ARMY RESEARCH LABORATORY

Trace Conserving Purification for Linear Scaling [O(N)] Methods: A First Enhancement to CP2K

by Jonathan Mullin

ARL-CR-0746 September 2014

prepared by

Jonathan Mullin

under contract

W911NF-12-2-0019

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5069

ARL-CR-0746 **September 2014**

Trace Conserving Purification for Linear Scaling [O(N)] Methods: A First Enhancement to CP2K

Jonathan Mullin
Weapons and and Materials Research, ARL

prepared by

Jonathan Mullin

under contract

W911NF-12-2-0019

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			
1. REPORT DATE *(DD-MM-YYYY)* September 2014	**2. REPORT TYPE** Final		**3. DATES COVERED (From - To)** August–November 2013
4. TITLE AND SUBTITLE Trace Conserving Purification for Linear Scaling [O(n)] Methods: A First Enhancement to CP2K			**5a. CONTRACT NUMBER**
			5b. GRANT NUMBER
			5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Jonathan Mullin			**5d. PROJECT NUMBER**
			5e. TASK NUMBER
			5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WML-B Aberdeen Proving Ground, MD 21005-5069			**10. SPONSOR/MONITOR'S ACRONYM(S)** ARL-CR-0746
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			
13. SUPPLEMENTARY NOTES			
14. ABSTRACT Implementation of the low-order purification method as an alternative to diagonalization for self-consistent density function theory calculations lowers the memory and time to solution for analysis of chemical systems. There is some improvement for low-sparsity systems, but the highest speed-up is seen in low band gap and highly sparse matrix problems. This work has been conducted and tested in the CP2K program.			
15. SUBJECT TERMS purification, CP2K, nonmonotonic, trace conserving, linear scaling, DFT, O(n)			

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Jonathan Mullin
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** (*Include area code*)
Unclassified	Unclassified	Unclassified	UU	20	410-306-0765

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

This work was supported by the US Army Research Laboratory Enterprise for Multiscale Research of Materials. Dr. Mullin's contract is coordinated through the Oak Ridge Associated Universities Maryland office. Timings were performed on the CRAY XE6 (Garnet) at the Engineer Research and Development Center of the US Department of Defense Supercomputing Resource Center. Dr. Mullin would like to thank the Gary Kedziora of the US Department of Defense User Productivity Enhancement, Technology Transfer, and Training program for helpful discussions.

INTENTIONALLY LEFT BLANK.

1. Introduction

A quantum mechanical (QM) approach to materials science provides a gold standard atomistic picture of the mechanisms responsible for a range of phenomena seen in macroscopic and experimental situations. The need to understand materials science problems from atomistic to macroscale was the impetus for the US Army Research Laboratory to initiate the Enterprise for Multiscale Material Research. This long-term project attempts to redefine how materials science questions are posed and solved. To support this goal, current state-of-the-art QM capabilities need to be extended in the number of atoms that can be treated and the length scale of the dynamics that can be simulated. This extension is referred to as large-scale QM—large both spatially and temporally. This will enable fundamental advances in the understanding of materials science problems.

This is the first in a series of reports that will be required to realize the new capabilities needed to model realistic materials science problems. There are several areas of QM theory that may be able to address these material properties, which present unique strengths. We hope to exploit these advantages to gain new insight into the basic physical processes governing macroscopic material properties from atomistic simulations. The first focus will be on expanding the linear scaling methods currently implemented within the atomistic simulation package CP2K.[1] Future work may expand on this work in linear scaling or adding functionality to the basic linear scaling approach. Other methods to achieve linear scaling involve orbital transform free methods, fragmentation methods, and new molecular dynamics ensembles. These may also be investigated as they present advantages outside the current focus.

In the past decade, CP2K has become a standard tool of the computational chemistry community for efficient QM molecular dynamics. One area where CP2K has been a leader in the field is in linear scaling methods. CP2K mixes plane waves and Gaussian descriptions of atoms and their interactions. This explicitly lends itself well to linear scaling methods based on sparse matrix density functional methods. One approach is to use purification of the Kohn-Sham matrix. This is currently implemented in CP2K, though there have been recent developments in this area that could provide significant advantages over the current state of the art.[2,3]

Purification is analogous to preconditioning in linear algebra, but the transformation is intended to resolve the Kohn-Sham matrix such that it becomes idempotent. This relies on predefined polynomials, which must balance the order of the polynomial with the number of iterations needed to create an idempotent matrix. A higher-order polynomial requires more matrix-matrix multiplications but may allow for fewer iterations. It has been shown that relatively low-order polynomials perform well.[4] CP2K currently employs a fourth-order polynomial, which monotonically transforms the matrix to idempotency and is known as the fourth-order trace resetting (TRS4) method.[5]

A recently published method provides a framework for using a chosen polynomial but with nonmonotonic constraints on the polynomial[2] that allow treatment of a low band gap system. The nonmonotonic approach we chose to implement in CP2K was based on the second-order trace conserving (TC2) method,[6] which allows for weighting of the polynomials based on the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO) energies. This allows the polynomials to adapt as the matrix approaches idempotentcy. The method implemented is a second-order polynomial reducing the amount of memory required and the matrix-matrix multiplications. This allows for a larger number of atoms to be treated than currently available methods in CP2K. The nonmonotonic nature allows the purification to perform as well as the higher-order methods and even outperform the current state of the art at low band gaps.

The nonmonotonic TC2 method therefore allows for a lower memory footprint for the calculation. The memory savings come from having to save each multiplication intermediate. Generally, for a given order of polynomial (N), N-1 intermediates must be saved. The benefit of this can be seen in the largest calculation capable of being run. A single molecular dynamics step on 100K atoms can be treated with TC2 but not using the current TRS4 method implemented in CP2K. Another advantage of nonmonotonic TC2 is the ability to adapt the purification to the band gap. This is important as the difference between the HOMO and LUMO orbitals becomes very small. As the band gap becomes small, the purification must differentiate how to spread out the values of matrix, which becomes difficult for an arbitrarily chosen polynomial. The nonmonotonic approach allows the flexibility in the transformation to resolve the small eigenvalues.

CP2K is a freely available program, combining several other open source libraries to form a complete simulation package. Our implementation of the TC2 method in CP2K is most easily understood by looking at the following pseudo code. A full subroutine is included in the appendix. However, it uses many conventions and variables specific to CP2K and is therefore hard to understand without context.

PSEUDO CODE

```
Initiate from self consistent field convergence

INPUTS: Kohn Sham Matrix (KS matrix), # occupied orbitals (nelectron),
value of highest occupied molecular orbital (homo) , value of lowest
unoccupied molecular orbital (lumo)

Initialize: X = (eps_n*I - H)/(eps_n - eps_0)  ... H* = S^-1/2*H*S^-1/2

Get largest/smallest eigenvalues for scaling

Scale KS matrix

Get X*X
Set alpha, beta and betaB = 1
```

```
    DO I=1,Max_iter
      IF(do_non_monotonic) THEN
         IF ( trace_fx  > nelectron ) THEN
           alpha=2/(2-beta)
           Xn+1 = (aX+ (1-a)I)^2
           beta =(alpha*beta +1-alpha)**2
           betaB=(alpha*betaB+1-alpha)**2
         ELSE
           alpha=2/(1+betaB)
           Xn+1 = 2aX-a^2*X^2
           beta =2*alpha*beta -(alpha*beta )**2
           betaB=2*alpha*betaB-(alpha*betaB)**2
      ELSE
         IF (trace_fx > nelectron) THEN
           Xn+1 = X*X
         ELSE
           Xn+1 = 2X-X*X
         ENDIF
      ENDIF
   Evaluate error associated with purification step. If below
threshold, exit loop
   ENDDO

Compute homo/lumo

Export X to an updated density

Return to self consistent field convergence
```

In this report, the nonmonotonic TC2 method of purification is compared with the TRS4 method already in CP2K. A series of calculations were run to gauge how effective the TC2 is for low and high band gap systems. The TC2 method lowers the memory requirements compared with TRS4, which allows larger systems to be simulated, as the current limiting factor on a commodity supercomputer tends to be the amount of memory per core. The nonmonotonic nature of the TC2 allows for faster convergence over traditional TC2 methods and is competitive with the higher order polynomial method TRS4. Further, the nonmonotonic TC2 outperforms TRS4 for low band gap systems where the purification can be adjusted based on the HOMO-LUMO gap in nonmonotonic TC2, leading to quicker idempotency of the Kohn-Sham matrix.

2. Results and Conclusions

Figure 1 shows a comparison of TC2 and TRS4 in which the time for each band gap is normalized to TRS4. Two systems were used to span the range of band gaps shown with real chemical examples rather than artificial matrix inputs. These are shown in Fig. 2. The 0.X band gap was a 1024 water case. This water box was constructed by an NVT (number, volume, temperature) simulation of TIP4P waters at 300 K, part of the test cases for CP2K. Subsequent calculations were performed for extensions of an alpha helix of alanine molecules. This was chosen because the band gap is lowered as the alpha helix is extended. This is an effect of the induced dipoles on the alanine aligning in the turns of the alpha helix. Therefore, by adding more repeating units, we can simulate a particular band gap. The chart represents a 20, 40, 80 alanine repeat. These timings were run on a CRAY XE6 on 512 processors. These represent the average of two runs for each system and may be influenced by normal variations in timings due to communication bottlenecks. Another test probed how large of a system was feasible. A test of 104K atoms was constructed by creating a supercell from multiples of the 1024 water box. Both TRS4 and TC2 were run on 2048 processors. The TRS4 method failed to run from an out-of-memory error, while the TC2 was able to complete an energy evaluation with a double-zeta valence polarized basis set. This stems from the fact that the TC2 method is able to use two fewer copies of the Kohn-Sham matrix to perform the purification, making a significant difference at high atom counts. It may also explain some of the timing differences.

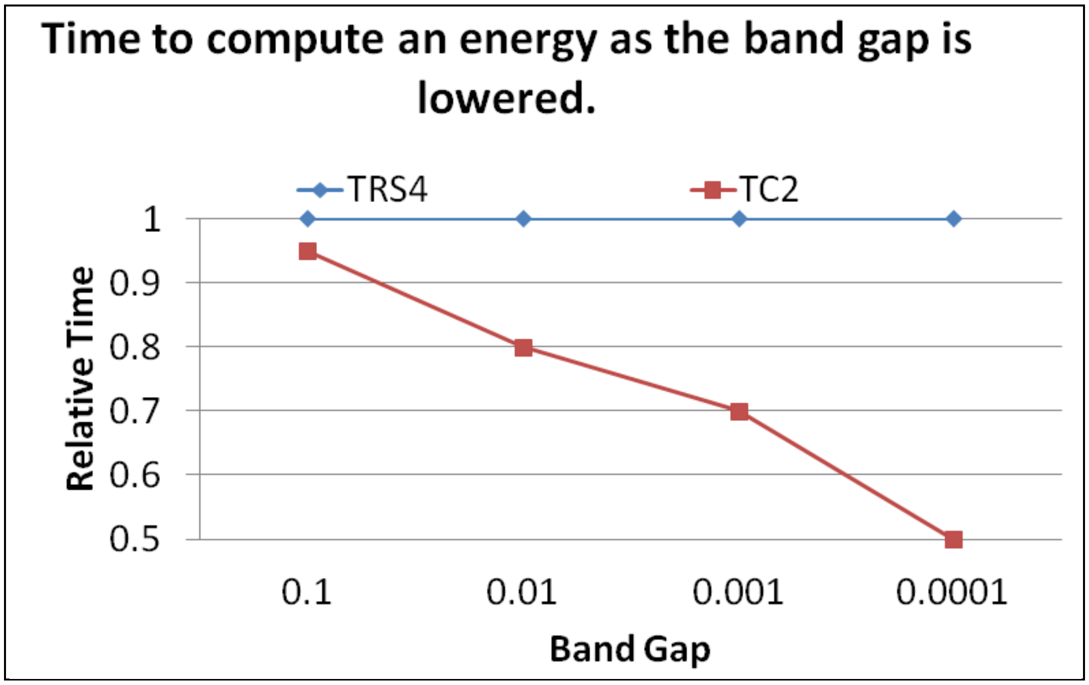

Fig. 1 Comparison of TRS4 and TC2 purification scheme times in CP2K. Timings are normalized to TRS4 for each band gap.

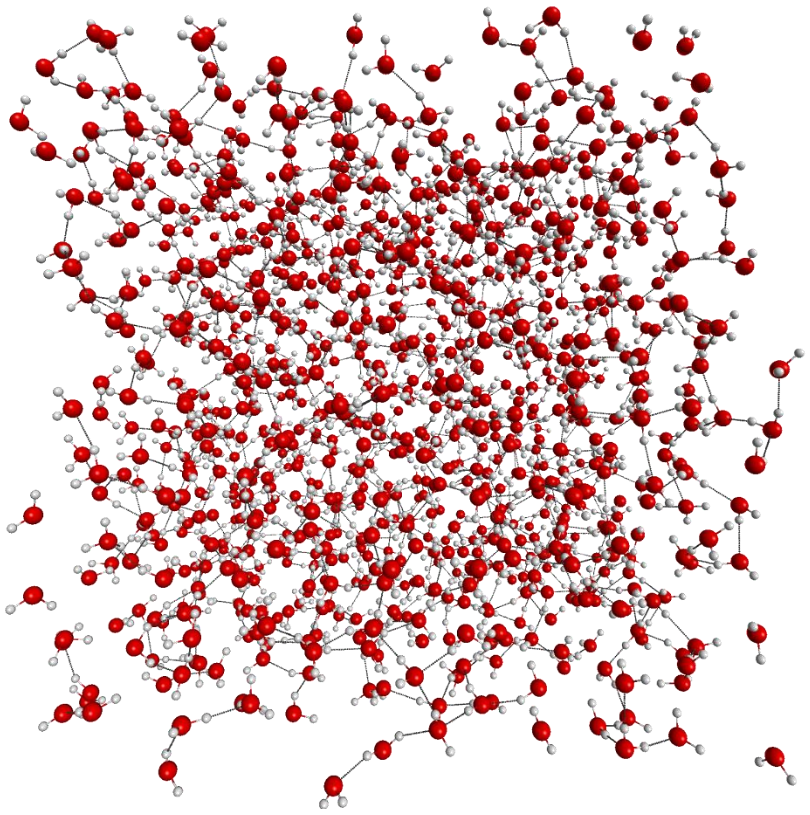
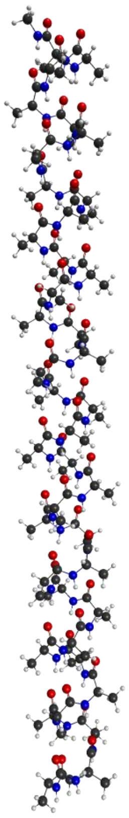

Fig. 2 Graphical representation of the 1024 water box (left) and a 40-subunit repeat of alanine in an alpha helix (right)

The TC2 method has been shown to provide significant advancements to the state of art in CP2K. There are some issues with premature identification of an idempotent matrix for the water case resulting in a 4-kcal/mol error for any size of water cluster. By lowering the eps_threshold (a numerical threshold for when to drop elements from the sparse matrix representation) slightly compared with TRS4, the TC2 method reproduces the TRS4 result. The deviation was consistent over an eps_threshold range of 1E-5 to 1E-7. This deviation of accuracy between the methods was not seen on diamond, alpha helixes, or graphene. It may be that the small contributions of the hydrogen bonding network are not being preserved as well as in the nonmonotonic approach. Current error estimation is designed for speed over accuracy, and this may not be an acceptable trade-off for the nonmonotonic polynomials. Future work might explore possible improvements such a using a Euclidean norm rather than a Frobenius norm in error estimation to alleviate premature idempotency identification for the nonmonotonic method.

3. References

1. Hutter J, Iannuzzi M, Schiffmann F, VandeVondele J. CP2K: atomistic simulations of condensed matter systems. Wiley Interdisciplinary Reviews: Computational Molecular Science. 2014;4(1):15.
2. Rubensson EHJ. Nonmonotonic recursive polynomial expansions for linear scaling calculation of the density matrix. Chem. Theory Comput. 2011:1233.
3. Suryanarayana P. Optimized purification for density matrix calculation. Chem. Phys. Lett. 2013;555:291.
4. Rubensson EH. Comment on “On the optimal symmetric purification scheme of the one-particle density matrix” [Chem. Phys. Lett. 2011;511:159–160]. Chem. Phys. Lett. 2012;527:227.
5. Niklasson AMN, Tymczak CJ, Challacombe M. Trace resetting density matrix purification in O(N) self-consistent-field theory. J. Chem Phys. 2003;118:8611.
6. Niklasson AMN. Expansion algorithm for the density matrix. Phys. Rev. B. 2002;66:155115.

Appendix. The Subroutine and Function Added to CP2K

This appendix appears in its original form, without editorial change.

Insignificant changes were made to input parsing routines to allow execution of these subroutines.

```
    ! ***************************************************************************
    !> \brief compute the density matrix using a non monotonic trace conserving algorithm
    !> \par History
    !>       2013.08 created [Jonathan Mullin]
    !> \author Jonathan Mullin
    ! ***************************************************************************
      SUBROUTINE density_matrix_tc2(matrix_p, matrix_ks, matrix_s_sqrt_inv, &
                                nelectron, threshold, e_homo, e_lumo, e_mu, &
                                dynamic_threshold,non_monotonic,&
                                matrix_ks_deviation,max_iter_lanczos,&
                                eps_lanczos, error)

        TYPE(cp_dbcsr_type), INTENT(INOUT)       :: matrix_p
        TYPE(cp_dbcsr_type), INTENT(IN)          :: matrix_ks, matrix_s_sqrt_inv
        INTEGER, INTENT(IN)                      :: nelectron
        REAL(KIND=dp), INTENT(IN)                :: threshold
        REAL(KIND=dp), INTENT(INOUT)             :: e_homo, e_lumo, e_mu
        LOGICAL, INTENT(IN), OPTIONAL            :: dynamic_threshold
        LOGICAL, INTENT(IN), OPTIONAL            :: non_monotonic
        TYPE(cp_dbcsr_type), INTENT(INOUT)       :: matrix_ks_deviation
        INTEGER, INTENT(IN)                      :: max_iter_lanczos
        REAL(KIND=dp), INTENT(IN)                :: eps_lanczos
        TYPE(cp_error_type), INTENT(inout)       :: error

        CHARACTER(LEN=*), PARAMETER :: routineN = 'density_matrix_tc2', &
          routineP = moduleN//':'//routineN
        INTEGER, PARAMETER                       :: max_iter = 100

        INTEGER                                  :: estimated_steps, nne, &
                                                    handle, i, j, unit_nr
        INTEGER(kind=int_8)                      :: flop1, flop2
        LOGICAL                                  :: converged, do_dyn_threshold, &
                                                   do_non_monotonic
        REAL(KIND=dp) :: current_threshold, eps_max, eps_min, est_threshold, &
          frob_id, frob_x, homo, lumo, max_eig, max_threshold, maxdev, &
          maxev, min_eig, minev, mmin, mu, mu_a, mu_b, mu_c, mu_fa, mu_fc, &
          occ_matrix, scaled_homo_bound, scaled_lumo_bound, t1, t2, trace_fx, &
           xi, alpha, beta, betaB
        TYPE(cp_dbcsr_type)                      :: matrix_k0, matrix_x, &
                                                    matrix_xidsq, matrix_xsq
        TYPE(cp_logger_type), POINTER            :: logger

        CALL timeset(routineN,handle)

        logger => cp_error_get_logger(error)
        IF (logger%para_env%mepos==logger%para_env%source) THEN
           unit_nr=cp_logger_get_default_unit_nr(logger,local=.TRUE.)
        ELSE
           unit_nr=-1
        ENDIF

        do_dyn_threshold = .FALSE.
        do_non_monotonic = .TRUE.
        IF(PRESENT(dynamic_threshold)) do_dyn_threshold = dynamic_threshold
        IF(PRESENT(non_monotonic)) do_non_monotonic = non_monotonic

        ! init X = (eps_n*I - H)/(eps_n - eps_0)  ... H* = S^-1/2*H*S^-1/2
        CALL cp_dbcsr_init(matrix_x,error=error)
        CALL cp_dbcsr_create(matrix_x, template=matrix_ks, matrix_type=dbcsr_type_no_symmetry,
error=error)

        CALL cp_dbcsr_multiply("N", "N", 1.0_dp, matrix_s_sqrt_inv, matrix_ks,&
                               0.0_dp, matrix_x, filter_eps=threshold,error=error)
        CALL cp_dbcsr_multiply("N", "N", 1.0_dp, matrix_x, matrix_s_sqrt_inv, &
                              0.0_dp, matrix_x, filter_eps=threshold,error=error)
        CALL cp_dbcsr_init(matrix_k0,error=error)
        CALL
cp_dbcsr_create(matrix_k0,template=matrix_ks,matrix_type=dbcsr_type_no_symmetry,error=error)
        CALL cp_dbcsr_copy(matrix_k0, matrix_x, error=error)

        ! compute the deviation in the mixed matrix, as seen in the ortho basis
        CALL cp_dbcsr_add(matrix_ks_deviation, matrix_x , -1.0_dp, 1.0_dp, error=error)
        CALL lanczos_alg_serial(matrix_ks_deviation, maxev, minev, max_iter=max_iter_lanczos,
threshold=eps_lanczos, &
                                converged=converged, error=error)
        maxdev = MAX(ABS(maxev), ABS(minev))
        IF (unit_nr>0) THEN
           WRITE(unit_nr, '(T6,A,1X,L12)')   "Lanczos converged:     ", converged
           WRITE(unit_nr, '(T6,A,1X,F12.5)') "change in mixed matrix: ", maxdev
           WRITE(unit_nr, '(T6,A,1X,F12.5)') "HOMO upper bound:      ", e_homo+maxdev
           WRITE(unit_nr, '(T6,A,1X,F12.5)') "LUMO lower bound:      ", e_lumo-maxdev
           WRITE(unit_nr, '(T6,A,1X,L12)')   "Predicts a gap ?        ", ((e_lumo-maxdev)-
(e_homo+maxdev))>0
        ENDIF
        ! save the old mixed matrix
        CALL cp_dbcsr_copy(matrix_ks_deviation, matrix_x , error=error)

        ! get largest/smallest eigenvalues for scaling
        CALL lanczos_alg_serial(matrix_x, max_eig, min_eig, max_iter=max_iter_lanczos,
threshold=eps_lanczos,&
                                converged=converged, error=error)
        IF (unit_nr>0) WRITE(unit_nr,'(T6,A,1X,2F12.5,1X,A,1X,L1)') "Est. extremal eigenvalues",
&  min_eig, max_eig," converged: ",converged

        eps_max = max_eig
        eps_min = min_eig

        ! scale KS matrix
        CALL cp_dbcsr_scale(matrix_x, -1.0_dp, error=error)
        CALL cp_dbcsr_add_on_diag(matrix_x, eps_max, error=error)
        CALL cp_dbcsr_scale(matrix_x, 1/(eps_max-eps_min), error=error)

        ! scale bounds for HOMO/LUMO
        scaled_homo_bound = (eps_max-(e_homo+maxdev))/(eps_max-eps_min)
        scaled_lumo_bound = (eps_max-(e_lumo-maxdev))/(eps_max-eps_min)
        current_threshold = threshold



        CALL cp_dbcsr_init(matrix_xsq,error=error)
        CALL
cp_dbcsr_create(matrix_xsq,template=matrix_ks,matrix_type=dbcsr_type_no_symmetry,error=error)

    cp_dbcsr_create(matrix_xidsq,template=matrix_ks,matrix_type=dbcsr_type_no_symmetry,error=erro
r)


        beta=e_lumo
        betaB=e_homo
        alpha=1.0_dp

        DO i=1,max_iter
          t1 = m_walltime()
          flop1 = 0; flop2 = 0

          ! get X*X
          CALL cp_dbcsr_multiply("N", "N", 1.0_dp, matrix_x, matrix_x,&
                                 0.0_dp, matrix_xsq, &
                                 filter_eps=current_threshold,flop=flop1,error=error)

          ! intermediate use matrix_p to compute = X*X-X
          CALL cp_dbcsr_copy(matrix_p, matrix_x,error=error)
          CALL cp_dbcsr_add(matrix_p, matrix_xsq, -1.0_dp, 1.0_dp, error=error)
          frob_id = cp_dbcsr_frobenius_norm(matrix_p)
          frob_x = cp_dbcsr_frobenius_norm(matrix_x)
         CALL cp_dbcsr_trace(matrix_x, trace_fx, error=error)
          ! quantities used for dynamic thresholding, when the estimated gap is larger than zero
          xi = (scaled_homo_bound-scaled_lumo_bound)
          IF (do_dyn_threshold .AND. xi > 0.0_dp) THEN
            mmin = 0.5*(scaled_homo_bound+scaled_lumo_bound)
            max_threshold = ABS(1-2*mmin)*xi

            scaled_homo_bound = evaluate_tc2_polynomial(scaled_homo_bound, trace_fx, nelectron)
            scaled_lumo_bound = evaluate_tc2_polynomial(scaled_lumo_bound, trace_fx, nelectron)
            estimated_steps = estimate_steps(scaled_homo_bound, scaled_lumo_bound, threshold)

            est_threshold =
(threshold/(estimated_steps+i+1))*xi/(1+threshold/(estimated_steps+i+1))
            est_threshold = MIN(max_threshold, est_threshold)
            IF (i > 1) est_threshold = MAX(est_threshold, 0.1_dp * current_threshold)
            current_threshold = est_threshold
          ELSE
            current_threshold = threshold
          ENDIF


         IF(do_non_monotonic) THEN
          IF ( trace_fx  > nelectron ) THEN
    ! Xn+1 = (aX+ (1-a)I)^2
            alpha=2/(2-beta)

            CALL cp_dbcsr_scale(matrix_xsq, alpha**2, error=error)
            CALL cp_dbcsr_add_on_diag(matrix_xsq, 1.0-alpha**2, error=error)
            CALL cp_dbcsr_copy(matrix_x, matrix_xsq, error=error)
            CALL cp_dbcsr_filter(matrix_x, current_threshold, error=error)

            beta =(alpha*beta +1-alpha)**2
            betaB=(alpha*betaB+1-alpha)**2


          ELSE
    ! Xn+1 = 2aX-a^2*X^2
            alpha=2/(1+betaB)

            CALL cp_dbcsr_add(matrix_x, matrix_xsq, 2.0_dp*alpha, -1.0_dp*alpha**2, error=error)
            CALL cp_dbcsr_filter(matrix_x, current_threshold, error=error)

            beta =2*alpha*beta -(alpha*beta )**2
            betaB=2*alpha*betaB-(alpha*betaB)**2

          ENDIF


         ELSE
    !
          IF (trace_fx > nelectron) THEN
            ! Xn+1 = X*X
            CALL cp_dbcsr_copy(matrix_x, matrix_xsq, error=error)
          ELSE
            ! Xn+1 = 2X-X*X
            CALL cp_dbcsr_add(matrix_x, matrix_xsq, 2.0_dp, -1.0_dp, error=error)
            CALL cp_dbcsr_filter(matrix_x, current_threshold, error=error)

          ENDIF
         ENDIF
          occ_matrix = cp_dbcsr_get_occupation(matrix_x)
          t2 = m_walltime()
          IF (unit_nr>0) THEN
               WRITE(unit_nr,&
                          '(T6,A,I3,1X,F10.8,E12.3,F12.3,F13.3,E12.3)') "TC2 it ", &
                          i, occ_matrix,  t2-t1,&
                          (flop1+flop2)/(1.0E6_dp*(t2-t1)), current_threshold
             CALL m_flush(unit_nr)
          ENDIF
```

```
          IF (unit_nr>0) WRITE(unit_nr,'(T6,A,5E12.3)')
"Err",frob_id,frob_x,threshold,ABS(frob_id/frob_x),SQRT(threshold)
          IF (ABS(frob_id/frob_x) < SQRT(threshold) ) EXIT

        END DO

        occ_matrix = cp_dbcsr_get_occupation(matrix_x)
        IF (unit_nr>0) WRITE(unit_nr, '(T6,A,I3,1X,F10.8)') 'Final TC2 iteration  ', i,
occ_matrix

        ! free some memory
        CALL cp_dbcsr_release(matrix_xsq, error=error)
    !    CALL cp_dbcsr_release(matrix_xidsq, error=error)

        ! output to matrix_p, P = inv(S)^0.5 X inv(S)^0.5
        CALL cp_dbcsr_multiply("N", "N", 1.0_dp, matrix_x, matrix_s_sqrt_inv,&
                               0.0_dp, matrix_p, filter_eps=threshold,error=error)
        CALL cp_dbcsr_multiply("N", "N", 1.0_dp, matrix_s_sqrt_inv, matrix_p,&
                               0.0_dp, matrix_p, filter_eps=threshold,error=error)

    compute_homo_lumo(matrix_k0,matrix_x,eps_min,eps_max,threshold,max_iter_lanczos,eps_lanczos,h
omo,lumo,unit_nr,error)

        e_homo = homo
        e_lumo = lumo

        beta=lumo
        betaB=homo

        CALL cp_dbcsr_release(matrix_x, error=error)
        CALL cp_dbcsr_release(matrix_k0, error=error)
        CALL timestop(handle)

      END SUBROUTINE density_matrix_tc2

      FUNCTION evaluate_tc2_polynomial(x, trace_fx,nelectron) RESULT (xr)
        REAL(KIND=dp)                           :: x,trace_fx
        INTEGER                                 :: i,nelectron
        REAL(KIND=dp)                           :: xr

        INTEGER                                 :: k

        i=1
        xr = x
        DO k=1,i
          IF (trace_fx < nelectron) THEN
            xr = 2*xr-xr**2
          ELSE
           xr = xr**2
          ENDIF
        ENDDO
      END FUNCTION evaluate_tc2_polynomial
```

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOV PRINTING OFC A MACHOTRA
1 (PDF)	DIR USARL RDRL WML B J MULLIN