

UvA-DARE (Digital Academic Repository)

A Quantitative Analysis of Light-Driven Charge Transfer Processes Using Voronoi Partitioning of Time Dependent DFT-Derived Electron Densities

Rombouts, J.A.; Ehlers, A.W.; Lammertsma, K.

DOI

[10.1002/jcc.24822](https://doi.org/10.1002/jcc.24822)

Publication date

2017

Document Version

Other version

Published in

Journal of Computational Chemistry

[Link to publication](#)

Citation for published version (APA):

Rombouts, J. A., Ehlers, A. W., & Lammertsma, K. (2017). A Quantitative Analysis of Light-Driven Charge Transfer Processes Using Voronoi Partitioning of Time Dependent DFT-Derived Electron Densities. *Journal of Computational Chemistry*, 38(20), 1811-1818. <https://doi.org/10.1002/jcc.24822>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)

Supporting Information

A Quantitative Analysis of Light-driven Charge Transfer Processes using Time Dependent DFT

Jeroen A. Rombouts, Andreas W. Ehlers, Koop Lammertsma

TABLE OF CONTENTS

General Remarks

Validation

Geometry Optimization and Analysis of Formaldehyde

Geometry Optimization and Analysis of *trans*-Azobenzene

Geometry Optimization and Analysis of a Pyridinium Dye Series

Geometry Optimization and Analysis of a Series of Chromophoric Metallosalphen Compounds

Python Script for Computation of Values from .cube Files

General Remarks

The practical aspects of calculating Q^{VECD} warrant further elaboration. Electron densities are available from most quantum chemical software packages as voxel files, containing a series of volume elements (voxels) that describe the charge density in a volume around the molecule. A script that performs these calculations on the popular .cube files (generated by e.g. ADF, NWChem or Gaussian), written in Python 3, is available at the end of this document.

Validation

In order to assess the consistency of the method across basis sets, several small molecules have been calculated using the B3LYP functional. The ground-state and first excited (singlet) state electron density was calculated using Dunning's correlation consistent polarized basis sets of double, triple, quadruple and quintuple zeta quality, with and without augmentation by diffuse functions. From the obtained densities, Q^{VECD}_A was computed for every atom in each molecule. In order to validate against computational errors, all atomic charges were summed into a single, absolute sum of charges for every compound, $|\sum Q^{\text{VECD}}|$ in equation 1. Given the neutral charge of all fragments, this quantity should always be zero.

$$(1) \quad |\sum Q^{\text{VECD}}| = |\sum_{\text{all atoms } A} Q^{\text{VECD}}_A|$$

For all molecules in the test set (cyanide anion, carbon monoxide, dinitrogen, hydrogen fluoride, nitrous oxide, ozone, FCCH, HCCH, LiCCH, LiCCF, formaldehyde, thioformaldehyde, selenoformaldehyde), the $|Q^{\text{VECD}}|$ value averaged over all basis sets is between 0.000 and 0.004 electron charge, provided that a suitably large integration box is used around the molecule, as shown in figure 1.

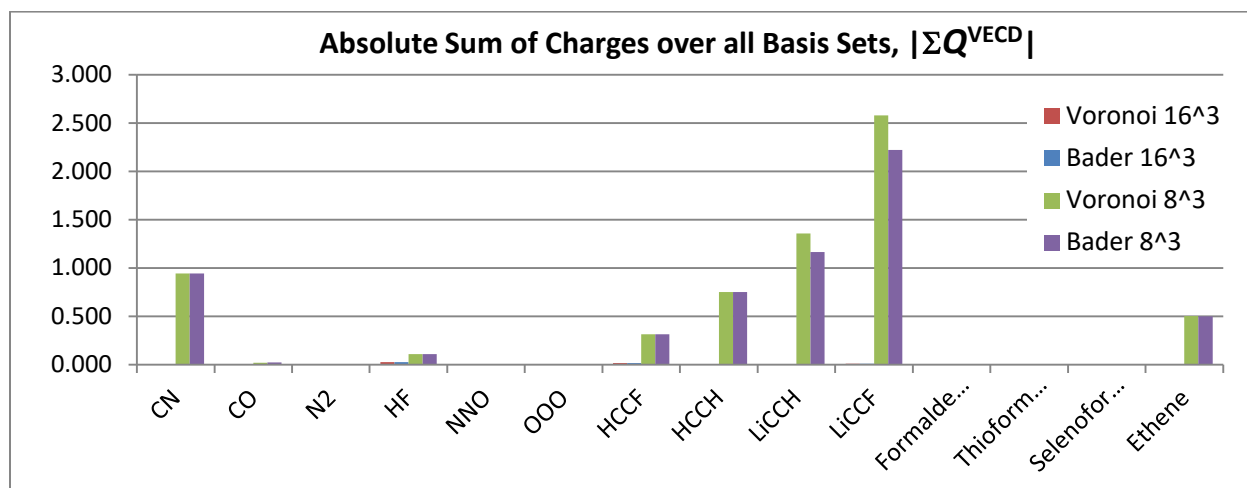


Figure 1: Absolute sum of charges for test molecules, analysis of the small molecules test set integrated using 8^3 \AA and 16^3 \AA boxes. Deviation from zero signifies a non-physical result wherein the total number of electrons in the system changes upon excitation.

Between basis sets, a considerable variance in computed Q^{VECD}_A is sometimes observed. This basis set dependence can be expressed (see equation 2) as the standard deviation in the Q^{VECD}_A values computed for a single atom A across basis sets.

$$(2) \quad \Sigma\sigma = \sum_{\text{all atoms } A} \sigma(Q^{\text{VECD}}_A), \quad \sigma(Q^{\text{VECD}}_A) = \text{Standard deviation of } Q^{\text{VECD}}_A \text{ across basis sets}$$

Figure 4 in the main text compares the basis set dependence of the Voronoi partitioning scheme against charge differences calculated using other schemes for obtaining Partial Atomic Charges, specifically Bader, Hirshfeld and Mulliken atomic charges. SI Table 1 below shows summed standard deviations underlying Figure 4 in the main text (left columns). The cross-basis set sum of charges according to eq. 1 are shown as well. Clearly the use of the Voronoi scheme leads to superior basis set independence, and acceptable rounding-off errors as assessed by the sum of charges.

Table 1: Color-coded performance indicators for PAC schemes on our test set. Colors range from red (= worst observed) to green (= best observed).

Performance indicators (higher = worse performance)	Summed per-atom cross-basis set standard deviation				cross-basis sum of charges			
	VECD	BECD	HECD	MECD	VECD	BECD	HECD	MECD
CO	0.014	0.021	0.007	0.186	0.000	0.001	0.008	0.000
OOO	0.001	0.011	0.036	0.026	0.005	0.003	0.000	0.000
OCH2	0.032	0.031	0.014	0.261	0.001	0.001	0.000	0.000
SCH2	0.010	0.134	0.005	0.091	0.001	0.000	0.000	0.000
SeCH2	0.009	0.205	0.005	0.109	0.003	0.006	0.000	0.000
N2	0.001	0.305	0.000	0.000	0.000	0.002	0.000	0.000
HF	0.015	0.425	0.031	0.170	0.027	0.026	0.000	0.000
NNO	0.016	0.455	0.009	0.116	0.001	0.001	0.000	0.000
LiCCF	0.025	0.044	0.577	1.518	0.009	0.009	0.007	0.000
HCCF	0.231	0.296	0.147	7.842	0.015	0.015	0.201	0.000
LiCCH	0.050	0.189	0.509	18.047	0.006	0.003	0.009	0.000
Ethene	0.245	0.441	0.236	1.175	0.002	0.000	0.141	0.000
HCCH	0.240	0.681	0.023	0.192	0.006	0.007	0.001	0.000
CN	0.359	0.521	0.225	3.832	0.000	0.000	0.715	0.000
Sum of performance indicators	1.246	3.759	1.825	33.564	0.076	0.073	1.084	0.000

Geometry Optimization and Analysis of Formaldehyde

The ground- and excited state geometries of formaldehyde were computed using (time dependent) Density Functional Theory using the B3LYP functional at cc-pVTZ level of theory. Frequency analysis was used to confirm the absence of imaginary vibrational frequencies larger than -50 cm^{-1} , indicative of sufficient convergence of the geometry optimization to a true energy minimum. Real-space density files were written using the NWChem 6.5 software package as input to the provided analytical script that computes electronic charges summed over the associated atom-centered voronoi polyhedra.

Table 2: Formaldehyde Voronoi Atomic Charges in ground- and excited state geometry minima (labeled G and E), computed on S_0 and S_1 energy surfaces. Shown below are the computed ΔQV values for excitation, emission and relaxation steps. Numbers are rounded to one hundredth of an elementary charge. Negative signs of ΔQV indicate movement of electrons towards the nucleus under consideration.

Absorption			Relaxation (S1)		
ATOM #	S1G Q(VORONOI)	Q(VECD)	ATOM #	S1E Q(VORONOI)	Q(VECD)
1 O	-8.212	0.164	1 O	-8.213	-0.001
2 C	-4.939	-0.186	2 C	-4.901	0.038
3 H	-1.422	0.012	3 H	-1.442	-0.019
4 H	-1.427	0.011	4 H	-1.445	-0.018
Relaxation (S0)			Emission		
ATOM #	S0G Q(VORONOI)	Q(VECD)	ATOM #	S0E Q(VORONOI)	Q(VECD)
1 O	-8.376	-0.018	1 O	-8.394	-0.181
2 C	-4.753	0.013	2 C	-4.740	0.161
3 H	-1.434	0.002	3 H	-1.432	0.010
4 H	-1.437	0.003	4 H	-1.434	0.010

Geometry Optimization and Analysis of Azobenzene

The *cis* and *trans* ground-state geometries of azobenzene were computed using Density Functional Theory using the B3LYP functional at cc-pVTZ level of theory. Frequency analysis was used to confirm the absence of imaginary vibrational frequencies (for *cis* and *trans* ground state geometries) larger than -50 cm^{-1} , indicative of sufficient convergence of the geometry optimization to a true energy minimum. The excited state intermediate was found by applying some torsion to one of the C-C-N-N dihedrals and optimizing the molecular structure on the S1 energy surface using TDDFT under relaxed optimization criterion (NWChem 6.5 geometry driver with “LOOSE” settings), without performing a frequency analysis. Real-space density files were written using the NWChem 6.5 software package as input to the provided analytical script that computes electronic charges summed over the associated atom-centered voronoi polyhedra. For clarity, atomic charges for phenyl groups were summed into one number for each ring.

Table 3: Cumulative VECD values for the entire azobenzene *trans* \rightarrow *cis* \rightarrow *trans* cycle outlined in the main text, starting and finishing at *trans* geometry. Shown below are the computed ΔQV values, numbers are rounded to one hundredth of an elementary charge. Negative signs of ΔQV indicate movement of electrons towards the nucleus under consideration. Label nomenclature (“ S_0^G ”) is as described in the main text.

Group	<i>trans</i> S_0^G	<i>trans</i> S_0^E	intermediate $S_1^{\text{Int.}}$	<i>cis</i> S_0^G	<i>cis</i> S_0^E	intermediate $S_1^{\text{Int.}}$	<i>trans</i> S_0^G
N1	0.000	0.059	0.040	-0.058	-0.074	0.040	0.000
N2	0.000	0.058	0.000	-0.048	-0.064	0.000	0.000
Ph1	0.000	-0.058	0.148	0.044	0.059	0.148	0.000
Ph2	0.000	-0.058	-0.187	0.063	0.079	-0.187	0.000

Table 4: Relevant DFT energies used to create the energy diagram in the main text. Energy values are translated to give a value of 0.00 hartree/eV at the lowest energy state. For reference [8] see main text.

State	DFT energies			Experimental ^[8]
	hartree	hartree, scaled	eV	
<i>trans</i> S_0^G	-572.959639	0.00	0.00	0
<i>trans</i> S_1^G	-572.864476	-0.10	2.59	2.8
intermediate $S_1^{\text{Int.}}$	-572.870910	-0.09	2.41	1.6 - 2.8
<i>cis</i> S_1^G	-572.836930	-0.12	3.34	3.5
<i>cis</i> S_0^G	-572.935231	-0.02	0.66	0.6

Table 5: Cartesian coordinates of Azobenzene structures used in this work

Azobenzene, intermediate excited state structure				Azobenzene, cis ground state structure			
N	-0.3583511	0.2927945	-0.3932536	N	-0.5026016	1.1215815	1.6155955
N	-0.5091749	0.8784608	0.7154813	N	0.7318847	1.1405771	1.5115271
C	-0.5230554	0.1593846	-1.6964117	C	-1.3823038	0.4969684	0.6726645
C	-0.6525229	-0.4566090	-4.4505450	C	-3.3307691	-0.5497826	-1.0181626
C	0.4162132	0.7281824	-2.6249994	C	-2.4384435	-0.2474536	1.1981354
C	-1.5579238	-0.6927160	-2.2161049	C	-1.3267096	0.7542386	-0.6982921
C	-1.6016498	-0.9971095	-3.5637227	C	-2.3061647	0.2368509	-1.5343567
C	0.3417821	0.4165495	-3.9704881	C	-3.3921161	-0.7920038	0.3505776
H	1.1914916	1.3887410	-2.2343959	H	-2.4951568	-0.3907358	2.2686214
H	-2.2780551	-1.1125054	-1.5111736	H	-0.5328069	1.3654597	-1.1020139
H	-2.3788581	-1.6655339	-3.9438571	H	-2.2672214	0.4474669	-2.5947839
H	1.0685781	0.8440818	-4.6655039	H	-4.1958374	-1.3887249	0.7606795
H	-0.6843310	-0.7188328	-5.5099817	H	-4.0865082	-0.9563978	-1.6762295
C	0.1167513	0.4241241	1.8425571	C	1.4717965	0.3841586	0.5454429
C	1.3167369	-0.4774403	4.2307332	C	3.1715880	-1.0207949	-1.1538688
C	-0.1153541	1.1060905	3.0680231	C	1.3038859	-0.9908201	0.3725003
C	0.9730743	-0.7149232	1.8399851	C	2.5182590	1.0434239	-0.0995159
C	1.5569125	-1.1481327	3.0218385	C	3.3461738	0.3469835	-0.9678100
C	0.4786289	0.6520570	4.2375211	C	2.1602777	-1.6867313	-0.4691120
H	-0.7691533	1.9802038	3.0648423	H	0.5202403	-1.5109245	0.9038808
H	1.1554123	-1.2335337	0.8973256	H	2.6664897	2.0980917	0.0886977
H	2.2095513	-2.0249639	3.0052241	H	4.1413851	0.8678336	-1.4840813
H	0.2861035	1.1785436	5.1754211	H	2.0351702	-2.7542625	-0.5922467
H	1.7743800	-0.8317289	5.1561867	H	3.8305715	-1.5676667	-1.8143284

Azobenzene, *trans* ground state structure

N	-0.0787578	-0.4077890	-0.4534707
N	0.0798095	0.4210556	0.4543911
C	-0.2413179	-0.2432227	-1.7873018
C	-0.5835777	0.0078069	-4.5548448
C	-0.2562671	1.0382798	-2.3919827
C	-0.4018998	-1.3907044	-2.5950491
C	-0.5697400	-1.2542097	-3.9583211
C	-0.4259270	1.1425755	-3.7579363
H	-0.1333037	1.9138151	-1.7715895
H	-0.3900469	-2.3653339	-2.1284974
H	-0.6916151	-2.1395567	-4.5682392
H	-0.4357371	2.1246623	-4.2126655
H	-0.7156791	0.1059201	-5.6227928
C	0.2418549	0.2497334	1.7879788
C	0.5827321	-0.0199688	4.5539088
C	0.4032559	1.3916549	2.6030438
C	0.2552646	-1.0355418	2.3843678
C	0.4242192	-1.1492848	3.7495305
C	0.5704130	1.2459020	3.9656303
H	0.3926044	2.3694493	2.1431493
H	0.1316369	-1.9067757	1.7581077
H	0.4328044	-2.1344614	4.1974719
H	0.6929439	2.1270951	4.5814209
H	0.7143069	-0.1256146	5.6212207

Geometry Optimization and Analysis of a Pyridinium Dye Series

The ground- and excited state geometries of DAMPI were computed using (time dependent) Density Functional Theory using the cam-B3LYP functional at cc-pVTZ level of theory. Frequency analysis was used to confirm the absence of imaginary vibrational frequencies larger than -50 cm^{-1} , indicative of sufficient convergence of the geometry optimization to a true energy minimum. Real-space density files were written using the NWChem 6.5 software package as input to the provided analytical script that computes electronic charges summed over the associated atom-centered voronoi polyhedra. The computed grouped Voronoi charge differences are as given in the main body of this manuscript. Because these are rather large geometries, the coordinates of the optimized structures are provided below for convenience.

Table 6: Cartesian coordinates of DAMPI and its homologues.

DAMPI-analogue, $n = 0$ smallest frequency $\nu = +14.3 \text{ cm}^{-1}$				DAMPI, $n = 1$ smallest frequency $\nu = -41.1 \text{ cm}^{-1}$, methyl rotation			
C	-0.0196113	-1.4831794	0.0152725	C	-0.6811462	-2.6335380	-0.0988896
C	-1.1375052	-2.2401118	-0.4190355	C	-1.8899470	-3.3672478	-0.2270194
C	-1.1302893	-3.6048956	-0.3925259	C	-1.8989623	-4.7308460	-0.2424806
N	-0.0552026	-4.3032719	0.0477316	N	-0.7521003	-5.4488850	-0.1362414
C	1.0371827	-3.6215841	0.4748305	C	0.4301835	-4.7868047	-0.0117866
C	1.0798858	-2.2581374	0.4672527	C	0.4947072	-3.4272762	0.0091731
H	-2.0194712	-1.7581545	-0.8077732	H	-2.8319240	-2.8461471	-0.3152442
H	-1.9727893	-4.1894131	-0.7290870	H	-2.8106032	-5.3000235	-0.3391569
C	-0.0830734	-5.7727379	0.1135287	C	-0.7668094	-6.9183941	-0.1512828
H	1.8641441	-4.2204194	0.8254149	H	1.3110671	-5.4058952	0.0689929
H	1.9764036	-1.7892735	0.8383801	H	1.4680380	-2.9738940	0.1104648
C	-0.0001714	-0.0433189	-0.0030700	C	-0.7101839	-1.2141676	-0.0860739
C	-1.1743110	0.7167269	-0.2037575	C	0.3821514	-0.3969492	0.0314657
C	-1.1679657	2.0868913	-0.2162019	C	0.4141873	1.0244408	0.0510636
C	0.0375986	2.8177869	-0.0383459	C	-0.7332547	1.8456830	-0.0526393
C	1.2235136	2.0599754	0.1574858	C	-0.6488430	3.2111080	-0.0278543
C	1.1937217	0.6901849	0.1787663	C	0.6067015	3.8704701	0.1045254
H	-2.1281834	0.2239129	-0.3230413	C	1.7621121	3.0514433	0.2085623
H	-2.1016258	2.6072504	-0.3547131	C	1.6566071	1.6851973	0.1816566
N	0.0554130	4.1691148	-0.0549998	H	-1.7112464	1.3966316	-0.1547663
H	2.1706697	2.5587751	0.2830598	H	-1.5546552	3.7895823	-0.1104213
H	2.1344218	0.1756514	0.3101640	N	0.6920210	5.2208385	0.1298154
C	1.3019506	4.8990642	0.1548989	H	2.7375091	3.4985381	0.3098623
C	-1.1714597	4.9263011	-0.2834974	H	2.5582999	1.0913019	0.2633303
H	-1.9039876	4.7487256	0.5067817	C	1.9867475	5.8792455	0.2679208
H	-1.6236927	4.6690264	-1.2430345	C	-0.5061335	6.0463958	0.0194718
H	-0.9366203	5.9850269	-0.2949214	H	-1.2008722	5.8537515	0.8395653
H	2.0284085	4.6844906	-0.6318205	H	-1.0252807	5.8724272	-0.9252375
H	1.7487671	4.6520908	1.1196486	H	-0.2199062	7.0918243	0.0589433
H	1.0947762	5.9635435	0.1424052	H	2.6514445	5.6296144	-0.5617108
H	0.8813559	-6.1669847	-0.1952891	H	2.4774713	5.6007091	1.2027315
H	-0.8496251	-6.1454539	-0.5586526	H	1.8391908	6.9537373	0.2710942
H	-0.3026859	-6.0978341	1.1294578	H	-0.1703003	-7.2866868	-0.9837613
				H	-1.7892374	-7.2628055	-0.2671197
				H	-0.3646967	-7.3033208	0.7840187
				H	-1.6925817	-0.7695498	-0.1785655
				H	1.3570356	-0.8640756	0.1226521

DAMPI analogue, n = 2				DAMPI analogue, n = 3			
smallest frequency $\nu = -45.4 \text{ cm}^{-1}$, methyl rotation				smallest frequency $\nu = -44.1 \text{ cm}^{-1}$, methyl rotation			
C	1.7568837	-3.2793878	1.4491944	C	0.5899464	-1.9622251	4.8461565
C	1.7912082	-3.0660513	2.8556472	C	0.4384170	-1.6638671	6.2295456
C	2.3301535	-3.9923205	3.6962672	C	0.7479469	-2.5737164	7.1940644
N	2.8583867	-5.1582682	3.2380086	N	1.2165015	-3.8117533	6.8814294
C	2.8467788	-5.4041340	1.8993536	C	1.3777574	-4.1410933	5.5688652
C	2.3206908	-4.5125847	1.0160797	C	1.0839706	-3.2704683	4.5666959
H	1.3913716	-2.1651855	3.2935420	H	0.0695839	-0.6955936	6.5347924
H	2.3637976	-3.8456113	4.7648443	H	0.6373251	-2.3589466	8.2458140
C	3.4360517	-6.1474991	4.1568286	C	1.5535790	-4.7871476	7.9247606
H	3.2777829	-6.3432297	1.5862737	H	1.7526294	-5.1351601	5.3759293
H	2.3370923	-4.7576577	-0.0357998	H	1.2370849	-3.6030180	3.5521061
C	-1.0091913	1.9128596	-0.6879570	C	-0.3403378	1.1666844	-3.2247975
C	-1.0885857	1.7910161	-2.0949709	C	-0.1297920	0.6959823	-4.5394960
C	-1.6465406	2.7681410	-2.8748890	C	-0.3849631	1.4669076	-5.6452053
C	-2.1750497	3.9581460	-2.2998880	C	-0.8790413	2.7898979	-5.5091677
C	-2.0982742	4.0862663	-0.8882012	C	-1.0928292	3.2710575	-4.1876192
C	-1.5356975	3.0965720	-0.1240576	C	-0.8320241	2.4859265	-3.0958895
H	-0.7015901	0.9064974	-2.5813028	H	0.2465286	-0.3094980	-4.6795218
H	-1.6813983	2.6240511	-3.9426705	H	-0.2041782	1.0522521	-6.6236166
N	-2.7278292	4.9267726	-3.0695942	N	-1.1371971	3.5651086	-6.5921901
H	-2.4833905	4.9671605	-0.4009025	H	-1.4663895	4.2701894	-4.0312797
H	-1.4933066	3.2266053	0.9499684	H	-1.0115279	2.8994650	-2.1133504
C	-2.7979191	4.7775318	-4.5185434	C	-0.9099275	3.0545513	-7.9381838
C	-3.2641408	6.1391283	-2.4622000	C	-1.6442090	4.9221232	-6.4316739
H	-2.4910368	6.6919282	-1.9241473	H	-0.9437634	5.5485827	-5.8748437
H	-4.0755141	5.9123599	-1.7671988	H	-2.6054153	4.9318341	-5.9130761
H	-3.6577223	6.7820657	-3.2421424	H	-1.7862795	5.3652130	-7.4116455
H	-1.8032855	4.6833525	-4.9595938	H	0.1386231	2.7914100	-8.0946409
H	-3.2722274	5.6562797	-4.9422320	H	-1.1761695	3.8212728	-8.6579837
H	-3.3880838	3.9036336	-4.8026183	H	-1.5216546	2.1724138	-8.1400002
H	3.3553279	-5.7786334	5.1741689	H	1.3368089	-4.3575379	8.8975091
H	2.8968870	-7.0897037	4.0762830	H	0.9601726	-5.6907073	7.7960790
H	4.4861054	-6.3078531	3.9189017	H	2.6120349	-5.0376508	7.8749198
C	-0.4401457	0.9277517	0.1643184	C	0.3575578	-1.1537742	2.5035840
C	0.1193360	-0.2784685	-0.1784294	C	0.2602940	-0.9974664	3.8696390
H	-0.4545490	1.1702002	1.2233228	H	0.7185902	-2.0965828	2.1049512
H	0.1660218	-0.5824708	-1.2172439	H	-0.1032523	-0.0465154	4.2417991
C	0.6461360	-1.1558102	0.7826325	C	-0.0546579	0.3126325	-2.1227856
C	1.2183845	-2.3750882	0.5039216	C	-0.1900401	0.5759768	-0.7827968
H	0.5855600	-0.8249585	1.8145805	H	0.3180590	-0.6730375	-2.3867076
H	1.2677572	-2.6843125	-0.5337169	H	-0.5568633	1.5403310	-0.4519290
				C	0.1327092	-0.3700467	0.2026463
				C	0.0211188	-0.1736038	1.5627316
				H	0.4998148	-1.3340152	-0.1396459
				H	-0.3435330	0.7821212	1.9242886

Geometry Optimization and Analysis of a Series of Chromophoric Metallosalphen Compounds

To analyze the properties of the series of compounds homologous to DATZnS, we took its DFT-optimized structure (In the ESI of Rombouts *et al*, *Chem. Eur. J.* 2014, see main text ref. 1b), optimized this again at cam-B3LYP/cc-pVDZ level of theory, and replaced the zinc atom with cobalt, copper, iron and nickel. For reasons of computational efficiency, the alkyl chains on the imide nitrogens are replaced by a hydrogen atom. The overall charge of the compounds was chosen to obtain a closed-shell spin = 0 electronic state. The cobalt-containing compounds were calculated separately as the overall monoanionic and monocationic species, the copper compound was calculated as its monoanion, and all other compounds were calculated as neutral compounds. The homologues were optimized at the same level of theory, using nwchem 6.5's "xfine" integration grid and convergence criteria of 10^{-6} (energy), 10^{-6} (density) and 10^{-5} (gradient). No frequency calculations could be performed due to the computational cost associated with the rather large size of these compounds.

For each compound, the singlet excitations were computed using TD-DFT. The VECD calculation was performed on the first transition encountered with a non-zero (meaning, larger than 0.1) dipole oscillator strength. This procedure is required for large pigments, in order to weed out dark (non-absorbing) transitions at low energies whenever these occur. Judicious selection of target roots prior to VECD calculation ensures that the actual, physically relevant optical transitions are investigated. All selected transitions were of a 'clean' orbital configuration, with only one occupied-to-unoccupied transition contributing to 75% or more of the excitation character (see table 5).

Table 7: TDDFT investigation of first transition with non-zero transition dipole

Compound name	Co(III)	Co(I)	Cu(I)	Fe(II)	Ni(II)	Zn(II)
	DATCoS ⁺¹	DATCoS ⁻¹	DATCuS ⁻¹	DATFeS	DATNiS	DATZnS
TDDFT root investigated using VECD	5	5	2	5	4	1
TDDFT energy	1.87 eV	1.53 eV	2.14 eV	2.59 eV	2.59 eV	2.57 eV
Orbital configuration	H-2 -> L 78%	H -> L+1 75%	H-1 -> L 85%	H -> L 82%	H -> L 88%	H -> L 89%
Oscillator strength	0.20	0.12	1.21	1.45	1.48	1.46

Python 3 Script for Computation of Values from .cube Files

You are free to use, modify or distribute this code without any further attribution. Save the code as, for example, cubereader.py, and execute this with

```
python3 cubereader.py final_state.cube initial_state.cube
```

to calculate Q^{VECD} using (final Voronoi charges) – (initial Voronoi charges).

```
1. import sys
2. import numpy as np
3. import math
4.
5. class Cartesians:
6.     def __init__(self):
7.         self.labels = None
8.         self.xyz = None
9.         self.n_atoms = None
10.        self.VECD = []
11.
12.    def set(self, coords, labels):
13.        assert len(coords) == len(labels)
14.        self.n_atoms = len(labels)
15.        self.labels = labels
16.        self.xyz = np.vstack(coords)
17.        assert self.xyz.shape[1] == 3
18.
19. class Cubereader:
20.     def __init__(self, uri, auto_process = True):
21.         self.voxels = None
22.         self.usesBohr = None
23.         self.voronoi = None
24.         self.has_voronoi = False
25.
26.         self.uri = uri
27.         self.n_atoms = 0
28.         self.cube_shape = []
29.         self.cartesians = Cartesians()
30.
31.         cartesians_to_set = [[], []]
32.         with open(self.uri) as this_file:
33.             for index, line in enumerate(this_file):
34.                 line = line.strip().split()
35.
36.                 if index == 2:
37.                     self.n_atoms = int(line[0])
38.                     self.VECD = [0.000]*self.n_atoms
39.                     if float(line[1]) < 0:
40.                         self.usesBohr = False
41.                     else:
42.                         self.usesBohr = True
43.
44.                 if index > 1 and index <= 5:
45.                     self.cube_shape.append([float(f) for f in line])
46.                 elif index > 5 and index <= 5 + self.n_atoms:
47.                     cartesians_to_set[0].append([float(f) for f in line[2:]])
48.                     cartesians_to_set[1].append(str(line[0]))
49.                 elif index > 5 + self.n_atoms:
50.                     break
51.
52.         self.cartesians.set(*cartesians_to_set)
53.         self.voxels = np.zeros(shape = tuple(int(i[0]) for i in self.cube_shape[1:4]), dtype = np.float_)
54.
55.         if auto_process:
56.             self.process_voxels()
57.             self.compute_voronoi()
58.             self.print_voronoi()
59.
60.     def process_voxels(self):
```

```

61.     assert type(self.voxels) != None and self.usesBohr != None
62.     i, j, k = (0, 0, 0)
63.     imax, jmax, kmax = self.voxels.shape
64.
65.     print("reading .cube file {0} into memory".format(self.uri))
66.     with open(self.uri) as this_file:
67.         for index, line in enumerate(this_file):
68.             line = line.strip().split()
69.             if index > 5 + self.n_atoms:
70.                 for element in line:
71.                     assert element[7] != '-' ## some .cubes with very small values throw an error here, eg
'1.23456-105'
72.                         self.voxels[i][j][k] = float(element)
73.                         k += 1
74.                         if k == kmax:
75.                             k = 0
76.                             j += 1
77.                         if j == jmax:
78.                             j = 0
79.                             i += 1
80.
81.     def compute_voronoi(self):
82.         shape = self.voxels.shape
83.         self.cube_origin = np.array([self.cube_shape[0][1], self.cube_shape[0][2], self.cube_shape[0][3]])
84.         self.voxel_size = np.array([self.cube_shape[1][1], self.cube_shape[2][2], self.cube_shape[3][3]])
85.         self.voxel_volume = self.voxel_size[0] * self.voxel_size[1] * self.voxel_size[2]
86.
87.         print("computing Voronoi charges for file {0}, this can take a while.".format(self.uri))
88.         for i in range(shape[0]):
89.             for j in range(shape[1]):
90.                 for k in range(shape[2]):
91.                     X = np.array([i,j,k]) * self.voxel_size + self.cube_origin
92.                     distances = [(i, d) for i, d in enumerate([np.linalg.norm(atom -
X) for atom in self.cartesians.xyz])]
93.                     argmin_distances = sorted(distances, key=lambda tpl: tpl[1])[0]
94.                     self.VECD[argmin_distances[0]] += self.voxels[i][j][k]
95.
96.         ## divide VECD by voxel volume
97.         self.voronoi = [i * self.voxel_volume for i in self.VECD]
98.         self.has_voronoi = True
99.
100.    def print_voronoi(self):
101.        assert self.has_voronoi
102.
103.        print("{0:^6} || {1:^10} | {2:^10} | {3:^10} || {4:^10}".format("label", "x", "y", "z", "Voronoi charge"))
104.        for index, label in enumerate(self.cartesians.labels):
105.            print("{0:^6} || {1:^10} | {2:^10} | {3:^10} || {4:^2.6}"\
106.                .format(label, self.cartesians.xyz[index][0], self.cartesians.xyz[index][1], self.cartesians.xyz[index][2],\
self.voronoi[index]))
107.
108.
109.    def __sub__(self, other):
110.        VECD = [0.000] * self.n_atoms
111.        assert len(self.voronoi) == len(other.voronoi)
112.        assert self.usesBohr == other.usesBohr
113.
114.        for index in range(self.n_atoms):
115.            ## by convention: negative VECD is accumulation of electrons
116.            VECD[index] = -1*self.voronoi[index] - -1*other.voronoi[index]
117.
118.            print("{0:^6} || {1:^10} | {2:^10} | {3:^10} || {4:^10}".format("label", "x", "y", "z", "VECD"))
119.            for index, label in enumerate(self.cartesians.labels):
120.                print("{0:^6} || {1:^10} | {2:^10} | {3:^10} || {4:^2.6}"\
121.                    .format(label, self.cartesians.xyz[index][0], self.cartesians.xyz[index][1], self.cartesians.xyz[index][2],\
VECD[index]))
122.
123.
124.        return VECD
125.
126. if __name__ == "__main__":
127.     print("Python 3 script for VECD calculation. Requires Numpy. Usage:\n\tpython3 script.py file_1.cube
file_2.cube\n")
128.     this_cube_uri = str(sys.argv[1])

```

```
129.     that_cube_uri = str(sys.argv[2])
130.
131.     this_cube = Cubereader(this_cube_uri)
132.     that_cube = Cubereader(that_cube_uri)
133.
134.     print("\n\t***RESULTS***\n")
135.     this_cube - that_cube
```