



A11104 190302

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

NIST
PUBLICATIONS

NISTIR 5296

Computer Systems Laboratory

**Computing Effects and Error
for Large Synthetic Perturbation
Screenings**

**Nathalie Drouin
Raghu Kacker
Gordon Lyon**

U. S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899

CMRF

COMPUTER MEASUREMENT
RESEARCH FACILITY
FOR HIGH PERFORMANCE
PARALLEL COMPUTATION

December 1993

~~QC~~

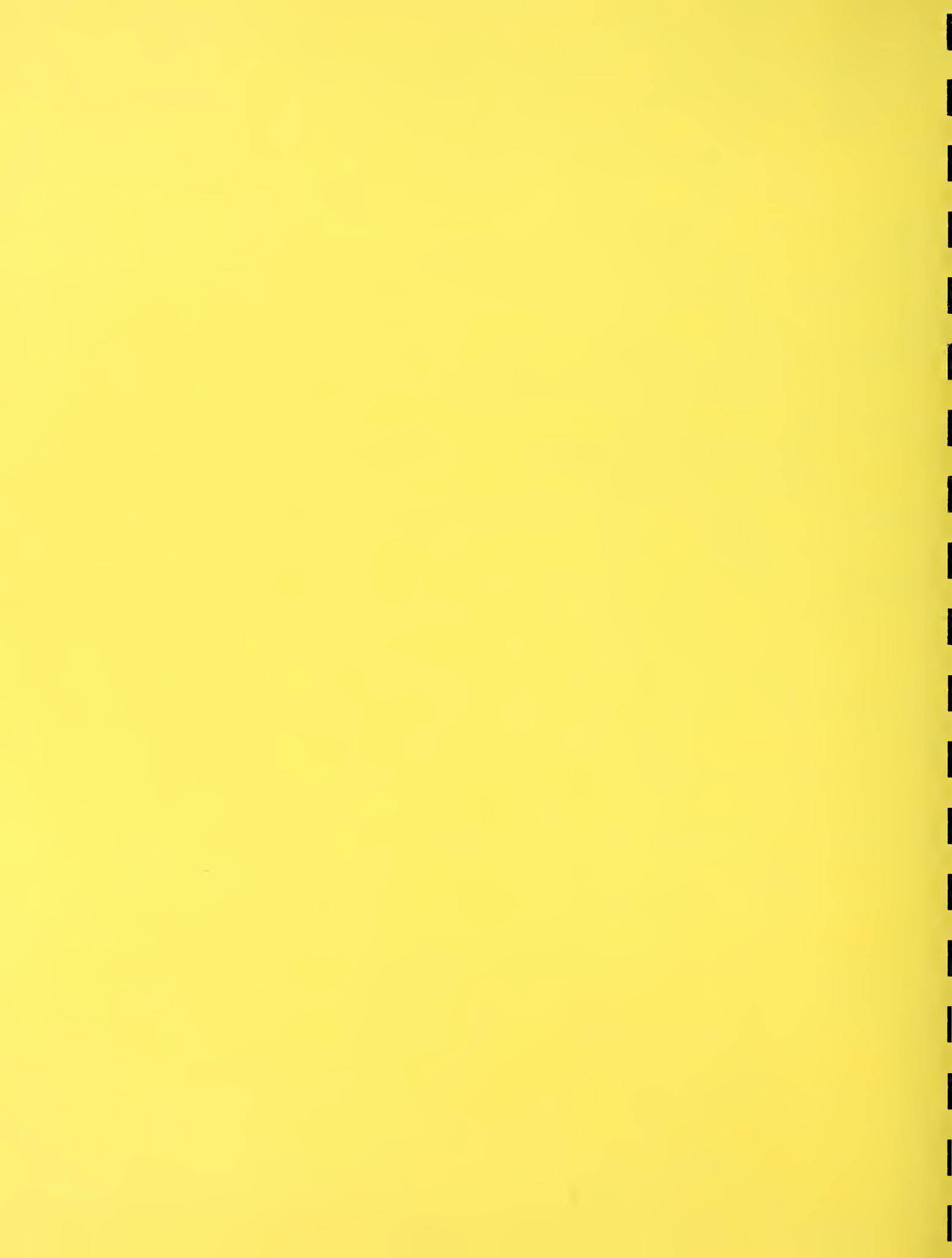
100

.U56

#5296

1993

Partially sponsored by the
Advanced Research Projects Agency



Computing Effects and Error for Large Synthetic Perturbation Screenings

**Nathalie Drouin
Raghu Kacker
Gordon Lyon**

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

December 1993



**U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary**

**TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology**

**NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director**

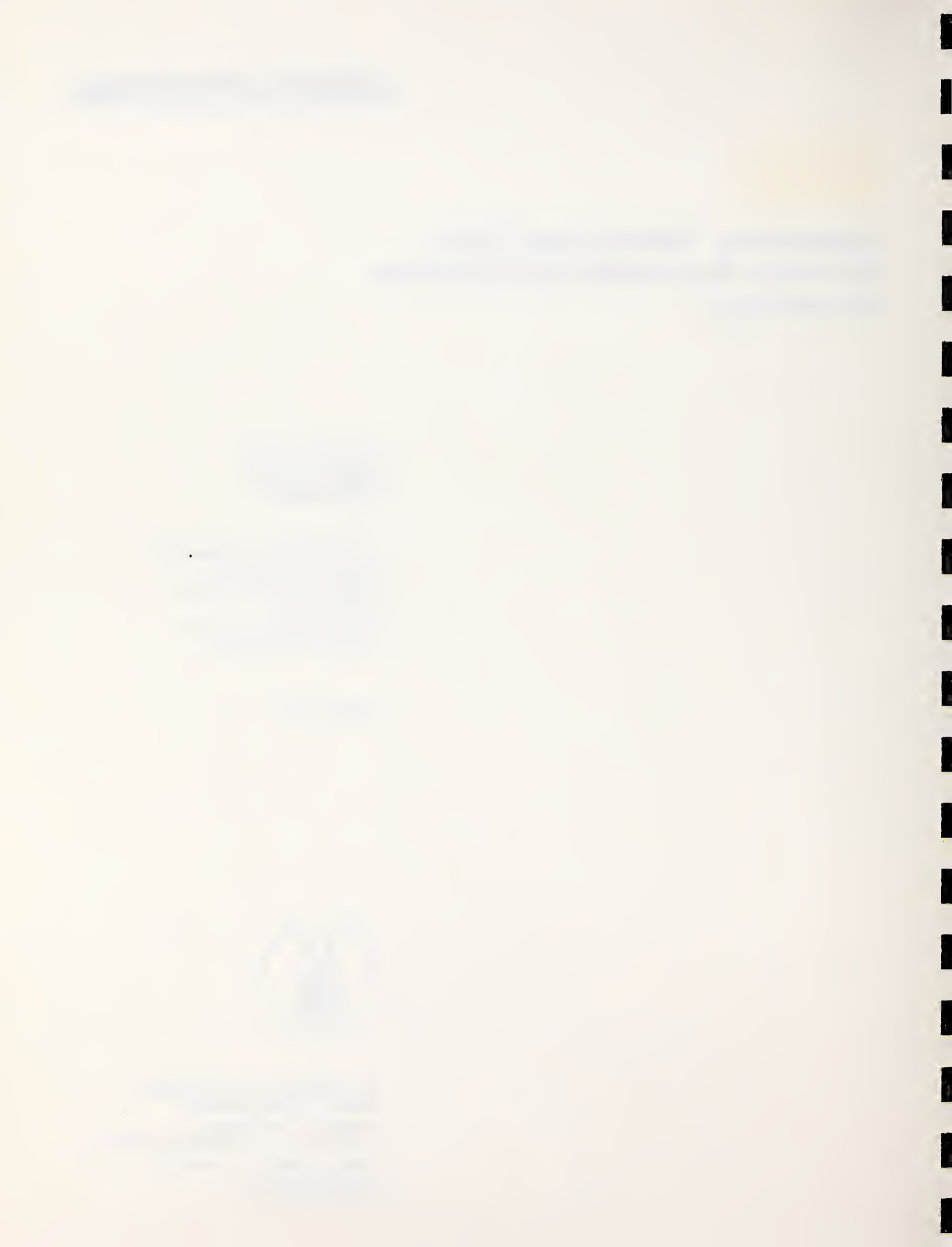
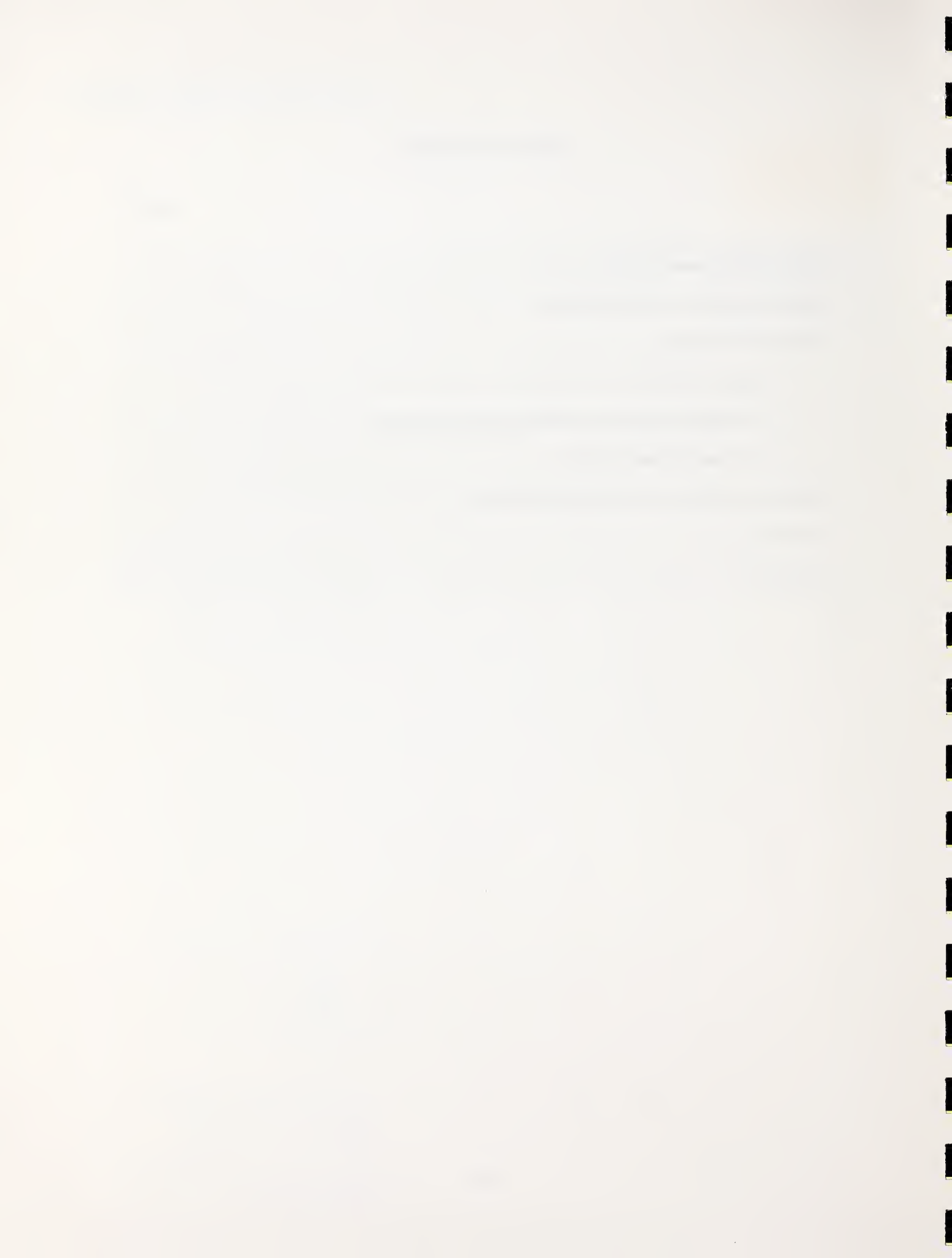


TABLE OF CONTENTS

	Page
1. Design of Experiments and SPS	1
2. Experimental Plans and Factorial Design	2
3. Computation of Effects	5
3.1 Effect of a Factor	5
3.2 Calculating Effects Using a Table of Contrast Coefficients	6
3.3 Variance and Standard Error	7
4. Hadamard Matrices and the Computation of Effects	9
5. Conclusion	11
6. References	12



Computing Effects and Error for Large Synthetic Perturbation Screenings

Nathalie Drouin *
Raghu Kacker
Gordon Lyon

Synthetic Perturbation Screening (SPS) is a powerful, statistically-based method of performance improvement for parallel computer programs. SPS uses designed statistical experiments to identify segments of code that consume significant computing resources or otherwise impede a parallel application. Analysis typically starts with a preliminary screening of a large number of code segments; SPS does this via fractional factorial designs based upon Hadamard matrices. Since Hadamard matrix designs are not commonly discussed in introductory texts, there is a need to survey briefly the calculation of their effects and especially, their standard error (S_p). The result is a practical, supplementary sketch that should help users of SPS automatic tools understand how investigations and analyses are being performed.

Key words: Analysis of Variance; Effect; Factorial Designs; Fold-Over; Hadamard Designs; Parallel Programming; Screening; Standard Error; Synthetic Perturbation Screening.

1. Design of Experiments and SPS

Statistically designed experiments (DEX) are strategies to maximize learning at minimum cost. A typical DEX involves a number of input variables called factors and one or more output variables called responses. The possible settings of the input variables delineate a multi-dimensional experimental region. This region is usually quite large. In its simplest form a DEX specifies the test points in the experimental region at which the response is evaluated. Combinatorial mathematics is used in specifying the test points, thereby ensuring a geometrically balanced coverage of the experimental region. A map of the response thus obtained provides an estimate of relationships between factors and responses. (Although certain tactics called randomization, blocking and replication are used in DEX to enhance the validity of the response map, these are not discussed here.) While DEX are usually used in physical science experimentation, their use in computer experiments is gaining momentum. The key difference is that in computer experiments, responses arise from execution of a computer program.

* N. Drouin is a visiting scientist with the Advanced Systems Division.

The DEX approach is an inexpensive, empirical alternative to analytical methods based upon a thorough understanding of the system under study. Cost alone is enough reason to try DEX as an initial strategy. The DEX statistical strategy can be applied to almost any system; it has been found to be effective even in the most complex multi-variable systems. Many analytical approaches tend to lose effectiveness as the complexity of a system increases. DEX do not. This attractive scaling property is of paramount importance for parallel systems and their study.

Objectives of a designed experiment include the following :

- 1 - Isolate important factors from many other candidates.
- 2 - Identify better, more promising, operating conditions for each factor.
- 3 - Identify better combinations of test settings.
- 4 - Map the relation between input factors and output response.
- 5 - Identify significant interactions and curvature effects of the factors.
- 6 - Find factors which affect variability and settings which minimize variation.

Primary focus here will be upon objective 1, which is called *screening*.

Synthetic Perturbation Screening, SPS, is a methodology for parallel programming sensitivity analysis [4]. It relies upon DEX for screening large sets of factors. Although the underlying statistical theory can be complex, the basic DEX application in SPS is simple to describe. A parallel program is treated as a *grey box* with *input parameters* and *outputs*. *Input parameters* are arguments to artificial perturbations (usually delays) that have been inserted into the program at potential *bottlenecks*. Bottlenecks are segments of code that impede computing. *Outputs* are measurable attributes of an execution, *e.g.*, program run times. The experimenter first chooses a number of factors for study. Each factor corresponds to a given location in the source code that might be a bottleneck. He then introduces synthetic perturbations (*e.g.*, artificial delays) at the factor's location. This is done in systematic patterns. A succession of such patterns constitutes a statistical plan of (parameter) search. By deliberately varying the perturbations (treatments) of the chosen factors in a planned manner, the experimenter can deduce the effects and interactions of the treatments on the program response. This information identifies performance bottlenecks.

2. Experimental Plans and Factorial Design

Given a sufficiently restricted set of input variables, one mainly needs a DEX investigation strategy that is reliable and accurate. In such circumstances, it can be assumed that all the inputs are important in varying degrees. Objective 5, above, would be such an instance. However, larger screenings (objective 1) as used in SPS definitely require DEX approaches that are, in addition, efficient. Thus, the traditional one-factor-at-a-time (1-FAT) method of analysis would be inordinately time consuming. Moreover, it can lead to incorrect conclusions, for 1-FAT does not capture information about interactions among factors that affect the measured response.

Full factorial designs are statistical designs for studying many different factors. They permit estimation of all main effects and interactions, but require a prohibitively large number of measurements when there are many factors. Consider for example, a two level design in 10 factors. Each factor assumes two levels (*e.g.*, *high* and *low*). Full factorial experiment involves $2^{10} = 1024$ different patterns of factor settings. With each pattern corresponding to a different run, this is a serious impediment, as the aim of SPS is

to give results from as few runs as possible. Furthermore, this design permits estimation of *all* interactions. Quite often high-order interactions are of negligible magnitude when compared to main effects and low-order interactions. Thus, a full factorial design is usually not appropriate for screening large sets of factors. One is mainly concerned with identifying and discarding factors that have no significant effects.

Fractional factorial designs trade knowledge of high-order interactions for a much diminished number of runs. Consequently, these designs may be viewed as efficient ways for conducting factorial experiments from a reduced set of measurements. The major assumption is that many interaction terms are not significant and can be ignored. Two-level fractional factorial designs can be generated from Hadamard matrices [1,10,13,14,15,16]. Hadamard matrices are intimately connected to factorial experiments in which each factor is at two levels; therefore they can be used in SPS (see Figure 1, next page) where treatment levels typically take two nominal values '-' (low) or '+' (high). The minimum number of runs required to conduct a meaningful investigation on the previous 10 factors example then drops from 1024 (full factorial design) to 12.

The greater the number of factors, the more revealing the economy in runs. The table below contrasts runs of a meaningful experiment for a *k*-factors model using full factorial and Hadamard designs. Resolution 3 designs based on Hadamard matrices will yield main effects assuming *all* interactions are negligible, while resolution 4 designs will yield main effects even when two-factor interactions are present. Note that resolution 3 designs confound main effects with two-factor interactions while designs of resolution 4 do not. Interactions will demand further runs, but the assumption is that with large sets of factors, this would be rare.

k-Factors	Minimum Number of Runs	
	Full Factorial	Hadamard Designs
		<i>res 3</i> <i>res 4</i>
2	4	4 4
3	8	4 8
4	16	8 16
5	32	8 16
10	1024	12 24
20	2 ²⁰	24 48
30	2 ³⁰	32 64
50	2 ⁵⁰	52 104
100	2 ¹⁰⁰	112 224
.....	...	
k ^{2^k}		Least integer <i>n</i> divisible by 4 such that $n \geq (k + 1)$ for <i>res 3</i> . For <i>res 4</i> # of runs is <i>n</i> for <i>k</i> = 2 and 2 <i>n</i> for <i>k</i> > 2.

Table 1.

In general, experiments are performed to measure the effects of one or more factors on a response. In the next section, a small example of a two-level, full factorial design with 3 factors illustrates how effects can be estimated. This exemplifies what SPS might do after screening and discarding a large number of factors. The example furthermore illustrates why larger sets of screened factors must be treated differently.

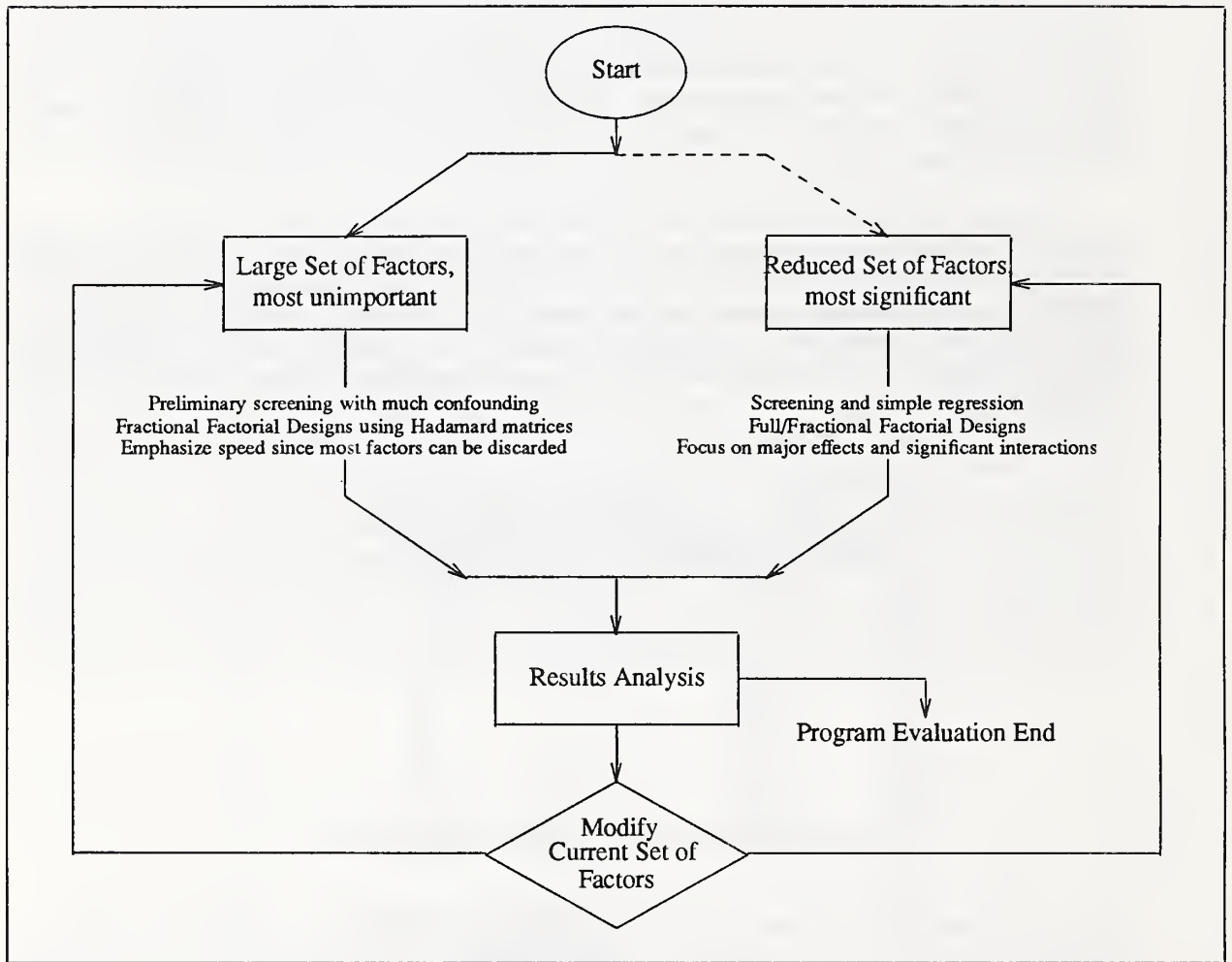


Figure 1. SPS Analysis Flow

3. Computation of Effects

3.1 Effect of a Factor

Let Prog1 be a MIMD program with 3 suspected bottlenecks: F1 (*for* loop), F2 (critical section) and F3 (*while* loop). A two-level full factorial plan with $k=3$ controlled factors is an arrangement consisting of $2^3 = 8$ possible combinations. Each factor may take two treatment levels, denoted by '-' and '+' respectively. Each of the 2^k combination is a run. The following 2^3 plan indicates how a 2^k plan can be written for any value of k .

Order of Running	Run	F1	F2	F3	Response, R
<i>random order</i>	1	-	-	-	17.05
	2	-	-	+	19.62
	3	-	+	-	23.19
	4	-	+	+	25.61
	5	+	-	-	17.08
	6	+	-	+	19.71
	7	+	+	-	23.34
	8	+	+	+	25.71
	Effect	0.09	6.09	2.49	

Table 2.

The headings in the above table refer respectively to the three factors F1, F2 and F3, the run identification (not order of running) and the measured response corresponding to the run. The entries in each column refer to the treatment levels of the factor (*i.e.*, high or low) associated with that column. The rows define the $2^3=8$ runs. The response, R , is the running time of the program under study.

The effect of a factor is a measure of the change in the response as we move from the '-' to the '+' treatment level of that factor. The main effect of a factor is the difference between two average responses, one corresponding to the treatments which have the '+' level of the factor and the other corresponding to the treatments which have the '-' level of the factor. Thus, the main effect of factor F3 is the average response for treatments 2, 4, 6, 8 minus the average response for treatments 1, 3, 5, and 7.

$$Effect(F3) = \left[\frac{(19.62+25.61+19.71+25.71)}{4} - \frac{(17.05+23.19+17.08+23.34)}{4} \right] = 2.49$$

Note that the algebraic sign of a main effect depends upon the arbitrary labeling of the two treatment levels. As a consequence, the absolute value of a main effect is a more meaningful measure of the relative importance of that factor. The sign of the effect is, of course, still needed to know if a factor treatment

degrades or improves the observed response.

Two common methods for quickly calculating effects are the Yates algorithm (*cf.* [9], [11]) and the table of contrast coefficients. The next section focus on the latter, which is more intuitive and applicable to the use of Hadamard matrices.

3.2 Calculating Effects Using a Table of Contrast Coefficients

Once factors of interest have been selected, systematic patterns of treatment levels are used to carry out investigations in a planned manner (see *Table 3*). As shown in 3.1, the effects of factors under study (or main effects) can be estimated in a very straightforward manner.

Run	F1	F2	F3	Response, <i>R</i>
1	-	-	-	17.05
2	-	-	+	19.62
3	-	+	-	23.19
4	-	+	+	25.61
5	+	-	-	17.08
6	+	-	+	19.71
7	+	+	-	23.34
8	+	+	+	25.71
Effect	0.09	6.09	2.49	

Table 3.

Although an estimate of the effects of interactions among factors on the measured response is not directly available from the above table, these can easily be computed using a table of signs, as in the one below:

Treatment	F1	F2	F3	F12	F13	F23	F123	Response, <i>R</i>
1	-	-	-	+	+	+	-	17.05
2	-	-	+	+	-	-	+	19.62
3	-	+	-	-	+	-	+	23.19
4	-	+	+	-	-	+	-	25.61
5	+	-	-	-	-	+	+	17.08
6	+	-	+	-	+	-	-	19.71
7	+	+	-	+	-	-	-	23.34
8	+	+	+	+	+	+	+	25.71
Effect	0.09	6.09	2.49	0.03	0.00	0.10	0.02	

Table 4.

Signs for the interactions are obtained by multiplying the signs of their respective factors. Thus, the column of signs for the F12 interaction is obtained by multiplying together the signs for F1 and F2.

In a general manner, k being the number of factors under study, the total number of possible p -factor interaction combinations is:

$$C_p^k = \frac{k!}{p!(k-p)!}$$

Thus, the total number of possible interactions is $\sum_{p=2}^k C_p^k$.

For example, there are exactly $C_2^3 = \frac{3!}{2!(3-2)!} = 3$ possible 2-factor combinations in 3 factors and a total of 4 possible interactions listed altogether.

The effect associated with column c_i is computed as follows:

$$Effect(c_i) = \left[\frac{\sum_{i=1}^q x_i * R_i}{q/2} \right]$$

where x_i , R_i and q are respectively the treatment level of the factor associated with F_i , the response corresponding to the run number i and the total number of runs. For example, the estimated effect for the F123 interaction is:

$$Effect(F 123) = \left[\frac{-17.05+19.62+23.19-25.61+17.08-19.71-23.34+25.71}{8/2} \right] = \frac{0.11}{4} = 0.02$$

Table 4 is called a *table of contrast coefficients*. The next section shows how effects can be used to identify factors that consume significant computing resources.

3.3 Variance and Standard Error

To decide which factor has a significant influence on the response R , it is necessary to evaluate the effects against background noise. Noise is measured by *standard error*, S_p , or *standard uncertainty*. Under certain assumptions it is possible to calculate the standard error for effects using higher-order (involving 3 or more factors) interactions. In particular, if interactions between factors are negligible, then these interactions measure differences arising principally from experimental error. This is the usual case in SPS.

For example, going back to Table 4, an estimate of the error can be obtained using columns corresponding to interactions between two and three factors, respectively F12, F13, F23 and F123. By doing so, they are assumed to be insignificant.

In the model used in SPS, the noise coefficient has an assumed normal distribution centered about zero. An estimated value for the variance of an effect of Table 4 is thus given by:

$$S_p^2 = \frac{\sum_{j=1}^m effect_j^2}{m} = \frac{(0.03^2 + 0^2 + 0.1^2 + 0.02^2)}{4} = 0.003$$

where m is the number of insignificant interactions. The estimated standard error of an effect is equal to the square root of the variance of the effect. Therefore:

$$S_p = \sqrt{0.003} = 0.05$$

From an statistical viewpoint, an effect is considered significant if it exceeds 2 or 3 times S_p . Anything within $\pm 1S_p$ is likely to be just noise.

Following the SPS methodology, segments of codes (*i.e.*, factors) or interactions are ranked quantitatively according to their sensitivity to perturbation. *SPS rank* is an ordered list of the magnitude of the main effects. The more important the main effect of a factor, the higher its SPS rank. Consequently, such code is more sensitive to perturbation.

Consider the following SPS rank taken from Table 2:

Factor	Effect	Rank
F2	6.09	1
F3	2.49	2
F1	0.09	3

$$S_p = \pm 0.05$$

Statistically, effects that are more than 3 times S_p can be called significant. Thus, in the above example, F2 and F3 are significant while F1 is not. However, statistical significance may not be a relevant criterion to decide whether or not a factor is a program bottleneck. To make the final decision, the programmer first needs to look at the potentially significant bottlenecks based on the ranking and then decides from the nature of the program segment whether a segment is indeed a candidate for further study.

As previously mentioned, Hadamard matrices are one possible scheme to generate statistical plans of analysis for SPS screenings of many factors. The number of factors may range up to 199 [1]. The next section gives a brief overview on the computation of effects with this particular type of design.

4. Hadamard Matrices and the Computation of Effects

Let Prog2 be a MIMD program with four potential *bottle-necks*: F1, F2, F3 and F4. Using the SPS methodology, systematic patterns of delays are produced once locations of interest have been identified. Hadamard matrices are one possible scheme that may be used to generate statistical plans of analysis. With $k=4$ being the number of selected locations in Prog2, the order n of the Hadamard matrix to be created is chosen as follows [1, p. 16]:

$$n \text{ is the first integer divisible by 4 such that } n \geq (k+1)$$

To evaluate Prog2, with a minimum number of runs, the smallest possible plan is used. Here, it is built from a Hadamard matrix of order 8. The corresponding plan is given in the table below.

Mean, μ	F1	F2	F3	F4	c1	c2	c3	Response, R
+	+	+	+	+	+	+	+	r1
+	-	+	-	+	-	+	-	r2
+	+	-	-	+	+	-	-	r3
+	-	-	+	+	-	-	+	r4
+	+	+	+	-	-	-	-	r5
+	-	+	-	-	+	-	+	r6
+	+	-	-	-	-	+	+	r7
+	-	-	+	-	+	+	-	r8

Table 5.

{ R } are the measured responses for Prog2. R varies about an overall mean μ computed with the first column of Table 5. In a Hadamard matrix of a given order n at most $n-1$ columns may be assigned a factor. Thus, with 4 factors and a column assigned to the computation of the mean, three left-over columns remain respectively $c1$, $c2$ and $c3$. Each of those 3 remaining columns measure differences arising principally from experimental error as long as interactions are negligible. Let $eff1$, $eff2$ and $eff3$ be the effects associated with $c1$, $c2$ and $c3$. An estimated value for the variance of an effect is given by:

$$S_p^2 = \frac{\sum_{j=1}^m effect_j^2}{m} = \frac{effect_{c1}^2 + effect_{c2}^2 + effect_{c3}^2}{3}$$

where j is an index over m residual columns.

The estimated standard error, S_p , is then obtained by taking the square root of the variance, S_p^2 .

If no restriction applies on the number of runs that can be performed, measurement accuracy may be increased by using a *fold-over* technique. By doing so, two-factor interactions are no longer confounded with main effects. With fold-over, the statistical plan includes both the initial design and its negative. Consequently, an evaluation of Prog2 with fold-over involves 16 runs: 8 runs correspond to the generated Hadamard matrix (Table 5), plus 8 runs that correspond to its negative (Table 6). The negative of a matrix is its signed opposite. This technique doubles the necessary number of runs.

Mean, μ	F1	F2	F3	F4	c1	c2	c3	Response, R
-	-	-	-	-	-	-	-	r9
-	+	-	+	-	+	-	+	r10
-	-	+	+	-	-	+	+	r11
-	+	+	-	-	+	+	-	r12
-	-	-	-	+	+	+	+	r13
-	+	-	+	+	+	+	-	r14
-	-	+	+	+	+	-	+	r15
-	+	+	-	+	-	-	+	r16

Table 6. Negative Entries for Fold-Over

When using fold-over, variance and standard error can be estimated in two ways, using: (i) left-over columns (*i.e.*, c1, c2 and c3), (ii) a table of contrast coefficients. However, without fold-over the table of contrast coefficients cannot be used because main effects are confounded (mixed) with two-factor effects.

To generate a table of contrast coefficients from an Hadamard matrix (see section 3.2), only columns of the plan associated with a factor are used. The table obtained for the evaluation of Prog2 is:

F1	F2	F3	F4	F12	F13	F14	F23	F24	F1234	Response, R
+	+	+	+	+	+	+	+	+			+	r1
-	+	-	+	-	+	-	-	+			+	r2
+	-	-	+	-	-	+	+	-			+	r3
-	-	+	+	+	-	-	-	-			+	r4
+	+	+	-	+	+	-	+	-			-	r5
-	+	-	-	-	+	+	-	-			-	r6
+	-	-	-	-	-	-	+	+			-	r7
-	-	+	-	+	-	+	-	+			-	r8
-	-	-	-	+	+	-	+	+			+	r9
+	-	+	-	-	+	-	-	+			+	r10
-	+	+	-	-	-	-	+	-			+	r11
+	+	-	-	+	-	+	-	-			+	r12
-	-	-	+	+	+	+	+	-			-	r13
+	-	+	+	-	+	-	-	-			-	r14
-	+	+	+	-	-	+	+	+			-	r15
+	+	-	+	+	-	-	-	+			-	r16

Table 7

Variance and standard error are estimated using effects computed on higher order interactions (see 5.3).

Note that the size of the table of contrast coefficients geometrically increases with the number of factors. As a result, the amount of necessary computation to produce the table rapidly becomes overwhelming, making the implementation inappropriate for a large number of factors. However, it is still possible to get an estimate of the error using a limited version, which is illustrated below. The generation of combinations is restricted to the multiplication of the first column of the plan (that one associated with the mean) with each column associated with a factor. In the present case, this gives:

Mean, μ	F1	F2	F3	F4	c1	c2	c3	c4	Response, R
+	+	+	+	+	+	+	+	+	r1
+	-	+	-	+	-	+	-	+	r2
+	+	-	-	+	+	-	-	+	r3
+	-	-	+	+	-	-	+	+	r4
+	+	+	+	-	+	+	+	-	r5
+	-	+	-	-	-	+	-	-	r6
+	+	-	-	-	+	-	-	-	r7
+	-	-	+	-	-	-	+	-	r8
-	-	-	-	-	+	+	+	+	r9
-	+	-	+	-	-	+	-	+	r10
-	-	+	+	-	+	-	-	+	r11
-	+	+	-	-	-	-	+	+	r12
-	-	-	-	+	+	+	+	-	r13
-	+	-	+	+	-	+	-	-	r14
-	-	+	+	+	+	-	-	-	r15
-	+	+	-	+	-	-	+	-	r16

Table 7.

Variance and standard error are again obtained by using effects associated with the generated columns (*i.e* c1, c2, c3 and c4).

5. Conclusion

Preliminary steps in Synthetic Perturbation Screening, SPS, often involve the statistical screening of 30-60 factors to discover which few are important. At this early stage of analysis, one is primarily concerned with discarding factors that have no significant effect on the program responses of interest. To handle such large screenings, one mainly needs an investigation strategy that is reliable and above all, efficient. Hadamard matrix designs fit these requirements. We have shown how to estimate main and interaction effects of factors when using experimental designs based on Hadamard matrices. We have also explained how to compute the standard error and to interpret the results to discard less important factors, thus focusing future experiments on those factors shown to be important. Note that the theory of Hadamard matrices [1,10,13,14,15,16] has not been covered herein. More conventional designs are found readily in numerous textbooks [7,8,9,11,12].

6. References

- [1] N.Drouin, "Building Hadamard Matrices in Steps of 4 to Order 200," NISTR 5121, April 1993.
- [2] R.Snelick, J.Ja'Ja, R.Kacker, G.Lyon, "Using Synthetic-Perturbation Techniques for Tuning Shared Memory Programs," NISTR 5139, March 1993.
- [3] G.Lyon, R.Snelick, R.Kacker, "Synthetic-Perturbation Tuning of MIMD Programs," NISTR 5131, February 1993.
- [4] G.Lyon, R.Snelick, R.Kacker, "Time-Perturbation Tuning of MIMD Programs," Performance Tools '92, R.Pooly and J.Hillston (eds.) Edinburgh University Press Ltd. Proceedings of the Sixth International Conference on Modeling Techniques and Tools for Computer Performance Evaluation , 16-18 September, 1992. Also as NISTR 4859, June 1992.
- [5] R.Candlin, J.Phillips, "Studying the Performance Properties of Concurrent Programs by Simulation Experiments on Synthetic Programs," Department of Chemical Engineering, University of Edinburgh, May 28 1992. (For the source text of this document contact rc#cs.ed.ac.uk).
- [6] G.Lyon, R.Snelick, R.Kacker, N.Drouin and J.Ja'Ja, "Portable Tuning Method for MIMD Programs," NFS Workshop, presentation 1 April 1992, Keystone CO.
- [7] R.Jain, **The Art of Computer Systems Performance Analysis**, John Wiley & Sons, Inc, 1991.
- [8] R.N.Kacker, E.S.Lagergren, J.J.Filliben, "Taguchi's Orthogonal Arrays Are Classical Designs of Experiments," Journal of Research of the National Institute of Standards and Technology, Vol.96, No.5, Sept-Oct 1991.
- [9] T.B.Barker, "Quality by Experimental Design," Marcel DEKKER Inc, 1985.
- [10] A.Hedayat, D.W.Wallis, "Hadamard matrices and their application," University of Illinois at Chicago Circle and University of Newcastle, The Annals of Statistics 1978, Vol.6, No.6, 1184-1238.
- [11] G.E.P Box, W.G.Hunter, J.S.Hunter, "Statistics for Experimenters," John Wiley & Sons, New York, 1978.
- [12] G.J.Hahn, "Some Things Engineers Should Know About Experimental Design," Journal of Quality Technology, Vol.9, No.1, January 1977.
- [13] L.D.Baumert, "Hadamard Matrices of order 116 and 232," Bull. Amer. Soc. (1966) 71 169-170
- [14] L.D.Baumert, M.Hall, Jr., "A new construction for Hadamard matrices," Bull. Amer. Math. Soc. (1965a) 71 169-171.
- [15] L.D.Baumert, M.Hall, Jr., "Hadamard Matrices of the Williamson Type," Math. Comp. (1965b) 19 442-447.
- [16] L.D.Baumert, S.W.Golomb, M.Hall, Jr., "Discovery of an Hadamard Matrix of Order 92," Bull. Amer. Math. Soc. (1962) 68 237-238.

