Using Parallel Supercomputers To Calculate Surface Energy Distributions

Brett J. Stanley,*,†,‡ Christian Halloy,§ and Georges Guiochon‡

Department of Chemistry, University of Tennessee, Knoxville, Knoxville, Tennessee 37996-1503, Chemical and Analytical Sciences Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-6120, and Joint Institute for Computational Science, University Tennessee, Knoxville, Knoxville, Tennessee 37996-1508

Received July 8, 1994⊗

The mathematically ill-posed problem of solving linear Fredholm integrals of the first kind for distribution functions, given experimental data, is performed with an iterative maximum-likelihood method called expectation-maximization (EM). The algorithm is programmed on two supercomputers of different architecture: the 4096 processor MP-2 from MasPar Corporation and the 32 processor CM-5 from Thinking Machines Corporation. Parallelization and use of the matrix-vector routines supplied by the vendors provides substantially faster run-times than that executed with the sequential code by other mainframe computers. This increase in computation efficiency provides a more practical use of the EM algorithm for these types of problems, which has been shown to be an optimal method. The MP-2 outperforms the more powerful CM-5 until the dimensions of the problem become fairly large.

INTRODUCTION

The linear Fredholm integral equation of the first kind

$$q(p) = \int_{\Omega} f(\epsilon)\theta(p,\epsilon)d\epsilon \qquad (1)$$

has been utilized extensively for the description of physical problems exhibiting heterogeneity or otherwise distributed data. Equation 1 is cast in terms of adsorption isotherms and adsorption energy distributions,[1,2] where $q(p)$ is the experimental, "global" isotherm, measured as a function of the partial pressure, p, of the adsorbing species; $\theta(p,\epsilon)$ is the specified "local" isotherm which serves as the model and kernel of the transform; and $f(\epsilon)$ is the desired distribution function of the adsorption energy, ϵ, which describes the heterogeneity of the surface energy. Ω is the range of energies specified by the range of pressures measured via an acceptable transform from pressure to energy, e.g., the condensation approximation.[1] This equation has also been used to estimate the heterogeneity of first-order kinetic rate constants describing desorption processes of metal solutes from chelating organic matter in soils, which may describe the binding and structural heterogeneity of these materials.[3] Other applications include image analysis in microscopy and astronomy.[4,5]

Optimal solution of eq 1 for the function $f(\epsilon)$ has been and is a matter of long debate; however, it is not the purpose of this paper to contribute to this quandary. Suffice it to say that several "optimal" methods have been proposed[6] which may simply be divided into two types of algorithms: iterative and noniterative algorithms. Noniterative techniques have the advantage of providing solutions directly with considerable savings in computation time. Of these methods, regularized regression appears to be an attractive choice.[6−8] The time savings are possible because of the linear algebra operations typically involved and the documented algorithms available for their efficient solution, e.g., LINPACK and EISPACK routines for various matrix decompositions and inversions.[9] However, the appropriate decomposition of experimental and numerical errors remains a challenging problem. For example, the programmed truncation of the significant set of singular values from a singular value decomposition may result in the inclusion of error information or the loss of true model information. These problems result in undesirable, artifactual peaks, or too featureless a solution, respectively, without a fool-proof method of control. The iterative expectation-maximization (EM) method[10] has been shown to converge toward the maximum-likelihood estimate with stability and the suppression of artifactual peaks.[11,12] As the solution approaches the global optimum, changes in the solution become smaller with each iteration; thus large changes which may correspond to artifacts cannot occur within a finite number of iterations. This inherent form of smoothing results in a small increase in estimation variance and a decrease in resolution. The initial guess is taken as a uniform distribution in this work, since no information exists on the correct distribution. However, if such information did exist, it could be incorporated into the initial guess to promote a more accurate solution. Since a stable and reliable convergence is obtained after several thousand iterations, this method should be superior if computation time is not of concern. Therefore, the increase in computation efficiency of the EM algorithm is the focus of the current paper.

ALGORITHM

The EM algorithm has been published previously.[10] It consists of upgrading the kth iteration estimate of the solution at each point, x_j^k, by nonzero multiplication

$$x_j^{k+1} = x_j^k \sum_{i=1}^{n} \frac{A_{ij} y_i}{\sum_{j=1}^{m} A_{ij} x_j} \qquad (2)$$

In equation 2, $\mathbf{A}$ is an $m \times n$ matrix specifying the model or

† Present address: Department of Chemistry, California State University, San Bernardino, 5500 University Drive, Sann Bernardino, CA 92407-2397.
‡ Department of Chemistry, UT, Knoxville and ORNL.
§ Joint Institute for Computational Science, UT, Knoxville.
⊗ Abstract published in *Advance ACS Abstracts,* November 15, 1994.

0095-2338/95/1635-0110$09.00/0

kernel of eq 1. Each column represents the adsorption vs pressure behavior at adsorption energy ϵ_j; each row represents the adsorption vs energy at pressure p_i. Trapezoidal quadrature is programmed in **A** as well, i.e., $A_{ij} = (\theta(p_i,\epsilon_j) + \theta(p_i,\epsilon_{j+1}))/2 \times (\epsilon_{j+1} - \epsilon_j)$. **y** is the $n \times 1$ data vector, $q(p)$, and **x** is the $m \times 1$ solution vector, $f(\epsilon)$, where n is the number of data points and m is the number of grid points on the energy axis; k is the iteration number. This step was achieved in three program lines:

(1) formation of the $n \times 1$ vector $\mathbf{y}^* = \mathbf{y}/\mathbf{Ax}$;

(2) formation of the $m \times 1$ correction vector $\beta = \mathbf{y}^{*T}\mathbf{A}/\sum_i A_i$, where T denotes transpose;

and

(3) correction of **x** with element-by-element multiplication: $x_j^{k+1} = x_j^k \times \beta_j$ for $j = 1, m$. The estimate of the data, $\mathbf{y}_{est}$, must be calculated for each iteration ($\sum_j A_{ij} x_j^k$ for $i = 1, n$), before or after these steps. The root-mean-squared error of the estimate from the data was also calculated for each iteration, so as to assure convergence. If a divergent step occurs, something is wrong, and the program should be aborted, as the EM method is guaranteed to converge at every iteration. However, once this behavior is assured, the error may be calculated every 1000 iterations or so in order to minimize calculations. Calculation of the model matrix, **A**, and other preliminary calculations occur before the main iteration loop, and posterior calculations such as moment detection and analysis occur after the main loop. It is this main iteration loop with these vector-matrix operations that compose the overwhelming majority of the computation time.

PARALLELIZATION

Parallel computing has recently come into being with the recognition that single processor computers are approaching their fundamental limits in terms of clock speed, and yet the complexity and magnitude of computational applications has no such bounds.[13] Thus the next logical step in computing has been to team several to thousands of processing elements (PEs) together, assign specific computing tasks within a given algorithm to a single PE, and combine results or communicate between PEs efficiently at specific points in the algorithm. Two architectural approaches have been realized with currently available commercial computers. The conventional approach has been to connect a few to a few dozen powerful central processing units (CPUs) together. Each of these CPUs may possess discrete vector units. The Multiple Instruction Multiple Data (MIMD) approach to parallel programming is conveniently applicable to this type of machine, and these machines are sometimes called MIMD machines. The increasingly popular approach has been to connect thousands of smaller CPUs together with tighter control. This approach has been dubbed the "massively" parallel approach, and the Single Instruction Multiple Data (SIMD) approach to parallel computing is particularly well suited to this type of machine. Correspondingly, these machines are often referred to as SIMD machines.

The EM algorithm is not obviously or inherently parallel. Each iteration as well as each step within each iteration as outlined above is dependent on the previous iteration or step, respectively. Therefore, the program cannot be broken up into multiple segments for independent, parallel execution, and the MIMD approach is not applicable. However, a parallel architecture and/or vector representation can be used to drastically speed up each individual computation. This is accomplished in the *data-parallel* or SIMD mode by mapping the data, **y**, the model, **A**, the distribution, **x**, and all temporary vectors across the PE array. In this way each data point is assigned to its own individual PE or subvector unit. Moreover, the vectors and matrices are mapped such that communication between PEs is optimized, e.g., $\mathbf{y}_{est}$ and **y** are mapped along the same PEs in the same order so that the point operations with these two vectors occur on the same PE, thus minimizing communication time. The vendors realize this type of parallelism is highly beneficial when operating with matrices and typically supply the routines and operations needed to achieve this optimization.

EXPERIMENTAL SECTION

The above two computer architectures and the SIMD approach to parallelization of the EM algorithm were implemented with a 4096 PE MP-2 supercomputer from MasPar Computer Corporation (Sunnyvale, CA), and a 32 PE CM-5 supercomputer from Thinking Machines Corporation (Cambridge, MA). Both computers are located in the Computer Science Department at the University of Tennessee, Knoxville. For the MP-2, each PE is a RISC type processor with 64 Kbyte of memory and 40 32-bit registers. With each node performing 32-bit floating point arithmetic, a maximum peak performance of 1.5 Gflops can be attained. The source code may be compiled using 1*k*, 2*k*, or 4*k* of the PEs. This was performed to assess the scaling characteristics of the program. "Maximum" optimization was specified upon compilation for all cases. The CM-5 PEs each consist of a 32 MHz SPARC processor with 32 Mbyte of local memory and four vector units of length 8 each, for a total of 1024 vector elements. These vector units were utilized in the data mapping for the CM-5 timings. Each of the nodes is capable of performing 64-bit floating point arithmetic at a rate of 128 Mflops, yielding a maximum aggregate performance of 4 Gflops. The sequential run-times were obtained with two VAX computers: the VAX 6420 and the VAX 7640 as well as with a DEC 7640 "AXP" computer.

The data tested for this study is the adsorption isotherm of diethyl ether (99.9%, Aldrich) on a chromatographic silica (Impaq RG1010, PQ Chromatography Products), at 50 °C. This isotherm was obtained by the elution-by-characteristic-points method of gas chromatography,[14,15] over a partial pressure range of 7.8×10^{-8}–3.5×10^{-3} atm diethyl ether, with the use of a wall-coated capillary column. In order to study the effect of the number of data points on the computation times and estimation variance of the EM program, the data was fit to an Akima spline (a nonsmoothing, interpolating spline available from the Numerical Algorithm Group library) and interpolated with 100, 200, 500, 1000, 2000, and 5000 points. The original isotherm possessed 418 points. The number of grid points on the energy axis of the adsorption energy distribution was set to 200. The number of iterations was set to 20 000. The Langmuir local isotherm was specified for the model.[16]

RESULTS AND DISCUSSION

The experimental data and the calculated distribution function are given in Figure 1. The estimated data and the experimental data agree to within 1% (rms = 0.009 52 with raw data; $n = m = 418$), and the individual plots agree to

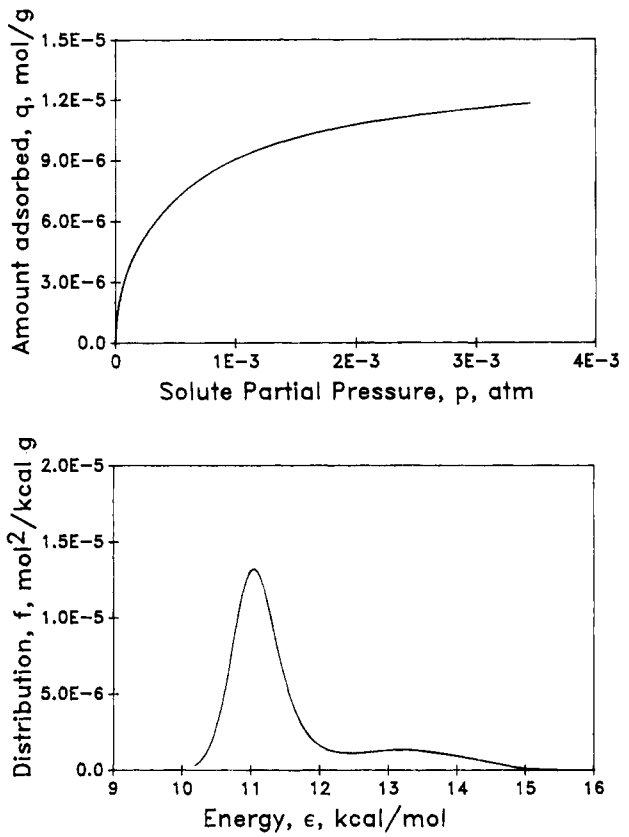

Figure 1. Experimental data and computational results of diethyl ether adsorption on Impaq RG1010 chromatographic silica: (a, top) adsorption isotherm taken at 50 °C and (b, bottom) adsorption energy distribution obtained with the EM method and the Langmuir local isotherm model.

Table 1. CPU Execution-Times of EM Algorithm for the Calculation of Adsorption Energy Distributions from Adsorption Isotherm Data as a Function of the Number of Data Points with Various Computers[a]

n (no. pts)	VAX 6420 (min)	VAX 7640 (min)	DEC 7640 (min)	MP-2 (min)[b]	CM-5 (min)[b,c]
100	39.7	6.60	3.55	1.31	15.4
200	80.2	13.4	7.22	1.78	8.33
500	205	35.3	17.9	2.71	10.4
1000	411	70.9	37.9	4.60	9.05
2000				8.25	9.23
5000				19.0	10.2

[a] Number of grid points = 200, number of iterations = 20 000. [b] Times optimized by specifying the exact number of data points, n, in the array dimension statements. [c] CM-5 timings are subject to variability due to time-sharing with other programs and users.

within the width of the lines shown. The sequential results and the parallel results agree to the sixth decimal place. The run-times for the EM code on the five machines tested are given in Table 1; the run-times on the MP-2 compiled with different numbers of PEs are given in Table 2. Graphical representation of these data are given in Figures 2–4.

As can be clearly seen from Figure 2, the parallel results on the MP-2 outperforms sequential calculation to varying degrees, depending on the CPU power. The enhanced efficiency is increased as the number of data points or the dimensions of the problem is increased. For the 100 × 200 problem (100 data points, 200 grid points), the MP-2 is only twice as fast as the DEC 7640. For the 1000 × 200 problem the MP-2 is a factor of 78 faster than the VAX 6420. For larger scale problems this increase is noteworthy, in that the

Table 2. CPU Execution Times of EM Algorithm for the Calculation of Adsorption Energy Distributions from Adsorption Isotherm Data for the MP-2 Supercomputer as a Function of the Number of PEs Utilized and the Number of Data Points[a]

n (no. pts)	MP-2, 1024 PEs (min)	MP-2, 2048 PEs (min)	MP-2, 4096 PEs (min)
100	3.10	2.69	1.75
200	4.08	3.40	2.24
500	7.09	5.06	3.21
1000	12.2	8.09	5.25

[a] Number of grid points = 200, number of iterations = 20 000.

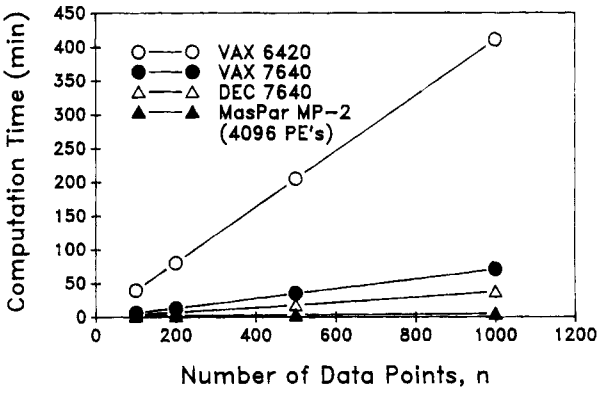

Figure 2. Computational efficiency of the EM algorithm with the computers used in this study: number of grid points = 200, number of iterations = 20 000; sequential computation with the VAX 6420, VAX 7640, and DEC 7640; and parallel computation with the MasPar MP-2 and 4096 processor elements (PEs).

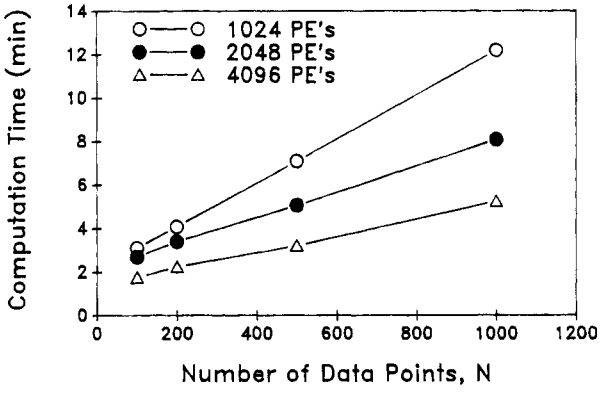

Figure 3. Computational efficiency of the EM algorithm with the massively parallel MP-2 as a function of the number of processor elements (PEs) utilized: number of grid points = 200, number of iterations = 20 000.

analyst can simply wait for the results at the terminal vs submitting a job for one half to several hours, depending on the available CPU. This allows the testing of multiple data sets and several models within one work session. Personal computing results can be inferred from experience, with 33 MHz 386SX CPUs lying closer to the VAX 6420 line and 66 MHz 486DX CPUs lying closer to the VAX 7640 line.

Figure 3 shows the effect of the number of available processors in parallel computing with the MP-2 for the current application. The efficiency increases as the number of PEs available increases. This enhancement also increases with increased data. The MP-2 at the University of Tennessee contains 4096 processors, and MasPar currently offers configurations of up to 16 384 processors. As of this writing, more than 100 of the MP-series computers have been sold. As parallel supercomputing increases in popularity, more of

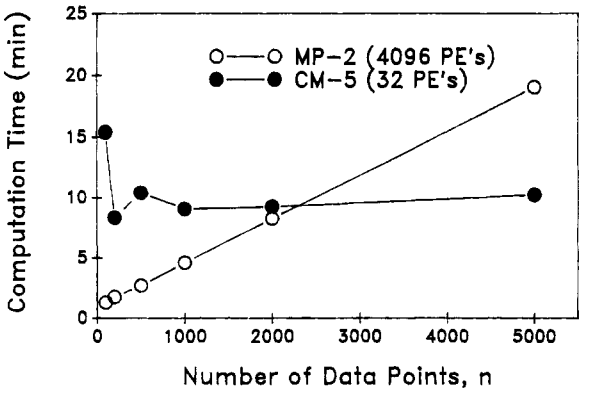

Figure 4. Computation efficiency of the EM algorithm with the supercomputers used in this study: number of grid points = 200; number of iterations = 20 000; massively parallel approach with the MP-2; and moderately parallel approach, with high-power CPUs, with the CM-5.

these types of massively parallel computers will become available at lower prices for time-sharing by physical scientists. It may be noted that the performance/PE for the different MP-2 configurations becomes significantly smaller as the number of PEs is increased, where performance/PE = (time × no. PEs)$^{-1}$ from Table 2. This is mainly due to underutilization of the machine or a load-balancing problem (i.e., 1024 PEs "fit" the data better, which only goes up to 1000 elements in Table 2). Such considerations may be important, however, when considering CPU time-charges or the initial overhead of purchasing larger arrays of PEs.

Figure 4 shows the results of the MP-2 (4096 PEs) vs the CM-5 supercomputer. The more powerful (and expensive) CM-5 computer does not outperform the MP-2 until very large data sets or dimensions are incurred, i.e., $m \times n > 2300 \times 200 = 460\,000$. For the current application of adsorption isotherm transformation to adsorption energy distributions, this dimensionality is rarely approached; however, it is quite conceivable that other types of problems with eq 1 could encounter such dimensions, e.g., rate constant distributions from raw first-order kinetic data.[17] The corresponding times in Table 1 were obtained with the array dimension declarations specified by the exact number of data and grid points for each specific case. This allowed for an increase in performance, as can be seen by comparing the MP-2 (4096 PE) times in Tables 1 and 2. It is interesting to note the apparent instability of the CM-5 execution for the smaller dimensional problems. Although time-sharing does cause variability in the reported execution times for the CM-5, several trials confirmed this variability to be insignificant in comparison to the fluctuation shown in Figure 4. The exact cause of this behavior is not known at present but is most likely due to improper or inefficient use of the vector units.

The estimation errors for the above experiments indicate that the error decreases only marginally as the number of points increases and becomes constant at large n. The same analysis holds true if n is held constant and the number of grid points, m, is varied, since the problem is symmetric in this manner. This indicates that it is not highly beneficial to increase the data density or the grid mesh size to very large values because the solution does not improve significantly, whereas the computation time increases greatly. This conclusion holds for the notion of data or mesh density; note that if a constant density is maintained and the total number of points must be increased to accommodate a larger range of data, the extra points are indeed needed. Also note that fitting the experimental data to a spline with subsequent analysis significantly increases the estimation error in this work (rms = 0.00952 for the raw data and $n = m = 418$ vs rms = 0.0177 for the splined data and $n = 500$, $m = 200$). This is mostly because the spline is evaluated at equal intervals across the data range, when in fact the experimental data density is not constant but more concentrated in the lower pressure region. The data in this lower pressure region is more accurate, hence the improved estimation error when using the experimental pressure points.

These results suggest that as parallel computing becomes more widespread, higher order numerical algorithms may be effectively utilized to solve problems which previously could only be executed practically for lower dimensional problems and with faster approximation algorithms. For applications of eq 1, iterative maximum-likelihood and maximum-entropy methods become executable on the minute scale, while retaining their desirable asymptotic properties. This attribute could accelerate research in the solution of these types of problems.

SUMMARY

The iterative EM method of solution of linear Fredholm integrals of the first kind has been demonstrated on parallel supercomputers with two different architectures. Due to the simplicity of the algorithm, parallelization of the code is straightforward. The increased computation efficiency places the method as a viable tool for these types of problems by allowing a solution within a few to several minutes. The massively parallel approach of the MasPar MP-2 yields the most impressive gains in computation times, surpassing that of the more powerful Thinking Machines CM-5 which possesses far fewer processing nodes; however, the CM-5 will eventually surpass and substantially outperform the MP-2 for very large problems. These gains were obtained for an algorithm which is not inherently or obviously parallel, suggesting that the computational solution of many other types of problems incorporating vector-matrix operations can be optimized with these types of computers.

ACKNOWLEDGMENT

The authors gratefully acknowledge Siddharthan Ramachandramurthi of the Joint Institute for Computational Science at the University of Tennessee, Knoxville, for the CM-5 timings, Sally Chase of MasPar Computer Corporation for technical support, the Joint Institute for Computational Science for accounts on the MP-2 and CM-5, the University of Tennessee Computing Center for VAXcluster accounts and support, DOE Grant DE-FG05-88ER13859, and the cooperative agreement between the University of Tennessee and Oak Ridge National Laboratory.

REFERENCES AND NOTES

(1) Rudzinski, W.; Everett, D. H. *Adsorption of Gases on Heterogeneous Surfaces*; Academic Press: New York, 1992.
(2) Jaroniec, M.; Madey, R. *Physical Adsorption on Heterogeneous Solids*; Elsevier: Amsterdam, 1988.
(3) Olson, S. L.; Shuman, M. S. Kinetic Spectrum Method for Analysis of Simultaneous, First-Order Reactions and Application to Copper-(II) Dissociation from Aquatic Macromolecules. *Anal. Chem.* **1983**, *55*, 1103−1107.

(4) Holmes, T. J.; Liu, Y-H. Richardson Lucy/Maximum Likelihood Image Restoration Algorithm for Fluorescence Microscopy: Further Testing. *Applied Optics* **1989**, *28*, 4930−4938.
(5) Richardson, W. H. Bayesian-Based Iterative Method of Image Restoration. *J. Opt. Soc. Am.* **1972**, *62*, 55−59.
(6) Tikhonov, A. N.; Arsenin, B. Ya. *Methods for Solution of III-posed Problems*; Nauka: Moscow, 1986.
(7) Wahba, G. Practical Approximate Solutions to Linear Operator Equations when the Data are Noisy. *SIAM J. Numer. Anal.* **1977**, *14*, 651−667.
(8) Provencher, S. W. A Constrained Regularization Method for Inverting Data Represented by Linear Algebraic or Integral Equations. *Comp. Phys. Comm.* **1982**, *27*, 213−227.
(9) Golub, G. H.; Van Loan, F. C. *Matrix Computations*; The Johns Hopkins University Press: Baltimore, 1983.
(10) Bialkowski, S. E. Expectation-Maximization Algorithm for Regression, Deconvolution and Smoothing of Shot-Noise Limited Data. *J. Chemometrics* **1991**, *5*, 211−225.
(11) Stanley, B. J.; Bialkowski, S. E.; Marshall, D. B. Analysis of First-Order Rate Constant Spectra with Regularized Least-Squares and Expectation-Maximization. 1. Theory and Numeration Characterization. *Anal Chem.* **1993**, *65*, 259−267.
(12) Stanley, B. J.; Guiochon, G. Numerical Estimation of Adsorption Energy Distributions from Adsorption Isotherm Data with the Expectation-Maximization Method. *J. Phys. Chem.* **1993**, *97*, 8098−8104.
(13) Almasi, G.; Gottlieb, A. *Highly Parallel Computing*; Benjamin/Cummings, 1989.
(14) Conder, J. R.; Young, C. L. *Physicochemical Measurement by Gas Chromatography*; Wiley: New York, 1979; Chapter 9.
(15) Roles, J.; Guiochon, G. Experimental Determination of Adsorption Isotherm Data for the Study of the Surface Energy Distributions of Various Solid Surfaces by Inverse Gas-Solid Chromatography. *J. Chromatogr.* **1992**, *591*, 233−243.
(16) Langmuir, I. *J.* Am. *Chem. Soc.* **1918**, *16*, 490.
(17) Stanley, B. J.; Topper, K.; Marshall, D. B. Analysis of the Heterogeneous Rate of Dissociation of Cu(II) from Humic and Fulvic Acids by Statistical Deconvolution. *Anal. Chem. Acta* **1994**, *287*, 25−34.

CI940086D