TOWARD A GENERAL PURPOSE LINEAR SCALING METHOD FOR EIGENVALUE PROBLEMS IN QUANTUM CHEMISTRY

Maxime BARRAULT - SEMINAIRE 2003

Eric Cancès, **CERMICS - Ecole Nationale des Ponts & Chaussées**
Guy Bencteux, Véronique Duwig, **EDF - Electricité de France**

Outline

→ Specificity of the eigenvalue problem arising in Quantum Chemistry

→ Reformulations of the problem : FOE and DMM

→ Insulators / Metallic systems

→ Numerical test on a basic model system

→ Perspectives

The eigenvalue problem in Quantum Chemistry

→ Notations

N number of electrons,

N_b number of basis functions.

Localized basis functions $\Longrightarrow N_b = kN$ with $k \simeq 3, 4.$

Let $\lambda \in \mathbb{R}^M$ be a family of parameters of the system (positions of the nuclei). **For each $\boldsymbol{\lambda}$, two SYMMETRIC matrices $\boldsymbol{S_\lambda}$ and $\boldsymbol{H_\lambda}$ of $\boldsymbol{\mathcal{M}^{N_b}(\mathbb{R})}$ are associated** in regular way.

S_λ is positive definite,

S_λ, H_λ are **SPARSE** without specific pattern.

The eigenvalue problem in Quantum Chemistry (2)

Let $\boldsymbol{D_\lambda} \in \boldsymbol{\mathcal{M}_S^{N_b}(\mathbb{R})}$ be the solution of the **Hartree-Fock or Kohn-Sham models** (95% of the calculations in computational chemistry)

$$
(\boldsymbol{\mathcal{P}}) \left\{ \begin{array}{l}
H_\lambda c_i = \epsilon_i S_\lambda c_i, \\
c_i^T S_\lambda c_j = \delta_{ij}, \\
\boldsymbol{\epsilon_1 \leq \ldots \leq \epsilon_N < \epsilon_{N+1} \leq \ldots \leq \epsilon_{N_b},} \\
D_\lambda = \sum_{i=1}^{\boldsymbol{N}} c_i c_i^T.
\end{array} \right.
$$

Two situations can occure

- (i) for λ_0, we want to calculate D_{λ_0},
- (ii) D_{λ_k} is known, we want to calculate $D_{\lambda_{k+1}}$ for $\lambda_{k+1} = \lambda_k + \delta\lambda_k$.

The eigenvalue problem in Quantum Chemistry (3)

$\longrightarrow$ Orders of magnitude

- Depending on the simulation, (*ii*) occurs between **10** (single point electronic energy) and **100000** (molecular dynamics) **times**,
- for "large" systems of interest, $\boldsymbol{N = 10000}$ (nanotubes), **100000** (DNA, proteins), ...

$\longrightarrow$ Standard computation

The resolution of $(\mathcal{P})$ is generally done by a standard diagonalization algorithm which scales as $\boldsymbol{N^3}$ (or N^2 if H_λ and S_λ are band matrices). This procedure cannot be used for large systems.

$\hookrightarrow$ **METHODS OF LOWER COMPLEXITY ARE NEEDED**

The eigenvalue problem in Quantum Chemistry (4)

$\longrightarrow$ Specificity of $(\mathcal{P})$

As the unknown of physical interest is D_λ on the pattern of H_λ, only the **projector on the subspace generated by the lowest N eigenvectors on the pattern of H_λ is needed**,

$\hookrightarrow$

- $*$ too much information is computed by a complete diagonalization,
- $*$ the lowest N eigenvectors are not really needed,
- $*$ not all the coefficients of D_λ have to be computed.

Such a specificity makes it possible to reformulate $(\mathcal{P})$. As S_λ and H_λ are sparse, lower complexity can be reached.

$\longrightarrow$ In the following

$$S_\lambda = I_{N_b}, \quad -1 = \epsilon_1 \leq \ldots \leq \epsilon_N < 0 < \epsilon_{N+1} \leq \ldots \leq \epsilon_{N_b} \leq 1,$$

$$\gamma = \frac{\epsilon_{N+1} - \epsilon_N}{\epsilon_{N_b} - \epsilon_1} \neq 0.$$

Fermi Operator Expansion algorithm (1)

→ Characterization of D_λ

$$
\begin{aligned}
\boldsymbol{D_\lambda} &= \mathcal{H}(-H_\lambda) \text{ with the Heaviside function } \mathcal{H}, \\
&\simeq \boldsymbol{p(H_\lambda)} \ \forall\, p \text{ such that } \boldsymbol{(*)} \left| \begin{array}{l} \boldsymbol{p(\epsilon_i) \simeq 1 \quad \forall\, i = 1, \ldots, N} \\ \boldsymbol{p(\epsilon_i) \simeq 0 \quad \forall\, i = N+1, \ldots, N_b} \end{array} \right.
\end{aligned}
$$

→ Choice of p : Chebyshev polynomials

→ Related approaches : Chebyshev Jackson polynomials (Saad, Chelikowsky), Rational approximation (Goedecker).

Fermi Operator Expansion algorithm (2)

→ Approximation of the Heaviside function $\mathcal{H}$ by Chebyshev polynomials

$$\boldsymbol{D_\lambda \simeq p_m(-H_\lambda) = \frac{1}{2} - \frac{2}{\pi}\sum_{n=1}^{m} \frac{1}{n} \sin\left(\frac{n\pi}{2}\right) T_n(-H_\lambda),}$$

with

$$\left| \begin{array}{l} T_0(-H_\lambda) = I_{N_b}, \\ T_1(-H_\lambda) = -H_\lambda, \\ \forall\, n \geq 1 \quad \boldsymbol{T_{n+1}(-H_\lambda) = -2H_\lambda T_n(-H_\lambda) - T_{n-1}(-H_\lambda).} \end{array} \right.$$

→ Choice of degree m : $\boldsymbol{m = \mathcal{O}(\gamma^{-1})}$.

GREATER the gap $\boldsymbol{\gamma}$, **LOWER** the degree $\boldsymbol{m}$.

Density Matrix Minimization algorithm (1)

→ Characterization of D_λ

$$\boldsymbol{D_\lambda = \mathrm{arginf}\left\{\mathrm{Tr}\big(H_\lambda D\big),\ D^2 = D,\ \mathrm{Tr}(D) = N,\ D \in \mathcal{M}_S^{N_b}(\mathbb{R})\right\}.}$$

→ **Theorem**

D_λ is the local minimum on $\mathcal{M}_S^{N_b}(\mathbb{R})$ of

$$\boldsymbol{\Omega(D) = \mathrm{Tr}\Big(H_\lambda\big(3D^2 - 2D^3\big)\Big).}$$

→ Optimization

- nonlinear conjugate gradient,
- good initial guess is needed.

Proof

- $\boldsymbol{D_\lambda}$ **is a local minimum of** $\boldsymbol{\Omega}$ **on** $\boldsymbol{\mathcal{M}_S^{N_b}(\mathbb{R})}$

$\forall\, h \in \mathcal{M}_S^{N_b}(\mathbb{R})$

$$\begin{aligned}\Big(\boldsymbol{\nabla\Omega(D_\lambda), h}\Big) &= \mathrm{Tr}\Big(6H_\lambda D_\lambda\big(1-D_\lambda\big)h\Big),\\ &\boldsymbol{= 0},\\ \frac{1}{2}\Big(h, \nabla^2\Omega(D_\lambda)h\Big) &= 3\mathrm{Tr}\big(H_\lambda h^2\big) - 4\mathrm{Tr}\big(H_\lambda D_\lambda h^2\big) - 2\mathrm{Tr}\big(H_\lambda h D_\lambda h\big).\end{aligned}$$

Let Q and $(E_{ij})_{i,j}$ be the matrix of eigenvectors of H_λ and the canonical basis of $\mathcal{M}^{N_b}(\mathbb{R})$. Let $\tilde{h}_{ij} = Q^T(E_{ij}+E_{ji})Q$, then

$\forall\, 1 \le i \le j \le N_b$

$$\begin{aligned}\frac{1}{2}\Big(\tilde{h}_{ij}, \nabla^2\Omega(D_\lambda)\tilde{h}_{ij}\Big) = 3(\epsilon_i+\epsilon_j) &- 2\epsilon_i(2\mathbb{I}_{i\le N} + \mathbb{I}_{j\le N})\\ &-2\epsilon_j(2\mathbb{I}_{j\le N} + \mathbb{I}_{i\le N}).\end{aligned}$$

Density Matrix Minimization algorithm (3)

As

$$\left|\begin{array}{lcll} \epsilon_i & < & 0 & \forall\, i = 1, \ldots, N \\ \epsilon_i & > & 0 & \forall\, i = N+1, \ldots, N_b \end{array}\right. ,$$

$$\forall\, 1 \leq i \leq j \leq N_b \quad \left(\tilde{h}_{ij}, \nabla^2 \Omega(D_\lambda) \tilde{h}_{ij}\right) > 0.$$

Thus, $$\boldsymbol{\forall\, h \in \mathcal{M}_S^{N_b}(\mathbb{R}) \quad \left(h, \nabla^2 \Omega(D_\lambda) h\right) > 0.}$$

- **it exists a unique local minimum of $\boldsymbol{\Omega}$ on $\boldsymbol{\mathcal{M}_S^{N_b}(\mathbb{R})}$.**

$\rightarrow$ Condition number : $\kappa = \mathcal{O}(\gamma^{-1})$

$$\Longrightarrow \boldsymbol{n_{it} \simeq \sqrt{\kappa} = \mathcal{O}(\gamma^{-1/2}).}$$

GREATER the gap $\boldsymbol{\gamma}$, **LOWER** the number of iterations $\boldsymbol{n_{it}}$.

Insulators / Metallic systems

→ Insulators

$$\left| \begin{array}{l} \gamma = \mathcal{O}(1) \Longrightarrow m, n_{it} \text{ are low,} \\ D_\lambda \text{ is sparse for physical reasons,} \end{array} \right.$$

↪ **FOE and DMM can scale linearly with N.**

→ Metallic systems

$$\left| \begin{array}{l} \gamma \ll 1 \Longrightarrow m, n_{it} \text{ are high,} \\ D_\lambda \text{ is dense for physical reasons,} \end{array} \right.$$

↪ **FOE and DMM scale as N^3 with a great prefactor : lower complexity can't be reached.**

Sparsity of the density matrix D_λ

Denote by $(\phi_i)_{i=1,\ldots,N_b}$ the set of the basis functions. The **density matrix operator** $\boldsymbol{\mathcal{D}}$ is defined by

$$\mathcal{D}(r,r') = \sum_{i=1}^{N_b}\sum_{j=1}^{N_b} D_{i,j}\phi_i(r)\phi_j(r').$$

→ **insulators**

$$\mathcal{D}(r,r') \simeq k_1 \frac{cos\Big(k_1|r-r'|\Big)}{|r-r'|^2}\boldsymbol{exp\Big(-k_2|r-r'|\Big)}.$$

→ **metallic** systems

$$\mathcal{D}(r,r') \simeq k_1 \frac{cos\Big(k_1|r-r'|\Big)}{\boldsymbol{|r-r'|^2}}.$$

Metallic systems : Adaptation of FOE and DMM (1)

→ Back to FOE and DMM

$$D_\lambda = q(H_\lambda) \quad \text{with} \quad \text{degree}(q) = \gamma^{-\alpha},\ \alpha > 0.$$

→ Idea

CREATE A GAP in H_λ in order to use FOE (DMM) with a LOW m (n_{it}).

→ How ? : calculation of d eigenmodes $(c_i, \epsilon_i)_{i=1,d}$ around 0 by IRA algorithm (Sorensen).

→ Problem

$$\bar{H}_\lambda = H_\lambda - \sum_{i=1}^{d} \epsilon_i c_i c_i^T \quad \text{is } \textbf{NOT} \text{ sparse}$$

Metallic systems : Adaptation of FOE and DMM (2)

→ Orthogonality of eigenvectors of H_λ

$$q(\bar{H}_\lambda) = q(H_\lambda) - \sum_{i=1}^{d} q(\epsilon_i) c_i c_i^T$$

→ Consequence

At each iteration k of FOE and DMM, D_k is of the form

$$\begin{aligned} D_k &= q_k\Big(H_\lambda - \sum_{i=1}^{d} \epsilon_i c_i c_i^T\Big) + \sum_{i=1}^{d} \mathbb{1}_{\epsilon_i<0} c_i c_i^T, \\ &= \underbrace{q_k(H_\lambda)}_{\textbf{sparse}} + \underbrace{\sum_{i=1}^{d} \Big(\mathbb{1}_{\epsilon_i<0} - q_k(\epsilon_i)\Big) c_i c_i^T}_{\textbf{dense}}. \end{aligned}$$

→ Obtention of D_λ

The reconstruction of D_λ is only done on the pattern of H_λ.

Metallic systems : DMM (1)

→ Notation : $\bar{H}_\lambda = H_\lambda - \sum_{i=1}^{d} \epsilon_i c_i c_i^T$.

→ Initialization : $D_0 = p_0(H_\lambda)$,

$$\bar{D}_0 = p_0(\bar{H}_\lambda) = D_0 - \sum_{i=1}^{d} p_0(\epsilon_i) c_i c_i^T.$$

→ Iterative processus

$$\begin{aligned} \bar{G}_k &= 6\bar{H}_\lambda \bar{D}_{k-1}(1 - \bar{D}_{k-1}) &&= G_k + \sum_{i=1}^{d} g_k(i) c_i c_i^T, \\ \bar{X}_k &= -\bar{G}_k + \beta_k^{CG} \bar{X}_{k-1} &&= X_k - \sum_{i=1}^{d} x_k(i) c_i c_i^T. \end{aligned}$$

Metallic systems : DMM (2)

$$\begin{aligned}\alpha_k &= \mathrm{arginf}_{\mathrm{loc}}\left\{\Omega\left(\bar{D}_{k-1}+\alpha\bar{X}_k\right),\ \alpha\in\mathbb{R}\right\},\\ &= \mathrm{arginf}_{\mathrm{loc}}\left\{\Omega\left(D_{k-1}+\alpha X_k\right)+\sum_{i=1}^{d}\epsilon_i\left(3f_k^\alpha(i)^2-2f_k^\alpha(i)^3\right),\ \alpha\in\mathbb{R}\right\},\end{aligned}$$

$$\bar{D}_k = \bar{D}_{k-1}+\alpha_k\bar{X}_k = D_k-\sum_{i=1}^{d}p_k(i)c_ic_i^T.$$

$\rightarrow$ Reconstruction after convergence

On the pattern of H_λ,

$$\begin{aligned}D_\lambda &= \bar{D}_\infty+\sum_{i=1}^{d}\mathbb{1}_{\epsilon_i<0}\ c_ic_i^T,\\ &= D_\infty+\sum_{i=1}^{d}\left(\mathbb{1}_{\epsilon_i<0}-p_\infty(i)\right)c_ic_i^T.\end{aligned}$$

Méthode d'Arnoldi simple

→ Principe : méthode de projection orthogonale sur un sous-espace de Krylov $\mathcal{K}^m(A, v_1)$ pour des matrices quelconques (non symétriques).

→ Algorithme simple

Choisir un vecteur initial v_1 normé

Pour $k = 1, \ldots,$ jusqu'à convergence (au plus n itérations)

Construction d'une base orthonormale V_k de $\mathcal{K}^k(A, v_1)$.

Construction de $H_k = V_k^t A V_k$.

Calculer des éléments propres $(\mu_i^k, z_i^k)_{i=1,k}$ de H_k.

Fin

Récupérer les éléments propres désirés : $(\mu_j^\infty, u_j^\infty = V_\infty z_j^\infty)_{j \in \mathcal{I}}$.

Problèmes de l'algorithme d'Arnoldi simple (et de Lanczos)

$\longrightarrow$ Perte d'orthogonalité, coût calcul et mémoire prohibitifs pour m grand.

$\hookrightarrow$ Redémarrage

$\longrightarrow$ On n'utilise pas l'information pour accélérer la convergence vers les modes propres cherchées.

$\hookrightarrow$ Redémarrage particulier

$\longrightarrow$ On garde les éléments propres convergés.

$\hookrightarrow$ Déflation de ces éléments

$\longrightarrow$ Difficulté à cerner les valeurs propres multiples (comme l'algorithme de Lanczos) et les clusters.

$\hookrightarrow$ Algorithme par bloc

Méthode d'Arnoldi avec redémarrage simple

Choisir un vecteur initial v_1 normé et m la dimension de l'espace $\mathcal{K}^m(A, v_1)$ de travail,

Pour $k = 1, \ldots,$ jusqu'à convergence

Construction d'une base orthonormale V_k de $\mathcal{K}^m(A, v_k)$.

Construction de $H_k = V_k^t A V_k$.

Calcul des éléments propres $(\mu_i^k, z_i^k)_{i=1,m}$ de H_k.

Calcul de v_{k+1} à l'aide des éléments propres de H_k.

Fin

Récupérer les éléments propres de A désirés : $(\mu_j^\infty, u_j^\infty = V_\infty z_j^\infty)_{j \in \mathcal{I}}$.

$\longrightarrow$ Remarque : Le redémarrage ralentit la convergence

$\hookrightarrow$ Compromis à réaliser à travers le choix de m.

Redémarrage : Calcul de v_{k+1}

$\longrightarrow$ Plusieurs redémarrages

- $*$ Redémarrage simple : $v_{k+1} = v_k^m$.
- $*$ Redémarrage implicite par Chebyshev : $v_{k+1} = P_k(A)v_k$.
- $*$ Redémarrage du à Saad :

$$v_{k+1} = \sum_{i=1}^{p} \alpha_i u_i$$

où u_i sont les meilleures estimations obtenues des d modes cherchés à l'itération k.

$\longmapsto$ Problème : Comment choisir les α_i ?

$\longrightarrow$ Problème lié au redémarrage : Il faut reconstruire à chaque itération $\mathcal{K}^m(A, v_{k+1})$.

Implicit Restart Arnoldi algorithm

→ Principle : calculation of d eigenmodes of a matrix A.

→ Our case : $\boldsymbol{A = H_\lambda^{-1}}$.

→ Algorithm

Let us consider a unit vector v_1, l such that $d+l$ is the work dimension space and $H^1_{d+1,d}$ the Hessenberg matrix obtained after d Arnoldi iterations with v_1 as starting vector.

For $k = 1, \ldots,$ until convergence

Generation of H^k_{d+l} by l additional Arnoldi iterations.

Diagonalization of H^k_{d+l}.

Calculation of v_{k+1} with the eigenmodes of H^k_{d+l}.

Generation of $H^{k+1}_{d+1,d}$ by d Arnoldi iterations with v_{k+1} as starting vector.

What is IRA ? (1)

→ Principle : IRA is the clever solution found by Sorensen to restart v_{k+1} while calculating $H^{k+1}_{d+1,d}$ at the same time.

→ Central point : use of l implicit QR factorization of $H^{k+1}_{d+l+1,d}$.

→ After Arnoldi

$$AV_{d+l} = V_{d+l}H_{d+l} + h_{d+l+1,d+l}v_{d+l+1}e^T_{d+l}.$$

→ Explicit QR Factorization with shift μ

$$H_{d+l} - \mu I_{d+l} = QR \text{ with Hessenberg } Q,$$
$$AV^+ - V^+H^+ = \tilde{h}^+_{d+l,d+l-1}v_{d+l}e^T_{d+l-1}$$

with $V^+ = V_{d+l}Q$ and $H^+ = RQ + \mu I_{d+l} = Q^T H_{d+l}Q$.

What is IRA ? (2)

$\rightarrow$ l implicit QR factorization with shifts μ_j

$$AV_{d+l}^{+} = V_{d+l}^{+} H_{d+l}^{+} + h_{d+l+1,d+l} v_{d+l+1} e_{d+l}^{T} Q$$

with $V_{d+l}^{+} = V_{d+l} Q$, $H_{d+l}^{+} = Q^T H_{d+l} Q$ and $Q = Q_1 \ldots Q_l$.

As Q_i Hessenberg for all $i = 1,\ l$, we have

$$h_{d+l+1,d+l} e_{d+l}^{T} Q = (\beta e_d^T, b^T), \quad b \in \mathbb{R}^l.$$

Therefore

$$\Longrightarrow \left| \begin{array}{lcl} AV_d^{+} & = & V_d^{+} H_d^{+} + h_{d+1,d}^{+} v_{d+1}^{+} e_d^T + \beta v_{d+l+1} e_d^T, \\ AV_d^{+} & = & V_d^{+} H_d^{+} + \tilde{h}_{d+1,d}^{+} s^{+} e_d^T \text{ with } \|s^{+}\|_2 = 1 \text{ and } s^{+} \perp V_d^{+}, \\ v_1^{+} & = & \gamma \prod_{i=1}^{l} (A - \mu_i I_n) v_1. \end{array} \right.$$

Metallic systems : Complexity

→ Question

Does it exist an OPTIMAL VALUE d_N for the number d of deflated eigenvectors such that the overall complexity scales as N^{α} with $\alpha < 3$ for a given accuracy ?

→ Overall complexity

The overall complexity is difficult to estimate because H_λ is sparse. However, if we assume a linear distribution of the eigenvalues of H_λ around 0, we can estimate it in two cases

- **(1) Band matrix**
- **(2) Calculus on a fixed band pattern**

$$\begin{array}{l|lcl} \text{Case 1} & d_N = \mathcal{O}(N^{3/5}) & \Longrightarrow & \boldsymbol{\mathcal{C}^t_{px} = \mathcal{O}(N^{9/5})} \\ \text{Case 2} & d_N = \mathcal{O}(N^{1/3}) & \Longrightarrow & \boldsymbol{\mathcal{C}^t_{px} = \mathcal{O}(N^{4/3})} \end{array}.$$

Complexity of the algorithm (2)

$\longrightarrow$ IRA : $l = \mathcal{O}(d)$

$$\boldsymbol{\mathcal{C}_{px}(IRA) = \mathcal{O}(Nd + d^3).}$$

$\longrightarrow p_m(H_\lambda)$

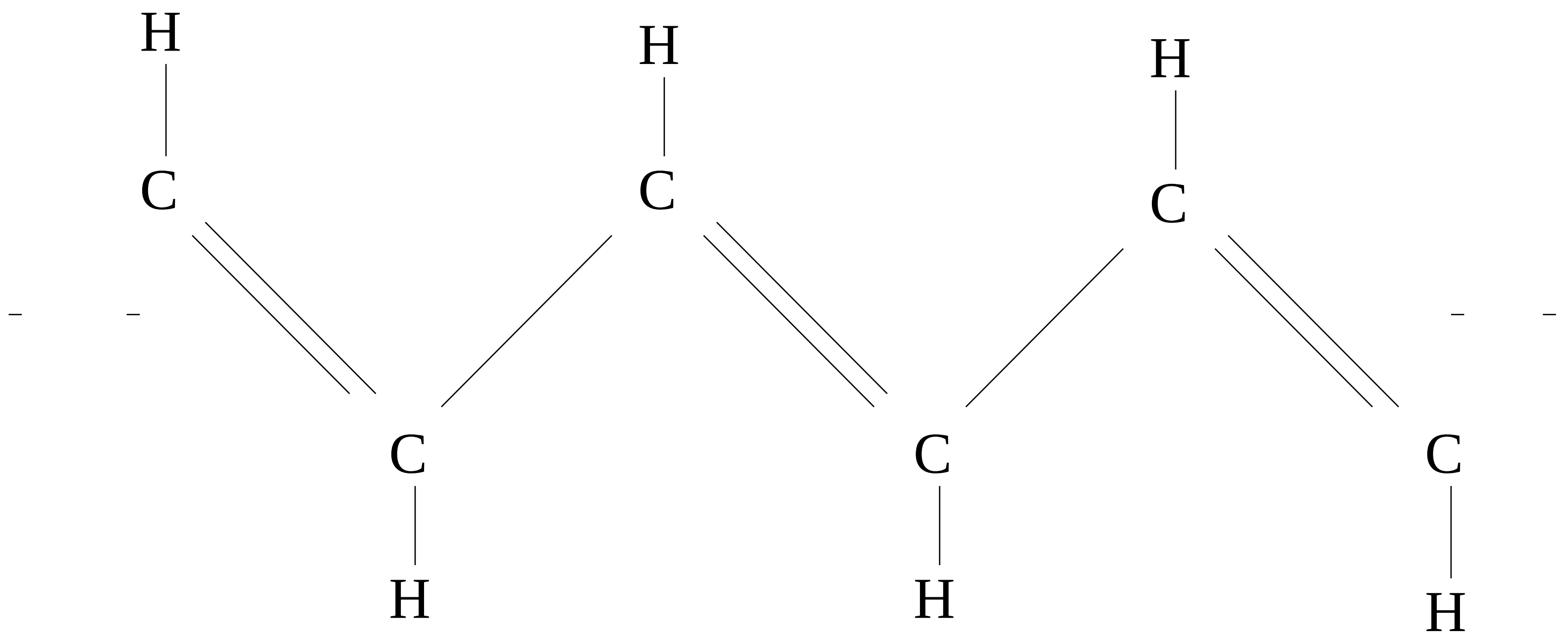
- **(1) Band matrix of bandwith $\boldsymbol{p}$** : $\mathcal{O}(p^2 N m^2)$,
- **(2) Calculus on a fixed pattern of bandwidth $\boldsymbol{p}$** : $\mathcal{O}(p^2 N m)$.

$\longrightarrow$ (3)

$$\boldsymbol{\mathcal{C}_{px}(3) = Nd}.$$

Numerical test on a basic model system (1)

$\rightarrow$ Polyen



$\rightarrow$ Model : semi-empirical (one basis function by electron)

$$\Longrightarrow N_b = 2 * N \quad \text{(N number of electron pairs).}$$

Numerical test on a basic model system (2)

$$H_\lambda = \begin{bmatrix} \alpha & \beta & 0 & \cdots & 0 \\ \beta & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta \\ 0 & \cdots & 0 & \beta & \alpha \end{bmatrix}, \ \gamma \sim \frac{1}{N_b}.$$

$\longrightarrow$ Error indicator : $\boldsymbol{\varepsilon_D = \|\Delta_r D\|_\infty}$

$$\left| \begin{array}{lll} (\Delta_r D)_{ij} & = \dfrac{D_{ij}^{exa} - D_{ij}^{sol}}{|D_{ij}^{exa}| + |D_{ij}^{sol}|} & \text{if } H_\lambda^{i,j} \neq 0 \text{ and } |D_{ij}^{exa}| + |D_{ij}^{sol}| \geq \varepsilon \\ & = \quad 0 & \text{else} \end{array} \right. .$$

$\longrightarrow$ Imposed pattern : in order to run the test case.

Numerical test on a basic model system (3)

→ Parameters

- $N_b = 10000 \to 40000$,
- $N = N_b/3$ ($N = N_b/2$ is a too symmetric model),
- pattern : bandwith 200,
- $(\varepsilon, \varepsilon_D^{max}) = (10^{-4}, 10^{-2})$.

→ Numerical complexity : $\boldsymbol{\mathcal{O}(N^{3/2})}$

Perspectives

$\longrightarrow$ Numerical investigations

- band matrices H_λ and S_λ (one dimensional polymer),
- sparse matrix H_λ and $S_\lambda = I_{N_b}$ (semi-empirical model),
- sparse matrices H_λ and S_λ.

$\longrightarrow$ Calculation of optimal d_N

- automatic strategy,

$\longrightarrow$ Pattern of work

- study of the effect of a fixed pattern,
- implementation of an adaptative strategy for DMM,

$\longrightarrow$ DMM

- Initial guess different from a polynomial of H_λ.