AN ABSTRACT OF THE THESIS OF

Ping-Jung Chou for the degree of Doctor of Philosophy in Biochemistry and Biophysics presented on November 3, 1993.

Title: Base Inclinations in Natural and Synthetic DNAs

Abstract Approved: Redacted for Privacy

Dr. W. Curtis Johnson, Jr.

A sophisticated computer program is developed to analyze flow linear dichroism data on nucleic acids for individual base inclinations. Measured absorption and linear dichroism data for synthetic AT and GC polymers and natural DNAs are analyzed. The reliability of the program is tested on data for the synthetic polymers, and the results are similar to earlier, more straightforward analyses. For the first time, specific base inclinations are derived for all bases individually from the linear dichroism data for natural deoxyribonucleic acids. For B-form DNA in aqueous solution at moderate salt concentrations, the inclinations from perpendicular are as follows: d(A) = 16.1 ± 0.5; d(T) = 25.0 ± 0.9; d(G) = 18.0 ± 0.6; d(C) = 25.1 ± 0.8 deg. Our results indicate that the bases in synthetic and natural DNAs are not perpendicular to the helix axis, even in the B form.

The mathematical bases and numerical analyses are presented in detail since both are the keys for successful spectral decompositions in this study, and could be applied to nonlinear optimization problems encountered in other types of biochemistry and biophysics measurements. The interplay between computer programming and scientific measurements can not be overemphasized for modern research.

Base Inclinations in Natural and Synthetic DNAs

by

Ping-Jung Chou

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirement for the
degree of

Doctor of Philosophy

Completed November 3, 1993

Commencement June 1994

APPROVED:

Redacted for Privacy

Professor of Biochemistry and Biophysics in charge of major

Redacted for Privacy

Head of Department of Biochemistry and Biophysics

Redacted for Privacy

Dean of Graduate School

Date thesis is presented November 3, 1993

Typed by Ping-Jung Chou for Ping-Jung Chou

ACKNOWLEDGEMENTS

I wish to thank J. D. Watson and F. H. C. Crick for bringing the DNA structure to the world. Their DNA model is the ultimate source and target for the studies leading to this thesis.

I would also like to thank Steve Jobs for his invention of personal computer. It is the APPLE II computer that showed me the fun of programming and the power of computing.

I'm in great appreciation of the country of the U.S.A., the state of Oregon and the town of Corvallis. They provided me with continuous financial support and unequalled living environment during the past five years.

I thank Dr. Joseph Nibler for showing me the beauty of physical chemistry: every H^+ in my body are spinning in coherence with the earth's magnetic field.

To Dr. Michael Schuyler with my heart. While everyone knows I'm a computer-holic, he appreciates how difficult it is to become one.

I thank Dr. Shing Ho for giving me a chance to prove myself that it pays to learn mathematics and think hard, and be responsible to myself.

I thank Dr. Curtis Johnson for raising me. He never said NO or MUST to me, although sometimes I wish he did.

I thank my wife, Shi-Jean Cho, for her HPLC (Heart, Patience, Love, and Care), and for giving birth to our daughter, Iris Chou.

TABLE OF CONTENTS

APPENDIX

LIST OF FIGURES

LIST OF TABLES

Base Inclinations in Natural and Synthetic DNAs

SECTION I

Introduction

Two Types of Experiments

Numerical data obtained from biochemical or biophysical experiments often required further processing and analysis in order to extract properties of the molecular system of interest. Some experiments are well designed in that the derivation of the formula describing the properties of the molecular system, the instruments and measurements that recording the data, and the analyses that process the data ascribed to the formula, are tightly coupled. Correct results can not be obtained if any of the three factors (formula, data and analysis) fails.

There are also experiments of the trial-and-error type. The formula is either missing or crudely constructed from certain assumptions, the instruments are not quite right for the job, or the method of analysis is not well specified. When these three factors are not well related to each other, the results are uncertain. Nevertheless, experiments of this type often pioneer new research directions.

This thesis focuses on the role and nature of the analyses for a well designed experiment.

Three Classes of Analyses

The way numerical data is processed can be divided into three classes. In the first class, experimental data require only simple numerical or statistical methods such as computing the average, standard deviation, or linear regression. For example, using linear regression to determine the Michaelis constant, K_M, and the maximal reaction velocity, V_{max}, from an enzyme-catalyzed reaction is a first class analysis. In the second class, the experimental data are functions of high complexity that are nonlinear in the

variables. Such functions demand robust and efficient optimization algorithms and extensive computer and human resources. Most importantly, the method of analysis must be developed, tested and proven. The mapping from X-ray diffraction pattern to atomic coordinates for the molecules is one example of a second class analysis. The third class of analysis is essentially the same as the second class, but differs in that the experimental data can not be fully analyzed in accordance to the specified formula because (1) computers with enough power are not available, and (2) the analysis applied to the experimental data must be designed around a downsized mathematical model. Disciplines of mathematics, numerical analysis and computer science are generally not emphasized for biochemistry and biophysics studies, and as a result, researchers may not be capable of developing and performing in-depth analyses for extracting reliable information from their measurements.

A Third Class Analysis

One example of a third class analysis is previous methods for analyzing the linear dichroism (LD) of DNA (this example is actually background for the study presented in SECTION II; see references 32 and 33 in BIBLIOGRAPHY, SECTION II for details). The goal of the experiment is to determine for DNA in solution: (1) whether the bases are perpendicular to the helical axis, and (2) if DNA bases are not perpendicular to the helical axis, then (2) at what angle the bases incline. In this experiment, absorption and LD spectra are measured in the UV for a DNA solution. For natural DNA with all four types of bases, there is a total of 16 absorption bands (four from adenine, three from thymine, four from guanine, and five from cytosine) that sum to give the spectra measured to 175 nm. If bands are assumed to be of Gaussian shape, they are determined by three variables: position (wavelength maximum), intensity and band width.

$$\mathrm{ABS}(\lambda) = \sum_{j=1}^{4} \sum_{i=1}^{N_j} G(\lambda, P_{ij}, I_{ij}, W_{ij})$$

in which N_j is the number of bands for base j, λ is wavelength, G is the Gaussian function, and P_{ij}, I_{ij} and W_{ij} are position, intensity and width, respectively, for the band i of base j. The LD spectrum is linked to the absorption spectrum through four pairs of geometrical parameters (one pair for each base) according to the following expression:

$$LD(\lambda) = \sum_{j=1}^{4} \sum_{i=1}^{N_i} G(\lambda, P_{ij}, I_{ij}, W_{ij}) 3[3\sin^2\alpha_j \sin^2(\chi_j - \delta_{ij})]/2$$

in which α_j and χ_j are the pair of parameters which define the inclination and axis of inclination, respectively, for base j. The δ_{ij} is a geometrical parameter defining the transition dipole associated with each band, and its value is determined from other experiments.

From the analytical point of view, the goal of the experiment can be stated as follows: given the formula for the absorption and LD spectra, and the measured absorption and LD spectra for DNA in solution, determine the most probable inclination angle, α, for each base of that DNA sample. Clearly, α for a given base can not be determined without the paired χ being determined at the same time, and both α and χ can not be determined without parameters for all bands of that base being determined first. Furthermore, all variables for the four bases must be determined simultaneously. In terms of numerical analysis, this is a nonlinear optimization problem with 56 variables to be determined.

Let us see how the task of optimizing 56 variables was accomplished in all previous work:

1. Instead of fitting absorption and LD spectra simultaneously, the LD spectrum was divided by the absorption spectrum to obtain the reduced LD spectrum (L), and L was used for the fitting.

$$L(\lambda) = LD(\lambda)/ABS(\lambda)$$

Since different pairs of dividend and divisor can give the same quotient,

important information is lost. Further more, a good fitting for the L spectrum does not automatically translate to a good fitting for both absorption and LD spectra.

2. Only synthetic polymers consisting of two bases (A/T and G/C) were computed. It would take too long and too much computer memory to compute four-base natural DNAs, and the results would have higher uncertainties.

3. The position and width of a band from a given base were fixed for all synthetic polymers containing that base. The number of variables was effectively reduced by more than half at the cost of using inaccurate fitting parameters.

4. The simplex algorithm was used to solve the nonlinear optimization problem. This is a lightweight algorithm: easy implementation, less progression per function evaluation, no estimate of errors for a computed solution, and easily trapped by local minima.

Clearly there is room for improvement in the way experimental LD data are analyzed. How the improvement can be achieved, and deficiencies be removed, in order to bring this analysis from the third class to the second class, is the major topic of this thesis.

The Requirements for a Successful Analysis

During the past few years the advance in computer technology (in both hardware and software) has not only brought powerful computers into the laboratory, but also provided a great opportunity for researchers to tackle complicated data analyses that were once thought to be very difficult or time consuming. The wave of better and faster computers should not be viewed as just tools for data processing, however. With this new technology, we are now freed from computational constraints, from having to truncate and cripple a complex but complete expression for analyzing an experiment. Researchers can now look at a large scale problem (in terms of computer resources), and

redesigned their experiments and methods of analysis to have their questions answered. It is a mutual interaction between the computer technology and the way of thinking in scientific research. It is not just an one-way application of computer programs to analyze experimental data.

However, it takes more than faster computers for a successful analysis. The skill in programming and the knowledge of numerical methods are essential in developing new analytical tools for a given experiment. The choice of the particular formula or algorithm, and the way they are implemented, influence not only the computing itself but also how we understand and interpret the results when they are obtained. The speed the computing progresses, the number of iterations required, and the way the error progresses, give not only the insight into the problem, but also into the suitability of the numerical methods used.

A complicated numerical analysis also demands an understanding of the architecture of the computer on which the program is running. Floating-point representation of real numbers on most modern computing machines has finite precision and as a result, roundoff errors arise and are passed on from one arithmetic operation to the next as the computation progresses. More often than not, the greatest loss in significant figures occurs when two numbers of about the same size are subtracted, so that most of the leading digits cancel out. How to estimate roundoff error (or better yet, how to avoid roundoff error), is essential for a stable and predictable implementation of numerical algorithm.

Also there is the task of debugging. While debugging of the initial program is not considered by many biochemistry and biophysics researchers, it is the core of programming. Experience of the programmer with the specific computer language, operating system, and algorithm, dictate the difficulty one can expect during the debugging process.

A Second Class Analysis

Now, let us see how the requirements for an analysis of LD experiments on DNA are met, so that it is reduced to second class:

1. Absorption and LD spectra are fitted simultaneously rather than using the reduced LD.

2. LD data are analyzed for all four bases in natural DNAs.

3. Since absorption bands are generally asymmetric, the log-normal shape is used instead of the Gaussian shape. A log-normal shape is determined by four variables (position, width, intensity and skewness). Further more, we treat all these as free variables instead of fixing them.

4. The most powerful problem solver for nonlinear optimization, the Levenberg-Marquard algorithm, is used for maximum performance.

5. Two more bands are added to adenine to make the fitting more realistic.

6. All aspects of the program can be fine tuned to fit our specific needs. For example, LU decomposition is used in this study to invert a positive definite and symmetric matrix, because it is faster than singular value decomposition and suffers less roundoff error than Gauss-Jordan elimination. Using singular value decomposition for the matrix inversion would slow down the program significantly, while using Gauss-Jordan elimination would accumulate roundoff error to such an extent that a matrix close to singular could not be inverted successfully.

Preview of SECTION II

The body of this thesis contained in SECTION II. It is based on the manuscript for a published paper in *J. Am. Chem. Soc.* **1993**, *115*, pp. 1205-1214, entitled: **Base inclinations in Natural and Synthetic DNAs.** An extensive literature review on the study of base inclinations in DNAs is in the INTRODUCTION section of the manuscript. Following the INTRODUCTION

section is the METHODS section, in which mathematical and numerical aspects of the analysis methods for the absorption and linear dichroism spectra for monomer and polymer DNAs are described. Since all spectra analyzed in this study were measured previously in our laboratory, sample preparations and measurements are left out from the METHODS section. It is then followed by the RESULTS and DISCUSSION section.

In order to make this thesis a complete presentation, there are some significant changes from the published paper. The heading **Algorithm** in the METHODS section has been replaced by **Nonlinear Least Squares Fitting,** which now contains the derivation of the nonlinear optimization algorithm described in more detail. A new heading (**LD Spectra for the Hypothetic Single Stranded Poly[d(T)]**) and a new figure (Figure 6) are added to the RESULTS and DISCUSSION section to illustrate how LD spectrum changes as a function of inclination angles and axes of inclination. Two figures (Figure 10a and 10b) and a paragraph are also added to the RESULTS and DISCUSSION section (**Repeated Fittings with Randomized Transition Dipole Directions**) to show that the inclination angles and axes of inclination obtained in this study are very stable relative to small variations in the transition dipole directions. A completely new section, RECENT DISCOVERIES, along with two figures (Figures 11 and 12) have been added to explain interesting questions observed in this study that were not understood when the original manuscript was written. Finally, the usage and source code of the two computer programs, *ABS-LD* and *PARSE-R*, written and used in this study, are included in an APPENDIX. These two programs are part of the methods and results; they deserve a place in this thesis.

SECTION II

Base Inclinations in Natural and Synthetic DNAs

Ping-Jung Chou and W. Curtis Johnson, Jr.

Department of Biochemistry and Biophysics
Oregon State University
Agricultural and Life Sciences 2011
Corvallis, OR 97331-7305

J. Am. Chem. Soc. **1993**, *115*, 1205-1214

ABSTRACT

A sophisticated computer program is developed to analyze flow linear dichroism data on nucleic acids for individual base inclinations. Measured absorption and linear dichroism data for synthetic AT and GC polymers and natural DNAs are analyzed. The reliability of the program is tested on data for the synthetic polymers, and the results are similar to earlier, more straightforward analyses. For the first time, specific base inclinations are derived for all bases individually from the linear dichroism data for natural deoxyribonucleic acids. For B-form DNA in aqueous solution at moderate salt concentrations, the inclinations from perpendicular are as follows: d(A) = 16.1 ± 0.5; d(T) = 25.0 ± 0.9; d(G) = 18.0 ± 0.6; d(C) = 25.1 ± 0.8 deg. Our results indicate that the bases in synthetic and natural DNAs are not perpendicular to the helix axis, even in the B form.

INTRODUCTION

Watson and Crick depicted their helical structure for DNA with 10 base pairs per turn with the bases perpendicular to the helix axis. This was consistent with Wilkins' X-ray patterns for fibers of DNA at high humidity, the B-form. Although the data in diffraction patterns from fibers are limited, subsequent model building indicated a 10-fold repeat with bases perpendicular to the helix axis for the B form.[1] However, DNA is known to be polymorphic,[2-9] with the particular structure sensitive to sequence, cation type, temperature, and solvent (or, in the case of fibers and crystals, the humidity). A structural model built on X-ray diffraction, however, may depend on packing forces, and not actually exist in solution where DNA molecules are relatively free.

Linear dichroism (LD) is a method for determining the inclination angle (the total effect of tilt, propeller twist, roll, and buckle) of a given kind of base in a DNA molecule in solution.[10-28] It is based on the fact that (1) each kind of base has different π-π^* transitions with dipole moments of known direction in the base plane; (2) the long DNA molecules can be aligned so that, at least on average, the helical axis lies in the direction of alignment; and (3) the anisotropic absorption of transition dipoles in a base can be expressed as a function of the base inclination angle from perpendicular to the helical axis.

DNA molecules are generally aligned either in films or fibers by the shear forces of flow or by their special polyelectrolyte properties in an orienting electric field. Of course, complete alignment is impossible. Base inclinations are deduced in the case of flow LD by modeling the shear forces in the flow cell, extrapolating to infinite shear, or making use the variation in LD as a function of wavelength. Base inclinations are usually deduced in the case of electric dichroism by making measurements at various fields and extrapolating to infinite field. The orientation problem may be further complicated by the

possible existence of tertiary superstructures, which would prevent complete alignment of the helix axis in the direction of alignment even in infinite shear or infinite field. Recent recognition that bent DNA does exist, typified by the kinetoplast fragments, means tertiary superstructures deserve serious consideration. Detailed reviews have been written covering these points.[12,14,15,27]

In an LD measurement the absorption is measured parallel and perpendicular to the direction of alignment at one or more wavelength, and the data are conveniently expressed as the reduced dichroism given by

$$L(\lambda) = \frac{[A_{\parallel}(\lambda) - A_{\perp}(\lambda)]}{A(\lambda)} = \frac{LD(\lambda)}{A(\lambda)} \qquad \text{Eq. 1}$$

where $A(\lambda)$ is the normal isotropic absorption at wavelength λ. If the base planes in B-form DNA are nearly perpendicular to the helix axis, then for complete alignment in the absence of complicating factors, $L(\lambda)$ will be -1.5 for the in-plane π-π^* transitions, regardless of the wavelength and the corresponding transition dipole directions.

Most electric dichroism work since 1978 has utilized samples of homogeneous length and reduced dichroism at the absorption maximum of 260 nm extrapolated to infinite field.[16,18-20] Measurements have been made on different DNA lengths, with the idea that it should be easier to obtain complete alignment for short lengths of DNA without exterior complications. However, considering all of the data together, it is clear that the shorter the DNA length the lower the magnitude of the negative L(260 nm). At one extreme Lee and Charney[19] obtained -1.41 for a DNA length of 9200 base pairs, while Hogan et al.[16] obtained -1.11 for a DNA length of 154 base pairs at the other extreme. Hogan et al.[16] interpreted their data in terms of a base inclination from perpendicular of about 17°. In contrast, Dieckmann et al.[20] and Lee and Charney[19] noted that a bent tertiary structure in the DNA would rationalize the

value for L(260 nm) as a function of DNA length; as the DNA length increases it is presumed that the DNA becomes increasingly straight in the orienting electric field. This data would still be consistent with the bases perpendicular to the helix axis if (1) extrapolation to infinite field are not correct or (2) the DNA has a tertiary superstructure so that complete alignment is impossible. Rau and Charney[29] has questioned the extrapolation to infinite field and have provide a model for the orientation of DNA as a function of field that explains the observed data. When everything is taken into consideration, Charney et al,[25] believe that L(260 nm) = -1.41 for the long DNA molecules is consistent with the Watson-Crick structure and an average base inclination of about 10° from X-ray studies on fibers.[1]

Flow LD measurements also give a negative reduced dichroism for B-form DNA.[10-12,17,22-24] The data are independent of wavelength between 280 and 250 nm, suggesting that the bases are perpendicular to the helix axis. The reduced dichroism is less negative in the 250 to 220 nm region, and this change in L has been presumed to be due to out-of-plane n-π^* transitions.[11] In general, workers have interpreted their LD data as being consistent with the Watson-Crick model. Our laboratory has extended the LD measurements of nucleic acids into the vacuum UV region to 175 nm.[24,30-33] Our data over this extended range show a reduced dichroism that varies with wavelength for natural B-form DNA,[24,31] indicating that the bases are not perpendicular to the helix axis.

It is not straightforward to relate either electric or flow LD data to base inclinations; the measurement depends not only on base inclinations, but also on the angle that the dipole for each transition makes with the axis around which the base is inclining. However, with this extended data we were able to compare the relative values of the reduced dichroism for the 260- and 220-nm π-π^* regions to obtain a minimum average base inclination from perpendicular of about 15° for standard B-form DNA.[24,31] We do not attempt to model our flow

or extrapolate our data to infinite alignment. The beauty of extending the LD data to shorter wavelengths is that absolute measurements are not necessary, and base inclinations can be determined from the wavelength dependence (overall spectral shape) of the data. DNA tertiary structure, such as a superhelical coil or simple bending, affects LD as a multiplicative factor, which affects the values at infinite field or flow but which does not affect the wavelength dependence of the data.[12,14,15,27]

We have also measured the LD of simple repeating double-stranded AT and GC polynucleotides from 320 to 175 nm.[32,33] This data can be decomposed into individual absorption bands, and since the transition dipole directions are known, it has been analyzed for inclination and axis of inclination for the various bases. The reduced dichroism for these double-stranded polynucleotides varies with wavelength, indicating that the base planes are not perpendicular to the helix axis. Many workers believe that loss of negative reduced dichroism around 230 nm is due to an n-π^* transition with an out-of-plane transition dipole. We analyze our data without the 245-212 nm spectral region, and the wavelength dependence of the data still predicted significant inclinations for the bases. Furthermore, the 230-nm feature in the reduced dichroism was found to be due to the angle that the π-π^* transition dipoles made with inclination axis in this region, and existence of an out-of-plane n-π^* need not be postulated to explain the measurements. If the minimum magnitude of the reduced dichroism for B-form DNA at 223 nm is compared with the maximum magnitude at 260 nm, a minimum average base inclination of about 19° is derived for natural DNA in the standard B form.[24]

Here we develop a sophisticated algorithm in order to analyze the LD data of natural nucleic acids as a function of wavelength for individual base inclinations and axes of inclination. With an algorithm that relies so heavily on the computer, it is important to be sure that the results are not an artifact generated by the computer. So we use this new method to reanalyze the data

for the synthetic AT and GC polynucleotides, which were analyzed in a more straightforward way in the original publications.[32,33] The results of this new method are in reasonable agreement with the results of the original, simpler analyses. Furthermore, the inclinations and axes of inclination that we derived for the individual bases in B-form DNA predict L(260 nm) of -1.40 for perfect alignment of the DNA helix axis along the direction of orientation. This agrees with the values obtained by extrapolating electric dichroism data to infinite field for monodispersed samples of long DNAs[19,20] and supports the argument that large electric field should overwhelm configurational and thermal bending for long DNAs.[27,29] The fact that L(260 nm) = -1.40 at perfect orientation can correspond to significant base inclinations, demonstrates that it is important to take into account the relative orientation of transition dipole to the axes around which the bases incline when interpreting LD data.

Flemming et al.[28] have used infrared LD to investigate the base inclination of A- and B-form DNA in oriented films. They find inclinations from perpendicular of 28-30° for the A form and 18-30° for the B form, in agreement with our work. Theoretical calculations support large base inclinations in DNA.[34,35] In particular, Sarai et al.[35] find that the origin of the B-form double helix can be attributed in large part to the atomic charge pattern in the base pairs. That is, the base pairs alone have a strong tendency to form a helical structure independent of the backbone. Further, propeller twisting is found to enhance the electrostatic interaction by positioning favored atom pairs closer together. One might expect that, in aqueous solution where the DNA is free of the packing effects found in crystals and fibers, bases may be freer to assume larger propeller twists with the concomitant larger base inclination in order to maximize favorable base-base interactions. Ansevin and Wang[36] have proposed a new model for the Z-form with a fair base inclination. Edmondson used the molecular mechanical program AMBER[37] to investigate the potential energy of conformations consistent with his LD results for poly[d(A)-d(T)].[38] He

found that the large 50° propeller twist maximizes intrastrand base-stacking interactions, and that the total potential energy was comparable to that calculated for X-ray diffraction models of DNA. Large propeller twists do not really preclude hydrogen bonding, because hydrogen bonds are not very directional.

Here we present the results of analyses of synthetic polymers and natural DNAs using our new algorithm and recently determined transition dipole directions. Large inclinations are confirmed for the bases in synthetic polymers, and specific inclinations are determined for the first time for the bases in natural DNAs.

METHODS

Nonlinear Least Squares Fitting

Suppose we are fitting a measured spectrum $y(\lambda_i)$, i=1,...,m, to an analytical function $Y(\lambda_i,\mathbf{x})$, where **x** is a vector of n parameters (unknown variables). The relationship between y and Y can be expressed as

$$y(\lambda_i) = Y(\lambda_i,\mathbf{x}) + e_i \qquad \text{Eq. 2}$$

in which e_i is the measurement error associated with $y(\lambda_i)$. If we assume that the e_i's are normally distributed and independently random, and apply Maximum Likelihood Estimation to Eq. 2, it can be shown that the best solution (the most likely values) for the n variables in **x** can be determined by minimizing

$$F(\mathbf{x}) = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y(\lambda_i) - Y(\lambda_i,\mathbf{x})]^2 \qquad \text{Eq. 3}$$

If the analytical function Y linearly depends on **x**, then Eq. 3 is a linear least squares minimization and the exact solution for **x** can be calculated by solving the following set of simultaneous equations,

$$\frac{\partial F(\mathbf{x})}{\partial x_j} = 0, \qquad j = 1,...,n \qquad \text{Eq. 4}$$

However, as we will see later, the function Y of this study is nonlinear and we have to resort to other indirect methods. Based on preliminary studies, the method we chose to solve this nonlinear least squares minimization problem is the Levenberg-Marquardt algorithm,[39] abbreviated as LM. First let's express

F(**x**) as a Taylor series expansion around an initial guess of **x**, $\mathbf{x}_0$, up to the second order,

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)\,\mathbf{F}'(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^2\,\mathbf{F}''(\mathbf{x}_0) \qquad \text{Eq. 5}$$

Now Eq. 5 is linear in **x**, so linear least squares minimization (Eq. 4) can be applied,

$$\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{F}'(\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)\,\mathbf{F}''(\mathbf{x}_0) = 0$$

$$\mathbf{x} = \mathbf{x}_0 - \mathbf{F}''(\mathbf{x}_0)^{-1}\mathbf{F}'(\mathbf{x}_0) \qquad \text{Eq. 6}$$

Because the Taylor series expansion of F is truncated after the second order, Eq. 6 will not bring us the optimum solution for **x** in just one step after an initial guess. Instead, if we replace **x** with $\mathbf{x}_{k+1}$ and $\mathbf{x}_0$ with $\mathbf{x}_k$, where k is a sequence of iterations, and apply the LM algorithm to Eq. 6, we can rewrite Eq. 6 as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [C_k\mathbf{D} + \mathbf{J}(\mathbf{x}_k)^T\mathbf{J}(\mathbf{x}_k)]^{-1}\,\mathbf{J}(\mathbf{x}_k)^T\mathbf{F}(\mathbf{x}_k) \qquad \text{Eq. 7}$$

in which C_k is the LM coefficient, $\mathbf{J}(\mathbf{x}_k)$ is an m-by-n numerical Jacobian matrix evaluated at $\mathbf{x}_k$, and D is a diagonal matrix with entries equivalent to the diagonal of $\mathbf{J}(\mathbf{x}_k)^T\mathbf{J}(\mathbf{x}_k)$. The i^{th} row (i=1,...,m) and j^{th} columns (j=1,...,n) of the Jacobian matrix at each iteration is calculated by

$$\frac{\mathbf{F}_i(\mathbf{x}+h\mathbf{u}_j) - \mathbf{F}_i(\mathbf{x})}{h}$$

in which u_j is the j^{th} unit vector and h is a small real number used to

approximate the first partial derivative of F_i with respect to x_j.

The Levenberg-Marquardt coefficient C_k in Eq. 7 is systematically updated according to results of the previous iteration. This allows the behavior of LM to switch smoothly between Gauss-Newton and steepest descent algorithms, and it is this flexibility that allows LM to locate the global minimum within a multidimensional space much faster than, say, Powell's conjugate gradient algorithm[40] used in our preliminary studies.

The diagonal elements of the n-by-n matrix $[J(\mathbf{x}_k)^T J(\mathbf{x}_k)]^{-1}$ are the variances of the elements in $\mathbf{x}_k$ at the k^{th} iteration if $\mathbf{F}$ is a linear function of $\mathbf{x}$, and measurement errors e_i's are normally distributed and independently random.[41] However, we do not know our error distribution, and as noted, our function is not linear. Although strictly speaking our diagonal elements are not the variances, they will be related to the true variances, and the difference between the diagonal elements for two consecutive iterations will still tell us whether $\mathbf{x}_k$ is more stable than $\mathbf{x}_{k-1}$. Of course, one can take the sum of squares error in fitting a spectrum to a minimum, but there is error in the data that is being fit so exactly. Instead we monitor the stability of $\mathbf{x}_k$ through the diagonal elements of $[J(\mathbf{x}_k)^T J(\mathbf{x}_k)]^{-1}$ and stop fitting when $\mathbf{x}_k$ is stable.

Fitting Monomer Absorption Spectra

To decompose a monomer absorption spectrum into its constituent bands, we must first choose an analytical function that can best describe the shape for each absorption band. Gaussian or Lorentzian functions are most often used in the decomposition of UV or IR spectra.[28,32] However, the shape of an UV absorption band is generally asymmetric, and this is well represented by the log-normal function.[42,43] With four parameters (band center μ, an integrated intensity ζ, width at half-height σ, and skewness ϱ), the log-normal function for a single band as a function of wavelength λ is

$$A(\lambda) = \zeta \exp\{-\frac{1}{2}[\frac{\ln(G/R)}{Z} - Z]^2\}/\sqrt{2\pi}ZG \qquad \text{if } G>0$$

$$A(\lambda) = 0 \qquad \text{if } G \leq 0$$

in which $G=\mu+R-\lambda$, $R=2\sigma\varrho/(\varrho^2-1)$, and $Z=\ln\varrho/(2\ln 2)^{1/2}$. In some cases, the skewness increased unreasonably to fit imperfect data perfectly. We limited ϱ to the range [1.0,1.5] and this limit barely affected the fit.

Thus, if a spectrum is to be decomposed into N individual bands, 4N variables would have to be determined, and the fitted spectrum (as opposing to measured spectrum) is

$$A_{base}(\lambda) = \sum_{i=1}^{N} A(\lambda, \mu_i, \zeta_i, \sigma_i, \rho_i)$$

Since we know from other work how many bands exist within the measured spectrum for each monomer,[44-48] we know the value of N for each base, which corresponds to the smallest number of bands necessary to give a satisfactory fit to the absorption spectrum.

To begin the decomposition, initial values for position μ, intensity ζ and width σ are taken from previous work[32,33,44-48] Skewness ϱ is arbitrarily assigned the value 1.2. Fittings to the monomer spectra by the LM algorithm is quite straightforward and the results are stable.

Fitting Polymer Absorption and LD Spectra

The parameters determined by fitting the absorption spectra are the initial guesses for simultaneously fitting the absorption and LD spectra for each type of polymer using the LM algorithm. The relation between isotropic absorption and LD for a transition dipole i of base j is given by[15,32,33]

$$LD_{ij}(\lambda) = A_{ij}(\lambda)3S[3\sin^2\alpha_j\sin^2(\chi_j-\delta_{ij}) - 1]/2 \qquad \text{Eq. 8}$$

in which δ_{ij} is the angle between transition dipole i and the vector $N_3 \rightarrow C_6$ if base j is a purine, or $N_1 \rightarrow C_4$ if pyrimidine; α_j is the inclination angle of base j from perpendicular to the helix axis (the result of both twist and tilt); χ_j is the angle between the in-plane axis (perpendicular to the helix axis) around which the base inclines and the vector to which δ_{ij} references; and S is the factor that makes up for imperfect orientation in the flow. The signs of χ, δ and α follows the right-handed Cartesian coordinate system, and the angles are illustrated in Figure 1.

Since our polymers in these studies contain more than one base, the absorption and LD spectra are as follows:

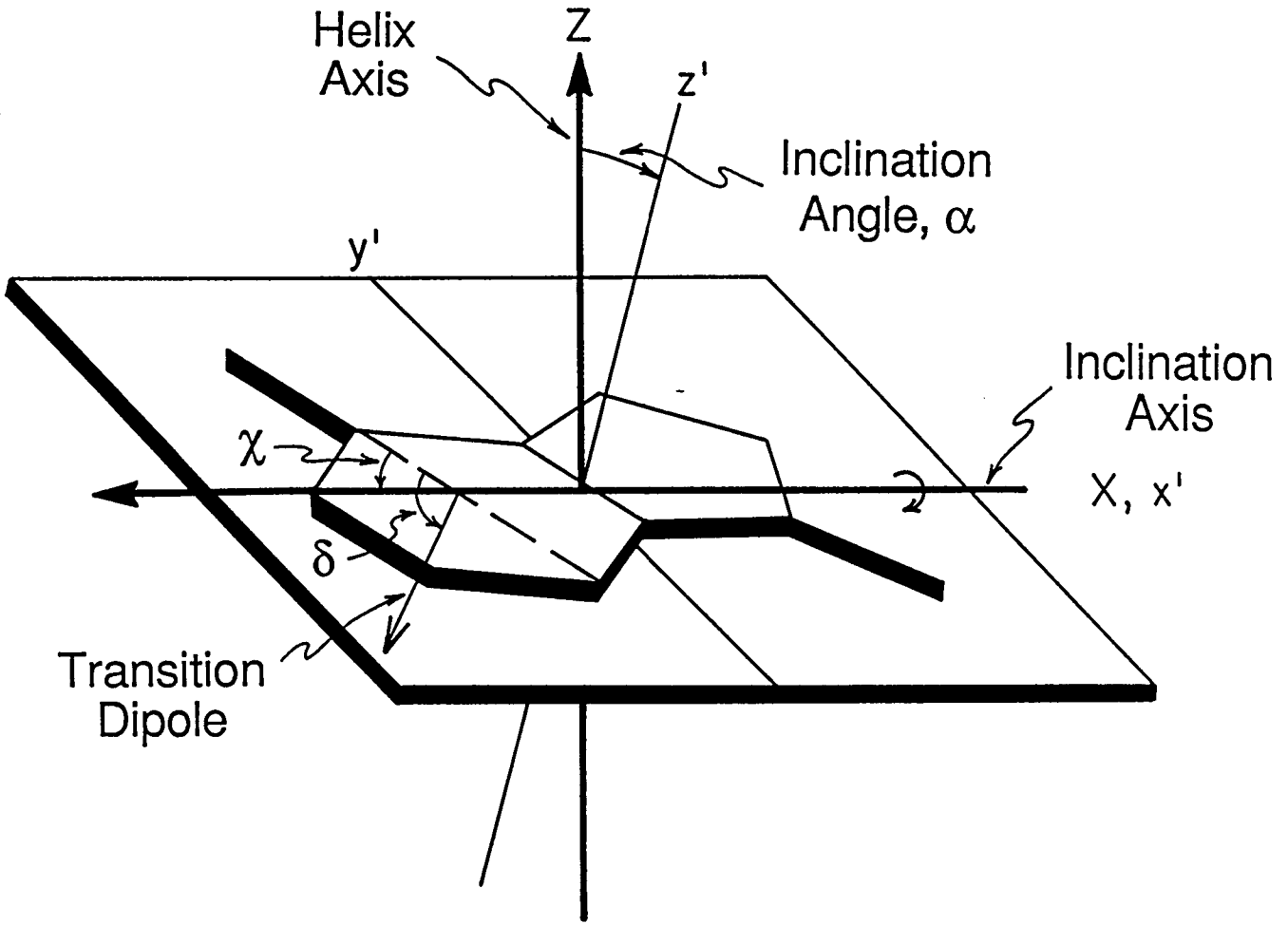
$$A_{poly}(\lambda) = \sum_{j=1}^{M}\sum_{i=1}^{N_j} A_{ij}(\lambda)$$

$$LD_{poly}(\lambda) = \sum_{j=1}^{M}\sum_{i=1}^{N_j} LD_{ij}(\lambda)$$

in which N_j is the number of transitions for the j^{th} base and M is the number of bases. We are analyzing the wavelength dependence of the data, so that imperfect orientation, including the effect of tertiary superstructures, does not affect our analysis.[12,14,15,17]

The objective is to determine parameters for all bands, and α and χ angles for the bases in the polymer, through the LM algorithm as described above, simultaneously fitting the absorption and LD spectra. We reiterate that, due to the large number of variables and different scales of measurement errors of the absorption and LD spectra, our chosen fit is at the unique point

Figure 1. Diagram showing the definition of α, χ and δ angles. Here a purine (adenine) is inclining around the X axis by an inclination angle, α. The angle between the reference $N_3 \rightarrow C_6$ and the inclination axis is χ, and between $N_3 \rightarrow C_6$ and the transition dipole is δ. Note that a pyrimidine would diagram like the six membered ring of the purine shown here, with the equivalent reference being $N_1 \rightarrow C_4$.


Helix
Axis
Z
z'
Inclination
Angle, α
y'
Inclination
Axis
χ
X, x'
δ
Transition
Dipole

along the minimization path not too far above the global minimum of residuals, at which most variables have the smallest variance. We monitor all variances after each iteration and choose the bottom of the multidimensional valley of variances as our end point.

The transition dipole direction, δ_{ij}, associating with transition i of base j, must be known to fit LD spectra, and these are taken from Clark and co-workers.[44-48] Initial values for parameters for the absorption and LD bands are those from our fitting of monomers (Table I). Initial α and χ angles for the synthetic polymers are from earlier work[24,32,33] and for DNA are from our results for the synthetic polymers.

Uncertainties in Transition Dipole Directions

The measured directions of the transition dipoles are assumed to be correct and unchanged for all polymers and DNAs studies. However, as mentioned in the reports of dipole direction measurements, there are uncertainties in these directions. To determine how the uncertainties affect the results, we repeated each fitting 100 times with transition dipole directions randomly varied within ±10°. The average value from the 100 runs for each variable (parameters for each band and α, χ angles of each base) is our reported value, and the standard deviation is for the 100 runs.

Validation of α and χ angles

A given base pair will have quantities that vary with α and χ angles of the pairing bases, such as hydrogen-bond distance and angle, distance between purine C_8 atom and pyrimidine C_6, distance between the two C_1' atoms, and propeller twist (the dihedral angle between base planes). By constructing base pairs from our α and χ angles, calculating these base-pair parameters, and comparing with published parameters, we can determine whether α and χ angles derived this way are reasonable. Another reason for

Table I. Decomposition of Monomer Absorption Spectra

Monomer	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	δ(deg)
dAMP	266.4	162.7	11.2	1.20	83[a]
	255.0	319.1	13.9	1.33	25[a]
	206.6	467.0	10.5	1.21	-45[a]
	195.3	78.7	6.1	1.38	15[a]
	184.9	282.7	7.6	1.29	72[a]
	173.6	60.2	4.5	1.00	-45[a]
TMP	265.1	363.0	18.0	1.25	-9[b]
	204.7	409.5	19.7	1.50	-53[b]
	176.6	190.7	5.8	1.42	-26[b]
dGMP	274.5	288.7	16.7	1.50	-4[c]
	248.5	309.5	13.9	1.10	-75[c]
	198.8	471.2	11.6	1.03	-71[c]
	183.2	449.4	11.6	1.50	41[c]
dCMP	269.0	301.2	15.3	1.12	6[d]
	228.1	319.2	19.8	1.31	-35[d]
	211.6	86.8	7.1	1.00	76[d]
	196.5	403.1	9.9	1.43	86[d]
	170.1	94.0	12.4	1.03	0[d]

[a]Clark[45,46]
[b]Novros and Clark.[48]
[c]Clark.[44]
[d]Zaloudek et al.[47]

this validation has to do with the sign of α. Because positive and negative α angles of the same magnitude would give the same LD spectrum, we investigated the four possible base pairings with the signs of the angles as +/+, +/-, -/+ and -/- for each base pair.

With atomic coordinates for the four bases taken from Arnott,[49] construction of a base pair begins by placing the bases in a plane (assigned to be the xy plane) perpendicular to the direction of light polarization (assigned to be the z axis). Each base plane is rotated about the inclination axis for α degrees. Because LD contains only information about a base instead of a base pair, we are free to move the two bases in space and rotate around the z axis, as long as we keep the angle between each base plane and xy plane constant. With minimal effort the two or three hydrogen-bond distances can be adjusted to an acceptable value, and then other base-pair parameters are calculated.

RESULTS and DISCUSSION

Decomposition of Monomer Absorption Spectra

Absorption spectra for dAMP, TMP, dGMP and dCMP are decomposed into 6, 3, 4, and 5 bands respectively, as shown in Figures 2-5. The position, intensity, width and skewness of each band are listed in Table I. Obviously, the parameters for the 173.6-nm band of dAMP and the 170.1-nm band of dCMP are neither well determined nor particularly relevant. However, the red end of a shorter wavelength band is necessary in this region to realistically fit the data. Also listed in Table I are the corresponding transition dipole directions, which are vital for successful decomposition of LD spectra.

During preliminary studies only four bands were used to decompose the dAMP spectrum, as with work previously done in our laboratory.[32] The result in calculating the spectrum fits the experiment except for the region between 190 and 210 nm, and according to Clark[46] a minor band is also in this region, which we now include in our fit (Figure 2). Fitting of the TMP spectrum is relatively easy because its three components are well separated (Figure 3), but these bands are not Gaussian and show that the log-normal function with its skewness parameters is more suitable to approximate electronic absorption bands. The major components of the dGMP and dCMP spectra can be distinguished as peaks or shoulders (Figures 4 and 5).

LD Spectra for a Hypothetical Single Stranded Poly[d(T)]

To illustrate how an LD spectrum changes as a function of α and χ angles, we computed two LD spectra according Eq. 8 for single stranded poly[d(T)], assuming $\alpha = 0°$, $\chi = 20°$ (LD 1) and $\alpha = 40°$, $\chi = 20°$ (LD 2). The alignment factor, S, in Eq. 8 is assumed to be 1.0 (perfect alignment), and we neglect base-base interactions to simplify visualizing the effect of base

Figure 2. Decomposition of dAMP absorption spectrum: (□□□) is measured, (▬▬) is fitted, and (——) is decomposed.

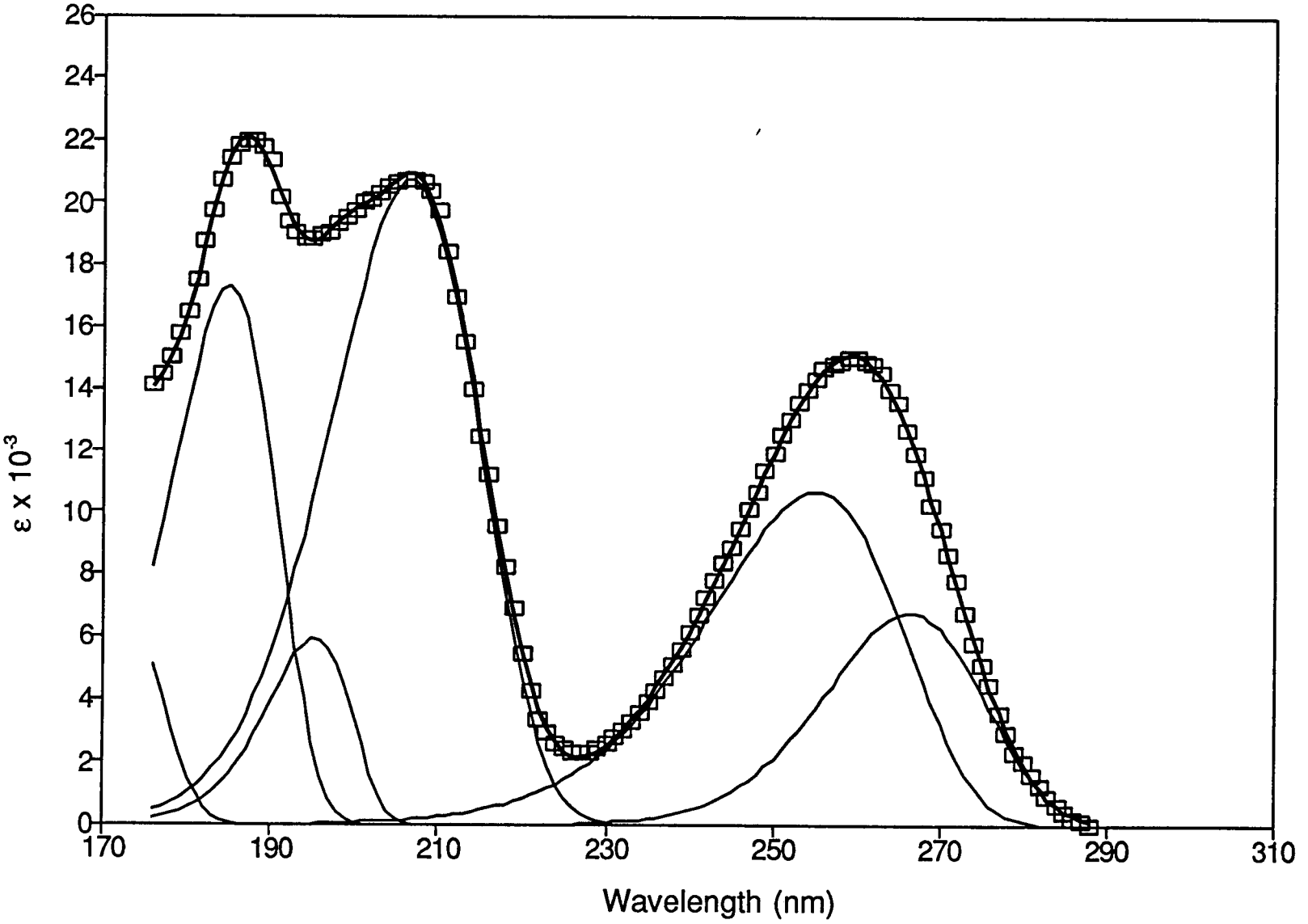

26
24
22
20
18
16
14
12
10
8
6
4
2
0
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Figure 3. Decomposition of TMP absorption spectrum: (□□□) is measured, (▬▬) is fitted, and (——) is decomposed.

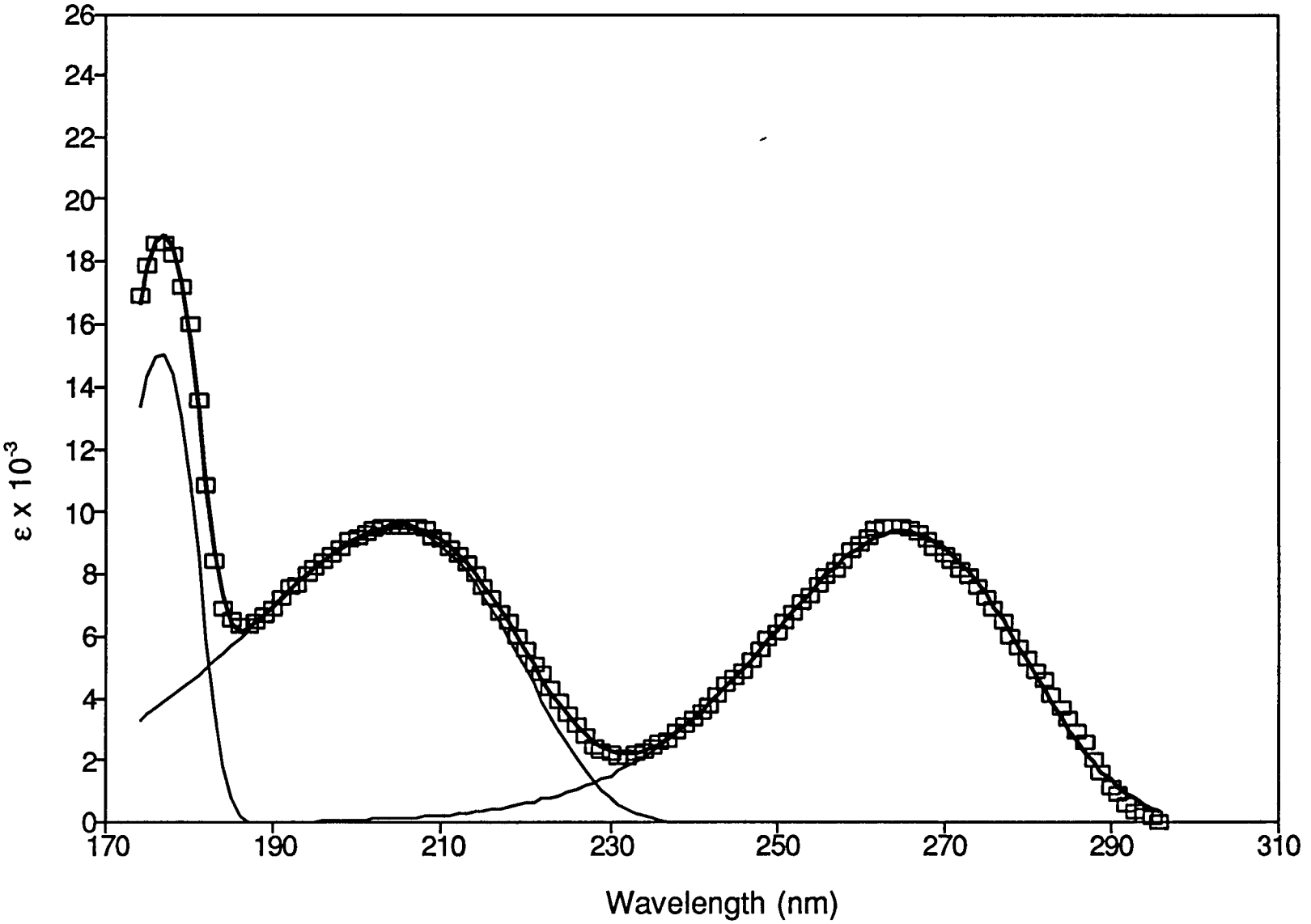

26
24
22
20
18
16
14
12
10
8
6
4
2
0
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Figure 4. Decomposition of dGMP absorption spectrum: (□□□) is measured, (▬▬) is fitted, and (——) is decomposed.

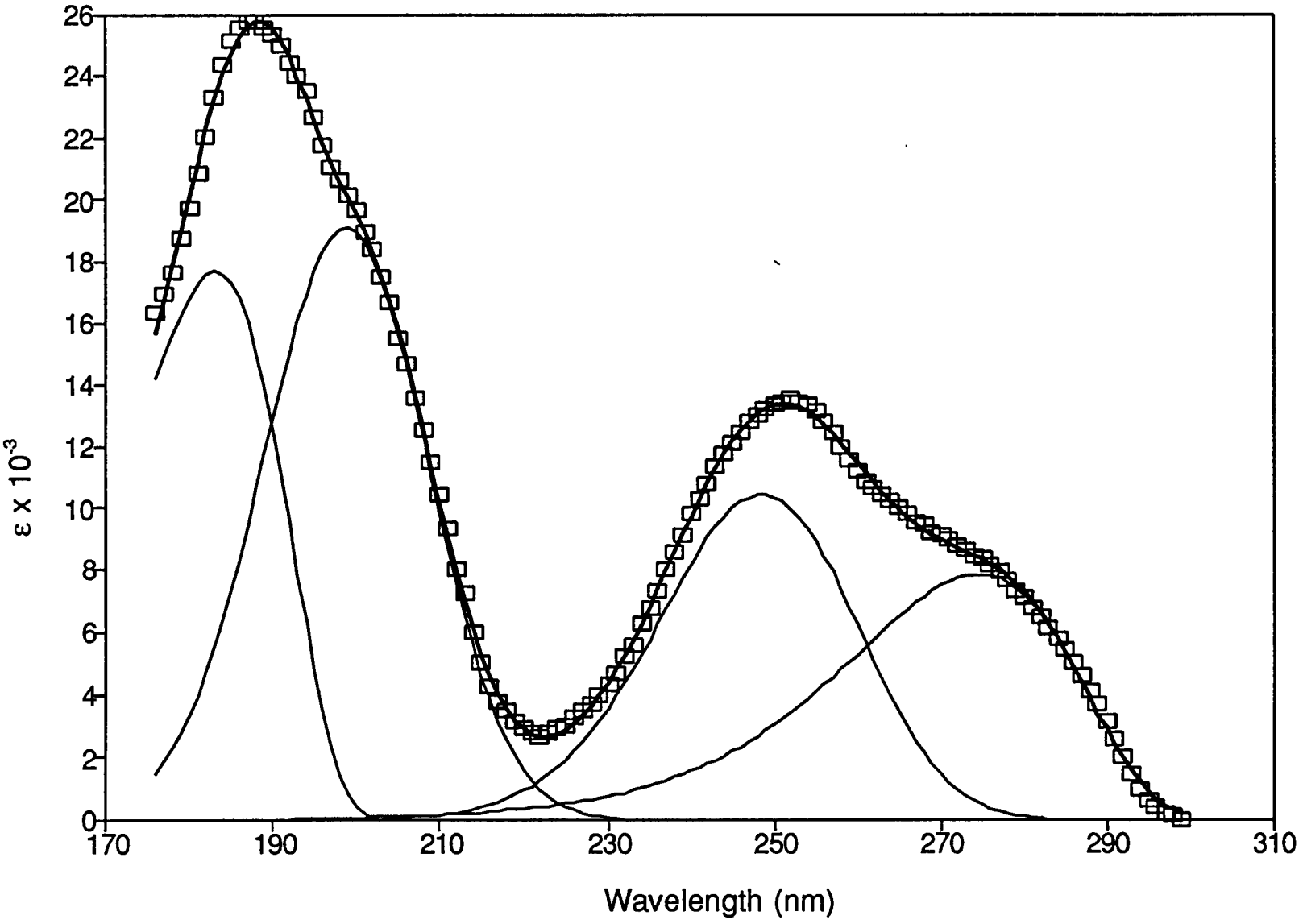
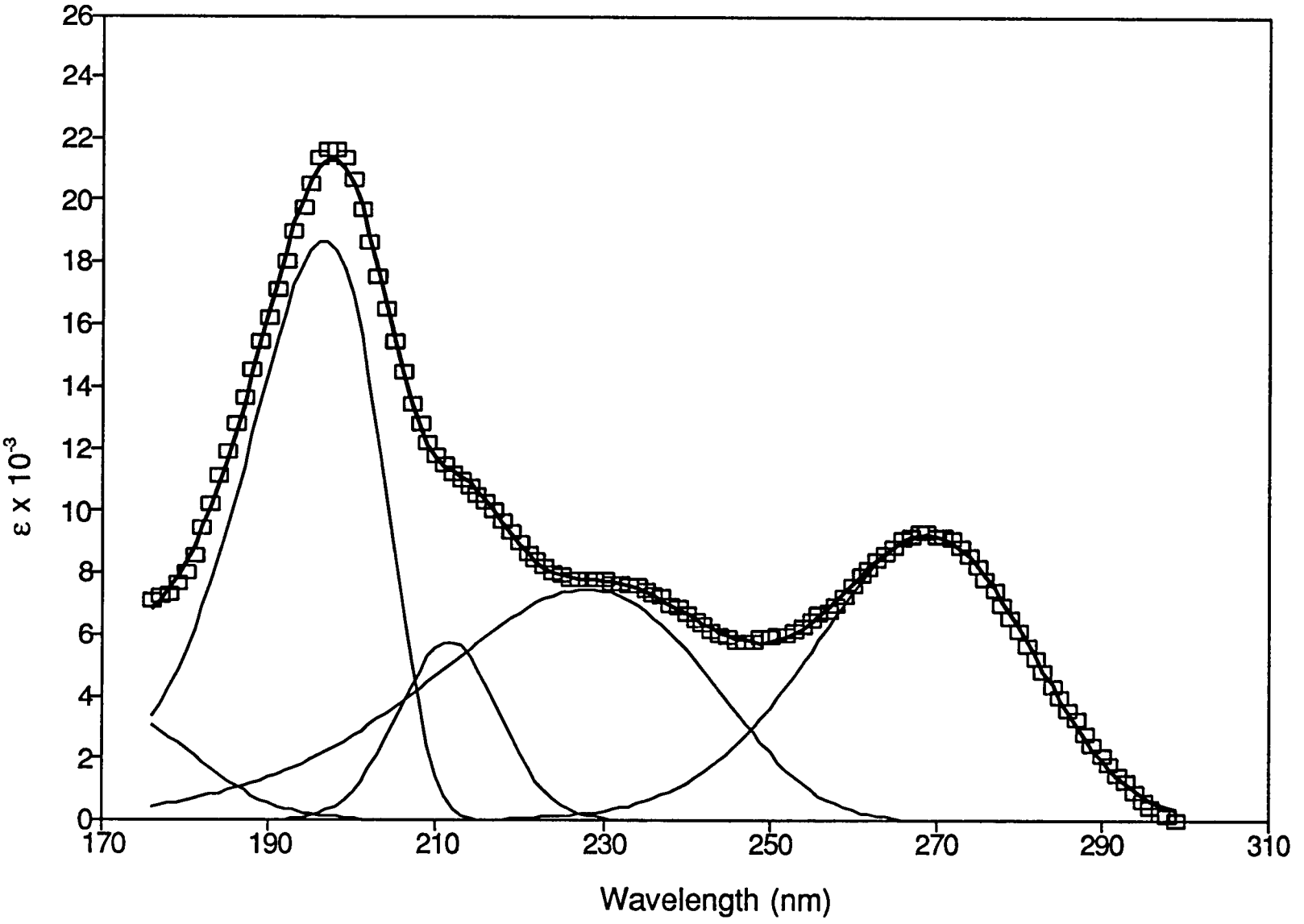

26
24
22
20
18
16
14
12
10
8
6
4
2
0
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Figure 5. Decomposition of dCMP absorption spectrum: (□□□) is measured, (▬▬) is fitted, and (——) is decomposed.


26
24
22
20
18
16
14
12
10
8
6
4
2
0
ε x 10⁻³
170
190
210
230
250
270
290
310
Wavelength (nm)

inclination on the CD. The three absorption bands for d(T) are taken from the decomposition of the TMP absorption spectrum (Figure 3), and the direction of these three transition dipoles are taken from Table I. In Figure 6, the LD 1 spectrum with sign reversed is plotted as a heavy solid line and its constituent LD bands are plotted as thin solid lines; the LD 2 spectrum with sign reversed is plotted as a heavy dotted line and its constituent LD bands are plotted as thin dotted lines.

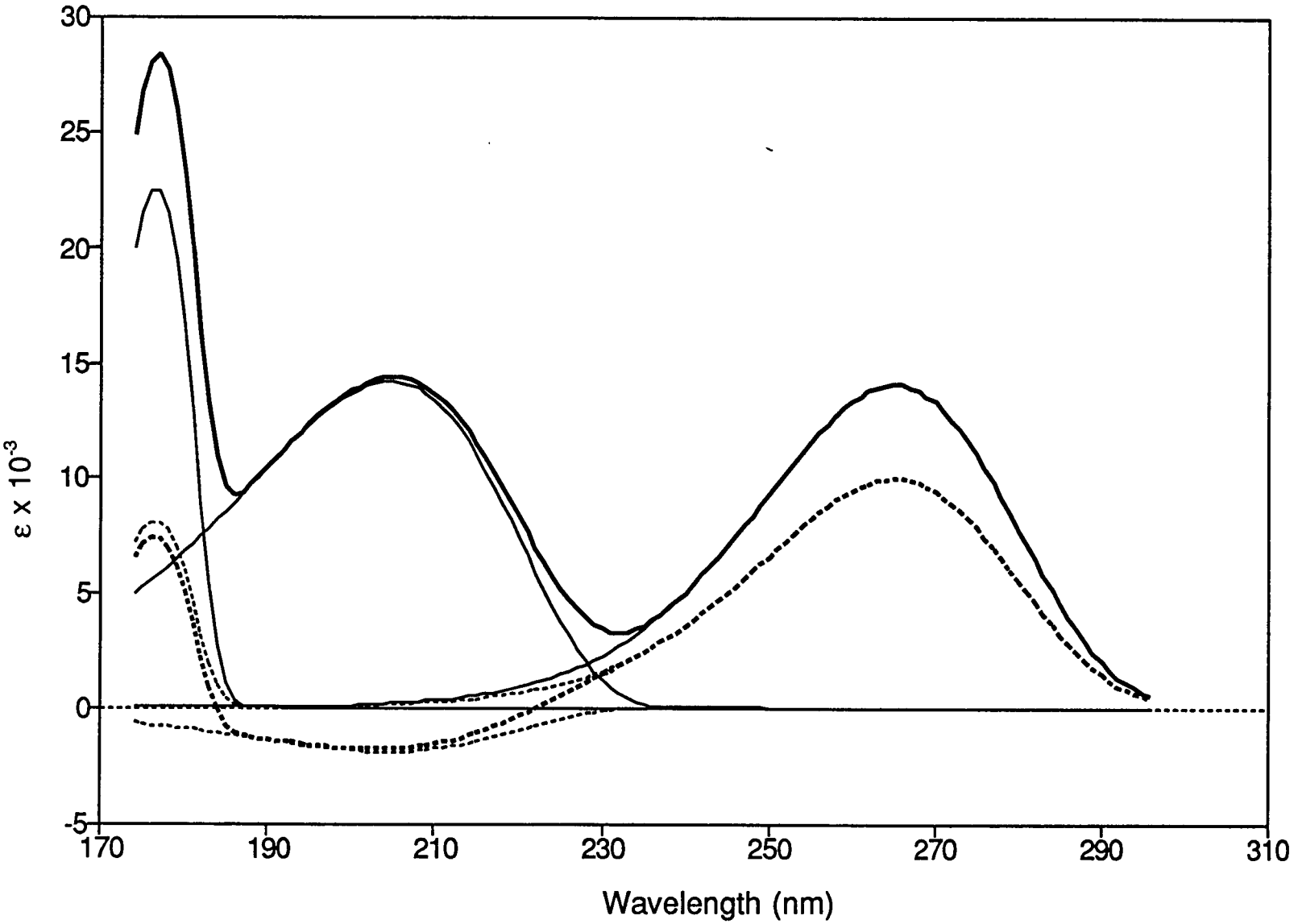
The overall shape (wavelength dependence) of the LD 1 spectrum is the same as that of the TMP absorption spectrum (Figure 3), and the intensity of the LD 1 spectrum is 1.5 times the intensity of the TMP absorption spectrum, as expected for an LD spectrum of a DNA polymer with bases perpendicular to the helix axis ($\alpha = 0°$). The LD 2 spectrum is significantly different from the LD 1 spectrum due to the different α angle, which also gives the χ angle relevance. As depicted in Eq. 8, when $\alpha = 0°$ (for LD 1), all of the three LD bands are obtained by multiplying their respective absorption bands to the same factor, -1.5, and the resulting LD spectrum is exactly -1.5 times the intensity of the absorption spectrum. On the other hand, when $\alpha \neq 0°$ (for LD 2), each LD band is obtained by multiplying its respective absorption band to a different factor, which depends on the χ angle of the d(T) and the δ angle associated with that band, and the resulting LD spectrum takes a unique shape.

Different α and χ angles (with $\alpha \neq 0°$) for DNA polymers give rise to different LD spectra. This is the basis for determining α and χ angles from LD spectra.

Decomposition of Synthetic Polymer Absorption and LD Spectra

Using the decomposition of the monomer absorption spectra as the initial guess (the ζ value is halved to approximately compensate for hyperchromism), a polymer absorption spectrum could be decomposed, and

Figure 6. LD spectra for a hypothetical single stranded poly[d(T)]: (▬▬) is LD 1 spectrum, (——) are LD bands of LD 1 spectrum; (•••••) is LD 2 spectrum, (·····) are LD bands of LD 2 spectrum. LD 1 and LD 2 spectra are calculated according to Eq. 8 with $\alpha = 0°$, $\chi = 20°$ and $\alpha = 40°$, $\chi = 20°$, respectively. The three absorption bands corresponding to the LD bands in LD 1 and LD 2 spectra are taken from the decomposition of the TMP absorption spectrum.


30
25
20
15
10
5
0
-5
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

the resulting μ, ζ, σ and ϱ parameters were used to decompose its LD spectrum with the α and χ's as the variables. The problem with this two-step procedure is that the fits to absorption and LD spectra are correlated. We also tried fitting both spectra at the same time with all of the variables, and the error in fitting the LD spectrum scaled to reflect the fact that the intensity of the LD spectrum is much smaller than that of the absorption spectrum. As the total sum of squares error for the fitting is minimized, the sum of squares error for the LD spectrum is nearly synchronous with that of absorption spectrum (Figure 7), and we can let the fitting proceed until the global minimum for sum of squares error is reached. The error in the fit will be less than the error in the measurements, and this unrealistic fitting results in some unrealistic band parameters. For example, a band position may move to 400 or 100 nm, a band intensity may become zero, or a band width may decrease to 0.1 nm. There is error in measurements, so we must stop the fitting before it is overdone. Thus, we choose to stop when the variables become stable, as described in the METHODS section. The advantage of this procedure is 2-fold: (1) the point along the minimization path is easily identified; and (2) the weight assigned to scale fitting errors of the LD spectrum has little effect on the fitting results, so we can weigh both absorption and LD spectra equally.

Figure 7 shows the standard deviation in α of the four bases for A-form DNA. We see that a stable solution with low standard deviation is achieved roughly when the log of the sum of squares error for the absorption (ABS.ssq) and LD (LD.ssq) is 6.0-6.4. The exactly choice does not affect the results significantly, and the method is stable. Further fitting leads to a larger standard deviation and instability, as Figure 7 shows. Note that ABS.ssq and LD.ssq are not perfectly correlated.

The results of decomposing the absorption and LD spectra for B-form poly[d(A)-d(T)] are shown in Figures 8a and 8b, and listed in Table II. For adenine, the first band of d(A) shifts toward longer wavelength by +10.5 nm

Figure 7. Sum of squares error for absorption (ABS.ssq) versus LD (LD.ssq), is correlated in this algorithm, □; LD.ssq decreases as ABS.ssq decreases, here for A-form DNA. To avoid overfitting the data, we look for a stable solution with small variances in the variables. Standard deviation for the inclination angle α of d(A) (▬ ▬), d(T) (——), d(G) (▬▬), and d(C) (• • • •) is minimized, as for the other variables, around LD.ssq of 6.0 to 6.4.

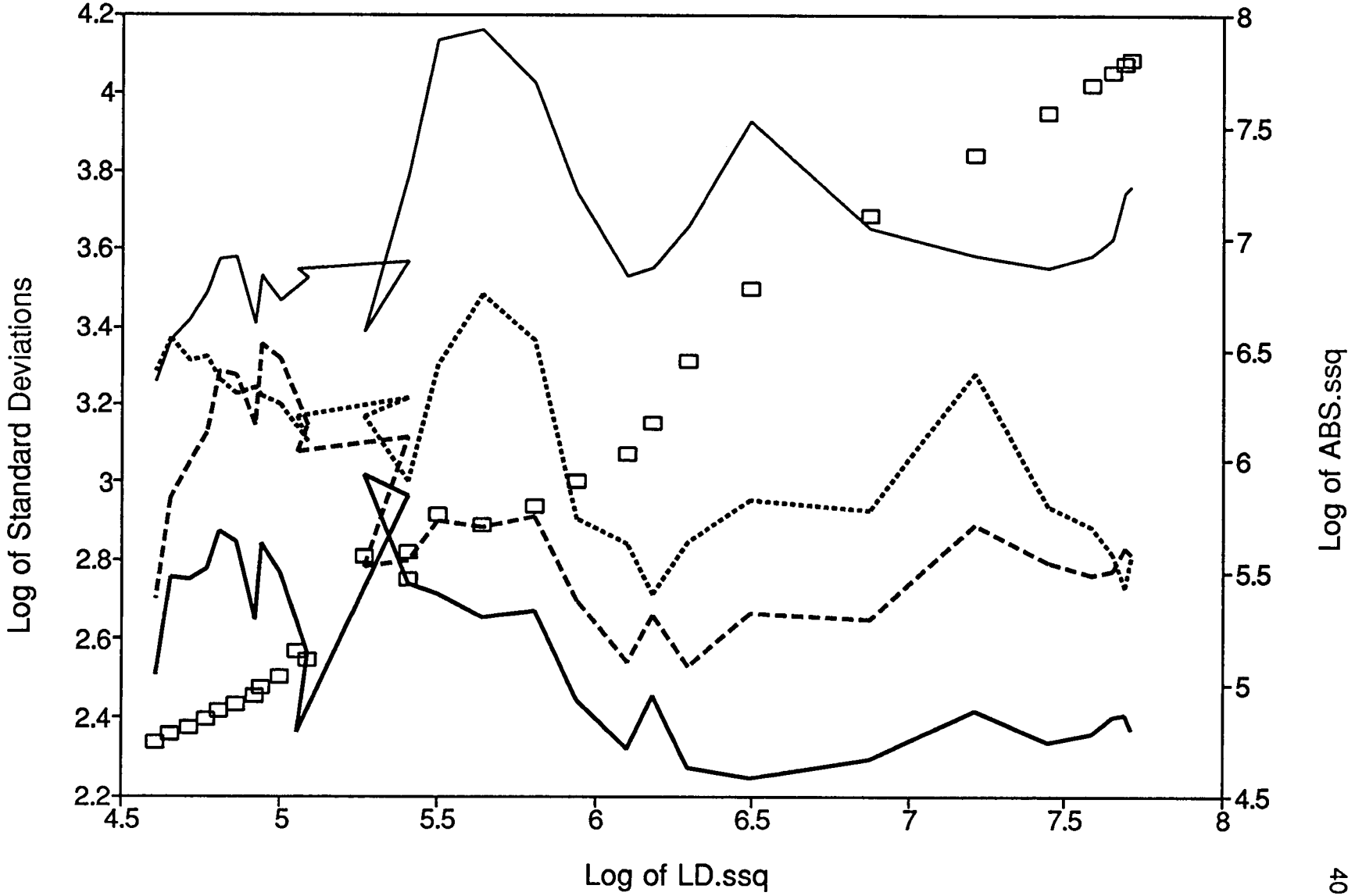

Log of Standard Deviations
Log of ABS.ssq
Log of LD.ssq
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
8
7.5
7
6.5
6
5.5
5
4.5
4.5
5
5.5
6
6.5
7
7.5
8

Figure 8a. Decomposition of poly[d(A)-d(T)] absorption spectrum: (□□□) is measured, (▬▬) is fitted, (——) is d(A) decomposed, and (·····) is d(T) decomposed.

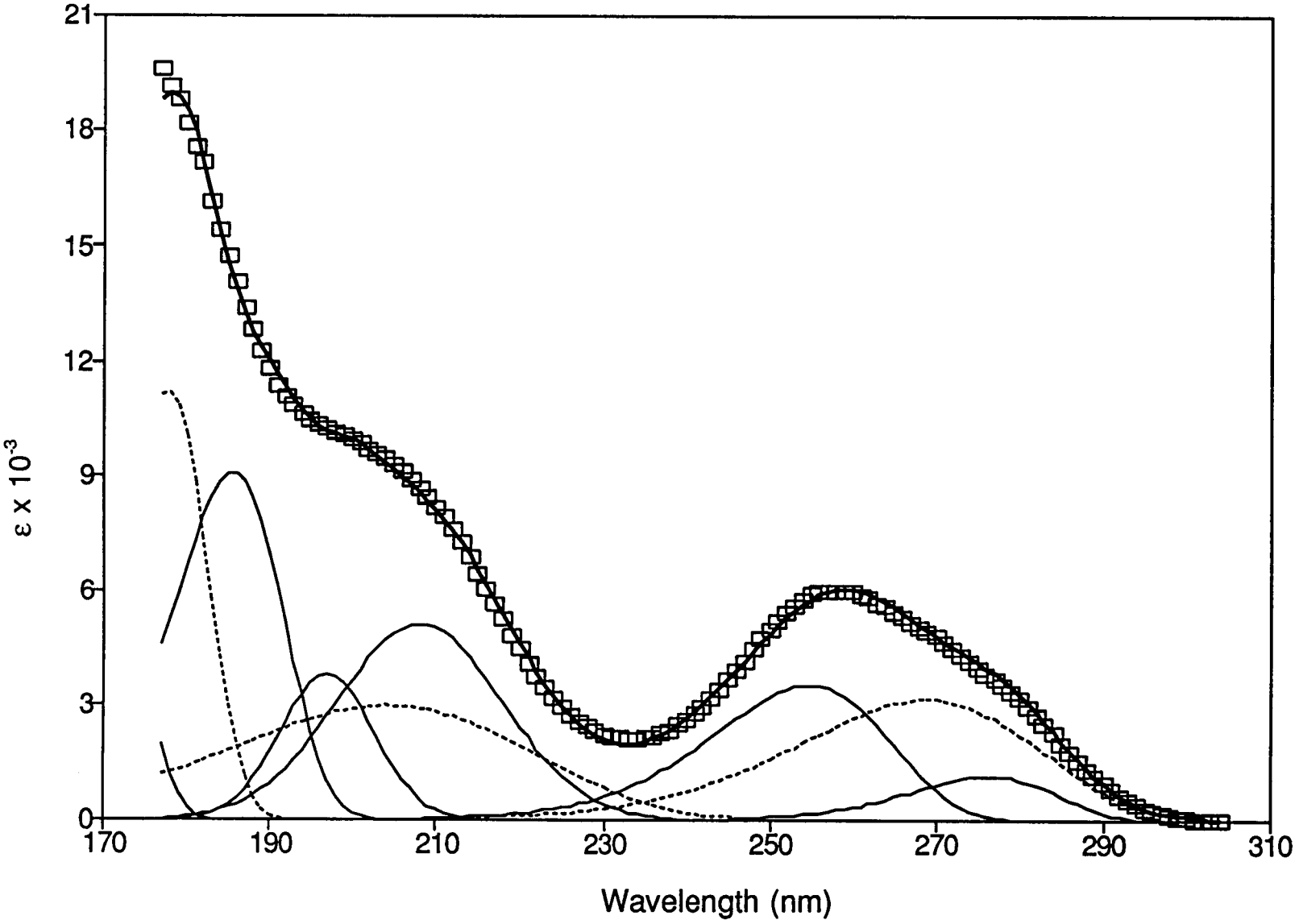

21
18
15
12
9
6
3
0
ε X 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Figure 8b. Decomposition of poly[d(A)-d(T)] normalized LD spectrum: (□□□) is measured, (▬▬) is fitted, (——) is d(A) decomposed, and (·····) is d(T) decomposed.

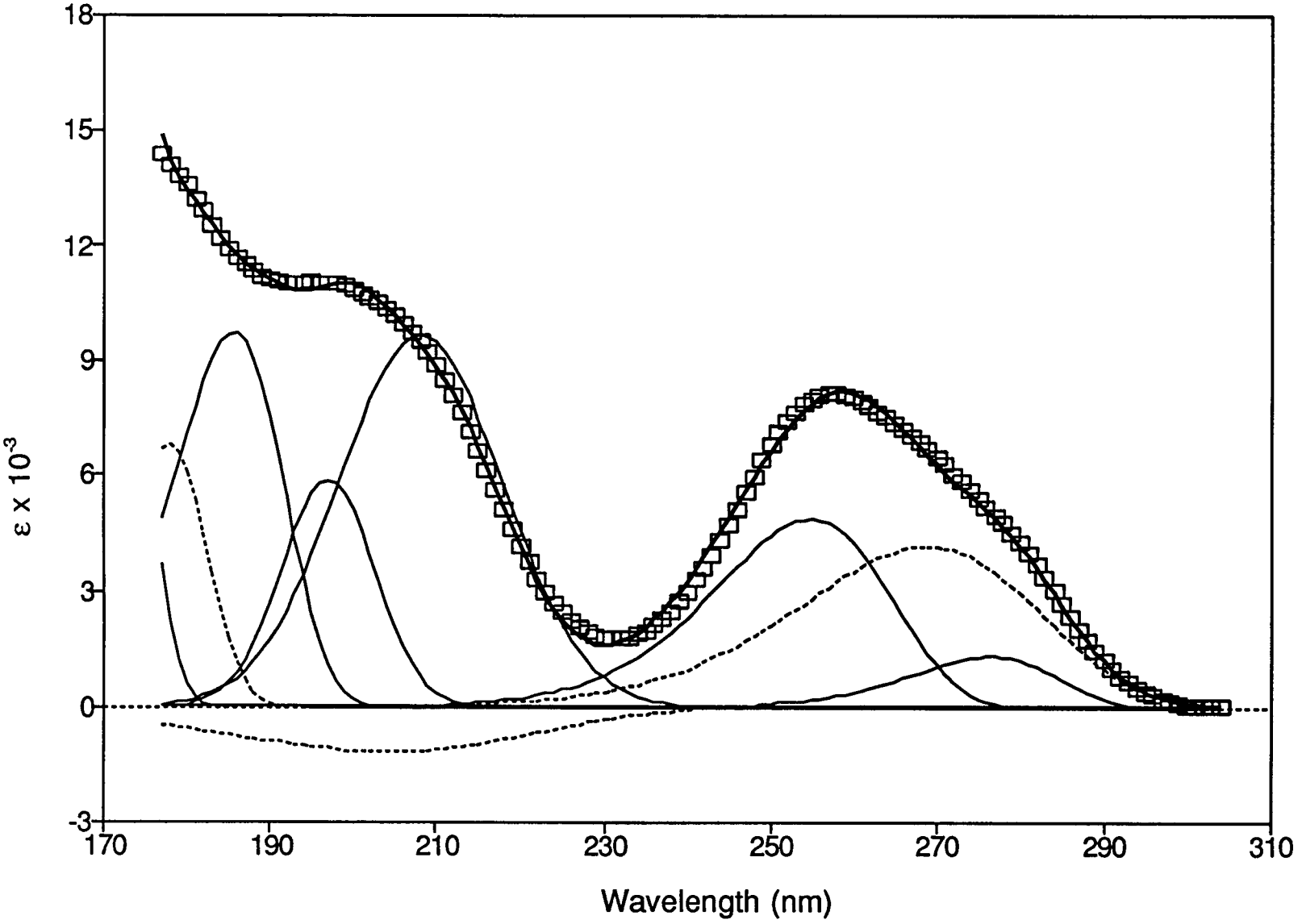

18
15
12
9
6
3
0
-3
ε x 10^-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Table II. Decomposition of Poly[d(A)-d(T)] Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(A)	276.9±1.0	26.1±4.6	10.1±0.5	1.20±0.11	23.2±0.8	-28.4±3.7
	255.2±0.8	94.8±2.2	12.4±0.3	1.23±0.09		
	207.9±0.2	124.7±5.5	11.7±0.3	1.01±0.00		
	196.9±0.2	52.8±2.8	6.7±0.1	1.02±0.01		
	185.4±0.2	148.2±4.9	7.9±0.2	1.24±0.02		
	172.5±0.2	40.5±6.2	4.1±0.3	1.10±0.06		
d(T)	268.0±2.3	113.6±2.5	17.9±1.4	1.32±0.14	42.1±2.5	21.1±3.2
	203.7±1.3	137.4±9.9	21.8±1.3	1.19±0.09		
	177.5±0.2	142.3±5.0	5.8±0.1	1.08±0.06		

μ: position of the band (wavelength of at the maximum height)
ζ: integrated intensity (area of the band)
σ: width at the half height of the band
ϱ: skewness

from that of dAMP, while the positions of all other bands remain about the same. The second band of d(T) is the only one in our studies having a positive LD band (remember that the resultant LD spectrum for a nucleic acid is negative everywhere). Numerically, the sign of an LD band depends on α and χ-δ angles, as can be seen from Eq. 8 in the METHODS section. The larger both angles are, the more likely that a transition will have a positive LD band. Since α and χ for the d(T) base from the best fit are 42.1° and 21.1° (Table II), and δ for the second band is -53° (Table I), the expression within brackets in Eq. 8 is positive. Detailed analyses on the relationship between LD and α, χ-δ angles can be found in RECENT DISCOVERIES section.

Table III lists the results for B-form poly[d(AT)-d(AT)] (decomposition not shown). Only the first band of d(A) and the second band of d(T) are significantly different from their counterparts in dAMP and TMP, respectively. If compared with results of poly[d(A)-d(T)] (Table II), we find that the third band of d(T) and all but the first band of d(A) are about the same for both polymers. The inclinations of d(A) and d(T) are somewhat smaller than those of poly[d(A)-d(T)].

Decompositions of B-form poly[d(G)-d(C)] (not shown) and poly[d(GC)-d(GC)] (Figures 9a and 9b) spectra gives very similar results in band parameters and α, χ angles (Table IV and V), but some band parameters deviate from those of dGMP and dCMP. The first band of d(G) shifts -5.2 nm with respect to that of dGMP, and the first four bands of d(C) shift -5.3, -6.7, -1.7 and -3.4 nm, respectively, from those of dCMP.

Table VI lists the results for decomposition (not shown) of the Z-form poly[d(GC)-d(GC)] spectra. Each band for both d(G) and d(C) resembles the corresponding one in B-form poly[d(G)-d(C)] and poly[d(GC)-d(GC)] (Table IV and V), except for a 3.9-nm difference in the position of the first band for d(G). Notice that we get almost the same results for d(C) for the three d(G)-d(C) polymers, including all of its bands and α, χ angles. This may indicate that

Figure 9a. Decomposition of poly[d(GC)-d(GC)] absorption spectrum: (□□□) is measured, (▬▬) is fitted, (——) is d(G) decomposed, and (·····) is d(C) decomposed.

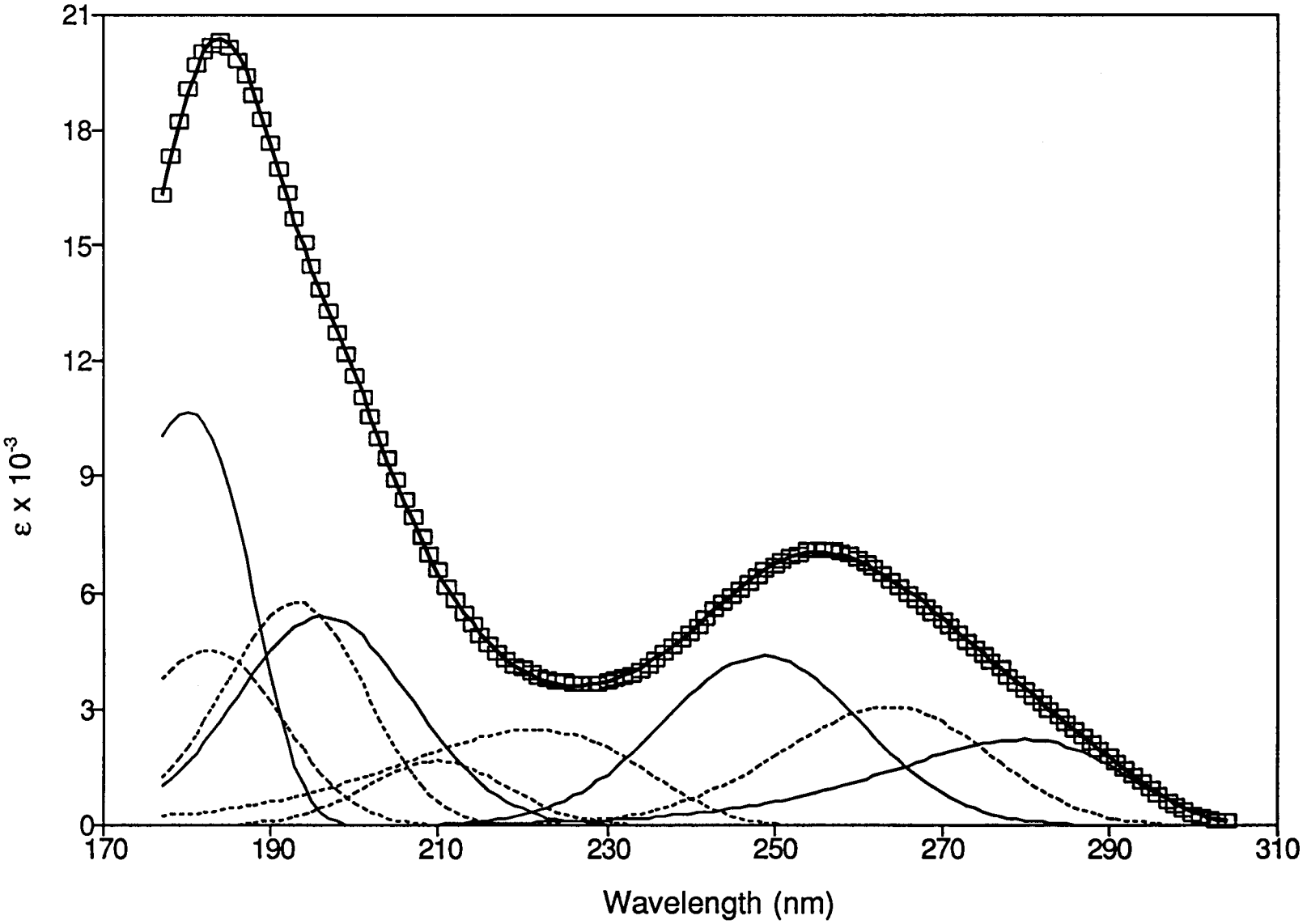

21
18
15
12
9
6
3
0
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Figure 9b. Decomposition of poly[d(GC)-d(GC)] normalized LD spectrum: (□□□) is measured, (▬▬) is fitted, (——) is d(G) decomposed, and (·····) is d(C) decomposed.

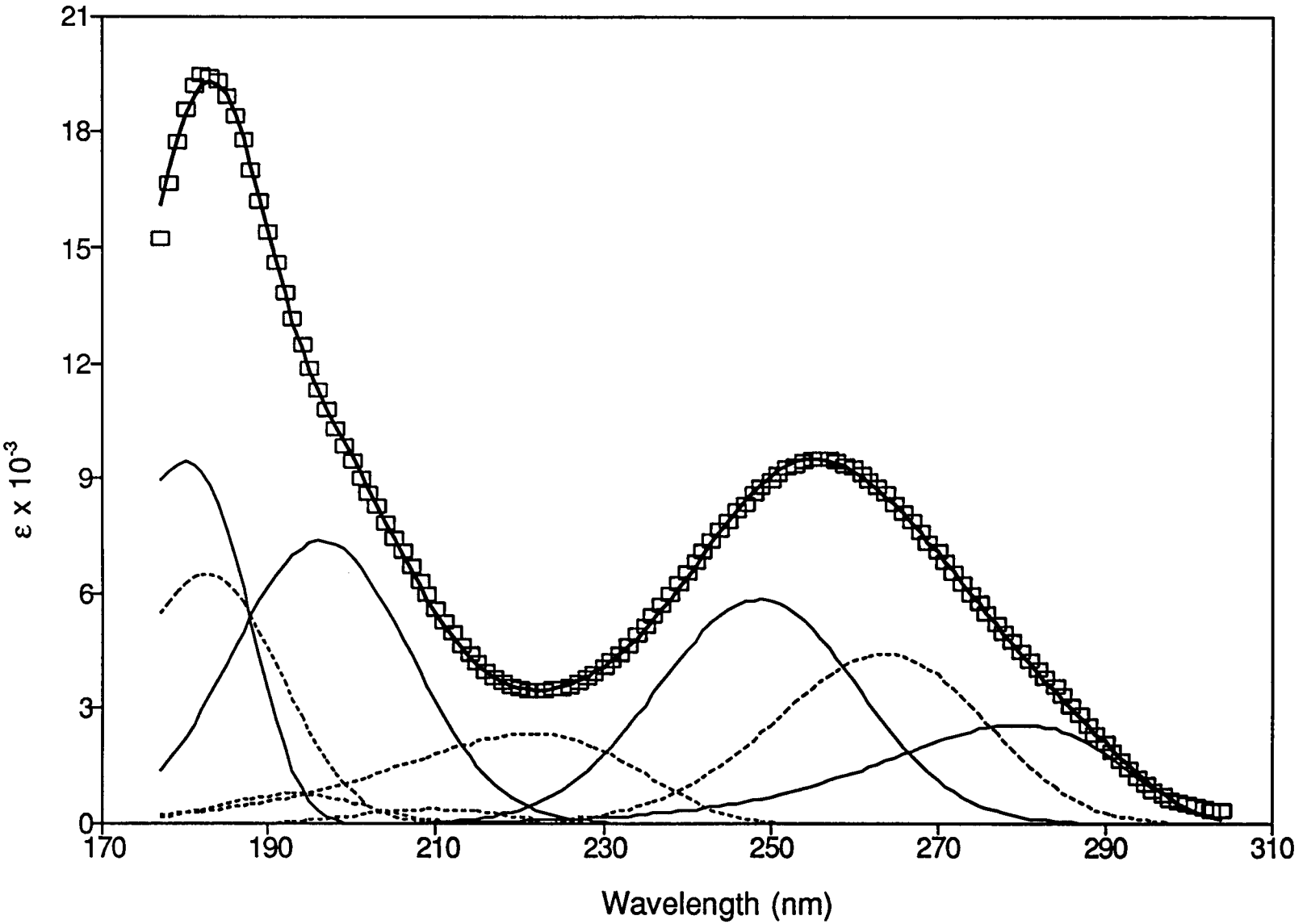

21
18
15
12
9
6
3
0
ε x 10-3
170
190
210
230
250
270
290
310
Wavelength (nm)

Table III. Decomposition of Poly[d(AT)-d(AT)] Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(A)	271.3±0.4	49.6±0.5	13.6±0.1	1.19±0.03	18.6±0.6	-16.1±3.4
	256.2±0.2	89.4±0.6	14.1±0.2	1.26±0.05		
	206.5±0.1	161.6±1.4	13.0±0.1	1.05±0.02		
	195.7±0.0	39.6±0.6	6.2±0.1	1.06±0.01		
	185.5±0.0	112.7±0.3	7.4±0.0	1.07±0.01		
	174.3±0.2	28.3±2.0	3.8±0.1	1.07±0.06		
d(T)	268.5±0.1	114.5±1.0	16.9±0.1	1.20±0.01	34.8±2.0	18.7±3.2
	203.7±0.7	159.6±3.8	23.9±0.4	1.41±0.02		
	177.2±0.0	78.7±0.9	5.5±0.1	1.13±0.00		

Table IV. Decomposition of B-form Poly[d(G)-d(C)] Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(G)	279.7±0.1	86.2±0.8	16.5±0.2	1.50±0.00	20.1±0.6	116.8±3.5
	248.3±0.1	135.7±0.6	13.5±0.1	1.00±0.00		
	196.4±0.2	145.7±1.0	12.0±0.2	1.01±0.01		
	179.8±0.0	234.4±1.5	10.1±0.1	1.39±0.01		
d(C)	263.6±0.1	98.4±0.4	15.1±0.2	1.15±0.01	33.8±1.0	189.8±3.8
	221.8±0.3	98.4±0.7	17.8±0.2	1.25±0.02		
	211.0±0.2	40.7±1.7	9.2±0.2	1.02±0.01		
	193.4±0.1	124.7±1.4	10.1±0.2	1.01±0.00		
	182.6±0.2	105.5±1.0	11.2±0.1	1.00±0.00		

Table V. Decomposition of B-form Poly[d(GC)-d(GC)] Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(G)	279.7±0.1	81.4±0.7	16.8±0.1	1.46±0.01	21.4±0.5	130.7±2.8
	248.7±0.1	130.1±0.4	14.1±0.1	1.02±0.01		
	196.3±0.2	140.7±1.1	12.4±0.1	1.01±0.01		
	180.0±0.0	232.7±0.7	10.1±0.0	1.35±0.01		
d(C)	263.7±0.1	96.1±0.4	14.7±0.1	1.10±0.01	34.0±0.7	184.0±3.2
	221.4±0.2	92.9±1.3	17.6±0.3	1.39±0.01		
	209.9±0.5	35.1±1.6	9.9±0.3	1.04±0.03		
	193.1±0.1	124.1±1.5	10.2±0.1	1.10±0.02		
	182.7±0.1	103.1±1.2	10.7±0.1	1.09±0.02		

Table VI. Decomposition of Z-form Poly[d(GC)-d(GC)] Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(G)	283.6±0.1	96.0±1.3	16.2±0.2	1.50±0.00	27.1±1.1	137.6±3.6
	249.3±0.2	126.8±1.9	14.7±0.2	1.01±0.01		
	197.9±0.6	143.3±3.0	12.6±0.2	1.04±0.04		
	177.3±0.5	222.1±1.8	11.4±0.2	1.05±0.02		
d(C)	265.4±0.3	97.1±1.3	15.4±0.4	1.04±0.02	32.1±1.7	201.5±2.8
	217.9±0.4	93.6±2.6	15.9±0.3	1.19±0.06		
	206.4±0.4	32.7±1.8	6.4±0.3	1.03±0.02		
	193.4±0.3	115.8±2.1	9.6±0.4	1.39±0.07		
	184.5±0.2	94.8±3.4	7.2±0.2	1.02±0.02		

cytosine is less sensitive to its environment or that it sees a similar surroundings in the three polymers.

Decomposition of DNA Absorption and LD Spectra

Natural DNA is typically studied in three different forms in solution. In aqueous solution with moderate salt (here 0.01 M Na^+ as phosphate buffer, pH 7) DNA exhibits a well-known conservative circular dichroism (CD) spectrum with a maximum at 275 nm and a minimum at 248 nm.[50] This B-form DNA has 10.4 bp/turn,[51,52] and we denote it as 10.4B-DNA. At high concentration of salt (here 5.5 M NH_4F), in 95% methanol, or when wrapped around histone cores, the 275-nm band of B-form DNA collapses, and this form has 10.2 bp/turn.[53] We denote this form as 10.2B-DNA. In 80% ethanol,[5,6] or here 80% 2,2,2-trifluoroethanol,[24] DNA has the nonconservative CD typical of A-form. The LD has been measured for all three forms,[31] and we analyze these LD spectra here for the first time.

Decomposition of DNA absorption and LD spectra presents another set of problems. First, the computer time required for each iteration is more than four times longer than that in fitting two-base polymers. Second, the step size between two iterations must be small enough so that the LM algorithm can find the path leading to the point of minimum variances and stay there through several iterations. Third, as the step size gets smaller, round-off errors become more significant in computing the Jacobian matrix and matrix inversion, resulting in meaningless variances.

We overcome these problems to obtain the results listed in Tables VII-IX for decomposition of absorption and LD spectra (not shown). Differences among the three DNAs for each band are generally small. Inclinations for the B forms are about 15° for the purine and 26° for the pyrimidines. As expected d(A), d(T) and d(C) have larger inclinations in A form, but our results indicate that d(G) is unchanged from the B form.

Table VII. Decomposition of 10.4B-DNA Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(A)	271.4±0.0	24.6±0.1	14.9±0.1	1.21±0.00	16.1±0.5	46.5±4.7
	255.2±0.1	43.4±0.2	13.3±0.0	1.27±0.01		
	206.4±0.0	73.2±0.1	12.8±0.0	1.05±0.00		
	195.5±0.0	17.5±0.2	6.4±0.0	1.01±0.00		
	185.5±0.0	50.8±0.1	7.4±0.0	1.09±0.01		
	174.4±0.0	13.9±0.1	3.5±0.0	1.13±0.00		
d(T)	268.5±0.0	54.7±0.2	17.4±0.1	1.19±0.01	25.0±0.9	1.8±3.3
	204.2±0.1	70.6±0.1	23.9±0.1	1.41±0.00		
	177.1±0.0	35.1±0.1	5.6±0.0	1.15±0.00		
d(G)	279.5±0.1	40.5±0.1	15.9±0.0	1.50±0.00	18.0±0.6	114.8±8.6
	248.9±0.0	61.7±0.2	13.2±0.0	1.04±0.01		
	196.3±0.1	63.9±0.2	12.4±0.1	1.01±0.01		
	180.0±0.0	105.2±0.1	10.2±0.0	1.35±0.01		
d(C)	263.2±0.0	45.8±0.1	15.2±0.0	1.05±0.00	25.1±0.8	215.8±3.0
	222.0±0.1	42.2±0.3	18.1±0.1	1.20±0.04		
	209.7±0.1	15.4±0.1	9.8±0.1	1.01±0.01		
	193.2±0.0	55.6±0.2	10.2±0.0	1.08±0.00		
	182.5±0.0	46.6±0.1	10.9±0.0	1.11±0.00		

Table VIII. Decomposition of 10.2B-DNA Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(A)	271.7±0.1	23.9±0.2	14.4±0.1	1.23±0.01	14.9±0.6	96.6±3.7
	255.1±0.0	42.6±0.1	13.2±0.1	1.18±0.01		
	206.7±0.1	68.8±0.2	13.1±0.1	1.01±0.01		
	194.9±0.0	16.4±0.1	6.7±0.1	1.01±0.00		
	185.4±0.1	45.8±0.2	8.1±0.1	1.01±0.01		
	174.1±0.1	11.7±0.5	3.6±0.1	1.15±0.04		
d(T)	268.3±0.2	53.6±0.4	17.8±0.2	1.09±0.01	28.1±1.3	31.9±3.0
	204.7±0.4	66.1±0.5	23.3±0.2	1.41±0.03		
	177.1±0.0	34.5±0.6	5.2±0.1	1.12±0.01		
d(G)	279.5±0.1	38.9±0.2	16.0±0.1	1.50±0.00	13.9±1.7	142.5±4.2
	248.8±0.1	58.6±0.2	13.4±0.1	1.08±0.01		
	196.5±0.1	59.5±0.1	12.2±0.1	1.01±0.01		
	179.7±0.0	96.1±0.2	10.5±0.0	1.27±0.00		
d(C)	263.5±0.1	43.7±0.2	15.4±0.1	1.07±0.01	27.7±0.7	201.2±2.5
	221.5±0.1	40.4±0.1	17.3±0.1	1.18±0.02		
	210.2±0.2	14.9±0.1	11.9±0.3	1.09±0.03		
	193.9±0.1	51.2±0.3	10.6±0.1	1.01±0.01		
	182.2±0.1	42.1±0.3	11.5±0.1	1.02±0.01		

Table IX. Decomposition of A-form DNA Absorption and LD Spectra

Base	μ (nm)	$\zeta \times 10^{-3}$	σ (nm)	ρ	α (deg)	χ (deg)
d(A)	270.3±0.2	25.8±0.7	16.9±0.7	1.13±0.01	27.8±1.0	7.0±1.3
	256.1±0.1	44.7±0.4	14.1±0.1	1.34±0.01		
	206.9±0.1	76.1±0.2	12.5±0.1	1.00±0.00		
	196.0±0.1	18.4±0.4	6.3±0.1	1.01±0.01		
	185.4±0.0	50.0±0.3	7.4±0.0	1.13±0.01		
	174.0±0.0	11.3±0.2	3.5±0.0	1.18±0.01		
d(T)	268.6±0.1	56.8±0.6	17.0±0.1	1.16±0.01	34.7±0.9	-5.4±1.4
	206.6±0.1	75.9±0.3	23.2±0.1	1.21±0.01		
	176.9±0.0	34.3±0.3	5.5±0.0	1.23±0.01		
d(G)	280.0±0.1	42.3±0.4	16.3±0.1	1.50±0.00	14.3±1.0	95.3±6.7
	248.9±0.1	61.7±0.4	14.6±0.1	1.06±0.01		
	196.8±0.1	65.2±0.5	12.0±0.1	1.04±0.02		
	179.9±0.1	103.7±0.4	10.3±0.0	1.36±0.01		
d(C)	263.4±0.1	47.7±0.2	14.7±0.0	1.06±0.01	35.2±0.5	216.1±1.4
	219.9±0.4	49.6±0.6	14.7±0.3	1.12±0.05		
	209.9±0.2	18.6±0.2	8.4±0.1	1.01±0.01		
	193.5±0.1	54.6±0.5	11.2±0.2	1.01±0.01		
	182.4±0.1	45.1±0.2	11.3±0.1	1.07±0.01		

Comparison to Previous Results

Angle α for the five synthetic polymers are similar to those calculated previously in our laboratory from the same data.[32,33] Differences in the inclination axis, χ, are not surprising as this parameter is not particularly sensitive to the data (see RECENT DISCOVERIES section). The new inclination angles, α, result in the same message: base pairs are inclined, even in the B form.

One factor that is responsible for any differences between this analysis and previous analyses is the different optimization algorithm. Although the advantage of simplex method used previously is that one can tell local ssq (sum of squares error) minima from the global minimum "by running the program several times",[32] the scale of the problem actually turns the advantage into a disadvantage, because the "several times" could mean an infinite number of times to assure the global minimum of ssq is found. Two other disadvantages in using the simplex algorithm are that (1) there is no correlation term defined for any two variables, and (2) the algorithm uses only ssq, and not individual squared errors. The LM algorithm used in this study has none of these drawbacks.

Furthermore, transition dipole directions are different, especially for the base adenine, which also has a different number of transitions. A skewness parameter is added to define the shape of an absorption band. Previous calculations aimed to fit one reduced LD spectrum (Eq. 1 in INTRODUCTION section), while this work fits absorption and LD spectra simultaneously. Previously, the position of a band for a given base is fixed for all polymers containing that base, but the position is variable now. Because every change we made departing from the previous study is an improvement, the current results should be more reliable and stable.

Repeated Fittings with Randomized Transition Dipole Directions

Two types of error can affect the results of spectral decomposition: the error in measuring absorption and LD spectra, and the error in determining transition dipole directions. The effects of the first type of error were minimized by using stability of the band parameters as the criterion for fitting the data. The effects of the second type of error can be studied through the Monte Carlo method. The direction of a transition dipole in a given base may not necessarily be the same for the monomer and a polymer containing the base. Thus, we repeated each of the fittings 100 times with each transition dipole direction randomly perturbed by a value sampled from an uniform distribution in the range ±10°. Averages and standard deviations are calculated from the 100 independently fitted results for each variable, and these are the results presented in Tables II-IX. The α and χ angles show no dependence on either ABS.ssq or LD.ssq, indicating that they are very stable around our chosen solution and are fairly insensitive to the ±10° variation in the transition dipole directions used to obtain these α and χ angles. Figures 10a and 10b illustrated the distributions of α and χ angles, respectively, of the four different bases resulting from 100 repeated fittings for 10.4B-DNA absorption and LD spectra.

In Tables II-IX, relatively large standard deviations for band parameters often occur at bands sitting near the ends of a spectrum, and for the second band of d(T). Standard deviations of χ angles in each table are always greater than those of α angles, especially for the three DNAs. The difference in stability between fitted α and χ angles was also observed in earlier studies, but its physical meaning was not clear until very recently. The explanation for this phenomenon is presented in the section RECENT DISCOVERIES.

Figure 10a. Distributions of inclination angles α for d(A) (□), d(T) (▌), d(G) (*) and d(C) (▬) from 100 repeated fittings of 10.4B-DNA absorption and LD spectra. In each repeated fitting, all transition dipole directions are each perturbed with a random number drawn from an uniform distribution in the range of ±10°.

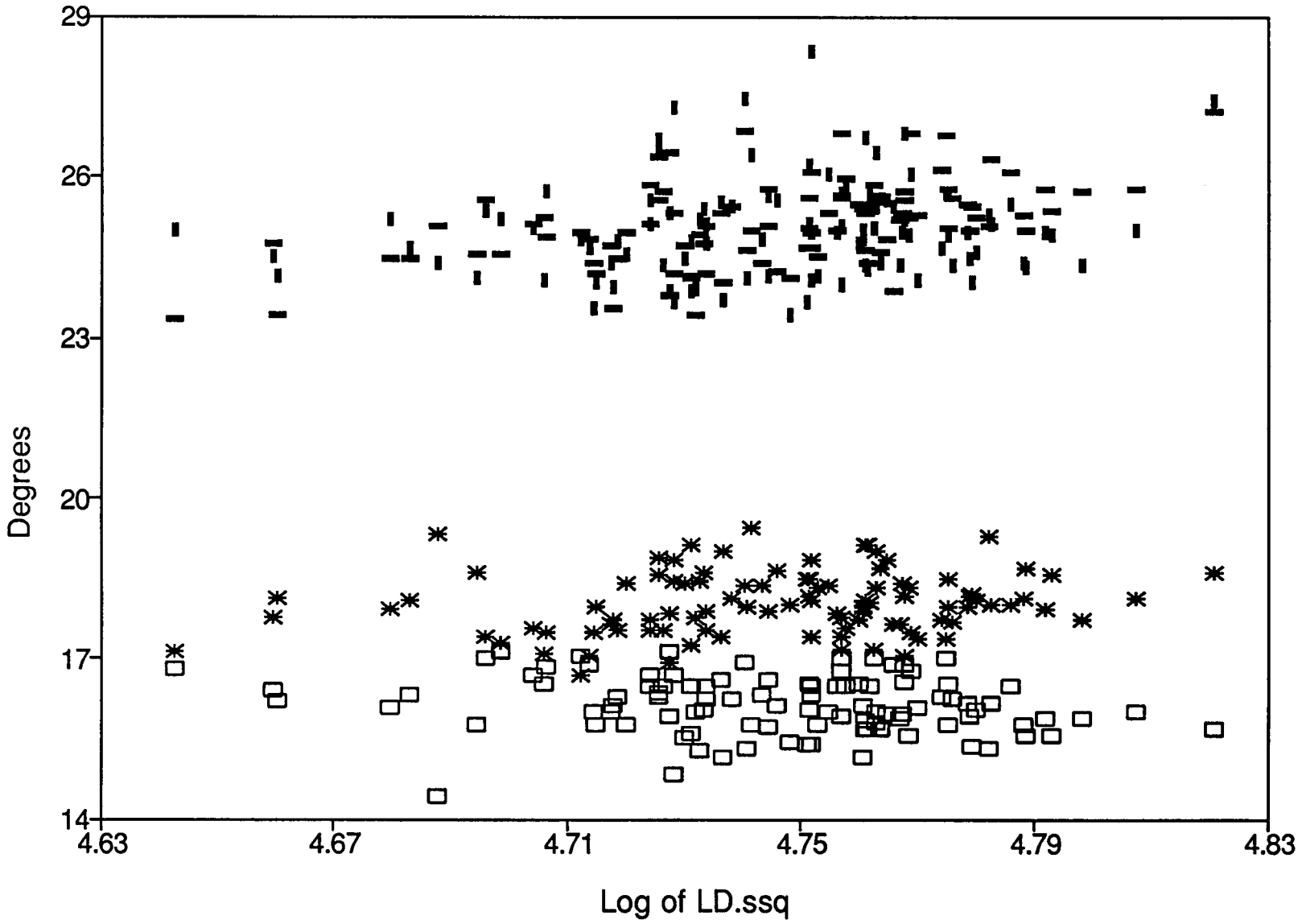
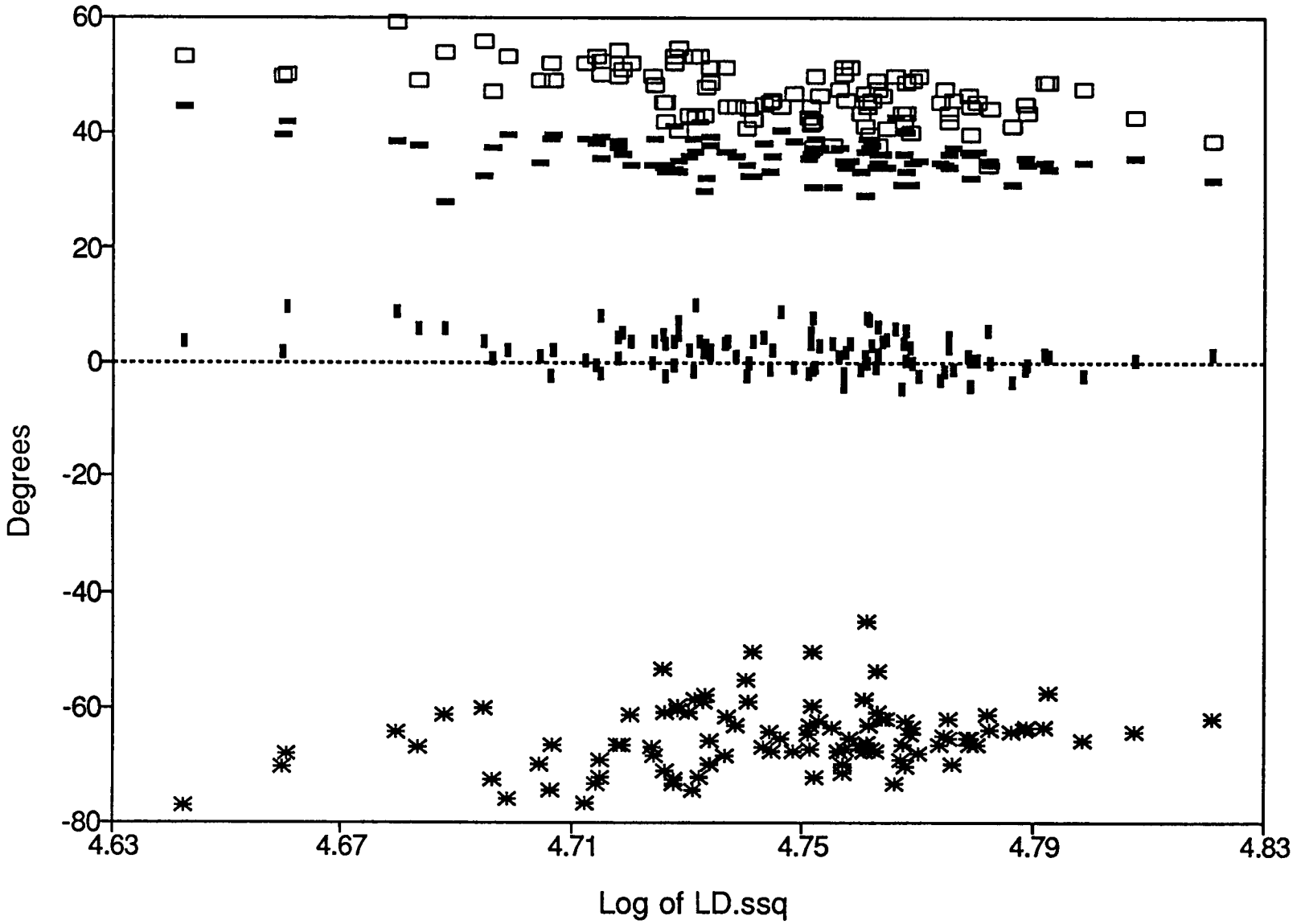

29
26
23
20
17
14
Degrees
4.63
4.67
4.71
4.75
4.79
4.83
Log of LD.ssq

Figure 10b. Distributions of inclination axes χ for d(A) (□), d(T) (▌), d(G) (*) and d(C) (—) from 100 repeated fittings of 10.4B-DNA absorption and LD spectra. In each repeated fitting, all transition dipole directions are each perturbed with a random number drawn from an uniform distribution in the range of ±10°.


60
40
20
0
-20
-40
-60
-80
Degrees
4.63
4.67
4.71
4.75
4.79
4.83
Log of LD.ssq

Building a Base-Pair Model

Table X lists parameters for the four base pairs built from α and χ angles determined for d(A) and d(T) in each of the two synthetic polymers and three DNAs. When building a base pair, positions and orientations of the two paired bases are adjusted so that the two hydrogen-bond lengths are between 2.80 and 3.00 Å, and hydrogen-bond angles ($A.C_6$-$A.N_6$-$T.O_4$ and $T.C_4$-$T.N_3$-$A.N_1$) are close to 120°. For each resulting base pair the propeller twist, the distance between the two C_1' atoms and the distance between $A.C_8$ and $T.C_6$ are determined. Because the imposed restrictions are not very tight in this procedure, for each base pair we can actually derive a number of possible conformations, each with slightly different values of the parameters. Thus the results presented in Table X are not unique, nor necessary the best, but simply possible.

One interesting feature regarding the uncertainty in the sign of α angles is that only poly[d(A)-d(T)] has a propeller twist in +/- or -/+ smaller than that in +/+ or -/-. It is independent of how the base pair is built, because of the large α angle for d(T). Another feature is that all four possible propeller twist angles for 10.4B-DNA are about the same as those of 10.2B-DNA, although the χ angles of d(A) and d(T) for 10.4B-DNA are significantly different from those for 10.2B-DNA. Equally strikingly is the fact that all parameters for the +/+ and -/- variations of 10.4B-DNA are virtually the same as their counterparts in 10.2B-DNA. From the two hydrogen-bond angles, the +/+ and -/- pairs in both B-form DNAs are considered more acceptable than the +/- and -/+ pairs. The +/+ and -/- pairs of A-form DNA are also our choices, because their $A.C_1'$-$T.C_1'$ and $A.C_8$-$T.C_6$ distances are more realistic than those of the +/- and -/+ pairs. In general, conformations with a smaller propeller twist for the AT pair have more favorable base-pair parameters, and these are found in the +/+ and -/- pairs for all but poly[d(A)-d(T)].

The GC base pairs were built for the three d(G)-d(C) polymers and three DNAs, and the parameters are listed in Table XI.a and XI.b. Among the four

Table X. A/T Base-Pair Parameters

	Sign of α for A/T pair	Hydrogen Bond				Propeller twist (deg)	$A.C_1'$ --- $T.C_1'$ (Å)	$A.C_8$ --- $T.C_6$ (Å)
		$A.N_6$---$T.O_4$		$A.N_1$---$T.N_3$				
		Length (Å)	Angle (deg)	Length (Å)	Angle (deg)			
poly [d(A) -d(T)]	+/+	2.80	102	3.00	94	53	10.84	9.67
	+/-	2.80	112	3.00	123	40	10.18	8.99
	-/+	2.80	110	2.91	121	40	10.44	9.22
	-/-	2.80	105	3.00	95	53	10.73	9.47
poly [d(AT) -d(AT)]	+/+	2.81	110	3.00	103	40	10.82	9.75
	+/-	2.80	114	3.00	120	38	10.38	9.26
	-/+	2.84	111	2.90	121	38	10.43	9.30
	-/-	2.80	113	3.00	103	40	10.74	9.81
10.4B-DNA	+/+	2.92	123	3.00	121	10	10.82	9.88
	+/-	3.00	109	2.95	121	41	10.29	9.37
	-/+	3.00	108	2.91	119	41	10.44	9.46
	-/-	2.87	124	3.00	121	10	10.78	9.86
10.2B-DNA	+/+	2.94	122	3.00	122	13	10.84	9.86
	+/-	3.00	107	2.81	115	43	10.71	9.72
	-/+	3.00	106	2.92	110	43	10.91	9.76
	-/-	2.85	123	3.00	120	13	10.80	9.84
A-form DNA	+/+	3.00	114	2.80	120	25	10.57	9.74
	+/-	2.80	109	3.00	123	58	9.27	8.39
	-/+	2.80	107	2.97	123	57	9.66	8.61
	-/-	3.00	115	2.80	121	25	10.48	9.76

Table XI.a. G/C Base-Pair Parameters

	Sign of α for G/C pair	Hydrogen Bond					
		$G.O_6$---$C.N_4$		$G.N_1$---$C.N_3$		$G.N_2$---$C.O_2$	
		Length (Å)	Angle (deg)	Length (Å)	Angle (deg)	Length (Å)	Angle (deg)
B-form poly [d(G) -d(C)]	+/+	3.00	112	2.80	113	3.37	109
	+/-	2.80	122	3.00	116	2.83	126
	-/+	2.80	118	3.00	114	3.13	124
	-/-	3.00	114	2.80	115	3.10	113
B-form poly [d(GC) -d(GC)]	+/+	3.00	111	2.80	112	3.30	108
	+/-	2.80	119	3.00	115	2.80	124
	-/+	2.80	113	2.81	115	2.80	124
	-/-	3.00	113	2.80	115	3.03	111
Z-form poly [d(GC) -d(GC)]	+/+	3.00	106	2.80	108	3.70	98
	+/-	2.80	120	2.98	113	2.80	123
	-/+	2.80	115	2.82	114	2.80	123
	-/-	3.00	107	2.80	110	3.54	102
10.4B-DNA	+/+	3.00	108	2.80	110	3.31	106
	+/-	2.80	119	2.93	113	2.80	123
	-/+	2.80	118	2.92	112	2.91	121
	-/-	3.00	108	2.80	111	3.17	110
10.2B-DNA	+/+	3.00	112	2.80	113	3.07	110
	+/-	2.80	121	2.95	114	2.80	124
	-/+	2.80	117	2.82	114	2.80	123
	-/-	2.91	114	2.80	114	3.00	113
A-form DNA	+/+	3.00	112	2.80	115	3.48	109
	+/-	2.80	123	2.84	119	2.80	125
	-/+	2.80	122	2.83	118	2.91	122
	-/-	3.00	111	2.80	115	3.32	113

Table XI.b. G/C Base-Pair Parameters

	Sign of α for G/C pair	Propeller twist (deg)	$G.C_1'$ --- $C.C_1'$ (Å)	$G.C_8$ --- $C.C_6$ (Å)
B-form poly [d(G) -d(C)]	+/+	49	10.55	9.70
	+/-	27	10.61	9.80
	-/+	25	10.74	9.81
	-/-	48	10.42	9.71
B-form poly [d(GC) -d(GC)]	+/+	45	10.44	9.70
	+/-	34	10.48	9.67
	-/+	34	10.28	9.53
	-/-	44	10.29	9.70
Z-form poly [d(GC) -d(GC)]	+/+	52	10.71	9.74
	+/-	31	10.53	9.73
	-/+	31	10.35	9.61
	-/-	52	10.63	9.75
10.4B-DNA	+/+	43	10.71	9.77
	+/-	11	10.67	9.89
	-/+	10	10.71	9.89
	-/-	43	10.66	9.77
10.2B-DNA	+/+	35	10.47	9.80
	+/-	26	10.61	9.82
	-/+	26	10.46	9.72
	-/-	36	10.48	9.79
A-form DNA	+/+	49	10.84	9.76
	+/-	21	10.58	9.86
	-/+	21	10.62	9.86
	-/-	49	10.80	9.76

sign possibilities for each base pair, there are always two (+/+ and -/-) that have the three hydrogen-bond lengths beyond the 2.8-3.0 Å range, primarily due to their large propeller twists. Since the amino group of d(G) rotates within the closed base pair,[54] we allowed the hydrogen-bond length of $G.N_2$-$C.O_2$ to be a larger value and adjusted the other two within the range of 2.8-3.0 Å. In all cases, the smaller propeller twist is also accompanied by better hydrogen-bond angles ($C.C_4$-$C.N_4$-$G.O_6$, $G.C_6$-$G.N_1$-$C.N_3$, and $G.C_2$-$G.N_2$-$C.O_2$). The $A.C_1'$-$T.C_1'$ and $A.C_8$-$T.C_6$ distances appear to be irregular for all four pairs and, thus, are of no help in determining which conformation is better.

One Step Further for Poly[d(A)-d(T)]

Even though base pairs can be built from the calculated α and χ angles for poly[d(A)-d(T)], the α angle of 42.1° for d(T) is rather large. Supporting evidence comes from the structure of poly[d(A)-d(T)] in the B-form, which has a large propeller twist so that an extra hydrogen bond forms between $A.N_6$ of one base pair and $T.O_4$ of the next pair.[55] To verify the existence of this cross-pair hydrogen bond in a B-form structure, we need to define some parameters. In our coordinate system, a standard B-form helix of DNA will have its helical axis at x = 0.86 and y = 2.40 Å.[56] The rise, dz, is 3.38 Å, and the rotational angle along the helical axis between two base pairs is +36°. Adding the twist and tilt from this work and then generating the second pair through rotation define a new helix axis displaced dx = +0.0 and dy = +0.3 Å.

Now, for each of the four possible AT pairs, namely, +/+, +/-, -/+, and -/-, we put a second pair (A2-T2) on top of the first one (A1-T1) according to the above parameters and calculate the length ($A1.N_6$-$T2.O_4$) and angle ($A1.C_6$-$A1.N_6$-$T2.O_4$) of this special hydrogen bond. The results listed in Table XII show that there indeed exists a hydrogen bond of length 2.87 Å and angle 118° between $A1.C_6$ and $T2.O_4$ atoms if the paired sign of α angles is -/-, and the bond distance is actually the shortest distance among any two atoms

Table XII. Cross-Pair Hydrogen Bond of poly[d(A)-d(T)]

Sign of α	A1.N_6---T2.O_4	
	Length (Å)	Angle (deg)
+/+	5.15	90
+/-	1.63	101
-/+	5.42	105
-/-	2.87	118

between bases of A1-A2, A1-T2, A2-T1 and T1-T2. For +/+, +/- and -/+ pairs, the hydrogen bond can barely form, and some atomic contact distances are too small to be acceptable. However, as have been stressed earlier, we have more degrees of freedom than required to determine a possible structure, and the results presented here should not be taken as unique. In the case of finding this cross-pair hydrogen bond, we tried only the smallest and most reasonable dx and dy that can give results satisfying our conditions, and the -/- pair appeared to be the one of choice.

We conclude that the large α angle for d(T), as well as the large propeller twist between d(A) and d(T), is possible, and the overall picture may be considered as the actual conformation of poly[d(A)-d(T)] in solution.

RECENT DISCOVERIES

We have described two interesting observations in the RESULTS and DISCUSSION section: (1) that the second LD band of d(T) for poly[d(A)-d(T)] is positive, and (2) that for all of the polymers examined in this study the uncertainty in the fitted χ angles is always larger than that of α angles for the same base. Now we present our very recent discoveries that will explain both observations.

First, we combine Eq. 1 and Eq. 8 and express the reduced linear dichroism for a transition dipole i of base j as follows:

$$L_{ij} = \frac{3}{2} [3 \sin^2\alpha_j \sin^2(\chi_j - \delta_{ij}) - 1] \qquad \text{Eq. 9}$$

where we take S = 1 for this illustration. We will omit the subscript i and j for clarity through this section, but one must not forget that for each base j, there are several dipoles i.

Depending on the three angles in this expression, the value for L can range from -1.5 to 3.0. Two configurations will render L equal to -1.5: (1) α = 0°, the base is perpendicular to the helical axis, and the direction of the transition dipole makes no difference (that is, the χ-δ term in Eq. 9 has no effect), and (2) χ-δ = 0°, the transition dipole falls in the inclination axis which in turn is defined to be perpendicular to the helix axis, and so it does not matter what the magnitude of the inclination angle is. For L to be 3.0 it is necessary that both α and χ-δ are 90°. The molecular configuration in this case would be that the base is parallel to the helical axis (which is very unlikely to happen) and the transition dipole is perpendicular to the inclination axis and hence also parallel to the helical axis.

In order to infer molecular configurations for L values other than the two

extremes, we plotted the L in gray shadings representing its values for α in the range of [0°,90°] in one dimension and χ-δ in the range of [-90°,90°] in the other dimension (Figure 11). This plot immediately suggests that: (1) L is more likely to be positive with larger α and $|\chi-\delta|$ angles, and (2) L is less sensitive to χ-δ with smaller α angles. We will explore these two points in more detail below.

Figure 12 visualizes why the second band of d(T) in poly[d(A)-d(T)] has a positive LD. The vertical line indicates the α value of d(T), which is 42.1° (Table II), and the three horizontal lines are χ-δ values (30.1°, 74.1° and 47.1° respectively) for the angles of three transition dipoles of d(T). The shaded areas at the intersections give the L values for the three bands. It is clear that with the large α angle and the large χ-δ value (74.1°), the L for the second band extends well into the positive shaded areas in Figure 12. The message here is that although the measured total LD spectra are always negative, one must not assume all individual LD bands are negative. Indeed, a simple calculation from Eq. 9 reveals that positive LD bands may exist and partially cancel the negative LD spectra if a base inclined more than 35.3°.

Figure 11 also explains why fitted χ angles have a larger standard deviation than α angles. At α = 15°, we see that gray-shade boundaries are crossed only a few times from the bottom of the box to the top. What this means is that very different χ-δ angles can give similar L values when α is small. On the other hand, for χ-δ = -75° across the box, we see that L changes more rapidly. Although L becomes more sensitive to χ-δ as α approaches 90° and less sensitive to α when χ-δ approaches 0°, most α angles are small (less than 35°) and χ-δ angles spread over the range of [-90°,90°] in this study, so it is inevitable for χ angles to have larger standard deviations than α angles.

Figure 11. Reduced linear dichroism, L, plotted as a function of inclination angle α and χ-δ angle, where χ is inclination axis and δ is transition dipole direction. L values in the graph box are presented as different level of gray shadings. The mapping from L's to shadings is shown at the right column.

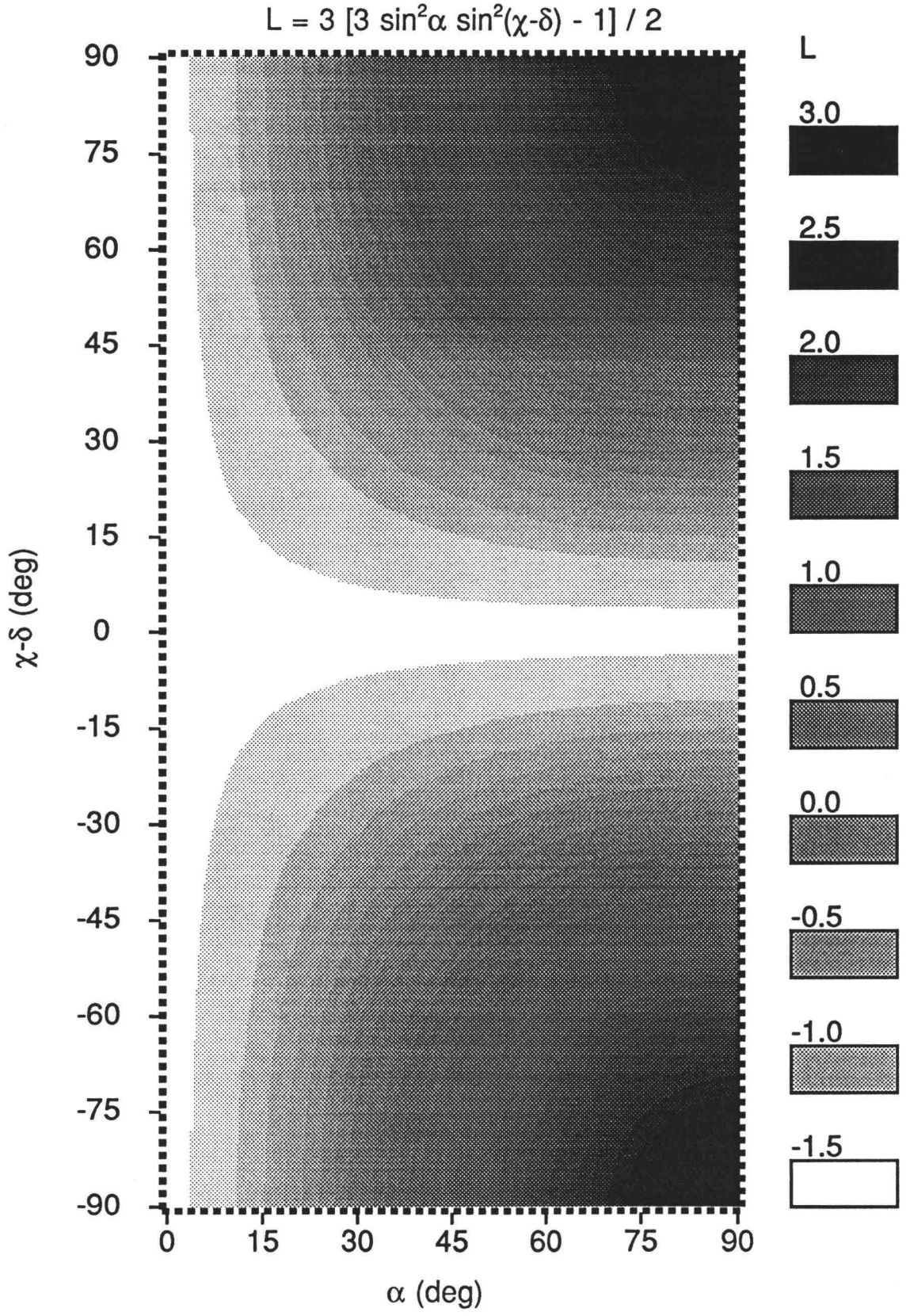
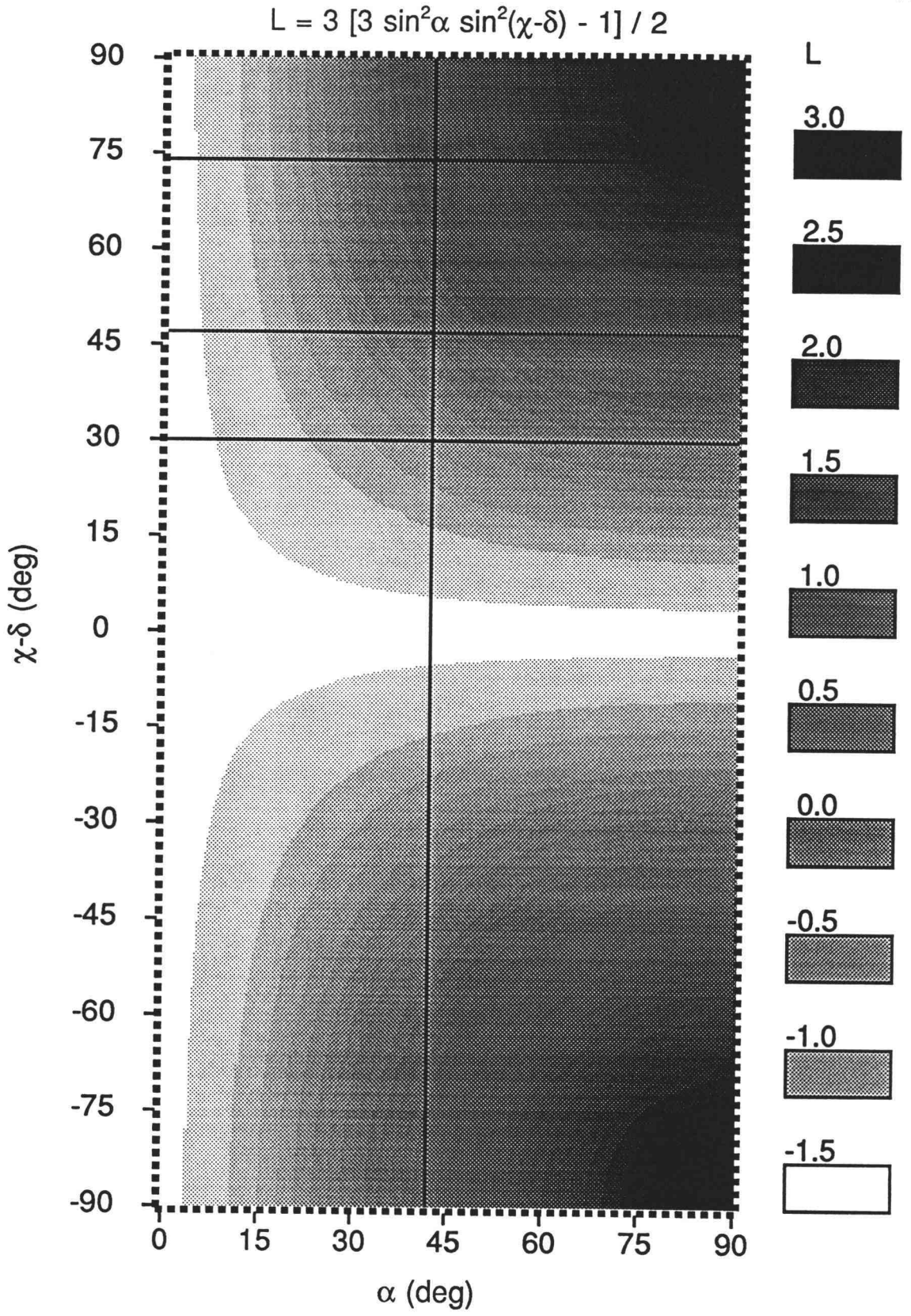

L = 3 [3 sin²α sin²(χ-δ) - 1] / 2
L
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
90
75
60
45
30
15
0
-15
-30
-45
-60
-75
-90
χ-δ (deg)
0
15
30
45
60
75
90
α (deg)

Figure 12. Graphical explanation of positive LD bands and insensitivity of χ to the LD data. The inclination angle α and the three χ-δ angles for d(T) of poly[d(A)-d(T)] are shown as a vertical line (α = 42°) and horizontal lines (χ-δ = 30°, 74° and 47° for the 268.0, 203.7 and 177.5 nm bands), respectively. The shaded areas at the intersections give the L values for the three bands.


L = 3 [3 sin²α sin²(χ-δ) - 1] / 2
90
75
60
45
30
15
0
-15
-30
-45
-60
-75
-90
χ-δ (deg)
0
15
30
45
60
75
90
α (deg)
L
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5

BIBLIOGRAPHY

1. Langridge, R.; Marvin, D. A.; Seeds, W. E.; Wilson, H. R.; Hooper, C. W.; Wilkins, M. H. F.; Hamilton, L. D. *J. Mol. Biol.* **1960**, *2*, 38-61.
2. Fuller, W.; Wilkins, M. H. F. *J. Mol. Biol.* **1965**, *12*, 60-80.
3. Arnott, S.; Hukins, H. W. L. *J. Mol. Biol.* **1972**, *149*, 761-786.
4. Bram, S.; Tougard, P. *Nature (London) New Biol.* **1972**, *239*, 128-131.
5. Girod, J. C.; Johnson, W. C., Jr.; Huntington, S. K.; Maestre, M. F. *Biochemistry* **1973**, *12*, 5092-5096.
6. Ivanov, V. I.; MinchenKova, L. E.; Schyolkina, A. K.; Poletayev, A. I. *Biopolymers* **1973**, *12*, 89-110.
7. Pohl, F. M. *Nature* **1976**, *260*, 365-366.
8. Leslie, A. G. W.; Arnott, S.; Chandrasekaran, R.; Ratliff, R. L. *J. Mol. Biol.* **1980**, *143*, 49-72.
9. Shakked, Z.; Kennard, O. In *Structural Biology*; McPherson, A., Jurnak, F., Ed.; Wiley: New York, 1983.
10. Cavalieri, L. F.; Rosenberg, B. H.; Rosoff, M. *J. Am. Chem. Soc.* **1956**, *78*, 5235-5238.
11. Gray, D. M.; Rubenstein, I. *Biopolymers* **1968**, *6*, 1605-1631.
12. Wada, A. *Appl. Spectrosc. Rev* **1972**, *6*, 1-30.
13. Ding, D.; Rill. R.; van Holde, K. E. *Biopolymers* **1972**, *11*, 2109-2124.
14. Hofrichter, J.; Eaton, W. A. *Annu. Rev. Biophys. Bioeng.* **1976**, *5*, 511-560.
15. Nordén, B. *Appl. Spectrosc. Rev.* **1978**, *14*, 157-248.
16. Hogan, M.; Dattagupta, N.; Crothers, D. M. *Proc. Natl. Acad. Sci. U.S.A.* **1978**, *75*, 195-199.
17. Rizzo, V,; Schellman, J. *Biopolymers* **1981**, *20*, 2143-2163.
18. Charney, E.; Yamaoka, K. *Biochemistry* **1982**, *21*, 834-842.
19. Lee, C.; Charney, E. *J. Mol. Biol.* **1982**, *161*, 289-303.
20. Diekmann, S.; Hillen, W.; Jung, M.; Wells, R. D.; Pörschke, D. *Biophys. Chem.* **1982**, *15*, 157-167.
21. Matsuda, K.; Yamaoka, K. *Bull. Chem. Soc. Jpn.* **1982**, *55*, 1727-1733.

22. Matsuoka, Y.; Nordén, B. *Biopolymers*, **1982**, *21*, 2433-2452.

23. Matsuoka, Y.; Nordén, B. *Biopolymers*, **1983**, *22*, 1731-1746.

24. Edmondson, S. P.; Johnson, W. C., Jr. *Biochemistry* **1985**, *24*, 4802-4806.

25. Charney, E.; Chen, H. H.; Henry, E. R.; Rau, D. C. *Biopolymers* **1986**, *25*, 885-904.

26. Charney, E.; Chen, H. H. *Proc. Natl. Acad. Sci. U.S.A.* **1987**, *84*, 1546-1549.

27. Charney, E. Q. *Rev. Biophys.* **1988**, *21*, 1-60.

28. Flemming, J.; Pohle, W.; Weller, K. *Int. J. Biol. Macromol.* **1988**, *10*, 248-254.

29. Rau, D. C.; Charney, E. *Biophys. Chem.* **1983**, *17*, 35-50.

30. Causley, G. C.; Johnson, W. C., Jr. *Biopolymers* **1982**, *21*, 1763-1780.

31. Dougherty, A. M.; Causley, G. C.; Johnson, W. C., Jr. *Proc. Natl. Acad. Sci. U.S.A.* **1983**, *80*, 2193-2195.

32. Edmondson, S. P.; Johnson, W. C., Jr. *Biopolymers* **1985**, *24*, 825-841.

33. Edmondson, S. P.; Johnson, W. C., Jr. *Biopolymers* **1985**, *25*, 2335-2348.

34. Levitt, M. *Proc. Natl. Acad. Sci. U.S.A.* **1978**, *75*, 640-644.

35. Sarai, A.; Jazur, J.; Nussinov, R.; Jernigan, R. L. *Biochemistry* **1988**, *27*, 8498-8502.

36. Ansevin, A. T.; Wang, A. H. *Nucleic. Acids Res.* **1990**, *18*, 6119-6126.

37. Weiner, P. K.; Kollman, P. A. *J. Comput. Chem.* **1981**, *2*, 287-303.

38. Edmondson, S. P. *Biopolymers* **1987**, *26*, 1941-1956.

39. Brown, K. M.; Dennis, J. E., Jr. *Numer. Math.* **1972**, *18*, 289-297.

40. Brent, R. P. In *Algorithms for Minimization Without Derivatives*; Thomas J. Watson Research Center: Yorktown Heights, NY, 1973.

41. Alper, J. S.; Gelb, R. I. *J. Phys. Chem.* **1990**, *94*, 4747-4751.

42. Siano, D. B.; Metzler, D. E. *J. Chem. Phys.* **1969**, *51*, 1856-1861.

43. Siano, D. B. *J. Chem. Educ.* **1972**, *49*, 755-757.

44. Clark, L. B. *J. Am. Chem. Soc.* **1977**, *99*, 3934-3938.

45. Clark, L. B. *J. Phys. Chem.* **1989**, *93*, 5345-5347.

46. Clark, L. B. *J. Phys. Chem.* **1990**, *94*, 2873-2879.

47. Zaloudek, F.; Novros, J. S.; Clark, L. B. *J. Am. Chem. Soc.* **1985**, *107*, 7344-7351.

48. Novros, J. S.; Clark, L. B. *J. Phys. Chem.* **1986**, *90*, 5666-5668.

49. Arnott, S. *Prog. Biophys. Mol. Biol.* **1970**, *21*, 265-319.

50. Brahms, J,; Mommaerts, W. F. H. M. *J. Mol. Biol.* **1964**, *10*. 73-88.

51. Wang, J. C. *Cold Spring Harbor Symp. Quant. Biol.* **1978**, *43*, 29-34.

52. Wang, J. C. *Proc. Natl. Acad. Sci. U.S.A.* **1979**, *76*, 200-203.

53. Baase, W. A.; Johnson, W. C., Jr. *Nucleic Acids Res.* **1979**, *6*, 797-814.

54. Williams, L. D.; Williams, N. G.; Shaw, B. R. *J. Am. Chem. Soc.* **1990**, *112*, 829-833.

55. Bhattacharyya, D.; Bansal, M. *J. Biomol. Struct. Dyn.* **1990**, *8*, 539-572.

56. Takusagawa, F. *J. Biomol. Struct. Dyn.* **1990**, *7*, 795-809.

57. Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. In *Numerical Recipes in C*; Cambridge University Press: New York, NY, 1988.

SECTION III
Conclusion

The study presented in SECTION II, that is, determination of DNA base inclinations, has it own significance in the field of DNA structure research. More importantly, this section of the thesis serves to illustrate how a comprehensive analysis can be achieved, and demonstrates that such an analysis is vital for a well designed experiment. In summary:

1. The analysis yield DNA base inclinations is a case study. Many other experiments with precious information buried in their measurements are still waiting for a conclusive analysis like this one.

2. Time is an important factor. Computer technology advances with time. Chemical and physical parameters of interest emerge with time. Mathematical and numerical methods evolve with time. These factors are certain to contribute new ideas to experimental designs.

3. The ability to develop in-house computer programs for data analysis in biochemistry and biophysics research should be recognized as an alternative to the ability to do on-the-bench experiments.

4. The combined power of computer hardware and software is not limited to data analysis. Simulation of molecular dynamics on computers has been done for years, and it is my personal belief that it will not be long before entire experiments (sample preparations, instrument, and measurements) will be simulated completely on computers.

APPENDIX

Overview of the *ABS-LD* and *PARSE-R* Programs

Two computer programs are used in this study. One is the fitting program *ABS-LD* and the other is *PARSE-R*, an auxiliary program for data conversion. The usage of *ABS-LD* is described in detail under the subtitles **Command Line Arguments for *ABS-LD* Program** and **Iteration Controls in *ABS-LD* Program**. The usage of *PARSE-R* is described under the subtitle **Command Line Arguments for *PARSE-R* Program**. The program source code for *ABS-LD* is given in Lists 1-5, and for *PARSE-R* in List 6. Both programs are written in C language.

The program *ABS-LD* is logically divided into 5 parts in the following short descriptions and in the lists for the purpose of clarity only; they should be concatenated in the presented order into a single program and compiled together.

Standard header files are listed in List 1. These files contain the function prototype and constant declarations used in the rest of the program.

List 2 is a modified version of the LU decomposition presented in the book *Numerical Recipes in C*.[57] LU decomposition is a commonly used technique to perform matrix inversion and solve linear equations. It is used in this program to invert the positive definite symmetry matrix generated by the Levenberg-Marquardt algorithm (Eq. 7 in METHODS section). Two additional functions are added to handle memory allocation and deallocation.

List 3 is our implementation of the Levenberg-Marquardt algorithm.[39] The main algorithm is implemented in the function **marquardt()**; the calculation of standard deviations for all variables is done in **marquardt_stdv()**. The remaining two functions are for memory management.

List 4 contains supporting functions. These are subroutines to perform input, output, band-shape calculation, spectra normalization and communication with the **marquardt()** function, plus some data structure

definitions.

List 5 is the **main()** function, the program entry point in C language. The behavior and response of the program are controllable through several run-time parameters and options provided in this function.

The program *PARSE-R* is used to convert the results of repeated fittings from *ABS-LD* to a format acceptable by spreadsheet programs. Most spreadsheet programs can not import a line of length longer than 256 characters, and each line of output from *ABS-LD* for repeated fittings is always longer than that. Hence the need of this program.

PARSE-R is listed in List 6.

Command Line Arguments for the *ABS-LD* Program

To run the program, type *ABS-LD* followed by your filename for the *PEAK*, *SPECTRUM* and *OUTPUT* files. For example, under the DOS operating system on a PC, type

```
ABS-LD P-CMP.11 DCMP.ABS O-CMP11.1
```

and hit <Return>. Here *P-CMP.11* is a *PEAK* file, *DCMP.ABS* is a *SPECTRUM* file, and *O-CMP11.1* is an *OUTPUT* file. Their formats are described below. The program expects exactly three command line arguments like this, otherwise it will print the message

```
need PEAK, SPECTRUM and OUTPUT files
```

and terminate immediately. If, for example, the program can't find or open the file *P-CMP.11*, which was entered as the *PEAK* file, the program will print the message

```
can't open PEAK file: P-CMP.11
```

and terminate immediately. The same error checking is applied to the *SPECTRUM* and *RESTRICTION* files. The *RESTRICTION* file is described in conjunction with the *PEAK* file below.

The *PEAK* File Format

All numerical fields in the *PEAK* file must be separated by at least one <Space> character or separated into different lines. Do not use <Comma> character to separate fields. The first integer number in the *PEAK* file is the number of bases involved in this fitting. If the number of bases is 1, the program assumes we are fitting an absorption spectrum only. If the number of bases is greater than 1, the program fits an absorption spectrum, and an LD spectrum using α and χ angles for each base. This means that if you want to fit both absorption and LD for a polymer with a single base, you must set the number of bases to 2, and repeat the data for the base. In all cases, the program reads an integer number as the number of transition dipoles for each base, and then reads five real numbers in a row as the initial guesses for position, intensity, width, and skewness, and the dipole direction for each band. This input process repeats for each band of a base and each base in the *PEAK* file.

Example *P-GMP.1* illustrates the *PEAK* file for one base and an absorption-only fit.

Example *P-GMP.1*

```
1
4
272.5   420100   22.8   1.50    -4
248.7   223400   12.5   1.01   -75
197.8   537500   13.3   1.24   -71
183.5   341600   11.6   1.50    41

from Table 1 of old paper
fix rho of the 1st and 4th band at 1.50 (originally 1.94 and 1.55)
restriction file: R-GMP.2
```

Note that the line starting "from Table ..." and there after are comments. The program will close the *PEAK* file immediately after all required data are read so the comments are effectively ignored. Comments can serve to indicate how this *PEAK* file is composed. The transition dipole directions are

not used by the program for decomposition of a monomer absorption spectrum.

After reading the data, the program then prompts you to enter the filename for a *RESTRICTION* file with the message `restriction file:`. The program use the information in the *RESTRICTION* file to identify if any band parameters are to be fixed at the values given in *PEAK* file, and not subject to the fitting. As shown in Example *R-GMP.1*, the *RESTRICTION* file for Example *P-GMP.1*, the skewness of the first and fourth bands are to be fixed. A *RESTRICTION* file contains as many lines as the number of bands of the corresponding *PEAK* file. Each line has four numbers (for band position, intensity, width, and skewness) of either 0 or 1 indicating a parameter is to be fixed or fitted, respectively.

Example *R-GMP.2*

```
1  1  1  0
1  1  1  1
1  1  1  1
1  1  1  0
```

When the number of bases is not 1 (the first integer in the *PEAK* file), the program will read two additional real numbers as the α and χ angles (in degrees) immediately after the integer for the number of dipoles for each base, as shown in Example *P-ZGCGC.3*.

Example *P-ZGCGC.3*

```
2
4  26.64  137.30
283.77   96882  16.47  1.500   -4
249.20  126800  14.72  1.018  -75
197.99  142769  12.72  1.023  -71
177.56  220647  11.44  1.066   41
5  31.94  202.16
265.46   97272  15.41  1.035    6
218.17   92848  15.89  1.238  -35
206.16   33702   6.13  1.088   76
193.44  116471   9.59  1.448   86
184.52   94241   7.20  1.035    0

taken from ZGCGC2.1 at the 39th iterations
fix rho of G1 at 1.5 with restriction file R-GC.2
```

Example *R-GC.2*

```
1 1
1 1 1 0
1 1 1 1
1 1 1 1
1 1 1 1
1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
```

To match the data in Example *P-ZGCGC.3*, two integers (1 1) are added for each base in Example *R-GC.2* to indicate that α and χ angles for both bases are to be fitted.

The following example is a *PEAK* file for fitting the absorption and LD spectra for natural DNA with four bases. There is no *RESTRICTION* file. When the program asks for the filename of the *RESTRICTION* file in this case, simply hit the <Return> key and the program will assume there is no *RESTRICTION* file and all variables (band parameters and α and χ angles) are subject to fitting.

Example *P-ZDNA.1*

```
4
6  24.52   6.67
270.85    50135   13.41   1.151    83
256.06    90265   14.45   1.306    25
206.39   161589   12.82   1.065   -45
195.62    39315    6.15   1.068    15
185.43   112444    7.39   1.067    72
174.32    28857    3.92   1.054   -45
3  33.70  -6.87
268.40   115561   17.01   1.191    -9
203.99   157246   23.67   1.416   -53
177.19    78019    5.59   1.131   -26
4  13.57  81.34
279.71    82034   16.84   1.454    -4
248.68   130043   14.03   1.017   -75
196.31   141293   12.35   1.007   -71
180.02   233621   10.11   1.365    41
5  35.73  216.72
263.75    96001   14.71   1.111     6
221.40    93564   17.43   1.394   -35
209.89    34261    9.72   1.020    76
193.07   123689   10.11   1.095    86
182.53   104113   10.84   1.085     0

taken from P-ATAT.2 and P-GCGC.2
```

The *SPECTRUM* File Format

The second argument to the program *ABS-LD* is your filename for the *SPECTRUM* file. The first two integers in a SPECTRUM file are the starting and ending wavelengths for the data that follows, with the shorter wavelength first and 1-nm interval assumed. If the number of bases (determined from *PEAK* file) is 1, then there is only one column of data representing the absorption spectrum. If the number of bases is not 1, then there are two columns of data; the first column is the absorption spectrum, and the second one is the LD spectrum. Partial lists of the *SPECTRUM* files for DAMP absorption spectrum and 10.4B-DNA absorption and LD spectra are shown in the following examples. Note that the LD spectrum in Example *104B-DNA.ALD* has been normalized to the same area as the absorption spectrum. All numerical fields in the *SPECTRUM* file must be separated by at

least one <Space> character or into different lines. Do not use the <Comma> character to separate fields.

Example *DAMP.ABS*

```
176  288
14111.12
14444.44
15000.00
15777.78
16444.44
17555.56
...
```

Example *104B-DNA.ALD*

```
177 300
13593.12463 13131.12435
13874.21217 13445.10512
14076.05695 13674.61011
14182.95993 13852.53255
14316.77554 14027.46469
14391.53287 14115.67834
...
```

The *OUTPUT* File Format

The third argument to the program *ABS-LD* is the filename of your *OUTPUT* file. All data generated during the program execution will be written to this file (see below for exceptions). In the following example, Example *O-AT2.1*, three lines of header are followed by the results of the first iteration of this fitting.

Example *O-AT2.1*

```
D:\DNA\ABS-LD.EXE  P-AT.2  AT.ABS  O-AT2.1

256  2  1e+010  3000000  0  (7)  [1]  M=256  N=40  <0.25>

orientation ABS/LD = 1.000000e+000

40  2.560000e+002   9.069433e+004  2.947606e+007
IAnormal=1.000616,  IAssq=1.272848e+007
LDnormal=-1.931784,        LDssq=1.665688e+007

 275.70 ( 18.62)   29182 (265487)  11.25 (24.60)  1.205 (1.621)  1.0573
 254.59 (  9.08)  100065 (377289)  13.86 (10.72)  1.332 (0.654)  1.3734
 208.44 (  1.44)  126690 (123047)  10.50 ( 2.57)  1.205 (0.415)  1.8150
 197.27 (  1.77)   52040 ( 50102)   6.12 ( 2.54)  1.379 (0.642)  1.5405
 185.69 (  5.68)  151272 (219676)   7.57 ( 8.24)  1.292 (1.185)  0.9820
 172.53 (142.03)   42533 (1800813)   4.49 (118.66)  1.002 (16.987)  1.8150
   24.05 ( 15.34)   -24.61 ( 35.97)

 268.16 ( 45.42)  117889 (395540)  18.00 (37.73)  1.253 (1.128)  1.2995
 206.43 (  7.15)  129761 (158944)  19.69 (31.78)  1.499 (1.943) -0.3294
 178.26 (  3.18)  131475 (250814)   5.80 ( 4.02)  1.420 (0.616)  0.6034
   40.37 ( 32.87)    21.66 ( 48.78)
```

The first line of the header is printed as a reflection of the command line, which started the program execution. The second line contains various control and status values for this fitting:

256	LMcoef
2	LMstep
1e+010	LMlimit
3000000	SSQlimit
0	SSQpercent
(7)	option
[1]	weight of ABS.ssq over LD.ssq
M=256	the number of data points
N=40	the number of variables
<0.25>	partial derivative control

The number of data points is all the data points being fitted; here both the absorption and LD spectra are being fitted simultaneously. If the fitting is for the monomer absorption spectrum only, then it will be equal to the number of data points of the absorption spectrum. The number of variables is the sum of the number of band parameters for all bands plus 2 (for α and χ) for each base, minus the number of variables to be fixed as specified in the *RESTRICTION* file. The meaning of all other values in the second header line

will be described in detail later.

The third line of the header is the scale factor calculated by the program to normalize the LD spectrum to the same area as the absorption spectrum. The program always normalizes the LD spectrum after the *SPECTRUM* file is read, and uses the normalized version of the LD spectrum for the fitting. Since the LD spectrum in the *SPECTRUM* file *AT.ABS* has already been normalized and its sign reversed, this number is 1.00.

At the end of each iteration, the program will generate a block of results similar to the one in the rest of Example *O-AT2.1* after the header. The meaning of each value in the first three lines of results are:

40	the number of active variables
2.560000e+002	current LMcoef
9.069433e+004	decrease of total ssq by this iteration
2.947606e+007	total ssq (ABS.ssq + LD.ssq) after this iteration
1.000616	should be close to 1
1.272848e+007	ABS.ssq after this iteration
-1.931784	normalization factor
1.665688e+007	LD.ssq after this iteration

The number of active variables will be smaller than the number of variables shown in the second line of header if, during this iteration, any of the variables have no effect on the fitting and are excluded by the program. Theoretically, this means these variables have reached their optimal values and the first partial derivatives of the minimization function with respect to these variables are zero. But in practice, this always means these variables have either very large or very small values, such that they no longer have any effect on the fitting. A typical example is that the position of a band shifts to 500 nm.

The current LMcoef will be explained later. The normalization factor is the scale factor that normalizes the calculated LD spectrum to the absorption spectrum. If the LD spectrum in the *SPECTRUM* file has not been normalized (that is, the original measured LD spectrum), then the alignment factor S in Eq. 8 can be calculated by dividing this normalization factor by the orientation

factor in the third line of the header.

Next in the results are, for each base, fitted values for band position, intensity, width, and skewness for all bands, and may be followed by α and χ angles if the LD spectrum is also fitted as in this example. The number in each parentheses is the square root of the diagonal element of the matrix $[\mathbf{J(x)}^T\mathbf{J(x)}]^{-1}$, corresponding to the variable immediately preceding the parentheses. See the heading **Nonlinear Linear Square Fitting** in METHODS section for the meaning of these numbers. The last number in each line of band parameters is the scale factor that, when it is multiplied by the absorption band, gives the corresponding LD band. So, for example, this factor for the second band of d(T) in Example *O-AT2.1* is -0.3294, and that means the LD band has changed sign and is 0.3294 the height of the corresponding absorption band.

Iteration Controls in the *ABS-LD* Program

After the data in the *PEAK* and *SPECTRUM* files are read, the program then asks values for the following controls on the screen,

```
eps, Lmcoef, Lmstep, LMlimit, SSQlimit, SSQpercent, SSQia/ld, option:
```

and they are entered from the keyboard, with at least one <Space> character separating two values.

eps

The eps is used by the Levenberg-Marquardt algorithm to compute h = u^{eps}, and h is the infinitesimal used to approximate the first partial derivatives of the minimization function with respect to all variables (heading **Nonlinear Least Squares Fitting** in METHODS section). The u is the unit-round of a double precision floating-point number of the host machine on which the program is executing, so its value may vary from system to system, but it is

always predefined in the C language header file <float.h> as the constant DBL_EPSILON. According to the IEEE 754 standard, a double precision floating-point number (which is the data type used in this program for fitting) will have its unit-round close to $2x10^{-16}$. To obtain a reasonable value for h, which is about 10^{-4} to 10^{-8} depending on the complexity of the minimization function, the eps should be set to at least 0.25 and at most 0.5 on a machine conforming to the IEEE 754 standard.

LMcoef, LMstep and LMlimit

The LMcoef, LMstep and LMlimit are used to control the behavior of the Levenberg-Marquardt algorithm (heading **Nonlinear Linear Square Fitting** in METHODS section). The Levenberg-Marquardt coefficient, LMcoef, should initially be a large value for a fitting, if the initial guesses for the variables are expected to be far from the optimal values. The larger the LMcoef, the smaller the adjustment for variables in each iteration. This strategy can prevent a poor initial guess from running wild.

After each successful iteration (the sum of squares error, ssq, is decreased), the LMcoef is divided by the LMstep to decrease the LMcoef and hence speed up the fitting with larger adjustments, so that LMstep must be greater than 1. If the ssq increases after an iteration, LMcoef is multiplied by the LMstep. This retraction usually results in a change of the direction of the minimization path, and thus prevents the algorithm from been trapped at a local minimum. When the fitting is moving forward (ssq is decreased), LMstep acts as an accelerator; when the algorithm steps into a local minimum, it acts as a snooper searching for other directions for the fitting. If a local minimum is very deep, the algorithm may not escape the trap even after it has been multiplied to the LMstep many times. In this case, we need to set an upper bound for the LMcoef, and this upper bound is LMlimit. If LMcoef becomes larger than the LMlimit, then the algorithm will stop. After all, a local minimum

this deep may well be considered as a global minimum.

In this study, LMcoef is set initially to 1024 or 256, LMstep to 2, and LMlimit to 10^{10}.

SSQlimit and SSQpercent

SSQlimit and SSQpercent are used as major criteria for stopping the algorithm. SSQlimit should be set to a reasonably small value, because when ssq is decreased to less than SSQlimit, the fitting will stop. Unless the minimal ssq can be estimated or is specifically known before the fitting starts, do not use this criterion. Instead, set SSQlimit to zero and SSQpercent to a small fraction. When the k^{th} iteration is successful and $(ssq_{k-1} - ssq_k)/ssq_{k-1}$ is less than the SSQpercent, then the fitting will stop.

Here we illustrate how to make use of the properties of SSQlimit and SSQpercent to obtain good fitting results. In this study, a fitting is always run at least twice. For the first run, SSQlimit is set to 0, and SSQpercent to 10^{-3} for natural DNA, 10^{-4} for synthetic polymer and 10^{-5} for monomer. This setting will keep the fitting proceeding well past the iteration that gives the optimal solution. The results of each iteration are manually examined, and an optimal solution is chosen. Then for the second run, the SSQlimit is set to a value larger than the ssq of the chosen optimal iteration, but smaller than the ssq of the iteration previous to the optimal iteration, and SSQpercent is set to 0. This time, the fitting will stop right after the chosen iteration and results can now be printed for this optimal solution.

SSQia/ld

The SSQia/ld can be used to weigh ABS.ssq and LD.ssq differently during preliminary studies if desired. Results presented in this thesis are from fittings with SSQia/ld set to 1. See heading **Decomposition of Synthetic Polymer Absorption and LD spectra** in the RESULTS and DISCUSSION

section for this reasoning.

option

Finally, the option. It is an integer taken as the sum of the following options:

1	print simple information after each iteration
2	print detailed information after each iteration
4	print simple and/or detailed information to the *OUTPUT* file
8	print the fitted spectra to the *OUTPUT* file
16	print all fitted bands to the *OUTPUT* file
32	proceed to repeated fittings after the first fitting is done

In Example *O-AT2.1*, the simple one-line information is

```
40  2.560000e+002   9.069433e+004  2.947606e+007
```

and the detailed information is

```
IAnormal=1.000616,  IAssq=1.272848e+007
LDnormal=-1.931784, LDssq=1.665688e+007

 275.70 ( 18.62)   29182 (265487)  11.25 (24.60)  1.205 (1.621)  1.0573
 254.59 (  9.08)  100065 (377289)  13.86 (10.72)  1.332 (0.654)  1.3734
 208.44 (  1.44)  126690 (123047)  10.50 ( 2.57)  1.205 (0.415)  1.8150
 197.27 (  1.77)   52040 ( 50102)   6.12 ( 2.54)  1.379 (0.642)  1.5405
 185.69 (  5.68)  151272 (219676)   7.57 ( 8.24)  1.292 (1.185)  0.9820
 172.53 (142.03)   42533 (1800813)   4.49 (118.66)  1.002 (16.987)  1.8150
   24.05 ( 15.34)   -24.61 ( 35.97)

 268.16 ( 45.42)  117889 (395540)  18.00 (37.73)  1.253 (1.128)  1.2995
 206.43 (  7.15)  129761 (158944)  19.69 (31.78)  1.499 (1.943) -0.3294
 178.26 (  3.18)  131475 (250814)   5.80 ( 4.02)  1.420 (0.616)  0.6034
   40.37 ( 32.87)    21.66 ( 48.78)
```

The format for the output spectra is three columns of data for absorption-only fitting, and five columns for absorption and LD fitting. In both formats, the first column is the wavelength, the second column is the original input absorption spectrum, and the third column is the fitted absorption spectrum. If the LD spectrum is also fitted, then the fourth column is the original input LD spectrum, and the fifth column is the fitted LD spectrum.

The format for the fitted bands written to the *OUTPUT* file also depends

on whether LD is involved. For absorption-only fitting (that is, there is only one base), all bands of the base are printed in columns (one band, one column), no wavelength column is inserted. If LD is present, then for each base, the absorption bands are printed first followed by one blank line and then the LD bands.

Different combinations of the above options will be useful for different fitting conditions. Here is a typical sequence of fittings. For the first round of fitting, in which the initial guesses for the variables may be pretty far away from their optimal values, the option could be set to 7, which is 1 + 2 + 3. This will cause all available information after each iteration to be written to the *OUTPUT* file. These results can be examined after the fitting to determine which iteration gives the best results. Then start the second fitting with the option set to 25, which is 1 + 8 + 16 (and with the new SSQlimit and SSQpercent, see above). This time, only one line of information is printed to the screen after each iteration so that the progression of the fitting can be monitored. At the end of the fitting, the detailed information of the last iteration is written to the *OUTPUT* file, followed by the input and fitted spectra, and then the spectra for all bands. A spreadsheet computer program can then import these spectra from the *OUTPUT* file, and plot the original and fitted spectra and bands for visual inspection. If repeated fittings are desired (heading **Uncertainties in Transition Dipole Directions** in the METHODS section and heading **Repeated Fittings with Randomized Transition Dipole Directions** in the RESULTS and DISCUSSION section), then the results of the second round of fitting can be edited into a new *PEAK* file to shorten the computing time. Set SSQlimit to any value larger than the ssq of the last results, set SSQpercent to 1, and set option to 32.

Repeated Fittings

The program starts the repeated fittings by first asking for the filename of a file to store the fitting results, the number of fittings to repeat, and the range of an uniform distribution from which random numbers are drawn:

```
input RANDOM filename, repeat &  range:
```

The program first records the number of repeats, the range, and the number of bases as well as the number of bands of each base in the output file so that it can be accessed by other programs. There is no output generated after each iteration, only an integer count printed to the screen indicating at which repeat the program is currently executing. At the end of each fitting, the program writes the final values of all variables to the output file in one line. First the ABS.ssq and LD.ssq, then for each base, the band position, intensity, width and skewness for all bands, and then α and χ angles. Note that no standard deviations are written to the file in this format, only the values of the variables.

Command Line Arguments for the *PARSE-R* Program

To run the program, type *PARSE-R* followed by your filename for *RANDOM* and *OUTPUT* files. The number of *OUTPUT* files must be equal to the number of bases in the *RANDOM* file. For example, under the DOS operating system on a PC, type

```
PARSE-R O-GCGC2.R O-GCGC2.G O-GCGC2.C
```

and hit <Return>. Here *O-GCGC2.R* is the *RANDOM* file, the results of repeated fittings from the program *ABS-LD*. Since *O-GCGC2.R* contains data for two bases (guanine and cytosine), two *OUTPUT* files are required, and they are *O-GCGC2.G* and *O-GCGC2.C*.

The program separates the data in *O-GCGC2.R*, and writes the data

belonging to a base to an *OUTPUT* file. The data in each *OUTPUT* file will be one line for each iteration; and in each line, the band position, intensity, width and skewness of each band, followed by the α and χ angles. The ABS.ssq and LD.ssq are stored as the first two numbers in the first *OUTPUT* file. The <Comma> character is used to separate each field.

The *OUTPUT* files can then be read into a spreadsheet program to calculate averages and standard deviations for each variable over all repeats.

List 1. *ABS-LD* Header Files

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
```

List 2. *ABS-LD* LU Decomposition

```
void initiate_lu( int n );
void lu_inverse( int n );
int  lu_decompose( int n );
void terminate_lu( void );


double **LUalpha, *LUbeta, *LUscale;
int    *LUindex, LUn=0;


void initiate_lu( int n )
{
  int i;

  if( LUn > 0 )
    terminate_lu();
  LUn = n;
  LUalpha = (double **)calloc( n, sizeof(double *) );
  for( i=0; i<n; i++ )
    LUalpha[i] = (double *)calloc( n, sizeof(double) );
  LUbeta = (double *)calloc( n, sizeof(double) );
  LUscale = (double *)calloc( n, sizeof(double) );
  LUindex = (int *)calloc( n, sizeof(int) );
}


void lu_inverse( int n )
{
  int    i, j, k, ii;
  double sum;

  ii = -1;
  for( i=0; i<n; i++ )
  {
    k = LUindex[i];
    sum = LUbeta[k];
    LUbeta[k] = LUbeta[i];
    if( ii >= 0 )
      for( j=ii; j<i; j++ )
        sum -= LUalpha[i][j] * LUbeta[j];
    else
      if( sum != 0.0 )
        ii = i;
    LUbeta[i] = sum;
  }

  for( i=n-1; i>=0; i-- )
  {
    sum = LUbeta[i];
    for( j=i+1; j<n; j++ )
      sum -= LUalpha[i][j] * LUbeta[j];
    LUbeta[i] = sum / LUalpha[i][i];
  }
}
```

```
int lu_decompose( int n )
{
  int    i, j, k, ii;
  double big, sum, temp;

  for( i=0; i<n; i++ )
  {
    big = 0.0;
    for( j=0; j<n; j++ )
    {
      temp = fabs( LUalpha[i][j] );
      if( temp > big )
        big = temp;
    }
    if( big == 0.0 )
      return i;
    LUscale[i] = 1.0 / big;
  }

  for( j=0; j<n; j++ )
  {
    for( i=0; i<j; i++ )
    {
      sum = LUalpha[i][j];
      for( k=0; k<i; k++ )
        sum -= LUalpha[i][k] * LUalpha[k][j];
      LUalpha[i][j] = sum;
    }

    big = 0.0;
    for( i=j; i<n; i++ )
    {
      sum = LUalpha[i][j];
      for( k=0; k<j; k++ )
        sum -= LUalpha[i][k] * LUalpha[k][j];
      LUalpha[i][j] = sum;
      temp = LUscale[i] * fabs( sum );
      if( temp >= big )
      {
        big = temp;
        ii = i;
      }
    }

    if( j != ii )
    {
      for( k=0; k<n; k++ )
      {
        temp = LUalpha[ii][k];
        LUalpha[ii][k] = LUalpha[j][k];
        LUalpha[j][k] = temp;
      }
      temp = LUscale[ii];
      LUscale[ii] = LUscale[j];
      LUscale[j] = temp;
    }

    LUindex[j] = ii;
    if( LUalpha[j][j] == 0.0 )
    {
```

```
      printf( "\n LU singularity at %d\n", j );
      exit(0);
    }
    temp = 1.0 / LUalpha[j][j];
    for( i=j+1; i<n; i++ )
      LUalpha[i][j] *= temp;
  }

  return -1;
}


void terminate_lu( void )
{
  int i;

  if( LUn > 0 )
  {
    free( LUindex );
    free( LUscale );
    free( LUbeta );
    for( i=0; i<LUn; i++ )
      free( LUalpha[i] );
    free( LUalpha );
    LUn = 0;
  }
}
```

List 3. *ABS-LD* Levenberg-Marquardt Algorithm

```
void initiate_marquardt( int m, int n );
void marquardt_stdv( int n, int jn );
int  marquardt( int m, int n, double x[], double parameters[], void
(*func)(), void (*output)() );
void terminate_marquardt( void );



double **MAjacobi, *MAbeta, *MAf, *MAff;
int    *MAindex, MAn=0;



void initiate_marquardt( int m, int n )
{
  int  i;

  if( MAn > 0 )
    terminate_marquardt();
  initiate_lu( n );
  MAn = n;
  MAjacobi = (double **)calloc( n, sizeof(double *) );
  for( i=0; i<n; i++ )
    MAjacobi[i] = (double *)calloc( m, sizeof(double) );
  MAbeta = (double *)calloc( m, sizeof(double) );
  MAf = (double *)calloc( m, sizeof(double) );
  MAff = (double *)calloc( m, sizeof(double) );
  MAindex = (int *)calloc( n, sizeof(int) );
}



void marquardt_stdv( int n, int jn )
{
  int i, j;

  for( i=0; i<jn; i++ )
    for( j=0; j<jn; j++ )
      LUalpha[i][j] = MAjacobi[i][j];
  lu_decompose( jn );

  for( j=0; j<jn; j++ )
  {
    for( i=0; i<jn; i++ )
      LUbeta[i] = 0.0;
    LUbeta[j] = 1.0;
    lu_inverse( jn );
    MAbeta[j] = LUbeta[j];
  }
  j = jn - 1;
  for( i=n-1; i>=0; i-- )
    if( MAindex[i] == 1 )
      MAbeta[i] = sqrt( MAbeta[j--] );
    else
      MAbeta[i] = 0.0;
}
```

```
int marquardt( int m, int n, double x[], double parameters[], void
(*func)(), void (*output)() )
{
  int    i, j, k, jn, result;
  double lmcoef, lmfactor, lmlimit, ssqlimit, ssqpercent;
  double ssqf, ssqff, ssqdif, xj, hu, fZEROeps, oldssqdif;

  if( m < n )
    return -3;

  lmcoef = parameters[0];
  lmfactor = parameters[1];
  lmlimit = parameters[2];
  ssqlimit = parameters[3];
  ssqpercent = parameters[4];
  fZEROeps = pow( DBL_EPSILON, parameters[6] );
  result = 0;

  func( x, MAf, n );
  oldssqdif = ssqf = 0.0;
  for( i=0; i<m; i++ )
    ssqf += MAf[i] * MAf[i];
  if( ssqf < ssqlimit )
    result = 1;

  while( result == 0 )
  {
    for( i=0; i<n; i++ )
      MAindex[i] = 0;
    jn = 0;
    for( j=0; j<n; j++ )
    {
      xj = x[j];
      hu = fabs(xj) * fZEROeps;
      if( hu < DBL_EPSILON )
        hu = fZEROeps;

      x[j] += hu;
      func( x, MAff, j );
      for( k=0; k<m; k++ )
      {
        ssqdif = MAjacobi[jn][k] = (MAff[k] - MAf[k]) / hu;
        if( fabs(ssqdif) > DBL_EPSILON )
          MAindex[j] = 1;
      }
      x[j] = xj;
      if( MAindex[j] == 1 )
        jn ++;
    }
    if( jn == 0 )
    {
      result = -2;
      break;
    }
    for( j=0; j<jn; j++ )
    {
      hu = 0.0;
      for( k=0; k<m; k++ )
        hu += MAjacobi[j][k] * MAf[k];
      LUbeta[j] = MAbeta[j] = hu;
    }
```

```
    for( i=0; i<jn; i++ )
    {
      for( j=i; j<jn; j++ )
      {
        hu = 0.0;
        for( k=0; k<m; k++ )
          hu += MAjacobi[i][k] * MAjacobi[j][k];
        LUalpha[i][j] = hu;
      }
      for( j=0; j<i; j++ )
        LUalpha[i][j] = LUalpha[j][i];
    }
    for( i=0; i<jn; i++ )
      for( j=0; j<jn; j++ )
        MAjacobi[i][j] = LUalpha[i][j];

short_cut:

    for( i=0; i<jn; i++ )
      LUalpha[i][i] *= (1.0 + lmcoef);

    lu_decompose( jn );
    lu_inverse( jn );

    j = jn - 1;
    for( i=n-1; i>=0; i-- )
      if( MAindex[i] == 1 )
        LUbeta[i] = x[i] - LUbeta[j--];
      else
        LUbeta[i] = x[i];

    func( LUbeta, MAff, n );
    ssqff = 0.0;
    for( i=0; i<m; i++ )
      ssqff += MAff[i] * MAff[i];

    ssqdif = ssqf - ssqff;
    if( ssqdif > 0.0 )
    {
      for( i=0; i<n; i++ )
        x[i] = LUbeta[i];
      for( i=0; i<m; i++ )
        MAf[i] = MAff[i];
      marquardt_stdv( n, jn );
      if( output != NULL )
        output( jn, lmcoef, ssqdif, ssqf );
      if( ssqff < ssqlimit )
        result = 1;
      else
        if( (ssqdif/ssqf < ssqpercent)  &&  (ssqdif < oldssqdif) )
          result = 2;
        else
          lmcoef /= lmfactor;
      ssqf = ssqff;
      oldssqdif = ssqdif;
    }
    else
    {
      if( output != NULL )
        output( jn, lmcoef, ssqdif, ssqf );
      lmcoef *= lmfactor;
```

```
      if( lmcoef < lmlimit )
      {
        for( i=0; i<jn; i++ )
        {
          for( j=0; j<jn; j++ )
            LUalpha[i][j] = MAjacobi[i][j];
          LUbeta[i] = MAbeta[i];
        }
        goto short_cut;
      }
      else
        result = -1;
    }
  }
  parameters[0] = lmcoef;
  parameters[5] = ssqf;

  return result;
}


void terminate_marquardt( void )
{
  int i;

  if( MAn > 0 )
  {
    free( MAindex );
    free( MAff );
    free( MAf );
    free( MAbeta );
    for( i=0; i<MAn; i++ )
      free( MAjacobi[i] );
    free( MAjacobi );
    MAn = 0;
    terminate_lu();
  }
}
```

List 4. *ABS-LD* Supporting Functions

```
typedef struct  { double mu, epsilon, sigma, rho, angle;
                  int    _mu_, _epsilon_, _sigma_, _rho_;
                  double beta;
                  double *pia, *pld; }  peak_t;

typedef struct  { peak_t *peak;
                  int    peaks;
                  double alpha, chi, gamma;
                  int    _alpha_, _chi_;
                  double *bia, *bld; }  base_t;

struct  { base_t *base;
          int    points, bases, peaks;
          double *wave, *ia, *ld;
          double ianormal, ldnormal, orientation;
          double *fit_ia, *fit_ld; }  spectrum;


void curve_shape( double mu, double ep, double si, double rho, double
x[], double y[], int points );
void input_data( char *argv[] );
void evaluate_iald( double x[], double f[], int nth );
void output_data( FILE *file, int option );
void adjust_epsilon( void );



void curve_shape( double mu, double ep, double si, double rho, double
x[], double y[], int points )
{
  static double half=1.177410023;       /* sqrt( 2*ln(2) ) */
  static double sqrt2pi=2.506628275;    /* sqrt( 2*M_PI ) */
  double g, r, s, t, z;
  int    i;

  r = 2.0 * si * rho / (rho * rho - 1.0);
  z = log( rho ) / half;
  t = ep / sqrt2pi / z;
  for( i=0; i<points; i++ )
  {
    g = mu + r - x[i];
    if( g > 0.0 )
    {
      s = log( g / r ) / z - z;
      s = 0.5 * s * s;
      y[i] = (s < 700.0) ? t / g / exp(s) : 0.0;
    }
    else
      y[i] = 0.0;
  }
}



double *fit_ia, *fit_ld, *bia, *bld, *pia, *pld;
```

```
void input_data( char *argv[] )
{
  char   restriction[32];
  FILE   *file;
  int    i, j, left, right, points, peaks, bases;
  base_t *bp;
  peak_t *pp;

  file = fopen( argv[1], "rt" );
  if( file == NULL )
  {
    printf( "\n can't open PEAKS file: %s\n", argv[1] );
    exit(0);
  }

  fscanf( file, "%d", &bases );
  spectrum.bases = bases;
  spectrum.peaks = 0;
  spectrum.base = (base_t *)calloc( bases, sizeof(base_t) );
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    fscanf( file, "%d", &peaks );
    bp->peaks = peaks;
    if( bases > 1 )
      fscanf( file, "%lf %lf", &bp->alpha, &bp->chi );
    bp->peak = (peak_t *)calloc( peaks, sizeof(peak_t) );
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      fscanf( file, "%lf %lf %lf %lf %lf", &pp->mu, &pp->epsilon,
&pp->sigma, &pp->rho, &pp->angle );
    }
    spectrum.peaks += peaks;
  }
  fclose( file );

  file = fopen( argv[2], "rt" );
  if( file == NULL )
  {
    printf( "\n can't open SPECTRUM file: %s\n", argv[2] );
    exit(0);
  }

  fscanf( file, "%d %d", &left, &right );
  spectrum.points = points = right - left + 1;
  spectrum.wave = (double *)calloc( points, sizeof(double) );
  spectrum.ia = (double *)calloc( points, sizeof(double) );
  spectrum.ld = (double *)calloc( points, sizeof(double) );
  spectrum.fit_ia = (double *)calloc( points, sizeof(double) );
  spectrum.fit_ld = (double *)calloc( points, sizeof(double) );

  if( bases == 1 )
    for( j=0; j<points; j++ )
      fscanf( file, "%lf", spectrum.ia+j );
  else
    for( j=0; j<points; j++ )
      fscanf( file, "%lf %lf", spectrum.ia+j, spectrum.ld+j );
  fclose( file );

  for( j=0; j<points; j++ )
```

```
    spectrum.wave[j] = left + j;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    bp->bia = (double *)calloc( points, sizeof(double) );
    bp->bld = (double *)calloc( points, sizeof(double) );
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      pp->pia = (double *)calloc( points, sizeof(double) );
      pp->pld = (double *)calloc( points, sizeof(double) );
    }
  }

  fit_ia = (double *)calloc( points, sizeof(double) );
  fit_ld = (double *)calloc( points, sizeof(double) );
  bia = (double *)calloc( points, sizeof(double) );
  bld = (double *)calloc( points, sizeof(double) );
  pia = (double *)calloc( points, sizeof(double) );
  pld = (double *)calloc( points, sizeof(double) );

  printf( "\n restriction file: " );
  gets( restriction );
  if( restriction[0] )
    if( (file = fopen( restriction, "rt" )) != NULL )
    {
      for( i=0; i<bases; i++ )
      {
        bp = &spectrum.base[i];
        peaks = bp->peaks;
        if( bases > 1 )
          fscanf( file, "%d %d", &bp->_alpha_, &bp->_chi_ );
        for( j=0; j<peaks; j++ )
        {
          pp = &bp->peak[j];
          fscanf( file, "%d %d %d %d", &pp->_mu_, &pp->_epsilon_,
&pp->_sigma_, &pp->_rho_ );
        }
      }
      fclose( file );
    }
    else
    {
      printf( "\n can't open RESTRICTION file: %s\n", restriction );
      exit(0);
    }
  else
  {
    for( i=0; i<bases; i++ )
    {
      bp = &spectrum.base[i];
      peaks = bp->peaks;
      if( bases > 1 )
        bp->_alpha_ = bp->_chi_ = 1;
      for( j=0; j<peaks; j++ )
      {
        pp = &bp->peak[j];
        pp->_mu_ = pp->_epsilon_ = pp->_sigma_ = pp->_rho_ = 1;
      }
    }
```

```
    printf( "\n No restriction file, all variables are fitted\n" );
  }
}



int    nvar;
double iabyld;


void evaluate_iald( double x[], double f[], int nth )
{
  static double pideg=0.017453292;      /* pi/180 */
  int    h, i, j, k, points, bases, peaks;
  double y, z, sumxx, sumxy, gamma, beta;
  base_t *bp;
  peak_t *pp;
  int    fbase, fpeak;

  bases = spectrum.bases;
  points = spectrum.points;

  h = 0;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      if( pp->_mu_ )
        pp->mu = fabs( x[h] );
      h++;
      if( pp->_epsilon_ )
        pp->epsilon = fabs( x[h] ) * 1000.0;
      h++;
      if( pp->_sigma_ )
        pp->sigma = fabs( x[h] );
      h++;
      if( pp->_rho_ )
        pp->rho = fabs( x[h] ) + 1.001;
      h++;
      if( nth < h )
      {
        fbase = i;
        fpeak = j;
        nth = 32767;
      }
    }
    if( bases > 1 )
    {
      if( bp->_alpha_ )
        bp->alpha = x[h];
      h++;
      if( bp->_chi_ )
        bp->chi = x[h];
      h++;
      if( nth < h )
      {
```

```
        fbase = i;
        fpeak = -1;
        nth = 32767;
      }
    }
  }

  h = 0;
  for( k=0; k<points; k++ )
    fit_ia[k] = fit_ld[k] = 0.0;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    if( nth == nvar  ||  fbase == i )
    {
      for( k=0; k<points; k++ )
        bia[k] = bld[k] = 0.0;
      if( bases > 1 )
        if( nth == nvar  ||  fpeak == -1 )
        {
          y = sin( bp->alpha * pideg );
          gamma = 3.0 * y * y;
          if( nth == nvar )
            bp->gamma = gamma;
        }
        else
          gamma = bp->gamma;
      for( j=0; j<peaks; j++ )
      {
        pp = &bp->peak[j];
        if( nth == nvar  ||  fpeak == j  ||  fpeak == -1 )
        {
          if( nth == nvar  ||  fpeak == j )
          {
            curve_shape( pp->mu, pp->epsilon, pp->sigma, pp->rho,
spectrum.wave, pia, points );
            if( nth == nvar )
              for( k=0; k<points; k++ )
                pp->pia[k] = pia[k];
          }
          else
            for( k=0; k<points; k++ )
              pia[k] = pp->pia[k];
          if( bases > 1 )
          {
            if( nth == nvar  ||  fpeak == -1 )
            {
              y = sin( (bp->chi - pp->angle) * pideg );
              beta = gamma * y * y - 1.0;
              if( nth == nvar )
                pp->beta = beta;
            }
            else
              beta = pp->beta;
            for( k=0; k<points; k++ )
              pld[k] = beta * pia[k];
            if( nth == nvar )
              for( k=0; k<points; k++ )
                pp->pld[k] = pld[k];
          }
```

```
        }
        else
        {
          for( k=0; k<points; k++ )
            pia[k] = pp->pia[k];
          if( bases > 1 )
            for( k=0; k<points; k++ )
              pld[k] = pp->pld[k];
        }
        for( k=0; k<points; k++ )
          bia[k] += pia[k];
        if( bases > 1 )
          for( k=0; k<points; k++ )
            bld[k] += pld[k];
      }
      if( nth == nvar )
      {
        for( k=0; k<points; k++ )
          bp->bia[k] = bia[k];
        if( bases > 1 )
          for( k=0; k<points; k++ )
            bp->bld[k] = bld[k];
      }
    }
    else
    {
      for( k=0; k<points; k++ )
        bia[k] = bp->bia[k];
      if( bases > 1 )
        for( k=0; k<points; k++ )
          bld[k] = bp->bld[k];
    }
    for( k=0; k<points; k++ )
      fit_ia[k] += bia[k];
    if( bases > 1 )
      for( k=0; k<points; k++ )
        fit_ld[k] += bld[k];
  }
  for( k=0; k<points; k++ )
  {
    z = spectrum.fit_ia[k] = fit_ia[k];
    f[h++] = spectrum.ia[k] - z;
  }
  if( bases > 1 )
  {
    sumxx = sumxy = 0.0;
    for( k=0; k<points; k++ )
    {
      z = fit_ld[k];
      sumxy += z * spectrum.ld[k];
      sumxx += z * z;
    }
    y = spectrum.ldnormal = sumxy / sumxx;
    for( k=0; k<points; k++ )
    {
      z = spectrum.fit_ld[k] = y * fit_ld[k];
      f[h++] = (spectrum.ld[k] - z) / labyld;
    }
  }
}
```

```
void output_data( FILE *file, int packed )
{
  int    i, j, k, pts, bases, points, peaks;
  double iassq, ldssq, z, stdvm, stdve, stdvs, stdvr, stdva, stdvc;
  base_t *bp;
  peak_t *pp;

  bases = spectrum.bases;
  points = spectrum.points;
  peaks = spectrum.peaks;

  pts = 0;
  iassq = ldssq = 0.0;
  for( k=0; k<points; k++ )
  {
    z = MAf[pts++];
    iassq += z * z;
  }
  if( packed )
    fprintf( file, "\n %le", iassq );
  else
  {
    fprintf( file, "\n IAnormal=%lf,\t IAssq=%le\n", spectrum.ianormal,
iassq );
    iassq = sqrt( iassq / (points - 4 * peaks) );
  }

  if( bases > 1 )
  {
    for( k=0; k<points; k++ )
    {
      z = MAf[pts++];
      ldssq += z * z;
    }
    ldssq *= (iabyld * iabyld);
    if( packed )
      fprintf( file, " %le", ldssq );
    else
    {
      fprintf( file, " LDnormal=%lf,\t LDssq=%le\n", spectrum.ldnormal,
ldssq );
      ldssq = sqrt( ldssq / (points - 2 * bases) );
    }
  }

  pts = 0;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      if( packed )
        fprintf( file, "\n %7.2lf %7.0lf %6.2lf %6.3lf %6.2lf", pp->mu,
pp->epsilon, pp->sigma, pp->rho, pp->angle );
      else
      {
        stdvm = MAbeta[pts++] * iassq;
        stdve = MAbeta[pts++] * iassq * 1000.0;
```

```
        stdvs = MAbeta[pts++] * iassq;
        stdvr = MAbeta[pts++] * iassq;
        fprintf( file, "\n %7.2lf (%6.2lf) %7.0lf (%6.0lf) %6.2lf
(%5.2lf) %6.3lf (%5.3lf)", pp->mu, stdvm, pp->epsilon, stdve,
pp->sigma, stdvs, pp->rho, stdvr );
        if( bases > 1 )
          fprintf( file, " %7.4lf", pp->beta*spectrum.ldnormal );
      }
    }
    if( bases > 1 )
      if( packed )
        fprintf( file, "  %7.2lf  %7.2lf", bp->alpha, bp->chi );
      else
      {
        stdva = MAbeta[pts++] * ldssq;
        stdvc = MAbeta[pts++] * ldssq;
        fprintf( file, "\n  %7.2lf (%6.2lf)  %7.2lf (%6.2lf)\n",
bp->alpha, stdva, bp->chi, stdvc );
      }
  }
  fprintf( file, "\n" );
}


FILE *outputfile;
int  option;


#define OP_SIMPLE    0x01
#define OP_DETAIL    0x02
#define OP_FILE      0x04
#define OP_SPECTRUM  0x08
#define OP_PEAKS     0x10
#define OP_RANDOMIZE 0x20
#define OP_PACKED    0x80


void check_marquardt( int jn, double lmcoef, double ssqdif, double ssqf
)
{
  FILE *of;

  of = (option & OP_FILE) ? outputfile : stdout;
  if( ssqdif < 0.0 )
    fprintf( of, " x " );
  else
  {
    if( option & OP_SIMPLE )
      fprintf( of, "\n %d  %le  % le  %le ", jn, lmcoef, ssqdif, ssqf
);
    if( option & OP_DETAIL )
      output_data( of, !OP_PACKED );
  }
}


void adjust_epsilon( void )
```

```
{
  int    i, j, k, bases, peaks, points;
  double x, sumxx, sumxy;
  base_t *bp;
  peak_t *pp;

  bases = spectrum.bases;
  points = spectrum.points;

  for( k=0; k<points; k++ )
    fit_ia[k] = 0.0;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      curve_shape( pp->mu, pp->epsilon, pp->sigma, pp->rho,
spectrum.wave, pia, points );
      for( k=0; k<points; k++ )
        fit_ia[k] += pia[k];
    }
  }
  sumxx = sumxy = 0.0;
  for( k=0; k<points; k++ )
  {
    x = fit_ia[k];
    sumxx += x * x;
    sumxy += x * spectrum.ia[k];
  }
  x = spectrum.ianormal = sumxy / sumxx;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    for( j=0; j<peaks; j++ )
      bp->peak[j].epsilon *= x;
  }
  if( bases > 1 )
  {
    sumxx = sumxy = 0.0;
    for( k=0; k<points; k++ )
    {
      x = spectrum.ld[k];
      sumxx += x * x;
      sumxy += x * spectrum.ia[k];
    }
    x = spectrum.orientation = sumxy / sumxx;
    for( k=0; k<points; k++ )
      spectrum.ld[k] *= x;
  }
}
```

List 5. *ABS-LD* Main Function

```
main( int argc, char *argv[] )
{
  FILE   *file;
  int    mfun, h, i, j, k, result, points, bases, peaks;
  double *varfit, para[7];
  base_t *bp;
  peak_t *pp;
  long   begintime, endtime;
  int    repeat, ranrepeat;
  double ranrange, ranscale, *varsave, *vangle;

  if( argc < 4 )
  {
    printf( "\n need PEAK, SPECTRUM and OUTPUT files\n" );
    exit(0);
  }

  input_data( argv );

  printf( "\n eps, LMcoef, LMstep, LMlimit, SSQlimit, SSQpercent,
SSQia/ld, option:\n" );
  scanf( "%lf %lf %lf %lf %lf %lf %lf %d", para+6, para+0, para+1,
para+2, para+3, para+4, &iabyld, &option );

  points = spectrum.points;
  bases = spectrum.bases;
  peaks = spectrum.peaks;

  mfun = points;
  nvar = 4 * peaks;
  if( bases > 1 )
  {
    mfun += points;
    nvar += 2 * bases;
  }
  varfit = (double *)calloc( nvar, sizeof(double) );
  vangle = (double *)calloc( peaks, sizeof(double) );

  adjust_epsilon();

  h = k = 0;
  for( i=0; i<bases; i++ )
  {
    bp = &spectrum.base[i];
    peaks = bp->peaks;
    for( j=0; j<peaks; j++ )
    {
      pp = &bp->peak[j];
      varfit[h++] = pp->mu;
      varfit[h++] = pp->epsilon / 1000.0;
      varfit[h++] = pp->sigma;
      varfit[h++] = pp->rho - 1.001;
      vangle[k++] = pp->angle;
    }
    if( bases > 1 )
    {
      varfit[h++] = bp->alpha;
      varfit[h++] = bp->chi;
```

```
    }
  }

  outputfile = file = fopen( argv[3], "wt" );
  initiate_marquardt( mfun, nvar );

  for( i=0; i<argc; i++ )
    fprintf( file, " %s ", argv[i] );
  fprintf( file, "\n\n" );
  for( i=0; i<5; i++ )
    fprintf( file, " %lg ", para[i] );
  fprintf( file, " (%d)  [%lg]  M=%d  N=%d  <%lg>\n", option, iabyld,
mfun, nvar, para[6] );
  fprintf( file, "\n orientation ABS/LD = %le\n", spectrum.orientation
);
  iabyld = sqrt( iabyld );

  time( &begintime );
  result = marquardt( mfun, nvar, varfit, para, evaluate_iald,
check_marquardt );
  evaluate_iald( varfit, MAf, nvar );
  time( &endtime );

  fprintf( file, "\n LMcoef=%le,\t SSQ=%le,\t Result=%d\n", para[0],
para[5], result );
  output_data( file, !OP_PACKED );

  if( option & OP_SPECTRUM )
  {
    for( k=0; k<points; k++ )
    {
      fprintf( file, "\n %3.0lf %le %le", spectrum.wave[k],
spectrum.ia[k], spectrum.fit_ia[k] );
      if( bases > 1 )
        fprintf( file, " %le %le", spectrum.ld[k], spectrum.fit_ld[k]
);
    }
    fprintf( file, "\n" );
  }

  if( option & OP_PEAKS )
  {
    fprintf( file, "\n" );
    for( i=0; i<bases; i++ )
    {
      bp = &spectrum.base[i];
      peaks = bp->peaks;
      for( k=0; k<points; k++ )
      {
        for( j=0; j<peaks; j++ )
          fprintf( file, " %le ", bp->peak[j].pia[k] );
        fprintf( file, "\n" );
      }
      if( bases > 1 )
      {
        fprintf( file, "\n" );
        for( k=0; k<points; k++ )
        {
          for( j=0; j<peaks; j++ )
            fprintf( file, " %le ", spectrum.ldnormal *
bp->peak[j].pld[k] );
```

```
          fprintf( file, "\n" );
        }
      }
      fprintf( file, "\n" );
    }
  }
  fclose( file );

  if( option & OP_RANDOMIZE )
  {
    para[3] = para[5];
    para[4] = 0.0;

    printf( "\n\n input RANDOM filename, repeat & %crange: ", 241 );
    scanf( "%s %d %lf", argv[0], &ranrepeat, &ranrange );

    ranscale = 2.0 * ranrange / ((double)RAND_MAX + 1.0);
    srand( (unsigned) time( &begintime ) );
    file = fopen( argv[0], "wt" );
    fprintf( file, "%d %lf %d", ranrepeat, ranrange, bases );
    for( i=0; i<bases; i++ )
      fprintf( file, " %d", spectrum.base[i].peaks );
    fclose( file );
    varsave = (double *)calloc( nvar, sizeof(double) );
    for( i=0; i<nvar; i++ )
      varsave[i] = varfit[i];
    printf( "\n" );

    time( &begintime );
    for( repeat=0; repeat<ranrepeat; repeat++ )
    {
      printf( "%4d", repeat );
      h = 0;
      for( i=0; i<bases; i++ )
      {
        bp = &spectrum.base[i];
        peaks = bp->peaks;
        for( j=0; j<peaks; j++ )
          bp->peak[j].angle = vangle[h++] + ranscale * rand() -
ranrange;
      }
      para[0] = 1024.0;
      para[1] = 2.0;
      para[2] = 1.0e10;
      marquardt( mfun, nvar, varfit, para, evaluate_iald, NULL );
      evaluate_iald( varfit, MAf, nvar );
      file = fopen( argv[0], "at" );
      output_data( file, OP_PACKED );
      fclose( file );
      for( i=0; i<nvar; i++ )
        varfit[i] = varsave[i];
    }
    time( &endtime );
    printf( "\n use time %ld sec\n", endtime-begintime );
  }

  terminate_marquardt();
  return 0;
}
```

List 6. *PARSE-R* Main Function

```
#include <stdio.h>
#include <stdlib.h>


main( int argc, char *argv[] )
{
  FILE   *fin, *fout[4];
  int    repeat, bases, peaks[4], i, j, k;
  double iassq, ldssq, mu, epsilon, sigma, rho, temp, alpha, chi;

  if( argc < 3 )
  {
    printf( "\n need RANDOM and OUTPUT files\n" );
    exit(0);
  }
  fin = fopen( argv[1], "rt" );
  fscanf( fin, "%d %lf %d", &repeat, &temp, &bases );

  if( argc != bases+2 )
    printf( "\n need %d OUT files\n", bases );
  else
  {
    for( i=0; i<bases; i++ )
    {
      fscanf( fin, "%d", peaks+i );
      fout[i] = fopen( argv[i+2], "wt" );
    }
    for( i=0; i<repeat; i++ )
    {
      fscanf( fin, "%lf %lf", &iassq, &ldssq );
      fprintf( fout[0], "%lg,%lg,", iassq, ldssq );
      for( j=0; j<bases; j++ )
      {
        for( k=0; k<peaks[j]; k++ )
        {
          fscanf( fin, "%lf %lf %lf %lf %lf", &mu, &epsilon, &sigma,
&rho, &temp );
          fprintf( fout[j], "%lg,%lg,%lg,%lg,", mu, epsilon, sigma, rho
);
        }
        fscanf( fin, "%lf %lf", &alpha, &chi );
        fprintf( fout[j], "%lg,%lg\n", alpha, chi );
      }
    }
    for( i=0; i<bases; i++ )
      fclose( fout[i] );
  }
  fclose( fin );
}
```