Convergence Analysis of Adaptive DIIS Algorithms with Application to Electronic Ground State Calculations

M. Talha Aziz

October 23, 2024

Outline

Self-Consistent Field (SCF) Method

- The Self-Consistent Field (SCF) method is an iterative process used to solve the electronic Schrödinger equation, crucial for Hartree-Fock methods.
- RHF (Restricted Hartree-Fock) applies SCF to calculate the total energy and molecular orbital coefficients for closed-shell systems.
- DIIS accelerates SCF iterations by ensuring faster convergence, which is crucial given the computational demands of electronic structure calculations.

RHF Workflow

- The core Hamiltonian H consists of kinetic energy matrix T and nuclear attraction matrix V:

$$H = T + V$$

- Initial density matrix D_0 is usually guessed or set to zero.
  - The Fock matrix is constructed as:

$$F_{ij} = H_{ij} + \sum_{kl} D_{kl}(2g_{ijkl} - g_{ikjl})$$

  - D_{kl}: density matrix elements; g_{ijkl}: two-electron integrals.

Solve the Roothan-Hall Equations

- Solve the Generalized Eigenvalue Problem (GEVP):

$$FC = SC\epsilon$$

  C: Molecular orbital coefficients; ϵ: Orbital energies.

- The density matrix D is updated as:

$$D_{ij} = 2 \sum_{a \in \text{occ}} C_{ia} C_{ja}^{\dagger}$$

  The factor of 2 accounts for closed-shell systems.

- The total electronic energy is:

$$E_{\text{elec}} = \frac{1}{2} \sum_{ij} D_{ij}(H_{ij} + F_{ij})$$

- Total energy also includes nuclear repulsion energy:

$$E_{\text{total}} = E_{\text{elec}} + E_{\text{nuc}}$$

Step 6: Check for Convergence

- Check convergence by:

$$\Delta E = |E_{\text{new}} - E_{\text{old}}| < \text{tol}$$

- If convergence is met, the SCF process ends, otherwise update D and repeat.

Introduction to Adaptive DIIS

- Anderson–Pulay acceleration methods include DIIS (Direct Inversion in the Iterative Subspace) and Anderson acceleration.
- Two adaptive variants of the Anderson–Pulay methods are proposed in this paper.
  - Restarted Anderson–Pulay acceleration.
  - Adaptive-depth Anderson–Pulay acceleration.
- These methods improve convergence by dynamically adjusting parameters during the iteration process, leading to faster and more stable results compared to the traditional fixed-depth DIIS.

Overview of the DIIS Algorithm

- DIIS (Direct Inversion in the Iterative Subspace) is a method developed by Peter Pulay in 1980 to accelerate the convergence of Self-Consistent Field (SCF) iterations in quantum chemistry.
- Fixed point Iteration:
  - We want to find a fixed-point x^* of a nonlinear function g from $\mathbb{R}^n$ to $\mathbb{R}^n$.
  - The fixed-point iterative scheme is:

$$x^{(k+1)} = g(x^{(k)})$$

  - At each step, an error vector $r^{(k)}$ is computed as:

$$r^{(k)} = f(x^{(k)})$$

  - Here, $f : \mathbb{R}^n \to \mathbb{R}^p$, where p may be different from n, and $f(x^*) = 0$.

DIIS Approximation

- DIIS accelerates convergence by assuming that a good approximation to x^* the fixed-point, can be obtained by combining previous iterates.
- At step $k+1$, form a combination of the previous iterates:

$$x^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} g(x^{(k-m_k+i)})$$

  $m_k = \min\{m, k\}$, where m is the maximal depth.
- The coefficients $c_i^{(k)}$ must minimize the norm of the linearized error vector:

$$\sum_{i=0}^{m_k} c_i^{(k)} r^{(k-m_k+i)}$$

- The minimization is subject to the constraint:

$$\sum_{i=0}^{m_k} c_i^{(k)} = 1$$

Lagrangian Formulation and Solving Normal Equations

- The Lagrangian function for minimizing the error is:

$$L(c_0, \ldots, c_{m_k}, \lambda) = \frac{1}{2} \left\| \sum_{i=0}^{m_k} c_i r^{(k-m_k+i)} \right\|^2 - \lambda \left(\sum_{i=0}^{m_k} c_i - 1 \right)$$

- This leads to the normal equations, which can be solved to find the coefficients c_i.
- In matrix form, the system becomes:

$$\begin{pmatrix} b_{00}^{(k)} & \cdots & b_{0m_k}^{(k)} & -1 \\ \vdots & \ddots & \vdots & \vdots \\ b_{m_k0}^{(k)} & \cdots & b_{m_km_k}^{(k)} & -1 \\ -1 & \cdots & -1 & 0 \end{pmatrix} \begin{pmatrix} c_0^{(k)} \\ \vdots \\ c_{m_k}^{(k)} \\ \lambda^{(k)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}$$

- This system can be solved for the coefficients $c^{(k)} = (c_0, \ldots, c_{m_k})$.

Algorithm: Pulay's DIIS for the acceleration of a fixed-point method based on a function g.

Algorithm 1: Pulay's DIIS [46] for the acceleration of a fixed-point method based on a function g.

Data: $x^{(0)}$, tol, m
$r^{(0)} = f(x^{(0)})$
$k = 0$
$m_k = 0$
while $\left\|r^{(k)}\right\|_2 > tol$ **do**
  solve the constrained least-squares problem for the coefficients $\left\{c_i^{(k)}\right\}_{i=0,\dots,m_k}$ using linear system (2.4)
  $x^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} g\left(x^{(k-m_k+i)}\right)$
  $r^{(k+1)} = f\left(x^{(k+1)}\right)$
  $m_{k+1} = \min(m_k + 1, m)$
  $k = k + 1$
end

- In Pulay's original DIIS, the error vector is defined as:

$$r^{(k)} = x^{(k+1)} - x^{(k)} = g(x^{(k)}) - x^{(k)}$$

- This choice links the error function $f(x)$ and the fixed-point function $g(x)$ by:

$$f(x) = g(x) - \mathrm{id}(x)$$

Handling Ill-Conditioning

- Generally, m is chosen to be a **small integer** to avoid complications.
- **Why a small m?**
  - Higher m increases computational cost.
  - Large m can lead to **ill-conditioning** of the Gramian matrix.
- Strategies to handle ill-conditioning:
  1. Remove old residuals until well-conditioned.
  2. Use an unconstrained formulation to improve conditioning.

From DIIS to CDIIS: The Need for Submanifold Preservation

- Why not just DIIS?
  - While DIIS is effective for accelerating convergence in general iterative schemes, it doesn't guarantee that iterates remain on constrained subspaces.
- The Challenge in SCF Calculations:
  - In SCF methods, matrices like the density matrix D need to satisfy physical constraints, such as idempotency: $D^2 = D$
- The Solution: CDIIS
  - CDIIS (Commutator DIIS) was developed to ensure that iterates remain within the submanifold of physically meaningful solutions, such as idempotent matrices in SCF methods.
  - CDIIS uses a **commutator** of matrices to define the error vector:

$$r^{(k)} = [F, D]$$

where F and D are matrices involved in SCF iterations.

Anderson Accelerations

- Anderson acceleration is a method used to speed up fixed-point iterations, such as solving nonlinear equations $h(x) = 0$. It improves convergence by using a combination of previous iterates, similar to DIIS but without the constraints applied in quantum chemistry.
- To solve a nonlinear equation $h(x) = 0$, a common approach is to perform fixed-point iteration:

$$g(x) = x + \beta h(x),$$

  where β is a sufficiently regular, homogeneous operator.
- Anderson acceleration, introduced for discretized nonlinear integral equations, is formulated as:

$$x^{(k+1)} = g(x^{(k)}) + \sum_{i=1}^{m_k} \theta_i^{(k)} \left(g(x^{(k-m_k-1+i)}) - g(x^{(k)}) \right),$$

Connection with DIIS

where the coefficients $\theta_i^{(k)}$ are chosen to minimize the norm of the residual:

$$\theta^{(k)} = \arg\min_{\theta \in \mathbb{R}^{m_k}} \left\| r^{(k)} + \sum_{i=1}^{m_k} \theta_i \left(r^{(k-m_k-1+i)} - r^{(k)} \right) \right\|_2 .$$

- The choice of error vectors $r^{(k)} = \beta h(x^{(k)})$ links the DIIS method with the Anderson acceleration.
- By setting $c_i^{(k)} = \theta_{i+1}^{(k)}$ and $c_{m_k}^{(k)} = 1 - \sum_{j=1}^{m_k} \theta_j^{(k)}$, we can reformulate Anderson acceleration in the DIIS form:

$$x^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} g(x^{(k-m_k+i)}).$$

- DIIS, when applied to fixed-point iteration, is essentially a reformulation of the Anderson acceleration.

Nonlinear Krylov Acceleration

- Nonlinear Krylov Acceleration (NKA) is another method similar to Anderson and DIIS, but it's specifically designed for accelerating nonlinear problems, such as those encountered in finite element methods.
- NKA constructs a Krylov subspace using past residuals, which enables faster convergence by solving a minimization problem over this subspace.
- NKA formulation:

$$x^{(k+1)} = g(x^{(k)}) - \sum_{i=1}^{m_k} \alpha_i^{(k)} \left(g(x^{(k-m_k+i)}) - g(x^{(k-m_k+i-1)}) \right),$$

where $\alpha^{(k)}$ is chosen to minimize:

$$\alpha^{(k)} = \arg \min_{\alpha \in \mathbb{R}^{m_k}} \left\| r^{(k)} - \sum_{i=1}^{m_k} \alpha_i \left(r^{(k-m_k+i)} - r^{(k-m_k+i-1)} \right) \right\|_2.$$

Relations between DIIS, Anderson Acceleration, and NKA

- All three methods—DIIS, Anderson Acceleration, and Nonlinear Krylov Acceleration—aim to improve the convergence of iterative methods by using past iterates or residuals.

Distinctions

- **DIIS:**
  - Specific to quantum chemistry and SCF methods.
  - Enforces constraints in optimization.
- **Anderson Acceleration:**
  - More general, applicable to various fixed-point problems.
  - Unconstrained approach for combining residuals.
- **NKA:**
  - Extends Krylov methods to nonlinear problems.
  - Constructs Krylov subspaces from past residuals.

While each method shares similar theoretical foundations, their application domains vary, making them each suited for different types of problems.

Quasi-Newton Methods Overview

- Quasi-Newton methods solve nonlinear systems by iteratively improving approximations of the Jacobian matrix or its inverse.
- These methods rely on the **secant condition**, which ensures consistency between changes in the solution and changes in the residuals:

$$B^{(k+1)}(x^{(k+1)} - x^{(k)}) = r^{(k+1)} - r^{(k)},$$

  where $B^{(k+1)}$ is the updated approximation of the Jacobian.
- This approach reduces the number of iterations needed for convergence compared to Newton's method, especially for large-scale problems.
- DIIS, while originally designed for electronic structure calculations, operates similarly to quasi-Newton methods by approximating the Jacobian through linear combinations of past iterates and residuals.

DIIS as a Multisecant Quasi-Newton Method

- In DIIS, we define two difference matrices at step k:

$$Y^{(k)} = \left[x^{(k-m_k+1)} - x^{(k-m_k)}, \ldots, x^{(k)} - x^{(k-1)} \right],$$

$$S^{(k)} = \left[r^{(k-m_k+1)} - r^{(k-m_k)}, \ldots, r^{(k)} - r^{(k-1)} \right],$$

  representing the changes in iterates and residuals over multiple previous steps.

- The DIIS update can be written as:

$$x^{(k+1)} = x^{(k)} - G^{(k)} r^{(k)},$$

  where the approximate inverse Jacobian is:

$$G^{(k)} = -I_n + (Y^{(k)} + S^{(k)})(S^{(k)^\top} S^{(k)})^{-1} S^{(k)^\top}.$$

- DIIS satisfies a **multisecant condition**, analogous to quasi-Newton methods:

$$G^{(k)} S^{(k)} = Y^{(k)},$$

  ensuring that updates are consistent with past changes in both iterates and residuals.

GMRES Method

- GMRES (Generalized Minimal Residual Method) is a well-known iterative method for solving $Ax = b$, where the goal is to minimize the residual.
- In GMRES, the solution at each step is the vector that minimizes the Euclidean norm of the residual within a Krylov subspace generated by the previous iterates.
- Finds $x^{(k+1)}$ by minimizing the Euclidean norm of the residual:

$$r_{\text{GMRES}}^{(k+1)} = b - Ax^{(k+1)}$$

in the affine space:

$$\mathcal{W}_{k+1} = \{v = x^{(0)} + z \mid z \in \mathcal{K}_{k+1}(A, r_{\text{GMRES}}^{(0)})\}.$$

where the Krylov subspace is defined as:

$$\mathcal{K}_j(A, r_{\text{GMRES}}^{(0)}) = \text{span}\{r_{\text{GMRES}}^{(0)}, Ar_{\text{GMRES}}^{(0)}, \ldots, A^{j-1} r_{\text{GMRES}}^{(0)}\}.$$

Equivalence with DIIS

- GMRES can be related to DIIS, as both methods rely on minimizing residuals, but GMRES is specifically used for solving linear systems, while DIIS is applied to nonlinear fixed-point problems in quantum chemistry.
- Theorem 2.2: Given $x_{\text{DIIS}}^{(0)} = x_{\text{GMRES}}^{(0)}$ and non-stagnating residuals:

$$r_{\text{GMRES}}^{(k-1)} \neq 0 \quad \text{and} \quad \|r_{\text{GMRES}}^{(j-1)}\|_2 > \|r_{\text{GMRES}}^{(j)}\|_2 \quad \text{for all } j < k,$$

the following holds:

$$x_{\text{GMRES}}^{(k)} = \sum_{i=0}^{k} c_i^{(k)} x_{\text{DIIS}}^{(i)}.$$

- Additionally:

$$x_{\text{DIIS}}^{(k+1)} = g(x_{\text{GMRES}}^{(k)}).$$

Anderson-Pulay Acceleration Methods

- Consider an error function f, a fixed-point function g, and a point $x^* \in \mathbb{R}^n$ such that $f(x^*) = 0$ and $g(x^*) = x^*$.
- The goal is to accelerate the convergence of a sequence $\{x^{(k)}\}$ generated by the fixed-point method.
- The acceleration is based on solving a least-squares problem at each iteration.

Algorithm 2: Fixed-depth Anderson–Pulay acceleration for a fixed-point method based on a function g.

Data: $x^{(0)}$, tol, m
$r^{(0)} = f\left(x^{(0)}\right)$
$k = 0$
$m_k = 0$
while $\left\|r^{(k)}\right\|_2 > tol$ **do**
  solve the constrained least-squares problem for the coefficients
  $\left\{c_i^{(k)}\right\}_{i=0,\dots,m_k} = \underset{\substack{(c_0,\dots,c_{m_k})\in\mathbb{R}^{m_k+1} \\ \sum_{i=0}^{m_k} c_i = 1}}{\arg\min} \left\|\sum_{i=0}^{m_k} c_i\, r^{(k-m_k+i)}\right\|_2$
  $x^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} g\left(x^{(k-m_k+i)}\right)$ **(version A)**
  or
  $x^{(k+1)} = g\left(\widetilde{x}^{(k+1)}\right)$ with $\widetilde{x}^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} x^{(k-m_k+i)}$ **(version P)**
  $r^{(k+1)} = f\left(x^{(k+1)}\right)$
  $m_{k+1} = \min(m_k + 1, m)$
  $k = k + 1$
end

Introduction to Adaptive Modifications

- This section introduces two adaptive modifications of the Anderson–Pulay acceleration:
  - Restarted Anderson–Pulay acceleration (Algorithm 3).
  - Adaptive-depth Anderson–Pulay acceleration (Algorithm 4).
- Both methods aim to improve convergence by adjusting the history of stored iterates.
- **Restart Strategy**
  - A restart occurs when a near-linear dependence among the stored iterates is detected.
  - When restarting, the history of iterates is reset, and the method starts afresh with a reduced number of stored iterates.
  - The key criterion for restarting is based on comparing the norm of the difference $\|s^{(k+1)} - \Pi_k s^{(k+1)}\|$ to the norm of $s^{(k+1)}$.
  - Define $s^{(k-m_k+i)} = r^{(k-m_k+i)} - r^{(k-m_k)}$ for $i = 1, \ldots, m_k$.

Restarted Depth

- Let Π_k denote the orthogonal projector onto the span of $\{s^{(k-m_k+1)}, \ldots, s^{(k)}\}$.
- Restart at step $k+1$ if:

$$\tau \|r^{(k+1)} - r^{(k-m_k)}\|_2 > \|(I - \Pi_k)(r^{(k+1)} - r^{(k-m_k)})\|_2,$$

  where τ is a parameter between 0 and 1.
- If this condition is not met, increase m_k by 1.
- In practice, the constrained vector of coefficients c is replaced by an unconstrained vector γ.
- Relationship between γ and c:

$$\sum_{i=0}^{m_k} c_i r^{(k-m_k+i)} = r^{(k-m_k)} + \sum_{i=1}^{m_k} \gamma_i s^{(k-m_k+i)}.$$

Restarted Depth

- Solving for γ simplifies the least-squares problem:

$$\gamma^{(k)} = \arg\min_{\gamma \in \mathbb{R}^{m_k}} \left\| r^{(k-m_k)} + \sum_{i=1}^{m_k} \gamma_i s^{(k-m_k+i)} \right\|_2 .$$

Algorithm 3: Restarted Anderson–Pulay acceleration for a fixed-point method based on a function g.

Data: $x^{(0)}$, tol, τ
$r^{(0)} = f(x^{(0)})$
$k = 0$
$m_k = 0$
while $\left\| r^{(k)} \right\|_2 > tol$ **do**
  if $m_k = 0$ **then**
    $x^{(k+1)} = g\left(x^{(k)}\right)$
  else
    solve the unconstrained least-squares problem for the coefficients $\left\{\gamma_i^{(k)}\right\}_{i=1,\dots,m_k} = \underset{(\gamma_1,\dots,\gamma_{m_k})\in\mathbb{R}^{m_k}}{\arg\min} \left\| r^{(k-m_k)} + \sum_{i=1}^{m_k} \gamma_i \, s^{(k-m_k+i)} \right\|_2$
    $x^{(k+1)} = g\left(x^{(k-m_k)}\right) + \sum_{i=1}^{m_k} \gamma_i^{(k)} \left(g\left(x^{(k-m_k+i)}\right) - g\left(x^{(k-m_k)}\right)\right)$ **(version A)**
    or
    $x^{(k+1)} = g\left(\widetilde{x}^{(k+1)}\right)$ with $\widetilde{x}^{(k+1)} = x^{(k-m_k)} + \sum_{i=1}^{m_k} \gamma_i^{(k)} \left(x^{(k-m_k+i)} - x^{(k-m_k)}\right)$ **(version P)**
  end
  $r^{(k+1)} = f\left(x^{(k+1)}\right)$
  $s^{(k+1)} = r^{(k+1)} - r^{(k-m_k)}$
  compute $\Pi_k s^{(k+1)}$ (the orthogonal projection of $s^{(k+1)}$ onto span$\left\{ s^{(k-m_k+1)}, \dots, s^{(k)} \right\}$)
  if $\tau \left\| s^{(k+1)} \right\|_2 > \left\| (\mathrm{id} - \Pi_k) s^{(k+1)} \right\|_2$ **then**
    $m_{k+1} = 0$
  else
    $m_{k+1} = m_k + 1$
  end
  $k = k + 1$
end

Adaptive Depth

- The adaptive modification adjusts the depth of stored iterates based on the relative size of the error norms.
- Set the depth m_{k+1} such that:

$$\delta \|r^{(i)}\|_2 < \|r^{(k+1)}\|_2 \quad \text{for } k+1-m_{k+1} \le i \le k,$$

where δ is a small positive parameter.

- This approach selectively retains iterates that contribute significantly to error reduction.

Algorithm 4: Adaptive-depth Anderson–Pulay acceleration for a fixed-point method based on a function g.

Data: $x^{(0)}$, tol, δ
$r^{(0)} = f(x^{(0)})$
$k = 0$
$m_k = 0$
while $\|r^{(k)}\|_2 > tol$ **do**
  if $m_k = 0$ **then**
    $x^{(k+1)} = g(x^{(k)})$
  else
    solve the unconstrained least-squares problem for the coefficients
    $\left\{\alpha_i^{(k)}\right\}_{i=1,\dots,m_k} = \underset{(\alpha_1,\dots,\alpha_{m_k})\in\mathbb{R}^{m_k}}{\arg\min} \left\|r^{(k)} - \sum_{i=1}^{m_k} \alpha_i\, s^{(k-m_k+i)}\right\|_2$
    $x^{(k+1)} = g(x^{(k)}) - \sum_{i=1}^{m_k} \alpha_i^{(k)} \left(g(x^{(k-m_k+i)}) - g(x^{(k-m_k+i-1)})\right)$ **(version A)**
    or
    $x^{(k+1)} = g(\widetilde{x}^{(k+1)})$ with $\widetilde{x}^{(k+1)} = x^{(k)} - \sum_{i=1}^{m_k} \alpha_i^{(k)} \left(x^{(k-m_k+i)} - x^{(k-m_k+i-1)}\right)$ **(version P)**
  end
  $r^{(k+1)} = f(x^{(k+1)})$
  $s^{(k+1)} = r^{(k+1)} - r^{(k)}$
  set m_{k+1} the largest integer $m \le m_k + 1$ such that for $k+1-m \le i \le k$, $\delta \|r^{(i)}\|_2 < \|r^{(k+1)}\|_2$
  $k = k+1$
end

Application to Electronic Ground State Calculations

The Anderson–Pulay acceleration is applied to compute the electronic ground state of molecular systems. Two main approximations are:

- Hartree–Fock Method (HF)
- Density Functional Theory (DFT)

Both the Hartree-Fock and Kohn-Sham equations are solved using the SCF method, an iterative fixed-point procedure.
The SCF algorithm iteratively updates the density matrix as follows:

$$\forall k \in \mathbb{N}, \begin{cases} F\left(D^{(k)}\right) C^{(k+1)} = S C^{(k+1)} E^{(k+1)} \\ \left(C^{(k+1)}\right)^* S C^{(k+1)} = I_N \\ D^{(k+1)} = C^{(k+1)} \left(C^{(k+1)}\right)^* \end{cases}$$

Convergence Acceleration

Assuming the uniform well-posedness property, the matrix $D^{(k+1)}$ is uniquely defined at each step and is the minimizer of a variational problem:

$$D^{(k+1)} = \operatorname{arginf} \left\{ \operatorname{tr} \left(F\left(D^{(k)}\right) D \right), D \in \mathcal{P}_N \right\}$$

This defines a fixed-point iteration process. To accelerate convergence, DIIS and CDIIS methods can be employed:

- DIIS uses the error vector $r^{(k)} = D^{(k)} - D^{(k-1)}$.
- CDIIS uses $r^{(k)} = \left[F\left(D^{(k)}\right), D^{(k)} \right]$.

CDIIS is popular due to its simplicity and fast convergence.

Forming the Next Pseudo-Density Matrix

CDIIS accelerates convergence by forming a linear combination of previous matrices:

$$\widetilde{D}^{(k+1)} = \sum_{i=0}^{m_k} c_i^{(k)} D^{(k-m_k+i)}$$

The coefficients $c_i^{(k)}$ are chosen to minimize the Frobenius norm of the error:

$$\boldsymbol{c}^{(k)} = \underset{\sum_{i=0}^{m_k} c_i = 1}{\arg\min} \left\| \sum_{i=0}^{m_k} c_i^{(k)} \left[F(D^{(k-m_k+i)}), D^{(k-m_k+i)} \right] \right\|_2$$

- The next density matrix $D^{(k+1)}$ is obtained by diagonalizing the Fock matrix $F(\widetilde{D}^{(k+1)})$ and applying the Aufbau principle.

Overview of Numerical Experiments

The goal of the numerical experiments is to illustrate the performance of the CDIIS variants introduced in the context of electronic ground state calculations, as discussed in Section 4. The experiments are implemented using the PySCF package

- For each CDIIS variant, the least-squares problem for the extrapolation coefficients is solved using QR factorization.
- The matrix involved is of tall-and-skinny type, and its factorization is updated step-by-step using the ‘scipy.linalg.qr‘ function.
- The QR decomposition is also used to compute the orthogonal projector, which plays a role in the restart condition.

Test Cases

- The experiments use molecular systems from benchmark datasets. The systems include challenging convergence tests for SCF algorithms.
- Molecular systems tested:
  A. Cadmium(II)-imidazole complex ([Cd(Im)]2+)
  B. Glycine (C2H5NO2)
  C. Dimethylnitramine (C2H6N2O2)
  D. Galactonolactone (C6H10O6)
- **Global Convergence Behaviour**
  - The first experiment tests the global convergence behaviour using an initial guess based on the core Hamiltonian.
  - The results show how CDIIS variants (restarted and adaptive-depth) perform compared to fixed-depth CDIIS.

Global Convergence Behaviour

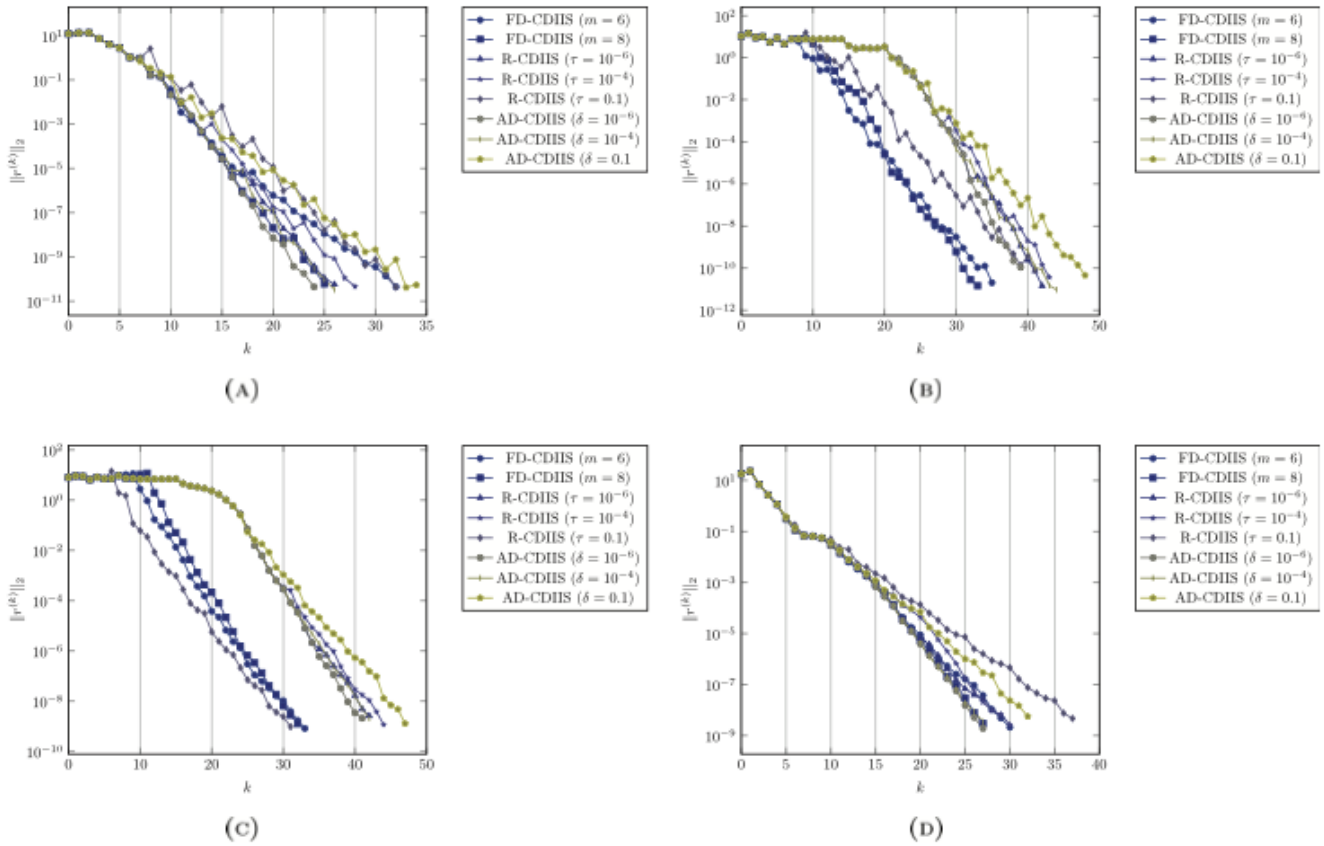

FIGURE 1. Residual norm convergence for the fixed-depth, restarted and adaptive-depth CDIIS on different molecular systems using an initial guess obtained by diagonalising the core Hamiltonian matrix. (A) Cadmium-imidazole complex in the RKS/B3LYP model with basis 3-21G. (B) Glycine molecule in the RKS/B3LYP model with basis 6-31Gs. (C) Dimethylnitramine molecule in the RHF model with basis 6-31G. (D) Galactonolactone molecule in the RHF model with basis 6-31G.

Figure: 1

Local Convergence Behaviour

- To assess local convergence, the CDIIS variants are combined with globally convergent methods, such as the EDIIS method.
- An improved initial guess is generated by the PySCF package.

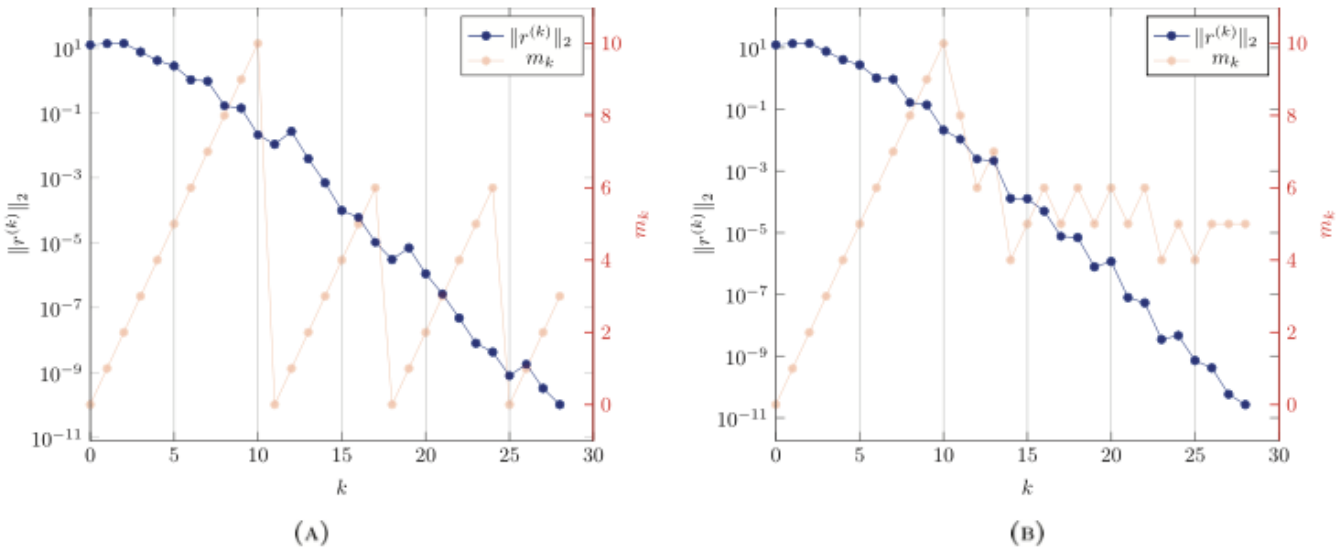

FIGURE 2. Residual norm convergence and corresponding depth value for the restarted and adaptive-depth CDIIS on the cadmium-imidazole complex in the RKS/B3LYP model with basis 3-21G, using an initial guess obtained by diagonalising the core Hamiltonian matrix. (A) Restarted CDIIS with $\tau = 10^{-4}$. (B) Adaptive-depth CDIIS with $\delta = 10^{-4}$.

Figure: 2

Mean Depth and Rate of Convergence

- Results shown for a set of molecules from Figure 1 with fewer parameter values.
- Restarted and adaptive-depth CDIIS outperform fixed-depth with well-chosen parameters.
- Efficiency observed by plotting error vector norms during experiments.
- Dimethylnitramine and glycine molecules were used for error vector norms.
- Observations:
  - Restart occurs after a significant error norm decrease.
  - Small slowdown in convergence post-restart.
  - Adaptive-depth avoids this issue.

Mean Depth and Rate of Convergence

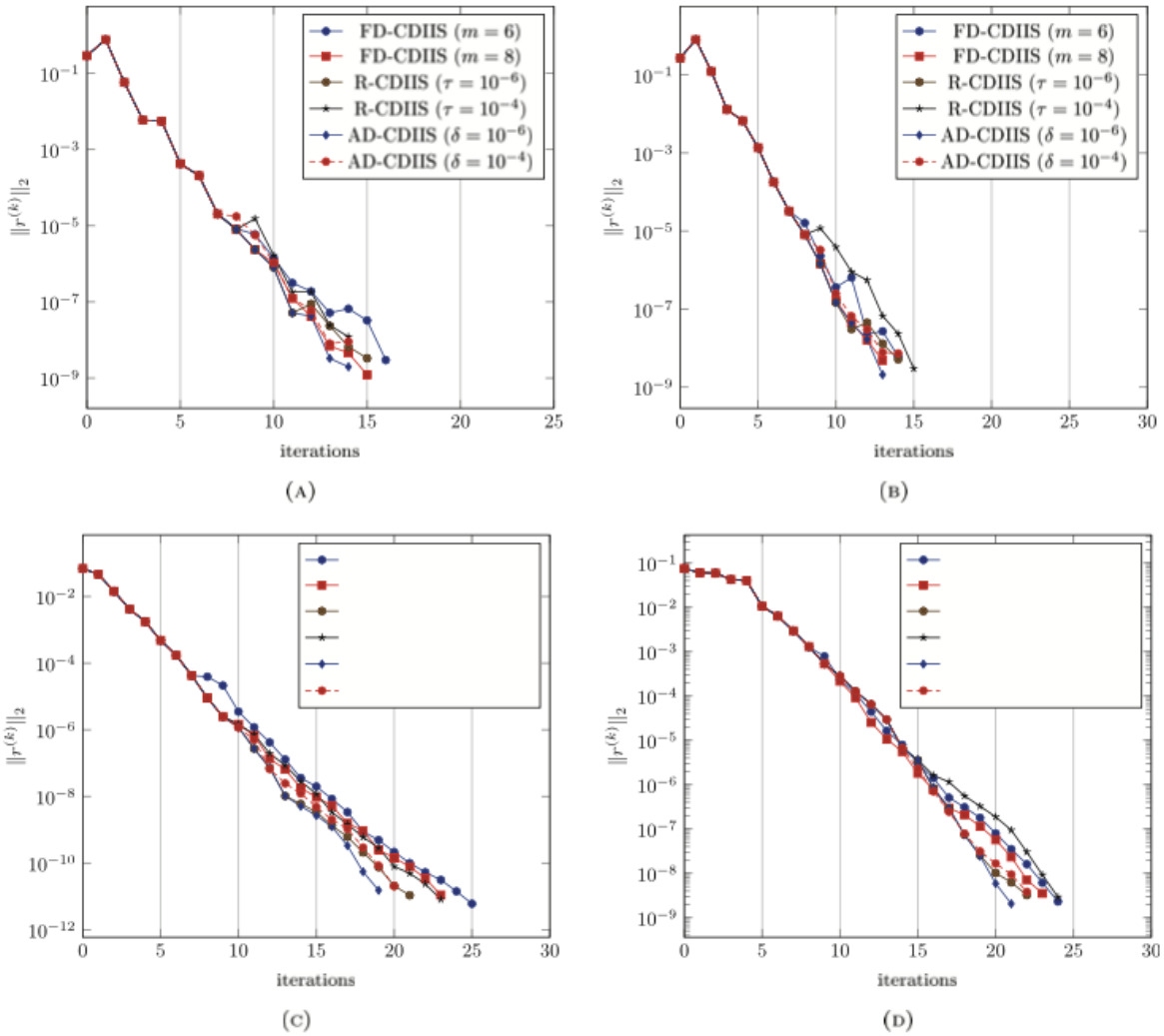

FIGURE 3. Residual norm convergence for the fixed-depth, restarted and adaptive-depth CDIIS on different molecular systems using an initial guess provided by a globally convergent method. (A) Cadmium-imidazole complex in the RKS/B3LYP model with basis 3-21G. (B) Glycine molecule in the RKS/B3LYP model with basis 6-31Gs. (C) Dimethylnitramine molecule in the RHF model with basis 6-31G. (D) Galactonolactone molecule in the RHF model with basis 6-31G.

Mean Depth and Rate of Convergence

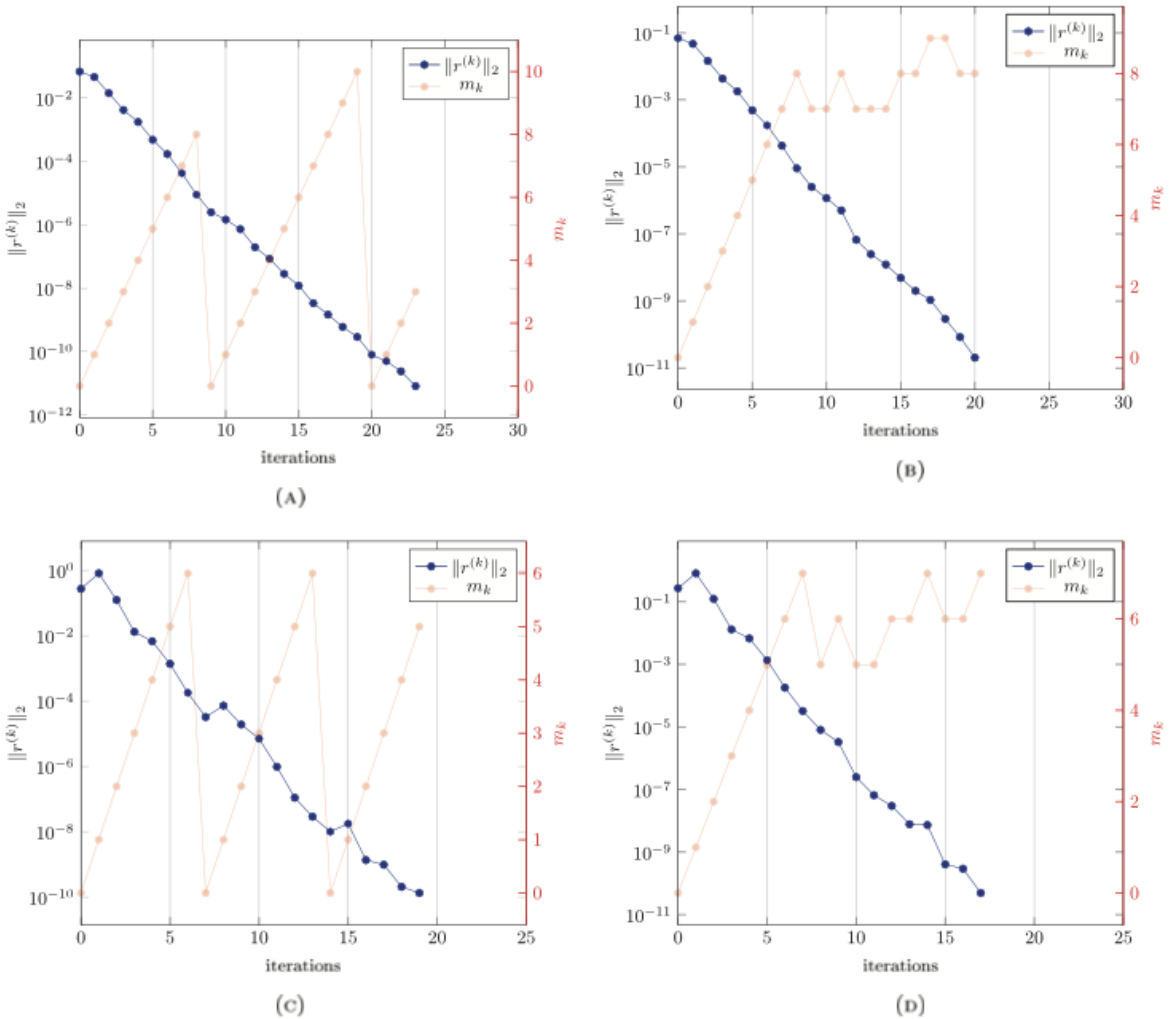

FIGURE 4. Residual norm convergence and corresponding depth for the restarted and adaptive-depth CDIIS on the dimethylnitramine and glycine molecules. (A) Restarted CDIIS with $\tau = 10^{-4}$ for the dimethylnitramine molecule in the RHF model with basis 6-31G. (B) Adaptive-depth CDIIS with $\delta = 10^{-4}$ for the dimethylnitramine molecule in the RHF model with basis 6-31G. (C) Restarted CDIIS with $\tau = 10^{-4}$ for the glycine molecule in the RKS/B3LYP model with basis 6-31Gs. (D) Adaptive-depth CDIIS with $\delta = 10^{-4}$ for the glycine molecule in the RKS/B3LYP model with basis 6-31Gs.

Mean Depth Comparison

- Mean depth $\bar{m}$ calculated as the average depth m_k during the experiments.
- Figure 5 shows the evolution of $\bar{m}$ as a function of restart parameter τ and adaptive-depth parameter δ.
- CDIIS variants (restarted and adaptive-depth) have lower storage/computation costs and better performance than fixed-depth.
- Figure 6: Rate of convergence computed using linear regression for restarted and adaptive-depth CDIIS.
- Rate increases as parameter values decrease.
- Smoother evolution seen for adaptive-depth variant.

Rate of Convergence

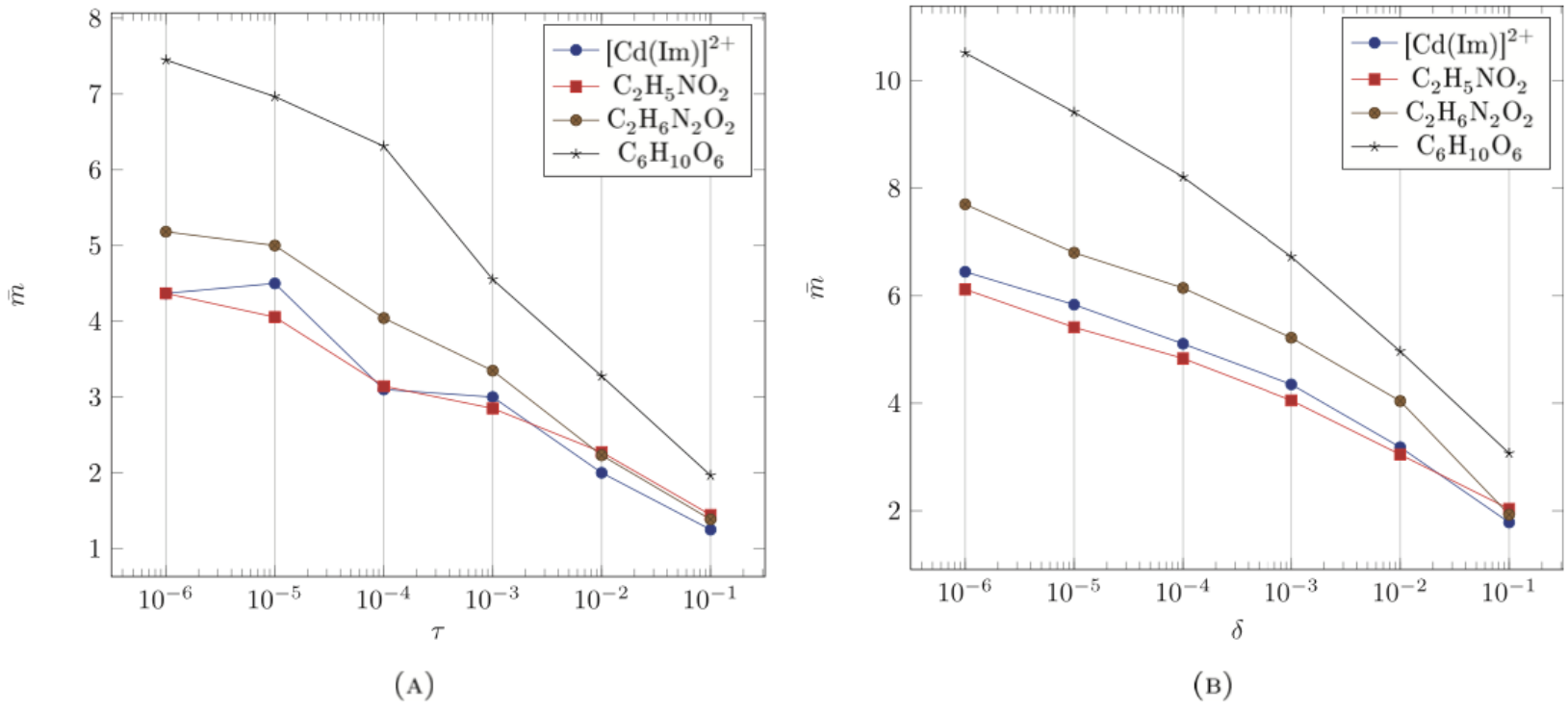

FIGURE 5. Evolution of the depth mean for different molecular systems and models. (A) Depth mean $\bar{m}$ as a function of the restart parameter τ. (B) Depth mean $\bar{m}$ as a function of the adaptive-depth parameter δ.

Figure: 5

Rate of Convergence

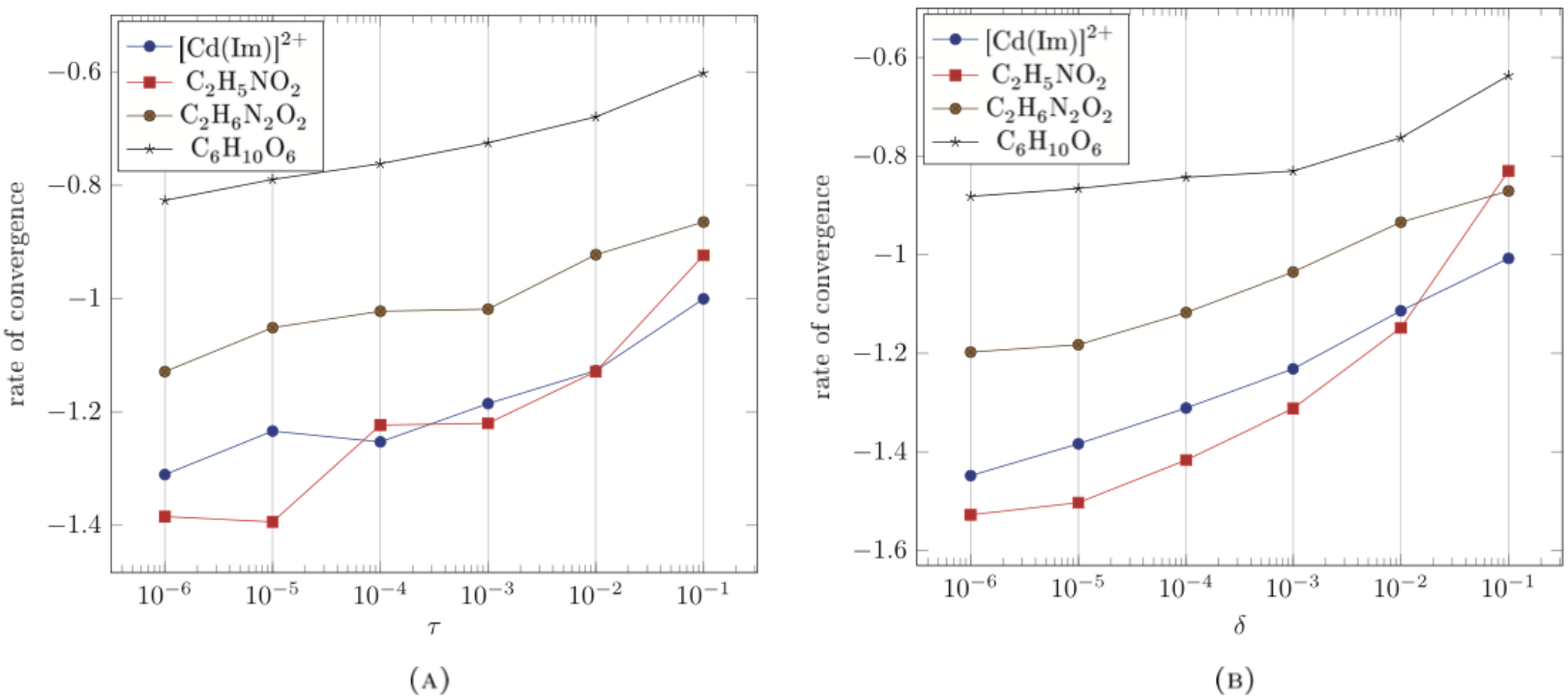

FIGURE 6. Evolution of the rate of convergence for different molecular systems. (A) Rate of convergence as a function of the restart parameter τ. (B) Rate of convergence as a function of the adaptive-depth parameter δ.

Figure: 6

Conclusion - Summary of Findings

- This work extends the DIIS and CDIIS techniques, originally introduced by Pulay, by proposing a general class of acceleration methods.
- We studied two variants:
  - Restarted method, based on a quasi-Newton approach.
  - Adaptive-depth method, using a new criterion.
- Our results rely on weaker assumptions than previous works on DIIS or Anderson acceleration, making them applicable to self-consistent field iterations in quantum chemistry.
- Numerical experiments show that both restarted and adaptive-depth CDIIS outperform fixed-depth methods, with adaptive-depth showing particular promise.
- Future work is needed to understand why practical parameter choices lead to better performance than theoretical estimates suggest.

Thank You!