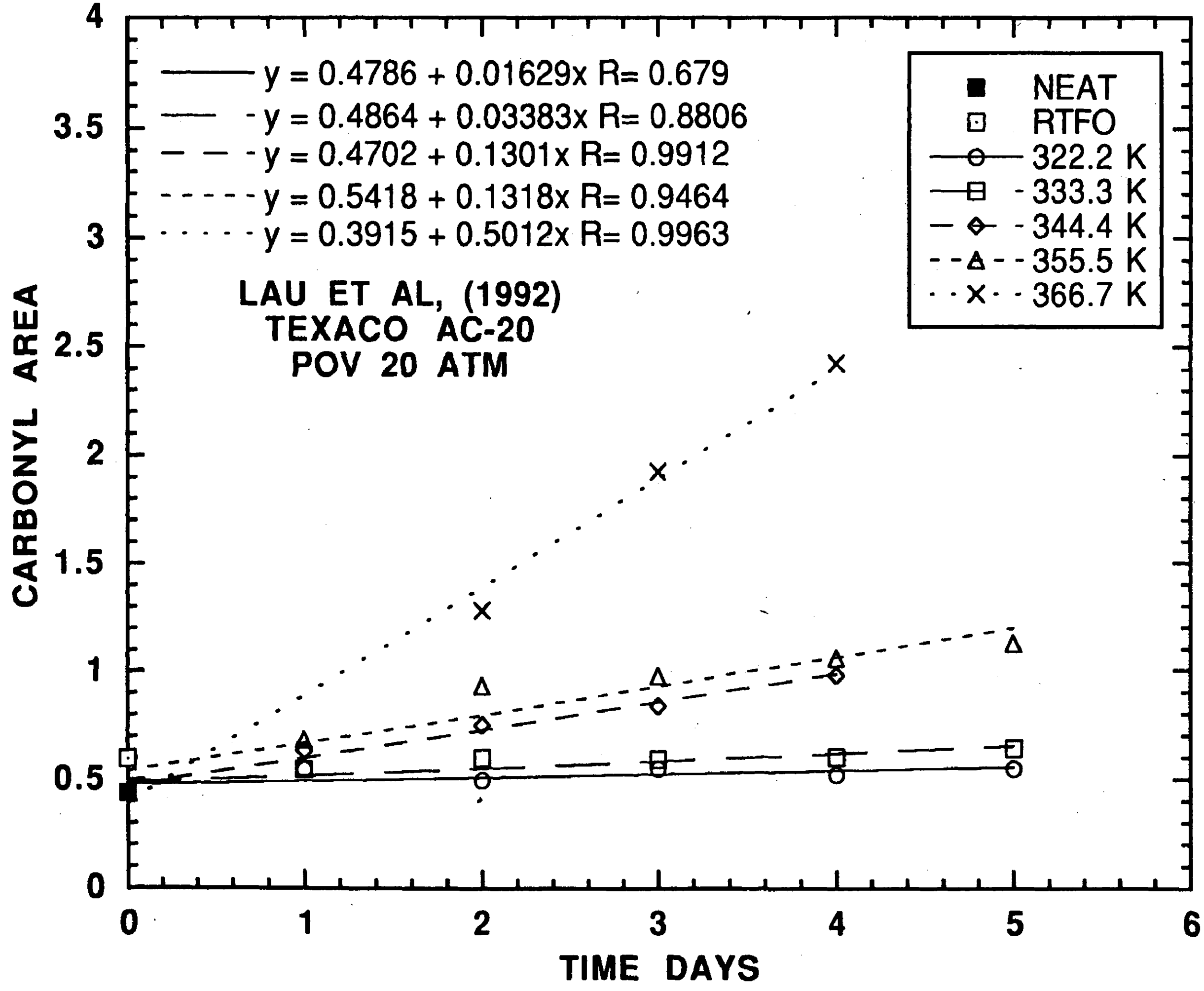

Figure G-4. *CA*s of neat, RTFO, and POV-aged Lau *et al.*, (1992) Texaco AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

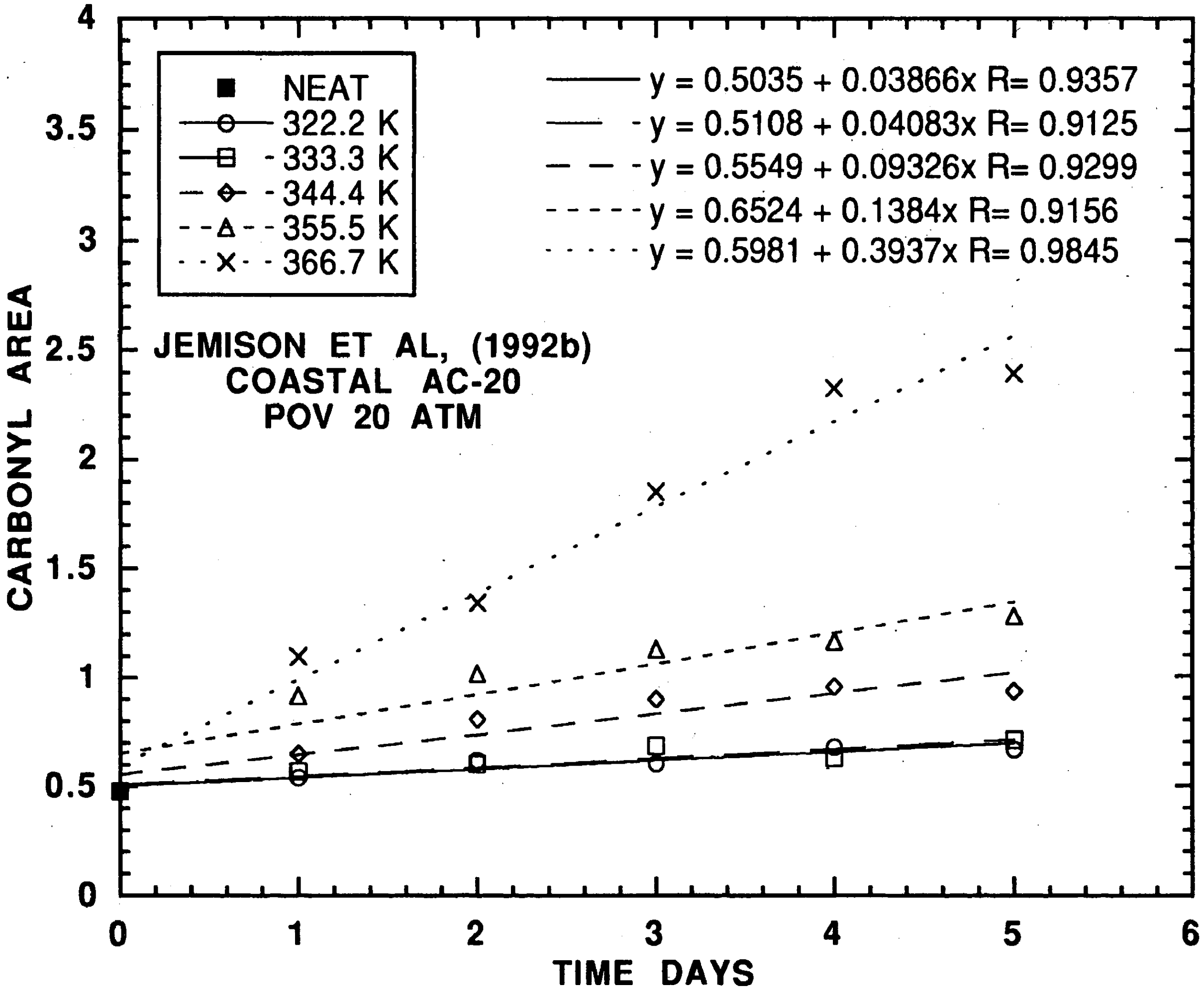

Figure G-5. *CA*s of neat and POV-aged Jemison *et al.*, (1992b) Coastal AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

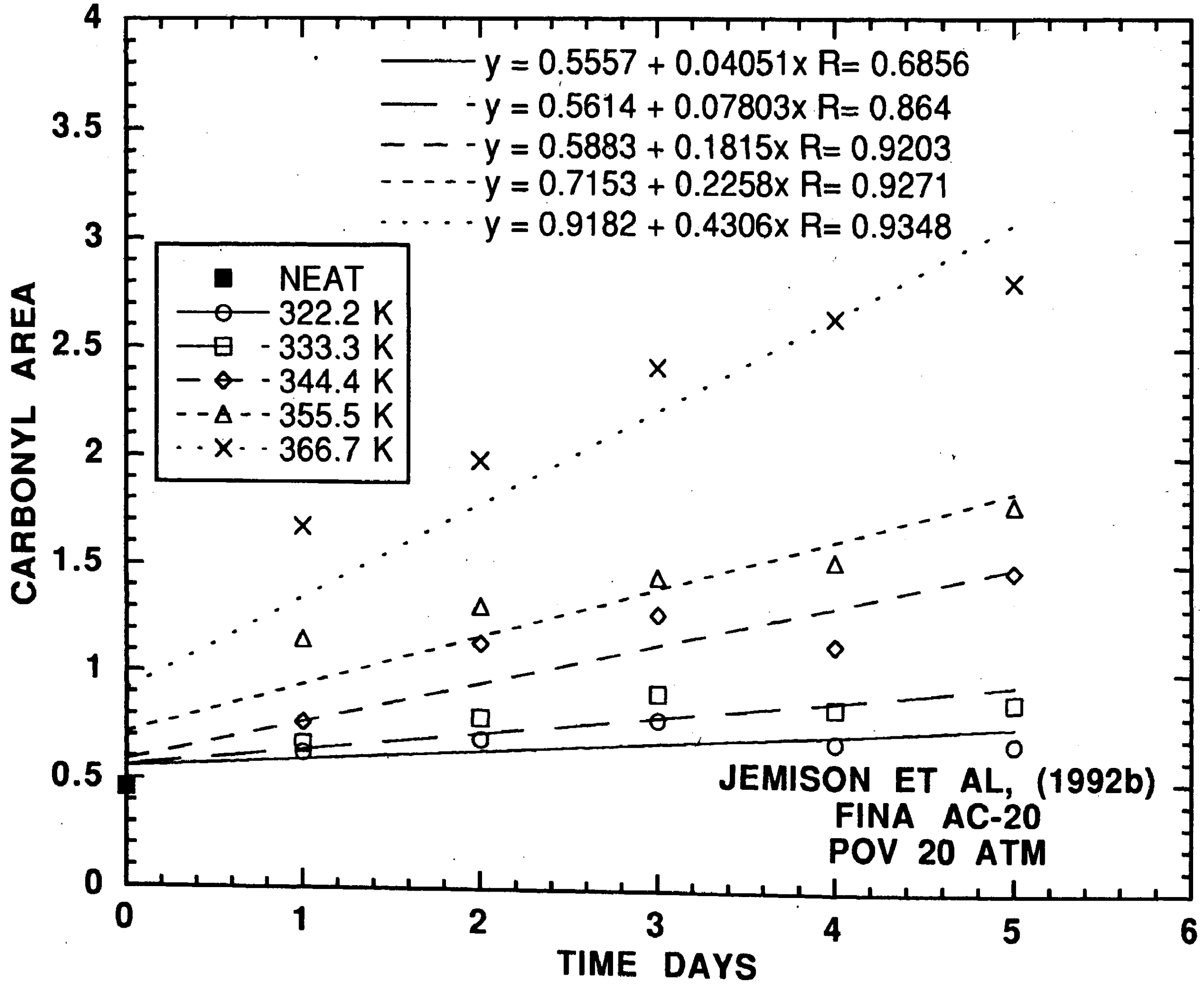

Figure G-6. *CAs* of neat and POV-aged Jemison *et al.*, (1992b) Fina AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

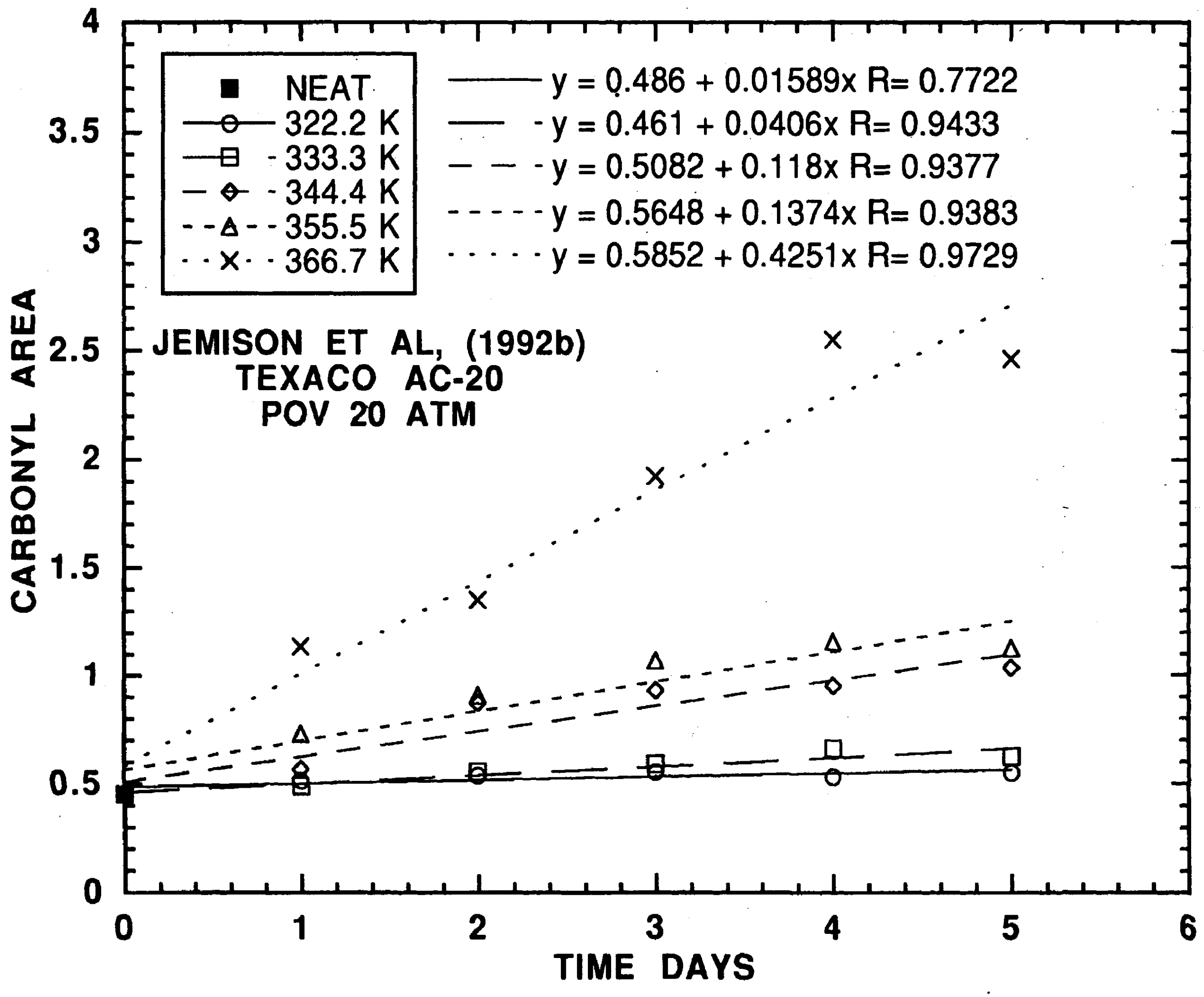

Figure G-7. *CA*s of neat and POV-aged Jemison *et al.*, (1992b) Texaco AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

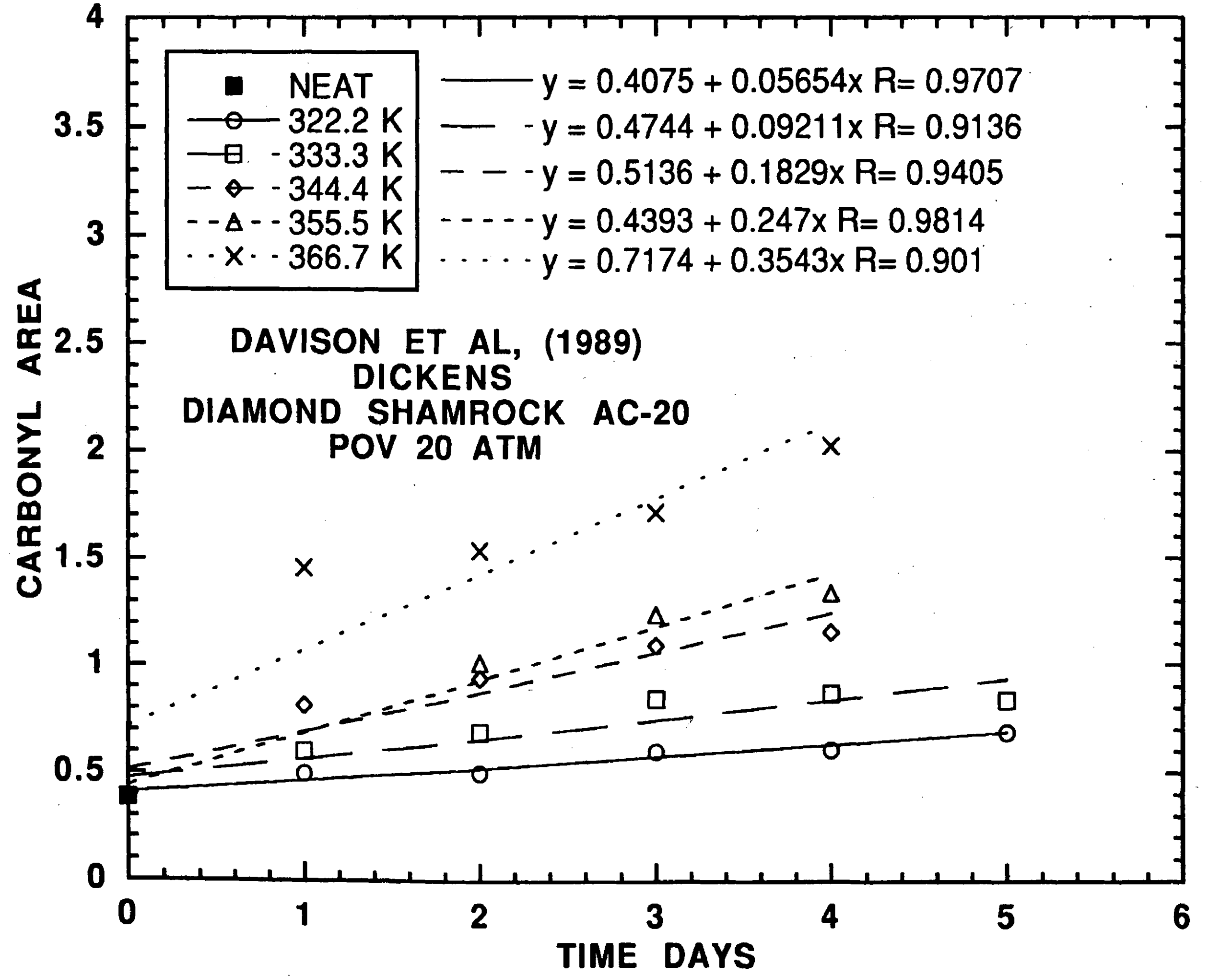

Figure G-8. *CA*s of neat and POV-aged Dickens Diamond Shamrock AC-20 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

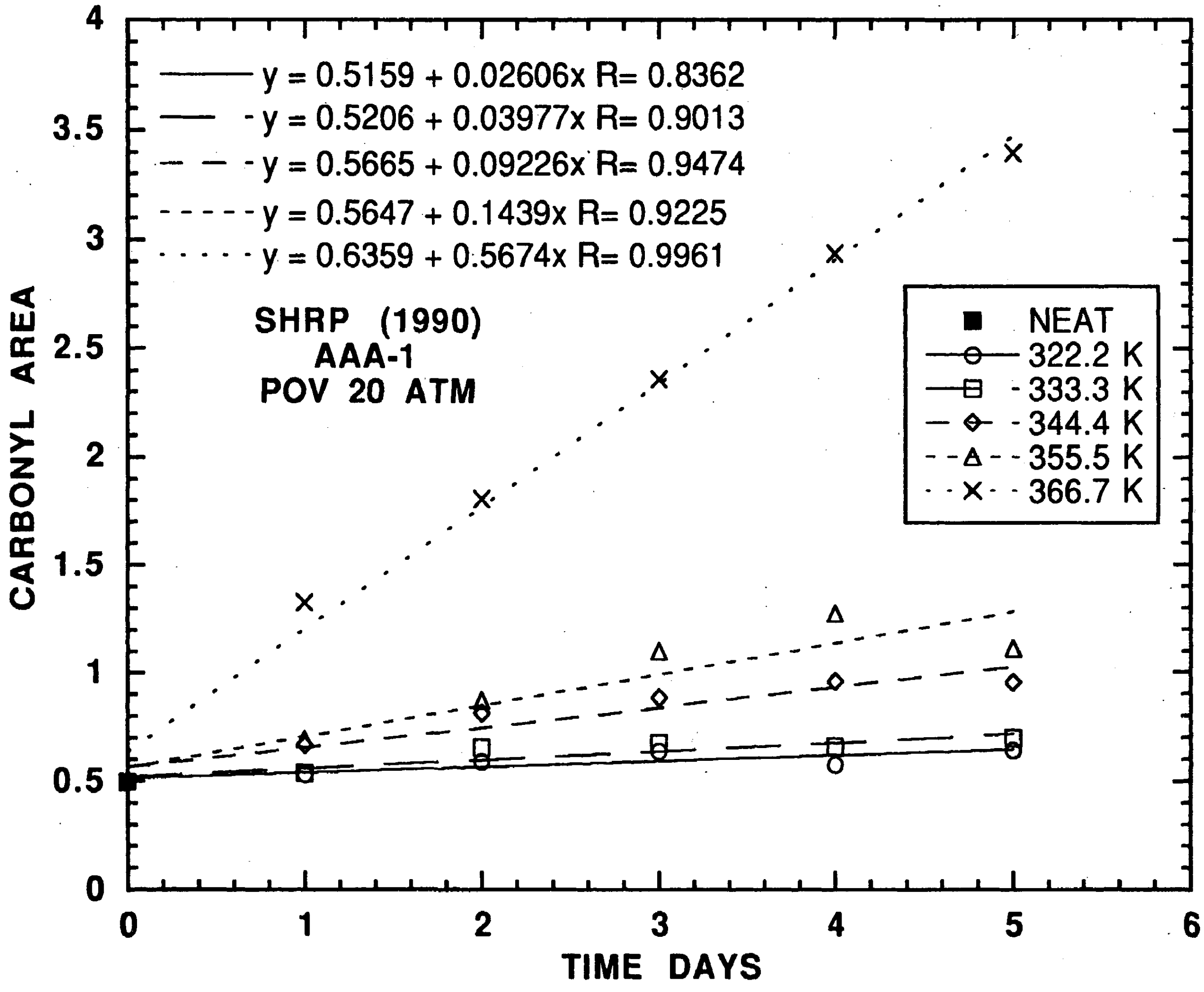

Figure G-9. *CA*s of neat and POV-aged SHRP (1990) AAA-1 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

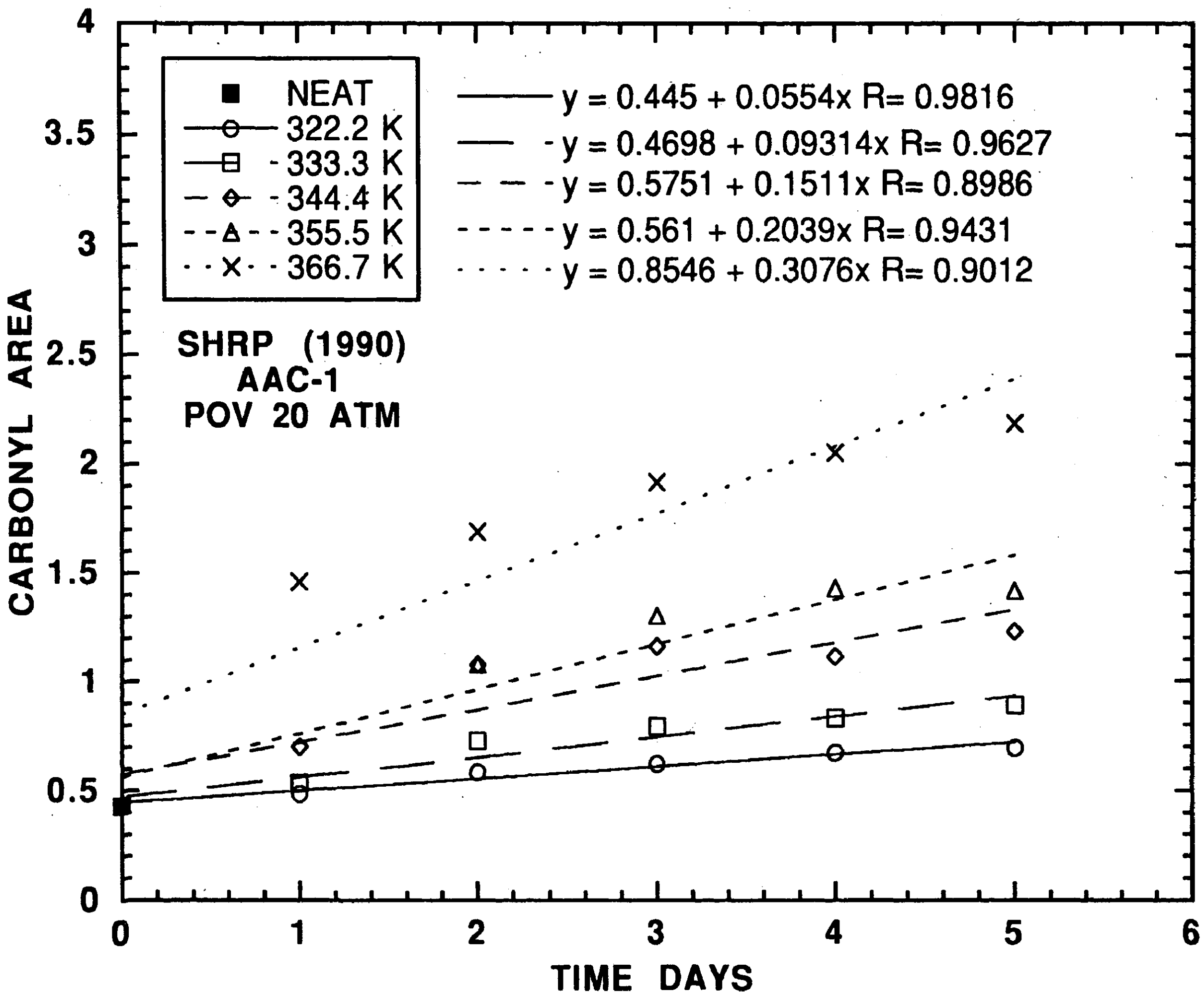

Figure G-10. *CA*s of neat and POV-aged SHRP (1990) AAC-1 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

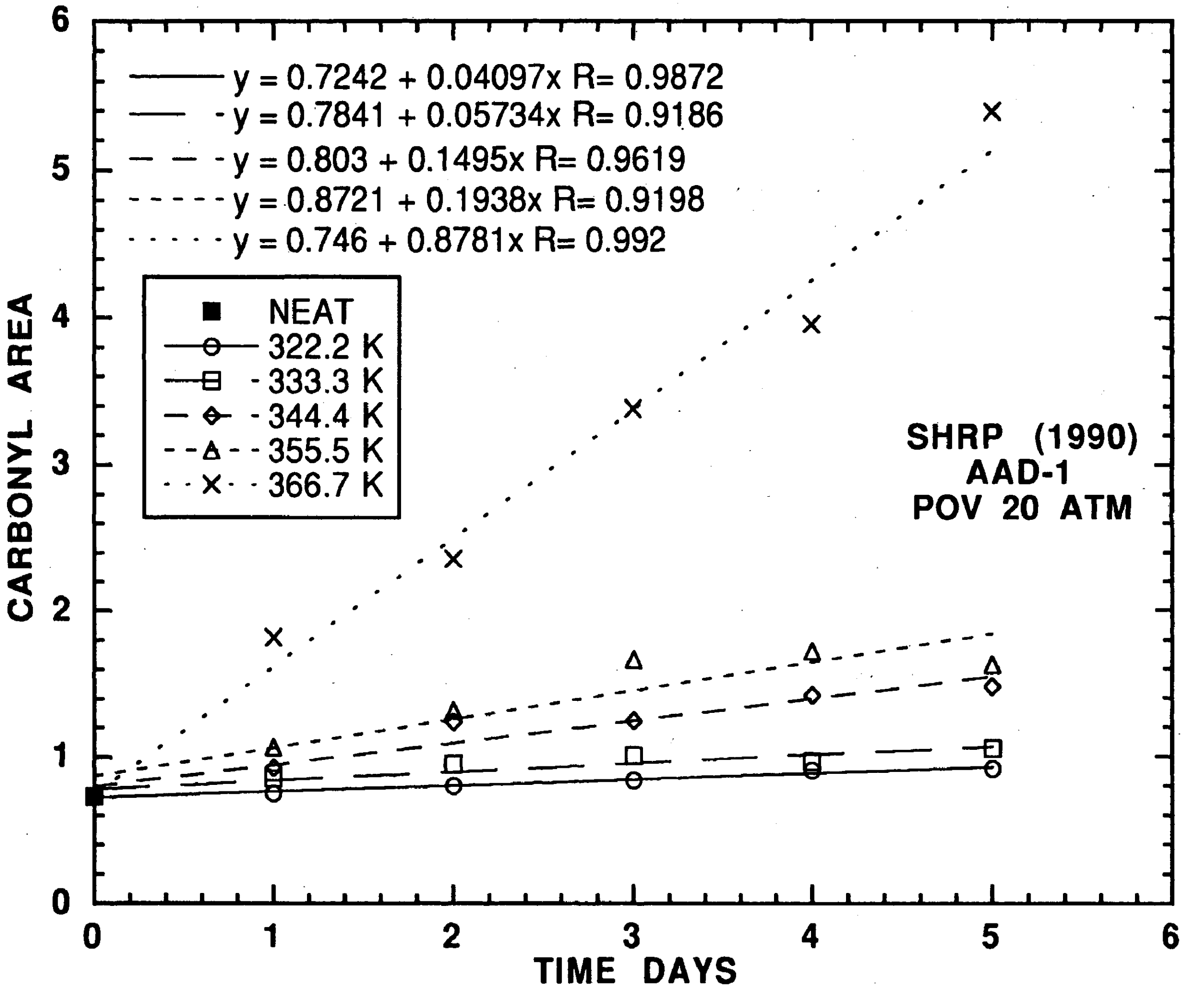

Figure G-11. *CA*s of neat and POV-aged SHRP (1990) AAD-1 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

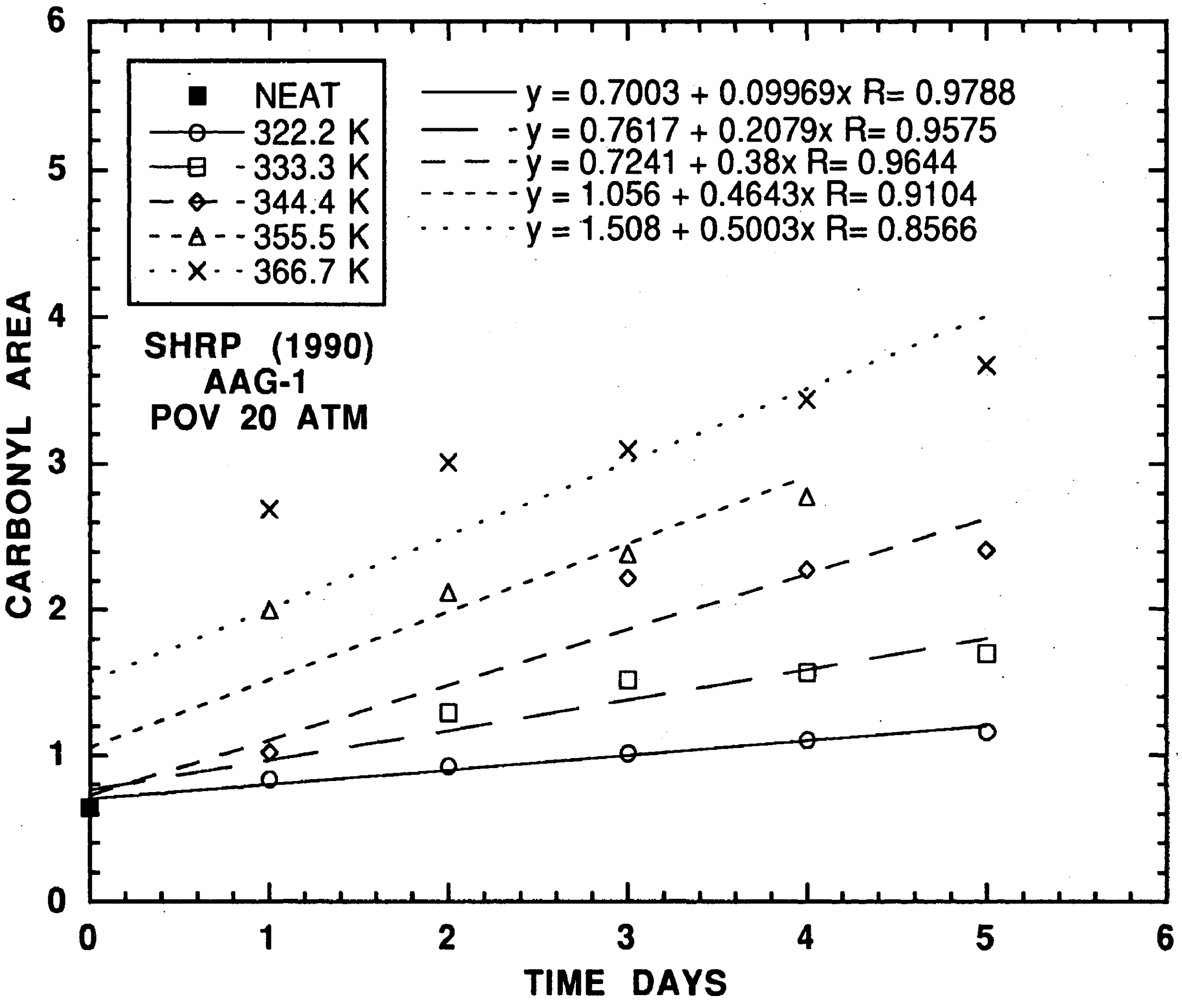

Figure G-12. *CA*s of neat and POV-aged SHRP (1990) AAG-1 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

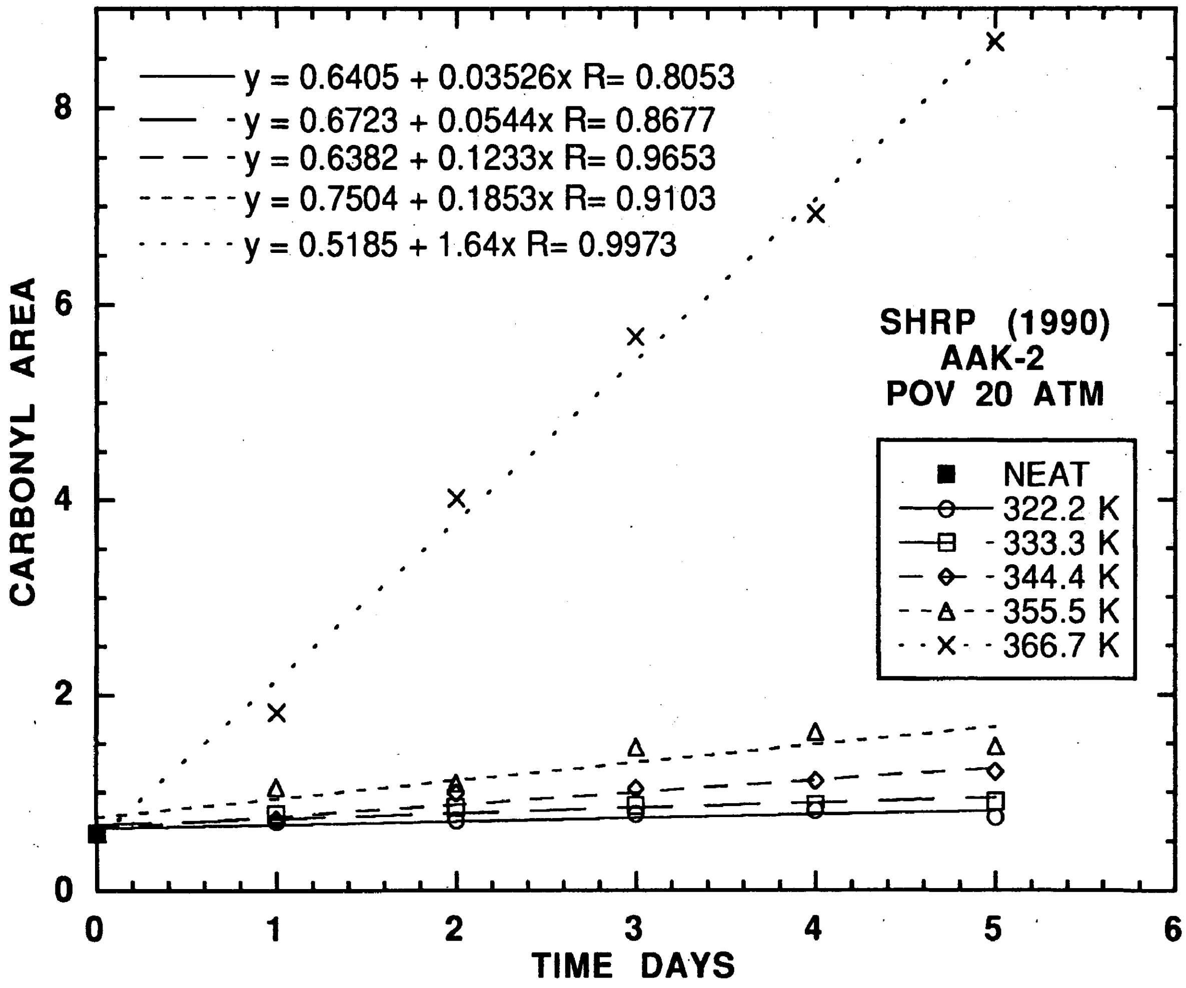

Figure G-13. *CA*s of neat and POV-aged SHRP (1990) AAK-2 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

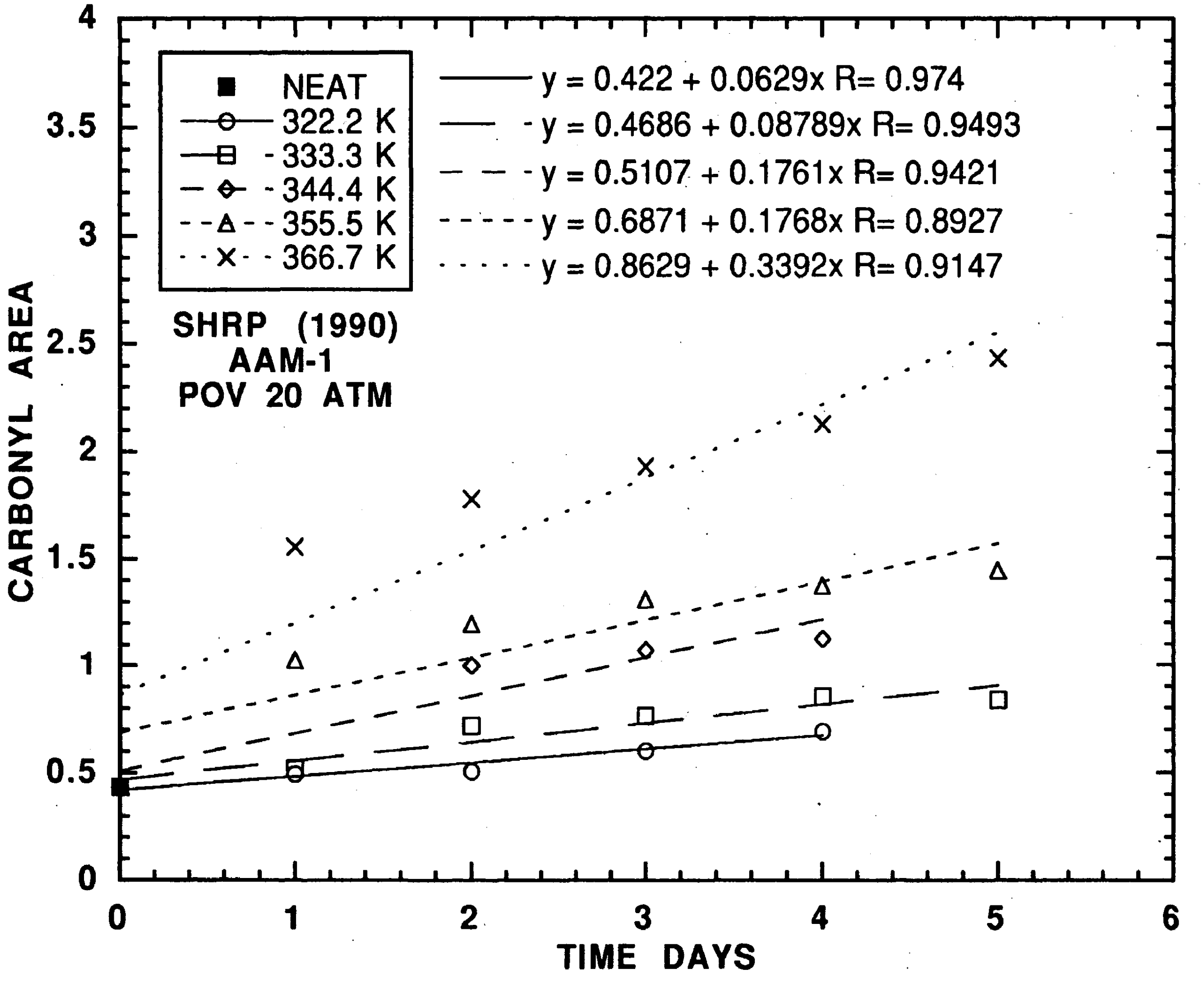

Figure G-14. *CA*s of neat and POV-aged SHRP (1990) AAM-1 at 322.2, 333.3, 344.4, 355.5, and 366.7 K at 20 atm from 1 to 5 days.

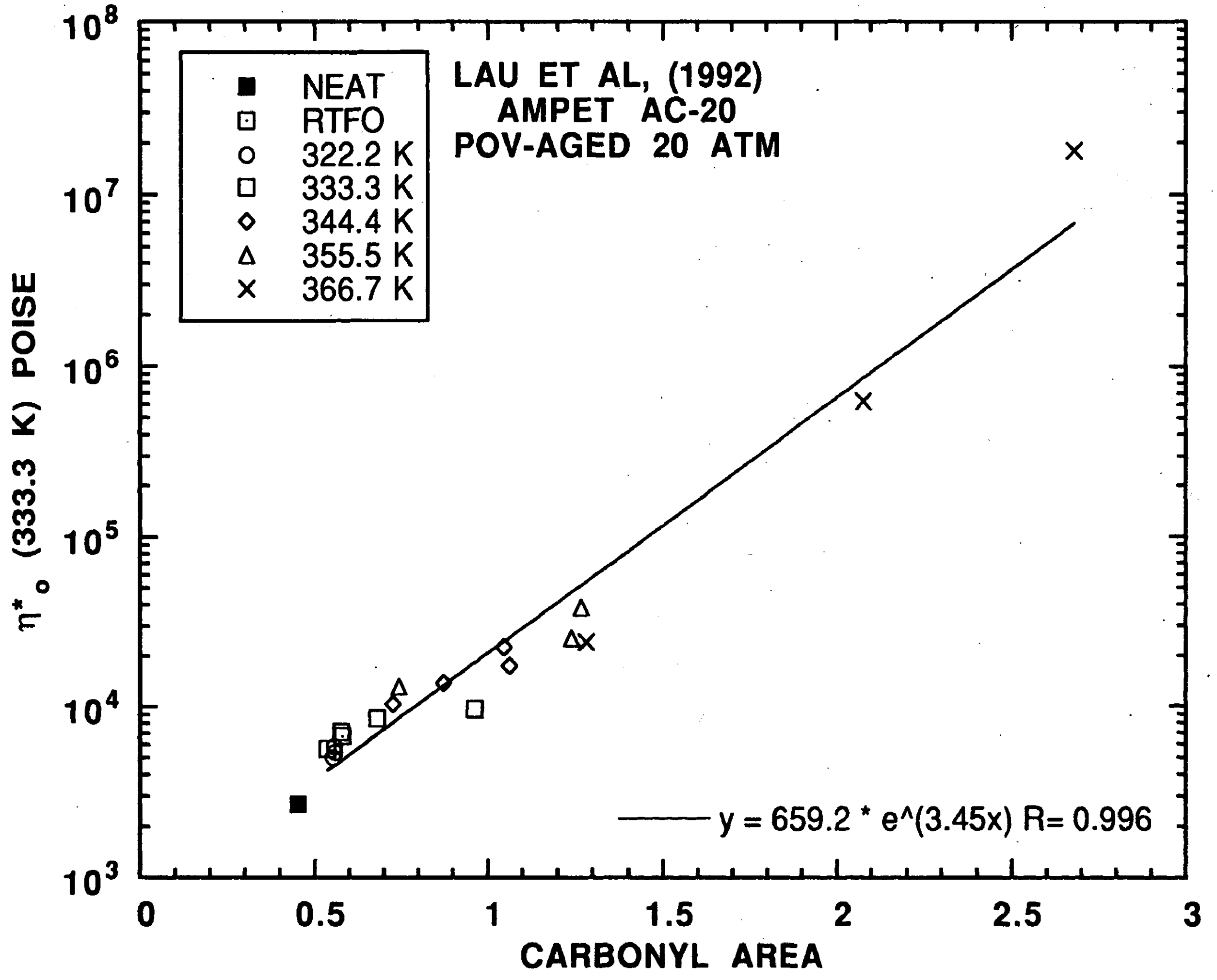

Figure G-15. η_0^* at 333.3 K and *CA* of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Ampet AC-20.

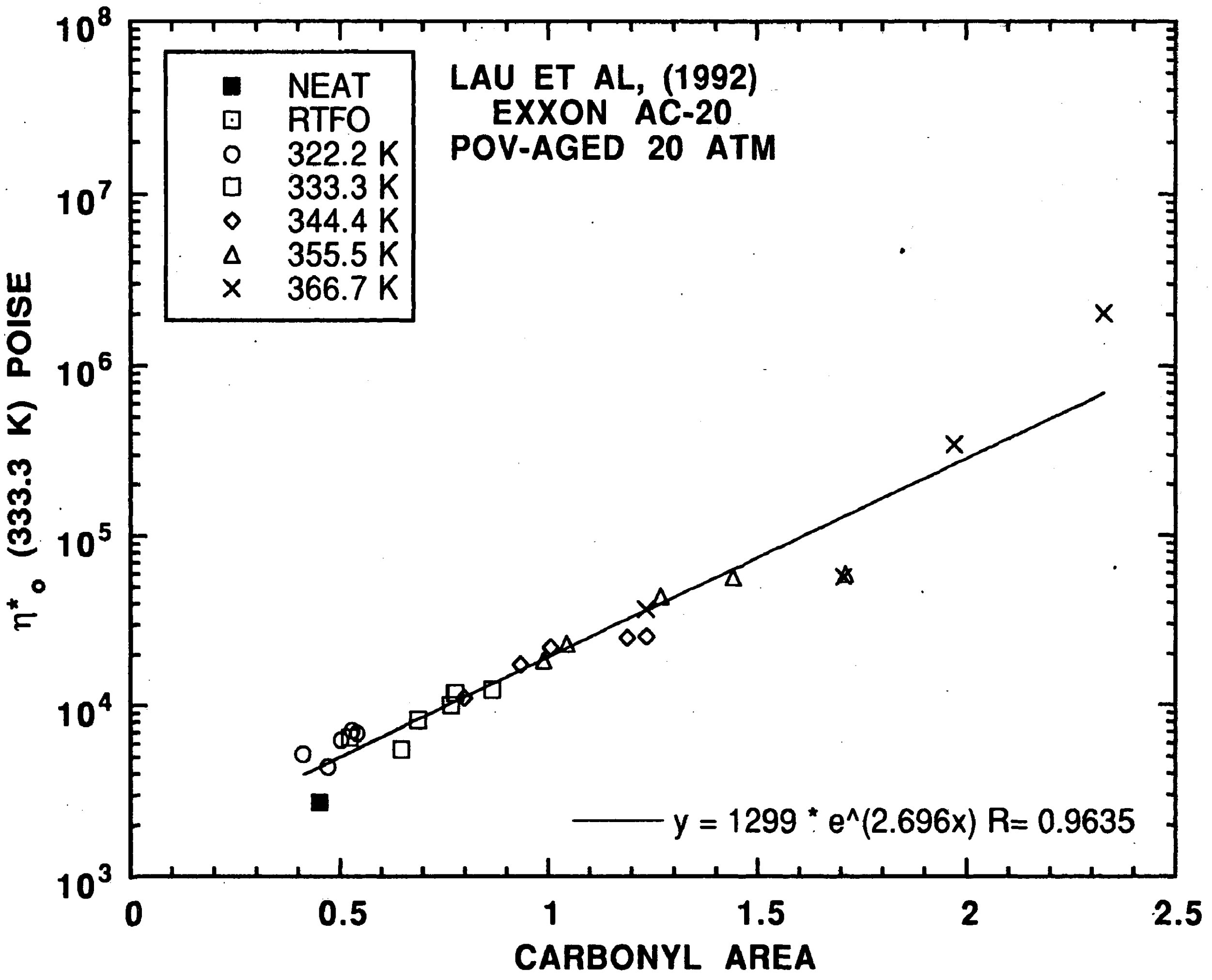

Figure G-16. η_0^* at 333.3 K and *CA* of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Exxon AC-20.

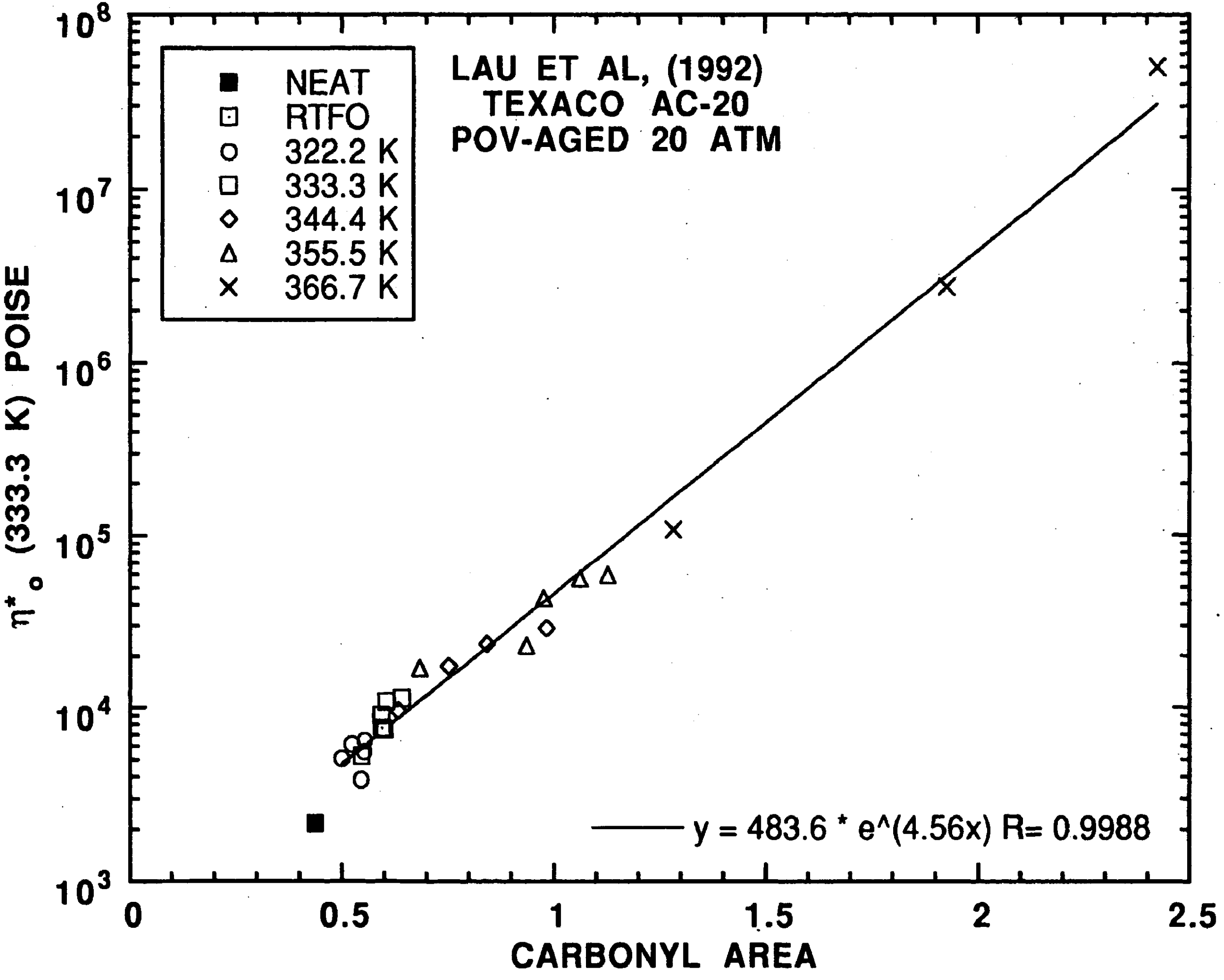

Figure G-17. η^*_0 at 333.3 K and *CA* of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Texaco AC-20.

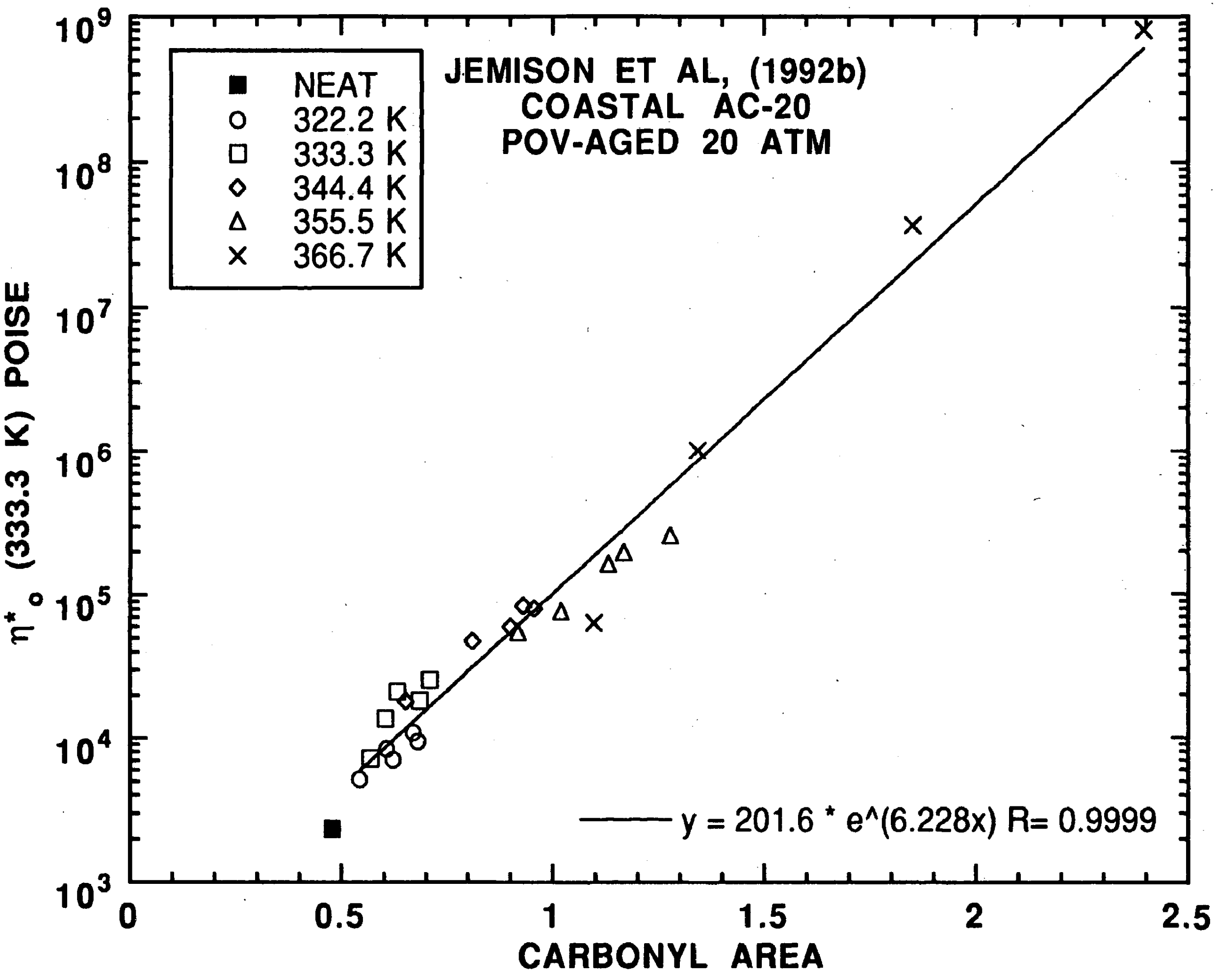

Figure G-18. η_o^* at 333.3 K and CA of neat and all short-term POV-aging conditions studied for Jemison *et al.*, (1992b) Coastal AC-20.

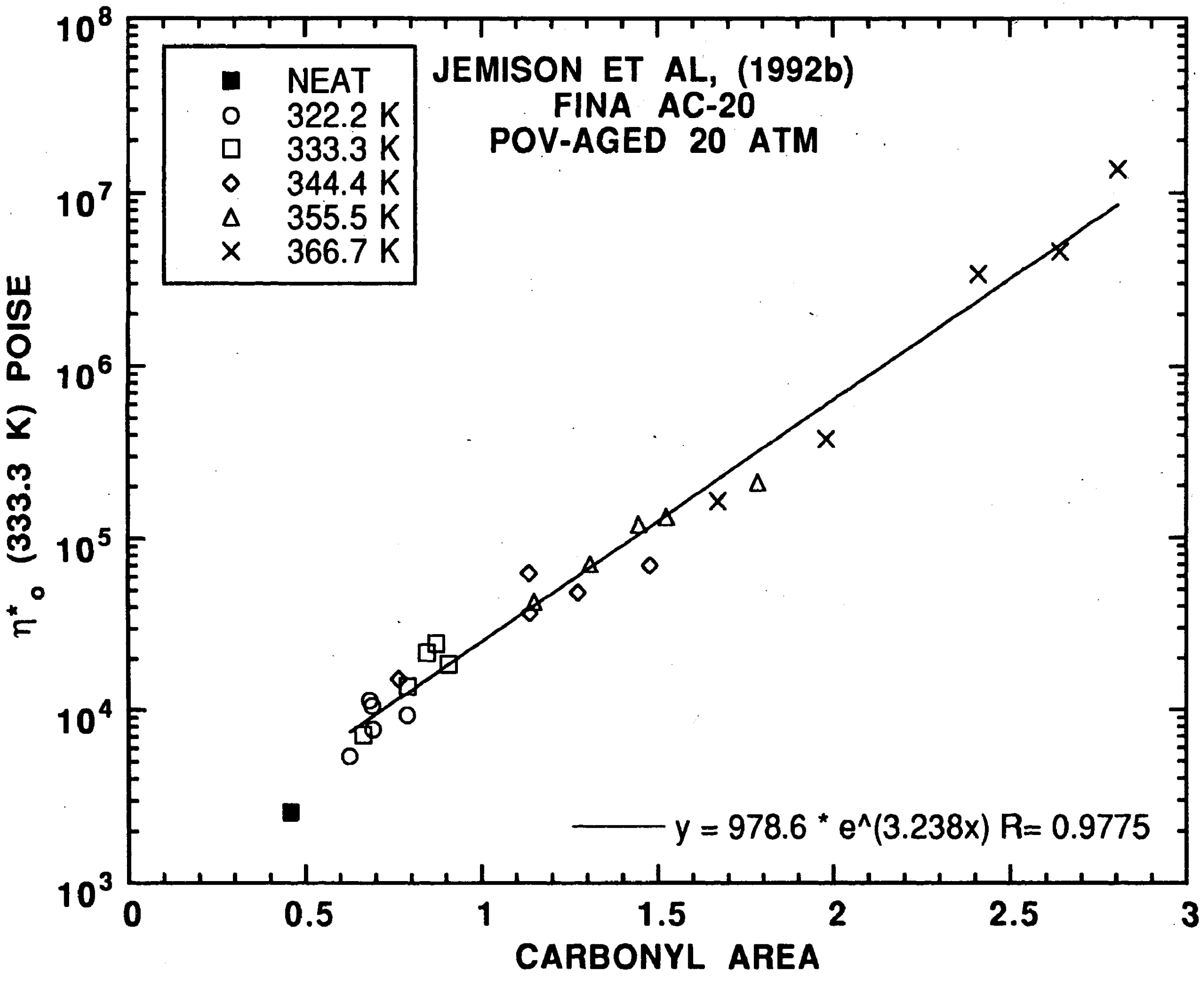

Figure G-19. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for Jemison *et al.*, (1992b) Fina AC-20.

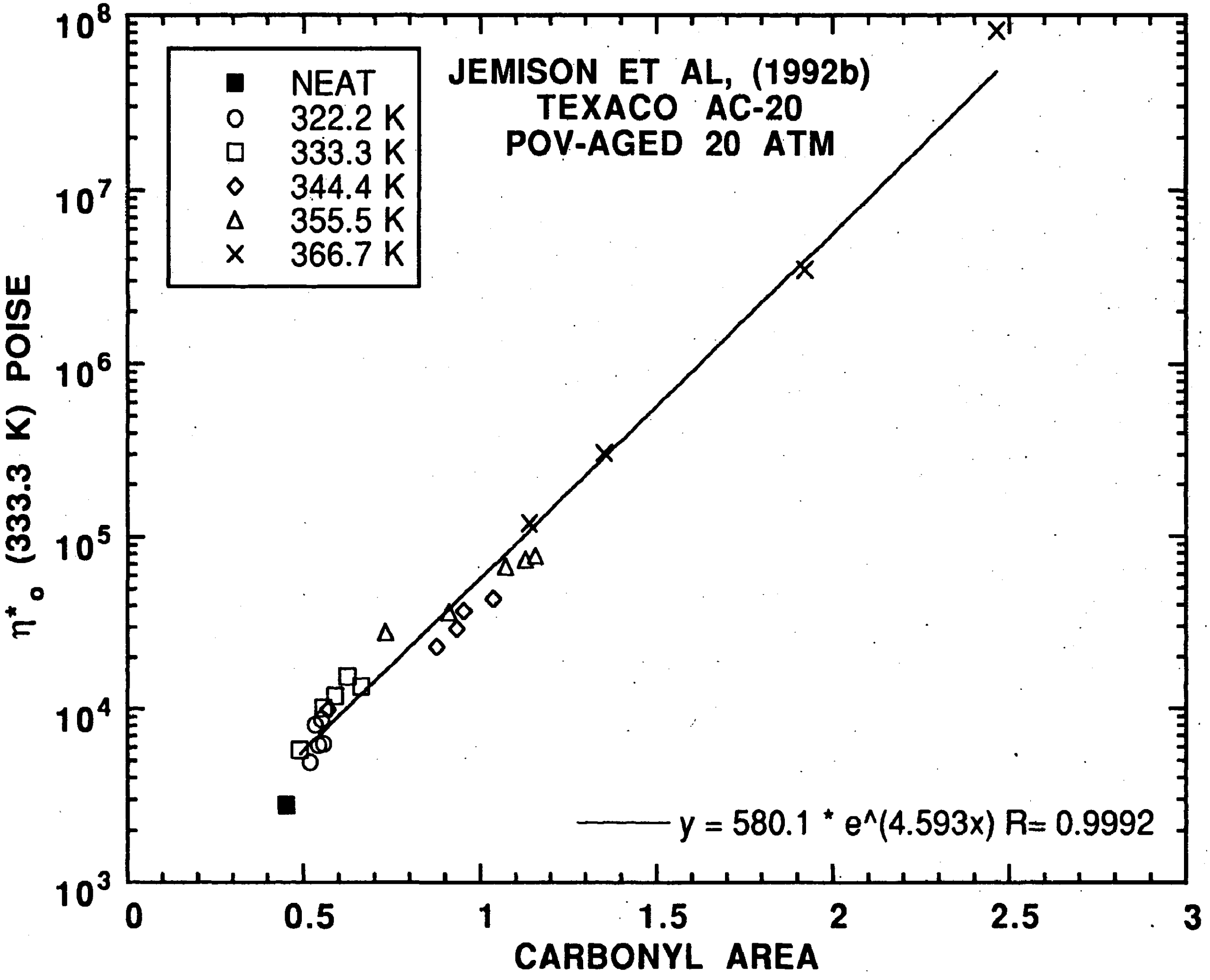

Figure G-20. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for Jemison *et al.*, (1992b) Texaco AC-20.

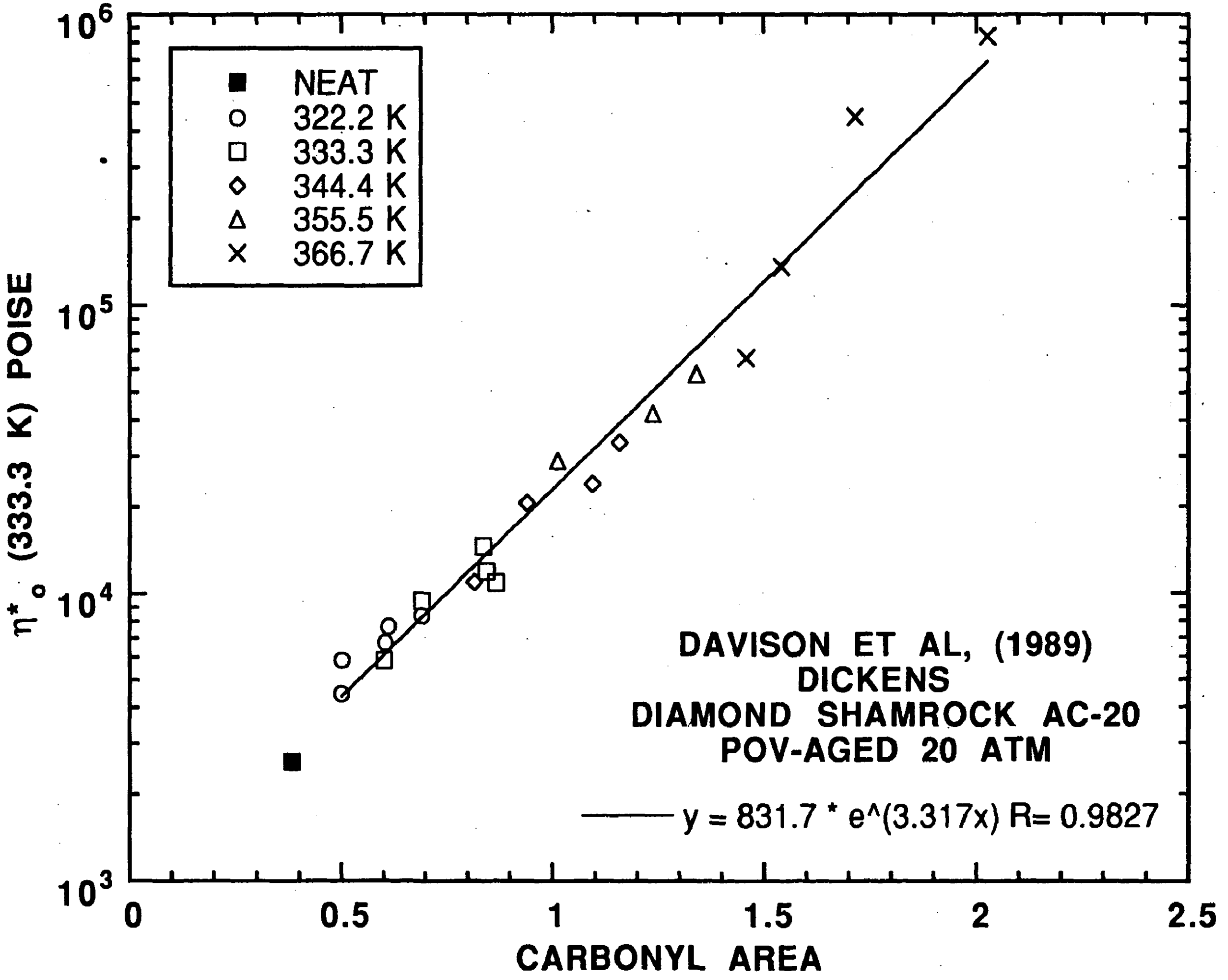

Figure G-21. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for Dickens Diamond Shamrock AC-20.

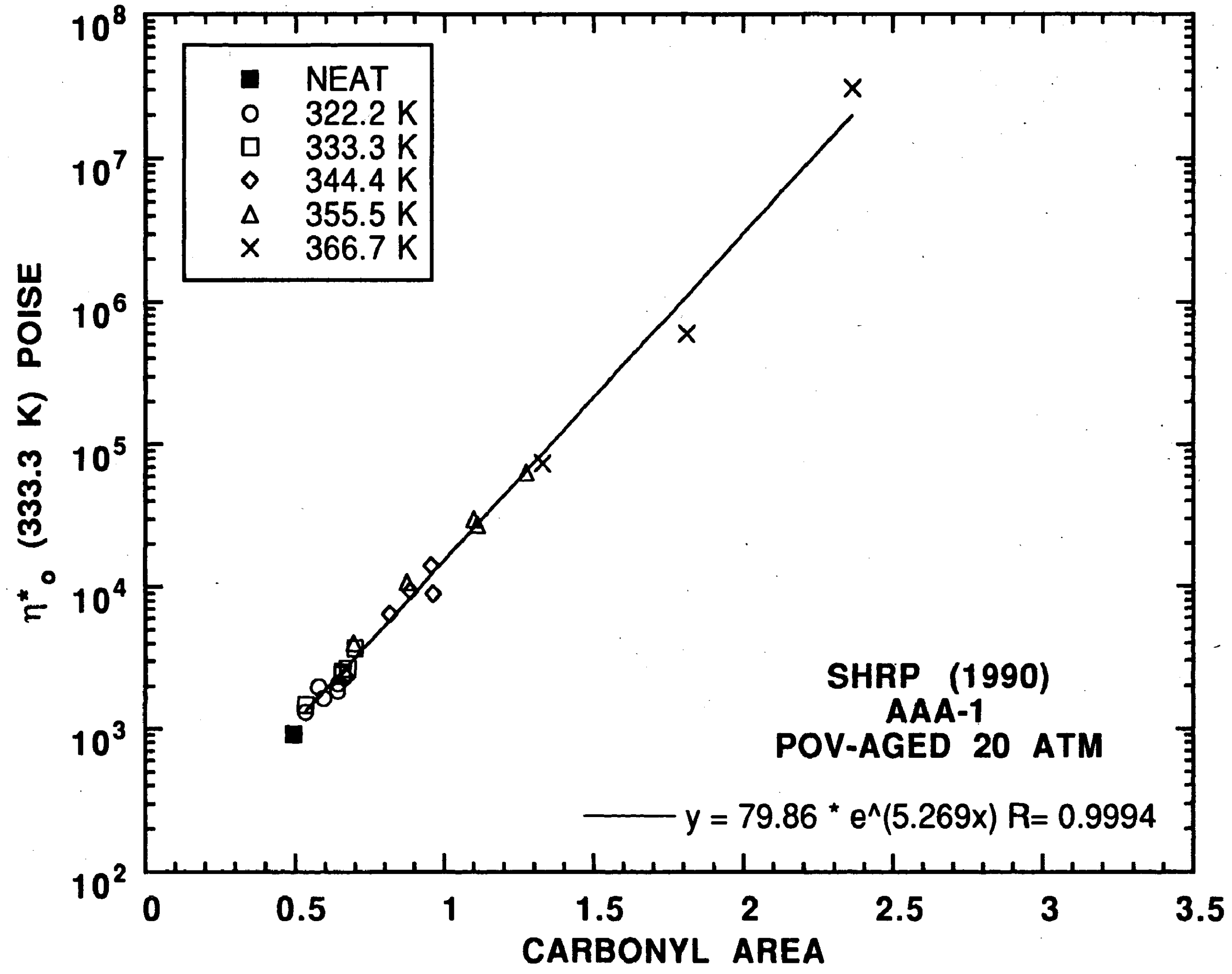

Figure G-22. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAA-1.

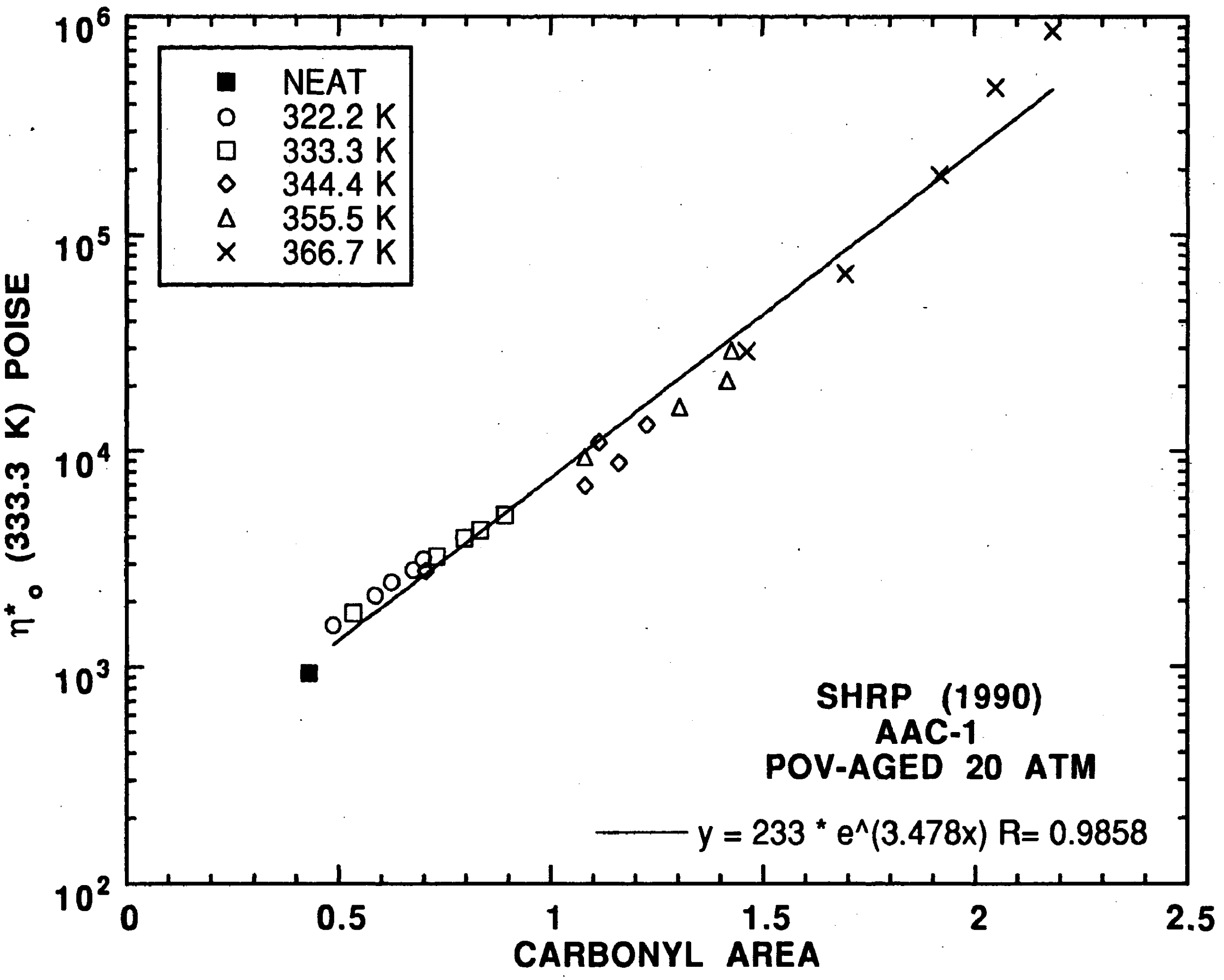

Figure G-23. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAC-1.

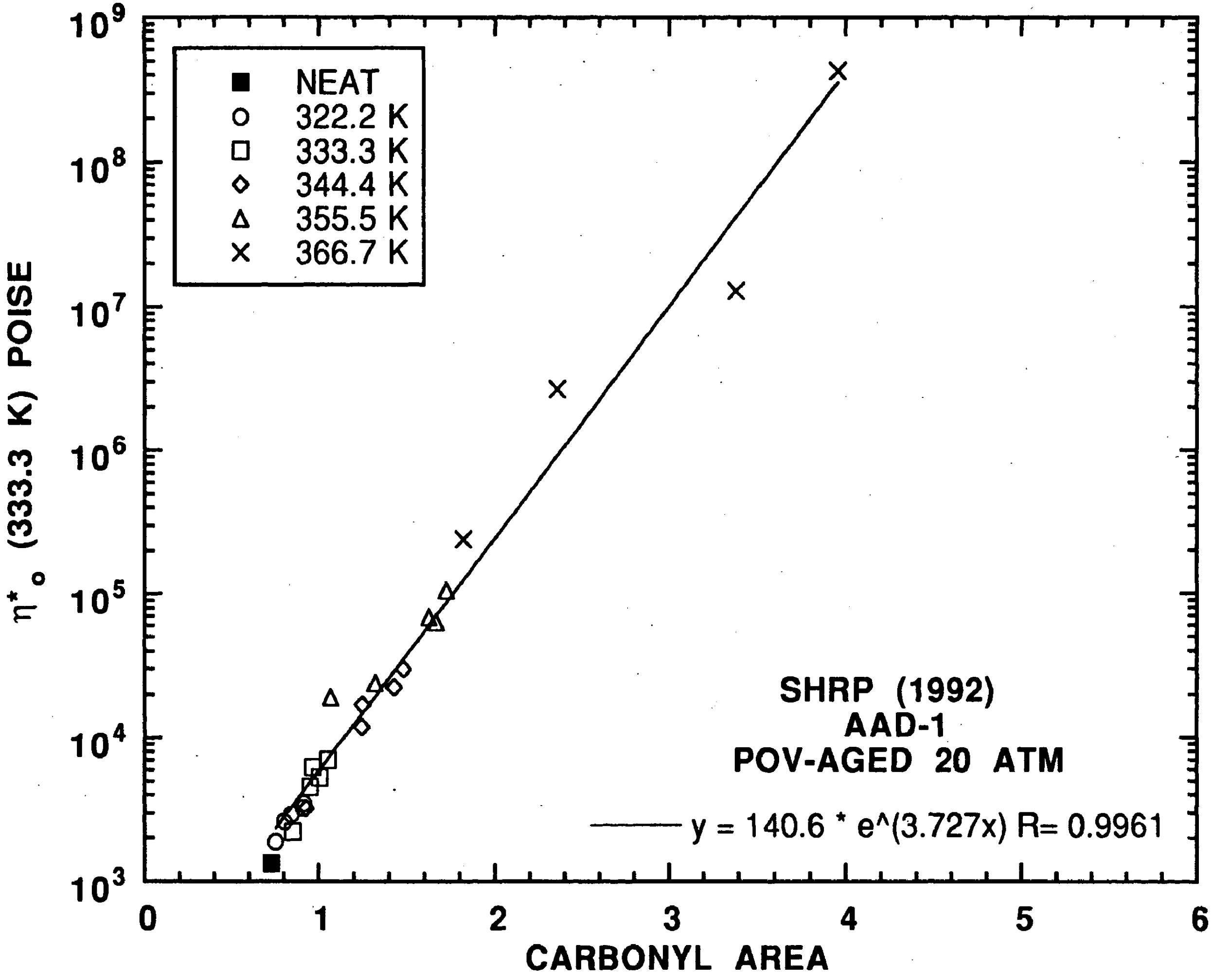

Figure G-24. η_o^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAD-1.

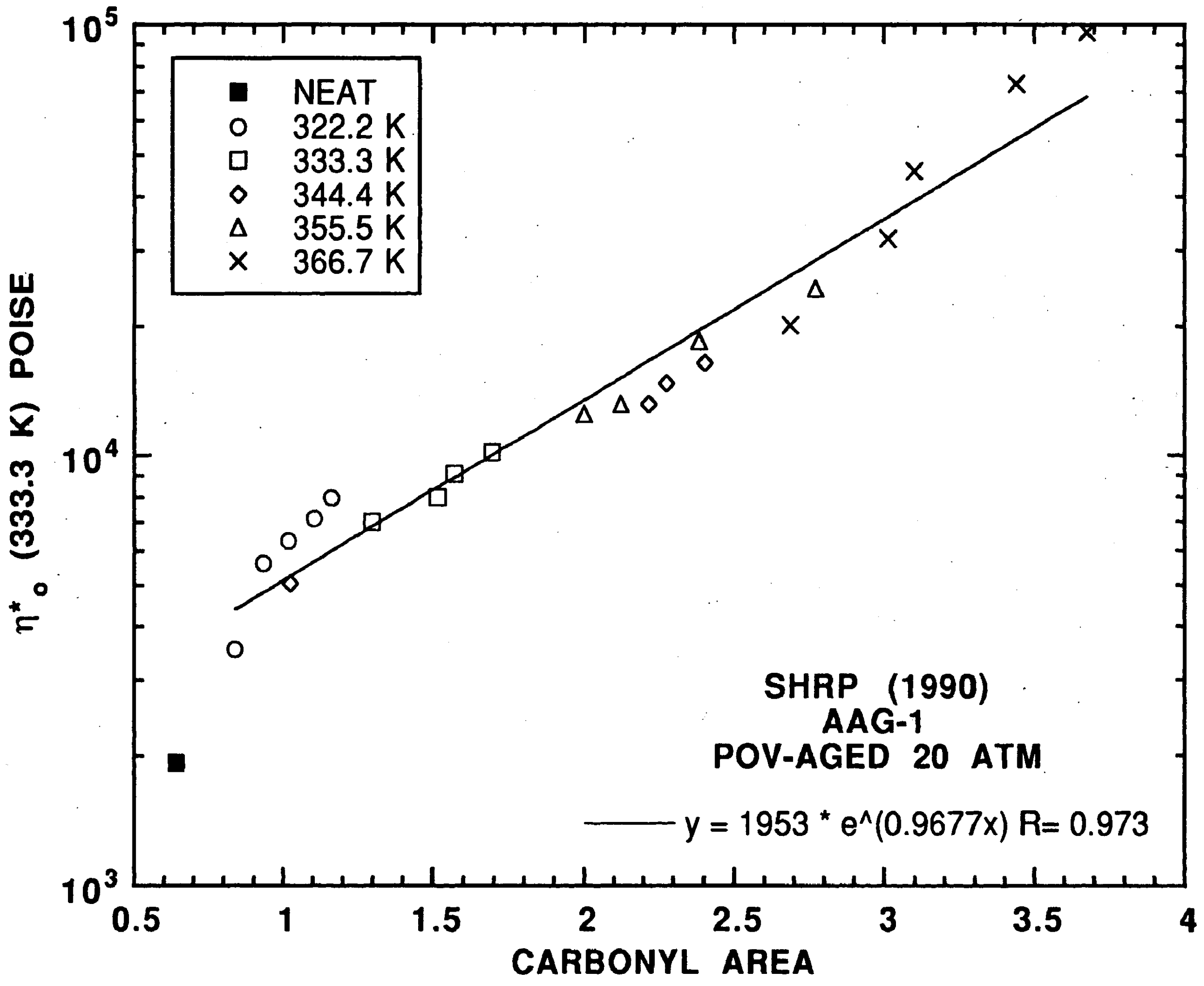

Figure G-25. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAG-1.

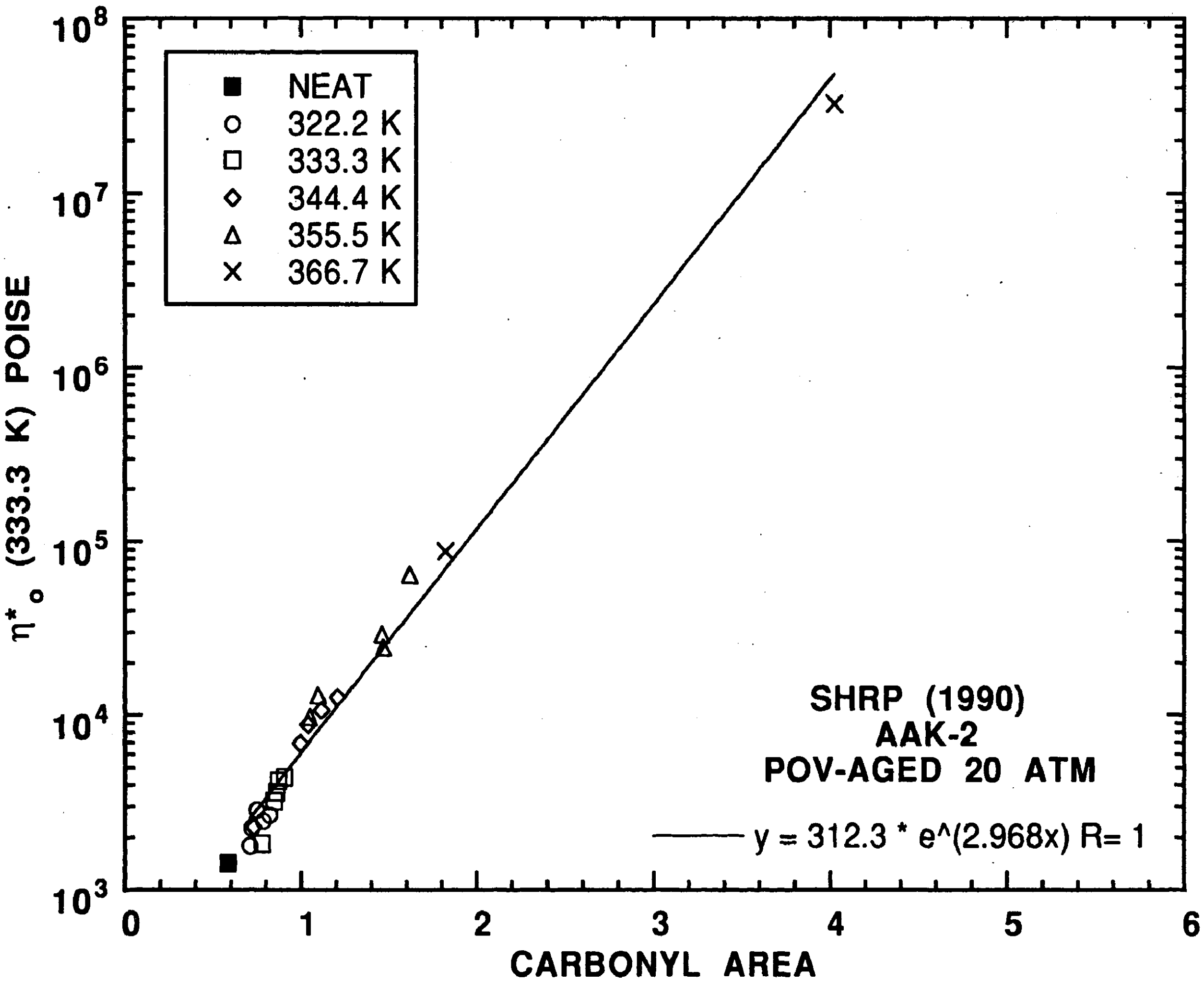

Figure G-26. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAK-2.

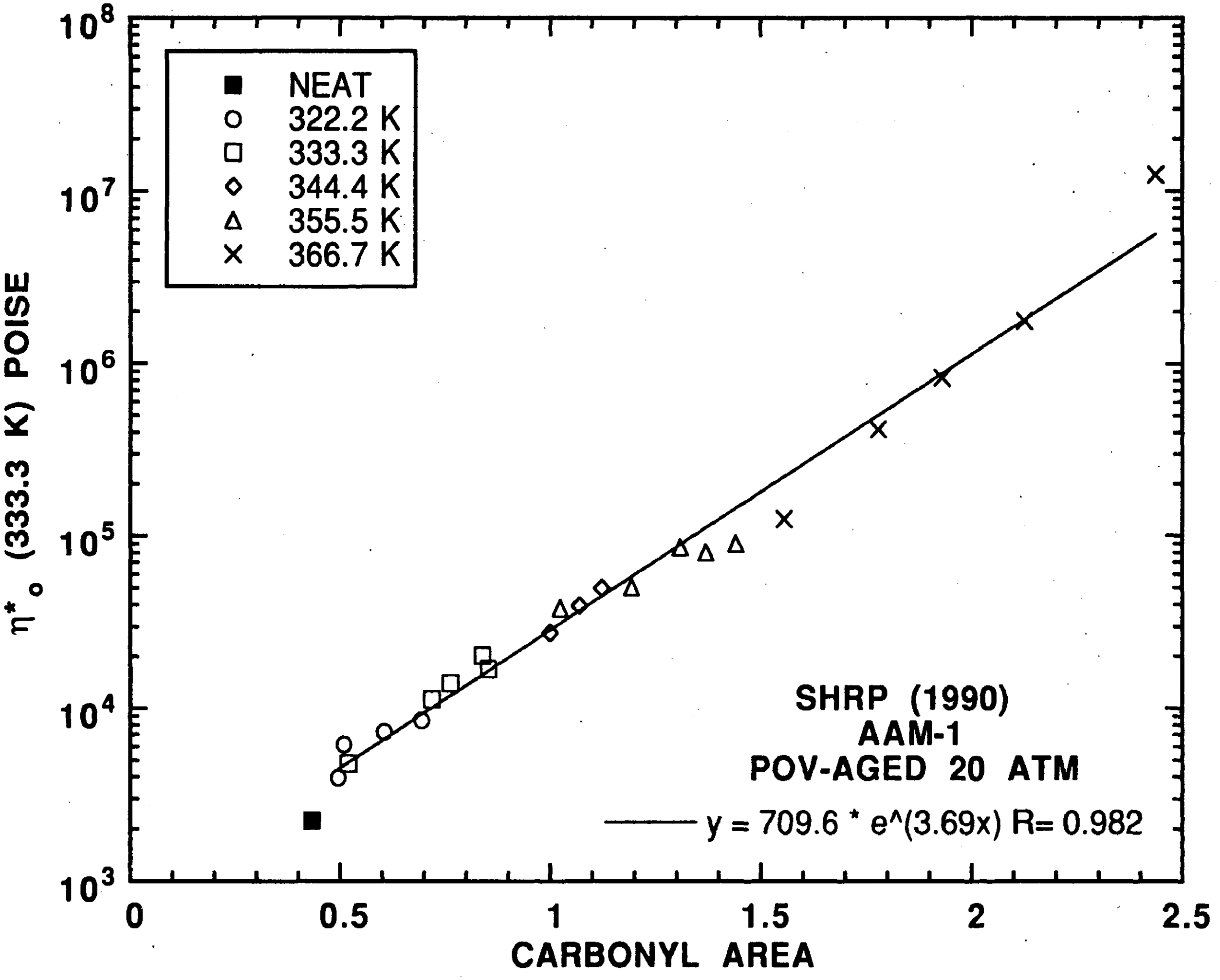

Figure G-27. η_0^* at 333.3 K and *CA* of neat and all short-term POV-aging conditions studied for SHRP (1990) AAM-1.

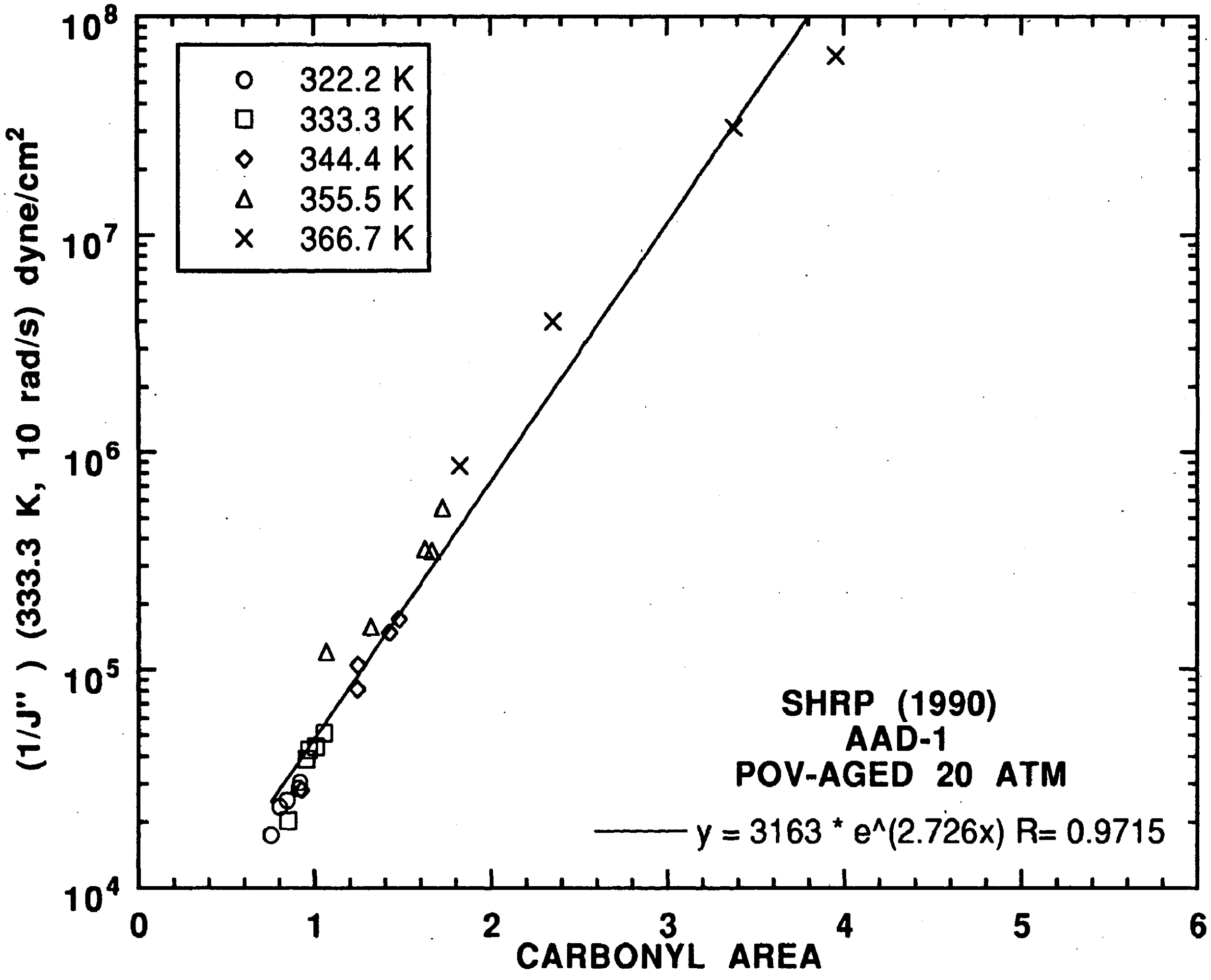

Figure G-28. $(1 / J'')$ at 333.3 K, 10 rad/s and CA of all short-term POV-aging conditions studied for SHRP (1990) AAD-1.

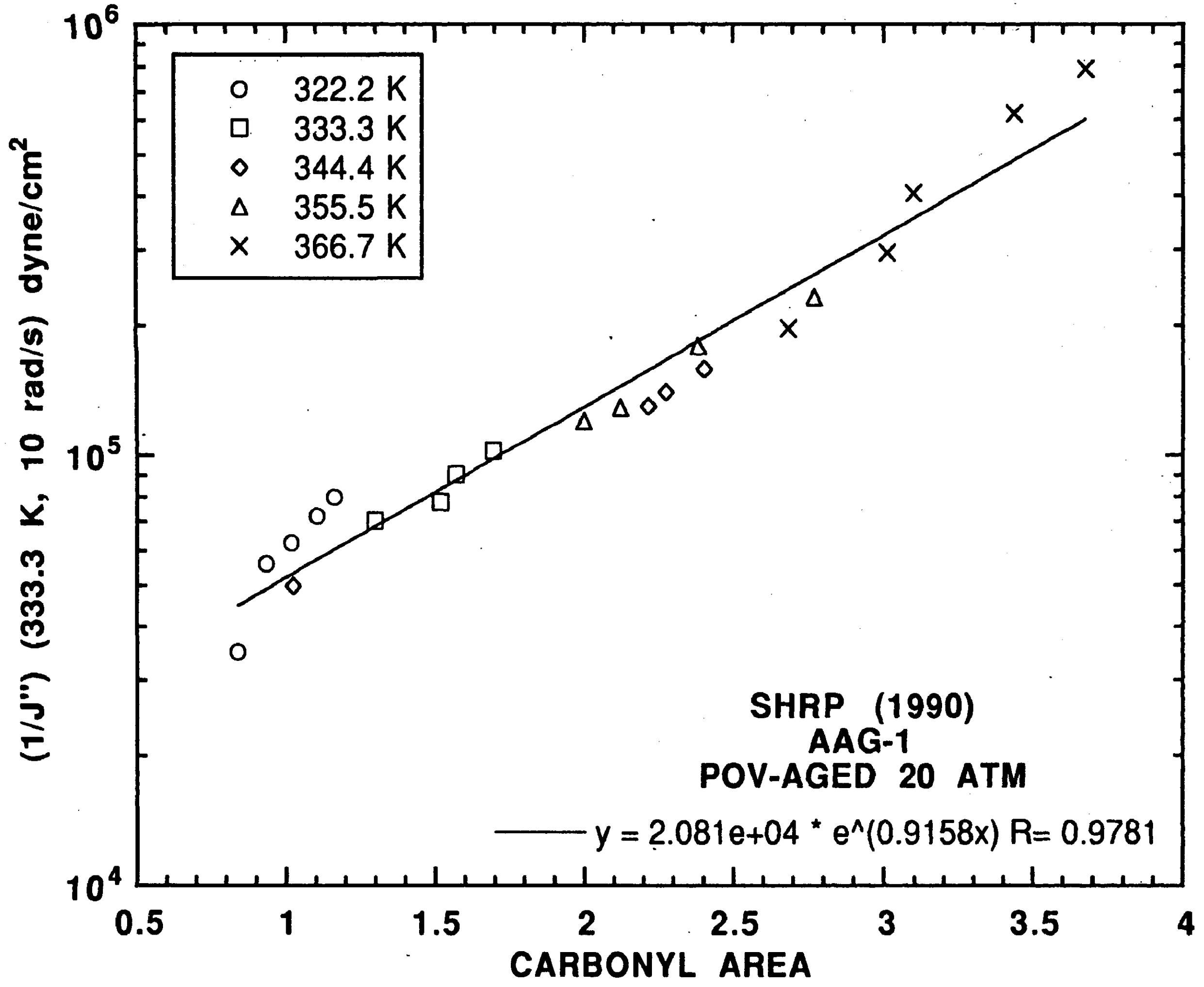

Figure G-29. $(1 / J'')$ at 333.3 K, 10 rad/s and CA of all short-term POV-aging conditions studied for SHRP (1990) AAG-1.

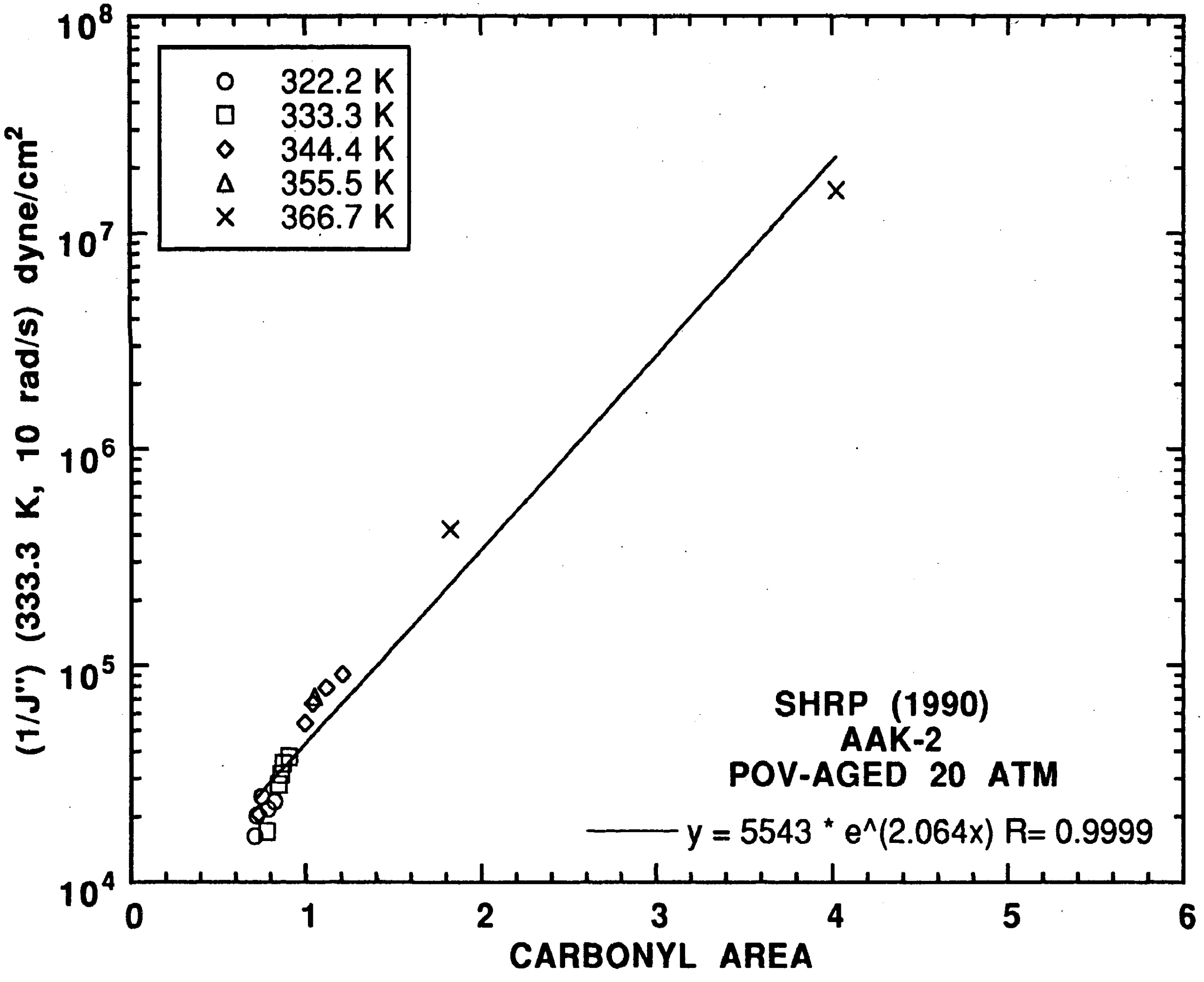

Figure G-30. $(1 / J'')$ at 333.3 K, 10 rad/s and *CA* of all short-term POV-aging conditions studied for SHRP (1990) AAK-2.

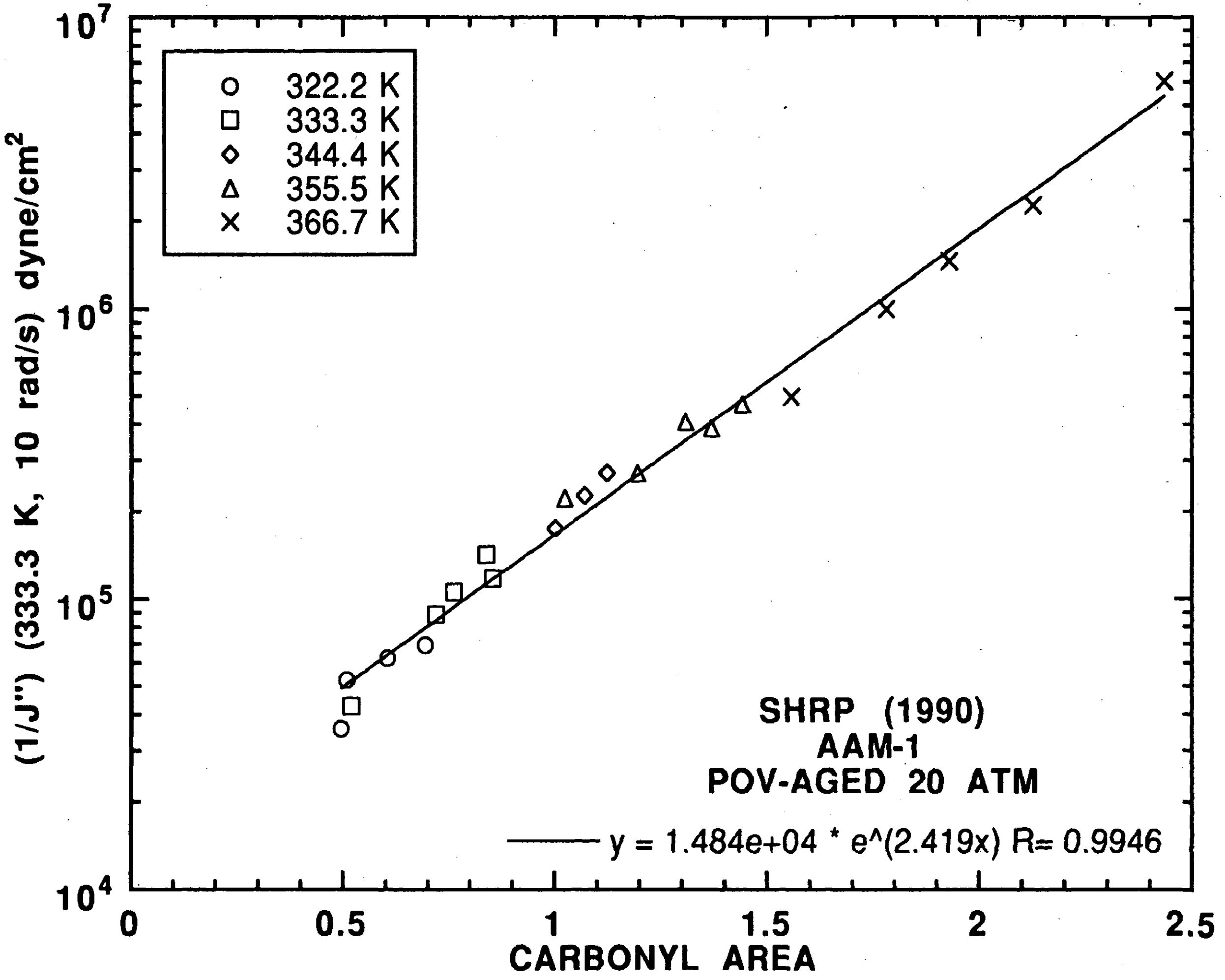

Figure G-31. $(1\ /\ J'')$ at 333.3 K, 10 rad/s and *CA* of all short-term POV-aging conditions studied for SHRP (1990) AAM-1.

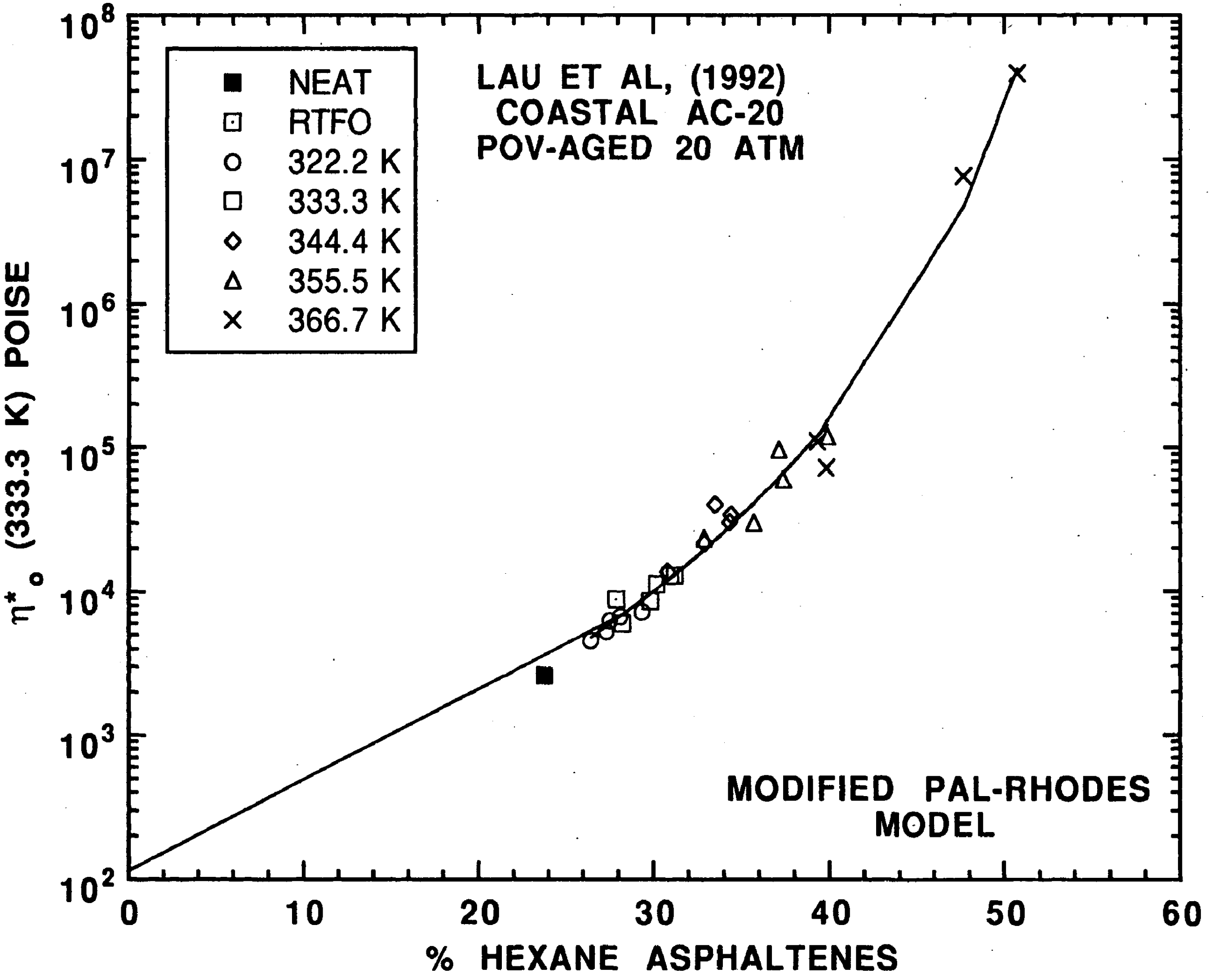

Figure G-32. η^*_0 at 333.3 K and % hexane asphaltenes of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Coastal AC-20.

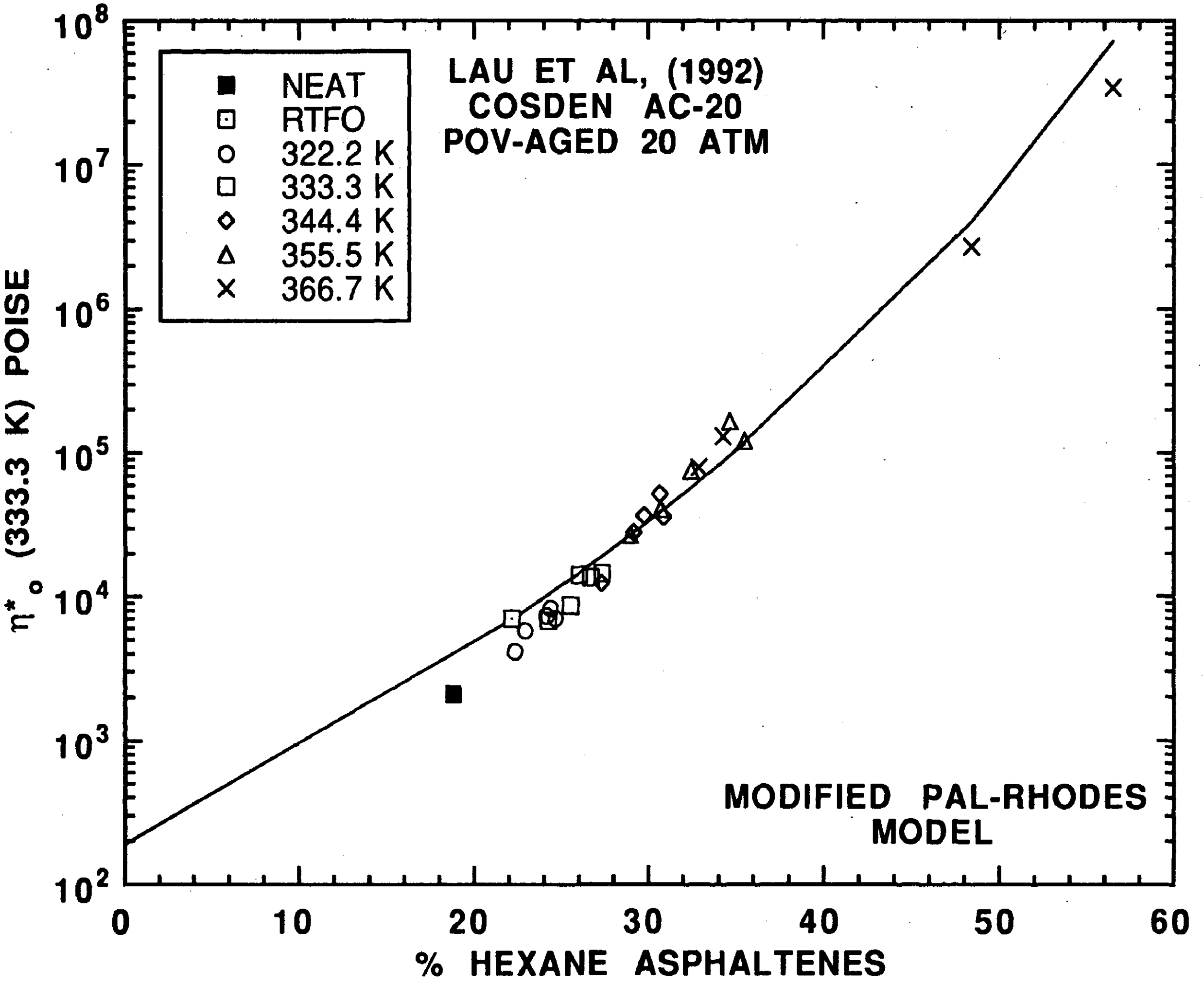

Figure G-33. η_0^* at 333.3 K and % hexane asphaltenes of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Cosden AC-20.

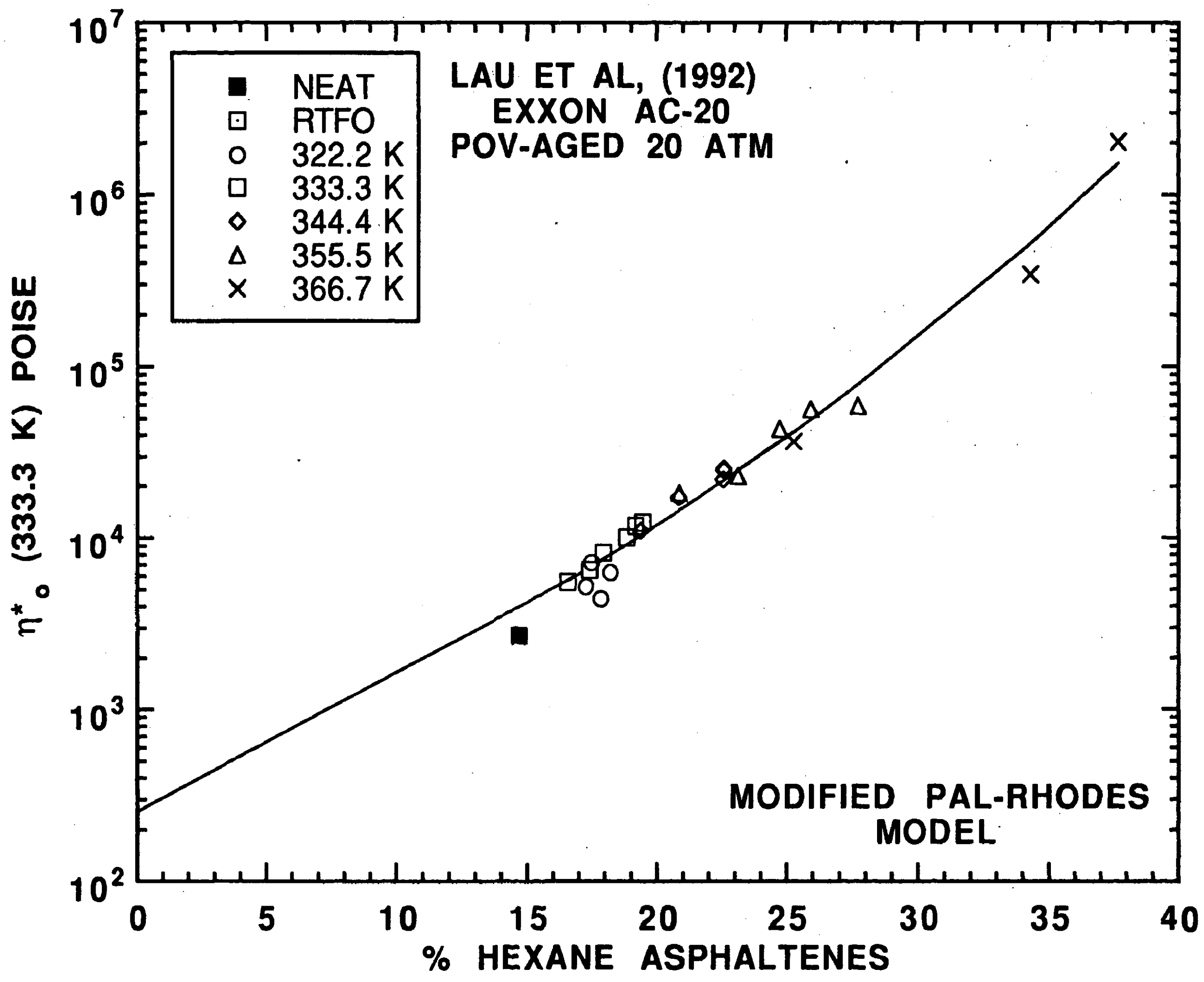

Figure G-34. η_0^* at 333.3 K and % hexane asphaltenes of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Exxon AC-20.

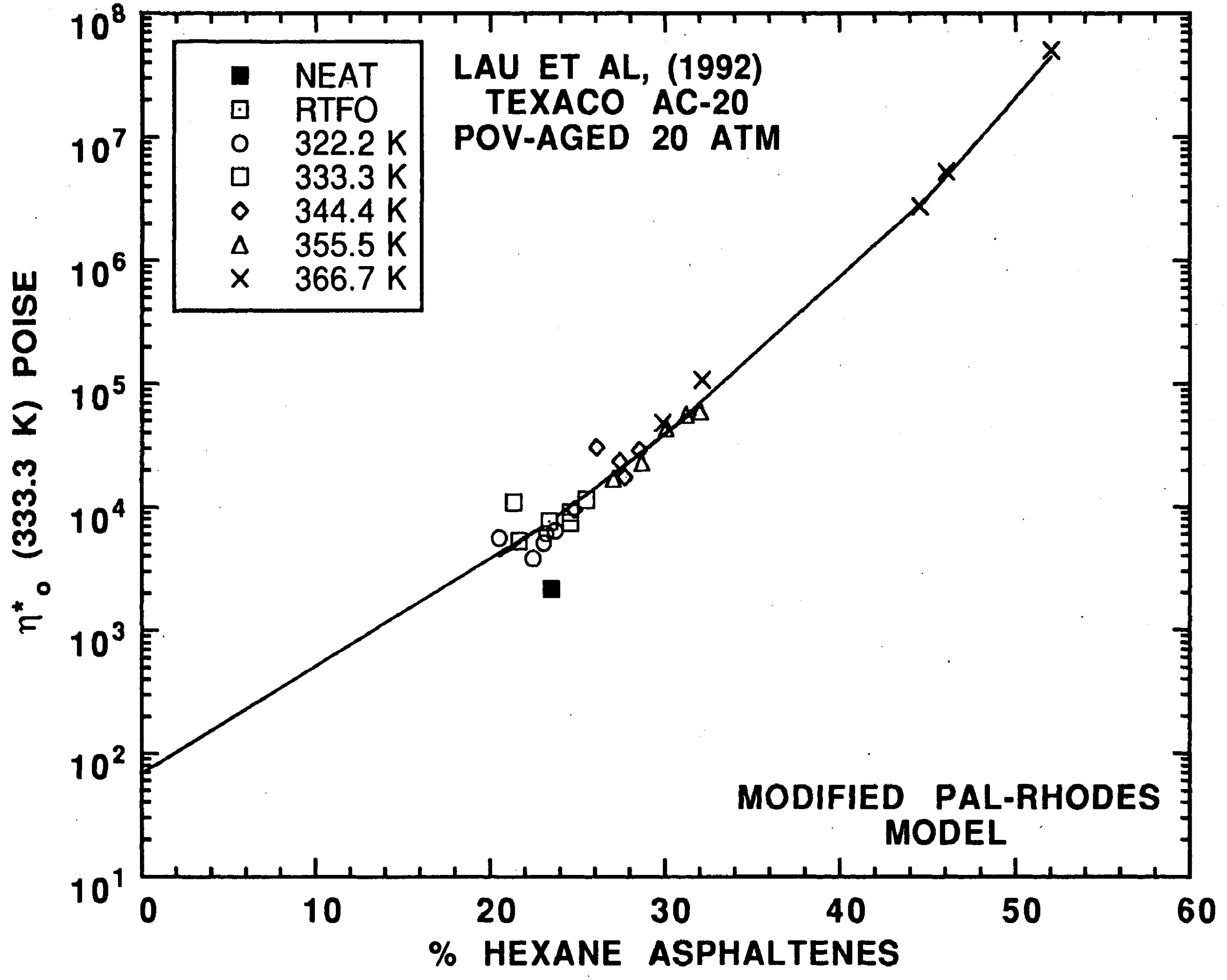

Figure G-35. η_0^* at 333.3 K and % hexane asphaltenes of neat, RTFO, and all short-term POV-aging conditions studied for Lau *et al.*, (1992) Texaco AC-20.