Supporting Information for

Simplification in the Acquisition and Analysis of Fluorescence Decays Acquired with Polarized Emission for Time-Resolved Fluorescence Anisotropy Measurements

Hunter Little,[1] Jingqi Wang,[1] Jean Duhamel,[1]*

Xuesong Li,[2] Nagula Markandeya,[2] Victor Maurizot,[2] Ivan Huc[2,3]

[1] Institute for Polymer Research, Waterloo Institute for Nanotechnology, Department of Chemistry, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada

[2] Université de Bordeaux, CNRS, Bordeaux Institut National Polytechnique, CBMN (UMR 5248), Institut Europeen de Chimie Biologie, 2 Rue Escarpit, 33600 Pessac, France

[3] Department Pharmazie, Ludwig-Maximilians-Universität München Butenandtstraße 5 - 13 D-81377 Munich, Germany

* To Whom correspondence should be addressed

Jean Duhamel: jduhamel@uwaterloo.ca

Victor Maurizot: victor.maurizot@u-bordeaux.fr

Ivan Huc : ivan.huc@cup.lmu.de

Table of Contents

A] Fit of the experimental fluorescence decays

The $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays acquired with toggling and static polarizers were fitted first, by using an in-house global analysis program that optimized the *G*-factor and second, according to the HORIBA analysis protocol, that measured the *G*-factor from the $I_{HV}(t)$ and $I_{HH}(t)$ decays before constructing and fitting the sum $I_S(t) = I_{VV}(t) + 2\times GI_{VH}(t)$ to characterize the natural fluorescence decay of the OPV-labeled foldamers and difference $I_D(t) = I_{VV}(t) - GI_{VH}(t)$ to obtain the ϕ and r_o parameters. Figures S1 and S2 provide an example of the fits that were obtained through global analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays acquired with toggling and static polarizers, respectively. As illustrated in Figures S1 and S2, the fits conducted by globally analyzing the $I_{VV}(t)$ and $I_{VH}(t)$ decays acquired with static or toggling polarizers were excellent. The main difference between the two sets of fluorescence decays was that the $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays had 10,000 counts at their decay maximum in Figures S1A and S1B whereas the $I_{VV}(t)$ decay had 10,000 more counts at its maximum in Figure S2A than $I_{VH}(t)$ at its decay maximum in Figure S2B. The difference in count number between the $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays in Figures S2A and S2B was a consequence of the standard acquisition procedure which required that the decays be obtained over a same excitation period. Analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays acquired with toggling polarizers through the standard method required determining the *G*-factor from the decay analysis of the $I_{HV}(t)$ and $I_{HH}(t)$ decays before calculating and analyzing the sum $I_S(t) = I_{VV}(t) + 2\times GI_{VH}(t)$ and difference $I_D(t) = I_{VV}(t) - GI_{VH}(t)$. The fits of the decay sum and difference were excellent as shown in Figures S3A and S3B, respectively.

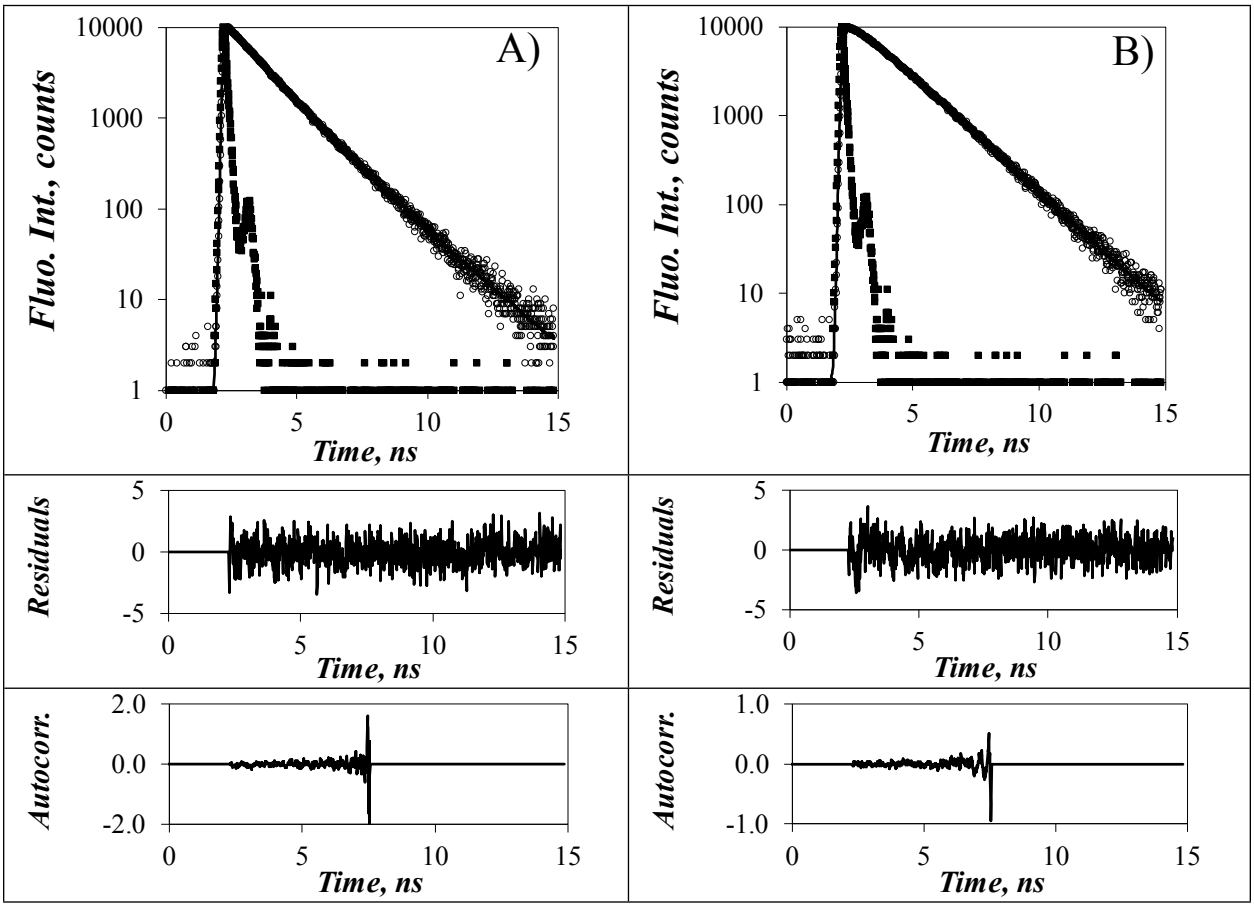

Figure S1. Global analysis of the A) $I_{VV}(t)$ and B) $I_{VH}(t)$ fluorescence decays acquired with static polarizers ($\chi^2 = 1.18$). [OPV-Q_{33}] = 1.4×10^{-5} M in chloroform.

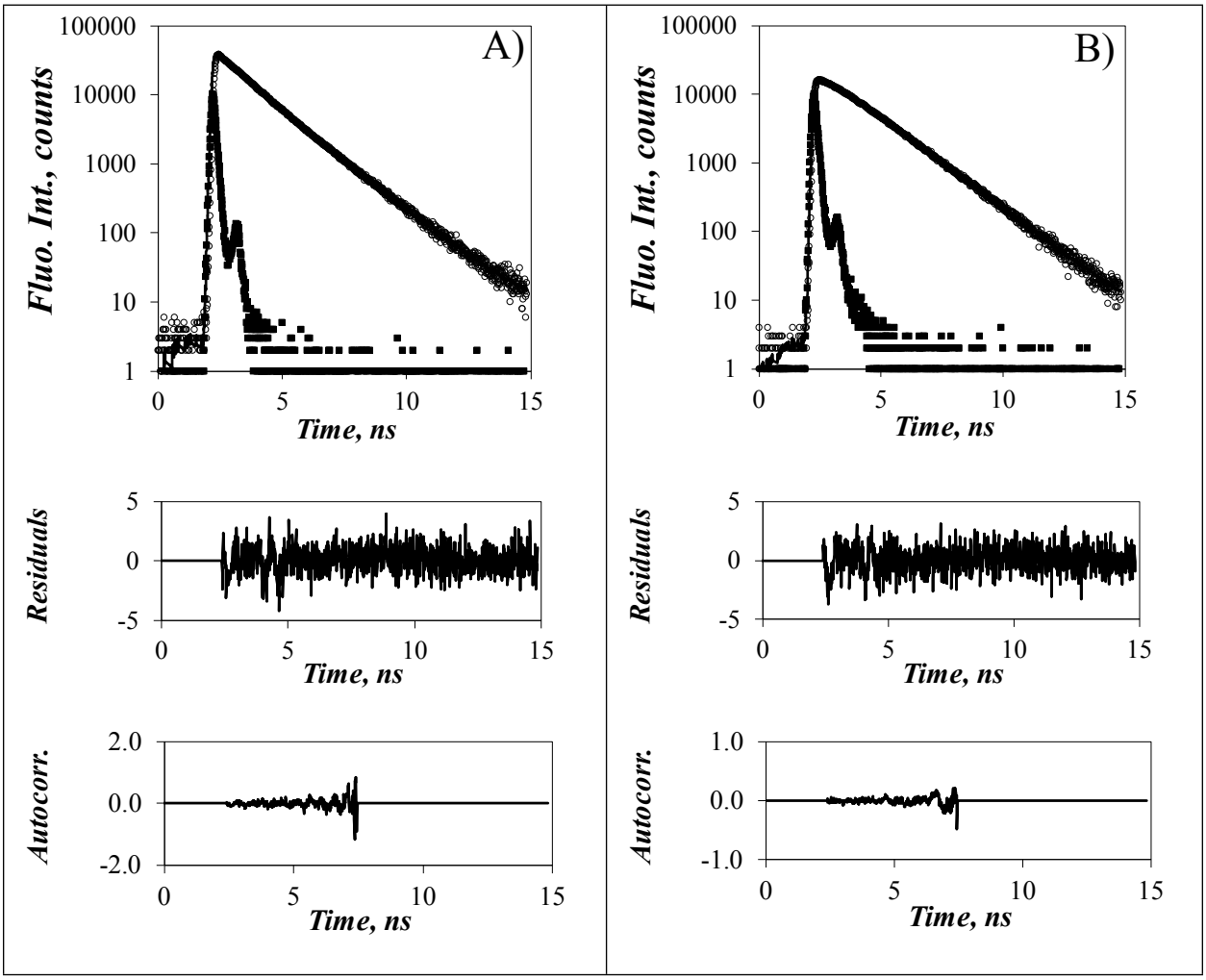

Figure S2. Global analysis of the A) $I_{VV}(t)$ and B) $I_{VH}(t)$ fluorescence decays acquired with toggling polarizers ($\chi^2 = 1.22$). [OPV-Q_{33}] = 1.4×10^{-5} M in chloroform.

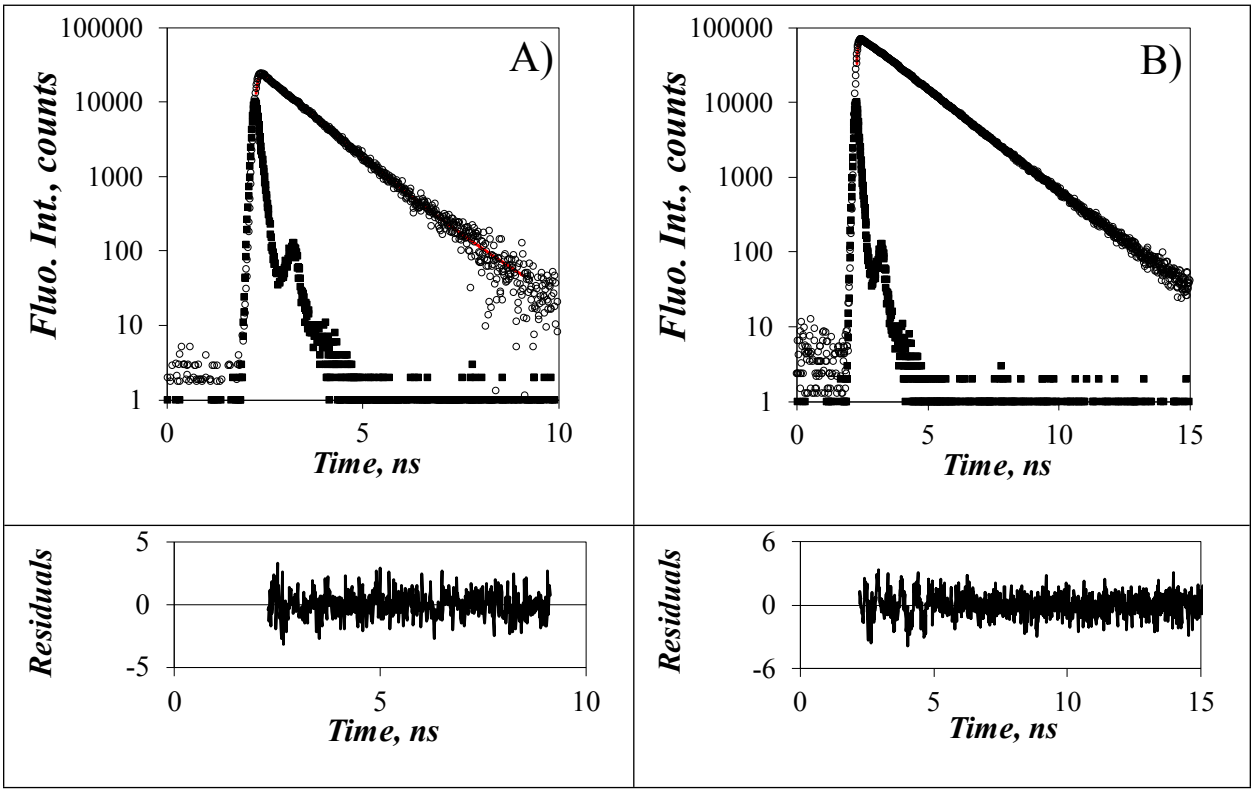

Figure S3. A) Sum $I_{HV}(t) + 2\times GI_{HH}(t)$ fitted with a single exponential (τ_o = 1.60 ns, χ^2 = 1.10) and B) difference $I_{HV}(t) - GI_{HH}(t)$ fitted with a biexponential where the $I_{VV}(t)$ and $I_{VH}(t)$ decays were acquired with toggling polarizers and fitted with the HORIBA programs. [OPV-Q_{33}] = 1.4×10^{-5} M in chloroform.

B] Simulations

In order to assess the ability of the proposed method to retrieve the rotational times for more complex macromolecules associated with multiexponential TRFAs such as symmetric top macromolecular objects (STMOs), $I_{VV}(t)$ and $I_{VH}(t)$ decays were simulated by convoluting an IRF with Equations 4 and 5 and assuming a TRFA that was either a mono-, bi-, or tri-exponential according to Equation S1 that was derived by Duhamel et al.[1] Equation S1 was obtained for a dye bound to STMOs that can wobble infinitely quickly between two reflecting barriers separated by an angle l. The (μ_A) absorption and (μ_E) emission dipole moments of the dye form an angle β_A and β_E with the main axis of the STMO and ξ represents the angle between the projection of μ_A and μ_E in the plane perpendicular to the main axis of the STMO. In Equation S1, rotational diffusion parallel and perpendicular to the main axis of the STMO is handled by the diffusion coefficients $D_{//}$ and $D_{\perp}$, respectively.

$$r(t) = 0.3\sin^2(\beta_A)\sin^2(\beta_E)\cos(2\xi)\times\frac{\sin^2 l}{l^2}\times\exp[-(4D_{//}+2D_{\perp})t]$$

$$+0.3\sin(2\beta_A)\sin(2\beta_E)\cos(\xi)\times\frac{\sin^2(l/2)}{(l/2)^2}\times\exp[-(D_{//}+5D_{\perp})t)$$

$$+0.1\times[3\cos^2(\beta_A)-1]\times[3\cos^2(\beta_E)-1]\times\exp[-6D_{\perp}t] \qquad \text{(S1)}$$

The initial intensity I_o in Equations 4 and 5 was adjusted so that the convolution of the IRF with Equation 4 would yield 10,000 (for a monoexponential TRFA) or 20,000 (for a bi- and triexponential TRFA) counts at the $I_{VV}(t)$ decay maximum. The same I_o value was then used to convolute the IRF with Equation 5 to yield the $I_{VH}(t)$ decay obtained for five different G-factors equal to 0.5, 0.75, 1.0, 1.5, and 2.0. Twenty different patterns of Poisson noise were added to each

convolution products to obtain the simulated $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays obtained for each STMO considered representing a total of 100 $I_{VV}(t)$ and 100 $I_{VH}(t)$ fluorescence decays. The decays were then fitted globally with the programs *aniso01c* for a monoexponential TRFA and *aniso02n-4* for a bi- and tri-exponential TRFA. As had been done earlier,[Error! Bookmark not defined.,Error! Bookmark not defined.] the program *aniso02n-4* assumed that the β_A, β_E, and ξ angles describing the orientation of the dye with respect to the molecular frame of the STMO were known from X-ray crystallography or molecular mechanics optimizations. The wobbling angle l was optimized by approximating the functions $sin^2 l/l^2$ and $sin(l/2)^2/(l/2)^2$ in Equation 6 as $1 - l^2/3$ and $1 - l^2/12$. These approximations were found to hold to within less than 10% error up to $l = 60^o$. Furthermore, *aniso02n-4* optimized the two diffusion coefficients $D_{//}$ and $D_\perp$ instead of the three rotational times $\phi_1 = (4D_{//}+2D_\perp)^{-1}$, $\phi_2 = (D_{//}+5D_\perp)^{-1}$, and $\phi_3 = (6D_\perp)^{-1}$, thus reducing the number of floating parameters. The $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays simulated with a bi- and tri-exponential TRFA were also fitted with the programs *aniso02d* and *aniso03g* which optimized the pre-exponential factors in the TRFA expression as free floating parameters.

Monoexponential TRFA: Setting $\beta_A = \beta_E = \xi = 0$ and $D = D_{//} = D_\perp = (6\phi)^{-1}$ in Equation S1 returned Equation 3 with $r_o = 0.4$ for a dye rigidly bound to a spherical object such as the OPV-Q_n foldamers shown in Figure 1.[2] The parameters $\tau_o = 1.6$ ns, $r_o = 0.4$, and ϕ values obtained from Equation S2 for the OPV-Q_n samples were then employed to simulate $I_{VV}(t)$ and $I_{VH}(t)$ decays with Equations 4 and 5, respectively.

$$\phi = 0.0598 \times NU + 0.3400 \qquad \text{(in ns)} \qquad \text{(S2)}$$

Biexponential TRFA: An earlier study showed that the TRFA of ethidium bromide (EB) intercalated between base pairs perpendicularly to the main axis of a DNA duplex could be handled

with the biexponential TRFA given in Equation S3 by setting $\beta_A = \beta_E = 90^o$, $\xi = 0$, and a wobbling angle $l = 35^o$ in Equation S1, and taking the $D_{//}$ and $D_\perp$ coefficients from Table S1.[1] The lifetime (τ_o) of EB intercalated between DNA base pairs was set to equal 23 ns. These parameters were used to simulate the $I_{VV}(t)$ and $I_{VH}(t)$ decays with Equations 4 and 5 for seven DNA duplexes, respectively.

$$r(t) = 0.254\times\exp(-t/\phi_1) + 0.100\times\exp(-t/\phi_3) \quad \text{(S3)}$$

Triexponential TRFA: It was generated by using the same parameters as for the biexponential TRFA except for the angles β_A and β_E which were set to equal 75°, resulting in Equation S4 which was used to generate the simulated $I_{VV}(t)$ and $I_{VH}(t)$ decays with Equations 4 and 5, respectively.

$$r(t) = 0.230\times\exp(-t/\phi_1) + 0.073\times\exp(-t/\phi_2) + 0.064\times\exp(-t/\phi_3) \quad \text{(S4)}$$

Table S1. $D_{//}$ and $D_\perp$ coefficients used for DNA duplexes.**Error! Bookmark not defined.**

Number of base of pairs	$D_{//}$ (μs^{-1})	$D_\perp$ (μs^{-1})	$\phi_1 = (4D_{//}+2D_\perp)^{-1}$ (ns)	$\phi_2 = (D_{//}+5D_\perp)^{-1}$ (ns)	$\phi_3 = (6D_\perp)^{-1}$ (ns)
11	62.1	37.0	3.1	4.0	4.5
16	54.7	18.7	3.9	6.7	8.9
17	48.4	14.4	4.5	8.3	11.6
19	47.9	12.8	4.6	8.9	13.0
28	33.8	5.9	6.8	15.8	28.1
29	35.3	6.3	6.5	14.9	26.3
32	27.8	4.7	8.3	19.4	35.2

C] Fit of the simulated fluorescence decays with aniso02n-4 using a triexponential TRFA

The $I_{VV}(t)$ and $I_{VH}(t)$ decays were simulated according to Equations 4 and 5, respectively, assuming the triexponential TRFA given in Equation S4 and the parameters provided in the Experimental section. As evident from the plots shown in Figure S4, the parameters $D_{//}$, $D_{\perp}$, l, and G obtained from the global analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ decays with *aniso02n-4* agreed perfectly with those used to simulate the decays. This result confirms the robustness of the procedure that optimizes the G-factor as a mere normalization parameter.

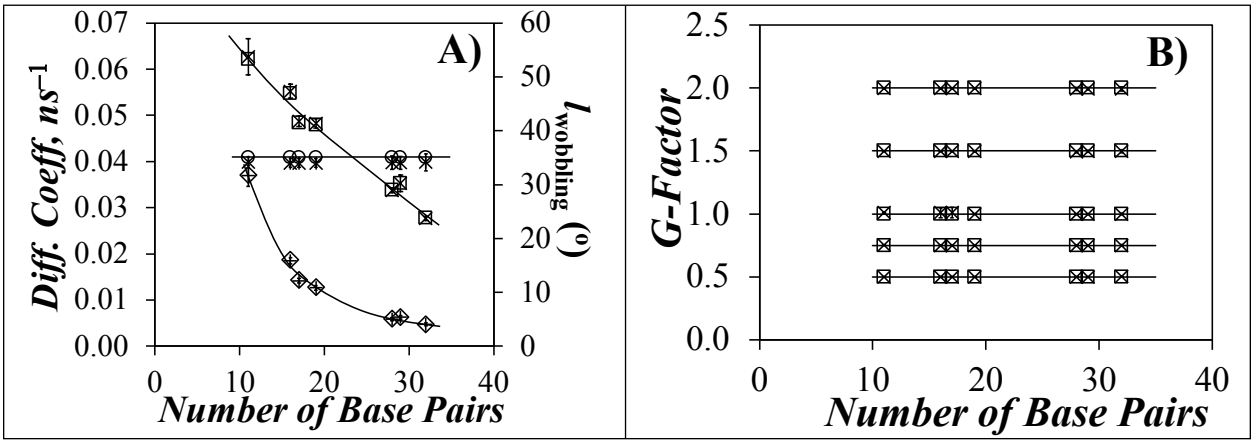

Figure S4. Plots of the inputted (×,+,∗) and calculated (□,◇,○) values of A) (□,×) $D_{//}$, (◇,+) $D_{\perp}$, and (○,*) r_o, and C) G-factor plotted as a function of the number of DNA base pairs.

D] Fit of the simulated fluorescence decays with aniso02d using a biexponential TRFA

The fluorescence decays simulated with Equation S3 were fitted with the program *aniso02d* according to the TRFA given in Equation S5 where the pre-exponential factors r_1 and r_2 and rotational times ϕ_1 and ϕ_2 were optimized along with the G-factor. Consequently, *aniso02d* used four independent parameters to handle the TRFA compared to three ($D_{//}$, $D_{\perp}$, l) for *aniso02n-4*.

$$r(t) = r_1 \times \exp(-t/\phi_1) + r_2 \times \exp(-t/\phi_2) \quad \text{(S5)}$$

The main effect of using more floating parameters with *aniso02d* was that the parameters r_1, r_2, and ϕ_1 retrieved from the global analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ fluorescence decays showed a bit more scatter in Figure S5A and B compared to those retrieved with *aniso02n-4* in Figure 4. Interestingly, when the number average rotational time $<\phi>$ ($=[r_1\phi_1+r_2\phi_2]/[r_1+r_2]$) and initial anisotropy r_o ($= r_1 + r_2$) were plotted as a function of the number of DNA base pairs in Figure S5C, a good agreement between the inputted and retrieved parameters was obtained. This outcome reflects the fact that the TRFA can be satisfyingly fitted with a larger set of floating parameters, but that the area under the curve defined by the TRFA, that is solely dependent on r_o and $<\phi>$, is retrieved accurately. As Figure S5D makes clear, the retrieved *G*-factor matches the inputted *G*-factors within minuscule error bars.

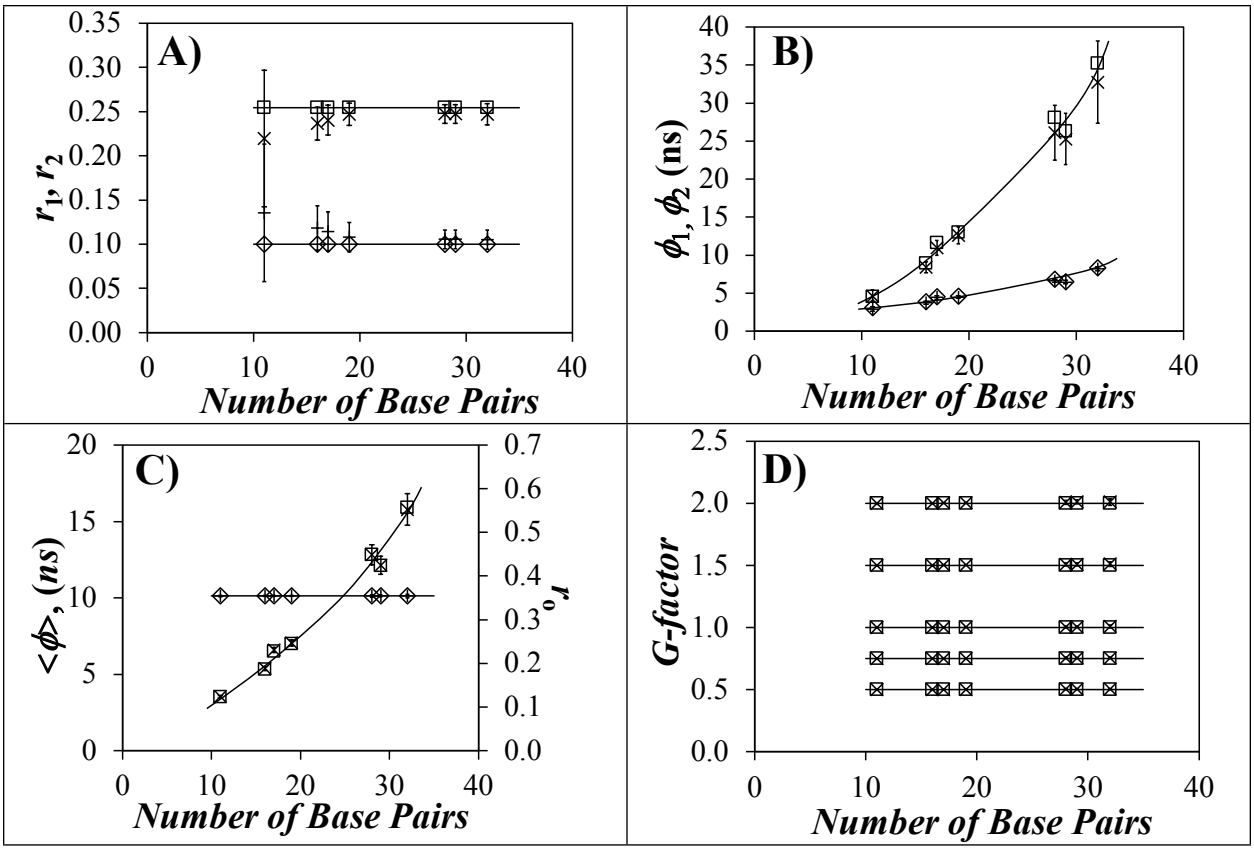

Figure S5. Plots of the (□, ◇) inputted and (×,+) calculated values of A) (□,×) r_1 and (◇,+) r_2, B) (□,×) ϕ_1 and (◇,+) ϕ_2, C) (□,×) $\langle\phi\rangle$ and (◇,+) r_o, and D) *G*-factor plotted as a function of the number of DNA base pairs. Biexponential TRFA obtained as described in the Experimental section.

E] Fit of the simulated fluorescence decays with aniso03g using a triexponential TRFA

The $I_{VV}(t)$ and $I_{VH}(t)$ decays simulated with the triexponential function given in Equation S4 were fitted globally with the program *aniso03g* according to Equation S6. The pre-exponential factors (r_i) and diffusion coefficients ($D_{//}$ and $D_{\perp}$) were allowed to float. The inputted and retrieved parameters are compared in Figures S6A – D. In this case, differences between inputted and retrieved parameters were more pronounced for the pre-exponential factors reflecting the fact that increasing the number of floating parameters decreases the accuracy in their retrieved value.

$$r(t) = r_1 \times \exp[-(4D_{//}+2D_{\perp})t] + r_2 \times \exp[-(D_{//}+5D_{\perp})t] + r_3 \times \exp[-(6D_{\perp})t] \quad \text{(S6)}$$

A much better agreement was obtained between the inputted and retrieved initial anisotropy (r_o) and number average rotational time ($<\phi>$) in Figure S6E. This result is expected since the parameters r_o and $<\phi>$ define the envelop of the TRFA which is well-represented by different sets of r_i, $D_{//}$, and $D_{\perp}$ values in Equation S6. Thus, the smaller accuracy obtained for the individual parameters does not affect r_o and $<\phi>$ which are obtained with good accuracy based on Figure S6E. Finally, the *G*-factors that were used to build the $I_{VV}(t)$ and $I_{VH}(t)$ decays and retrieved with *aniso03g* were compared in Figure S6F. The agreement between inputted and retrieved *G*-factors was excellent demonstrating that optimization of the *G*-factor in the global analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ decays is a robust procedure.

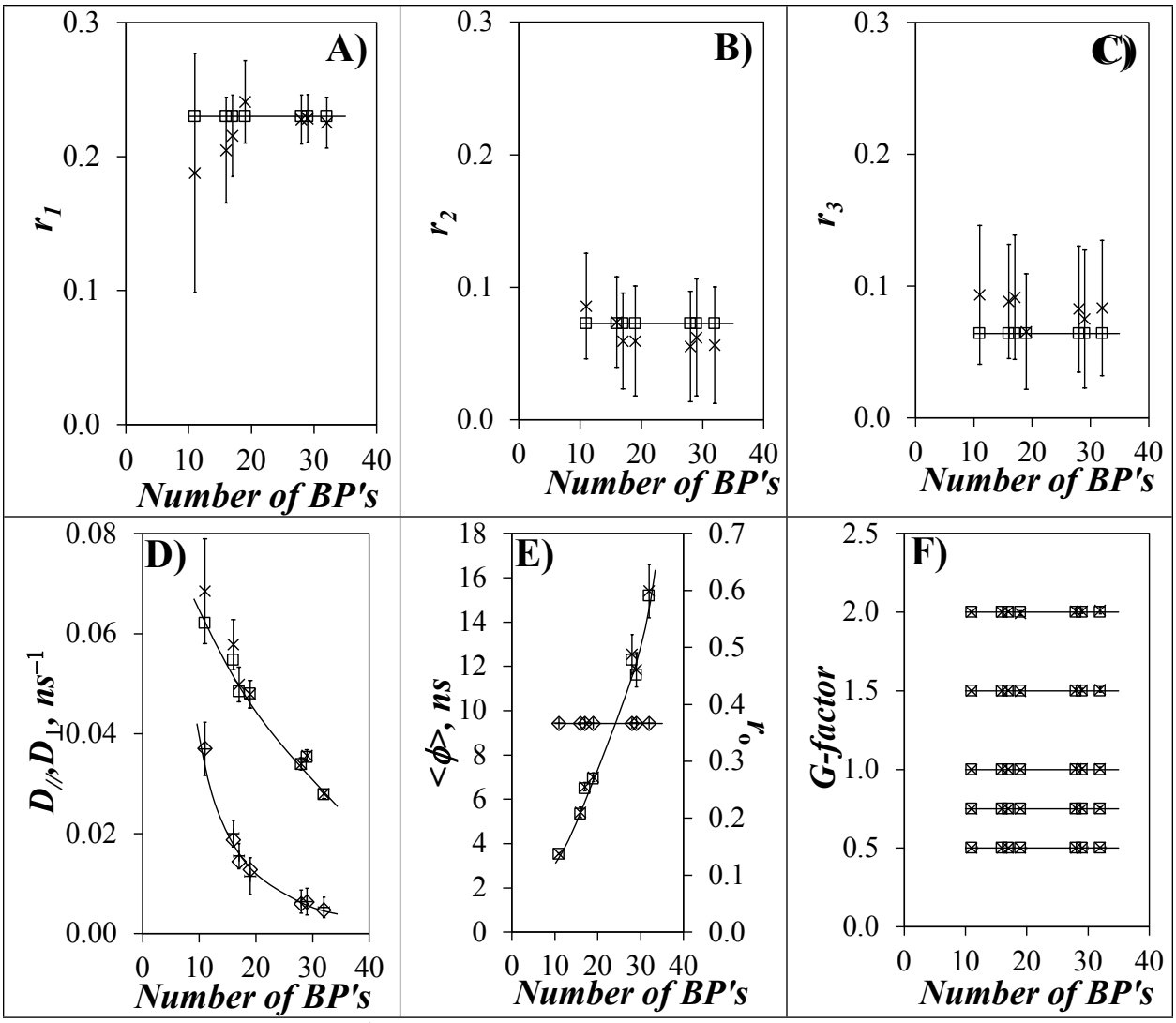

Figure S6. Plots of the (□,◇) inputted and (×,+) calculated values of A) r_1, B) r_2, C) r_3, D) (□,×) $D_{//}$ and (◇,+) $D_\perp$, E) (□,×) <ϕ> and (◇,+) r_o, and D) *G*-factor plotted as a function of the number of DNA base pairs. Triexponential TRFA obtained as described in the Experimental section.

F] Parameters obtained from the experimental fluorescence decay analysis.

Table S2. Parameters r_o, ϕ, and G retrieved from the analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ decays acquired with toggling polarizers until the difference between the maxima of the $I_{VV}(t)$ and $I_{VH}(t)$ decays reached 10,000 counts and fitted according to the protocol provided by HORIBA assuming that the anisotropy is a monoexponential (aniso01c). Solutions of 1.4×10^{-5} M OPV-Q_n constructs in chloroform.

Sample	r_0	ϕ (ns)	τ_o* (ns)	G	χ^2
OPV-Q_4	0.40	0.55	1.71	0.84	1.11
	0.38	0.50	1.72	0.88	1.21
	0.40	0.57	1.69	0.86	1.01
OPV-Q_7	0.37	0.73	1.58	0.89	0.98
	0.37	0.73	1.57	0.89	1.18
	0.43	0.72	1.58	0.89	0.93
OPV-Q_{17}	0.35	1.28	1.59	0.86	1.09
	0.36	1.27	1.57	0.78	1.12
	0.36	1.26	1.58	0.90	1.25
OPV-Q_{24}	0.35	1.74	1.59	0.90	1.25
	0.35	1.69	1.59	0.90	1.04
	0.35	1.70	1.62	0.90	0.96
OPV-Q_{33}	0.35	2.12	1.57	0.91	1.15
	0.35	2.09	1.60	0.91	1.10
	0.35	2.05	1.60	0.91	1.15

* τ_o was determined from the monoexponential analysis of the $I_{VM}(t)$ fluorescence decays of the OPV-Q_n solutions in chloroform.

Table S3. Parameters r_o, ϕ, and G retrieved from the analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ decays acquired with toggling polarizers until the difference between the maxima of the $I_{VV}(t)$ and $I_{VH}(t)$ decays reached 10,000 counts and fitted globally assuming that the anisotropy is a monoexponential (aniso01c). Solutions of 1.4×10^{-5} M OPV-Q_n constructs in chloroform. The G-factor was optimized in the analysis.

Sample	r_0	ϕ (ns)	τ_o* (ns)	G	χ^2
OPV-Q_4	0.42	0.58	1.71	0.84	1.19
	0.37	0.60	1.72	0.86	1.15
	0.38	0.58	1.69	1.02	1.29
OPV-Q_7	0.41	0.76	1.58	0.98	1.27
	0.38	0.75	1.57	0.89	1.28
	0.40	0.75	1.58	0.89	1.25
OPV-Q_{17}	0.37	1.32	1.59	0.83	1.25
	0.36	1.32	1.57	0.90	1.26
	0.36	1.32	1.58	0.91	1.25
OPV-Q_{24}	0.34	1.77	1.59	0.91	1.28
	0.34	1.70	1.59	0.92	1.24
	0.34	1.74	1.62	0.92	1.21
OPV-Q_{33}	0.32	2.25	1.57	0.97	1.27
	0.35	2.45	1.60	0.93	1.22
	0.32	2.23	1.60	0.95	1.29

* τ_o was determined from the monoexponential analysis of the $I_{VM}(t)$ fluorescence decays of the OPV-Q_n solutions in chloroform.

Table S4. Parameters r_o, ϕ, and G retrieved from the analysis of the $I_{VV}(t)$ and $I_{VH}(t)$ decays acquired with static polarizers until both decays reached 10,000 counts at the decay maximum and fitted globally assuming that the anisotropy is a monoexponential (aniso01c). Solutions of 1.4×10^{-5} M OPV-Q_n constructs in chloroform. The G-factor was optimized in the analysis.

Sample	r_0	ϕ (ns)	τ_o* (ns)	G	χ^2
OPV-Q_4	0.39	0.57	1.71	0.45	1.17
	0.42	0.59	1.72	0.42	1.20
	0.39	0.57	1.69	0.43	1.24
OPV-Q_7	0.41	0.76	1.58	0.42	1.24
	0.40	0.74	1.57	0.43	1.28
	0.40	0.75	1.58	0.43	1.27
OPV-Q_{17}	0.34	1.24	1.59	0.42	1.21
	0.36	1.28	1.57	0.41	1.28
	0.36	1.34	1.58	0.40	1.26
OPV-Q_{24}	0.34	1.78	1.59	0.40	1.27
	0.34	1.68	1.59	0.41	1.28
	0.34	1.66	1.62	0.41	1.24
OPV-Q_{33}	0.33	2.28	1.57	0.43	1.28
	0.33	2.36	1.60	0.41	1.25
	0.32	2.13	1.60	0.41	1.18

* τ_o was determined from the monoexponential analysis of the $I_{VM}(t)$ fluorescence decays of the OPV-Q_n solutions in chloroform.

G] REFERENCES

1. Duhamel, J.; Kanyo, J.; Dinter-Gottlieb, G.; Lu, P. Fluorescence Emission of EB intercalated in Defined DNA Duplexes: Hydrodynamics and Base Pair Twisting Components. *Biochemistry* **1996**, *35*, 16687-16697.
2. Wang, J.; Little, H.; Duhamel, J.; Li, X.; Nagula, M.; Maurizot, V.; Huc, I. Application of Time-Resolved Fluorescence Anisotropy to Probe Quinoline-Based Foldamers Labeled with Oligo(phenylene-vinylene). *Macromolecules* **2019**, *52*, 5829-5837.

H] Listing of *aniso01c*

```
c     Program created April 9, 2015

c     This program fits the I(para;para) and I(para;per) decays
c       globally assuming a monoexponential anisotropy.
c       There is NO function for non-covalently attached dyes.
c       There is a function for covalently attached dyes.
c       The G-factor is calculated in the analysis.

      implicit double precision (a-h,o-z)

      parameter (mfit=8,ma=8,nca=8)

      dimension y1(10000),lista(mfit),a(mfit),
     & covar(mfit,mfit),alpha(mfit,mfit),
     & e1(10000),da(mfit),beta(mfit)
     & ,dyda(mfit),ym(10000),res1(10000),auto1(10000)
     & ,e2(10000),y2(10000),res2(10000),auto2(10000)
     & ,nstart1(3),nstart2(3)
     & ,a1(10),a2(10),tau1(10),tau2(10),f1pa(10000),f1pe(10000)

      real*8 taum,chisq,tpc1,tpc2


10    format(A)
      character *30 DD1,DD2

      write (*,*) 'How many chanels do you want to work with
     & for the VV decay?'
      read(*,*) ndata1

c     write (*,*) 'What is the name of your lamp file ?'
c     read(*,10) DD

c     open (1,file=DD,status='old')

c     read(1,*) (e(i),i=1,ndata)

c     close(1)

c     write (*,*) 'what is the name of your fluorescence decay?'
c     read(*,10) DD

c     open(1,file=DD,status='old')

c     read(1,*) (y(i),i=1,ndata)

c     close(1)

c     write (*,*) 'What is your file''s name for VV?'
c     read(*,10) DD1
```

```
c     open(1,file=DD1,status='old')

c     do 15 i=1,ndata1

c     read(1,*) e1(i),y1(i)

c15   continue

c     close (1)

      write (*,*) 'What is your lamp''s filename for
     & the VV decay?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 13 i=1,9
13    read(1,*)

      do 15 i=1,ndata1

      read(1,*) x,e1(i)

15    continue

      close (1)


      write (*,*) 'What is your VV decay''s filename?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 14 i=1,9
14    read(1,*)

      do 16 i=1,ndata1

      read(1,*) x,y1(i)

16    continue

      close (1)

      write (*,*) 'How many chanels do you want to work with
     & for the VH decay?'
      read (*,*) ndata2


c     write (*,*) 'What is your file''s name for VH?'
c     read(*,10) DD2

c     open(1,file=DD2,status='old')
```

```
c     do 16 i=1,ndata2

c     read(1,*) e2(i),y2(i)

c16   continue

c     close (1)


      write (*,*) 'What is your lamp''s filename for
     & the VH decay?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 313 i=1,9
313   read(1,*)

      do 315 i=1,ndata2

      read(1,*) x,e2(i)

315   continue

      close (1)


      write (*,*) 'What is your VH decay''s filename?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 314 i=1,9
314   read(1,*)

      do 316 i=1,ndata2

      read(1,*) x,y2(i)

316   continue

      close (1)

      write (*,*) 'What is the lifetime of the reference compound
     & for the VV decay?'
      read(*,*) taur1

      write (*,*) 'What is your time per chanel for the VV
     & decay?'
      read(*,*) tpc1

      taur1 = exp(-tpc1/taur1)
```

```
      alold2 = e1(1)

      do 40 i=2,ndata1

      alold1 = e1(i)
      e1(i) = e1(i) - alold2*taur1
      alold2 = alold1

      if(e1(i).lt.0) e1(i)=0.0001

40    continue


      write (*,*) 'What is the lifetime of the reference compound
     & for the VH decay?'
      read(*,*) taur2

      write (*,*) 'What is your time per chanel for the VH
     & decay?'
      read(*,*) tpc2

      taur2 = exp(-tpc2/taur2)

      alold2 = e2(1)

      do 41 i=2,ndata2

      alold1 = e2(i)
      e2(i) = e2(i) - alold2*taur2
      alold2 = alold1

      if(e2(i).lt.0) e2(i)=0.0001

41    continue

      imax = 0

      do 500 i=2,ndata1

      if (e1(i).gt.emax) then
      imax = i
      emax = e1(i)
      endif

500   continue


      write (*,511)
      write (*,*) '_____________________________________________'
511   format(10H   channel,10H      lamp,10H     decay)

      do 520 k=imax-15,imax+10
```

```
      if (k.gt.1) then
      write (*,512) k,e1(k),y1(k)
512   format (3H   ,I4,3H   ,2(2H  ,F7.1,1H ))
      endif
520   continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VV decay?'
      read(*,*) nstart1(3)

      write (*,*) 'At which channel does the background noise start
     & for the VV decay?'
      read(*,*) nstart1(1)

      write (*,*) 'At which channel does the background noise end
     & for the VV decay?'
      read(*,*) nstart1(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 860 i=nback1,nback2

c     sume = sume + e1(i)
c     sumy = sumy + y1(i)

c860  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 870 i=1,ndata1

c     e1(i) = e1(i) - sume
c     y1(i) = y1(i) - sumy

c     if(e1(i).lt.0.0) e1(i) = 0.00001
c     if(y1(i).lt.0.0) y1(i) = 0.00001

c870  continue


      imax = 0
      emax = 0.0

      do 501 i=2,ndata2

      if (e2(i).gt.emax) then
      imax = i
      emax = e2(i)
```

```
      endif

501   continue


      write (*,511)
      write (*,*) '_____________________________________________________'

      do 1522 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e2(k),y2(k)
      endif
1522  continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VH decay?'
      read(*,*) nstart2(3)

      write (*,*) 'At which channel does the background noise start
     & for the VH decay?'
      read(*,*) nstart2(1)

      write (*,*) 'At which channel does the background noise end
     & for the VH decay?'
      read(*,*) nstart2(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 861 i=nback1,nback2

c     sume = sume + e2(i)
c     sumy = sumy + y2(i)

c861  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 871 i=1,ndata2

c     e2(i) = e2(i) - sume
c     y2(i) = y2(i) - sumy

c     if(e2(i).lt.0.0) e2(i) = 0.00001
c     if(y2(i).lt.0.0) y2(i) = 0.00001

c871  continue

        write (*,*) 'How many decay times do you need to represent
```

```
     & the decay of the dye?'
        read (*,*) nexp1

        tauav1 = 0.0

        do 400 k1=1,nexp1
        write (*,*) 'What is the',k1,'th decaytime?'
        read (*,*) tau1(k1)
        write (*,*) 'What is its normalized pre-exponential factor?'
        read (*,*) a1(k1)

        tauav1 = tauav1 + a1(k1)*tau1(k1)

400     continue

c        write (*,*) 'How many decay times do you need to represent
c     & the decay of the dye?'
c        read (*,*) nexp2

c        tauav2 = 0.0

c        do 405 k1=1,nexp2
c        write (*,*) 'What is the',k1,'th decaytime?'
c        read (*,*) tau2(k1)
c        write (*,*) 'What is its normalized pre-exponential factor?'
c        read (*,*) a2(k1)

c        tauav2 = tauav2 + a2(k1)*tau2(k1)

c405     continue

        do 410 i1=1,10000

        f1pa(i1) = 0.0
        f1pe(i1) = 0.0

        do 415 k1=1,nexp1

        ai1 = i1
        f1pa(i1) = f1pa(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc1/tau1(k1))
        f1pe(i1) = f1pe(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc2/tau1(k1))

415     continue

410     continue



      write (*,*) 'What is the first decay time in the anisotropy?'
      read (*,*) a(2)
```

```
      write (*,*) 'What is its pre-exponential factor?'
      read(*,*) a(1)

        write (*,*) 'What is your scaling factor with the instrument
     & lamp with the VV-decay?'
        read (*,*) a(5)

        write (*,*) 'What is your scaling factor with the instrument
     & lamp with the HH-decay?'
        read (*,*) a(8)

      a(6) = .1
      a(7) = .1

      a(3) = 10.0
      a(4) = 10.0


      do 524 i=1,mfit
      lista(i) =i
524   continue

513   alambda = -0.1
      itera = 0
      kchi = 0

514   continue


      if (itera.ne.0) goto 540

515   write (*,521)
521   format(10Hiteration ,20H    amplitude       ,
     &20H    lifetime        ,20H    scattering      ,10H chisquare)
      write(*,*)'_____________________________________________________
     ____________'

      a(1) = sqrt(a(1))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))

540   continue

      call mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,covar,alpha,
     &      nca,chisq,alambda,nstart1,nstart2,e1,e2,itera,da,beta
     &  ,dyda,tpc1,tpc2,f1pa,f1pe)

      a(1) = a(1)*a(1)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
```

```
      a(8) = a(8)*a(8)

      write (*,*)

      write (*,542) itera,a(1),a(2),a(5),a(8),
     &     chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
542   format(I5,5H     ,4(3H   ,F10.3,7H       ),F10.3)

      write (*,543) a(3),a(4)
543   format(10HBackground,2(3H   ,F10.3,7H       ))

      write (*,545) a(6),a(7)
545   format(10HScattering,2(3H   ,F10.3,7H       ))

      write (*,544) a(5)/a(8)
523   format(10H          ,2(3H   ,F10.3,7H       ))
544   format(10HG-factor  ,3H   ,F10.3,7H       )

c     write(*,*) 'alambda = ',alambda

      a(1) = sqrt(a(1))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))

      if (itera.eq.1) goto 514
      if(chicca.eq.chisq) kchi = kchi+1
      if (chicca.ne.chisq) chicca = chisq
      if(chicca.ne.chisq) kchi = 0
      if (kchi.gt.20) goto 550
      goto 540

550   continue

c550  write (*,*) 'Do you want to try new amplitudes and lifetimes
c     & or starting analysis channel?'
c     write (*,*) 'yes = 1'
c     read(*,*) nn
c     if (nn.eq.1) then

c     do 525 i=1,nexp
c          ii = nexp+i
c          write (*,*) 'what is your new ',i,'th lifetime?'
c          read(*,*) a(ii)
c          write (*,*) 'what is your new ',i,'th amplitude?'
c          read(*,*) a(i)
c525  continue

c     write (*,*) 'what is your new scattering factor correction?'
c     read(*,*) a(mfit)
```

```
c           write (*,*) 'what is your new starting analysis channel?'
c           read(*,*) nstart
c           goto 513
c     else
            write (*,*) 'this is the end, my friend!'
c     endif


      do 1000 i=1,ndata1

      call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa)

      ym(i) = ymod
      res1(i) = y1(i) - ymod

      if (ymod.lt.1.0) then

            res1(i) = 0.0

      else

            res1(i) = res1(i)/sqrt(ymod)

      endif

      if(i.lt.nstart1(3)) res1(i) = 0.0

1000  continue

      do 1010 i=1,ndata1

      sum = sum + res1(i)*res1(i)

1010  continue

      n3 = ndata1 - nstart1(3) + 1
      an3 = ndata1 - nstart1(3) + 1

      do 1020 j=nstart1(3),n3-1

            do 1030 i=nstart1(3),ndata1-j

            auto1(j) = auto1(j) + res1(i)*res1(i+j)

1030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'
```

```
      endif

      auto1(j) = an3*auto1(j)/(am*sum)

1020  continue

      open(2,file='plot1.dat',status='old')

      do 1040 i=1,ndata1

      time = i*tpc1
      write (2,1050) time,e1(i),y1(i),ym(i),res1(i),auto1(i)

1050  format (4F10.2,2E12.3)

1040  continue

      close(2)

      a(1) = a(1)*a(1)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)

      open(2,file='plotres1',status='old')

      atcha = 0.0
      write(2,10) DD1
      write (2,1070) a(1),a(2)
      write (2,1070) atcha,atcha
      write (2,1070) a(5),a(5)/a(8)
      write (2,1070) frac,tauav1
      write (2,1070) atcha,atcha
      write (2,1080) a(3)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart1(3)

1070  format(2F10.5)
1075  format (I5)
1080  format (F10.5)

      close(2)

      a(1) = sqrt(a(1))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))


      do 2000 i=1,ndata2
```

```
      call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,f1pe)

      ym(i) = ymod
      res2(i) = y2(i) - ymod

      if (ymod.lt.1.0) then

            res2(i) = 0.0

      else

            res2(i) = res2(i)/sqrt(ymod)

      endif

      if(i.lt.nstart2(3)) res2(i) = 0.0

2000  continue

      do 2010 i=1,ndata2

      sum = sum + res2(i)*res2(i)

2010  continue

      n3 = ndata2 - nstart2(3) + 1
      an3 = ndata2 - nstart2(3) + 1

      do 2020 j=nstart2(3),n3-1

            do 2030 i=nstart2(3),ndata2-j

            auto2(j) = auto2(j) + res2(i)*res2(i+j)

2030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto2(j) = an3*auto2(j)/(am*sum)

2020  continue

      open(2,file='plot2.dat',status='old')

      do 2040 i=1,ndata2
```

```
      time = i*tpc2
      write (2,1050) time,e2(i),y2(i),ym(i),res2(i),auto2(i)

2040  continue

      close(2)


      a(1) = a(1)*a(1)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)

      atcha = 0.0

      open(2,file='plotres2',status='old')

      atcha = 0.0
      write(2,10) 'aniso01'
      write(2,10) DD2
      write (2,1070) a(1),a(2)
      write (2,1070) atcha,atcha
      write (2,1070) a(5),a(5)/a(8)
      write (2,1070) frac,tauav1
      write (2,1070) atcha,atcha
      write (2,1080) a(4)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart2(3)

      close(2)


      end







      subroutine foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1
     & ,f1pa)

      implicit double precision (a-h,o-z)

      real*8 a(ma),y1(10000),e1(10000),dyda(ma),f1pa(ndata1)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc1,ymod,ymod1,
     & ymod2,ymod3,tau0

c     if (i.eq.1) then
```

```
c     write (*,*) 'We are in foncs1!'
c     write (*,*) 'taum = ',taum
c     endif

c     if (i.eq.1) then
c     write (*,*) 'in foncs1, tpc1 = ',tpc1
c     endif


      a(1) = a(1)*a(1)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue

      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc1/a(2))
      tadi1 = 1.0

      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
      ymod2 = ymod2 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(1)*tadi1

      dyda(2) = dyda(2) + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*a(1)*tadi1*(akk-1.0)

      tadi1 = tadi1*tad1

20    continue

25    continue

      dyda(1) = 2.0*sqrt(a(1))*ymod2/a(1)
```

```
      dyda(2) = dyda(2)*tpc1/(a(2)*a(2))
      dyda(3) = 2.0*sqrt(a(3))
      dyda(4) = 0.0
      dyda(5) = 2.0*sqrt(a(5))*(ymod1+ymod2)/a(5)
      dyda(6) = e1(i)
      dyda(7) = 0.0
      dyda(8) = 0.0

      ymod = ymod1 + ymod2 + a(6)*e1(i) + a(3)

      a(1) = sqrt(a(1))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))

c        if(i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif

c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)
      return

      end


      subroutine foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2
     & ,f1pe)

      implicit double precision (a-h,o-z)

      real*8 a(ma),e2(10000),dyda(ma),f1pe(ndata2)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc2,ymod,ymod1,
     & ymod2,ymod3,tau0,aa1,aa2,aa3,aa4,aa5,aa6,akk,tauS


      a(1) = a(1)*a(1)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue



      ymod1 = 0.0
      ymod2 = 0.0
```

```
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc2/a(2))
      tadi1 = 1.0



      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
      ymod2 = ymod2 - amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
     & *a(1)*tadi1

      dyda(2) = dyda(2) - amol*e2(i-k+1)*tpc2*a(8)
     & *f1pe(k)*a(1)*tadi1*(akk-1.0)


      tadi1 = tadi1*tad1

20    continue

25    continue

      dyda(1) = 2.0*sqrt(a(1))*ymod2/a(1)
      dyda(2) = dyda(2)*tpc2/(a(2)*a(2))
      dyda(3) = 0.0
      dyda(4) = 2.0*sqrt(a(4))
      dyda(5) = 0.0
      dyda(6) = 0.0
      dyda(7) = e2(i)
      dyda(8) = 2.0*sqrt(a(8))*(ymod1+ymod2)/a(8)

      ymod = ymod1 + ymod2 + a(7)*e2(i) + a(4)

      a(1) = sqrt(a(1))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))

c        if (i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
```

```
c        endif
c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)


      return

      end







      subroutine mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & covar,alpha,nca,chisq,alamda,nstart1,nstart2,e1,e2,itera,
     & da,beta,dyda,tpc1,tpc2,f1pa,f1pe)


      implicit double precision (a-h,o-z)

      parameter (mmax=20)


      dimension
y1(ndata1),y2(ndata2),a(ma),lista(ma),e1(ndata1),e2(ndata2),
     &
      covar(mfit,mfit),alpha(mfit,mfit),atry(mmax),beta(mfit),da(mfit)
     &  ,dyda(mfit),nstart1(3),nstart2(3),f1pa(ndata1),
     &  f1pe(ndata2)

      real*8 taum,chisq,alamda,tpc1,tpc2,tauS

      save ochisq

c     write (*,*) 'we are in mrqmin. taum = ',taum
c     write (*,*) 'we are in mrqmin. tpc2 = ',tpc2

      if (alamda.lt.0) then
            kk = mfit+1
            do 12 j=1,ma
                  ihit=0
                  do 11 k=1,mfit
                        if(lista(k).eq.j) ihit=ihit+1
11                continue
                  if(ihit.eq.0) then
                        lista(kk)=j
                        kk = kk+1
                  else if (ihit.gt.1) then
                        pause 'improper permutation in lista'
```

```
                endif
12          continue
      if(kk.ne.(ma+1)) pause 'improper permutation in lista'
      alamda = 0.001

      call mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,alpha,beta,nca,chisq
     &      ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe)

c     write(*,43) itera,a(1),a(2),chisq/(ndata1+ndata2-nstart1-nstart2-
mfit)
c43   format(I5,5H     ,3(3H   ,F10.3,7H       ))


      do 13 j=1,ma
            atry(j)=a(j)
13    continue
      endif
      ochisq = chisq
      itera = itera+1


      do 15 j=1,mfit
            do 14 k=1,mfit
                  covar(j,k)=alpha(j,k)
14          continue
      covar(j,j)=alpha(j,j)*(1.+alamda)
      da(j) = beta(j)
15    continue

c     write (*,*) 'Just before Gaussj.'

      call gaussj(covar,mfit,nca,da,1,1)

c     write (*,*) 'Just after Gaussj.'


      if(alamda.eq.0) then
            call covsrt(covar,nca,ma,lista,mfit)
            return
      endif
      do 16 j=1,mfit
            atry(lista(j)) = a(lista(j))+da(j)
16    continue


      call
mrqcof(y1,y2,ndata1,ndata2,atry,ma,lista,mfit,covar,da,nca,chisq
     &  ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe)


      if (chisq.lt.ochisq) then
      alamda = 0.1*alamda
      ochisq=chisq
```

```
      do 18 j=1,mfit
            do 17 k=1,mfit
                  alpha(j,k)=covar(j,k)
17          continue
      beta(j)=da(j)
      a(lista(j))=atry(lista(j))
18    continue
      else
      alamda = 10.*alamda
      chisq=ochisq
      endif
      return
      end


      subroutine mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & alpha,beta,nalp,chisq
     &      ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe)

      implicit double precision (a-h,o-z)

      dimension y1(10000),y2(10000),alpha(nalp,nalp),beta(mfit),
     &  dyda(mfit),lista(mfit),a(ma),e1(10000),e2(10000)
     & ,nstart1(3),nstart2(3),f1pa(ndata1)
     & ,f1pe(ndata2)
      real*8 taum,chisq,tpc1,tpc2,tauS

c     write (*,*) 'we are in mrqcof. taum = ',taum
c     write (*,*) 'we are in mrqcof. tpc2 = ',tpc2

      do 112 j=1,mfit
            do 111 k=1,j
            alpha(j,k) = 0.
111         continue
      beta(j) = 0.
112   continue
      chisq=0.

120   do 115 i=nstart1(3),ndata1

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 114 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 113 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
113                     continue
            beta(j)=beta(j)+dy*wt
114         continue
      chisq=chisq+dy*dy*sig2i
```

```
115   continue

320   do 315 i=nstart1(1),nstart1(2)

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 314 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 313 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
313                     continue
            beta(j)=beta(j)+dy*wt
314         continue
      chisq=chisq+dy*dy*sig2i
315   continue

c     write (*,*) 'Chisq value after monomer: ',chisq

220   do 215 i=nstart2(3),ndata2

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe)

c           if (i.lt.nstart2) goto 215
            sig2i = 1./y2(i)
            dy = y2(i)-ymod
            do 214 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 213 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
213                     continue
            beta(j)=beta(j)+dy*wt
214         continue
      chisq=chisq+dy*dy*sig2i
215   continue

420   do 415 i=nstart2(1),nstart2(2)

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe)

c           if (i.lt.nstart2) goto 215
            sig2i = 1./y2(i)
            dy = y2(i)-ymod
            do 414 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 413 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
413                     continue
            beta(j)=beta(j)+dy*wt
```

```
414         continue
      chisq=chisq+dy*dy*sig2i
415   continue

c     write (*,*) 'Chisq value after excimer: ',chisq


      do 117 j=2,mfit
            do 116 k=1,j-1
            alpha(k,j)=alpha(j,k)
116         continue
117   continue
      return
      end




      subroutine covsrt(covar,ncvm,ma,lista,mfit)

      implicit double precision (a-h,o-z)

      dimension covar(ncvm,ncvm),lista(mfit)

      do 212 j=1,ma-1
            do 211 i=j+1,ma
            covar(i,j) = 0.
211         continue
212   continue
      do 214 i=1,mfit-1
            do 213 j=i+1,mfit
                  if(lista(j).gt.lista(i)) then
                        covar(lista(j),lista(i))=covar(i,j)
                  else
                        covar(lista(i),lista(j))=covar(i,j)
                  endif
213         continue
214   continue
      swap=covar(1,1)
      do 215 j=1,ma
      covar(1,j) = covar(j,j)
      covar(j,j) = 0.
215   continue
      covar(lista(1),lista(1))=swap
      do 216 j=2,mfit
      covar(lista(j),lista(j))=covar(1,j)
216   continue
      do 218 j=2,ma
            do 217 i=1,j-1
            covar(i,j)=covar(j,i)
217   continue
```

```
218   continue
      return
      end




      subroutine gaussj(a,n,np,b,m,mp)

      implicit double precision (a-h,o-z)

      parameter (nmax=50)
      dimension a(np,np),b(np),ipiv(nmax),indxr(nmax),indxc(nmax)

      do 311 j=1,n
            ipiv(j) = 0
311   continue

      do 322 i=1,n
            big = 0.

            do 313 j=1,n

                  if (ipiv(j).ne.1) then

                  do 312 k=1,n

                  if(ipiv(k).eq.0) then

                        if (abs(a(j,k)).ge.big) then
                        big = abs(a(j,k))
                        irow = j
                        icol = k
                        endif

                        else if (ipiv(k).gt.1) then
                        pause 'singular matrix'
                        endif

312                 continue

                  endif

313         continue

      ipiv(icol) = ipiv(icol) + 1

      if (irow.ne.icol) then

            do 314 l=1,n
```

```
                dum = a(irow,l)
                a(irow,l) = a(icol,l)
                a(icol,l) = dum
314         continue

                dum = b(irow)
                b(irow) = b(icol)
                b(icol) = dum

      endif

      indxr(i) = irow
      indxc(i) = icol

      if (a(icol,icol).eq.0)  pause 'singular matrix'

      pivinv = 1./a(icol,icol)

      a(icol,icol) = 1.

      do 316 l=1,n
            a(icol,l) = a(icol,l)*pivinv
316   continue

            b(icol) = b(icol)*pivinv

      do 321 ll=1,n
            if (ll.ne.icol) then

                  dum = a(ll,icol)
                  a(ll,icol) = 0.

                  do 318 l=1,n
                        a(ll,l) = a(ll,l) - a(icol,l)*dum
318               continue

                        b(ll) = b(ll) - b(icol)*dum

            endif

321   continue

322   continue

      do 324 l=n,1,-1
            if (indxr(l).ne.indxc(l)) then

                  do 323 k=1,n
                        dum = a(k,indxr(l))
                        a(k,indxr(l)) = a(k,indxc(l))
                        a(k,indxc(l)) = dum
323               continue
```

```
            endif

324   continue

      return
      end
```

I] Listing of *aniso02n-4*

```
c     Program created April 9, 2015

c     This program fits the I(para;para) and I(para;per) decays
c       globally assuming a biexponential anisotropy.
c       There is a function for non-covalently attached dyes.
c       There is a function for covalently attached dyes.
c       The free dyes have a different lifetime as the bound dyes.
c       The G-factor is optimized
c       There is no free dye in the solution.
c       The long lifetime is optimized in Ipara and Iper.
c       r(t) = rox(r1xexp(-t/taur1) + r2xexp(-t/taur2)
c       + r3xexp(-t/taur3)). The pre-exponential factors
c       r1, r2, and r3 are fixed in the analysis. The rotational
c       times taur1, taur2, and taur3 are optimized according
c       to the diffusion coefficients Dpara and Dper.
c       r0 is fixed in the analysis.
c       Wobbling angle is optimized.

      implicit double precision (a-h,o-z)

      parameter (mfit=9,ma=9,nca=9)

      dimension y1(10000),lista(mfit),a(mfit),
     & covar(mfit,mfit),alpha(mfit,mfit),
     & e1(10000),da(mfit),beta(mfit)
     & ,dyda(mfit),ym(10000),res1(10000),auto1(10000)
     & ,e2(10000),y2(10000),res2(10000),auto2(10000)
     & ,nstart1(3),nstart2(3)
```

```
     & ,a1(10),tau1(10),f1pa(10000),f1pe(10000)

      real*8 taum,chisq,tpc1,tpc2


10    format(A)
      character *30 DD1,DD2

      write (*,*) 'How many chanels do you want to work with
     & for the VV decay?'
      read (*,*) ndata1

c     write (*,*) 'What is the name of your lamp file ?'
c     read(*,10) DD

c     open (1,file=DD,status='old')

c     read(1,*) (e(i),i=1,ndata)

c     close(1)

c     write (*,*) 'what is the name of your fluorescence decay?'
c     read(*,10) DD

c     open(1,file=DD,status='old')

c     read(1,*) (y(i),i=1,ndata)

c     close(1)

c     write (*,*) 'What is your file''s name for VV?'
c     read(*,10) DD1

c     open(1,file=DD1,status='old')

c     do 15 i=1,ndata1

c     read(1,*) e1(i),y1(i)

c15   continue

c     close (1)

      write (*,*) 'What is your lamp''s filename for
     & the VV decay?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 13 i=1,9
13    read(1,*)

      do 15 i=1,ndata1
```

```
      read(1,*) x,e1(i)

15    continue

      close (1)


      write (*,*) 'What is your VV decay''s filename?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 14 i=1,9
14    read(1,*)

      do 16 i=1,ndata1

      read(1,*) x,y1(i)

16    continue

      close (1)

      write (*,*) 'How many chanels do you want to work with
     & for the VH decay?'
      read (*,*) ndata2


c     write (*,*) 'What is your file''s name for VH?'
c     read(*,10) DD2

c     open(1,file=DD2,status='old')

c     do 16 i=1,ndata2

c     read(1,*) e2(i),y2(i)

c16   continue

c     close (1)


      write (*,*) 'What is your lamp''s filename for
     & the VH decay?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 313 i=1,9
313   read(1,*)

      do 315 i=1,ndata2
```

```
      read(1,*) x,e2(i)

315   continue

      close (1)


      write (*,*) 'What is your VH decay''s filename?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 314 i=1,9
314   read(1,*)

      do 316 i=1,ndata2

      read(1,*) x,y2(i)

316   continue

      close (1)

      write (*,*) 'What is the lifetime of the reference compound
     & for the VV decay?'
      read(*,*) taur1

      write (*,*) 'What is your time per chanel for the VV
     & decay?'
      read(*,*) tpc1

      taur1 = exp(-tpc1/taur1)

      alold2 = e1(1)

      do 40 i=2,ndata1

      alold1 = e1(i)
      e1(i) = e1(i) - alold2*taur1
      alold2 = alold1

      if(e1(i).lt.0) e1(i)=0.0001

40    continue


      write (*,*) 'What is the lifetime of the reference compound
     & for the VH decay?'
      read(*,*) taur2

      write (*,*) 'What is your time per chanel for the VH
     & decay?'
```

```
      read(*,*) tpc2

      taur2 = exp(-tpc2/taur2)

      alold2 = e2(1)

      do 41 i=2,ndata2

      alold1 = e2(i)
      e2(i) = e2(i) - alold2*taur2
      alold2 = alold1

      if(e2(i).lt.0) e2(i)=0.0001

41    continue

      imax = 0

      do 500 i=2,ndata1

      if (e1(i).gt.emax) then
      imax = i
      emax = e1(i)
      endif

500   continue


      write (*,511)
      write (*,*) '_____________________________________________________'
511   format(10H   channel,10H      lamp,10H     decay)

      do 520 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e1(k),y1(k)
512   format (3H   ,I4,3H   ,2(2H  ,F7.1,1H ))
      endif
520   continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VV decay?'
      read(*,*) nstart1(3)

      write (*,*) 'At which channel does the background noise start
     & for the VV decay?'
      read(*,*) nstart1(1)

      write (*,*) 'At which channel does the background noise end
     & for the VV decay?'
      read(*,*) nstart1(2)

c     sume = 0.0
```

```
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 860 i=nback1,nback2

c     sume = sume + e1(i)
c     sumy = sumy + y1(i)

c860  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 870 i=1,ndata1

c     e1(i) = e1(i) - sume
c     y1(i) = y1(i) - sumy

c     if(e1(i).lt.0.0) e1(i) = 0.00001
c     if(y1(i).lt.0.0) y1(i) = 0.00001

c870  continue


      imax = 0
      emax = 0.0

      do 501 i=2,ndata2

      if (e2(i).gt.emax) then
      imax = i
      emax = e2(i)
      endif

501   continue


      write (*,511)
      write (*,*) '_____________________________________________________'

      do 1522 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e2(k),y2(k)
      endif
1522  continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VH decay?'
      read(*,*) nstart2(3)

      write (*,*) 'At which channel does the background noise start
```

```
     & for the VH decay?'
      read(*,*) nstart2(1)

      write (*,*) 'At which channel does the background noise end
     & for the VH decay?'
      read(*,*) nstart2(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 861 i=nback1,nback2

c     sume = sume + e2(i)
c     sumy = sumy + y2(i)

c861  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 871 i=1,ndata2

c     e2(i) = e2(i) - sume
c     y2(i) = y2(i) - sumy

c     if(e2(i).lt.0.0) e2(i) = 0.00001
c     if(y2(i).lt.0.0) y2(i) = 0.00001

c871  continue

        write (*,*) 'How many decay times do you need to represent
     & the decay of the dye bound to the macromolecule.'
        read (*,*) nexp1

        tauav1 = 0.0

        do 400 k1=1,nexp1
        write (*,*) 'What is the',k1,'th decaytime?'
        read (*,*) tau1(k1)
        write (*,*) 'What is its normalized pre-exponential factor?'
        read (*,*) a1(k1)

        tauav1 = tauav1 + a1(k1)*tau1(k1)

400     continue


        do 410 i1=1,10000

        f1pa(i1) = 0.0
        f1pe(i1) = 0.0
```

```
         do 415 k1=1,nexp1

         ai1 = i1
         f1pa(i1) = f1pa(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc1/tau1(k1))
         f1pe(i1) = f1pe(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc2/tau1(k1))

415      continue

410      continue


       write (*,*) 'What is Dpara (in ns^-1)?'
       read (*,*) a(2)

       write (*,*) 'What is Dper (in ns^-1)?'
       read (*,*) a(9)

         write (*,*) 'What is ro? (fixed in the analysis)'
         read (*,*) ar0

         write (*,*) 'Calculation of the pre-exponential factors
     & for the anisotropy function.'
         write (*,*) 'What is the angle beta-A?'
         read (*,*) betaA
         betaA = betaA*3.1416/180.0

         write (*,*) 'What is the angle beta-E?'
         read (*,*) betaE
         betaE= betaE*3.1416/180.0

         write (*,*) 'What is the angle xi between the diploe
     & moments uA and uE?'
         read (*,*) xi
         xi = xi*3.1416/180.0

         ar1 = ar0*0.75*sin(2.0*betaA)*sin(2.0*betaE)*cos(xi)
         ar2 = ar0*0.75*sin(betaA)*sin(betaA)*sin(betaE)
     & *sin(betaE)*cos(2.0*xi)
         ar3 = ar0*0.25*(3.0*cos(betaA)*cos(betaA)-1.0)*
     & (3*cos(betaE)*cos(betaE)-1.0)

         write (*,551) ar1,1.0/(a(2)+5.0*a(9))
         write (*,552) ar2,1.0/(4.0*a(2)+2.0*a(9))
         write (*,553) ar3,1.0/(6.0*a(9))

         write (*,*) 'What is your wobbling angle? (< 55o)'
         read (*,*) a(1)

         a(1) = a(1)*3.1416/180.0
```

```
        write (*,*) 'What is your scaling factor for Ipara?'
        read (*,*) a(5)

        write (*,*) 'What is your scaling factor for Iper?'
        read (*,*) a(8)

      a(6) = .1
      a(7) = .1

      a(3) = 10.0
      a(4) = 10.0

      do 524 i=1,mfit
      lista(i) =i
524   continue

513   alambda = -0.1
      itera = 0
      kchi = 0

514   continue



      if (itera.ne.0) goto 540

515   write (*,521)
521   format(10Hiteration ,20H    amplitude       ,
     &20H    lifetime        ,20H    scattering      ,10H chisquare)
      write(*,*)'_____________________________________________________
     ____________'

c     write (*,*) 'just before entering mrqmin. taum = ',taum

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))

540   continue

      call mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,covar,alpha,
     &     nca,chisq,alambda,nstart1,nstart2,e1,e2,itera,da,beta
     &  ,dyda,tpc1,tpc2,f1pa,f1pe,aG,ar1,ar2,ar3)

      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
```

```
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)

      write (*,*)

      write (*,542) itera,a(1)*180/3.1316,a(2),a(9),
     &     chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
542   format(I5,5H     ,3(3H   ,F10.3,7H       ),F10.3)

547   format(10Hr02       ,(3H   ,F10.3,7H       ),
     & 10Htaur2     ,(3H   ,F10.3,7H       ))

      write (*,543) a(3),a(4)
543   format(10HBackground,2(3H   ,F10.3,7H       ))

      write (*,545) a(6),a(7)
545   format(10HScattering,2(3H   ,F10.3,7H       ))

      write (*,546) a(5),a(8)
546   format(15Hscaling factors,2(3H   ,F10.3,7H       ))

      write (*,544) a(5)/a(8)
523   format(10H          ,2(3H   ,F10.3,7H       ))
544   format(10HG-factor  ,3H   ,F10.3,7H       )

        write (*,551) ar1,1.0/(a(2)+5.0*a(9))
        write (*,552) ar2,1.0/(4.0*a(2)+2.0*a(9))
        write (*,553) ar3,1.0/(6.0*a(9))

551     format(10H r1 =     ,F10.5,10H taur1   ,F10.5)
552     format(10H r2 =     ,F10.5,10H taur2   ,F10.5)
553     format(10H r3 =     ,F10.5,10H taur3   ,F10.5)


      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))

      if (itera.eq.1) goto 514
      if(chicca.eq.chisq) kchi = kchi+1
      if (chicca.ne.chisq) chicca = chisq
      if(chicca.ne.chisq) kchi = 0
      if (kchi.gt.100) goto 550
      goto 540

550   continue
```

```
c550  write (*,*) 'Do you want to try new amplitudes and lifetimes
c     & or starting analysis channel?'
c     write (*,*) 'yes = 1'
c     read(*,*) nn
c     if (nn.eq.1) then

c     do 525 i=1,nexp
c          ii = nexp+i
c          write (*,*) 'what is your new ',i,'th lifetime?'
c          read(*,*) a(ii)
c          write (*,*) 'what is your new ',i,'th amplitude?'
c          read(*,*) a(i)
c525  continue

c     write (*,*) 'what is your new scattering factor correction?'
c     read(*,*) a(mfit)

c          write (*,*) 'what is your new starting analysis channel?'
c          read(*,*) nstart
c          goto 513
c     else
          write (*,*) 'this is the end, my friend!'
c     endif


      do 1000 i=1,ndata1

      call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,aG,
     & ar1,ar2,ar3)

      ym(i) = ymod
      res1(i) = y1(i) - ymod

      if (ymod.lt.1.0) then

           res1(i) = 0.0

      else

           res1(i) = res1(i)/sqrt(ymod)

      endif

      if(i.lt.nstart1(3)) res1(i) = 0.0

1000  continue

      do 1010 i=1,ndata1

      sum = sum + res1(i)*res1(i)

1010  continue
```

```
      n3 = ndata1 - nstart1(3) + 1
      an3 = ndata1 - nstart1(3) + 1

      do 1020 j=nstart1(3),n3-1

            do 1030 i=nstart1(3),ndata1-j

            auto1(j) = auto1(j) + res1(i)*res1(i+j)

1030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto1(j) = an3*auto1(j)/(am*sum)

1020  continue

      open(2,file='plot1.dat',status='old')

      do 1040 i=1,ndata1

      time = i*tpc1
      write (2,1050) time,e1(i),y1(i),ym(i),res1(i),auto1(i)

1050  format (4F10.2,2E12.3)

1040  continue

      close(2)

      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)

      open(2,file='plotres1',status='old')

      atcha = 0.0
      write(2,10) DD1
      write (2,1070) a(1)*180/3.1416,tauav1
      write (2,1070) a(2),a(9)
      write (2,1070) atcha,atcha
      write (2,1070) atcha,atcha
```

```
      write (2,1070) a(5),a(5)/a(8)
      write (2,1080) a(3)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart1(3)

1070  format(2F10.5)
1075  format (I5)
1080  format (F10.5)

      close(2)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))


      do 2000 i=1,ndata2

      call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,f1pe,aG,
     & ar1,ar2,ar3)

      ym(i) = ymod
      res2(i) = y2(i) - ymod

      if (ymod.lt.1.0) then

            res2(i) = 0.0

      else

            res2(i) = res2(i)/sqrt(ymod)

      endif

      if(i.lt.nstart2(3)) res2(i) = 0.0

2000  continue

      do 2010 i=1,ndata2

      sum = sum + res2(i)*res2(i)

2010  continue

      n3 = ndata2 - nstart2(3) + 1
      an3 = ndata2 - nstart2(3) + 1

      do 2020 j=nstart2(3),n3-1
```

```
            do 2030 i=nstart2(3),ndata2-j

            auto2(j) = auto2(j) + res2(i)*res2(i+j)

2030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto2(j) = an3*auto2(j)/(am*sum)

2020  continue

      open(2,file='plot2.dat',status='old')

      do 2040 i=1,ndata2

      time = i*tpc2
      write (2,1050) time,e2(i),y2(i),ym(i),res2(i),auto2(i)

2040  continue

      close(2)


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)

      atcha = 0.0

      open(2,file='plotres2',status='old')

      atcha = 0.0
      write(2,10) 'aniso02nb'
      write(2,10) DD2
      write (2,1070) a(1)*180/3.1416,tauav1
      write (2,1070) a(2),a(9)
      write (2,1070) atcha,atcha
      write (2,1070) atcha,atcha
      write (2,1070) a(8),a(5)/a(8)
      write (2,1080) a(4)
```

```
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart2(3)

      close(2)


      end







      subroutine foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1
     & ,f1pa,aG,ar1,ar2,ar3)

      implicit double precision (a-h,o-z)

      real*8 a(ma),y1(10000),e1(10000),dyda(ma),f1pa(ndata1)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc1,ymod,ymod1,
     & ymod2,ymod3,tau0

c     if (i.eq.1) then
c     write (*,*) 'We are in foncs1!'
c     write (*,*) 'taum = ',taum
c     endif

c     if (i.eq.1) then
c     write (*,*) 'in foncs1, tpc1 = ',tpc1
c     endif


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue

      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0
      ymod5 = 0.0
      ymod6 = 0.0
```

```
      if (i.eq.1) goto 25

      tad1 = dexp(-tpc1*(a(2)+5.0*a(9)))
      tadi1 = 1.0

      tid1 = dexp(-tpc1*(4.0*a(2)+2.0*a(9)))
      tidi1 = 1.0

      ted1 = dexp(-tpc1*(6.0*a(9)))
      tedi1 = 1.0

      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
      ymod2 = ymod2 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*ar1*tadi1
      ymod3 = ymod3 - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(1)*a(1)*ar1*tadi1/12.0
      ymod4 = ymod4 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*ar2*tidi1
      ymod5 = ymod5 - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(1)*a(1)*ar2*tidi1/3.0
      ymod6 = ymod6 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*ar3*tedi1

      dyda(2) = dyda(2) - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*((1.0-a(1)*a(1)/12.0)*ar1*tadi1 +
     &  4.0*(1.0-a(1)*a(1)/3.0)*ar2*tidi1)*(akk-1.0)

      dyda(9) = dyda(9) - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*(5.0*(1.0-a(1)*a(1)/12.0)*ar1*tadi1 +
     &  2.0*(1.0-a(1)*a(1)/3.0)*ar2*tidi1+6.0*ar3*tedi1)
     & *(akk-1.0)

        tadi1 = tadi1*tad1
      tidi1 = tidi1*tid1
      tedi1 = tedi1*ted1

20    continue

25    continue

      dyda(1) = 4.0*sqrt(a(1))*(ymod3+ymod5)/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc1
      dyda(3) = 2.0*sqrt(a(3))
```

```
      dyda(4) = 0.0
      dyda(5) = 2.0*sqrt(a(5))*(ymod1+ymod2+ymod3+ymod4
     & +ymod5+ymod6)/a(5)
      dyda(6) = e1(i)
      dyda(7) = 0.0
        dyda(8) = 0.0
      dyda(9) = 2.0*sqrt(a(9))*dyda(9)*tpc1

      ymod = ymod1 + ymod2 + ymod3 + ymod4 + ymod5 + ymod6
     & + a(6)*e1(i) + a(3)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))

c        if(i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif

c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)
      return

      end


      subroutine foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2
     & ,f1pe,aG,ar1,ar2,ar3)

      implicit double precision (a-h,o-z)

      real*8 a(ma),e2(10000),dyda(ma),f1pe(ndata2)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc2,ymod,ymod1,
     & ymod2,ymod3,tau0,aa1,aa2,aa3,aa4,aa5,aa6,akk,tauS


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
        a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue
```

```
 ymod1 = 0.0
 ymod2 = 0.0
 ymod3 = 0.0
 ymod4 = 0.0
 ymod5 = 0.0
 ymod6 = 0.0

 if (i.eq.1) goto 25

 tad1 = dexp(-tpc2*(a(2)+5.0*a(9)))
 tadi1 = 1.0

 tid1 = dexp(-tpc2*(4.0*a(2)+2.0*a(9)))
 tidi1 = 1.0

 ted1 = dexp(-tpc2*(6.0*a(9)))
 tedi1 = 1.0

 do 20 k=1,i

 akk = k

 if((k.eq.1).or.(k.eq.i)) then
 amol = 0.5
 else
 amol = 1.0
 endif

 ymod1 = ymod1 + amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
 ymod2 = ymod2 - amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
& *ar1*tadi1
 ymod3 = ymod3 + amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
& *a(1)*a(1)*ar1*tadi1/12.0
 ymod4 = ymod4 - amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
& *ar2*tidi1
 ymod5 = ymod5 + amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
& *a(1)*a(1)*ar2*tidi1/3.0
 ymod6 = ymod6 - amol*e2(i-k+1)*tpc2*a(8)*f1pe(k)
& *ar3*tedi1

 dyda(2) = dyda(2) + amol*e2(i-k+1)*tpc2*a(8)
& *f1pe(k)*((1.0-a(1)*a(1)/12.0)*ar1*tadi1 +
&  4.0*(1.0-a(1)*a(1)/3.0)*ar2*tidi1)*(akk-1.0)

 dyda(9) = dyda(9) + amol*e2(i-k+1)*tpc2*a(8)
& *f1pe(k)*(5.0*(1.0-a(1)*a(1)/12.0)*ar1*tadi1 +
&  2.0*(1.0-a(1)*a(1)/3.0)*ar2*tidi1
&  + 6.0*ar3*tedi1)*(akk-1.0)

 tadi1 = tadi1*tad1
```

```
      tidi1 = tidi1*tid1
      tedi1 = tedi1*ted1

20    continue

25    continue

      dyda(1) = 4.0*sqrt(a(1))*
     &  (ymod3+ymod5)/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc2
      dyda(3) = 0.0
      dyda(4) = 2.0*sqrt(a(4))
      dyda(5) = 0.0
      dyda(6) = 0.0
      dyda(7) = e2(i)
      dyda(8) = 2.0*sqrt(a(8))*(ymod1+ymod2+ymod3+ymod4
     & ymod5+ymod6)/a(8)
      dyda(9) = 2.0*sqrt(a(9))*dyda(9)*tpc2

      ymod = ymod1 + ymod2 + ymod3 + ymod4 + ymod5 + ymod6
     & + a(7)*e2(i) + a(4)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))

c        if (i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif
c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)


      return

      end




      subroutine mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & covar,alpha,nca,chisq,alamda,nstart1,nstart2,e1,e2,itera,
     & da,beta,dyda,tpc1,tpc2,f1pa,f1pe,aG,
     & ar1,ar2,ar3)
```

```
      implicit double precision (a-h,o-z)

      parameter (mmax=20)


      dimension
y1(ndata1),y2(ndata2),a(ma),lista(ma),e1(ndata1),e2(ndata2),
     &
      covar(mfit,mfit),alpha(mfit,mfit),atry(mmax),beta(mfit),da(mfit)
     &  ,dyda(mfit),nstart1(3),nstart2(3),f1pa(ndata1),
     &  f1pe(ndata2)

      real*8 taum,chisq,alamda,tpc1,tpc2,tauS

      save ochisq

c     write (*,*) 'we are in mrqmin. taum = ',taum
c     write (*,*) 'we are in mrqmin. tpc2 = ',tpc2

      if (alamda.lt.0) then
            kk = mfit+1
            do 12 j=1,ma
                  ihit=0
                  do 11 k=1,mfit
                        if(lista(k).eq.j) ihit=ihit+1
11                continue
                  if(ihit.eq.0) then
                        lista(kk)=j
                        kk = kk+1
                  else if (ihit.gt.1) then
                        pause 'improper permutation in lista'
                  endif
12          continue
      if(kk.ne.(ma+1)) pause 'improper permutation in lista'
      alamda = 0.001

      call mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,alpha,beta,nca,chisq
     &     ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG,
     & ar1,ar2,ar3)

c     write(*,43) itera,a(1),a(2),chisq/(ndata1+ndata2-nstart1-nstart2-
mfit)
c43   format(I5,5H     ,3(3H   ,F10.3,7H       ))


      do 13 j=1,ma
            atry(j)=a(j)
13    continue
      endif
      ochisq = chisq
      itera = itera+1
```

```
      do 15 j=1,mfit
            do 14 k=1,mfit
                  covar(j,k)=alpha(j,k)
14          continue
      covar(j,j)=alpha(j,j)*(1.+alamda)
      da(j) = beta(j)
15    continue

c     write (*,*) 'Just before Gaussj.'

      call gaussj(covar,mfit,nca,da,1,1)

c     write (*,*) 'Just after Gaussj.'


      if(alamda.eq.0) then
            call covsrt(covar,nca,ma,lista,mfit)
            return
      endif
      do 16 j=1,mfit
            atry(lista(j)) = a(lista(j))+da(j)
16    continue


      call
mrqcof(y1,y2,ndata1,ndata2,atry,ma,lista,mfit,covar,da,nca,chisq
     &  ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG,
     & ar1,ar2,ar3)


      if (chisq.lt.ochisq) then
      alamda = 0.1*alamda
      ochisq=chisq
      do 18 j=1,mfit
            do 17 k=1,mfit
                  alpha(j,k)=covar(j,k)
17          continue
      beta(j)=da(j)
      a(lista(j))=atry(lista(j))
18    continue
      else
      alamda = 10.*alamda
      chisq=ochisq
      endif
      return
      end


      subroutine mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & alpha,beta,nalp,chisq
     &      ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe
```

```
     &  ,aG,ar1,ar2,ar3)

      implicit double precision (a-h,o-z)

      dimension y1(10000),y2(10000),alpha(nalp,nalp),beta(mfit),
     &  dyda(mfit),lista(mfit),a(ma),e1(10000),e2(10000)
     & ,nstart1(3),nstart2(3),f1pa(ndata1)
     & ,f1pe(ndata2)
      real*8 taum,chisq,tpc1,tpc2,tauS

c     write (*,*) 'we are in mrqcof. taum = ',taum
c     write (*,*) 'we are in mrqcof. tpc2 = ',tpc2

      do 112 j=1,mfit
            do 111 k=1,j
            alpha(j,k) = 0.
111         continue
      beta(j) = 0.
112   continue
      chisq=0.

120   do 115 i=nstart1(3),ndata1

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG,ar1,ar2,ar3)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 114 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 113 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
113                     continue
            beta(j)=beta(j)+dy*wt
114         continue
      chisq=chisq+dy*dy*sig2i
115   continue

320   do 315 i=nstart1(1),nstart1(2)

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG,ar1,ar2,ar3)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 314 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 313 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
313                     continue
            beta(j)=beta(j)+dy*wt
```

```
314         continue
      chisq=chisq+dy*dy*sig2i
315   continue

c     write (*,*) 'Chisq value after monomer: ',chisq

220   do 215 i=nstart2(3),ndata2

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe,aG,ar1,ar2,ar3)

c           if (i.lt.nstart2) goto 215
            sig2i = 1./y2(i)
            dy = y2(i)-ymod
            do 214 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 213 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
213                     continue
            beta(j)=beta(j)+dy*wt
214         continue
      chisq=chisq+dy*dy*sig2i
215   continue

420   do 415 i=nstart2(1),nstart2(2)

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe,aG,ar1,ar2,ar3)

c           if (i.lt.nstart2) goto 215
            sig2i = 1./y2(i)
            dy = y2(i)-ymod
            do 414 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 413 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
413                     continue
            beta(j)=beta(j)+dy*wt
414         continue
      chisq=chisq+dy*dy*sig2i
415   continue

c     write (*,*) 'Chisq value after excimer: ',chisq


      do 117 j=2,mfit
            do 116 k=1,j-1
            alpha(k,j)=alpha(j,k)
116         continue
117   continue
      return
      end
```

```
      subroutine covsrt(covar,ncvm,ma,lista,mfit)

      implicit double precision (a-h,o-z)

      dimension covar(ncvm,ncvm),lista(mfit)

      do 212 j=1,ma-1
            do 211 i=j+1,ma
            covar(i,j) = 0.
211         continue
212   continue
      do 214 i=1,mfit-1
            do 213 j=i+1,mfit
                  if(lista(j).gt.lista(i)) then
                        covar(lista(j),lista(i))=covar(i,j)
                  else
                        covar(lista(i),lista(j))=covar(i,j)
                  endif
213         continue
214   continue
      swap=covar(1,1)
      do 215 j=1,ma
      covar(1,j) = covar(j,j)
      covar(j,j) = 0.
215   continue
      covar(lista(1),lista(1))=swap
      do 216 j=2,mfit
      covar(lista(j),lista(j))=covar(1,j)
216   continue
      do 218 j=2,ma
            do 217 i=1,j-1
            covar(i,j)=covar(j,i)
217   continue
218   continue
      return
      end




      subroutine gaussj(a,n,np,b,m,mp)

      implicit double precision (a-h,o-z)

      parameter (nmax=50)
```

```
      dimension a(np,np),b(np),ipiv(nmax),indxr(nmax),indxc(nmax)

      do 311 j=1,n
            ipiv(j) = 0
311   continue

      do 322 i=1,n
            big = 0.

            do 313 j=1,n

                  if (ipiv(j).ne.1) then

                  do 312 k=1,n

                  if(ipiv(k).eq.0) then

                        if (abs(a(j,k)).ge.big) then
                        big = abs(a(j,k))
                        irow = j
                        icol = k
                        endif

                        else if (ipiv(k).gt.1) then
                        pause 'singular matrix'
                        endif

312               continue

                  endif

313         continue

      ipiv(icol) = ipiv(icol) + 1

      if (irow.ne.icol) then

            do 314 l=1,n
                  dum = a(irow,l)
                  a(irow,l) = a(icol,l)
                  a(icol,l) = dum
314         continue

                  dum = b(irow)
                  b(irow) = b(icol)
                  b(icol) = dum

      endif

      indxr(i) = irow
      indxc(i) = icol

      if (a(icol,icol).eq.0)  pause 'singular matrix'
```

```
      pivinv = 1./a(icol,icol)

      a(icol,icol) = 1.

      do 316 l=1,n
            a(icol,l) = a(icol,l)*pivinv
316   continue

            b(icol) = b(icol)*pivinv

      do 321 ll=1,n
            if (ll.ne.icol) then

                  dum = a(ll,icol)
                  a(ll,icol) = 0.

                  do 318 l=1,n
                        a(ll,l) = a(ll,l) - a(icol,l)*dum
318               continue

                        b(ll) = b(ll) - b(icol)*dum

            endif

321   continue

322   continue

      do 324 l=n,1,-1
            if (indxr(l).ne.indxc(l)) then

                  do 323 k=1,n
                        dum = a(k,indxr(l))
                        a(k,indxr(l)) = a(k,indxc(l))
                        a(k,indxc(l)) = dum
323               continue

            endif

324   continue

      return
      end
```

J] Listing of *aniso02d*

```
c     Program created April 9, 2015
c     This program fits the I(para;para) and I(para;per) decays
```

```
c       globally assuming a biexponential anisotropy.
c       There is NO function for non-covalently attached dyes.
c       There is a function for covalently attached dyes.
c       Pre-exponential factors of the anisotropy can be negative.
c       the G factor is omitted and it is calculated.

      implicit double precision (a-h,o-z)

      parameter (mfit=10,ma=10,nca=10)

      dimension y1(10000),lista(mfit),a(mfit),
     & covar(mfit,mfit),alpha(mfit,mfit),
     & e1(10000),da(mfit),beta(mfit)
     & ,dyda(mfit),ym(10000),res1(10000),auto1(10000)
     & ,e2(10000),y2(10000),res2(10000),auto2(10000)
     & ,nstart1(3),nstart2(3)
     & ,a1(10),a2(10),tau1(10),tau2(10),f1pa(10000),f1pe(10000)

      real*8 taum,chisq,tpc1,tpc2


10    format(A)
      character *30 DD1,DD2

      write (*,*) 'How many chanels do you want to work with
     & for the VV decay?'
      read (*,*) ndata1

c     write (*,*) 'What is the name of your lamp file ?'
c     read(*,10) DD

c     open (1,file=DD,status='old')

c     read(1,*) (e(i),i=1,ndata)

c     close(1)

c     write (*,*) 'what is the name of your fluorescence decay?'
c     read(*,10) DD

c     open(1,file=DD,status='old')

c     read(1,*) (y(i),i=1,ndata)

c     close(1)

c     write (*,*) 'What is your file''s name for VV?'
c     read(*,10) DD1

c     open(1,file=DD1,status='old')

c     do 15 i=1,ndata1
```

```
c     read(1,*) e1(i),y1(i)

c15   continue

c     close (1)

      write (*,*) 'What is your lamp''s filename for
     & the VV decay?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 13 i=1,9
13    read(1,*)

      do 15 i=1,ndata1

      read(1,*) x,e1(i)

15    continue

      close (1)


      write (*,*) 'What is your VV decay''s filename?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 14 i=1,9
14    read(1,*)

      do 16 i=1,ndata1

      read(1,*) x,y1(i)

16    continue

      close (1)

      write (*,*) 'How many chanels do you want to work with
     & for the VH decay?'
      read (*,*) ndata2


c     write (*,*) 'What is your file''s name for VH?'
c     read(*,10) DD2

c     open(1,file=DD2,status='old')

c     do 16 i=1,ndata2

c     read(1,*) e2(i),y2(i)
```

```
c16   continue

c     close (1)


      write (*,*) 'What is your lamp''s filename for
     & the VH decay?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 313 i=1,9
313   read(1,*)

      do 315 i=1,ndata2

      read(1,*) x,e2(i)

315   continue

      close (1)


      write (*,*) 'What is your VH decay''s filename?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 314 i=1,9
314   read(1,*)

      do 316 i=1,ndata2

      read(1,*) x,y2(i)

316   continue

      close (1)

      write (*,*) 'What is the lifetime of the reference compound
     & for the VV decay?'
      read(*,*) taur1

      write (*,*) 'What is your time per chanel for the VV
     & decay?'
      read(*,*) tpc1

      taur1 = exp(-tpc1/taur1)

      alold2 = e1(1)

      do 40 i=2,ndata1
```

```
      alold1 = e1(i)
      e1(i) = e1(i) - alold2*taur1
      alold2 = alold1

      if(e1(i).lt.0) e1(i)=0.0001

40    continue


      write (*,*) 'What is the lifetime of the reference compound
     & for the VH decay?'
      read(*,*) taur2

      write (*,*) 'What is your time per chanel for the VH
     & decay?'
      read(*,*) tpc2

      taur2 = exp(-tpc2/taur2)

      alold2 = e2(1)

      do 41 i=2,ndata2

      alold1 = e2(i)
      e2(i) = e2(i) - alold2*taur2
      alold2 = alold1

      if(e2(i).lt.0) e2(i)=0.0001

41    continue

      imax = 0

      do 500 i=2,ndata1

      if (e1(i).gt.emax) then
      imax = i
      emax = e1(i)
      endif

500   continue


      write (*,511)
      write (*,*) '_____________________________________________'
511   format(10H   channel,10H      lamp,10H     decay)

      do 520 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e1(k),y1(k)
512   format (3H   ,I4,3H   ,2(2H  ,F7.1,1H ))
```

```
      endif
520   continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VV decay?'
      read(*,*) nstart1(3)

      write (*,*) 'At which channel does the background noise start
     & for the VV decay?'
      read(*,*) nstart1(1)

      write (*,*) 'At which channel does the background noise end
     & for the VV decay?'
      read(*,*) nstart1(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 860 i=nback1,nback2

c     sume = sume + e1(i)
c     sumy = sumy + y1(i)

c860  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 870 i=1,ndata1

c     e1(i) = e1(i) - sume
c     y1(i) = y1(i) - sumy

c     if(e1(i).lt.0.0) e1(i) = 0.00001
c     if(y1(i).lt.0.0) y1(i) = 0.00001

c870  continue


      imax = 0
      emax = 0.0

      do 501 i=2,ndata2

      if (e2(i).gt.emax) then
      imax = i
      emax = e2(i)
      endif

501   continue
```

```
      write (*,511)
      write (*,*) '_____________________________________________________'

      do 1522 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e2(k),y2(k)
      endif
1522  continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VH decay?'
      read(*,*) nstart2(3)

      write (*,*) 'At which channel does the background noise start
     & for the VH decay?'
      read(*,*) nstart2(1)

      write (*,*) 'At which channel does the background noise end
     & for the VH decay?'
      read(*,*) nstart2(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 861 i=nback1,nback2

c     sume = sume + e2(i)
c     sumy = sumy + y2(i)

c861  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 871 i=1,ndata2

c     e2(i) = e2(i) - sume
c     y2(i) = y2(i) - sumy

c     if(e2(i).lt.0.0) e2(i) = 0.00001
c     if(y2(i).lt.0.0) y2(i) = 0.00001

c871  continue

        write (*,*) 'How many decay times do you need to represent
     & the decay of the dye bound to the macromolecule.'
        read (*,*) nexp1

        tauav1 = 0.0
```

```
        do 400 k1=1,nexp1
        write (*,*) 'What is the',k1,'th decaytime?'
        read (*,*) tau1(k1)
        write (*,*) 'What is its normalized pre-exponential factor?'
        read (*,*) a1(k1)

        tauav1 = tauav1 + a1(k1)*tau1(k1)

400     continue


        do 410 i1=1,10000

        f1pa(i1) = 0.0
        f1pe(i1) = 0.0

        do 415 k1=1,nexp1

        ai1 = i1
        f1pa(i1) = f1pa(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc1/tau1(k1))
        f1pe(i1) = f1pe(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc2/tau1(k1))

415     continue

410     continue



      write (*,*) 'What is the first decay time in the anisotropy?'
      read (*,*) a(2)

      write (*,*) 'What is its pre-exponential factor?'
      read(*,*) a(1)

      write (*,*) 'What is the second decay time in the anisotropy?'
      read (*,*) a(4)

      write (*,*) 'What is its pre-exponential factor?'
      read(*,*) a(3)

        write (*,*) 'What is your scaling factor with the instrument
     & lamp for the VV decay?'
        read (*,*) a(5)

        write (*,*) 'What is your scaling factor with the instrument
     & lamp for the VH decay?'
        read (*,*) a(10)

      a(6) = .1
```

```
      a(7) = .1

      a(8) = 10.0
      a(9) = 10.0

      do 524 i=1,mfit
      lista(i) =i
524   continue

513   alambda = -0.1
      itera = 0
      kchi = 0

514   continue



      if (itera.ne.0) goto 540

515   write (*,521)
521   format(10Hiteration ,20H    amplitude       ,
     &20H    lifetime        ,20H    scattering      ,10H chisquare)
      write(*,*)'_____________________________________________________
     ____________'

c     write (*,*) 'just before entering mrqmin. taum = ',taum

      a(2) = sqrt(a(2))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))

540   continue

      call mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,covar,alpha,
     &      nca,chisq,alambda,nstart1,nstart2,e1,e2,itera,da,beta
     &  ,dyda,tpc1,tpc2,f1pa,f1pe,aG)

      a(2) = a(2)*a(2)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)

      write (*,*)

      write (*,542) itera,a(1),a(2),a(5),a(10),
     &      chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
```

```
542   format(I5,5H     ,4(3H   ,F10.3,7H       ),F10.3)

      write (*,543) a(8),a(9)
543   format(10HBackground,2(3H   ,F10.3,7H       ))

      write (*,545) a(6),a(7)
545   format(10HScattering,2(3H   ,F10.3,7H       ))

      write (*,523) a(3),a(4)
523   format(10H          ,2(3H   ,F10.3,7H       ))
        write (*,544) a(5)/a(10)
544   format(10HG-factor  ,3H   ,F10.3,7H       )

c     write(*,*) 'alambda = ',alambda

      a(2) = sqrt(a(2))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))

      if (itera.eq.1) goto 514
      if(chicca.eq.chisq) kchi = kchi+1
      if (chicca.ne.chisq) chicca = chisq
      if(chicca.ne.chisq) kchi = 0
      if (kchi.gt.20) goto 550
      goto 540

550   continue

c550  write (*,*) 'Do you want to try new amplitudes and lifetimes
c     & or starting analysis channel?'
c     write (*,*) 'yes = 1'
c     read(*,*) nn
c     if (nn.eq.1) then

c     do 525 i=1,nexp
c           ii = nexp+i
c           write (*,*) 'what is your new ',i,'th lifetime?'
c           read(*,*) a(ii)
c           write (*,*) 'what is your new ',i,'th amplitude?'
c           read(*,*) a(i)
c525  continue

c     write (*,*) 'what is your new scattering factor correction?'
c     read(*,*) a(mfit)

c           write (*,*) 'what is your new starting analysis channel?'
c           read(*,*) nstart
c           goto 513
c     else
            write (*,*) 'this is the end, my friend!'
```

```
c     endif


      do 1000 i=1,ndata1

      call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,aG)

      ym(i) = ymod
      res1(i) = y1(i) - ymod

      if (ymod.lt.1.0) then

            res1(i) = 0.0

      else

            res1(i) = res1(i)/sqrt(ymod)

      endif

      if(i.lt.nstart1(3)) res1(i) = 0.0

1000  continue

      do 1010 i=1,ndata1

      sum = sum + res1(i)*res1(i)

1010  continue

      n3 = ndata1 - nstart1(3) + 1
      an3 = ndata1 - nstart1(3) + 1

      do 1020 j=nstart1(3),n3-1

            do 1030 i=nstart1(3),ndata1-j

            auto1(j) = auto1(j) + res1(i)*res1(i+j)

1030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto1(j) = an3*auto1(j)/(am*sum)

1020  continue
```

```
      open(2,file='plot1.dat',status='old')

      do 1040 i=1,ndata1

      time = i*tpc1
      write (2,1050) time,e1(i),y1(i),ym(i),res1(i),auto1(i)

1050  format (4F10.2,2E12.3)

1040  continue

      close(2)

      a(2) = a(2)*a(2)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)

        aG = a(5)/a(10)

      open(2,file='plotres1',status='old')

      atcha = 0.0
      write(2,10) DD1
      write (2,1070) a(1),a(2)
      write (2,1070) a(3),a(4)
      write (2,1070) a(5)/a(10),atcha
      write (2,1070) frac,tauav1
      write (2,1070) atcha,atcha
      write (2,1080) a(8)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart1(3)

1070  format(2F10.5)
1075  format (I5)
1080  format (F10.5)

      close(2)

      a(2) = sqrt(a(2))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))


      do 2000 i=1,ndata2
```

```
      call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,f1pe,aG)

      ym(i) = ymod
      res2(i) = y2(i) - ymod

      if (ymod.lt.1.0) then

            res2(i) = 0.0

      else

            res2(i) = res2(i)/sqrt(ymod)

      endif

      if(i.lt.nstart2(3)) res2(i) = 0.0

2000  continue

      do 2010 i=1,ndata2

      sum = sum + res2(i)*res2(i)

2010  continue

      n3 = ndata2 - nstart2(3) + 1
      an3 = ndata2 - nstart2(3) + 1

      do 2020 j=nstart2(3),n3-1

            do 2030 i=nstart2(3),ndata2-j

            auto2(j) = auto2(j) + res2(i)*res2(i+j)

2030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto2(j) = an3*auto2(j)/(am*sum)

2020  continue

      open(2,file='plot2.dat',status='old')

      do 2040 i=1,ndata2
```

```
      time = i*tpc2
      write (2,1050) time,e2(i),y2(i),ym(i),res2(i),auto2(i)

2040  continue

      close(2)


      a(2) = a(2)*a(2)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)

      atcha = 0.0

      open(2,file='plotres2',status='old')

      atcha = 0.0
      write(2,10) 'aniso02d'
      write(2,10) DD2
      write (2,1070) a(1),a(2)
      write (2,1070) a(3),a(4)
      write (2,1070) a(5)/a(10),atcha
      write (2,1070) frac,tauav1
      write (2,1070) atcha,atcha
      write (2,1080) a(9)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart2(3)

      close(2)


      end







      subroutine foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1
     & ,f1pa,aG)

      implicit double precision (a-h,o-z)

      real*8 a(ma),y1(10000),e1(10000),dyda(ma),f1pa(ndata1)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc1,ymod,ymod1,
     & ymod2,ymod3,tau0

c     if (i.eq.1) then
```

```
c     write (*,*) 'We are in foncs1!'
c     write (*,*) 'taum = ',taum
c     endif

c     if (i.eq.1) then
c     write (*,*) 'in foncs1, tpc1 = ',tpc1
c     endif


      a(2) = a(2)*a(2)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue

      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc1/a(2))
      tadi1 = 1.0

      tod1 = dexp(-tpc1/a(4))
      todi1 = 1.0

      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
      ymod2 = ymod2 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(1)*tadi1
      ymod3 = ymod3 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(3)*todi1

      dyda(2) = dyda(2) + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*a(1)*tadi1*(akk-1.0)
      dyda(4) = dyda(4) + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*a(3)*todi1*(akk-1.0)
```

```
      tadi1 = tadi1*tad1
      todi1 = todi1*tod1

20    continue

25    continue

      dyda(1) = ymod2/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc1/(a(2)*a(2))
      dyda(3) = ymod3/a(3)
        dyda(4) = 2.0*sqrt(a(4))*dyda(4)*tpc1/(a(4)*a(4))
      dyda(5) = 2.0*sqrt(a(5))*(ymod1+ymod2
     & +ymod3)/a(5)
      dyda(6) = e1(i)
      dyda(7) = 0.0
      dyda(8) = 2.0*sqrt(a(8))
      dyda(9) = 0.0

      ymod = ymod1 + ymod2 + ymod3
     & + a(6)*e1(i) + a(8)

      a(2) = sqrt(a(2))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))

c        if(i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif

c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)
      return

      end


      subroutine foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2
     & ,f1pe,aG)

      implicit double precision (a-h,o-z)

      real*8 a(ma),e2(10000),dyda(ma),f1pe(ndata2)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc2,ymod,ymod1,
     & ymod2,ymod3,tau0,aa1,aa2,aa3,aa4,aa5,aa6,akk,tauS


      a(2) = a(2)*a(2)
```

```
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue



      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc2/a(2))
      tadi1 = 1.0

      tod1 = dexp(-tpc2/a(4))
      todi1 = 1.0


      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
      ymod2 = ymod2 - amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
     & *a(1)*tadi1
      ymod3 = ymod3 - amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
     & *a(3)*todi1

      dyda(2) = dyda(2) - amol*e2(i-k+1)*tpc2*a(10)
     & *f1pe(k)*a(1)*tadi1*(akk-1.0)
      dyda(4) = dyda(4) - amol*e2(i-k+1)*tpc2*a(10)
     & *f1pe(k)*a(3)*todi1*(akk-1.0)


      tadi1 = tadi1*tad1
      todi1 = todi1*tod1

20    continue
```

```
25    continue

      dyda(1) = ymod2/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc2/(a(2)*a(2))
      dyda(3) = ymod3/a(3)
      dyda(4) = 2.0*sqrt(a(4))*dyda(4)*tpc2/(a(4)*a(4))
      dyda(5) = 0.0
      dyda(6) = 0.0
      dyda(7) = e2(i)
      dyda(8) = 0.0
      dyda(9) = 2.0*sqrt(a(9))
      dyda(10) = 2.0*sqrt(a(10))*(ymod1+ymod2
     & +ymod3)/a(10)

      ymod = ymod1 + ymod2 + ymod3
     & + a(7)*e2(i) + a(9)

      a(2) = sqrt(a(2))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))

c        if (i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif
c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)


      return

      end







      subroutine mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & covar,alpha,nca,chisq,alamda,nstart1,nstart2,e1,e2,itera,
     & da,beta,dyda,tpc1,tpc2,f1pa,f1pe,aG)


      implicit double precision (a-h,o-z)

      parameter (mmax=20)
```

```
      dimension
y1(ndata1),y2(ndata2),a(ma),lista(ma),e1(ndata1),e2(ndata2),
     &
      covar(mfit,mfit),alpha(mfit,mfit),atry(mmax),beta(mfit),da(mfit)
     &  ,dyda(mfit),nstart1(3),nstart2(3),f1pa(ndata1),
     &  f1pe(ndata2)

      real*8 taum,chisq,alamda,tpc1,tpc2,tauS

      save ochisq

c     write (*,*) 'we are in mrqmin. taum = ',taum
c     write (*,*) 'we are in mrqmin. tpc2 = ',tpc2

      if (alamda.lt.0) then
            kk = mfit+1
            do 12 j=1,ma
                  ihit=0
                  do 11 k=1,mfit
                        if(lista(k).eq.j) ihit=ihit+1
11                continue
                  if(ihit.eq.0) then
                        lista(kk)=j
                        kk = kk+1
                  else if (ihit.gt.1) then
                        pause 'improper permutation in lista'
                  endif
12          continue
      if(kk.ne.(ma+1)) pause 'improper permutation in lista'
      alamda = 0.001

      call mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,alpha,beta,nca,chisq
     &     ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG)

c     write(*,43) itera,a(1),a(2),chisq/(ndata1+ndata2-nstart1-nstart2-
mfit)
c43   format(I5,5H     ,3(3H   ,F10.3,7H       ))


      do 13 j=1,ma
            atry(j)=a(j)
13    continue
      endif
      ochisq = chisq
      itera = itera+1


      do 15 j=1,mfit
            do 14 k=1,mfit
                  covar(j,k)=alpha(j,k)
14          continue
      covar(j,j)=alpha(j,j)*(1.+alamda)
      da(j) = beta(j)
```

```
15    continue

c     write (*,*) 'Just before Gaussj.'

      call gaussj(covar,mfit,nca,da,1,1)

c     write (*,*) 'Just after Gaussj.'


      if(alamda.eq.0) then
            call covsrt(covar,nca,ma,lista,mfit)
            return
      endif
      do 16 j=1,mfit
            atry(lista(j)) = a(lista(j))+da(j)
16    continue


      call
mrqcof(y1,y2,ndata1,ndata2,atry,ma,lista,mfit,covar,da,nca,chisq
     &  ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG)


      if (chisq.lt.ochisq) then
      alamda = 0.1*alamda
      ochisq=chisq
      do 18 j=1,mfit
            do 17 k=1,mfit
                  alpha(j,k)=covar(j,k)
17          continue
      beta(j)=da(j)
      a(lista(j))=atry(lista(j))
18    continue
      else
      alamda = 10.*alamda
      chisq=ochisq
      endif
      return
      end


      subroutine mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & alpha,beta,nalp,chisq
     &     ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe
     &  ,aG)

      implicit double precision (a-h,o-z)

      dimension y1(10000),y2(10000),alpha(nalp,nalp),beta(mfit),
     &  dyda(mfit),lista(mfit),a(ma),e1(10000),e2(10000)
     & ,nstart1(3),nstart2(3),f1pa(ndata1)
     & ,f1pe(ndata2)
      real*8 taum,chisq,tpc1,tpc2,tauS
```

```
c     write (*,*) 'we are in mrqcof. taum = ',taum
c     write (*,*) 'we are in mrqcof. tpc2 = ',tpc2

      do 112 j=1,mfit
            do 111 k=1,j
            alpha(j,k) = 0.
111         continue
      beta(j) = 0.
112   continue
      chisq=0.

120   do 115 i=nstart1(3),ndata1

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 114 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 113 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
113                     continue
            beta(j)=beta(j)+dy*wt
114         continue
      chisq=chisq+dy*dy*sig2i
115   continue

320   do 315 i=nstart1(1),nstart1(2)

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 314 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 313 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
313                     continue
            beta(j)=beta(j)+dy*wt
314         continue
      chisq=chisq+dy*dy*sig2i
315   continue

c     write (*,*) 'Chisq value after monomer: ',chisq

220   do 215 i=nstart2(3),ndata2

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
```

```
     & f1pe,aG)

c          if (i.lt.nstart2) goto 215
           sig2i = 1./y2(i)
           dy = y2(i)-ymod
           do 214 j=1,mfit
                wt=dyda(lista(j))*sig2i
                     do 213 k=1,j
                     alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
213                  continue
           beta(j)=beta(j)+dy*wt
214        continue
      chisq=chisq+dy*dy*sig2i
215   continue

420   do 415 i=nstart2(1),nstart2(2)

           call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe,aG)

c          if (i.lt.nstart2) goto 215
           sig2i = 1./y2(i)
           dy = y2(i)-ymod
           do 414 j=1,mfit
                wt=dyda(lista(j))*sig2i
                     do 413 k=1,j
                     alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
413                  continue
           beta(j)=beta(j)+dy*wt
414        continue
      chisq=chisq+dy*dy*sig2i
415   continue

c     write (*,*) 'Chisq value after excimer: ',chisq


      do 117 j=2,mfit
           do 116 k=1,j-1
           alpha(k,j)=alpha(j,k)
116        continue
117   continue
      return
      end




      subroutine covsrt(covar,ncvm,ma,lista,mfit)

      implicit double precision (a-h,o-z)
```

```
      dimension covar(ncvm,ncvm),lista(mfit)

      do 212 j=1,ma-1
            do 211 i=j+1,ma
            covar(i,j) = 0.
211         continue
212   continue
      do 214 i=1,mfit-1
            do 213 j=i+1,mfit
                  if(lista(j).gt.lista(i)) then
                        covar(lista(j),lista(i))=covar(i,j)
                  else
                        covar(lista(i),lista(j))=covar(i,j)
                  endif
213         continue
214   continue
      swap=covar(1,1)
      do 215 j=1,ma
      covar(1,j) = covar(j,j)
      covar(j,j) = 0.
215   continue
      covar(lista(1),lista(1))=swap
      do 216 j=2,mfit
      covar(lista(j),lista(j))=covar(1,j)
216   continue
      do 218 j=2,ma
            do 217 i=1,j-1
            covar(i,j)=covar(j,i)
217   continue
218   continue
      return
      end




      subroutine gaussj(a,n,np,b,m,mp)

      implicit double precision (a-h,o-z)

      parameter (nmax=50)
      dimension a(np,np),b(np),ipiv(nmax),indxr(nmax),indxc(nmax)

      do 311 j=1,n
            ipiv(j) = 0
311   continue

      do 322 i=1,n
            big = 0.
```

```
            do 313 j=1,n

                  if (ipiv(j).ne.1) then

                  do 312 k=1,n

                  if(ipiv(k).eq.0) then

                        if (abs(a(j,k)).ge.big) then
                        big = abs(a(j,k))
                        irow = j
                        icol = k
                        endif

                        else if (ipiv(k).gt.1) then
                        pause 'singular matrix'
                        endif

312               continue

                  endif

313         continue

      ipiv(icol) = ipiv(icol) + 1

      if (irow.ne.icol) then

            do 314 l=1,n
                  dum = a(irow,l)
                  a(irow,l) = a(icol,l)
                  a(icol,l) = dum
314         continue

                  dum = b(irow)
                  b(irow) = b(icol)
                  b(icol) = dum

      endif

      indxr(i) = irow
      indxc(i) = icol

      if (a(icol,icol).eq.0)  pause 'singular matrix'

      pivinv = 1./a(icol,icol)

      a(icol,icol) = 1.

      do 316 l=1,n
            a(icol,l) = a(icol,l)*pivinv
316   continue
```

```
            b(icol) = b(icol)*pivinv

      do 321 ll=1,n
            if (ll.ne.icol) then

                  dum = a(ll,icol)
                  a(ll,icol) = 0.

                  do 318 l=1,n
                        a(ll,l) = a(ll,l) - a(icol,l)*dum
318               continue

                        b(ll) = b(ll) - b(icol)*dum

            endif

321   continue

322   continue

      do 324 l=n,1,-1
            if (indxr(l).ne.indxc(l)) then

                  do 323 k=1,n
                        dum = a(k,indxr(l))
                        a(k,indxr(l)) = a(k,indxc(l))
                        a(k,indxc(l)) = dum
323               continue

            endif

324   continue

      return
      end
```

K] Listing of *aniso03g*

```
c     Program created April 9, 2015
c     This program fits the I(para;para) and I(para;per) decays
c       globally assuming a triexponential anisotropy.
c       There is NO function for non-covalently attached dyes.
c       There is a function for covalently attached dyes.
c       Pre-exponential factors of the anisotropy can be negative.
c       the G factor is omitted and it is calculated.
c       The diffusion coefficients are optimized.

      implicit double precision (a-h,o-z)

      parameter (mfit=11,ma=11,nca=11)

      dimension y1(10000),lista(mfit),a(mfit),
     & covar(mfit,mfit),alpha(mfit,mfit),
     & e1(10000),da(mfit),beta(mfit)
     & ,dyda(mfit),ym(10000),res1(10000),auto1(10000)
     & ,e2(10000),y2(10000),res2(10000),auto2(10000)
     & ,nstart1(3),nstart2(3)
     & ,a1(10),a2(10),tau1(10),tau2(10),f1pa(10000),f1pe(10000)

      real*8 taum,chisq,tpc1,tpc2


10    format(A)
      character *30 DD1,DD2

      write (*,*) 'How many chanels do you want to work with
     & for the VV decay?'
      read (*,*) ndata1

c     write (*,*) 'What is the name of your lamp file ?'
```

```
c     read(*,10) DD

c     open (1,file=DD,status='old')

c     read(1,*) (e(i),i=1,ndata)

c     close(1)

c     write (*,*) 'what is the name of your fluorescence decay?'
c     read(*,10) DD

c     open(1,file=DD,status='old')

c     read(1,*) (y(i),i=1,ndata)

c     close(1)

c     write (*,*) 'What is your file''s name for VV?'
c     read(*,10) DD1

c     open(1,file=DD1,status='old')

c     do 15 i=1,ndata1

c     read(1,*) e1(i),y1(i)

c15   continue

c     close (1)

      write (*,*) 'What is your lamp''s filename for
     & the VV decay?'
      read(*,10) DD1

      open(1,file=DD1,status='old')

      do 13 i=1,9
13    read(1,*)

      do 15 i=1,ndata1

      read(1,*) x,e1(i)

15    continue

      close (1)


      write (*,*) 'What is your VV decay''s filename?'
      read(*,10) DD1

      open(1,file=DD1,status='old')
```

```
      do 14 i=1,9
14    read(1,*)

      do 16 i=1,ndata1

      read(1,*) x,y1(i)

16    continue

      close (1)

      write (*,*) 'How many chanels do you want to work with
     & for the VH decay?'
      read (*,*) ndata2


      write (*,*) 'What is your file''s name for VH?'
      read(*,10) DD2

c     open(1,file=DD2,status='old')

c     do 16 i=1,ndata2

c     read(1,*) e2(i),y2(i)

c16   continue

c     close (1)


      write (*,*) 'What is your lamp''s filename for
     & the VH decay?'
      read(*,10) DD2

      open(1,file=DD2,status='old')

      do 313 i=1,9
313   read(1,*)

      do 315 i=1,ndata2

      read(1,*) x,e2(i)

315   continue

      close (1)


      write (*,*) 'What is your VH decay''s filename?'
      read(*,10) DD2

      open(1,file=DD2,status='old')
```

```
      do 314 i=1,9
314   read(1,*)

      do 316 i=1,ndata2

      read(1,*) x,y2(i)

316   continue

      close (1)

      write (*,*) 'What is the lifetime of the reference compound
     & for the VV decay?'
      read(*,*) taur1

      write (*,*) 'What is your time per chanel for the VV
     & decay?'
      read(*,*) tpc1

      taur1 = exp(-tpc1/taur1)

      alold2 = e1(1)

      do 40 i=2,ndata1

      alold1 = e1(i)
      e1(i) = e1(i) - alold2*taur1
      alold2 = alold1

      if(e1(i).lt.0) e1(i)=0.0001

40    continue


      write (*,*) 'What is the lifetime of the reference compound
     & for the VH decay?'
      read(*,*) taur2

      write (*,*) 'What is your time per chanel for the VH
     & decay?'
      read(*,*) tpc2

      taur2 = exp(-tpc2/taur2)

      alold2 = e2(1)

      do 41 i=2,ndata2

      alold1 = e2(i)
      e2(i) = e2(i) - alold2*taur2
      alold2 = alold1

      if(e2(i).lt.0) e2(i)=0.0001
```

```
41    continue

      imax = 0

      do 500 i=2,ndata1

      if (e1(i).gt.emax) then
      imax = i
      emax = e1(i)
      endif

500   continue


      write (*,511)
      write (*,*) '_____________________________________________________'
511   format(10H   channel,10H      lamp,10H     decay)

      do 520 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e1(k),y1(k)
512   format (3H   ,I4,3H   ,2(2H  ,F7.1,1H ))
      endif
520   continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VV decay?'
      read(*,*) nstart1(3)

      write (*,*) 'At which channel does the background noise start
     & for the VV decay?'
      read(*,*) nstart1(1)

      write (*,*) 'At which channel does the background noise end
     & for the VV decay?'
      read(*,*) nstart1(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 860 i=nback1,nback2

c     sume = sume + e1(i)
c     sumy = sumy + y1(i)

c860  continue

c     sume = sume/anback
c     sumy = sumy/anback
```

```
c     do 870 i=1,ndata1

c     e1(i) = e1(i) - sume
c     y1(i) = y1(i) - sumy

c     if(e1(i).lt.0.0) e1(i) = 0.00001
c     if(y1(i).lt.0.0) y1(i) = 0.00001

c870  continue


      imax = 0
      emax = 0.0

      do 501 i=2,ndata2

      if (e2(i).gt.emax) then
      imax = i
      emax = e2(i)
      endif

501   continue


      write (*,511)
      write (*,*) '_____________________________________________________'

      do 1522 k=imax-15,imax+10

      if (k.gt.1) then
      write (*,512) k,e2(k),y2(k)
      endif
1522  continue

      write (*,*) 'From which channel do you wish to start
     &     your analysis for the VH decay?'
      read(*,*) nstart2(3)

      write (*,*) 'At which channel does the background noise start
     & for the VH decay?'
      read(*,*) nstart2(1)

      write (*,*) 'At which channel does the background noise end
     & for the VH decay?'
      read(*,*) nstart2(2)

c     sume = 0.0
c     sumy = 0.0

c     anback = nback2-nback1 + 1

c     do 861 i=nback1,nback2
```

```
c     sume = sume + e2(i)
c     sumy = sumy + y2(i)

c861  continue

c     sume = sume/anback
c     sumy = sumy/anback

c     do 871 i=1,ndata2

c     e2(i) = e2(i) - sume
c     y2(i) = y2(i) - sumy

c     if(e2(i).lt.0.0) e2(i) = 0.00001
c     if(y2(i).lt.0.0) y2(i) = 0.00001

c871  continue

        write (*,*) 'How many decay times do you need to represent
     & the decay of the dye bound to the macromolecule.'
        read (*,*) nexp1

        tauav1 = 0.0

        do 400 k1=1,nexp1
        write (*,*) 'What is the',k1,'th decaytime?'
        read (*,*) tau1(k1)
        write (*,*) 'What is its normalized pre-exponential factor?'
        read (*,*) a1(k1)

        tauav1 = tauav1 + a1(k1)*tau1(k1)

400     continue


        do 410 i1=1,10000

        f1pa(i1) = 0.0
        f1pe(i1) = 0.0

        do 415 k1=1,nexp1

        ai1 = i1
        f1pa(i1) = f1pa(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc1/tau1(k1))
        f1pe(i1) = f1pe(i1) + a1(k1)*
     & dexp(-(ai1-1.0)*tpc2/tau1(k1))

415     continue

410     continue
```

```
      write (*,*) 'Give an estimate of Dpara.'
      read (*,*) a(2)

      write (*,*) 'Give an estimate of Dper.'
      read (*,*) a(4)

      taurot1 = 1.0/(a(2)+5.0*a(4))
      taurot2 = 1.0/(4.0*a(2)+2.0*a(4))
      taurot3 = 1.0/(6.0*a(4))

      write (*,*) 'taurot1 = ',taurot1
      write (*,*) 'taurot2 = ',taurot2
      write (*,*) 'taurot3 = ',taurot3

     write (*,*) 'What is the pre-exponential factor for taurot1 =
     & ',taurot1,'?'
     read(*,*) a(1)

     write (*,*) 'What is the pre-exponential factor for taurot2 =
     & ',taurot2,'?'
     read(*,*) a(3)

     write (*,*) 'What is the pre-exponential factor for taurot3 =
     & ',taurot3,'?'
     read(*,*) a(11)

      write (*,*) 'What is your scaling factor with the instrument
     & lamp for the VV decay?'
      read (*,*) a(5)

      write (*,*) 'What is your scaling factor with the instrument
     & lamp for the VH decay?'
      read (*,*) a(10)

     a(6) = .1
     a(7) = .1

     a(8) = 10.0
     a(9) = 10.0

     do 524 i=1,mfit
     lista(i) =i
524  continue

513  alambda = -0.1
     itera = 0
     kchi = 0

514  continue
```

```
      if (itera.ne.0) goto 540

515   write (*,521)
521   format(10Hiteration ,20H    amplitude       ,
     &20H    lifetime        ,20H    scattering      ,10H chisquare)
      write(*,*)'_____________________________________________________
     ____________'

c     write (*,*) 'just before entering mrqmin. taum = ',taum

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))
      a(11) = sqrt(a(11))

540   continue

      call mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,covar,alpha,
     &      nca,chisq,alambda,nstart1,nstart2,e1,e2,itera,da,beta
     &  ,dyda,tpc1,tpc2,f1pa,f1pe,aG)

      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)
      a(11) = a(11)*a(11)

      write (*,*)

      write (*,542) itera,a(5),a(10),a(2),a(4),
     &      chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
542   format(I5,5H     ,4(3H   ,F10.3,7H       ),F10.3)

      write (*,543) a(8),a(9)
543   format(10HBackground,2(3H   ,F10.3,7H       ))

      write (*,545) a(6),a(7)
545   format(10HScattering,2(3H   ,F10.3,7H       ))

        write (*,544) a(5)/a(10)
544   format(10HG-factor  ,3H   ,F10.3,7H       )

        write (*,551) a(1),1.0/(a(2)+5.0*a(4))
```

```
        write (*,552) a(3),1.0/(4.0*a(2)+2*a(4))
        write (*,553) a(11),1.0/(6.0*a(4))

551   format(10Hro1 =     ,3H   ,F10.3,7H       ,10Htaurot1 = ,F10.3)
552   format(10Hro2 =     ,3H   ,F10.3,7H       ,10Htaurot2 = ,F10.3)
553   format(10Hro3 =     ,3H   ,F10.3,7H       ,10Htaurot3 = ,F10.3)

c     write(*,*) 'alambda = ',alambda

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))
      a(11) = sqrt(a(11))

      if (itera.eq.1) goto 514
      if(chicca.eq.chisq) kchi = kchi+1
      if (chicca.ne.chisq) chicca = chisq
      if(chicca.ne.chisq) kchi = 0
      if (kchi.gt.20) goto 550
      goto 540

550   continue

c550  write (*,*) 'Do you want to try new amplitudes and lifetimes
c     & or starting analysis channel?'
c     write (*,*) 'yes = 1'
c     read(*,*) nn
c     if (nn.eq.1) then

c     do 525 i=1,nexp
c           ii = nexp+i
c           write (*,*) 'what is your new ',i,'th lifetime?'
c           read(*,*) a(ii)
c           write (*,*) 'what is your new ',i,'th amplitude?'
c           read(*,*) a(i)
c525  continue

c     write (*,*) 'what is your new scattering factor correction?'
c     read(*,*) a(mfit)

c           write (*,*) 'what is your new starting analysis channel?'
c           read(*,*) nstart
c           goto 513
c     else
            write (*,*) 'this is the end, my friend!'
c     endif
```

```
      do 1000 i=1,ndata1

      call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,aG)

      ym(i) = ymod
      res1(i) = y1(i) - ymod

      if (ymod.lt.1.0) then

            res1(i) = 0.0

      else

            res1(i) = res1(i)/sqrt(ymod)

      endif

      if(i.lt.nstart1(3)) res1(i) = 0.0

1000  continue

      do 1010 i=1,ndata1

      sum = sum + res1(i)*res1(i)

1010  continue

      n3 = ndata1 - nstart1(3) + 1
      an3 = ndata1 - nstart1(3) + 1

      do 1020 j=nstart1(3),n3-1

            do 1030 i=nstart1(3),ndata1-j

            auto1(j) = auto1(j) + res1(i)*res1(i+j)

1030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto1(j) = an3*auto1(j)/(am*sum)

1020  continue

      open(2,file='plot1.dat',status='old')
```

```
      do 1040 i=1,ndata1

      time = i*tpc1
      write (2,1050) time,e1(i),y1(i),ym(i),res1(i),auto1(i)

1050  format (4F10.2,2E12.3)

1040  continue

      close(2)

      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)
      a(11) = a(11)*a(11)

        aG = a(5)/a(10)

      open(2,file='plotres1',status='old')

      atcha = 0.0
      write(2,10) DD1
      write (2,1070) a(1),1.0/(a(2)+5.0*a(4))
      write (2,1070) a(3),1.0/(4.0*a(2)+2.0*a(4))
      write (2,1070) a(11),1.0/(6.0*a(4))
      write (2,1070) a(5)/a(10),atcha
      write (2,1070) frac,tauav1
      write (2,1080) a(8)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart1(3)

1070  format(2F10.5)
1075  format (I5)
1080  format (F10.5)

      close(2)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))
      a(11) = sqrt(a(11))
```

```
      do 2000 i=1,ndata2

      call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,f1pe,aG)

      ym(i) = ymod
      res2(i) = y2(i) - ymod

      if (ymod.lt.1.0) then

            res2(i) = 0.0

      else

            res2(i) = res2(i)/sqrt(ymod)

      endif

      if(i.lt.nstart2(3)) res2(i) = 0.0

2000  continue

      do 2010 i=1,ndata2

      sum = sum + res2(i)*res2(i)

2010  continue

      n3 = ndata2 - nstart2(3) + 1
      an3 = ndata2 - nstart2(3) + 1

      do 2020 j=nstart2(3),n3-1

            do 2030 i=nstart2(3),ndata2-j

            auto2(j) = auto2(j) + res2(i)*res2(i+j)

2030        continue

      am = j
      am = an3 - am

      if(am.eq.0.0) then

      write (*,*) 'there is a problem!'

      endif

      auto2(j) = an3*auto2(j)/(am*sum)

2020  continue

      open(2,file='plot2.dat',status='old')
```

```
      do 2040 i=1,ndata2

      time = i*tpc2
      write (2,1050) time,e2(i),y2(i),ym(i),res2(i),auto2(i)

2040  continue

      close(2)


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)
      a(11) = a(11)*a(11)

      atcha = 0.0

      open(2,file='plotres2',status='old')

      atcha = 0.0
      write(2,10) 'aniso02d'
      write(2,10) DD2
      write (2,1070) a(1),1.0/(a(2)+5.0*a(4))
      write (2,1070) a(3),1.0/(4.0*a(2)+2.0*a(4))
      write (2,1070) a(11),1.0/(6.0*a(4))
      write (2,1070) a(5)/a(10),atcha
      write (2,1070) frac,tauav1
      write (2,1080) a(9)
      write (2,1080) chisq/(ndata1+ndata2-nstart1(3)-nstart2(3)+
     & (nstart1(2)-nstart1(1))+(nstart2(2)-nstart2(1))-mfit)
      write (2,1075) nstart2(3)

      close(2)


      end







      subroutine foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1
     & ,f1pa,aG)

      implicit double precision (a-h,o-z)
```

```
      real*8 a(ma),y1(10000),e1(10000),dyda(ma),f1pa(ndata1)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc1,ymod,ymod1,
     & ymod2,ymod3,tau0

c     if (i.eq.1) then
c     write (*,*) 'We are in foncs1!'
c     write (*,*) 'taum = ',taum
c     endif

c     if (i.eq.1) then
c     write (*,*) 'in foncs1, tpc1 = ',tpc1
c     endif


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)
      a(11) = a(11)*a(11)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue

      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc1*(a(2)+5.0*a(4)))
      tadi1 = 1.0

      tod1 = dexp(-tpc1*(4.0*a(2)+2.0*a(4)))
      todi1 = 1.0

      ted1 = dexp(-tpc1*(6.0*a(4)))
      tedi1 = 1.0

      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
```

```
      endif

      ymod1 = ymod1 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
      ymod2 = ymod2 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(1)*tadi1
      ymod3 = ymod3 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(3)*todi1
      ymod4 = ymod4 + amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)*
     & 2.0*a(11)*tedi1

      dyda(2) = dyda(2) - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*(a(1)*tadi1+4.0*a(3)*todi1)*(akk-1.0)
      dyda(4) = dyda(4) - amol*e1(i-k+1)*tpc1*a(5)*f1pa(k)
     & *2.0*(5.0*a(1)*tadi1+2.0*a(3)*todi1+6.0*a(11)*tedi1)
     & *(akk-1.0)

      tadi1 = tadi1*tad1
      todi1 = todi1*tod1
      tedi1 = tedi1*ted1

20    continue

25    continue

      dyda(1) = 2.0*sqrt(a(1))*ymod2/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc1
      dyda(3) = 2.0*sqrt(a(3))*ymod3/a(3)
        dyda(4) = 2.0*sqrt(a(4))*dyda(4)*tpc1
      dyda(5) = 2.0*sqrt(a(5))*(ymod1+ymod2
     & +ymod3+ymod4)/a(5)
      dyda(6) = e1(i)
      dyda(7) = 0.0
      dyda(8) = 2.0*sqrt(a(8))
      dyda(9) = 0.0
        dyda(10) = 0.0
      dyda(11) = 2.0*sqrt(a(11))*ymod4/a(11)

      ymod = ymod1 + ymod2 + ymod3 + ymod4
     & + a(6)*e1(i) + a(8)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))
      a(11) = sqrt(a(11))

c        if(i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif
```

```
c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)
      return

      end


      subroutine foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2
     & ,f1pe,aG)

      implicit double precision (a-h,o-z)

      real*8 a(ma),e2(10000),dyda(ma),f1pe(ndata2)
      real*8 tedi1,ted1,tadi1,tad1,tidi1,tid1,amol,tpc2,ymod,ymod1,
     & ymod2,ymod3,tau0,aa1,aa2,aa3,aa4,aa5,aa6,akk,tauS


      a(1) = a(1)*a(1)
      a(2) = a(2)*a(2)
      a(3) = a(3)*a(3)
      a(4) = a(4)*a(4)
      a(5) = a(5)*a(5)
      a(8) = a(8)*a(8)
      a(9) = a(9)*a(9)
      a(10) = a(10)*a(10)
      a(11) = a(11)*a(11)

      do 5 k=1,ma
      dyda(k) = 0.
5     continue



      ymod1 = 0.0
      ymod2 = 0.0
      ymod3 = 0.0
      ymod4 = 0.0

      if (i.eq.1) goto 25

      tad1 = dexp(-tpc2*(a(2)+5.0*a(4)))
      tadi1 = 1.0

      tod1 = dexp(-tpc2*(4.0*a(2)+2.0*a(4)))
      todi1 = 1.0

      ted1 = dexp(-tpc2*(6.0*a(4)))
      tedi1 = 1.0
```

```
      do 20 k=1,i

      akk = k

      if((k.eq.1).or.(k.eq.i)) then
      amol = 0.5
      else
      amol = 1.0
      endif

      ymod1 = ymod1 + amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
      ymod2 = ymod2 - amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
     & *a(1)*tadi1
      ymod3 = ymod3 - amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
     & *a(3)*todi1
      ymod4 = ymod4 - amol*e2(i-k+1)*tpc2*a(10)*f1pe(k)
     & *a(11)*tedi1

      dyda(2) = dyda(2) + amol*e2(i-k+1)*tpc2*a(10)
     & *f1pe(k)*(a(1)*tadi1+4.0*a(3)*todi1)*(akk-1.0)
      dyda(4) = dyda(4) + amol*e2(i-k+1)*tpc2*a(10)
     & *f1pe(k)*(5.0*a(1)*tadi1+2.0*a(3)*todi1+6.0*a(11)*tedi1)
     & *(akk-1.0)


      tadi1 = tadi1*tad1
      todi1 = todi1*tod1
      tedi1 = tedi1*ted1

20    continue

25    continue

      dyda(1) = 2.0*sqrt(a(1))*ymod2/a(1)
      dyda(2) = 2.0*sqrt(a(2))*dyda(2)*tpc2
      dyda(3) = 2.0*sqrt(a(3))*ymod3/a(3)
      dyda(4) = 2.0*sqrt(a(4))*dyda(4)*tpc2
      dyda(5) = 0.0
      dyda(6) = 0.0
      dyda(7) = e2(i)
      dyda(8) = 0.0
      dyda(9) = 2.0*sqrt(a(9))
      dyda(10) = 2.0*sqrt(a(10))*(ymod1+ymod2
     & +ymod3+ymod4)/a(10)
      dyda(11) = 2.0*sqrt(a(11))*ymod4/a(11)

      ymod = ymod1 + ymod2 + ymod3 + ymod4
     & + a(7)*e2(i) + a(9)

      a(1) = sqrt(a(1))
      a(2) = sqrt(a(2))
      a(3) = sqrt(a(3))
      a(4) = sqrt(a(4))
```

```
      a(5) = sqrt(a(5))
      a(8) = sqrt(a(8))
      a(9) = sqrt(a(9))
      a(10) = sqrt(a(10))
      a(11) = sqrt(a(11))

c        if (i.eq.300) then
c     write (*,*) i,ymod,ymod1,ymod2,ymod3,ymod4
c        endif
c     write (*,*) dyda(1),dyda(2),dyda(3),dyda(4),dyda(5)

c     do 111 ik=1,ma
c111  write (*,*) 'a(',ik,' = ',a(ik)


      return

      end







      subroutine mrqmin(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & covar,alpha,nca,chisq,alamda,nstart1,nstart2,e1,e2,itera,
     & da,beta,dyda,tpc1,tpc2,f1pa,f1pe,aG)


      implicit double precision (a-h,o-z)

      parameter (mmax=20)


      dimension
y1(ndata1),y2(ndata2),a(ma),lista(ma),e1(ndata1),e2(ndata2),
     &
      covar(mfit,mfit),alpha(mfit,mfit),atry(mmax),beta(mfit),da(mfit)
     &  ,dyda(mfit),nstart1(3),nstart2(3),f1pa(ndata1),
     &  f1pe(ndata2)

      real*8 taum,chisq,alamda,tpc1,tpc2,tauS

      save ochisq

c     write (*,*) 'we are in mrqmin. taum = ',taum
c     write (*,*) 'we are in mrqmin. tpc2 = ',tpc2

      if (alamda.lt.0) then
            kk = mfit+1
            do 12 j=1,ma
                  ihit=0
```

```
                do 11 k=1,mfit
                      if(lista(k).eq.j) ihit=ihit+1
11              continue
                if(ihit.eq.0) then
                      lista(kk)=j
                      kk = kk+1
                else if (ihit.gt.1) then
                      pause 'improper permutation in lista'
                endif
12          continue
      if(kk.ne.(ma+1)) pause 'improper permutation in lista'
      alamda = 0.001

      call mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,alpha,beta,nca,chisq
     &      ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG)

c     write(*,43) itera,a(1),a(2),chisq/(ndata1+ndata2-nstart1-nstart2-
mfit)
c43   format(I5,5H     ,3(3H   ,F10.3,7H       ))


      do 13 j=1,ma
            atry(j)=a(j)
13    continue
      endif
      ochisq = chisq
      itera = itera+1


      do 15 j=1,mfit
            do 14 k=1,mfit
                  covar(j,k)=alpha(j,k)
14          continue
      covar(j,j)=alpha(j,j)*(1.+alamda)
      da(j) = beta(j)
15    continue

c     write (*,*) 'Just before Gaussj.'

      call gaussj(covar,mfit,nca,da,1,1)

c     write (*,*) 'Just after Gaussj.'


      if(alamda.eq.0) then
            call covsrt(covar,nca,ma,lista,mfit)
            return
      endif
      do 16 j=1,mfit
            atry(lista(j)) = a(lista(j))+da(j)
16    continue
```

```
      call
mrqcof(y1,y2,ndata1,ndata2,atry,ma,lista,mfit,covar,da,nca,chisq
     &  ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe,aG)


      if (chisq.lt.ochisq) then
      alamda = 0.1*alamda
      ochisq=chisq
      do 18 j=1,mfit
            do 17 k=1,mfit
                  alpha(j,k)=covar(j,k)
17          continue
      beta(j)=da(j)
      a(lista(j))=atry(lista(j))
18    continue
      else
      alamda = 10.*alamda
      chisq=ochisq
      endif
      return
      end


      subroutine mrqcof(y1,y2,ndata1,ndata2,a,ma,lista,mfit,
     & alpha,beta,nalp,chisq
     &     ,nstart1,nstart2,e1,e2,tpc1,tpc2,dyda,f1pa,f1pe
     &  ,aG)

      implicit double precision (a-h,o-z)

      dimension y1(10000),y2(10000),alpha(nalp,nalp),beta(mfit),
     &  dyda(mfit),lista(mfit),a(ma),e1(10000),e2(10000)
     & ,nstart1(3),nstart2(3),f1pa(ndata1)
     & ,f1pe(ndata2)
      real*8 taum,chisq,tpc1,tpc2,tauS

c     write (*,*) 'we are in mrqcof. taum = ',taum
c     write (*,*) 'we are in mrqcof. tpc2 = ',tpc2

      do 112 j=1,mfit
            do 111 k=1,j
            alpha(j,k) = 0.
111         continue
      beta(j) = 0.
112   continue
      chisq=0.

120   do 115 i=nstart1(3),ndata1

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG)

c           if (i.lt.nstart1) goto 115
```

```
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 114 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 113 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
113                     continue
            beta(j)=beta(j)+dy*wt
114         continue
      chisq=chisq+dy*dy*sig2i
115   continue

320   do 315 i=nstart1(1),nstart1(2)

            call foncs1(i,a,ymod,dyda,ma,ndata1,e1,tpc1,f1pa,
     & aG)

c           if (i.lt.nstart1) goto 115
            sig2i = 1./y1(i)
            dy = y1(i)-ymod
            do 314 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 313 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
313                     continue
            beta(j)=beta(j)+dy*wt
314         continue
      chisq=chisq+dy*dy*sig2i
315   continue

c     write (*,*) 'Chisq value after monomer: ',chisq

220   do 215 i=nstart2(3),ndata2

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
     & f1pe,aG)

c           if (i.lt.nstart2) goto 215
            sig2i = 1./y2(i)
            dy = y2(i)-ymod
            do 214 j=1,mfit
                  wt=dyda(lista(j))*sig2i
                        do 213 k=1,j
                        alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
213                     continue
            beta(j)=beta(j)+dy*wt
214         continue
      chisq=chisq+dy*dy*sig2i
215   continue

420   do 415 i=nstart2(1),nstart2(2)

            call foncs2(i,a,ymod,dyda,ma,ndata2,e2,tpc2,
```

```
     & f1pe,aG)

c          if (i.lt.nstart2) goto 215
           sig2i = 1./y2(i)
           dy = y2(i)-ymod
           do 414 j=1,mfit
                wt=dyda(lista(j))*sig2i
                     do 413 k=1,j
                     alpha(j,k)=alpha(j,k)+wt*dyda(lista(k))
413                  continue
           beta(j)=beta(j)+dy*wt
414        continue
      chisq=chisq+dy*dy*sig2i
415   continue

c     write (*,*) 'Chisq value after excimer: ',chisq


      do 117 j=2,mfit
           do 116 k=1,j-1
           alpha(k,j)=alpha(j,k)
116        continue
117   continue
      return
      end




      subroutine covsrt(covar,ncvm,ma,lista,mfit)

      implicit double precision (a-h,o-z)

      dimension covar(ncvm,ncvm),lista(mfit)

      do 212 j=1,ma-1
           do 211 i=j+1,ma
           covar(i,j) = 0.
211        continue
212   continue
      do 214 i=1,mfit-1
           do 213 j=i+1,mfit
                if(lista(j).gt.lista(i)) then
                     covar(lista(j),lista(i))=covar(i,j)
                else
                     covar(lista(i),lista(j))=covar(i,j)
                endif
213        continue
214   continue
      swap=covar(1,1)
      do 215 j=1,ma
```

```
      covar(1,j) = covar(j,j)
      covar(j,j) = 0.
215   continue
      covar(lista(1),lista(1))=swap
      do 216 j=2,mfit
      covar(lista(j),lista(j))=covar(1,j)
216   continue
      do 218 j=2,ma
            do 217 i=1,j-1
            covar(i,j)=covar(j,i)
217   continue
218   continue
      return
      end




      subroutine gaussj(a,n,np,b,m,mp)

      implicit double precision (a-h,o-z)

      parameter (nmax=50)
      dimension a(np,np),b(np),ipiv(nmax),indxr(nmax),indxc(nmax)

      do 311 j=1,n
            ipiv(j) = 0
311   continue

      do 322 i=1,n
            big = 0.

            do 313 j=1,n

                  if (ipiv(j).ne.1) then

                  do 312 k=1,n

                  if(ipiv(k).eq.0) then

                        if (abs(a(j,k)).ge.big) then
                        big = abs(a(j,k))
                        irow = j
                        icol = k
                        endif

                        else if (ipiv(k).gt.1) then
                        pause 'singular matrix'
                        endif
```

```
312               continue

                  endif

313         continue

      ipiv(icol) = ipiv(icol) + 1

      if (irow.ne.icol) then

            do 314 l=1,n
                  dum = a(irow,l)
                  a(irow,l) = a(icol,l)
                  a(icol,l) = dum
314         continue

                  dum = b(irow)
                  b(irow) = b(icol)
                  b(icol) = dum

      endif

      indxr(i) = irow
      indxc(i) = icol

      if (a(icol,icol).eq.0)  pause 'singular matrix'

      pivinv = 1./a(icol,icol)

      a(icol,icol) = 1.

      do 316 l=1,n
            a(icol,l) = a(icol,l)*pivinv
316   continue

            b(icol) = b(icol)*pivinv

      do 321 ll=1,n
            if (ll.ne.icol) then

                  dum = a(ll,icol)
                  a(ll,icol) = 0.

                  do 318 l=1,n
                        a(ll,l) = a(ll,l) - a(icol,l)*dum
318               continue

                        b(ll) = b(ll) - b(icol)*dum

            endif

321   continue
```

```
322   continue

      do 324 l=n,1,-1
            if (indxr(l).ne.indxc(l)) then

                  do 323 k=1,n
                        dum = a(k,indxr(l))
                        a(k,indxr(l)) = a(k,indxc(l))
                        a(k,indxc(l)) = dum
323               continue

            endif

324   continue

      return
      end
```