GronOR: Scalable and Accelerated Non-Orthogonal Configuration Interaction for Molecular Fragment Wave Functions

T. P. Straatsma,[*,†,‡] R. Broer,[¶] A. Sánchez-Mansilla,[§] C. Sousa,[‖] and C. de Graaf[¶,§,⊥]

†*National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6373, U. S. A.*

‡*Department of Chemistry and Biochemistry, University of Alabama, Tuscaloosa, AL 35487-0336, U. S. A.*

¶*Theoretical Chemistry Group, Zernike Institute for Advanced Materials, University of Groningen, Groningen, The Netherlands*

§*Department of Physical and Inorganic Chemistry, Universitat Rovira i Virgili, C. Marcel·lí Domingo 1, 43007 Tarragona, Spain*

‖*Department of Physical Chemistry and Institut de Química Teòrica i Computacional, Universitat de Barcelona, Spain*

⊥*ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain*

E-mail: str@ornl.gov

Abstract

GronOR is a program package for non-orthogonal configuration interaction calculations. Electronic wave functions are constructed in terms of anti-symmetrized products of multi-configuration molecular fragment wave functions. The computational complexity of the non-

orthogonal methodologies implemented in GronOR applied to large molecular assemblies requires a design that takes full advantage of massively parallel supercomputer architectures and accelerator technologies. This work describes the implementation strategy and resulting performance characteristics. In addition to parallelization and acceleration, the software development strategy includes aspects of fault resiliency and heterogeneous computing. The program was designed for large-scale supercomputers, but also runs effectively on small clusters and workstations for small molecular systems. GronOR is available as open source to the scientific community.

Introduction

Electron and excitation energy transfer in molecular systems constitute fundamental chemical processes in a wide range of important phenomena, including the interconversion of photo-excitation and electric energy found in natural biomolecular photosynthesis and man-made photo-voltaic electronic devices. Understanding transfer rates in processes such as electron transfer, multiple exciton generation,[1] intermolecular Coulombic decay,[2] and exciton diffusion[3] is critical for the design of new materials with high energy conversion efficiencies.[4] Key to quantifying transfer rates is the ability to accurately determine the electronic coupling between states with different electronic configurations.[5–7] Electronic coupling can only indirectly be estimated from experiments, but in theoretical approaches are directly expressed in terms of elements of the Hamiltonian and overlap matrices in a diabatic representation of the electronic states involved. Among the many theoretical and computational methods that are available for efficiently evaluating electronic coupling, non-orthogonal configuration interaction with multiconfigurational fragment wave functions representing the diabatic states (NOCI-F) is one of the most rigorous approaches.[8] Expressed in an independently optimized, and therefore in principle non-orthogonal orbital set, each diabatic state is ensured to be unbiased and to include full orbital relaxation. Our implementation of NOCI is one among the many that have been developed over the last decade.[9–19] The idea of expressing each electronic configuration in its own set of orbitals is not new, but only recently computers have

become sufficiently powerful to efficiently perform such calculations.

NOCI-F naturally includes static correlation corrections in the expansion of the fragment states in terms of multiconfigurational wave functions, typically from Complete Active Space Self-Consistent Field (CASSCF) calculations, but the direct inclusion of dynamic correlation effects would lead to unmanageable and computationally prohibitively expensive computations. An alternative approach taken here is to shift the diagonal elements of the NOCI matrix with dynamic correlation energy corrections (DCEC) calculated for the multiconfiguration fragment states used to construct the many-electron basis functions (MEBF). The DCEC are evaluated using a perturbative method such as Complete Active Space Second-order Perturbation Theory (CASPT2). Alternatively, the coefficients of the relatively short Complete Active Space (CAS) wave functions can be dressed with the effect of dynamic correlation with the help of intermediate effective Hamiltonian theory.[20] Taking the effects of dynamic correlation into account in NOCI-F calculations has been demonstrated to significantly change electronic coupling in singlet fission studies and to give very accurate results in a study of the magnetic coupling strength in organic radicals.[21]

Although the construction of MEBFs from independently optimized but non-orthogonal multiconfiguration fragment wave functions leads to relatively compact wave function expansions in terms of Slater determinants or combinations thereof, the evaluation of the Hamiltonian matrix for such an expansion involves the evaluation of contributions from large numbers of non-orthogonal determinant pairs. These contributions can be evaluated independently and, therefore, a massively parallel procedure emerges naturally. With singlet fission studies as the primary research target, this original version was limited to the construction of MEBFs of singlet spin symmetry, and used labeled two-electron integrals in an atomic orbital (AO) basis.

The present program package, GronOR, is an open source program[22,23] developed through a collaboration between the University of Groningen and Oak Ridge National Laboratory to be a massively parallel and GPU-accelerated software application to perform non-orthogonal configuration interaction calculations very efficiently on modern computer architectures. The first version of GronOR was developed on the 27PF Titan supercomputer with NVIDIA Kepler accelerators,

installed in 2013, and its successor the 200PF Summit supercomputer with NVIDIA Volta accelerators, installed in 2019, at the Oak Ridge Leadership Computing Facility (OLCF). The GronOR project was part of OLCF's Early Science Program for the development of scalable applications,[24] and is a feature application project in the Accelerated Data and Computing (ADAC) Institute.[25] The University Rovira i Virgili in Tarragona, Spain, joined the development team, and the application includes new features and has undergone a series of significant methodological and performance improvements. New features include the ability to construct general spin states beyond just singlet spin states, the treatment of dynamic correlation energy corrections, the ability to construct multi-fragment states, and the deposition of cml-formatted results into the iochem-BD repository for easy sharing with the scientific community.[26,27] Methodological improvements include the transformation to a common orbital basis and associated two-electron integral transformation resulting in a significant reduction in the computational cost of processing two-electron integrals. Implementation improvements include a more efficient base-state generator, and options to use the most efficient linear algebra solvers available for the problem size as well as the computer architecture used.

The NOCI implementation in GronOR provides certain features that distinguishes it from other NOCI codes. First, there is the possibility to treat ensembles of molecules through the construction of the many-electron basis functions as anti-symmetrized spin-adapted products of fragment wave functions. This ensures a pure diabatic description of the different electronic configurations with full orbital relaxation. This is in principle also possible in other codes, such as the state interaction implemented in OpenMolcas, but there the active space of the different MEBFs must have the same size and the orbitals must be of similar character. These restrictions do not exist in GronOR. As long as the fragment wave functions can be written as a linear combination of Slater determinants, GronOR can calculate the Hamiltonian and overlap matrix elements of the resulting MEBFs. Second, GronOR can handle medium-sized systems without making any approximation in the calculation of the matrix elements. The recent improvements described below have significantly increased the maximum system size for which rigorous NOCI calculations can be performed. GronOR can

handle systems with up to 150 atoms in an almost routine fashion. Other implementations either introduce approximations (ab initio Frenkel-Davydov approach) or have only reported results for small systems with very short wave function expansions of the fragment wave functions.[11,28–30]

Methodology

Construction of spin-adapted antisymmetrized products of fragment wave functions

The ensemble wave function Ψ is expanded in terms of the MEBFs Φ_i that can be Slater determinants or combinations thereof,

$$\Psi = \sum_{i=1}^{N} c_i \Phi_i \tag{1}$$

where the MEBFs are spin-adapted linear combinations of antisymmetrized products of fragment orbitals ϕ_{jk}

$$\Phi_i = \sum_j a_{ij} \hat{A} \prod_k \phi_{jk} \tag{2}$$

with $\hat{A}$ the antisymmetrization operator and a_{ij} the MEBF expansion coefficients.

In early versions of GronOR the spin coupling was limited to singlet MEBFs formed by two singlet, two doublet or two triplet fragments.[8,24] While sufficient for the study of phenomena related to singlet fission, it strongly limits other possible applications of NOCI-Fragments. GronOR now includes an improved spin coupling algorithm based on Clebsch-Gordan coupling coefficients, which allows to generate MEBFs with any spin moment from fragment wave functions with arbitrary spin, compatible with the total spin of the MEBF. This opens opportunities for a much wider range of applications, such as magnetic interactions in molecular or extended systems. As the wave functions of fragments A and B are expressed in determinants of maximum M_S, first a list is generated with all possible M_S-determinants between $M_{S,max}$ and $-M_{S,max}$ by subsequent applications of the step-down operator $\hat{S}^-$. Then the determinants of the two fragments are combined

multiplying only those determinants for which the sum of the respective M_S-values add-up to the total spin of the MEBF. Spin symmetry is assured by multiplying with the appropriate Clebsch-Gordan coefficient $C(S_1, M_{S1}, S_2, M_{S2}, S, M_S)$, calculated through a recursive algorithm as outlined by Zuo, Humbert and Esling.[31] In GronOR, this procedure can be extended to the study of systems composed of more than two fragments by providing the intermediate spin coupling for pairs of fragments. The spin adaptation of the MEBFs is in principle similar to the construction of the configuration state functions (CSF) in standard multiconfigurational approaches such as CASSCF, although the details of the implementation differ.

Factorization of the cofactor matrix

The resulting Hamiltonian of the non-orthogonal MEBFs, requires the evaluation of non-orthogonal determinant matrix elements. The computationally most demanding contributions are those from the two-electron elements

$$g_{\alpha\beta} = \langle \Phi_\alpha | \, \bar{g}_{12} \, | \Phi_\beta \rangle = \sum_{i<k} \sum_{j<l} \langle \phi_i \phi_j | \, \bar{g}_{12} \, | \psi_k \psi_l \rangle \, S(ik, jl) \tag{3}$$

with $\bar{g}_{12}$ the two-electron operator and S(ik,jl) the second-order co-factor of the overlap matrix of the molecular orbitals. The non-orthogonal determinant matrix elements are effectively evaluated using a method based on factorization of the transformed second order cofactor matrix as developed by van Montfort[32] and by Broer and Nieuwpoort.[33,34] The method was implemented for single determinant matrix elements in the General Non-Orthogonal Matrix Element (GNOME) computer program, and forms the basis for efficient calculation of large wave function expansions in GronOR. The core of the method is a transformation to corresponding orbitals $\tilde{\phi}$ and $\tilde{\psi}$ with a diagonal overlap matrix, expansion in a common orbital set $\{\chi\}$,[35] and factorization of the trans-

formed cofactor matrix,

$$\langle \Phi_\alpha | \bar{g}_{12} | \Phi_\beta \rangle = \sum_{i<k} \sum_{j<l} \langle \tilde{\phi}_i \tilde{\phi}_j | \bar{g}_{12} | \tilde{\psi}_k \tilde{\psi}_l \rangle \prod_{m \neq i,k} \lambda_m \quad (4)$$

$$= \sum_{p,q,r,s} \langle \chi_p \chi_r | \bar{g} | \chi_q \chi_s \rangle (1 - \hat{p}_{qs}) \sum_i \tilde{c}_{ip} \tilde{d}_{iq} \sum_{k \neq i} \tilde{c}_{kr} \tilde{d}_{ks} \prod_{m \neq i,k} \lambda_m \quad (5)$$

$$= (1 - \hat{p}_{pr})(1 - \hat{p}_{qs}) F_{pq}(\omega) G_{rs}(\omega) \quad (6)$$

in which λ_m is the overlap between $\tilde{\phi}_m$ and $\tilde{\psi}_m$. The number of singularities ω (i.e., the number of λ's equal to zero) determines the functional form of the matrices F and G in terms of the expansion coefficients $\tilde{c}$ and $\tilde{d}$.[33,34] Their explicit forms are given in the supplementary information. A transformation of the two-electron integrals in terms of corresponding orbitals is not required.

Common orbital transformation

In configuration interaction approaches based on orthogonal Slater determinants, the computational cost can be significantly reduced by transforming the two-electron integrals from the atomic to the molecular orbital basis. However, the fact that NOCI expresses each state in a different set of MOs complicates such integral transformations. This required the design and implementation of a novel procedure to generate a common MO basis that can be used to describe all the electronic states and to express the two-electron integrals in this new basis.[35] The overlap matrix of the orbitals with non-zero occupation numbers of all electronic states is diagonalized and the eigenvectors with small eigenvalues are discarded to remove the linear dependencies from the basis. The size and the accuracy depend on the threshold for linear dependency, but the size of the new common MO basis is typically only slightly larger than half the number of electrons in the system. This new approach significantly reduces the size of the two-electron integral file, often by two orders of magnitude for moderately large AO basis sets. As a result of the use of this common MO basis procedure, NOCI roughly scales as N^4 in the number of electrons of the system instead of the number of basis functions. It makes possible the study of much larger molecular systems using this methodology.

Dynamic correlation correction

NOCI-Fragments is designed such that the fragment (or monomer) wave functions used to construct the MEBFs can be calculated with virtually any post-Hartree-Fock scheme. In the applications described here, fragment wave functions are calculated from a CASSCF approach, but other choices such as the ICE-CI (or CIPSI),[36,37] ORMAS,[38] XASSCF (X=R, G or L),[39,40] stochastic CI[41,42] or any other multiconfigurational expansions can also be used. The wave function expansions for each fragment wave function can be chosen differently. These multiconfigurational fragment wave functions take into account non-dynamic (or static) electron correlation, but dynamic electron correlation is more complicated to include. The inclusion of dynamic electron correlation can be accomplished in two different ways. The first shifts the diagonal matrix elements of the NOCI Hamiltonian matrix with the DCEC of the fragment electronic states, typically calculated with multiconfigurational second-order perturbation theory (CASPT2, NEVPT2).[43,44] The Hamiltonian is expressed in an orthogonalized MEBF basis, then the shifts are applied on the diagonal and subsequently the matrix is transformed back to the original non-orthogonal basis. The second approach also uses dynamic electron correlation modified wave functions through explicit inclusion of this effect in the wave function using effective Hamiltonian theory to dress the reference wave function with dynamic electron correlation effects as implemented in the dynamic electron correlation dressed complete active space (DCD-CAS) of Pathak, Lang and Neese.[45]

Design Strategy

The implementation of GronOR is based on four basic design principles. The first design principle is to develop for massive parallelism in order to be able to effectively use the largest supercomputers available for open science. This requires an implementation that minimizes load imbalance through the use of the appropriate programming model for the algorithm. The best choice for evaluating the large expansions of NOCI-F properties in terms of independent contributions is a task-based master-worker model. Parallelism is also significantly affected by communication re-

quirements, which in GronOR are minimized by keeping the large integral list memory resident once moved from the file system, and by controlling the number and size of messages by evaluating the matrix element contributions in batches. Finally, collective communication event such as synchronizations and reduction operations are negatively impacting load imbalance and are, after the initial setup phase, completely avoided in GronOR's design. For all communication operations the ubiquitously available Message Passing Interface (MPI) is used.

The second design principle is to take full advantage of accelerators. To ensure maximum portability of the code, GPU acceleration is achieved through a directive-based approach using OpenACC or OpenMP target off-loading. With the latest compiler technologies this can be done with minimal impact on performance in comparison to the less portable CUDA. Acceleration on CPU-based system is accomplished using OpenMP threading directives. The GNOME algorithm depends on two linear algebra solvers, a singular value decomposition and an eigenvalue solver, and provides a CPU-version for these. Whenever available, a computer vendor's optimized linear algebra library provides these solvers that are highly optimized for their architecture, such as the Math Kernel Library (MKL) for Intel architectures and the CUSOLVER library for NVIDIA GPU accelerators. GronOR provides options to use both of these alternatives. Because all computational work in GronOR can be carried out on GPU accelerators, it is computationally most efficient for GPU execution to use libraries that expect input data to already reside in GPU memory and to also leave the output data in GPU memory, as does the NVIDIA CUSOLVER library. Other options exist, such as the MAGMA[46] or SLATE[47] libraries, and the code could easily be adapted to using these, especially as the calling procedure is usually similar if not identical. However, some of the alternative library routines include copying input data from and output data to host memory, which for GronOR is not required as all data already resides in GPU memory and would lead to unnecessary data transfer.

The third design principle is to implement fault resiliency where possible and appropriate. The task-based master-worker model provides for an effective way to achieve hard fault resilience by duplicating towards the end of a run any still outstanding tasks and incorporating the first results

returned to the master rank. Storing each Hamiltonian matrix element as soon as completed in a check-point restart file provides a means for resiliency against network failures, power failures and job execution time limit terminations.

The fourth and final design principle is to deliver a heterogeneous computing capable implementation. In the current implementation in GronOR, most routines exist in two versions. One version is optimized to use GPU accelerators, while the other is optimized to execute effectively on CPUs. The versions can be used simultaneously to support accelerated and non-accelerated ranks on the same node to use all available compute resources. Another application mode is to compile independent accelerated and non-accelerated executables for use on compute clusters with accelerated and non-accelerated partitions on the same interconnect network. The ability to run with independent executables is needed as runtime systems and libraries can be expected to be different on these partitions, but needs to be supported by the job scheduler.

Implementation

Generating integrals and fragment wave functions

For the generation of the molecular integrals and the fragment wave functions, GronOR is interfaced to OpenMolcas.[48,49] The multi-configuration method most often used is CASSCF, followed by CASPT2 to generate the DCEC. Part of the GronOR software are auxiliary programs for extracting the required data from OpenMolcas files, and to carry out the common orbital transformation of the wave functions and integrals. This transformation is not required, but significantly reduces both the number of basis functions in the wave function expansion as well as, and more importantly, the number of two-electron repulsion integrals. Since this transformation also significantly reduces the number of near-zero integrals, GronOR can process them in canonical order which further reduces the memory requirements as no integral labels need to be stored.

Input/output

The input data to GronOR consists of the molecular orbital coefficients and determinant lists for each of the fragment wave functions, and the integrals for the full system, all in either the AO or common MO basis set. Of these the integrals are always the largest amount of data, typically contained in multiple files, and reading and distributing the integrals constitute a large component of the setup time. In this work, the wall clock time required for reading and transfer of coefficients and integrals is always included in the reported setup times, not in the times reported for Hamiltonian calculation. For large systems and very large node-count runs, this can negatively impact the overall scalability. When all integrals can be stored in the memory available to a worker rank, a single worker rank reads the integrals from file and MPI_Bcast is used to distribute them to all other worker ranks. If the availability of memory requires the integrals to be divided over multiple worker ranks, worker ranks in a single group read the integrals from file in parallel and distribute their portion of the integrals to equivalent ranks in the other groups using MPI_Bcast. When multiple ranks, executing on one node, require the same batch of integrals, a different integral distribution mechanism can be used in which integrals are distributed to a single rank on each node in a first step, followed by intra-node distribution to other ranks on the same node. This two-step mechanism reduces the setup time on large node-count computer systems.

Fragment molecular orbital coefficients and determinant lists are read by the master rank and distributed using MPI_Bcast to all worker ranks.

Task-based execution model

Each of the Hamiltonian and overlap matrix elements, as well as their large number of individual contributions in terms of determinant pairs can be independently calculated. This makes parallelization using a task-based execution model a particularly efficient strategy of achieving load balance provided the implementation is designed to avoid any form of synchronization between ranks, and the frequency and size of messages between master and worker ranks do not create communication contention on the network. The work-flow in GronOR is schematically depicted

in Figure 1.

After the setup phase, the master rank enters an iterative loop (labeled 1 in Figure 1) consisting of a blocking MPI_Recv from ANY_SOURCE, determining the source of a received buffer and accumulating the results in the appropriate arrays, and selecting and sending with a non-blocking MPI_iSend a 32-byte buffer identifying the next task to be evaluated back to the source worker rank. Once all tasks have been dispatched, the master enters another iterative loop (labeled 2 in Figure 1) consisting of a blocking MPI_Recv from ANY_SOURCE, determining the source of a received buffer and, if not previously received and processed, accumulating the results in the appropriate arrays, and selecting and sending with a non-blocking MPI_iSend a duplicate of a still outstanding task back to the source worker rank. Once all required results buffers have been received, the master sends with a non-blocking MPI_iSend a terminate signal to each of the worker ranks (labeled 3 in Figure 1). Part of the processing of results data on the master rank is writing a record on a checkpoint restart file every time the calculation of one of MEBF in the run has completed. This file is read on subsequent restart runs of the same job such that repeated calculation of completed Hamiltonian matrix elements is avoided. In addition, the master process periodically writes a status message onto a so-called dayfile which provides information about the progress of a running job.

Each of the worker ranks after receiving a task definition from the master rank (labeled 1 in Figure 2) executes for each item in the task list in an iterative loop the evaluation of factorize cofactor matrices and overlap and one-electron integral contributions (labeled 2 in Figure 2) and in batches the contributions from the two-electron integrals (labeled 3 in Figure 2) using different algorithms for cases without or with one or two singularities, which are then returned as a 64-byte accumulated results buffer with Hamiltonian, overlap and timing information to the master (labeled 4 in Figure 2) using a non-blocking MPI_iSend. Acknowledgment of the MPI_iSend is not required as the following MPI_Recv can only receive a next task if the master rank successfully processed the results buffer. The process on the worker ranks terminates after receiving from the master, instead of a new task, the corresponding terminate signal (labeled 5 in Figure 2).

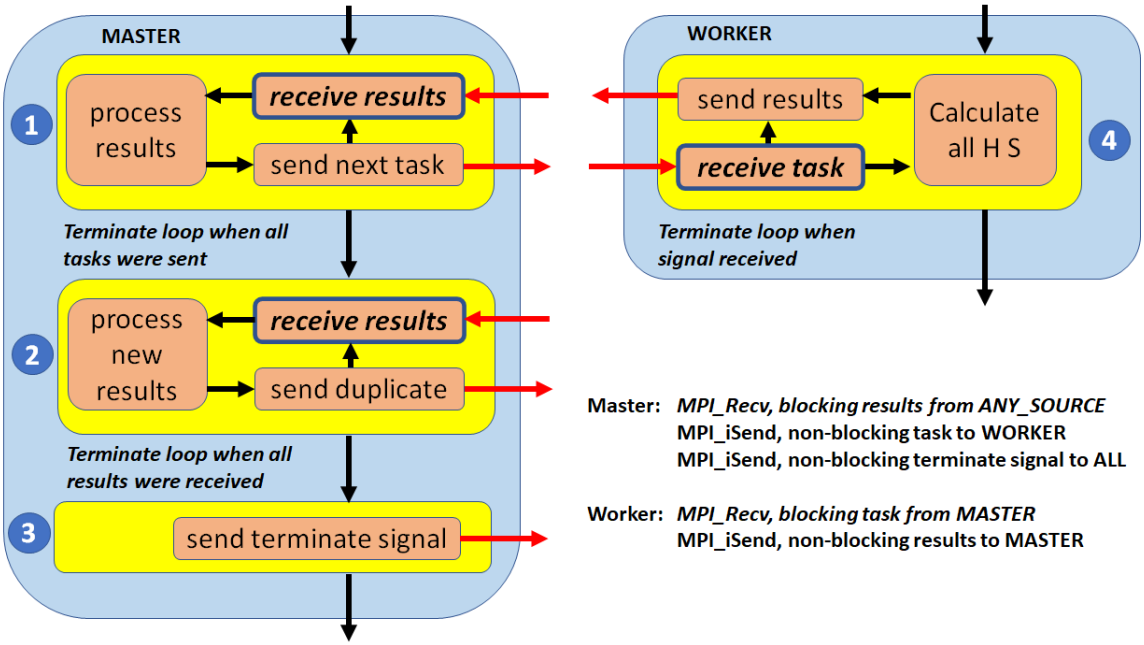


Figure 1: Schematic of the task-based execution workflow. 1: iterative process on the master rank receiving and processing results buffers and sending new tasks; 2: iterative process on the master rank receiving and processing results not previously received and sending duplicates of still outstanding tasks; 3: master rank sending terminate signals to all other ranks once all results have been received and processed; 4: iterative process on each worker rank receiving a task and returning calculated results to the master rank until a terminate signal is received from the master rank. H and S indicate the Hamiltonian and overlap matrix element contributions, respectively.

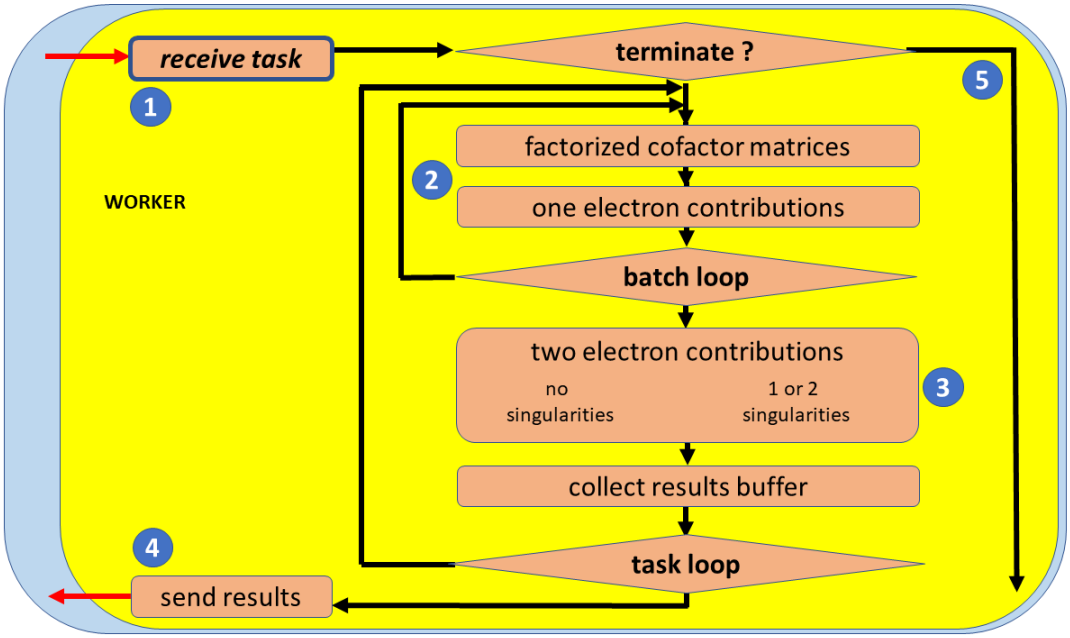


Figure 2: Schematic of the task-based execution workflow on the worker ranks. 1: task definition received from master; 2: evaluation of factorized cofactor matrices, optionally in batches, and accumulation of overlaps and one-electron contributions; 3: evaluation and accumulation of two-electron contributions requiring a single or multiple passes through the two-electron integral list for each batch; 4: returning the accumulated results back to the master rank; 5: termination after receiving the appropriate signal from the master rank.

The size of a task is user-defined, and should be chosen large enough to avoid network contention but small enough to avoid load imbalance at the end of the run when the master rank is waiting for the last few contributing results.

Fault resiliency

The definition of MPI_Recv calls on the master in loops 1 and 2 in Figure 1 is for messages from ANY_SOURCE. This guarantees that the master will not stall as long as at least one worker rank is still sending results buffers. The non-blocking MPI_iSend of tasks by the master rank is not followed by an MPI_Wait to avoid creating a synchronization between the master and any of the worker ranks. Consequently, in order to avoid premature use of the buffer sent by the non-blocking MPI_iSend for dispatching a task to a different worker rank, the master rank is using separate buffers for each worker rank it communicates with. Since the task buffer communicated by the master rank consists of only four integers, namely the Hamiltonian matrix indices and the first and last index into the determinant pair list for that element in this task, this has little consequence for the memory requirements. This design guarantees that a task buffer for a particular worker rank is only reused after the results from the previous use of the buffer was received and used by that worker rank. This is particularly important when using an MPI implementation that does not copy the communication buffer before returning from a non-blocking MPI_iSend.

The blocking MPI_Recv in loop 4 in Figure 1 on each of the worker ranks can only receive tasks from the master rank. Once a task is received, the computationally demanding calculation of Hamiltonian and overlap contributions is carried out. When completed, the results are returned to the master using a non-blocking MPI_iSend. There is no need to issue an MPI_Wait as no new task will be communicated unless the master rank has received and processed the previously sent results buffer.

Unless the run-time system captures hardware faults on the computer system and terminates the running job or the master ranks stalls, this MPI communication strategy avoids any synchronization that would stall the progress of the computation if one or more of the worker ranks would stall.

After each completion of a Hamiltonian matrix element, the master rank writes a checkpoint restart file, so that a job can be restarted in case of a power or network failure or job termination as a result of a time limit.

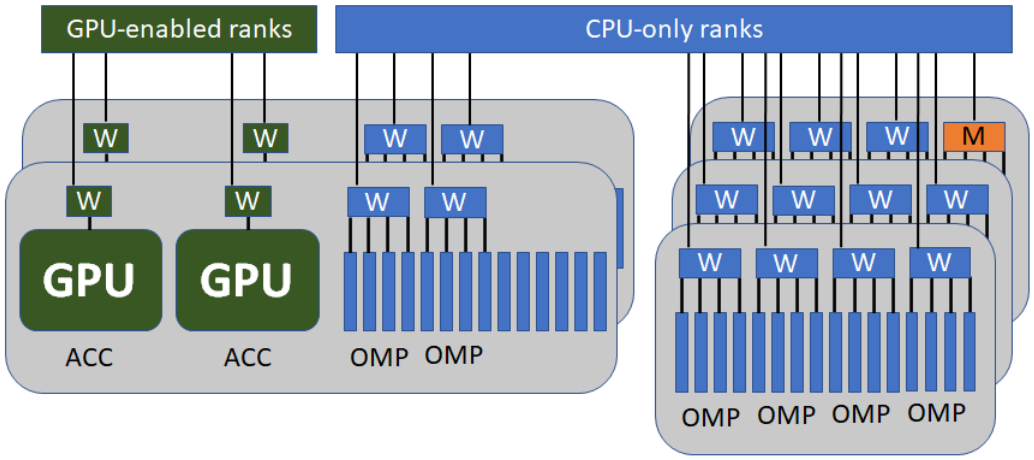


Figure 3: MPI processes are grouped into GPU-enabled (green) and CPU-only (blue) ranks. The single master rank (orange) orchestrates the distribution of tasks and collects the results returned. M, W denote master and worker ranks, respectively. ACC and OMP denote OpenACC and OpenMP instrumented code running on the labeled ranks.

Rank assignments

At the start of a GronOR calculation the available MPI ranks are grouped into GPU-enabled and CPU-only ranks, as depicted in Figure 3. The number of ranks per node should be chosen such that each rank has access to roughly the same amount of memory, and GPU-enabled ranks should have this memory available both on the node and on the accelerator. In cases where integrals need to be divided among multiple ranks, each rank sharing the integrals are of the same type, GPU-enabled or CPU-only, so that on hybrid systems the master rank can assign tasks equally. The master rank is always the last one in the list so that the assignment of ranks sharing integrals in some cases is easier to accomplish without crossing node boundaries. On systems with NVIDIA GPUs, the Multi-Process Server (MPS) capability can be used to let GPU-enabled ranks share an accelerator in situations where this optimizes memory utilization.

Task size and batch size

The number of determinant pair contributions evaluated in a single task is user-defined, and is chosen such to avoid network contention while also small enough to avoid load imbalance at the end of the run when the master rank is waiting for the last few contributing results. On CPU-only ranks that typically have access to more memory than ranks executing on accelerators, intermediate results can be accumulated within each task and used in batches when processing the two-electron integral list. The size of these batches is separately user-defined and used to manage the memory required for the intermediate arrays. In particular on CPU-only ranks the reduction in the number of times the two-electron integral list is traversed has been found to improve the computational performance significantly. The optimal choice of task and batch sizes depend on two aspects of the hardware used. The task size should be sufficiently large such that network contention resulting from many small messages is avoided while not creating load balancing towards the end of calculations from unnecessarily large tasks the master rank is waiting for to complete on a few of the worker ranks. The size of the batches is determined by the size and bandwidth of available memory and caches to worker ranks, which is typically larger on the host than the accelerator. In practical calculations done thus far, a choice of 32 contributions per task appears to be reasonable, and an effective batch size is found to be one on GPU-accelerators, where caches are small, and 32 on CPU hosts. These choices are set as defaults in GronOR and have been used in all calculations reported in this work.

Build environment

GronOR is written in Fortran90 with a small number of utility routines written in C. All communication is implemented using MPI, and architecture and accelerator specific optimizations are included through compiler directives.

Compiling GronOR is facilitated through a cmake-based build environment. GronOR has been compiled using the Intel, PGI, NVIDIA HPC SDK, IBM XL and Gnu compiler suites. The available build options are listed in Table 1. The source code has been instrumented with explicit calls

to timer routines in a timer library that is part of the GronOR git repository. Output of the collected timings is an input option.

Table 1: Options used by the cmake build system.

Option	Description	Details
OPENACC	OpenACC enabled	Recommended for GPU-accelerated computers
CUSOLVER	QR algorithms enabled	Recommended for GPU-accelerated computers
CUSOLVERJ	QR and Jacobi algorithms enabled	Recommended for GPU-accelerated computers
OPENMP	OpenMP threading enabled	
OMP_TARGET	OpenMP target offloading enabled	Mutually exclusive with OPENACC
MKL	Intel MKL library enabled	

Optimization of calculating Hamiltonian matrix contributions

Compiler directives are used to exploit parallelism within each MPI rank. For GPU accelerators the use of OpenACC or OpenMP target directives can be specified at compile time. Such directives are embedded in the source code with preprocessor macros in such a way that both the combination of OpenACC for GPU off-loading and OpenMP threading and the combination of OpenMP target off-loading and OpenMP threading can be used. This approach was chosen to ensure portability of the code to a wide range of computer architectures. In order to allow execution on hybrid architectures, i.e., simultaneous use on accelerated and CPU-only partitions, multiple versions of a number of routines exist.

For calculations in which the two-electron integral list fits more than once on the memory of available NVIDIA GPU accelerators, multiple ranks can share a GPU via the NVIDIA MPS multiple process server. This option can be used in a very flexible manner. For example, if the integral list fits three times in the combined memory of two GPUs, the integral list can be divided over two ranks, and three groups of two ranks can share the two GPUs using the MPS server with three ranks executing per GPU.

For execution on NVIDIA GPUs the singular value decomposition and eigenvalue solvers in the factorization of the second order cofactor matrices can optionally be carried out using the QR or iterative Jacobi solvers provide by the CUSOLVER library. These solvers significantly improved

performance of GronOR, in particular for molecular systems with more than 100 electrons.

Testing and validation

Nearly all parts of the code are exercised in every calculation performed. Only in the generation of the MEBF base states different paths through the routine are taken depending on the number of fragments and the spin states. The test suite contains example runs for each of these combinations. Further, the routines used in the calculation of the Hamiltonian matrix element contributions exist in two versions, an OpenMP instrumented version for execution on CPUs, and an OpenACC/OpenMP-offloading instrumented version for execution on GPUs. Both versions are compiled for GPU-enabled systems, such that GPU-accelerated ranks and CPU-only can be used in the same run, while for systems that are not GPU-enabled the CPU-only version is compiled. This build mechanism allows for optimal flexibility in using hybrid computer systems. The test suite consists of example inputs that can be used on any of these homogeneous or heterogeneous computer systems. The test suite contains examples runs that have been validated with independent Molcas State Interaction calculations. Since GronOR is build using CMake, the test suite can be run using CTest with periodic upload to a test dashboard. In addition to test input, a set of benchmark runs is run to ensure the scalability and accelerated performance. These benchmarks are run less frequently as they require more computing resources.

Accuracy and Performance of Thresholds and Workflow Parameters

Common orbital transformation threshold

The effect of the threshold in the generation of the common molecular orbital basis is illustrated for an anthracene dimer. The geometry of the fragment was optimized with BLYP/def2-SVP and the second anthracene molecule is displaced by $\Delta x = \Delta y$ = 0.97 Å, Δz = 3.75 Å with respect to

the first fragment. The atomic orbital basis used to optimize the orbitals of the fragment states contains 312 basis functions and was taken from the ANO-RCC basis library of OpenMolcas. A 4s,3p,1d contraction was used for carbon and 3s,1p for hydrogen. Each fragment wave function was obtained with an active space of 6 orbitals and 6 electrons and a threshold of 10^{-6} was used in the selection of the determinants pairs contributing to the matrix elements between the MEBFs. Six spin singlet coupled MEBFs have been constructed, including one with both fragments in the ground state, S_0S_0, two with one fragment in an excited state, S_0S_1 and S_1S_0, one with two fragments in a triplet state, T_1T_1, and two charge transfer states built from the lowest doublet fragment states, D^+D^- and D^-D^+.

Fig. 4 shows the normalized energies of the MEBFs and the number of two-electron integrals as function of the threshold (τ_{MO}) used to eliminate the linear dependencies in the common MO basis. The MEBF energies converge very fast and a threshold of 10^{-4} results in energies that are practically indistinguishable from those obtained with smaller thresholds. Taking the energies with the smallest threshold as reference, the largest deviation in the total MEBF energy is 24 meV at $\tau_{MO} = 10^{-4}$. The relative energies of the final NOCI states are even less sensitive to the value of τ_{MO}. The maximum difference in excitation energy is 6 meV for $\tau_{MO} = 10^{-4}$. For higher thresholds the deviations start to grow and the excitation energies for the higher threshold tested ($5 \cdot 10^{-3}$) differ by more than 0.5 eV with respect to the reference values obtained with the smallest threshold.

The influence of τ_{MO} on the time-to-solution can already be anticipated from the number of two-electron integrals in Fig. 4 that increase exponentially with smaller thresholds, but is quantified in a more detailed manner in Table 2. Using the smallest threshold results in a basis set of 380 functions to describe the different electronic states of the dimer, which in turn leads to a two-electron integral file of 21.0 GB. This is too large to be held in the HBM memory of a single V100 GPU used for this calculation, and therefore, the strategy of dividing the integrals over two ranks was followed, as indicated in the last column of Table 2. For $\tau_{MO} \geq 10^{-6}$, the two-electron integral file fits multiple times in the memory of the GPU. The timings in Table 2 are for 100 summit node

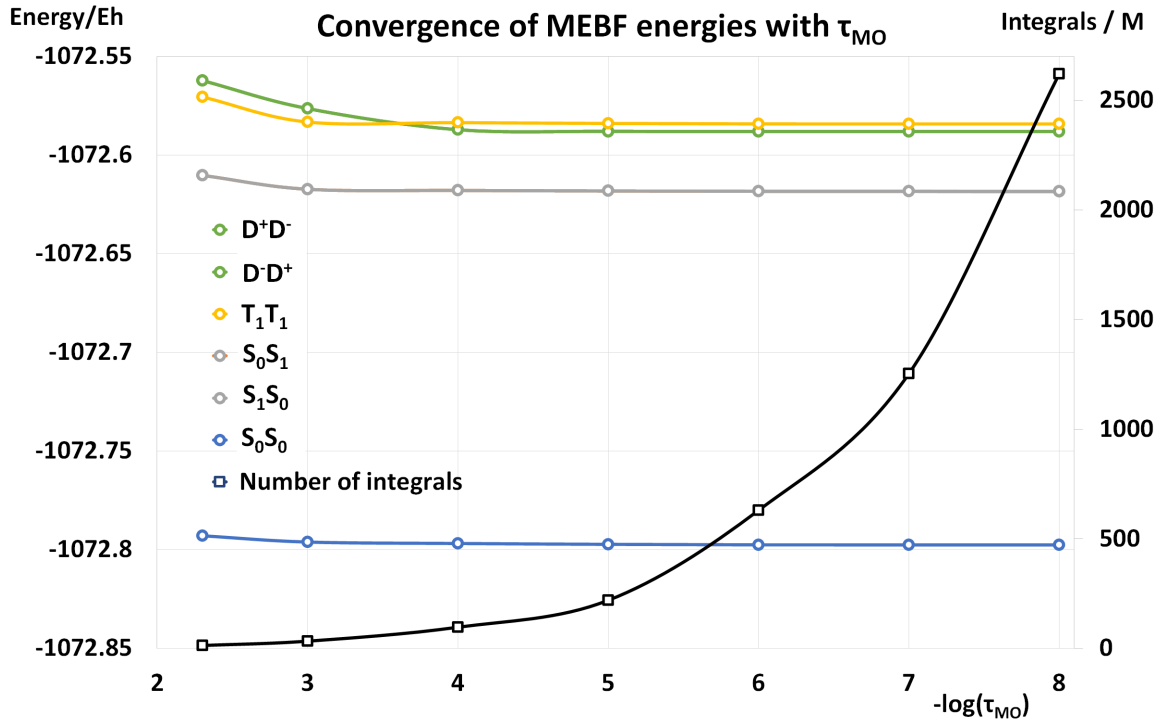


Figure 4: Convergence of the normalized MEBF energies of a anthracene dimer at 3.75 Å separation (circles; blue = S_0S_0, gray = S_0S_1/S_1S_0, gold = T_1T_1, green = D^+D^-/D^-D^+) and the number of two-electron integrals (black squares) as a function of the threshold for linear dependencies in the common MO basis.

Table 2: Wall clock time to solution (in seconds) for the calculation of the 6x6 NOCI matrix of the anthracene dimer as a function of τ_{MO} on 100 Summit nodes, each with six NVIDIA V100 accelerators.

τ_{MO}	basis functions	number of 2-el. int. in millions	Number of ranks 18 (3/GPU)	12 (2/GPU)	6 (1/GPU)	6 (1/GPU) (2/Int.List)
$5{\cdot}10^{-3}$	102	13.8	1025	1519	2989	
$1{\cdot}10^{-3}$	138	34.1	1048	1550	3043	
$1{\cdot}10^{-4}$	166	96.1	1114	1634	3203	
$1{\cdot}10^{-5}$	204	218.6	1207	1807	3536	
$1{\cdot}10^{-6}$	366	630.5		2426	4503	
$1{\cdot}10^{-7}$	316	1254.3			6232	6410
$1{\cdot}10^{-8}$	380	2620.2				13288
AO basis	624	19013.6				

runs with access to six V100 GPUs on the node, with 1, 2 or 3 ranks allocated per GPU using the MPS protocol, resulting in 6, 12 or 18 ranks used per node. The measured speed-up is close to the ideal factor of 2 and 3 for the runs with 12 and 18 ranks, respectively, demonstrating the efficiency of the MPS protocol. There is a factor of about ten between the time to solution for the $\tau_{MO} = 10^{-4}$ (18 ranks) calculation and the one with the smallest threshold, with virtually the same results obtained. Even for the smallest threshold, the number of basis functions is still only 60% of the full AO basis, which illustrates that NOCI calculations are always best executed with transformation to the common MO basis. While among the 19.0 billion two-electrons integrals expressed in the AO basis there will be an important fraction of very small integrals that can be disregarded, it is still more efficient to make a transformation to the common MO basis and use the integrals in canonical order and avoid doubling the memory requirement when using labeled integrals.

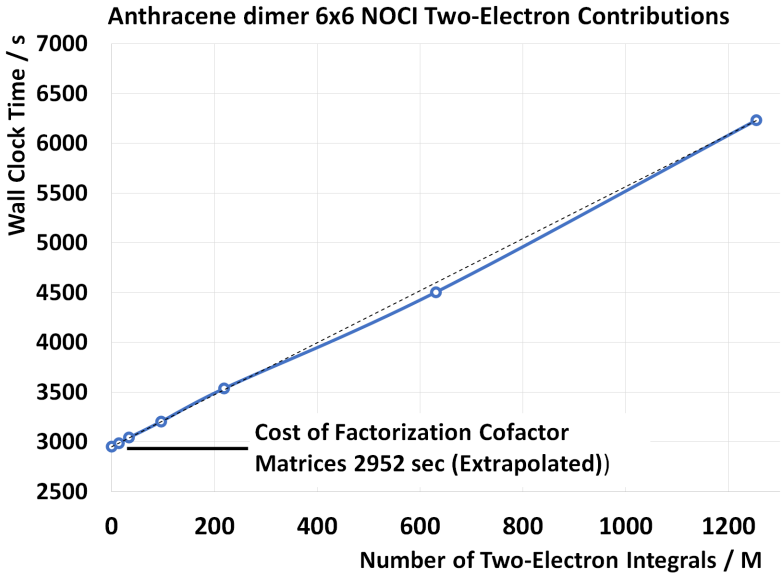


Figure 5: Wall clock cost of evaluating the two-electron contributions to the 6x6 Hamiltonian for an Anthracene dimer on 100 Summit nodes with one rank per GPU, as a function of the number of two-electron integrals resulting from using different thresholds τ_{MO}. The dashed line indicates linear behavior.

In Figure 5, the wall clock time required for evaluating the two-electron contributions to the 6x6 Hamiltonian for an Anthracene dimer on 100 Summit nodes with one rank per GPU, as a

function of the number of two-electron integrals resulting from using different thresholds τ_{MO} is depicted. The timings plotted correspond to the sixth column in Table 2. The wall clock cost of the factorization of the cofactor matrices is independent of τ_{MO} and is 2952 seconds from extrapolation to zero integrals. The cost of evaluating the two-electron integral contributions to the Hamiltonian matrix elements is demonstrated to be completely linear with the number of integrals.

Wave function expansion coefficients threshold

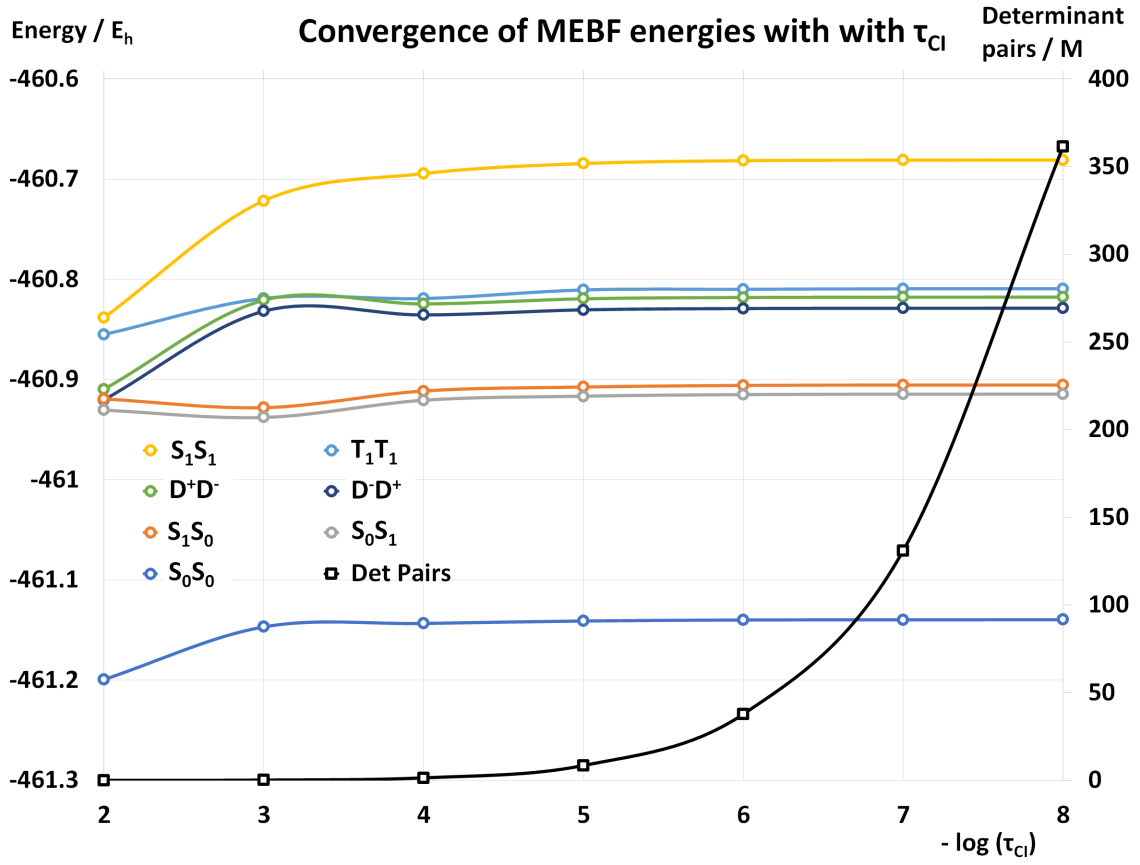


Figure 6: Convergence of the normalized state energies of a benzene dimer at 8 Ångstrom separation as a function of the CI threshold with the associated execution wall clock time on 12 Summit nodes.

A wave function expansion coefficient threshold to reduce the length of the wave function expansion is applied in two steps. First, the threshold is applied to the coefficients in the construction of the determinant expansion of the MEBFs. Second, the same threshold is applied to the gen-

eration of the determinant pair list. This threshold strongly determines the computational cost, as illustrated in Figure 6. As in the case for the common orbital transformation threshold, it is also not possible to perform the calculation for the full list of determinant pairs. Therefore, the calculation with the smallest threshold ($\tau_{CI} = 10^{-8}$) is taken as reference to check the stability of the results with larger thresholds. The relative MEBF energies stay nearly constant up to τ_{CI} values of 10^{-5}, for which the largest deviation is found for the S_1S_1 MEBF differing by 0.06 eV from the reference value. Increasing the threshold to 10^{-4} decreases the relative energy of the S_1S_1 and ^1TT MEBFs by 0.26 eV and 0.17 eV, while the changes in other MEBFs are still smaller than 0.1 eV. Larger thresholds lead to unreliable results. Figure 6 also shows how the number of determinant pairs decreases rapidly with increasing threshold, leading to a reduction of a factor of $\sim$50 for $\tau_{CI} = 10^{-5}$ compared to the reference value.

More examples of the dependency of the results on the two thresholds described in this and the previous subsection can be found in the main text and the supplementary information of Refs. 21 and 35. Based on the experience gained by those test cases, the ones described in the present study and other (so far unpublished) calculations, the default values of τ_{MO} and τ_{CI} have been put to 10^{-5}, which is somewhat on the safe side, especially for τ_{MO}, where a ten times larger threshold would not lead to noticeable changes in most cases.

Singular value decomposition and eigenvalue solvers

The GronOR implementation relies on two linear algebra solvers, a singular value decomposition and an eigenvalue solver. Optimized versions of these solvers for NVIDIA accelerators are available in the CUSOLVER library, both as a QR and an iterative Jacobi implementation. The relative computational efficiency of these implementations depends on the size of the symmetric matrices, i.e., for GronOR calculations the number of electrons in the system. In Figure 7 this efficiency relative to the CPU-only versions is given as a function of the number of electrons for a 7x7 Hamiltonian for the dimers of benzene (84 electrons), naphthalene (136), anthracene (188), tetracene (240) and pentacene (292) with a single rank per GPU, and of a 4x4 Hamiltonian for the

dimers of 5,5'-difluoro-indigo (304), 5,5'-dichloro-indigo (336) and 5,5'-dibromo-indigo (408) with two ranks per GPU on the JFZ Juwels Booster. Calculations of the systems with 188 or more electrons show an increasing speedup with the number of electrons when using the GPU-resident CUSOLVER routines. The indigo derivatives were run with two ranks sharing a GPU to increase the overall efficiency of the runs. The slight drop in speedup between a single and dual ranks on a GPU indicates the competition for resources on the device. While below 188 electrons the matrices are too small for the CUSOLVER routines to compete with the CPU-resident routines, for the larger systems the CUSOLVER routines are significantly more efficient. The speedup results presented in Figure 7 are for the entire GronOR NOCI calculations, and not for the solver component only.

Freezing core orbitals

Another approach that can reduce time-to-solution is to remove the core orbitals from the NOCI computation. To deal with the fact that the core orbitals are not strictly the same for all electronic states, the average density matrix due to the core electrons is constructed. The eigenvectors of this average density matrix are used to represent the core orbitals of all electronic states before the transformation of the integrals to the common MO basis. The energy contribution of these frozen core electrons is absorbed in the nuclear potential energy and their interaction with the valence electrons in the one- and two-electron integrals.

Table 3: Comparing the NOCI results of an anthracene dimer with and without freezing the 28 C-1s orbitals. γ (in meV) is the electronic coupling between the $S_0S_1$and T_1T_1 MEBFs and ΔE_{ij} (in eV) are the relative NOCI energies. SNH stands for Summit node hours.

	C-1s not frozen	C-1s frozen	difference
γ	2.492	2.491	$0.98{\cdot}10^{-3}$
$\Delta E_{S_0S_1}$	4.80	4.80	$1.3{\cdot}10^{-3}$
$\Delta E_{T_1T_1}$	5.69	5.69	$-4.7{\cdot}10^{-5}$
SNH	31.0	11.6	19.4
number of 2-el. integrals	96,070,591 (845 MB)	45,998,436 (405 MB)	

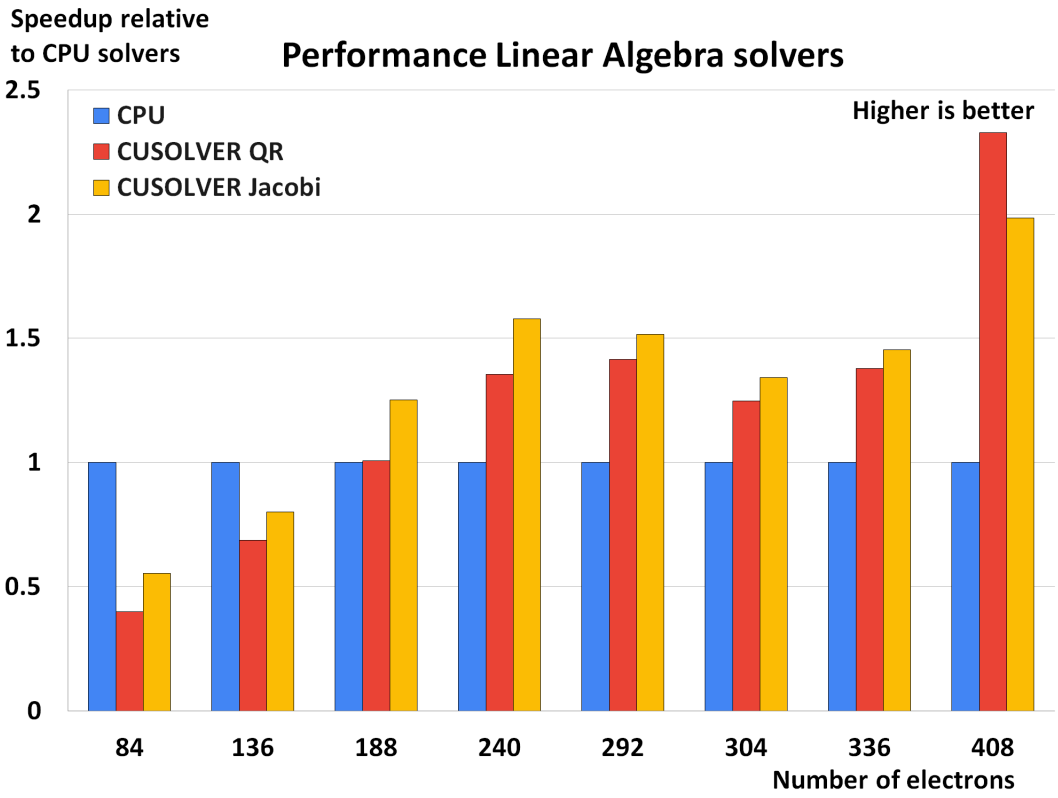


Figure 7: Speedup resulting from using the CUSOLVER QR or Jacobi singular value decomposition and eigenvalue decomposition solvers relative to using the CPU resident SVD and TRED2/TQL solvers, as a function of the number of electrons. Results obtained for GronOR calculations of a 7x7 Hamiltonian for benzene (84), naphthalene (136), anthracene (188), tetracene (240) and pentacene (292) dimers with a single rank per GPU, and of a 4x4 Hamiltonian for 5,5'-difluoro-indigo (304), 5,5'-dichloro-indigo (336) and 5,5'-dibromo-indigo (408) with two ranks per GPU on the JFZ Juwels Booster.

The results in Table 3, obtained for the previously described anthracene dimer with $\tau_{CI} = 10^{-6}$ and $\tau_{MO} = 10^{-4}$, show that freezing the core orbitals does not affect the outcomes of the NOCI calculation, both the electronic couplings (only one is shown here, but all other couplings follow the same trend) and the relative energies of the NOCI wave functions are virtually identical with and without freezing the core. The only aspect that undergoes a significant change is the time-to-solution, presented in the table as Summit node hours (SNH). The calculations were done on 100 nodes with 18 accelerated ranks per node in both cases, leading to a wall-clock time-to-solution of 1116 and 418 seconds, respectively. A speed-up was measured of a factor of 2.7 by removing the 28 C-1s orbitals of the anthracene dimer, mostly caused by the reduction in the number of two-electron integrals from 96M to 45M.

Scalability and Accelerated Performance on Supercomputers

Supercomputer Systems Used in the Benchmarks

For the GronOR performance analysis, benchmark calculations were performed on massively parallel hybrid supercomputers representing two generations of NVIDIA GPU accelerators, namely the Volta V100 on Summit and the Ampere A100 on Juwels Booster.

Summit at OLCF

Summit is the 4670-node accelerated supercomputer at the Oak Ridge leadership Computing Facility (OLCF) at the Department of Energy's Oak Ridge National Laboratory in Oak Ridge, Tennessee, USA. The IBM AC922 nodes consist of two IBM POWER9 CPUs and six NVIDIA Volta V100 GPU accelerators. A fast on-node NVIDIA NVLINK interconnect provides 50 GB/s bandwidth between each CPU and three GPU accelerators. The single coherent memory domain consists of 512 GB of DDR-4 memory and 16 GB of high bandwidth memory (HBM2) on each of the GPUs. The aggregate peak performance of Summit is more than 200 PFlop, which placed the system first on the TOP-500 list of supercomputers from June 2018 to November 2019, and is at

present the most powerful supercomputer in the United States.

Juwels Booster at JSC

The Juwels Booster module is a 936-node partition of the Juwels supercomputer at the Jülich Supercomputer Center (JSC) at the Jülich Forschungszentrum (JFZ) in Jülich, Germany. The Bull Sequana XH2000 Juwels Booster module consists of nodes with two AMD EPYC Rome 7402 CPUs and four NVIDIA Ampere A100 accelerators. Each node has 512 GB of memory and each GPU has 40 GB of high bandwidth memory (HBM2e). The peak performance of the Booster module partition of Juwels is 73 PFlop. Juwels Booster placed number 7 on the TOP-500 list of supercomputers in November 2020, and is at present the fastest supercomputer in Europe.

Benchmark Results

Naphthalene Dimer

The very first implementation of GronOR demonstrated the flexibility of combining independently optimized multi-configuration fragment wave function expansions for NOCI studies and illustrated how the algorithm can be effectively used on parallel, accelerated computers.[8] With the task-based design and significant parts of the code base ported to GPU accelerators, code performance was demonstrated for a dimer of naphthalene molecules in which the wave function was expressed as the product of molecular wave functions at the CAS(8,8) for one, and CAS(4,4) for the other molecule. This original benchmark calculation of a 4x4 Hamiltonian $(S_0S_0, S_0S_1, S_1S_0, T_1T_1)$ of the CAS(8+4,8+4) naphthalene dimer required 1,771 seconds on 4604 nodes on Summit, or 2,265 Summit node-hours.[8] Based on those promising initial performance characteristics of the code, the additional methodological and implementation improvements were made as described in the current work. These improvements, in particular the reduction of the list of two-electron integrals as a result of the common molecular orbital transformation, the reduction of the determinant pairs lists as a result of applying a threshold on the CI expansion coefficients, the transfer of all

computationally demanding parts of the calculations to the GPU accelerators and the use of GPU-accelerated optimized solver libraries, have resulted in the significant performance improvements reported here. With the current implementation of GronOR, the same naphthalene dimer benchmark now requires 881 seconds on 32 Summit nodes, or 7.8 Summit node-hours. This is a 290-fold improvement of the computational efficiency and demonstrates how the recent methodological developments and code optimizations make the use of NOCI calculations feasible for much larger molecular systems.

Indolonaphthyridine Dimer

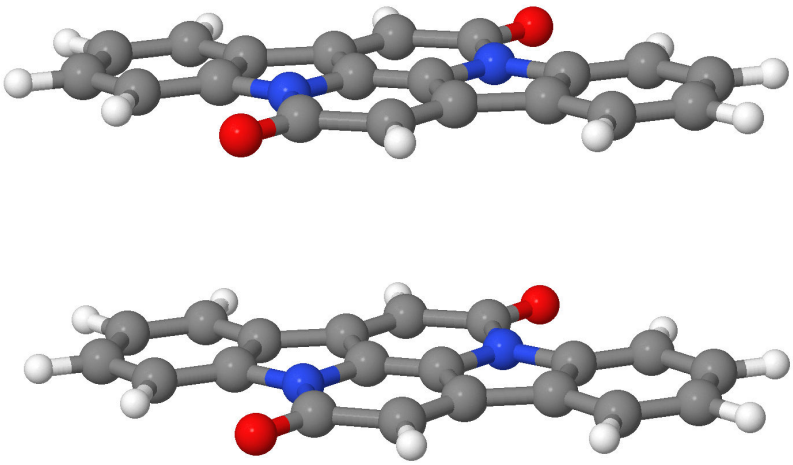


Figure 8: Indolonaphthyridine dimer at 4.0 Ångstrom separation

Indolonaphthyridine derivatives have been suggested as potential candidates for singlet fission capable molecules in photovoltaic materials.[50] One such derivative is cibalackrot which has been studied using non-orthogonal approaches by Ryerson et al.[51]

A dimer of the core indolonaphthyridine at 4.0 Å separation, shown in Figure 8, with 320 electrons was used here to obtain GronOR release 21.04 benchmarks.

Total execution times, including setup, data transfer, and evaluation of the full Hamiltonian matrix, as a function of the number of nodes used for the calculation of the 7x7 Hamiltonian (S_0S_0, S_1S_0, S_0S_1, S_1S_1, T_1T_1, D^-D^+, D^+D^-) with fragment states S_0, S_1, T_1, D^- and D^+ obtained at the

Table 4: Timings in seconds obtained for the indolonaphthyridine dimer at CAS(4+4,4+4) as a function of nodes on Summit.

	CPU-only			GPU + CPU solvers		
Nodes	Setup	Hamiltonian	Total	Setup	Hamiltonian	Total
16				9.634	3790.689	3800.324
32				12.820	1888.576	1901.395
64	9.423	4957.121	4966.544	11.550	944.697	956.246
128	10.112	2495.345	2505.458	10.643	476.305	486.947
256	10.579	1330.581	1341.161	11.655	238.943	250.598
	GPU + cu-QR			GPU + cu-Jacobi		
Nodes	Setup	Hamiltonian	Total	Setup	Hamiltonian	Total
16	11.115	1894.798	1905.905	10.742	1182.536	1192.291
32	11.134	946.032	956.166	10.796	593.079	603.875
64	10.342	473.336	483.679	10.172	295.478	306.174
128	10.503	238.324	248.826	11.060	149.962	159.984
256	11.101	120.443	131.544	11.602	75.429	86.582

Table 5: Timings obtained for the indolonaphthyridine dimer at CAS(4+4,4+4) as a function of nodes on Juwels.

	GPU + cu-Jacobi		
Nodes	Setup	Hamiltonian	Total
16	13.278	914.969	928.247
32	13.921	458.176	472.097
64	13.891	230.466	244.356
128	13.140	117.118	130.258
256	14.294	60.417	74.711

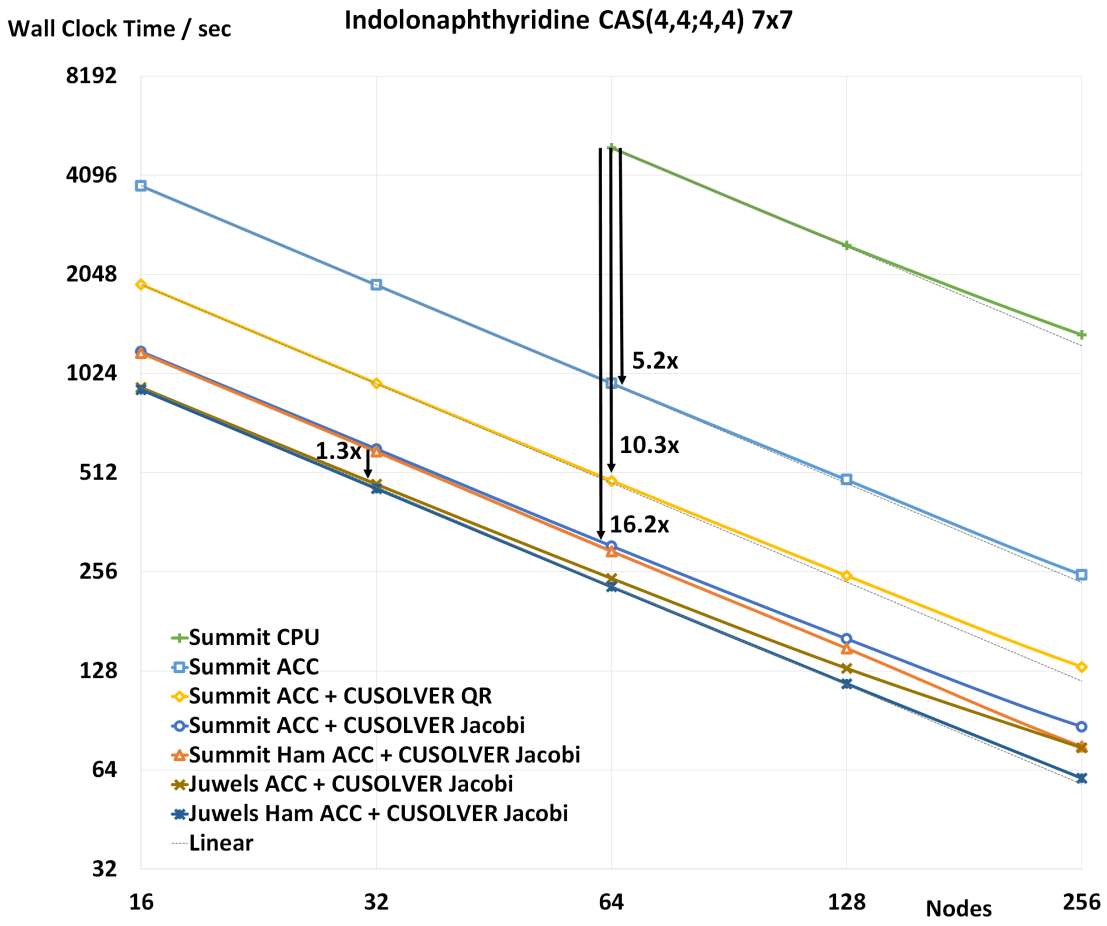


Figure 9: Accelerated performance on Summit of the indolonaphthyridine dimer 7x7 Hamiltonian at CAS(4+4,4+4), showing a 5.2-fold speedup of the OpenACC instrumented code, 10.3-fold speedup using the cusolver QR solvers, and 16.2-fold speedup using the cusolver iterative Jacobi solvers used on six ranks per node, compared to CPU-only runs with 30 ranks per node. Dashed lines indicate ideal (i.e., linear) scaling. Runs using the cusolver Jacobi solvers are 1.3 times faster on Juwels compared to Summit.

CAS(4,4) level using OpenMolcas are given in Tables 4 and 5 for Summit and Juwels respectively, and in Figure 9. Calculations at the CAS(4+4,4+4) level significantly reduce the number of determinant pairs contributing to Hamiltonian and overlap matrices compared to more accurate and more representative calculations at the CAS(8+8,8+8) level, and makes it reasonable to still carry out CPU-only runs to assess the performance impact of GPU-acceleration. The computational requirement for each individual determinant pair is comparable for both cases, however. This makes the CAS(4+4,4+4) calculation a good test case to illustrate the improvement of performance from GPU acceleration. On 64 nodes, GPU acceleration on Summit using 6 ranks per node with SVD and EVD solvers still carried on the CPU leads to a speedup factor of 5.2 compared to a 30 rank per node optimized CPU-only run. When in addition the the SVD and EVD are executed on the GPU using the QR or Jacobi solvers provided by the CUSOLVER library the speedup factors are 10.3 and 16.2, respectively. The efficiency of the cusolver SVD and EVD solvers, compared to the same solvers running on the CPU, increases with the size of the input matrices which in GronOR depends on the number of electrons in the molecular system. For the indolonaphthyridine dimer with 320 electrons the iterative cusolver Jacobi solvers are clearly more efficient than either the CPU-based or cusolver QR solvers. For systems with around 100 electrons, the cusolver routines exhibit similar efficiency as the CPU-based solvers, and for systems with fewer electrons the cusolver routines are less efficient.

For the calculations using the different solver options, the execution wall clock times measured on Summit for the setup, the Hamiltonian calculation and the total wall clock times are given in Table 4, and for the calculations using the cusolver Jacobi solvers depicted in Figure 9. Setup time is defined as the wall clock time measured on the master rank between the start of the calculation and the first request for a task from one of the worker ranks. The setup time is more or less constant, appears independent of the number of nodes used in the calculation, but becomes on the larger node-counts a significant fraction of the total wall clock time because of the relatively short execution time needed to calculate the Hamiltonian matrix elements for this small CAS space. The scalability of the Hamiltonian calculation remains near-ideal for this chemical system from 16 to

256 nodes.

A comparison of V100 and A100 GPU runs using the cusolver Jacobi solvers on Summit and the Juwels Booster system, respectively, is given in Figure 10. The Juwels Booster A100 GPUs are about 15% faster than the V100, but with six V100 GPUs per node on Summit and four A100 GPUs per node on Juwels-Booster, per node Summit is expected to be 1.3x faster. While calculations are performed as much as is possible on the GPUs, wave function coefficients need to be transferred from host to device memory at the start of each determinant pair calculation, and OpenACC GPU kernel invocation overheads also involve communication between host and device. As a result, Summit nodes are in practice found to be 1.5 times faster, which in part can be explained by the faster communication between host and device, which is NVLINK on Summit and PCIe on Juwels. More effective end-to-end connectivity on Summit also appears to contribute to better observed scalability on Summit. For both computer systems, however, the setup time becomes a noticeable fraction of the total execution time for this small benchmark.

Table 6: Timings obtained for the indolonaphthyridine dimer at CAS(8+8,8+8) as a function of nodes on Summit.

	GPU + cu-Jacobi				
Nodes	Setup	Hamiltonian	Total	Eff. Ham.	Eff. Total
128	36.097	19281.201	19317.298	100	100
256	32.543	9684.957	9717.501	99.5	99.4
512	36.991	4831.476	4868.458	99.8	99.2
1024	39.205	2419.037	2458.242	99.6	98.2
2048	44.371	1212.410	1256.781	99.4	96.1
3072	36.214	807.297	843.511	99.5	95.4
4096	39.707	605.582	645.289	99.5	93.5

Table 7: Timings obtained for the indolonaphthyridine dimer at CAS(8+8,8+8) as a function of nodes on Juwels.

	GPU + cu-Jacobi				
Nodes	Setup	Hamiltonian	Total	Eff. Ham.	Eff. Total
128	26.314	16357.383	16383.516	100	100
256	27.904	8171.100	8199.758	100.1	99.9
384	28.107	5460.332	5488.439	99.9	99.5
512	28.969	4097.502	4216.471	99.8	97.1

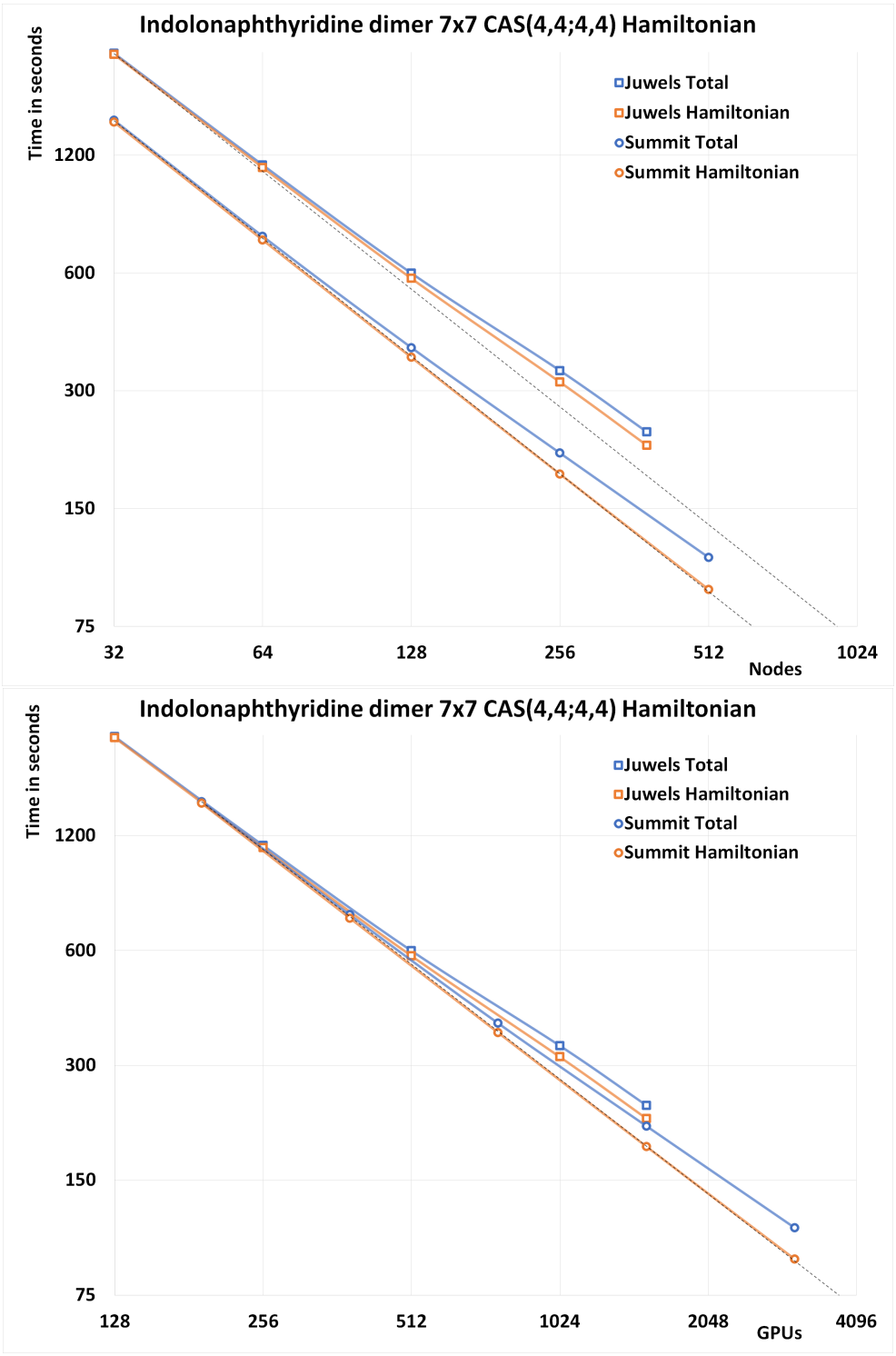


Figure 10: Timings for the Hamiltonian calculation and total execution wall clock times on Summit and Juwels for the indolonaphthyridine dimer 7x7 Hamiltonian at CAS(4+4,4+4), showing the relative performance of the Summit V100 and Juwels-Booster A100 accelerators. Top panel: Performance as function of nodes. Bottom panel: Performance as function of GPUs. Dashed lines indicate ideal, i.e., linear, scaling. The relationship between timings per node and per GPU is 4:1 for Juwels Booster, and 6:1 for Summit.

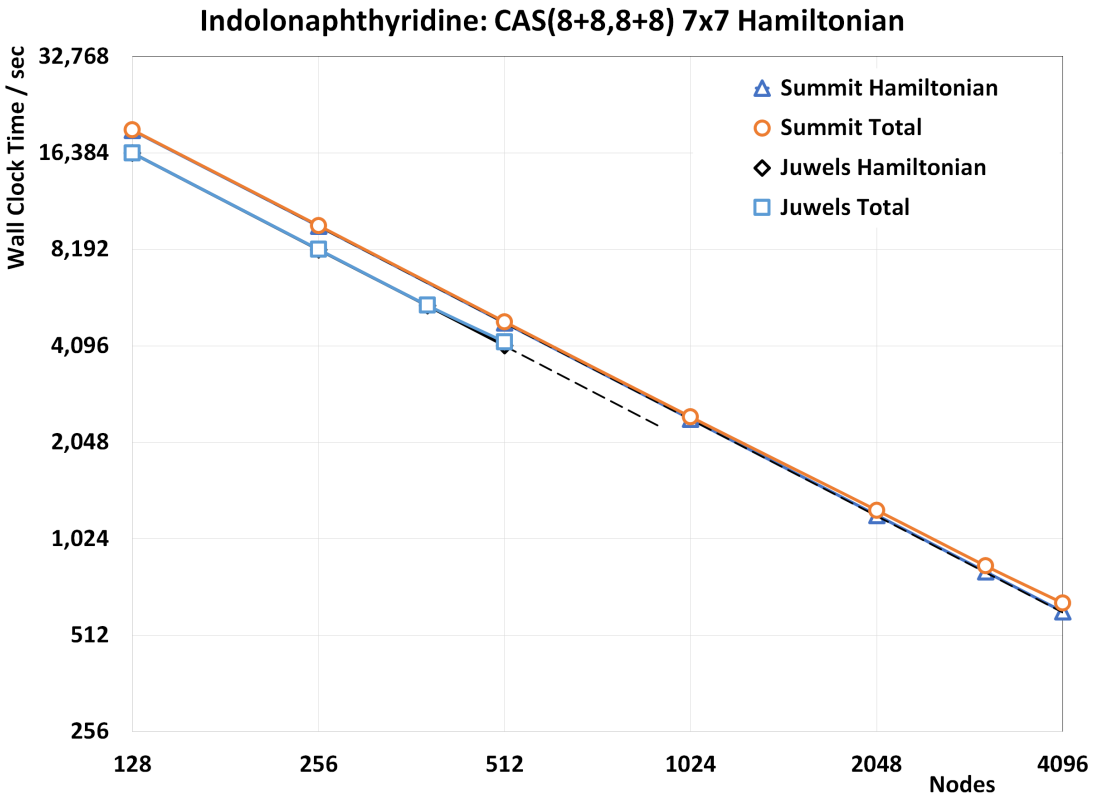


Figure 11: Timings for the Hamiltonian calculation and total execution wall clock times on Summit and Juwels for the indolonaphthyridine dimer 7x7 Hamiltonian at CAS(8+8,8+8), showing a parallel scalability efficiency of 93.5% when run on 4096 Summit nodes and of 97.1% when run on 512 Juwels nodes, with 128 node runs as the baseline.Dashed lines indicate ideal (i.e., linear) scaling, which for Juwels is shown up to 938, the number of GPU-capable nodes.

The second indolonaphthyridine dimer benchmark was carried out at the CAS(8+8,8+8) level of wave function expansion, which represents a more common production usage of the method. Timings obtained on 128 to 4096 Summit nodes and on 128 to 920 Juwels nodes are given in Tables 6 and 7, respectively, and depicted in Figure 11. With 128 node runs as baseline, the parallel scalability efficiency of the Hamiltonian calculation is near-ideal at 99.5% and 99.8% on 4096 Summit nodes and 512 Juwels nodes, respectively. The setup time on both systems is also for these calculations independent of the number of nodes used, and leading to total parallel scalability efficiencies of 93.5% and 97.1% on Summit and Juwels, respectively.

Heterogeneous Computer Systems

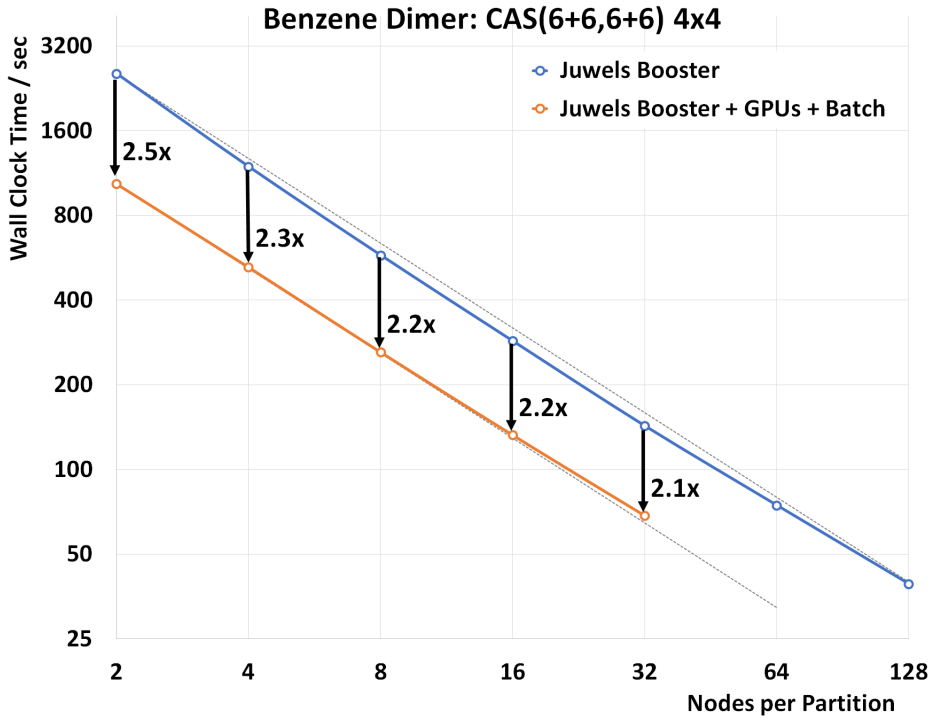


Figure 12: Timings for total execution wall clock times for a 4x4 Hamiltonian calculations for a benzene dimer at CAS(6+6,6+6) executed on the Juwels Booster partition compared to a hybrid calculation on the Juwels resource running across the Juwels Booster, Juwels GPUs and Juwels Cluster partitions. The Juwels Booster partition runs (blue curve) use four ranks per node, i.e., four GPUs per node as given on the horizontal axis. In the hybrid runs (orange curve) for each data point the same number of nodes on the V100 and cluster partitions are added, i.e., four A100 GPUs, four V100 GPUs and the cluster CPUs for each of the nodes on the horizontal axes.

The task-based implementation of the algorithm in GronOR with completely independent and asynchronous execution of contributions by worker ranks makes it possible to effectively run NOCI-F calculations on heterogeneous compute clusters. The Juwels machine at JFZ is an example of a compute resource with CPU and GPU partitions on the same interconnect. To illustrate how GronOR can take effective advantage of different partitions, a small benzene dimer at CAS(6+6,6+6) calculation was conducted across different partitions and compared with use of the Juwels Booster partition only. Two sets of runs were conducted. As baseline, a Juwels Booster only runs were carried out on 2 to 128 nodes using 4 ranks per node, each assigned to one of the A100 GPUS. Slightly super-linear scalability is observed caused by the single master rank allocated but not using a GPU. Timings are compared in Figure 12 with timings for similar runs using in addition the same number of nodes on the V100 GPU and CPU-only Cluster partitions, each with the same number of ranks per node. In these runs the master rank was executed on a CPU-only node of the Cluster partition, and no super-linear scalability is observed. Timings for the hybrid runs are, as expected, more than two times faster. For each data point in Figure 12, the comparison is between using, per node, four A100 GPUs and four each of A100, V100 GPUs and four CPU ranks. For example, the speedup given in Figure 12 is comparing 2 nodes on the Booster with six nodes used in hybrid mode, i.e., 2 nodes on the Booster, 2 nodes on the V100 partition, and 2 nodes on the CPU-only partition. In the hybrid runs, the A100 GPUs were found to consistently contribute 47%, the V100 GPUs 49%, and the CPU-ranks 4% of the total number of matrix element contributions. Since the A100 GPUs are faster than the V100 GPUs, this result could again point to more efficient data transfer on Summit's NVLINK compared to the PCIe bus on Juwels Booster.

Clusters and Workstations

While large-scale massively parallel accelerated supercomputers have been the primary target in the development of GronOR, the code also runs quite effectively on small compute clusters and workstations for small molecular problems. On computer systems with a single or a few GPU

accelerators the same CUSOLVER routines can be used together with the NVHPC compilers, while for non-accelerated systems the MKL solver routines with the Intel compilers, or the solver routines included in the software with Gnu compilers can be used.

Tetracene is one of the original materials found to exhibit singlet fission, and still serves as a key material in computational studies.[52] A tetracene dimer, with 240 electrons, represents the upper limit of system sizes that can be studied on small computer systems with one or a few GPU-accelerated nodes. Production calculations of the 6x6 Hamiltonian of a tetracene dimer at the CAS(8+8,8+8) level could be carried out on a departmental cluster node with two NVIDIA V100 accelerators in just over 24 node-hours using 10 accelerated ranks.

Discussion

Non-orthogonal configuration interaction based on independently optimized multi-configuration diabatic fragment wave functions provides a rigorous and effective methodology for the computational study of a range of important processes involving multiple electron excitation, electron and excitation energy transfer, exciton diffusion, intermolecular Coulombic decay. Understanding these and other processes is fundamental to finding new techniques for exploiting photosynthesis for novel materials or photo-excitation dynamics for energy conversion in photovoltaic devices. While non-orthogonal methods are substantially more computationally demanding than their orthogonal alternatives, they offer a number of advantages in the form of simpler expansions of wave functions in terms of constituent fragment states and a much more intuitive interpretation of the importance of the fragment states for NOCI calculated properties. The motivation for the development of GronOR described in this work is to provide highly efficient computational software that make NOCI-F calculations feasible for sufficiently large molecular systems to benefit research focused on energy conversion and transfer applications.

Currently, GronOR scalability and accelerated performance has been demonstrated on NVIDIA GPU accelerators. Once available to the development team, GronOR is planned be ported using

the same strategies to other accelerators, such as the AMD and Intel GPUs which are expected to be part of the next generation of supercomputers.

GronOR is available as open source application under the Apache 2.0 license to the scientific community from its GitLab repository.[23]

Acknowledgments

This work used resources of the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, through the Director's Discretionary program and the INCITE project chm154.

Benchmark calculations on the Juwels Booster module and Juwels GPUs and Cluster partitions at the Jülich Supercomputer Center (JSC) were made possible through test project 22180 granted through the Gauss Center for Supercomputing (GCS), as well as production runs carried out under the PRACE project 2021240033/pra129. The authors thank Dr. Herten of JSC for his support and assistance.

This work was supported in part by the (Shell NWO) research program of the Foundation for Fundamental Research on Matter (FOM), which is part of the Netherlands Organization for Scientific Research (NWO).

Financial support has also been provided by the Spanish Administration (Projects PID2020-113187GB-I00, RTI2018-095460-B-I00 and MDM-2017-0767) and the Generalitat de Catalunya (Projects 2017-SGR629 and 2017SGR13).

This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public

access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

1. Nozik, A. J.; Beard, M. C.; Luther, J. M.; Law, M.; Ellingson, R. J.; Johnson, J. C. Semiconductor Quantum Dots and Quantum Dot Arrays and Applications of Multiple Exciton Generation to Third-Generation Photovoltaic Solar Cells. *Chem. Rev.* **2010**, *110*, 6873–6890.

2. Jahnke, T.; Hergenhahn, U.; Winter, B.; Dörner, R.; Frühling, U.; Demekhin, P. V.; Gokhberg, K.; Cederbaum, L. S.; Ehresmann, A.; Knie, A.; Dreuw, A. Interatomic and Intermolecular Coulombic Decay. *Chem. Rev.* **2020**, *120*, 11295–11369.

3. Feron, K.; Belcher, W. J.; Fell, C. J.; Dastoor, P. C. Organic Solar Cells: Understanding the Role of Förster Resonance Energy Transfer. *Int. J. of Mol. Sci.* **2012**, *13*, 17019–17047.

4. Scholes, G. D.; Fleming, G. R.; Olaya-Castro, A.; van Grondelle, R. Lessons from nature about solar light harvesting. *Nature Chemistry* **2011**, *3*, 763–774.

5. Hsu, C.-P. The Electronic Couplings in Electron Transfer and Excitation Energy Transfer. *Acc. Chem. Res.* **2009**, *42*, 509–518.

6. You, Z.-Q.; Hsu, C.-P. Theory and Calculation for the Electronic Coupling in Excitation Energy Transfer. *Int. J. Quantum Chem.* **2014**, *114*, 102–115.

7. Chou, H.-H.; Yang, C.-H.; T., L. J.; Hsu, C.-P. First-Principle Determination of Electronic Coupling and Prediction of Charge Recombination Rates in Dye-Sensitized Solar Cells. *J. Phys. Chem. C* **2016**, 983992.

8. Straatsma, T. P.; Broer, R.; Faraji, S.; Havenith, R. W. A.; Aguilar Suarez, L. E.; Kathir, R. K.; Wibowo, M.; de Graaf, C. GronOR: Massively Parallel and GPU-Accelerated Non-Orthogonal Configuration Interaction for Large Molecular Systems. *J. Chem. Phys.* **2020**, *152*, 064111.

9. Thom, A. J. W.; Head-Gordon, M. Hartree-Fock solutions as a quasidiabatic basis for nonorthogonal configuration interaction. *J. Chem. Phys.* **2009**, *131*, 124113.

10. Yost, S. R.; Kowalczyk, T.; van Voorhis, T. A multireference perturbation method using non-orthogonal Hartree-Fock determinants for ground and excited states. *J. Chem. Phys.* **2013**, *139*, 174104.

11. Morrison, A. F.; Herbert, J. M. Analytic derivative couplings and first-principles exciton/phonon coupling constants for an ab initio Frenkel-Davydov exciton model: Theory, implementation, and application to compute triplet exciton mobility parameters for crystalline tetracene. *J. Chem. Phys.* **2017**, *146*, 224110.

12. Kähler, S.; Olsen, J. Dynamic correlation for non-orthogonal reference states: Improved perturbational and variational methods. *J. Chem. Phys.* **2018**, *149*, 144104.

13. Oosterbaan, K. J.; White, A. F.; Head-Gordon, M. Non-Orthogonal Configuration Interaction with Single Substitutions for Core-Excited States: An Extension to Doublet Radicals. *J. Chem. Theory Comput.* **2019**, *15*, 2966–2973.

14. Nite, J.; Jiménez-Hoyos, C. A. Low-Cost Molecular Excited States from a State-Averaged Resonating Hartree-Fock Approach. *J. Chem. Theory Comput.* **2019**, *15*, 5343–5351.

15. Glebov, I. O.; Kozlov, M. I.; Poddubnyy, V. V. Comparison of the Coulomb and non-orthogonal appraoches to the construction of the exciton Hamiltonian. *Comput. Theor. Chem.* **2019**, *1153*, 12–18.

16. Burton, H. G. A.; Thom, A. J. W. Reaching Full Correlation through Nonorthogonal Configuration Interaction: A Second-Order Perturbative Approach. *J. Chem. Theory Comput.* **2020**, *16*, 5586–5600.

17. Burton, H. G. A. Generalized nonorthogonal matrix elements: Unifying Wick's theorem and the Slater–Condon rules. *J. Chem. Phys.* **2021**, *154*, 144109.

18. Zhao, R.; Grofe, A.; Wang, Z.; Bao, P.; Chen, X.; Liu, W.; Gao, J. Dynamic-then-Static Approach for Core Excitations of Open-Shell Molecules. *J. Phys. Chem. Lett.* **2021**, *12*, 7409–7417.

19. Mahler, A. D.; Thompson, L. M. Orbital optimization in nonorthogonal multiconfigurational self-consistent field applied to the study of conical intersections and avoided crossings. *J. Chem. Phys.* **2021**, *154*, 244101.

20. Malrieu, J.-P.; Durand, P.; Daudey, J.-P. Intermediate Hamiltonians as a new class of effective Hamiltonians. *J. Phys. A* **1985**, *18*, 809–826.

21. Sánchez-Mansilla, A.; Sousa, C.; Kathir, K. R.; Broer, R.; Straatsma, T. P.; de Graaf, C. On the role of dynamic correlation in the electronic coupling calculated through nonorthogonal configuration interaction with fragments. *Phys. Chem. Chem. Phys.* **2022**, *in press, 10.1039/D2CP00772J*.

22. Straatsma, T. P.; Broer, R.; de Graaf, C.; Sousa, C.; Sánchez-Mansilla, A.; Kathir, K. R. `http://www.gronor.org/`.

23. `https://gitlab.com/gronor/gronor`.

24. Straatsma, T. P.; Broer, R.; Faraji, S.; Havenith, R. W. A. GronOR Nonorthogonal Configuration Interaction Calculations at Exascale. *Ann. Rep. Comput. Chem.* **2018**, *14*, 77–91.

25. Accelerated Data Analytics and Computing Institute. `https://adac.ornl.gov/`.

26. Álvarez-Moreno, M.; de Graaf, C.; Lopez, N.; Maseras, F.; Poblet, J. M.; Bo, C. Managing the Computational Chemistry Big Data Problem: The ioChem-BD Platform. *J. Chem. Inform. Mod.* **2015**, *55*, 95–103.

27. ioChem-BD is a public web-based repository for computational chemistry calculations developed by the Institute of Chemical Research of Catalonia (ICIQ) in collaboration with the Rovira i Virgili University (URV), Tarragona, Spain. `http://iochem-bd.org`.

28. Frenkel, J. On the Transformation of light into Heat in Solids. I. *Phys. Rev.* **1931**, *37*, 17–44.

29. Davydov, A. S. Excitons in thin crystals. *Soviet Phys.-Usp.* **1964**, *18*, 496–499.

30. Morrison, A. F.; You, Z.-Q.; Herbert, J. M. Ab Initio Implementation of the Frenkel-Davydov Exciton Model: A Naturally Parallelizable Approach to Computing Collective Excitations in Crystals and Aggregates. *J. Chem. Theory Comput.* **2014**, *10*, 5366–5376.

31. Zuo, L.; Humbert, M.; Esling, C. An effective algorithm for calculation of the Clebsch-Gordan coefficients. *J. Appl. Cryst.* **1993**, *26*, 302–304.

32. van Montfort, J. T. Photo-electron spectroscopy. General theoretical aspects and the calculation of peak positions and intensities in some simple systems. Ph.D. thesis, University of Groningen, 1980.

33. Broer, R.; Nieuwpoort, W. C. Broken orbital symmetry and the description of hole states in tetrahedral $[CrO_4]^{2-}$ anion. I. Introductory considerations and calculations on oxygen 1s hole states. *Chem. Phys.* **1981**, *54*, 291–303.

34. Broer, R.; Nieuwpoort, W. C. Broken orbital symmetry and the description of valence hole states in the tetrahedral $[CrO_4]^{2-}$ anion. *Theor. Chim. Acta* **1988**, *73*, 405–418.

35. Kathir, R. K.; de Graaf, C.; Broer, R.; Havenith, R. W. A. Reduced Common Molecular Orbital Basis for Nonorthogonal Configuration Interaction. *J. Chem. Theory Comput.* **2020**, *16*, 2941–2951.

36. Neese, F.; Wennmohs, F.; Becker, U.; Riplinger, C. The ORCA quantum chemistry program package. *J. Chem. Phys.* **2020**, *152*, 224108.

37. Huron, B.; Malrieu, J.-P.; Rancurel, P. Iterative perturbation calculations of ground and excited state energies from multiconfigurational zeroth-order wavefunctions. *J. Chem. Phys.* **1973**, *58*, 5745–5759.

38. Ivanic, J. Direct configuration interaction and multiconfigurational self-consistent-field method for multiple active spaces with variable occupations. I. Method. *J. Chem. Phys.* **2003**, *119*, 9364–9376.

39. Olsen, J.; Roos, B. O.; Jørgensen, P.; Jensen, H. J. A. Determinant based configuration interaction algorithms for complete and restricted configuration interactions apces. *J. Chem. Phys.* **1988**, *96*, 2185–2192.

40. Hermes, M. R.; Gagliardi, L. Multiconfigurational Self-Consistent Field Theory with Density Matrix Embedding: The Localized Active Space Self-Consistent Field Method. *J. Chem. Theory Comput.* **2019**, *15*, 972–986.

41. Li Manni, G.; Smart, S. D.; Alavi, A. Combining the Complete Active Space Self-Consistent Field Method and the Full Configuration Interaction Quantum Monte Carlo within a Super-CI Framework, with Application to Challenging Metal- Porphyrins. *J. Chem. Theory Comput.* **2016**, *12*, 1245–1258.

42. Weser, O.; Guther, K.; Ghanem, K.; Li Manni, G. Stochastic Generalized Active Space Self-Consistent Field: Theory and Application. *J. Chem. Theory Comput.* **2022**, *18*, 251–272.

43. Andersson, K.; Roos, B. O. Excitation energies in the nickel atom studied with the complete active space SCF method and second-order perturbation theory. *Chem. Phys. Lett.* **1992**, *191*, 507–514.

44. Angeli, C.; Cimiraglia, R.; Evangelisti, S.; Leininger, T.; Malrieu, J.-P. Introduction of n-electron valence states for multireference perturbation theory. *J. Chem. Phys.* **2001**, *114*, 10252–10264.

45. Pathak, S.; Lang, L.; Neese, F. A dynamic correlation dressed complete active space method: Theory, implementation, and preliminary applications. *J. Chem. Phys.* **2017**, *147*, 234109.

46. Farhan, M. A.; Abdelfattah, A.; Tomov, S.; Gates, M.; Sukkari, D.; Haidar, A.; Rosenberg, R.; Dongarra, J. MAGMA template for scalable linear algebra on emerging architectures. *International Journal of High Performance Computing Applications* **2021**, *34 (6)*, 645–658.

47. Gates, M.; Charara, A.; Kurzak, J.; YarKhan, A.; Al Farhan, M.; Sukkari, D.; Dongarra, J. *SLATE Users' Guide, SWAN No. 10*; 2020; revision 07-2020.

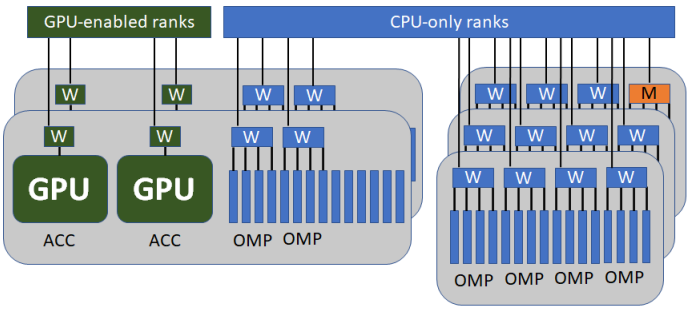
48. Galván, I. F. et al. OpenMolcas: From Source Code to Insight. *J. Chem. Theory Comput.* **2019**, *15*, 5925–5964.

49. Aquilante, F. et al. Modern quantum chemistry with [Open]Molcas. *J. Chem. Phys.* **2020**, *152*, 214117.

50. Fallon, K. J. et al. Exploiting Excited-State Aromaticity To Design Highly Stable Singlet Fission Materials. *J. Am. Chem. Soc.* **2019**, *141*, 13867–13876.

51. Ryerson, J. L. et al. Structure and photophysics of indigoids for singlet fission: Cibalackrot. *J. Chem. Phys.* **2019**, *151*, 184903.

52. Aguilar Suarez, L. E.; Menger, M. F. S. J.; Faraji, S. Singlet fission in tetracene: an excited state analysis. *Mol. Phys.* **2019**, *MQM2019*, 1–14.



For Table of Contents Only.

GronOR: Scalable and Accelerated Non-Orthogonal Configuration Interaction for Molecular Fragment Wave Functions

Supplementary information

T. P. Straatsma,[*,a,b], R. Broer,[c] A. Sánchez-Mansilla,[d] C. Sousa,[e] and C. de Graaf[c,d,f]

(a) National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6373, U. S. A.

(b) Department of Chemistry and Biochemistry, University of Alabama, Tuscaloosa, AL 35487-0336, U. S. A.

(c) Zernike Institute of Advanced Materials University of Groningen, Netherlands

(d) Departament de Química Física i Inorgànica Universitat Rovira i Virgili, Tarragona, Spain.

(e) Departament de Química Física and Institut de Química Teòrica i Computacional, Universitat de Barcelona, Spain

(f) ICREA, Pg. Lluís Companys 23, Barcelona, Spain

E-mail: str@ornl.gov

May 26, 2022

1 GNOME algorithm

The calculation of the interaction between non-orthogonal Slater determinants is based on the General Non-Orthogonal Matrix Elements (GNOME) algorithm developed by Broer, Nieuwpoort and van Montfort in the 1980s.[1–3] Here, the focus is on the two-electron part of the matrix elements, the simpler one-electron part is analogous and the expressions can be found in the original articles.

The basic expression for the calculation of the two-electron part of the interaction between two non-orthogonal Slater determinants reads

$$\langle\Phi_\alpha|\bar{g}_{12}|\Phi_\beta\rangle = \sum_{i<k}\sum_{j<l}\langle\phi_i\phi_j|\bar{g}_{12}|\psi_k\psi_l\rangle S(ik,jl) \tag{1}$$

with $S(ik,jl)$ the second-order co-factor of the overlap matrix of the orbitals. By applying a corresponding orbital transformation

$$\tilde{\phi}_i = \sum_j \phi_j U_{ji} \qquad \tilde{\psi}_i = \sum_j \psi_j V_{ji} \tag{2}$$

the overlap matrix becomes diagonal and the two-electron contribution to the Hamiltonian matrix element becomes

$$\langle\Phi_\alpha|\bar{g}_{12}|\Phi_\beta\rangle = \sum_{i<k}\langle\tilde{\phi}_i\tilde{\phi}_k|\bar{g}_{12}|\tilde{\psi}_i\tilde{\psi}_k\rangle \prod_{m\neq i,k}\lambda_m \tag{3}$$

with $\lambda_m = \langle\tilde{\phi}_m|\tilde{\psi}_m\rangle$. Substituting $\tilde{\phi}$ and $\tilde{\psi}$ by their expansion in basis functions χ (either AO basis or the common MO basis, see below) leads to

$$\langle\Phi_\alpha|\bar{g}_{12}|\Phi_\beta\rangle = \sum_{\mu<\nu}\sum_{\rho<\sigma}\langle\chi_\mu\chi_\rho|\bar{g}_{12}|\chi_\nu\chi_\sigma\rangle B(\mu\rho,\nu\sigma) \tag{4}$$

with

$$B(\mu\rho,\nu\sigma) = \frac{1}{2}(1-\hat{P}_{\mu\rho})(1-\hat{P}_{\nu\sigma})\sum_{k,i}\tilde{c}_{i\mu}\tilde{c}_{k\rho}\tilde{d}_{\nu i}\tilde{d}_{\sigma k}\prod_{m\neq i,k}\lambda_m \tag{5}$$

the transformed second-order co-factor. This super-matrix with approximately $N^4/8$ elements can be written as the product of two $N\times N$ matrices $F(\omega)$ and $G(\omega)$

$$B(\mu\rho,\nu\sigma) = (1-\hat{P}_{\mu\rho})(1-\hat{P}_{\nu\sigma})F(\omega)_{\mu\nu}G(\omega)_{\rho\sigma} \tag{6}$$

This is the factorized transformed second-order co-factor and forms the basis of the GNOME algorithm. F and G depend on ω, the number of zeros in the overlap matrix of the corresponding orbitals.

$$F(0)_{\mu\nu} = \frac{1}{2}\sum_i \tilde{c}_{i\mu}\tilde{d}_{\nu i}\lambda_i^{-1} \qquad G(0)_{\mu\nu} = 2F(0)_{\mu\nu}\prod_i \lambda_i \tag{7}$$

$$F(1)_{\mu\nu} = \sum_{i \neq m} \tilde{c}_{i\mu} \tilde{d}_{\nu i} \lambda_i^{-1} \qquad G(1)_{\mu\nu} = \tilde{c}_{m\mu} \tilde{d}_{\nu m} \prod_{i \neq m} \lambda_i \qquad (\lambda_m = 0) \tag{8}$$

$$F(2)_{\mu\nu} = \tilde{c}_{n\mu} \tilde{d}_{q\nu} \qquad G(2)_{\mu\nu} = \tilde{c}_{m\mu} \tilde{d}_{\nu m} \prod_{i \neq m,n} \lambda_i \qquad (\lambda_{m,n} = 0) \tag{9}$$

For $\omega > 2$, both F and G are zero.

2 Common MO basis

The construction of the common molecular orbital basis to express the non-orthogonal states is described in detail in Ref. 4. A short version is given here for completeness.

To facilitate the calculation of the matrix elements among the non-orthogonal Slater determinants, the one- and two-electron integrals need to be expressed in a common basis set. The standard atomic orbital (AO) basis functions provide such a common basis, but have the disadvantage of being very large and severely limiting the applicability of the NOCI approach as implemented in GronOR. The first step towards a more compact basis set of molecular orbitals is to collect the doubly occupied and active orbitals of all the (non-orthogonal) electronic states of a fragment

$$\Upsilon = \{\phi_1, \phi_2, \ldots, \phi_k, \psi_1, \psi_2, \ldots, \psi_l, \omega_1, \omega_2, \ldots, \omega_m, \ldots\} \tag{10}$$

where ϕ_i, ψ_i, ω_i, ... are the optimal orbitals used to describe the diabatic fragment states Φ, Ψ, Ω, ... Although Υ defines a complete basis to express the fragment states, it is a non-orthogonal and, more importantly, a strongly linear dependent basis set.

To construct a more compact common MO basis, the $n \times n$ overlap matrix of the orbitals in Υ is calculated, where $n = k + l + m + \ldots$, the sum of all the inactive and active orbitals.

$$S_{ij} = \langle \psi_i | \psi_j \rangle \tag{11}$$

The eigenvectors $\mathbf{U}$ of this overlap matrix also constitute a complete basis

$$S_d = U^\dagger S U, \tag{12}$$

with the difference that the linear dependencies can be removed from the basis by considering only those eigenvectors whose eigenvalue λ_α is larger than the user-defined threshold τ_{MO}.

$$U \xrightarrow{\lambda_\alpha > \tau_{MO}} V \tag{13}$$

The reduced basis V is expressed in the AO basis to obtain the final common MO basis set

$$W = C(S'_d)^{-1/2}V \tag{14}$$

where C is the coefficient matrix of the orbitals in Υ.

This process is repeated for all fragments and the resulting basis sets are accumulated. Once all fragments are processed, the integrals are expressed in the new common MO basis.

References

[1] R. Broer and W. C. Nieuwpoort. Broken orbital symmetry and the description of hole states in tetrahedral $[CrO_4]^{2-}$ anion. I. Introductory considerations and calculations on oxygen 1s hole states. *Chem. Phys.*, 54:291–303, 1981.

[2] R. Broer and W. C. Nieuwpoort. Broken orbital symmetry and the description of valence hole states in the tetrahedral $[CrO_4]^{2-}$ anion. *Theor. Chim. Acta*, 73:405–418, 1988.

[3] J. T. van Montfort. *Photo-electron spectroscopy. General theoretical aspects and the calculation of peak positions and intensities in some simple systems.* PhD thesis, University of Groningen, 1980.

[4] R. K. Kathir, C. de Graaf, R. Broer, and R. W. A. Havenith. Reduced common molecular orbital basis for nonorthogonal configuration interaction. *J. Chem. Theory Comput.*, 16:2941–2951, 2020.