

Sources of Bias in Chemical Warfare Simulation

Mr. Dale Malabarba and Mr. Raymond V. Spring
US Army Natick Research and Development Center

CHEMCAS-II is a computer assisted model which simulates multiple chemical weapon attacks on arrays of military forces. The model estimates chemical casualties and equipment contamination caused by specific weapon system/chemical agent combinations delivered against troop units. Further, this is a one sided chemical warfare model which can portray both offensive and defensive situations. The description of the weapon systems, chemical agents, meteorological conditions, unit deployment, protective postures, and toxicological response of personnel are defined by the user.

Historically, CHEMCAS-II evolved from the TECH models of the 1960's, which were single target/single weapon system models developed to support JTCG/ME studies. CHEMCAS-II, however, was extended to include the use of percent-of-knowledge and target location error (TLE) lookup tables for determining target detection and location errors for input to offline manual fireplanning, a simplified model of munition delivery accuracy, a new model of chemical alarm performance, multiple allowable orientations of wind direction, target facing azimuth, munition line-of-flight, and accounting for non-simultaneous arrival of attacks from different sources.¹

The model throughout its component programs utilizes a Monte Carlo simulation technique. The programs are:

- Main 0 - manages the target arrays and performs the target detection and error computations.
- Main 1 - produces the chemical cloud.
- Main 2 - performs the weapons delivery and attack overlay computations.
- Main 3 - assesses personnel casualties and equipment contamination.
- PRTRTN - produces summary reports for the simulation.¹

The validity of results obtained from CHEMCAS rest explicitly on the accuracy of the user defined inputs and implicitly on the statistical properties of the pseudo-random number generator used in the model. In CHEMCAS, as in any simulation model, results can be misleading and inaccurate due to bias attributable to a bad generator.

Specifically, in CHEMCAS bias caused by the pseudo-random number generator can be introduced in MAIN 0, MAIN 2, or MAIN 3. In MAIN 0 a file of perceived target locations is produced using a version of the generator which produces pseudo-random normal numbers. This file is then used to manually prepare a fireplan for a chemical attack against these perceived target locations. In MAIN 2 the impact points of the chemical rounds are determined using this

generator and the associated ballistic errors. Finally, in MAIN 3 a user defined number of random points is sampled within each target, based on the version producing pseudo-random uniform numbers, to evaluate the dosage and deposition at each of these points. The dosage and deposition values are then used to determine personnel casualties and equipment contamination.

From the preceding discussion, it is apparent that bias can be introduced in each of these programs through the pseudo-random number generator and severely impact the results of the simulation and all subsequent analyses. Therefore, to assess the potential bias in CHEMCAS, the routines used to generate the pseudo-random numbers have been analyzed.

- Pseudo-Random Number Generation -

The CHEMCAS simulation model generates two types of pseudo-random numbers, uniform and normal. The uniform numbers are generated on the interval zero to one and subsequently provide input to the pseudo-random normal number generator. This latter procedure should then produce a standard normal distribution, that is a normal distribution with a mean of zero and a standard deviation of one. Again, these numbers are used in the determination of perceived target locations for fireplanning and, with ballistic error distributions, to locate the impact points of the chemical rounds.

The pseudo-random uniform number generator is an assembly language function, RANDM, called from various programs and subroutines in CHEMCAS. The routine is a basic linear congruent generator which stores the results of the current call for use as the seed in the next function call.

The pseudo-random normal numbers are generated in a loop, where a bivariate normal x,y is produced. Here, RANDM is called twice, the first call for x and the second for y . This loop executes 12 times with subsequent values for x and y being added to the previous values obtained from RANDM. Finally, after the twelfth iteration the number 6 is subtracted from the summed values of x and y . So, for this two dimensional x,y point the x value would be the sum of the odd number calls to RANDM (e.g., 1st call value + 3rd call value + ... + 23rd call value) and the y value the sum of all the even calls. Thus, if this generator has serial correlation problems with even number lags, the results obtained from the pseudo-random normal generator would also be affected.

The validity of these generators has been evaluated by various statistical methods. First, to evaluate the pseudo-random uniform number generator, a program written by Crigler and Shields² of the Naval Surface Weapons Center in Dahlgren, Virginia has been used. This program was used to conduct a series of 11 different statistical tests on a distribution of ten thousand numbers produced by the pseudo-random uniform number generator. These tests are briefly described below:

1. Mean and Variance Tests - Since the uniform distribution has a mean of 0.5 and a variance of $1/12$ a determination of whether the distribution has this mean and variance is done using a t-test for comparing the mean and an F-test for a ratio of the sample and expected variances.

2. Frequency Test - The uniform distribution assumes that all its values are equally distributed over the interval zero to one. In this test, the ten thousand pseudo-random uniform numbers are divided into 100 categories and a chi-square test is performed to determine the equality of the distribution in the categories.
3. Kolmogorov-Smirnov (K-S) Test - Compares the observed cumulative frequency distribution (cdf) of the ten thousand observations in the sample with the expected cdf of an assumed uniform distribution. This test of the goodness of fit will determine how well the pseudo-random uniform numbers actually fit the uniform distribution.³
4. Maximum of T Test - The K-S test is performed on the cdf of 100 categories of the sample to determine how well these categories fit the expected category cdf of the uniform distribution. The rationale of breaking the uniform distribution into different categories is to show that no matter what the size of the category, if the distribution is uniform, the same number of observations is expected in each category.
5. Gap Test - Using the assumption for the uniform distribution that the gap between each of the ten thousand pseudo-random uniform numbers is equidistant, the length of the gap between each number is tested to determine if they meet this equidistant assumption.
6. Poker Test - The 0,1 uniform range is segmented into five equal sized categories. Two hundred sets of five consecutive pseudo-random uniform numbers are then chosen from the sample of ten thousand. The numbers in each category are tabulated and tested to show an equal number of points in each category.
7. Coupon Collector's Test - Analogous to the Poker Test except that the entire sample of ten thousand is tested to show how evenly they are distributed into the five categories.
8. Permutation Test - the original set of ten thousand pseudo-random uniform numbers is segmented into groups of three successively generated numbers. For each of these groups there are six possible permutations of order (i.e. high, medium, low; high, low, medium; etc) representing six different categories. Subsequently, each group of numbers is placed in the appropriate category. Finally, a test is performed to determine if category sizes are equal.
9. Runs Test - A run is defined as a group of successive numbers that either increase or decrease. To demonstrate that this pseudo-random uniform number generator is truly random, there should be no large groups or large number of runs of the same magnitude. The test looks for discrepancies such as a large number of runs or detects an unlikely large run group in the sample.
10. Serial Test for Successive Pairs - The 0,1 range is divided into ten equal sized intervals and then the sample of ten thousand pseudo-random uniform numbers is broken into five thousand pairs. For each number in a pair, a determination is made of the interval in which that number falls. An observation is then entered into a 10 X 10 matrix where the row corresponds to the

interval for the first number in the pair and the column corresponds to the second. Finally, a chi-square test for independence is performed on the five thousand observations in the 10 X 10 matrix, which will determine the randomness of the generator.

11. Serial Correlation Test - The serial correlation between paired observations from the sample of pseudo-random uniform numbers is evaluated for nine different pairing schemes. In these schemes the second value in a pair is lagged from two to ten. Since the numbers should be random, no matter how the numbers are paired there should be no correlation between them. Thus, each of the pairing schemes is tested to determine if the correlation differs from zero. This series of eleven tests has been conducted for ten sets of ten thousand pseudo-random uniform numbers generated by RANDM with different seeds. The results of the testing, shown in Table 1, indicate no reason to reject RANDM as an acceptable pseudo-random uniform number generator.

The pseudo-random normal number generator was subsequently tested through the use of a sample K-S test. The sample size selected for this test was one thousand in order to approximate the number of pseudo-random normal values generated in a typical CHEMCAS simulation.

The overall number of K-S tests performed and samples generated was thirty with the samples coming from ten different seed values. The results, shown in Table 2, demonstrate that five of thirty test sets were rejected at the 0.05 level of significance. This can be interpreted to mean that approximately 16.7 percent of the sets generated were not from a normal (0,1) distribution. Also, the five rejected samples were produced from five different seed values indicating that fifty percent of the input seeds could lead to biased results.

This finding does raise questions about results obtained from studies that use CHEMCAS and, more importantly, dictates that the cause of nonnormality be investigated and an alternative method of generating pseudo-random normal numbers be found. Toward this end, the size of the string of summed pseudo-random uniform numbers was increased to twenty and thirty with the appropriate subtractions being made for each. In both instances, all thirty samples fail the normality test. The results are detailed in Tables 3 and 4.

Further study used the original pseudo-random normal number generating methodology to produce samples from those seeds which caused the normality test failures. These samples can then be normalized by transforming to a mean of zero and standard deviation of 1 (Table 5). This not only eliminates the non-normality, but also indicates that the generator produces samples which are normal but do not have a mean of zero or standard deviation of one. To determine if RANDM is the cause, the test series was rerun using the seeds that produced the nonnormal sample. These seeds passed all tests, leading to the conclusion that the cause of nonnormality rests not with RANDM, but with the technique to generate pseudo-random normal numbers. As a result a new generating technique must be identified.

One possibility is that suggested by Box and Muller⁴ which takes two pseudo-random uniform numbers and transforms them using the log, sine, and cosine functions in the following manner:

$$X = (-2.0 \cdot \log_e U_1)^{\frac{1}{2}} \cos \pi \cdot U_2$$

$$Y = (-2.0 \cdot \log_e U_1)^{\frac{1}{2}} \sin 2\pi \cdot U_2$$

Where U_1 = pseudo-random uniform number 1

U_2 = pseudo-random uniform number 2

This method, when used to generate samples of one thousand pseudo-random normal numbers, can be tested with the K-S test. As Table 6 shows, the number of rejected samples, though reduced, is still in excess of a five percent level and can not be considered acceptable.

A second method, based on the recommendations of Ahrens and Dieter⁵ and implemented in a FORTRAN routine taken from "A Guide to Simulation", by Bradley, Fox and Schrage⁶ was used to generate samples of one thousand pseudo-random normal numbers. Values were generated from different seeds for X, Y, and Z. The results are well within acceptable limits and are shown in Table 7.

In conclusion, the current method for generating pseudo-random normal numbers in the CHEMCAS model is unacceptable and should be revised. Any replacement must consider both run time and the quality of the numbers generated. The second method generates acceptable values. If this routine is fast as well as statistically acceptable, it should serve as a method to replace the existing technique.

This paper should warn simulation modelers not to accept any pseudo-random number generator. Even those adopted from the literature should be tested to ensure their statistical validity. Compilers for different hardware configurations operate with a set of explicit and implicit assumptions which can affect the results obtained from any pseudo-random number routine. Thus, the modelers should ensure the adequacy of their generator.

Table 1

Results from Use of RANDOM Program on 10,00 Numbers Uniformly Distributed

Seed #	critical values	Mean .4943- .5057	S.D. Var.	Frequency	K-S	Max-T	Gap	Poker	Coupon	Permute	Runs	Serial
7147398565		.49827	.08286	123.2253	.0136	.1340	15.5073	9.4877	18.3070	11.0705	12.5916	123.2253
8123098453		.49864	.08444	77.6200	.0075	.0600	6.4139	2.0068	27.6830*	8.5644	6.2411 9.3272	107.1600
9192939477		.49962	.08309	105.7400	.0079	.0700	4.3442	2.5716	18.1141	.9172	5.8449 2.1469	104.7600
0205873195		.50009	.08341	104.0600	.0052	.0900	7.1079	8.3604	12.8949	4.1611	3.7532 3.1709	92.0800
6274531897		.49766	.08322	83.5600	.0058	.0800	4.0982	5.1549	12.1638	5.5545	2.8335 2.1546	96.7600
4321568979		.50110	.08201	96.6400	.0081	.0700	2.0554	3.4281	15.8192	7.7759	7.4109 3.3019	93.2800
3241598763		.50063	.08334	102.6600	.0083	.0500	6.2168	7.8193	12.9987	4.2007	4.5387 3.6160	98.6000
2426282921		.50183	.08357	73.4600	.0053	.0600	4.3644	1.6299	12.3015	6.2457	3.3965 4.7576	79.3200
1092837465		.49645	.08288	86.4600	.0060	.0700	10.4368	2.2698	6.1736	1.7633	1.2649 12.8375*	103.4000
5134267881		.50051	.08315	127.1000	.0102	.0600	5.1206	4.7997	5.6608	4.3375	4.3197 5.0232	81.2800
					.0080	.1000	8.0942	5.0141	12.2439	7.5419	1.1494 3.2145	86.6400

* Significant at 0.05 level

Table 2

Results of Kologorov-Smirnov One Sample Test on 1,000 Normal (0,1) Random Numbers Generated by CHEMCAS Routine Summing 12 Values and Subtracting 6

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
1029384567	.00720	.01058	.99988	.03958	.00915	.08716	.01453	.02764	.42976
7685940315	.00159	.04225	.05631	.01357	.03259	.23867	.02427	.02119	.59773
2126205197	.04617	.01790	.02817	.02121	.01216	.75920	.02271	.01603	.68085
5642971739	.04317	.00161	.04808	.01553	.02300	.66535	.02502	.01310	.55864
0192348765	.01133	.02375	.62531	.00572	.03557	.15923	.02346	.00556	.64090
4178926351	.01088	.02504	.55742	.02236	.01466	.69923	.02008	.01829	.81493
8302957123	.04545	.01833	.03211	.01963	.01130	.83585	.01433	.01435	.98623
3576409123	.02007	.00326	.81544	.01353	.01234	.99307	.02770	.02256	.42691
6153047829	.00346	.02528	.54523	.03185	.00907	.26240	.02106	.04426	.03980
5371928461	.00273	.07133	.00008	.03190	.01774	.26072	.03827	.00644	.10695

PDIF 2 = Largest Positive Difference

PDIF 3 = Largest Negative Difference

PDIF 6 = 2-sided Significance Level

* Significant at 0.05 level

*

Table 3.

Results of Kolmogorov-Smirnov One Sample Test on 1000 Normal (0,1) Random Numbers Generated by CHEMCAS Routine Summing 20 Values and Subtracting 10

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
1029384567	.09130	.07789	.00000	* .06659	.07030	.00010	* .06046	.06827	.00018
7685940315	.05584	.08015	.00001	* .04466	.07864	.00001	* .06668	.06239	.00027
2126205197	.09620	.06027	.00000	* .05853	.07692	.00001	* .06407	.07047	.00010
5642971739	.07750	.05881	.00001	* .05856	.07999	.00001	* .05841	.07669	.00002
0912348765	.06692	.08661	.00000	* .06676	.08587	.00000	* .06317	.08113	.00000
4178926351	.05976	.07446	.00003	* .07441	.05663	.00003	* .07300	.06150	.00005
8302957123	.08822	.06845	.00000	* .05242	.07742	.00001	* .08935	.07098	.00000
3576409123	.07595	.08208	.00000	* .08157	.08029	.00000	* .07515	.07410	.00002
6153047829	.06262	.05977	.00078	* .08408	.04815	.00000	* .06648	.10110	.00000
5371928461	.04945	.11501	.00000	* .08826	.07524	.00000	* .06185	.06361	.00061

PDIF 2 = Largest Positive Difference

PDIF 3 = Largest Negative Difference

PDIF 6 = 2-sided Significance Level

* Significant at 0.05 level

Table 4
 Results of Kolmogorov-Smirnov One Sample Test on 1000 Normal (0,1) Random Numbers Generated
 by CHEMCAS Routine Summing 30 values and Subtracting 15

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
1029384567	.10990	.09728	.00000	.09439	.12294	.00000	.12239	.09695	.00000
7685940315	.10666	.13386	.00000	.09505	.12569	.00000	.11139	.10179	.00000
2126205197	.14832	.11130	.00000	.10658	.11455	.00000	.10660	.11325	.00000
5642971739	.12184	.11598	.00000	.10387	.11908	.00000	.11079	.12712	.00000
0912348765	.10296	.11603	.00000	.10186	.12332	.00000	.11281	.11302	.00000
4178926351	.12668	.09788	.00000	.10178	.08907	.00000	.11927	.10648	.00000
8302957123	.12981	.09486	.00000	.11177	.10988	.00000	.12075	.11380	.00000
3576409123	.11296	.12062	.00000	.11634	.11445	.00000	.12786	.10917	.00000
6153047829	.10958	.09583	.00000	.12276	.10208	.00000	.11594	.14574	.00000
537198461	.08969	.13326	.00000	.11405	.11593	.00000	.12268	.10734	.00000

PDIF 2 = Largest Positive Difference

PDIF 3 = Largest Negative Difference

PDIF 6 = 2-sided Significance Level

* Significant at 0.05 level

Table 5

Results of Kolmogorov-Smirnov One Sample Test on 1000 Normal (0,1) Random Numbers Generated by CHEMCAS Routine Summing 12 values and Subtracting 6 and then normalizing. Seeds used caused Rejections in Table 2.

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
5371928461	.02664	.03254	.24019	.01742	.02304	.66336	.00959	.01305	.99567
6153047829	.01979	.01456	.82815	.01953	.01968	.83336	.02564	.02378	.52679
8302957123	.02992	.02203	.33239	.01581	.01404	.96398	.01744	.01186	.92144
5642971739	.01624	.03163	.26980	.01712	.01493	.93117	.02541	.01315	.53816
2126205197	.02274	.01794	.67910	.00992	.01228	.99819	.02416	.01724	.60354
2655856677	.02672	.01557	.47300	.02362	.01396	.63258	.01730	.01538	.92571
0123456789	.01352	.01052	.99316	.02334	.01365	.64742	.01687	.01555	.93850

PDIF 2 = Largest Positive Difference

PDIF 3 = Largest Negative Difference

PDIF 6 = 2-sided Significance Level

* Significant at 0.05 level

Table 6

Results of Kolmogorov-Smirnov One Sample Test on 1000 Normal (0,1) Random Numbers Generated by Box-Muller Method

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
10289384567	.04549	.01518	.03188 *	.02050	.01029	.79483	.02764	.02281	.42971
7685940315	.01581	.00997	.96390	.01806	.01592	.90023	.02457	.01720	.58181
2126205197	.03194	.01054	.25945	.02567	.01644	.52502	.00637	.02781	.42168
5642971739	.02842	.01479	.39460	.02261	.00471	.68612	.01183	.02336	.64607
0912348765	.03867	.01366	.10043	.02072	.00988	.78392	.01798	.00983	.90309
4178926351	.02464	.01378	.57844	.02278	.01734	.67735	.01219	.01479	.98092
8576409123	.02899	.02366	.34996	.02050	.01777	.79463	.01597	.01981	.82725
6153047829	.01707	.03426	.19106	.03223	.02227	.24987	.01196	.01944	.84412
5371928461	.02799	.01678	.41349	.00428	.04473	.03655 *	.01262	.01476	.98141

PDIF = Largest Positive Difference

PDIF = Largest Negative Difference

PDIF = 2-sided Significance level

* Significant at 0.05 level

Table 7

Results of Kolmogorov-Smirnov One Sample Test on 100 Normal (0,1) Random Numbers Generated by Method of Ahrens and Dieter

Seed #	X			Y			Z		
	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6	PDIF 2	PDIF 3	PDIF 6
1029384567	.01277	.02602	.50727	.01488	.02261	.68603	.02719	.00876	.45063
7685940315	.00948	.01828	.89190	.00329	.02965	.34312	.00446	.02737	.44216
2126205197	.03497	.00440	.17319	.02594	.00586	.51170	.02956	.00953	.34666
5642971739	.01023	.01800	.90230	.02237	.01307	.69903	.02475	.01886	.57275
0192348765	.01018	.03453	.18422	.02294	.02731	.44495	.02370	.01577	.62793
4178926351	.03626	.00879	.14409	.03075	.00619	.30094	.00538	.02806	.41031
8302957123	.01263	.01275	.99685	.00942	.02693	.46306	.02013	.01341	.81269
3576409123	.02631	.02048	.49306	.01912	.01144	.85829	.01076	.01282	.99661
6153047829	.01802	.00995	.90161	.01043	.03278	.23293	.00572	.02780	.42232
5371928461	.02737	.00616	.44210	.00843	.03406	.19637	.05342	.01495	.00665

PDIF 2 = Largest Positive Difference

PDIF 3 = Largest Negative Difference

PDIF 6 = 2-sided Significance Level

* Significant at 0.05 level

- 1 J.L. Burroughs and K. Williams, "Techniques For War Game Assessment of Chemical Operations", BDM Services Company and Vector Research Inc., Leavenworth, Kansas, September 1975, AD B021809.
- 2 J.R. Crigler and P.A. Shields, "RANDOM: A Computer Program for Evaluating Pseudo-Uniform Random Number Generators", Naval Surface Weapons Center, August 1982, NSWC TR 82-93.
- 3 F.J. Massey, Jr., "The Kolmogorov-Smirnov Test For Goodness of Fit", Journal American Statistical Association, Volume 46, p. 68-78.
- 4 G.E.P Box and M.E. Muller, "A Note on the Generation of Random Normal Deviates", Analysis of Mathematical Statistics 29, p. 610-611.
- 5 J.H. Ahrens and U. Dieter, "Computer Methods for Sampling from the Exponential and Normal Distributions", Communications of the ACM 15, p. 873-882.
- 6 P. Bratley, B.L. Fox and L.E. Schrage, A Guide to Simulation, Springer-Verlag, 1983.

PUBLIC RELEASE APPROVAL/DISAPPROVAL
PAPERS - AORS XXIII

RIT P

- REFERENCES: AR340-17 (Release of Information and Records from Army Files)
 AR380-5 (DA Information Security Program Regulation)
 AR380-10 (Policy for Disclosure of Military Information for Foreign Governments)
 AR530-1 (OPSEC)

DATE August 13, 1984

SUBJECT/TITLE OF PAPER Sources of Bias in Chemical Warfare Simulation

SPONSORING COMMAND US Army Natick Research & Development Center
Directorate for Systems Analysis and Concept Development
 ATTN: STRNC-OI (ofc symbol)
Natick, MA. 01760

and/or - AUTHOR(s) Dale Malabarba, Raymond Spring

OFFICE ADDRESS Individual Protection and Chemical Defense Division

Telephone #(AV & COMM) AV-256-4340 COMM(617)651-4340

RELEASE: Pursuant to the above cited references, manuscript/paper has been reviewed and is approved/authorized for public release. Yes No

NONRELEASE: Request from sources outside the DA that invoke the Freedom of Information Act (FOIA) are governed by AR340-17; therefore, rationale for nonrelease for manuscript/paper must be cited. Please attach additional sheet with specific rationale and applicable exemption. For classified papers, provide a copy(ies) of security classification guidelines.

OPSEC review completed. Paper is releasable to Australia, Canada, UK, Republic of Korea, and the Netherlands. Yes No

Dale Malabarba
 Author(s)
Raymond Spring
 Author(s)

16 August 84
 Date
14 August 1984
 Date

Approved - AMSAA Security Officer
Title: _____

Date _____

Not Government Funded

Harold S. Manogian
 Approved - Sponsor's Security Officer

Harold S. Manogian
 Approved - Sponsor's PAO Office

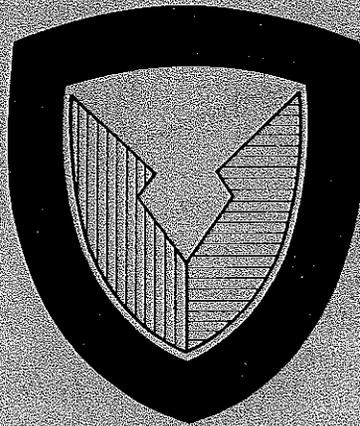
Mark Malabarba
 Approved - Sponsor's FOIA Office

PROCEEDINGS

TWENTY-THIRD ANNUAL
US ARMY OPERATIONS RESEARCH SYMPOSIUM

AORS XXIII

2 OCTOBER - 4 OCTOBER 1984
FORT LEE VIRGINIA



VOLUME II

Distribution limited to US Government Agencies only; test and evaluation October 1984. Other requests for this document must be referred to Director, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, Maryland 21005-5071.

SPONSORED BY

United States Army Materiel Command
Alexandria, Virginia

PROCEEDINGS
OF THE
TWENTY-THIRD ANNUAL
US ARMY OPERATIONS RESEARCH SYMPOSIUM

"EXCELLENCE IN ARMY ANALYSIS"

SPONSORED BY

US Army Materiel Systems Analysis Activity

CO-HOSTS

US Army Logistics Center
Fort Lee, VA

US Army Quartermaster Center
and Fort Lee, Fort Lee, VA

US Army Logistics Management Center
Fort Lee, VA

SYMPOSIUM CHAIRMAN

Mr. Keith A. Myers

ADMINISTRATIVE COORDINATOR

Ms. Glenna D. Tingle