BigDFT - linear scaling

Paul Boulanger
Stephan Mohr

21.10.2011
1 Philosophy of linear scaling
Philosophy of linear scaling
Why are wavelets suited for O(N)?

The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook

Table of contents

1 Philosophy of linear scaling
2 Why are wavelets suited for O(N)?
# Table of contents

1. Philosophy of linear scaling
2. Why are wavelets suited for \(O(N)\)?
3. The TMB method (Trace Minimizing Basis functions)
Table of contents

1. Philosophy of linear scaling
2. Why are wavelets suited for O(N)?
3. The TMB method (Trace Minimizing Basis functions)
4. Parallelization strategy
Table of contents

1. Philosophy of linear scaling
2. Why are wavelets suited for \( O(N) \)?
3. The TMB method (Trace Minimizing Basis functions)
4. Parallelization strategy
5. Performance results
# Table of contents

1. Philosophy of linear scaling
2. Why are wavelets suited for $O(N)$?
3. The TMB method (Trace Minimizing Basis functions)
4. Parallelization strategy
5. Performance results
6. Outlook
Outline

1 Philosophy of linear scaling
2 Why are wavelets suited for O(N)?
3 The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4 Parallelization strategy
5 Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6 Outlook
We solve the Kohn-Sham equation for the KS orbitals $\psi_{nk}(r)$:

$$\hat{H}_\text{KS} \psi_{nk}(r) = \epsilon_{nk} \psi_{nk}(r),$$

which are the eigenstates of the Kohn-Sham Hamiltonian:

$$\hat{H}_\text{KS} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + V_{\text{ext}}(r) + V_H(r) + V_{xc}(r) \right].$$

- **OUR GOAL**: diagonalize $\langle G | \hat{H}_\text{KS} | G' \rangle$
- Direct Diagonalization: required memory scales as $O(N_G^2)$ and CPU time as $O(N_G^3)$
- Since $N_G > N_{\text{occ}} > N_{\text{atoms}}$, only for very small systems are possible.
Instead of DD, obtain Kohn-Sham orbitals by minimizing the DFT total energy with respect to the wavefunction coefficients [1].

- Only occupied KS orbitals are needed: memory requirements decrease to $O(N_{occ}N_G)$
- Gradient of the total energy: $|g_{nk}⟩ = (\hat{H}^{KS} - \epsilon_{nk}) |\psi_{nk}(r)⟩$
- Need the density: $\rho(r) = \sum_{nk} f_{nk} \psi_{nk}^*(r) \psi_{nk}(r)$
- If we do this, we need $O(N_{occ}N_G^2)$ operations.
- We also need to orthogonalize the KS orbitals, which costs $O(N_{occ}^2N_G)$ operations.

Since the 80’s, Rise of the KS-DFT simulations:

- Increase of computer power: time for solving KS equations has become smaller and smaller.
- System sizes of DFT studies has become larger and larger.

However, in the last 5-10 years:

- Growth of computer power comes from increase of NCPU
- While the speed of each CPU remains unchanged.

Moral of this story:
⇒ We will need an approach with a good parallel efficiency
⇒ Standard DFT approaches will hit the CUBIC WALL
KS-DFT scheme: \( \psi_n(r) \) used to ease the calculation of \( E_{kin} \)

Hohenberg-Kohn theorem: only the density is required

If we look at the **density matrix**, defined as:

\[
\rho(r, r') = \sum_n f_n \psi_n^*(r) \psi_n(r')
\]  \( (1) \)

we find that it is ranged \([1]\) (i.e. it is fairly local):

\[
\rho(r, r') \to 0, \text{ when } |r - r'| \to \infty
\]

- The details of the decay are complex and material dependent
- Can be related to the localisation of Wannier functions
- **Kohn’s principle of near-sightedness** \([1]\)

---

\[ \mathcal{O}(N) \text{ codes all make the same approximation:} \]

\[ \rho(r, r') = 0, \text{ if } |r - r'| > r_{\text{cut}} \]  \hspace{1cm} (2)

This basically means that we can cut our system into \( r_{\text{cut}} \) sized sub-systems and solve each sub-system separately for the density matrix.


- For other methods (FOE,KPM): S.Y. Wu and C.S. Jayanthi, Phys.
Support functions

The density matrix is usually rewritten in terms of localized orbitals (or Wannier functions) that are called support functions:

$$\rho(r, r') = \sum_{I\alpha, J\beta} \phi_{I\alpha}(r) K_{I\alpha J\beta} \phi_{J\beta}(r')$$ (3)

Here, the near-sightedness is used as a criterion on which pairs of support functions are included.

- Crossover point increases with $N_{SF}$
- Proper convergence can be slow if $N_{SF}$ too small
The ONETEP approach

- Cut the system into subsystems of radii $r_{loc}^{α}$

$$E = 2K_{IαJβ}H_{JβIα} + E_{DC}[n]$$

$$\phi_{Iα}(r) = \sum_{μ} D_{μ}(r)c_{Iα}^{μ}$$

- Minimize the energy functional in two nested loops:
  - Inner loop: Minimize wrt the elements of the density kernel $K_{IαJβ}$
  - Outer loop: Minimize wrt the localized functions
Numerical considerations

Central to all these methods is:
- Sparse matrices: sparse matrix format and operations
- High Parallelization: need for communication

A lot of work is being done to develop efficient parallel sparse matrix operations.

To do this:
- Good load balancing
- Communication and calculation overlap
- Minimize global redistribution of data
Outline

1 Philosophy of linear scaling
2 Why are wavelets suited for $O(N)$?
3 The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4 Parallelization strategy
5 Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6 Outlook
One of the most important points for each electronic structure code is the underlying basis set.

Since we aim to reach linear scaling with our code, we need a basis set with compact support. There are several basis sets exhibiting this property, but the scaling function / wavelet basis used in our code combines several advantages:

- they are the only known functions which have compact support and are orthogonal at the same times
- they are a systematic basis set and the solution can therefore be obtained with arbitrary precision
- they allow for an adaptive resolution
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for O(N)?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
1 Philosophy of linear scaling
2 Why are wavelets suited for $O(N)$?
3 The TMB method (Trace Minimizing Basis functions)
   - the basis functions
     - flow chart
     - some details
4 Parallelization strategy
5 Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6 Outlook
We assume that there is some set of well localized basis functions \( \{ \phi_j \} \) such that we can write the physical orbitals \( \Psi_i \) as linear combinations of these:

\[
\Psi_i = \sum_j c_{ij} \phi_j
\]

If these basis functions are of good quality, we only need few of them and we have reduced the complexity of our system by some orders of magnitude (compared to representing \( \Psi_i \) in the original scaling function / wavelet basis).
general idea

We assume that there is some set of well localized basis functions \{\phi_j\} such that we can write the physical orbitals \(\Psi_i\) as linear combinations of these:

\[
\Psi_i = \sum_j c_{ij} \phi_j
\]

If these basis functions are of good quality, we only need few of them and we have reduced the complexity of our system by some orders of magnitude (compared to representing \(\Psi_i\) in the original scaling function / wavelet basis).
The basis functions themselves are given in the scaling function / wavelet basis, i.e.

\[ \phi_i(r) = \sum_{j1,j2,j3} s_{j1,j2,j3}^i \phi_{j1,j2,j3}(r) + \sum_{j1,j2,j3} \sum_{\nu=1}^{7} d_{j1,j2,j3;\nu}^i \psi_{j1,j2,j3}(r) \]

If the basis functions decay fast enough, the coefficients \(s\) and \(d\) are negligibly small far enough from the center of the basis function.

We can therefore set all coefficients equal to zero outside of a sphere around the center of each basis functions.

This procedure still conserves the adaptivity of the basis set (i.e. higher resolution close to the center of the localization region).
localization of the basis functions

The basis functions themselves are given in the scaling function / wavelet basis, i.e.

\[ \phi_i(r) = \sum_{j1,j2,j3} s_{j1,j2,j3}^i \phi_{j1,j2,j3}(r) + \sum_{j1,j2,j3} \sum_{\nu=1}^7 d_{j1,j2,j3;\nu}^i \psi_{j1,j2,j3}(r) \]

If the basis functions decay fast enough, the coefficients \( s \) and \( d \) are negligibly small far enough from the center of the basis function.
We can therefore set all coefficients equal to zero outside of a sphere around the center of each basis functions.

This procedure still conserves the adaptivity of the basis set (i.e. higher resolution close to the center of the localization region).
localization of the basis functions

The basis functions themselves are given in the scaling function / wavelet basis, i.e.

$$
\phi_i(r) = \sum_{j1,j2,j3} s^i_{j1,j2,j3} \phi_{j1,j2,j3}(r) + \sum_{j1,j2,j3} \sum_{\nu=1}^7 d^i_{j1,j2,j3;\nu} \psi_{j1,j2,j3}(r)
$$

If the basis functions decay fast enough, the coefficients $s$ and $d$ are negligibly small far enough from the center of the basis function.

We can therefore set all coefficients equal to zero outside of a sphere around the center of each basis functions.

This procedure still conserves the adaptivity of the basis set (i.e. higher resolution close to the center of the localization region).
localization of the basis functions

Figure: The grid for a small molecule. We see the global box (coarse and fine) and the localization region (coarse and fine).
How do these basis functions \( \{\phi_j\} \) look like?

We want them to have the following properties:

- they have to be strictly localized
- they should adapt themselves to their chemical environment, i.e. they are not fixed, but rather flexible
How do these basis functions $\{\phi_j\}$ look like?

We want them to have the following properties:

- they have to be strictly localized
- they should adapt themselves to their chemical environment, i.e. they are not fixed, but rather flexible
form of the basis functions

How do these basis functions \( \{ \phi_j \} \) look like?

We want them to have the following properties:

- they have to be strictly localized
- they should adapt themselves to their chemical environment, i.e. they are not fixed, but rather flexible
creation of the basis functions

To create our basis functions, we take the standard DFT Hamiltonian and add a confining potential to it (which is different for each localization region):

$$\mathcal{H}^i = \mathcal{H}^{DFT} + \mathcal{V}^i.$$ 

This implies that each localization region has its own Hamiltonian.

The standard DFT Hamiltonian is given by

$$\mathcal{H}^{DFT} = -\frac{1}{2} \nabla^2 + \mathcal{V}_H[\rho] + \mathcal{V}_{XC}[\rho] + \mathcal{V}_{ex}$$

where $\mathcal{V}_H$ is the Hartree potential, $\mathcal{V}_{XC}$ the exchange correlation potential and $\mathcal{V}_{ex}$ the external ionic potential.
To create our basis functions, we take the standard DFT Hamiltonian and add a confining potential to it (which is different for each localization region):

$$\mathcal{H}^i = \mathcal{H}^{DFT} + \mathcal{V}^i.$$

This implies that each localization region has its own Hamiltonian.

The standard DFT Hamiltonian is given by

$$\mathcal{H}^{DFT} = -\frac{1}{2} \nabla^2 + \mathcal{V}_H[\rho] + \mathcal{V}_{XC}[\rho] + \mathcal{V}_{ex}$$

where $\mathcal{V}_H$ is the Hartree potential, $\mathcal{V}_{XC}$ the exchange correlation potential and $\mathcal{V}_{ex}$ the external ionic potential.
creation of the basis functions

The confining potential $\mathcal{V}^i$ is different for each localization region and has the form

$$\mathcal{V}^i = c(r - \mathbf{R}_i)^4$$

where $\mathbf{R}_i$ is the center of the localization region $i$.

This confining potential ensures that the basis functions are well localized, but close to their center they feel the “correct” Hamiltonian and should therefore be of excellent quality.
The confining potential $\mathcal{V}^i$ is different for each localization region and has the form

$$\mathcal{V}^i = c(r - R_i)^4$$

where $R_i$ is the center of the localization region $i$.

This confining potential ensures that the basis functions are well localized, but close to their center they feel the “correct” Hamiltonian and should therefore be of excellent quality.
creation of the basis functions

Since the Hamiltonian is different for each localization region, we can not search for eigenfunctions to determine the basis functions $\phi_j$.

Instead we have to minimize the trace of the Hamiltonian, i.e. our basis functions are given by the condition

$$\min_{\phi_i} \sum_i \langle \phi_i \mid \mathcal{H}^i \mid \phi_i \rangle \quad \text{with} \quad \langle \phi_i \mid \phi_j \rangle = \delta_{ij}$$
creation of the basis functions

Since the Hamiltonian is different for each localization region, we can not search for eigenfunctions to determine the basis functions $\phi_j$.

Instead we have to minimize the trace of the Hamiltonian, i.e. our basis functions are given by the condition

$$\min_{\phi_i} \sum_i \langle \phi_i | H^i | \phi_i \rangle \quad \text{with} \quad \langle \phi_i | \phi_j \rangle = \delta_{ij}$$
the expansion coefficients

Once we have determined the basis functions $\phi_j$, we can calculate the Hamiltonian matrix in this basis, i.e.

$$H_{jk}^{DFT} = \langle \phi_j | \mathcal{H}^{DFT} | \phi_k \rangle$$

Diagonalizing this matrix to get the eigenvectors $H^{DFT} c_i = \epsilon_i c_i$ provides us with the expansion coefficients for the physical orbitals, i.e.

$$\Psi_i = \sum_j c_{ij} \phi_j$$

We will never (at the moment almost never) explicitly build this linear combination, but always work with the localized basis functions $\phi_j$ and the expansion coefficients $c_i$. 
the expansion coefficients

Once we have determined the basis functions $\phi_j$, we can calculate the Hamiltonian matrix in this basis, i.e.

$$H^{DFT}_{jk} = \langle \phi_j | H^{DFT} | \phi_k \rangle$$

Diagonalizing this matrix to get the eigenvectors $H^{DFT} c_i = \epsilon_i c_i$ provides us with the expansion coefficients for the physical orbitals, i.e.

$$\Psi_i = \sum_j c_{ij} \phi_j$$

We will never (at the moment almost never) explicitly build this linear combination, but always work with the localized basis functions $\phi_j$ and the expansion coefficients $c$. 
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for $O(N)$?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
overview of the method

- Input guess for
  - basis functions \( \phi \)
  - charge density
  - potential
overview of the method
overview of the method

- Input guess for
  - basis functions \( \phi \)
  - charge density
  - potential

- communicate potential
- communicate orbitals
- calculate Hamiltonian and diagonalize
- calculate charge density
overview of the method

- Input guess for
  - basis functions $\phi$
  - charge density
  - potential

- calculate Hamiltonian
- and diagonalize

- calculate charge density

- mix the density

- communicate potential
- communicate orbitals
overview of the method

- Input guess for
  - basis functions phi
  - charge density
  - potential

- calculate Hamiltonian and diagonalize
- calculate charge density
- mix the density
- update the potential

communicate potential
communicate orbitals
overview of the method

- Input guess for
  - basis functions \( \phi \)
  - charge density
  - potential

- calculate Hamiltonian and diagonalize
- calculate charge density
- mix the density
- update the potential
- reached self consistency
- standard SCF cycle
- communicate potential
- communicate orbitals
overview of the method

- Input guess for:
  - basis functions \( \phi \)
  - charge density
  - potential

- calculate Hamiltonian and diagonalize
- calculate charge density
- mix the density
- update the potential
- reached self consistency

- standard SCF cycle
  - communicate potential
  - communicate orbitals

- reoptimize basis functions
- no

- the basis functions
- flow chart
- some details
overview of the method
overview of the method

The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook
overview of the method

- Input guess for basis functions, charge density, potential
- Orthonormalization
- Hamiltonian application
- Orthoconstraint
- Preconditioning
- Communication (overlap matrix)
- Communicate potential
- Communicate orbitals
- Calculate Hamiltonian and diagonalize
- Calculate charge density
- Mix the density
- Update the potential
- Reached self-consistency
- Reoptimize basis functions

Standard SCF cycle:
- Yes
- No

Parallelization strategy
Performance results
Outlook
overview of the method

The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook

Input guess for
- basis functions PHI
- charge density
- potential

Orthonormalization
Hamiltonian application
Orthoconstraint
Preconditioning
Improve basis functions

Communicate overlap matrix
Communicate potential
Communicate orbitals

Calculate Hamiltonian and diagonalize
Calculate charge density
Mix the density
Update the potential

Reached self consistency

Reoptimize basis functions

Standard SCF cycle
overview of the method

- input guess for
  - basis functions \( \phi \)
  - charge density
  - potential

- orthonormalization
- Hamiltonian application
- orthoconstraint
- preconditioning
- improve basis functions
- stop optimizing basis functions

- reoptimize basis functions
- yes
- no

- loop to create basis functions
- communication (overlap matrix)
- standard SCF cycle
- communicate potential
- calculate Hamiltonian and diagonalize
- calculate charge density
- mix the density
- update the potential
- reached self consistency
- no
overview of the method
**Philosophy of linear scaling**

Why are wavelets suited for $O(N)$?

The TMB method (Trace Minimizing Basis functions)

Parallelization strategy

Performance results

Outlook

overview of the method
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for $O(N)$?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
There are many possibilities for creating an initial guess for the basis functions $\phi_j$.

We decided to do it in the subspace of some atomic orbitals by writing each basis function as a linear combination of these atomic orbitals $\chi_l$, i.e.

$$\phi_j = \sum_l d_{jl} \chi_l$$

The coefficients $d$ are given by

$$\min \sum_i \sum_i \langle d_i | H^i | d_i \rangle \quad \text{with} \quad \langle d_i | d_j \rangle = \delta_{ij}$$

where $H^i$ is the Hamiltonian of the localization region $i$ in the basis of the atomic orbitals.
There are many possibilities for creating an initial guess for the basis functions $\phi_j$.

We decided to do it in the subspace of some atomic orbitals by writing each basis function as a linear combination of these atomic orbitals $\chi_l$, i.e.

$$\phi_j = \sum_l d_{jl} \chi_l$$

The coefficients $d$ are given by

$$\min_d \sum_i \langle d_i | H^i | d_i \rangle \quad \text{with} \quad \langle d_i | d_j \rangle = \delta_{ij}$$

where $H^i$ is the Hamiltonian of the localization region $i$ in the basis of the atomic orbitals.
The total charge density is given by

\[ \rho(r) = \sum_i \Psi_i(r)\Psi_i(r) = \sum_i \sum_{jk} c_{ij} c_{ik} \phi_j(r)\phi_k(r) = \sum_{jk} M_{jk} \phi_j(r)\phi_k(r) \]

with \( M_{jk} = \sum_i c_{ij} c_{ik} \).

Since the basis functions \( \phi \) are strictly localized, this summation scales as \( O(N) \).
calculation of the charge density

The total charge density is given by

\[ \rho(r) = \sum_i \Psi_i(r) \Psi_i(r) \]
\[ = \sum_i \sum_{jk} c_{ij} c_{ik} \phi_j(r) \phi_k(r) \]
\[ = \sum_{jk} M_{jk} \phi_j(r) \phi_k(r) \quad \text{with} \quad M_{jk} = \sum_i c_{ij} c_{ik} \]

Since the basis functions \( \phi \) are strictly localized, this summation scales as \( \mathcal{O}(N) \).
The basis functions $\phi_i$ are orthonormalized by using a Loewdin orthonormalization, i.e.

$$\left| \tilde{\phi}_i \right> = \sum_j S_{ij}^{-\frac{1}{2}} \left| \phi_j \right>$$

with $S_{ij} = \langle \phi_i | \phi_j \rangle$.

Due to its sparsity the overlap matrix $S$ can be calculated with $O(N)$. However the inversion of $S$ scales cubically. There are two ways to avoid this problem:
The basis functions $\phi_i$ are orthonormalized by using a Loewdin orthonormalization, i.e.

$$|\tilde{\phi}_i\rangle = \sum_j S_{ij}^{-\frac{1}{2}} |\phi_j\rangle \quad \text{with} \quad S_{ij} = \langle \phi_i | \phi_j \rangle.$$ 

Due to its sparsity the overlap matrix $S$ can be calculated with $O(N)$. However the inversion of $S$ scales cubically. There are two ways to avoid this problem:
Use some parallel Linear Algebra package to speed up the calculation (but in principle the scaling remains cubically)

Use a Taylor approximation of $S^{-\frac{1}{2}}$, which only requires sparse matrix-matrix multiplications. Due to the strict localization of the basis functions, exact orthogonality can not be achieved, and so the error introduced by this approximation is acceptable.
Use some parallel Linear Algebra package to speed up the calculation (but in principle the scaling remains cubically).

Use a Taylor approximation of $S^{-\frac{1}{2}}$, which only requires sparse matrix-matrix multiplications. Due to the strict localization of the basis functions, exact orthogonality can not be achieved, and so the error introduced by this approximation is acceptable.
The gradient is obtained by applying the Hamiltonian, i.e.

\[ |g_i\rangle = \mathcal{H}^i |\phi_i\rangle. \]

This operation can be entirely done with \( O(N) \):

- for our scaling function / wavelet basis, the kinetic energy is a convolution with a filter of finite length.
- the potential needs only be applied within the localization region of the basis function
The gradient is obtained by applying the Hamiltonian, i.e.

\[ |g_i\rangle = \mathcal{H}^i |\phi_i\rangle. \]

This operation can be entirely done with \( O(N) \):

- for our scaling function / wavelet basis, the kinetic energy is a convolution with a filter of finite length.
- the potential needs only be applied within the localization region of the basis function.
gradient calculation

The gradient is obtained by applying the Hamiltonian, i.e.

\[ |g_i\rangle = \mathcal{H}^i |\phi_i\rangle. \]

This operation can be entirely done with \( O(N) \):
- for our scaling function / wavelet basis, the kinetic energy is a convolution with a filter of finite length.
- the potential needs only be applied within the localization region of the basis function
orthonormality constraint

The orthonormality constraint looks a bit more complicated than usual since:

- the Hamiltonian is not symmetric due to the orbital dependent potential
- the overlap matrix among the basis functions is not exactly the identity

It has the form

$$|\tilde{g}_i\rangle = |g_i\rangle - \frac{1}{2} \sum_j \Lambda'_{ij} |\phi_j\rangle - \frac{1}{2} \sum_j \Lambda'_{ji} |\phi_j\rangle$$

with $$\Lambda' = S^{-1}\Lambda, S_{ij} = \langle \phi_i | \phi_j \rangle, \Lambda_{ij} = \langle \phi_i | g_j \rangle$$

The same strategies as for the orthonormalization are used to reduce the scaling.
The orthonormality constraint looks a bit more complicated than usual since:

- the Hamiltonian is not symmetric due to the orbital dependent potential
- the overlap matrix among the basis functions is not exactly the identity

It has the form

$$|\tilde{g}_i\rangle = |g_i\rangle - \frac{1}{2} \sum_j \Lambda_{ij}^\prime |\phi_j\rangle - \frac{1}{2} \sum_j \Lambda_{ji}^\prime |\phi_j\rangle$$

with $\Lambda^\prime = S^{-1}\Lambda$, $S_{ij} = \langle\phi_i | \phi_j\rangle$, $\Lambda_{ij} = \langle\phi_i | g_j\rangle$

The same strategies as for the orthonormalization are used to reduce the scaling.
To speed up the convergence of the basis functions, we have to precondition the gradient. To do so, the following equation has to be solved:

\[
\left( -\frac{1}{2} \nabla^2 + \gamma + c(r - R_i)^4 \right) |\tilde{g}_i\rangle = |\tilde{\tilde{g}}_i\rangle
\]

This equation does not have to be solved exactly, but only approximately (using a CG procedure).

Due to the scaling function / wavelet basis, all these operations involve only convolutions with a filter of finite length. Therefore the preconditioning can be done with $O(N)$. 
To speed up the convergence of the basis functions, we have to precondition the gradient. To do so, the following equation has to be solved:

\[
\left(-\frac{1}{2}\nabla^2 + \gamma + c(r - R_i)^4\right)|\tilde{g}_i\rangle = |\tilde{g}_i\rangle
\]

This equation does not have to be solved exactly, but only approximately (using a CG procedure).

Due to the scaling function / wavelet basis, all these operations involve only convolutions with a filter of finite length. Therefore the preconditioning can be done with \(\mathcal{O}(N)\).
To improve the basis functions update them with the preconditioned gradient, for instance with steepest descent:

\[ |\phi_i\rangle \leftarrow |\phi_i\rangle - \alpha |\tilde{g}_i\rangle \]

In practice this update is done with DIIS rather than SD.
diagonalization of the Hamiltonian

With the basis functions $\phi_i$ we can then calculate the Hamiltonian matrix with the same techniques as described above.

If we have $n$ basis functions, we can diagonalize this $n \times n$ matrix to get the $m$ lowest eigenvalues and eigenvectors, where $m$ is the number of physical orbitals.

With these eigenvectors $c_i$ the physical orbitals are then given by

$$\Psi_i = \sum_j c_{ij} \phi_j$$

This diagonalization scales cubically with system size, but again using some parallel Linear Algebra package can alleviate the problem.
diagonalization of the Hamiltonian

With the basis functions $\phi_i$ we can then calculate the Hamiltonian matrix with the same techniques as described above.

If we have $n$ basis functions, we can diagonalize this $n \times n$ matrix to get the $m$ lowest eigenvalues and eigenvectors, where $m$ is the number of physical orbitals.

With these eigenvectors $c_i$ the physical orbitals are then given by

$$\Psi_i = \sum_j c_{ij} \phi_j$$

This diagonalization scales cubically with system size, but again using some parallel Linear Algebra package can alleviate the problem.
With the basis functions $\phi_i$ we can then calculate the Hamiltonian matrix with the same techniques as described above.

If we have $n$ basis functions, we can diagonalize this $n \times n$ matrix to get the $m$ lowest eigenvalues and eigenvectors, where $m$ is the number of physical orbitals.

With these eigenvectors $c_i$ the physical orbitals are then given by

$$\Psi_i = \sum_j c_{ij} \phi_j$$

This diagonalization scales cubically with system size, but again using some parallel Linear Algebra package can alleviate the problem.
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for $O(N)$?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
Our code is parallelized using MPI (distributed memory). Adding OpenMP (shared memory) is planned for the future.

In our implementation there are two quantities that are parallelized: the basis functions and the charge density / potential. Every MPI process has a few orbitals and does all calculations on these orbitals. In addition every process handles one slice of the potential.

The situation is depicted below for 5 orbitals on 5 MPI processes.
Our code is parallelized using MPI (distributed memory). Adding OpenMP (shared memory) is planned for the future.

In our implementation there are two quantities that are parallelized: the basis functions and the charge density / potential. Every MPI process has a few orbitals and does all calculations on these orbitals. In addition every process handles one slice of the potential.

The situation is depicted below for 5 orbitals on 5 MPI processes.
Communication between the processes is required for the following operations:

- Calculation of overlap matrix / Hamiltonian matrix
- Calculation of the charge density
- Building the potential for the Hamiltonian application

Almost all communications are done using non-blocking point to point communication.
Overlap matrix

To calculate the overlap matrix, we need to communicate between the processes handling two overlapping basis functions. Only that overlap will be communicated.

In this way each process calculates a small part of the matrix. Using a collective communication call brings the entire matrix to all processes.
calculation of charge density

To calculate the charge density each process will do the job for one slice. No additional communication is required later, since the Poisson solver uses the same slices distribution.

Therefore each process has to gather all orbitals extending into its slice. Only the part in the range of that slice will be communicated.
Each process needs that part of the potential that has to be applied to its orbital.

However each process has only the potential of its slice, whereas the orbital may extend over several slices. Therefore each process needs to gather the potential from additional (sub)slices belonging to other processes.
overlapping communication with computation

The communication for the calculation of the charge density can be overlapped with computation in an efficient way and is hardly visible at all.

The communication of the potential has fewer overlap, but still seems to be fast enough.

However the communication for the calculation of the overlap matrix consumes a lot of time. Efforts to improve this part are under way.
overlapping communication with computation

The communication for the calculation of the charge density can be overlapped with computation in an efficient way and is hardly visible at all.

The communication of the potential has fewer overlap, but still seems to be fast enough.

However the communication for the calculation of the overlap matrix consumes a lot of time. Efforts to improve this part are under way.
The communication for the calculation of the charge density can be overlapped with computation in an efficient way and is hardly visible at all.

The communication of the potential has fewer overlap, but still seems to be fast enough.

However the communication for the calculation of the overlap matrix consumes a lot of time. Efforts to improve this part are under way.
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for O(N)?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for O(N)?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
To examine the scaling with respect to the size of the system we did calculations for alkanes $C_nH_{2n+2}$ for values of $n$ in between 100 and 600 for both the cubic and the linear TMB version.

Due to the chain-like structure of the alkanes, these systems are an excellent candidate for $O(N)$ methods.
Philosophy of linear scaling
Why are wavelets suited for O(N)?
The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook

scaling for alkanes $C_nH_{2n+2}$: runtime

![Graph showing execution time vs number of C atoms for cubic, linear TMB, and linear TMB without initializations]
Philosophy of linear scaling
Why are wavelets suited for $O(N)$?
The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook

scaling for alkanes $C_nH_{2n+2}$: memory

![Graph showing memory scaling for alkanes $C_nH_{2n+2}$ with system size and number of processes.](image_url)
Outline

1 Philosophy of linear scaling
2 Why are wavelets suited for O(N)?
3 The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4 Parallelization strategy
5 Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6 Outlook
To compare the scaling with respect to the number of MPI processes used, we calculated the total energy of the alkane $C_{200}H_{402}$ for a different number of processors.
Philosophy of linear scaling
Why are wavelets suited for O(N)?
The TMB method (Trace Minimizing Basis functions)
Parallelization strategy
Performance results
Outlook

scaling with system size
scaling with number of processes
accuracy

scaling with processes for $C_{200}H_{402}$

![Graph showing speedup vs. number of processes for $C_{200}H_{402}$](image)

Paul Boulanger Stephan Mohr
BigDFT - linear scaling
One sees that some parts of the code which are not yet optimized very much prevent a better speedup.

Here is a similar plot for the cubic version.
One sees that some parts of the code which are not yet optimized very much prevent a better speedup.

Here is a similar plot for the cubic version.
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for O(N)?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook

Paul Boulanger Stephan Mohr

BigDFT - linear scaling
accuracy check for alkanes

To check the accuracy we did two calculations for two slightly different systems: A standard alkane and an alkane that is twisted in the middle.
We calculated the energy difference between these molecules for various lengths. It seems that this difference is quite independent of the length of the alkane.

The energy differences for both the cubic and the linear version are shown below.
Outline

1. Philosophy of linear scaling
2. Why are wavelets suited for $O(N)$?
3. The TMB method (Trace Minimizing Basis functions)
   - the basis functions
   - flow chart
   - some details
4. Parallelization strategy
5. Performance results
   - scaling with system size
   - scaling with number of processes
   - accuracy
6. Outlook
As the code is still under development, its functionality is limited.

At the moment, only the electronic ground state properties can be calculated. Forces are also available, but do not contain any Pulay correction. This correction is always required if the basis depends on the positions of the atoms, as it is the case here. Tests whether the absence of this correction limits the usage of the code or not are under way.

In the future any functionality available with the standard cubic version of BigDFT should also be possible with the linear scaling version.
Outlook

As the code is still under development, its functionality is limited.

At the moment, only the electronic ground state properties can be calculated. Forces are also available, but do not contain any Pulay correction. This correction is always required if the basis depends on the positions of the atoms, as it is the case here. Tests whether the absence of this correction limits the usage of the code or not are under way.

In the future any functionality available with the standard cubic version of BigDFT should also be possible with the linear scaling version.
Outlook

As the code is still under development, its functionality is limited.

At the moment, only the electronic ground state properties can be calculated. Forces are also available, but do not contain any Pulay correction. This correction is always required if the basis depends on the positions of the atoms, as it is the case here. Tests whether the absence of this correction limits the usage of the code or not are under way.

In the future any functionality available with the standard cubic version of BigDFT should also be possible with the linear scaling version.