

corresponds exactly to the standard 6-311G(d,p) basis set.

For smaller basis-set correlated calculations, it may often be advantageous to use the double- $\zeta$ -valence plus polarization 6-31G<sup>+</sup> and 6-31G<sup>++</sup> basis sets in place of 6-31G(d) and 6-31G(d,p), particularly for processes that are likely to be accompanied by a significant change in the d-function exponent. The standard exponents for 6-31G<sup>++</sup> (and 6-31G<sup>+</sup>) are the same as those for 6-311G(MC)(d,p).

Heats of formation for second-row hydrides, calculated by using the procedure of Pople et al. at the MP4 level with an augmented

6-311G(MC)(d,p) basis set, are generally in good agreement with available experimental values and, in many cases, would represent the best estimates of the heats of formation currently available.

**Registry No.** H, 12385-13-6; H<sub>2</sub>, 1333-74-0; Na, 7440-23-5; Al, 7429-90-5; Si, 7440-21-3; P, 7723-14-0; S, 7704-34-9; Cl<sub>2</sub>, 7782-50-5; NaH, 7646-69-7; MgH, 14332-53-7; MgH<sub>2</sub>, 7693-27-8; AlH, 13967-22-1; AlH<sub>2</sub>, 14457-65-9; AlH<sub>3</sub>, 7784-21-6; SiH, 13774-94-2; SiH<sub>2</sub>, 13825-90-6; SiH<sub>3</sub>, 13765-44-1; SiH<sub>4</sub>, 7803-62-5; PH, 13967-14-1; PH<sub>2</sub>, 13765-43-0; PH<sub>3</sub>, 7803-51-2; SH, 13940-21-1; SH<sub>2</sub>, 7783-06-4; ClH, 7647-01-0.

## Optimized Two-Electron-Integral Transformation Procedures for Vector-Concurrent Computer Architectures

Jonathan N. Hurley, David L. Huestis,<sup>†</sup> and William A. Goddard, III\*

Arthur Amos Noyes Laboratory of Chemical Physics,<sup>‡</sup> California Institute of Technology, Pasadena, California 91125 (Received: December 10, 1987)

Various approaches for the transformation of two-electron integrals are discussed. Using the full index symmetry (FIS) of the array to be transformed for a case with  $N$  basis functions, the  $N$ -to- $N$  transformation yields an operation count of  $(25/24)N^5$ , while transformation to a small set with  $M$  orbitals yields  $(1/8)N^4M$  if  $N \gg M$ . We discuss the vector-optimized implementation of this (FIS) procedure for concurrent architectures. Tests were carried out on an Alliant FX8/8 mini-supercomputer (eight processors sharing one memory) and compared with the performance of DEC VAX scalar machines.

### I. Introduction

New computer architectures allowing multiple processors to work in parallel on a given problem are being developed. Examples include hypercubes with distributed memory (e.g., Intel, Caltech, FPS) and with shared memory (the Alliant FX8). The Alliant is especially promising for studying electronic structure problems because it has an effective, vector-concurrent optimizing Fortran compiler allowing straightforward coding, since data passing and synchronization between processors are taken care of automatically by the operating system. Furthermore, the large global memory available (currently up to 256 Mbytes) allows large systems ( $\sim 200$  basis functions) to be tackled. This, combined with use of effective core potentials, provides the capability for studying large clusters of transition metals (say, Pt<sub>10</sub>).

In this paper we report progress in developing an optimized integral transformation program (FISTRN) for parallel machines. This involved restructuring the algorithms to take maximum advantage of concurrency, vectorization, and huge memory. In addition, the program must be designed specifically to take advantage of automatic vector-concurrent optimizing Fortran compilers.

### II. Quantum Mechanics

For the calculation of correlated wave functions, it is essential to have an efficient procedure for transforming integrals over atomic or symmetry orbitals (AO's) into integrals over molecular orbitals (MO's). The AO two-electron integrals

$$A_{ij,kl} \equiv \left\langle \psi_i(1) \psi_k(2) \left| \frac{1}{r_{12}} \right| \psi_j(1) \psi_l(2) \right\rangle \\ \equiv \int d^3x_1 \psi_i(1) \psi_j(1) \int d^3x_2 \frac{1}{r_{12}} \psi_k(2) \psi_l(2) \quad (1)$$

are transformed into the corresponding two-electron integrals over a new set of basis functions

$$Z_{\alpha\beta,\gamma\delta} \equiv \left\langle \phi_\alpha(1) \phi_\gamma(2) \left| \frac{1}{r_{12}} \right| \phi_\beta(1) \phi_\delta(2) \right\rangle \quad (2)$$

where

$$\phi_\alpha = \sum_{n=1}^N \psi_n C_{n\alpha} \quad (3)$$

Taking the number of basis functions in each set as  $N$ , then, depending on how the  $Z$  integrals are calculated in terms of the  $A$  integrals, the number of multiplications (or the operation count, OpC) may increase as rapidly as  $qN^8$  or as slowly as  $q'N^5$ . In addition, the prefactors  $q$  and  $q'$  may vary by a factor of 4. Herein we will consider in detail the techniques for such transformations.

In particular, the array of integrals  $A$  is symmetric in the indexes  $i$  and  $j$ , in the indexes  $k$  and  $l$ , and in the pairs  $ij$  and  $kl$ . In order to minimize the prefactor  $q$ , this index symmetry is exploited during the transformation.

### III. Review of Two-Electron-Integral Transformation Theory<sup>1</sup>

The two-electron integrals of eq 1 and 2 are related by

$$Z_{\alpha\beta,\gamma\delta} = \sum_{ijkl} C_{i\alpha} C_{j\beta} A_{ijkl} C_{k\gamma} C_{l\delta} \quad (4)$$

where  $C$  is the  $N \times M$  transformation matrix connecting the  $N$  basis functions  $\{\psi_i\}$  and the  $M$  molecular functions  $\{\phi_\alpha\}$ .

*A. Direct Multiplication* ( $(1/16)N^8$ ). An obvious but inefficient method of accomplishing the transformation (4) would be to calculate the quantity  $C_{i\alpha} C_{j\beta} A_{ijkl} C_{k\gamma} C_{l\delta}$  for each  $i, j, k, l, \alpha, \beta, \gamma, \delta$ . Each of these terms would then be accumulated in the appropriate  $Z_{\alpha\beta,\gamma\delta}$ . This technique requires the largest number of operations,  $\sim (1/16)N^4M^4$ , even if the full index symmetries of both  $A$  and  $Z$  are used.

<sup>†</sup> Current address: Chemical Physics Laboratory, SRI-International, Menlo Park, CA 94025.

<sup>‡</sup> Contribution No. 7710.

(1) Huestis, D. L.; Goddard III, W. A., unpublished. See: Huestis, D. L. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, 1972.

TABLE I: Number of Operations for the Four Steps Involved in Transforming the Two-Electron Integrals with FIS<sup>5</sup>

step	summation	approximate OpC
1	$X_{ijkl} = \sum_{\substack{l=k \\ ij \leq kl}}^N A_{ijkl} C_{l\beta}$	$(1/8)N^4M$
2	$F_{ij\gamma\delta} + F_{ij\beta\gamma} = \sum_{k=i}^N X_{ijk\delta} C_{k\gamma} + X_{ijk\gamma} C_{k\delta}$	$(1/6)N^3M^2$
3	$Y_{i\beta\gamma\delta} = \sum_{j=1}^i [F_{ij\gamma\delta} + F_{ij\beta\gamma}] C_{j\beta}$	$(1/4)N^2M^3$
4	$Z_{\alpha\beta\gamma\delta} = \sum_{i=1}^N [Y_{i\beta\gamma\delta} C_{i\alpha} + Y_{i\alpha\gamma\delta} C_{i\beta} + Y_{i\beta\alpha\delta} C_{i\gamma} + Y_{i\beta\gamma\alpha} C_{i\delta}]$	$(1/2)NM^4$

B. Density Matrices ((3/16)N<sup>6</sup>). One method of improving upon direct multiplication is to use density matrices, which are defined by

$$D_{ij,\alpha\beta} = [C_{i\alpha}C_{j\beta} + C_{i\beta}C_{j\alpha}] \quad i \neq j$$

$$= C_{i\alpha}C_{j\beta} \quad i = j \quad (5)$$

for all  $i \geq j$ ,  $\alpha \geq \beta$ . Then eq 4 becomes

$$Z_{\alpha\beta\gamma\delta} = \sum_{i \geq j} \sum_{k \geq l} D_{ij,\alpha\beta} A_{ijkl} D_{kl,\gamma\delta} \quad (6)$$

Considering each pair  $ij$  as a single subscript,  $ij = \frac{1}{2}i(i-1) + j$ , and similarly for  $\alpha\beta$ , and setting  $ms = \frac{1}{2}M(M+1)$ ,  $ns = \frac{1}{2}N(N+1)$ , we obtain

$$Z_{\alpha\beta,\gamma\delta} = \sum_{ij=1}^{ns} \sum_{kl=1}^{ns} D_{ij,\alpha\beta} A_{ijkl} D_{kl,\gamma\delta} \quad (7)$$

This similarity transformation of a symmetric matrix requires at least  $\frac{1}{2}ns(ms+1)ms + ms(ns)^2$  multiplications. Expanding in terms of  $N$  and  $M$

$$\text{OpC} = \frac{1}{16}N(N+1)(N^2+N+2)M(M+1) + \frac{1}{8}N(N+1)M(M+1)(M(M+1)+2) \quad (8)$$

or approximately  $(1/16)N^4M^2 + (1/8)N^2M^4$ . This becomes  $(3/16)N^6$  if  $N = M$  and  $(1/16)N^4M^2$  if  $N \gg M$  (ref 2).

C. Successive Transformations ( $4N^5$ ). An alternative approach involves saving the intermediate result after each summation

$$Z_{\alpha\beta\gamma\delta} = \sum_i C_{i\alpha} \sum_j C_{j\beta} \sum_k C_{k\gamma} \sum_l A_{ijkl} C_{l\delta} \quad (9)$$

When no index symmetry is used, this takes  $N^4 + N^3M^2 + N^2M^3 + NM^4$  multiplications, which leads to dominant terms of  $4N^5$  for  $N = M \gg 1$  and  $N^4M$  for  $N \gg M$  (ref 3).

D. Modified Full Index Symmetry<sup>1,5</sup> ( $(25/24)N^5$ ). Using the full index symmetry (FIS) of each intermediate result, i.e., never calculating anything more than once, can reduce the OpC for the process in (9) to  $(1/8)N^4M + (1/6)N^3M^2 + (1/4)N^2M^3 + (1/2)NM^4$ , which reduces to  $(25/24)N^5$  and  $(1/8)N^4M$  for  $N = M \gg 1$  and  $N \gg M$ , respectively. The discussion of this FIS procedure is as follows.

The given problem (4) can be written in symbolic tensor notation:

$$\hat{Z} = C \otimes C \cdot \hat{A} \cdot C \otimes C \quad (10)$$

Let  $\hat{Q}$  be any fourth rank tensor; then the possible index permutations (analogous to matrix transposition) are defined by

$$\hat{Q}_{ijkl}^{\tau} = \hat{Q}_{\tau(ijkl)} \quad (11)$$

TABLE II: Number of Operations for the Four Steps Involved in Transforming the Two-Electron Integrals with FIS Utilizing Only  $A_{ijkl}$   $ij \geq kl$ 

step	summation	approximate OpC
1	$X_{ijk\delta} = \sum_{\substack{l=1 \\ ij \geq kl}}^k A_{ijkl} C_{l\delta}$	$(1/8)N^4M$
2	$F_{ij\gamma\delta} + F_{ij\beta\gamma} = \sum_{k=1}^i X_{ijk\delta} C_{k\gamma} + X_{ijk\gamma} C_{k\delta}$	$(1/3)N^3M^2$
3	$Y_{i\beta\gamma\delta} = \sum_{j=1}^i [F_{ij\gamma\delta} + F_{ij\beta\gamma}] C_{j\beta}$	$(1/4)N^2M^3$
4	$Z_{\alpha\beta\gamma\delta} = \sum_{i=1}^N [Y_{i\beta\gamma\delta} C_{i\alpha} + Y_{i\alpha\gamma\delta} C_{i\beta} + Y_{i\beta\alpha\delta} C_{i\gamma} + Y_{i\beta\gamma\alpha} C_{i\delta}]$	$(1/2)NM^4$

TABLE III: Comparison of the Total Number of Operations Involved in the Two-Electron-Integral Transformation for the Different Methods Discussed Above

method	$N = M$	$M \ll N$
direct multiplication	$(1/16)N^8$	$(1/16)N^4M^4$
density matrices	$(3/16)N^6$	$(1/16)N^4M^2$
successive transformations	$4N^5$	$N^4M$
FIS	$(11/8)N^5$	$(1/2)N^4M$
modified FIS	$(25/24)N^5$	$(1/8)N^4M$

where  $\tau$  is any permutation from  $S_4$  (the symmetric group of dimension 4). For example,

$$\hat{Q}_{ijkl}^{(12)} = \hat{Q}_{jikl} \quad (12)$$

The symmetries of the array of two-electron integrals can be written as

$$\hat{A}^{\tau} = \hat{A} \quad \text{for } \tau \in \Gamma \quad (13)$$

where  $\Gamma$  is the subgroup of  $S_4$  generated by  $\{(12), (34), (13)(24)\}$ . We decompose  $\hat{A}$  as

$$\hat{A} = \sum_{\tau \in \Gamma} \hat{T}^{\tau} \quad (14)$$

uniquely by defining  $T$  such that  $T_{ijkl} = 0$  if the indexes  $i, j, k, l$  are out of their standard canonical order.<sup>3</sup> Since

$$[C \otimes C \cdot \hat{T} \cdot C \otimes C]^{\tau} = C \otimes C \cdot \hat{T}^{\tau} \cdot C \otimes C \quad (15)$$

we define

$$\hat{B} = C \otimes C \cdot \hat{T} \cdot C \otimes C \quad (16)$$

so that

$$\hat{Z} = \sum_{\tau} \hat{B}^{\tau} \quad (17)$$

The four successive summations are illustrated in Table I. All of the final index symmetry is introduced by the sum over permutations at the end (17). This means, for instance, that  $F_{ij\gamma\delta} \neq F_{ij\beta\gamma}$  for  $\gamma \neq \delta$ . Earlier introduction of some of the final index symmetry reduces the number of operations in steps 3 and 4 and this is how the calculation is actually performed (see Table I). We call this procedure modified FIS. The final result is

$$\hat{z} = \hat{V} + \hat{V}^{(12)} + \hat{V}^{(13)(24)} + \hat{V}^{(12)(13)(24)} \quad (18)$$

leading to a grand total

$$\text{OpC} = \frac{1}{8}N^4M + \frac{1}{6}N^3M^2 + \frac{1}{4}N^2M^3 + \frac{1}{2}NM^4 \quad (19)$$

(ignoring terms of fourth order or less, i.e.,  $N, M \gg 1$ ). This reduces to  $(25/24)N^5 = 1.04N^5$  for  $N = M \gg 1$  and  $(1/8)N^4M$  for  $N \gg M$ . This is the method of Saunders et al.<sup>5</sup> where the first transformation runs over  $l = k, N$  and the second has  $k = i, N$ . This results in a saving of  $(1/6)N^3M^2$  in the second step

(2) Nesbet, R. K. *Rev. Mod. Phys.* **1963**, *35*, 552.

(3) McLean, A. *Proceedings of the Conference on Potential Energy Surfaces in Chemistry, University of California, Santa Cruz, August, 1970*; p 87. Published as IBM Report RA 18 (1971).

(4) Whiteside, R. A.; Binkley, J. S.; Colvin, M. E.; Schaefer III, H. F. *J. Chem. Phys.* **1987**, *86*, 2185.

(5) Saunders, V. R.; Van Lenthe, J. H. *Mol. Phys.* **1983**, *48*, 923.

**TABLE IV: Storage Requirements in kilobytes for the Transformation of Two-Electron Integrals with Full Index Symmetry<sup>a</sup>**

step	array	storage	$N = 1000$	$N = 120$
			$M = 64$	$M = 120$
1	$\hat{T}$	$(1/2)N(N+1)$	3910	56.72
	$\hat{C}$	$NM$	500.0	112.5
	$\hat{X}$	$NM$	500.0	112.5
2	$\hat{F}$	(use $\hat{T}$ )	0	0
3	$\hat{Y}$	$(1/2)M^2(M+1)$	1040	6806
4	$\hat{Z}$	$(1/2)ms(ms+1)$	8454	102959
		$[ms = (1/2)M(M+1)]$		
total			14404	110046

<sup>a</sup>The output integrals are Real\*4.

because as the number of executions of the loop increases ( $j = 1, i$  with  $i = 1, N$ ) the length of the loop decreases ( $k = i, N$ ). The only disadvantage of this method over the original (Table II) is that it requires AO integrals  $A_{ijkl}$ ,  $kl \geq ij$ , whereas our SCF code requires  $A_{ijkl}$ ,  $kl \leq ij$ . Since our integral programs put out all  $kl$  integrals, however, this is not a problem. We believe that this version of modified FIS is the most efficient integral transformation algorithm available (Table III).

Bender<sup>6</sup> has also proposed general two-electron-integral transformations using index symmetry. This paper discusses a method using index symmetry but does not estimate the operation count. It appears that the method is the same as Huestis,<sup>1</sup> yielding an operation count of  $(29/24)N^5$ . However, Bender does give formulas that allow one to take into account the spatial symmetry of the molecule to reduce the operation count. For systems of high symmetry, particularly atoms, this will result in enormous savings. We have not implemented code for taking advantage of symmetry owing to the difficulty of vectorizing it efficiently.

#### IV. Programming Considerations

It was desired to run the FISTRN program concurrently on all eight processors so as to minimize the total elapsed time. To obtain concurrent operation on an Alliant computer there are two procedures that can be followed: (1) *vector-concurrent*, where a single do-loop is split across all eight vector processors; or (2) *concurrent outer vector inner (COVI)*, where each iteration of the second inner loop is performed concurrently on a different processor.

COVI is the preferred procedure for this code since the inner loop is of length NBF (number of basis functions), where NBF is typically  $< \sim 100$  (the vector length of each vector processor is 32); thus we need vectors lengths  $> 256$  for efficient vector-concurrent operation.

The index symmetry is brought in at the second step in order to halve the OpC for steps 3 and 4 and so that we get the largest possible inner loop lengths. Table I gives the transformations that are performed.

The format of the program is (1) the transformation matrix C from AO's to MO's is read in, (2) one-electron integrals are read and transformed, and (3) the two-electron integrals  $A_{ijkl}$  are read in one  $ij$  block at a time.

Because one block of  $ij$  integrals transforms into all  $\alpha\beta$  blocks, to employ the algorithm described in section III, it is most efficient to store all  $Z_{\alpha\beta\gamma\delta}$  in core. Since full index symmetry is utilized, we need only store  $(1/8)N(N+1)[N(N+1)+2]$  such integrals. Table IV gives some data for cases with various numbers of basis functions. Using Real\*4 variables for the output integrals, a 110 Mbyte system can accommodate  $M = 120$  output basis functions. However, the memory requirement is largely independent of the number of input functions (cf.  $N = 1000$  and  $M = 64$ ). Hence, if one is doing a contraction of the basis set (for instance, for a restricted CI), the program can deal with an almost unlimited size input basis set. It was decided to employ this strategy of storing all output integrals in core, since intermediate storage on disk requires wasteful and complex sorting routines and because projected increases in the size of solid-state memory mean that

**TABLE V: CPU Times (in seconds) for Integral Transformations with Various Numbers of Basis Functions<sup>a</sup>**

	SiH <sub>3</sub> 32 <sup>c</sup>	C <sub>2</sub> F <sub>4</sub> 54 <sup>c</sup>	C <sub>2</sub> F <sub>6</sub> 72 <sup>c</sup>	C <sub>3</sub> F <sub>8</sub> 99 <sup>c</sup>	C <sub>4</sub> F <sub>10</sub> 122 <sup>c</sup>	av ratio
DEC 8650	56	728	6430	26088	64878	0.563
FX8-1 ce	48	544	2170	11080	29113	1.000
FX8-2 ce's	24	277	1132	5795	15026	1.946
FX8-4 ce's	13	154	625	3150	8312	3.521
FX8-8 ce's	8.6	99	415	2001	5318	5.462
Intel Hypercube <sup>b</sup>	320					0.150
DEC $\mu$ VAX II	330					0.145

<sup>a</sup>FX indicates the Alliant FX-8. <sup>b</sup>Intel Hypercube as programmed by Whiteside et al.<sup>4</sup> Largest case that could be run was 32 basis functions. <sup>c</sup>Number of basis functions.

in-core programs will have the ability to run most systems of future interest.

An example of the transformation code is shown below.

```

c perform 1st index transformation (AAAA → AAAM)
  x = 0.0d0
  kl = i * (i-1)/2
  do k = i, nbf
    iofmo = (k-1) * norb
    lmin = 1
    if (k.eq.i) lmin = j
    do is = 1, norb
      ioftr = (is-1) * nbf
      do l = lmin, k
        x(iofmo+is) = x(iofmo+is) + vj(kl+l) * trans(ioftr+l)
      &
      end do
    end do
    kl = kl + k
  end do

```

This loop (and the other three loops that perform the remaining three index transformations) is performed once for each block of  $ij$  input integrals. The inner loop has an average length of about  $NBF/2$ , which, for systems with greater than 64 basis functions, is greater than the vector length of the vector processor. The second inner loop is of length  $norb$  (the number of output basis functions) which again is typically much larger than the number of processors (8). Thus our code is well suited for a vector-concurrent architecture. In contrast, a straightforward vector processor would not perform as well since, in general, they have longer vector lengths (up to 128) and greater vector startup times, all of which factor much longer vectors.

In carrying out operations on canonical lists of integrals and other symmetric matrices, attention must be paid to handling of diagonal elements that would be counted twice (for example, loops over  $i = 1$  to  $NBF$  and  $j = 1$  to  $i$ ). Left unaltered, this could lead to a number of conditional statements to cope with the cases of  $i = j$ ,  $k = l$ , and  $ij = kl$ . Instead, we follow the common practice of multiplying (before the transformation) each integral by a factor of  $1/2$  for each equality satisfying  $i = j$ ,  $k = l$ , or  $ij = kl$  (so, for instance, 3333 is multiplied by  $1/8$ ). This then eliminates all conditionals except the one shown above (for the case  $k = i$ , the limit on  $l$  is the minimum of  $j$  and  $k$ ).

#### V. Results

Table V gives timing data for  $N$ -to- $N$  transformation of various sizes run on our reference VAX 8650, different numbers of processors on the Alliant, and one case run on an Intel Hypercube quoted by Whiteside et al.<sup>4</sup> In general, with a multiprocessor Alliant, the amount of time used by each processor will be different because of scalar parts to the code. The CPU times quoted are for the processor with the maximum CPU time. These timings show appreciable concurrency in the code (four processors  $\sim 3.5$  times faster than one), but the raw speed of one processor over the 8650 of about 1.8 (LINPACK benchmarks on a single processor Alliant lead to speeds about 3 times the 8650) is rather disappointing. We also note that the speedup drops off quite significantly with more than four processors. This is found in general for codes that perform vector operations rather than matrix operations and is due to memory contention (the bandwidth is

(6) Bender, C. F. *J. Comput. Phys.* **1971**, *9*, 547.

not large enough to handle all the I/O since for every term of the form  $c = a + b$  there are two reads and a write). The data for 32 basis functions also indicate that with present technology, an Alliant eight-processor system is 35 times faster than a 32-processor Intel Hypercube. Further reason for using the Alliant is the ease of programming since all interprocessor communication and synchronization problems are automatically dealt with by the compiler and the Alliant utilizes an optimizing Fortran compiler.

## VI. Conclusions

We have demonstrated that with fairly straightforward reprogramming the FISTRN integral transformation routine performs well on a vector-concurrent architecture. We believe that it is necessary to redevelop quantum chemistry programs to be optimal on parallel computers in order for quantum chemistry to solve

the important chemical problems of the next decades. One strategy involves making use of large memory since the most efficient algorithms rely on random access to all the integrals. Because of the continuing decline in cost of solid-state memory, this will become a viable approach to solve I/O bound problems. Of the two types of architecture, global memory (e.g., the Alliant) and local memory (e.g., the Intel Hypercube), we believe that global memory machines are more effective for quantum chemistry methods since current hardware and software technology already allow effective concurrency with only straightforward restructuring of code.

*Acknowledgment.* This work was supported by a contract (No. N00014-86-K-0735) from the Office of Naval Research and Defense Advanced Research Projects Agency.

## Correlation between $pK_a$ Values and Nuclear Magnetic Resonance Dilution Shifts for the Hydroxylic Proton of Acids in Pyridine

Tsuneo Ikenoue<sup>†</sup>

*Chemical Research Institute of Non-Aqueous Solutions, Tohoku University, Katahira 2-1-1, Sendai, 980, Japan*

and Nobuyuki Yoshida\*

*Department of Chemistry, International Christian University, Mitaka, Tokyo, 181, Japan  
(Received: December 30, 1987)*

The NMR dilution shifts for the hydroxylic proton of carboxylic and phenolic acids as well as proton shifts for HI, HBr, and HCl in pyridine were studied at low acid concentrations by using thoroughly dried pyridine. The chemical shift values ( $\delta^0$ ) for the hydroxylic or hydrohalic proton of these acids at the limit of zero acid concentration were determined, and the correlation between the  $\delta^0$  and  $pK_a$  values was examined. Correlations were found not only for monobasic acid systems but also for average  $pK_a$  values in mixed acid and dibasic acid systems.

## Introduction

A number of reports have been published on NMR studies of chemical shift measurements for acid-base complex formation in solution. Linear relationships between the "complexation shift" and the enthalpy of the hydrogen-bonding interaction have been obtained by Yablonskii<sup>1</sup> for weak hydrogen-bonding interactions, by Eyman and Drago<sup>2</sup> for systems of moderate strength interactions with  $\Delta H$  in the 3-10 kcal/mol range, and by Hadzi and Smerkolj<sup>3</sup> for strongly interacting systems.

The correlations between complexation shift and  $\Delta pK_a$  have also been investigated for interactions between halogenated acetic acids and substituted pyridines,<sup>4-6</sup> pyridine *N*-oxides,<sup>7,8</sup> quinoline *N*-oxides,<sup>9-11</sup> substituted arsine oxide, or phosphine oxides<sup>12</sup> in various solvents. These investigations were based on the NMR measurement of characteristic chemical shifts for hydroxylic protons in hydrogen-bonded acid-base complexes dissolved in various inert solvents and the measurement of the characteristic NMR dilution shifts for the hydroxylic protons of the free (non-hydrogen-bonded) acid in the corresponding acid-solvent binary systems. The complexation shift has been defined as the difference between the chemical shift of the hydroxylic proton in the hydrogen-bonded complex and in the free acid.

We have reported previously<sup>13</sup> that in rigorously dried pyridine, the NMR dilution shift curve for the hydroxylic proton in the acetic acid-pyridine binary system could be readily extrapolated

to the limit of infinitely dilute acid, and that  $\delta^0$  (the chemical shift for the hydroxylic proton at the extrapolated limit of zero acid concentration) could be interpreted to be the characteristic chemical shift resulting from the 1:1 acid-pyridine interaction. This paper describes the determination of the  $\delta^0$  values for acids and phenols and examines the correlation between the  $\delta^0$  values and the  $pK_a$  values of the acids and phenols. Since the characteristic chemical shift for the hydroxylic proton in free acids and phenols in pyridine cannot be determined, it is impossible to derive

- (1) Yablonskii, O. P. *Zh. Fiz. Khim.* **1975**, *49*, 2538.
- (2) Eyman, D. P.; Drago, R. S. *J. Am. Chem. Soc.* **1966**, *88*, 1617.
- (3) Hadzi, D.; Smerkolj, R. *J. Chem. Soc., Faraday Trans. 1* **1975**, *72*, 1188.
- (4) Dega-Szafran, Z.; Dulewicz, E. *Org. Magn. Reson.* **1981**, *16*, 214.
- (5) Dega-Szafran, Z.; Kunzendorf, J. *Pol. J. Chem.* **1979**, *53*, 623.
- (6) Dega-Szafran, Z.; Grech, E.; Naskret-Barciszewska, M. Z.; Szafran, M. *J. Chem. Soc., Perkin Trans. 2* **1975**, 250.
- (7) Dega-Szafran, Z.; Naskret-Barciszewska, M. Z.; Szafran, M. *J. Chem. Soc., Perkin Trans. 2* **1974**, 763.
- (8) Brycki, B.; Dega-Szafran, Z.; Szafran, M. *Adv. Mol. Relax. Interact. Processes* **1979**, *15*, 71.
- (9) Dega-Szafran, Z.; Grech, E.; Szafran, M. *J. Chem. Soc., Perkin Trans. 2* **1972**, 1839.
- (10) Dega-Szafran, Z.; Grech, E.; Naskret-Barciszewska, M. Z.; Szafran, M. *Adv. Mol. Relax. Interact. Processes* **1973**, *5*, 89.
- (11) Brycki, B.; Dega-Szafran, Z.; Szafran, M. *Pol. J. Chem.* **1980**, *54*, 221.
- (12) Grundwald, M.; Szafran, M.; Rychlewski, J. *Pol. J. Chem.* **1979**, *53*, 829.
- (13) Ikenoue, T.; Yoshida, N. *J. Phys. Chem.* **1987**, *91*, 6440.

<sup>†</sup> Present address: TRD Co. Ltd., Imozawa, Miyagimachi, Miyagi 989-32, Japan.