Analytical Gradients for the Range-Separated Random Phase Approximation Correlation Energies Using a Lagrangian Framework

Bastien Mussard[a], János G. Ángyán[a], Péter G. Szalay[b]

[a] CRM^2, Université de Lorraine, Nancy, France
[b] Laboratory of Theoretical Chemistry, Eötvös Loránd University (ELTE), Budapest, Hungary

bastien.mussard@univ-lorraine.fr

9 juin 2013

Analytical Gradients for the Range-Separated Random Phase Approximation Correlation Energies Using a Lagrangian Framework

Bastien Mussard[a], János G. Ángyán[a], Péter G. Szalay[b]

[a] CRM^2, Université de Lorraine, Nancy, France
[b] Laboratory of Theoretical Chemistry, Eötvös Loránd University (ELTE), Budapest, Hungary

bastien.mussard@univ-lorraine.fr

9 juin 2013

Range-Separated context

Observation

- DFT : good at short range
- WF methods : suitable at long range

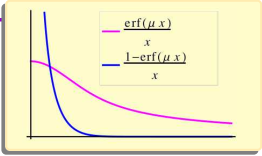
Range-Separated context

Observation

- DFT : good at short range
- WF methods : suitable at long range

The idea **Mix the two** :

$$\boxed{E^{\mathrm{RSH}} = \min_{\phi} \left\{ \langle\phi|\ \hat{T} + \hat{V}_{ne} + \hat{W}_{ee}^{\mathrm{lr},\mu}\ |\phi\rangle + E_{\mathrm{Hxc}}^{\mathrm{sr},\mu}[n_\phi] \right\}}$$



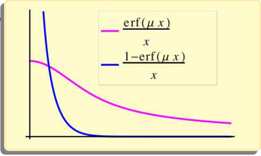
Range-Separated context

Observation

- DFT : good at short range
- WF methods : suitable at long range

The idea **Mix the two** :

$$\boxed{E^{\mathrm{RSH}} = \min_{\phi} \left\{ \langle \phi | \, \hat{T} + \hat{V}_{ne} + \hat{W}_{ee}^{\mathrm{lr},\mu} \, | \phi \rangle + E_{\mathrm{Hxc}}^{\mathrm{sr},\mu}[n_{\phi}] \right\}}$$



where ϕ_0 via a Euler-Lagrange equation

with : $\hat{H}_0 = \hat{T} + \hat{V}_{ne} + \hat{V}_{\mathrm{Hx,HF}}^{\mathrm{lr}} + \hat{V}_{\mathrm{Hxc}}^{\mathrm{sr}}$

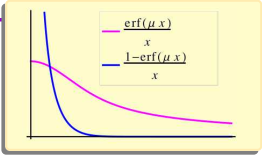
Range-Separated context

Observation

- DFT : good at short range
- WF methods : suitable at long range

The idea **Mix the two** :

$$E^{\mathrm{RSH}} = \min_{\phi} \left\{ \langle \phi | \, \hat{T} + \hat{V}_{ne} + \hat{W}_{ee}^{\mathrm{lr},\mu} \, | \phi \rangle + E_{\mathrm{Hxc}}^{\mathrm{sr},\mu}[n_\phi] \right\}$$



where ϕ_0 via a Euler-Lagrange equation

with : $\hat{H}_0 = \hat{T} + \hat{V}_{ne} + \hat{V}_{\mathrm{Hx,HF}}^{\mathrm{lr}} + \hat{V}_{\mathrm{Hxc}}^{\mathrm{sr}}$

Long-range correlation

$$E = E^{\mathrm{RSH}} + E_c^{\mathrm{lr}} = \langle \mathbf{d}^{(0)} \mathbf{f}^{\mathrm{lr}} \rangle - \langle \mathbf{d}^{(0)} \mathbf{DC}^{\mathrm{lr}} \rangle + E_{\mathrm{Hxc}}^{\mathrm{sr},\mu}[n_{\phi_0}] + E_c^{\mathrm{lr}}$$
$$= \langle \mathbf{d}^{(0)} \left(\mathbf{f}^{\mathrm{lr}} + \mathbf{f}^{\mathrm{sr}} \right) \rangle - \langle \mathbf{d}^{(0)} \mathbf{DC} \rangle + E_c^{\mathrm{lr}}$$

Random Phase Approximation

Random Phase Approximation

RPA

- especially useful for dispersion energies (vdW forces)
- many different *flavors*, the simplest being :

$$\boxed{\begin{cases} E_c^{\text{dRPA-I}} & = \langle \mathbf{K}\mathbf{T} \rangle \\ 0 & = 2\left(\mathbf{K} + \mathbf{K}\mathbf{T} + \mathbf{T}\mathbf{K} + \mathbf{T}\mathbf{K}\mathbf{T}\right) + \left(\boldsymbol{\epsilon}\mathbf{T} + \mathbf{T}\boldsymbol{\epsilon}\right) \end{cases}}$$

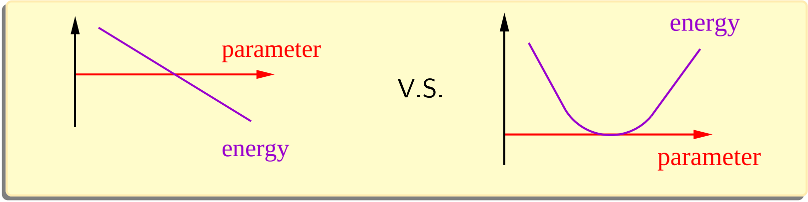
Random Phase Approximation

RPA

- especially useful for dispersion energies (vdW forces)
- many different *flavors*, the simplest being :

$$\begin{cases} E_c^{\text{dRPA-I}} &= \langle \mathbf{KT} \rangle \\ 0 &= 2\left(\mathbf{K} + \mathbf{KT} + \mathbf{TK} + \mathbf{TKT}\right) + \left(\epsilon\mathbf{T} + \mathbf{T}\epsilon\right) \end{cases}$$

A non-variational method



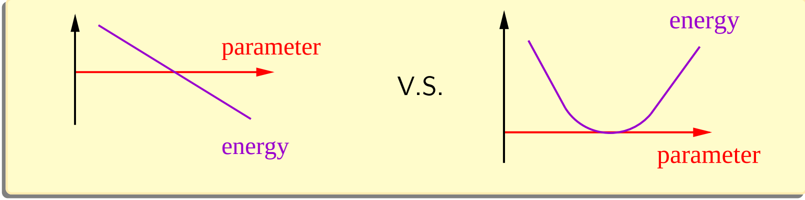
Random Phase Approximation

RPA

- especially useful for dispersion energies (vdW forces)
- many different *flavors*, the simplest being :

$$\boxed{\begin{cases} E_c^{\text{dRPA-I}} &= \langle \mathbf{KT} \rangle \\ 0 &= 2\left(\mathbf{K} + \mathbf{KT} + \mathbf{TK} + \mathbf{TKT}\right) + \left(\epsilon\mathbf{T} + \mathbf{T}\epsilon\right) \end{cases}} \quad = \mathbf{R}(\mathbf{T})$$

A non-variational method



Random Phase Approximation

RPA

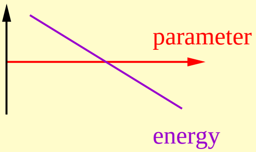
- especially useful for dispersion energies (vdW forces)
- many different *flavors*, the simplest being :

$$\begin{cases} E_c^{\text{dRPA-I}} & = \langle \mathbf{KT} \rangle \\ 0 & = 2\left(\mathbf{K} + \mathbf{KT} + \mathbf{TK} + \mathbf{TKT}\right) + \left(\epsilon\mathbf{T} + \mathbf{T}\epsilon\right) \end{cases}$$

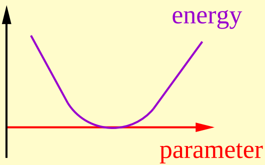
$= \langle \mathbf{IT} \rangle$

$= \mathbf{R}(\mathbf{T})$

A non-variational method



V.S.



A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

HF gradient

$\frac{\partial E_{HF}}{\partial \kappa} = \langle HF | \frac{\partial \hat{H}}{\partial \kappa} | HF \rangle$

A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

HF gradient

$$\hat{H} = \underline{S^{-1}_{\mu\alpha} h_{\alpha\beta} S^{-1}_{\nu\beta}} \chi^\dagger_\mu \chi_\nu + \frac{1}{2} \underline{S^{-1}_{\mu\alpha} S^{-1}_{\nu\beta} (\alpha\beta|\gamma\delta) S^{-1}_{\gamma\sigma} S^{-1}_{\delta\lambda}} \chi^\dagger_\mu \chi^\dagger_\nu \chi_\sigma \chi_\lambda$$

$$\frac{\partial E_{HF}}{\partial \kappa} = \langle HF | \frac{\partial \hat{H}}{\partial \kappa} | HF \rangle$$

A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

HF gradient

$$\hat{H} = \underline{S^{-1}_{\mu\alpha} h_{\alpha\beta} S^{-1}_{\nu\beta}} \chi^{\dagger}_{\mu} \chi_{\nu} + \frac{1}{2} \underline{S^{-1}_{\mu\alpha} S^{-1}_{\nu\beta} (\alpha\beta|\gamma\delta) S^{-1}_{\gamma\sigma} S^{-1}_{\delta\lambda}} \chi^{\dagger}_{\mu} \chi^{\dagger}_{\nu} \chi_{\sigma} \chi_{\lambda}$$

$$\begin{aligned} \frac{\partial E_{HF}}{\partial \kappa} &= \langle HF | \frac{\partial \hat{H}}{\partial \kappa} | HF \rangle \\ &= \underline{\delta h_{\alpha\beta}} P_{\alpha\beta} + \frac{1}{2} \underline{\delta(\mu\lambda|\nu\sigma)} (P_{\mu\lambda} P_{\nu\sigma} - P_{\mu\sigma} P_{\nu\lambda}) + \underline{\delta S_{\mu\nu}} S^{-1}_{\nu\lambda} F_{\lambda\sigma} P_{\sigma\mu} \end{aligned}$$

- HF gradient *straightforwardly* found by the Hellman-Feynman theorem

A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

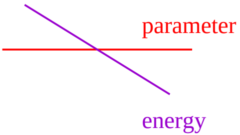
HF gradient

$$\hat{H}=S^{-1}_{\mu\alpha} h_{\alpha\beta} S^{-1}_{\nu\beta} \chi^\dagger_\mu \chi_\nu + \frac{1}{2} S^{-1}_{\mu\alpha} S^{-1}_{\nu\beta} (\alpha\beta|\gamma\delta) S^{-1}_{\gamma\sigma} S^{-1}_{\delta\lambda} \chi^\dagger_\mu \chi^\dagger_\nu \chi_\sigma \chi_\lambda$$

$$\begin{aligned} \frac{\partial E_{HF}}{\partial \kappa} &= \langle HF | \frac{\partial \hat{H}}{\partial \kappa} | HF \rangle \\ &= \delta h_{\alpha\beta} P_{\alpha\beta} + \frac{1}{2} \delta(\mu\lambda|\nu\sigma)(P_{\mu\lambda}P_{\nu\sigma} - P_{\mu\sigma}P_{\nu\lambda}) + \delta S_{\mu\nu} S^{-1}_{\nu\lambda} F_{\lambda\sigma} P_{\sigma\mu} \end{aligned}$$

- HF gradient *straightforwardly* found by the Hellman-Feynman theorem

For non-variational wavefunctions... (RPA)



A Simple Gradient : HF

Hellmann-Feynman theorem

- valid for **variational** wavefunctions

$$\boxed{\frac{\partial E}{\partial \kappa} = \langle \Psi | \frac{\partial \hat{H}}{\partial \kappa} | \Psi \rangle}$$

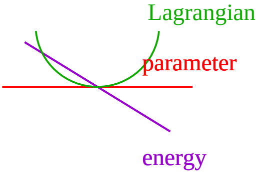
HF gradient

$$\hat{H} = S^{-1}_{\mu\alpha} h_{\alpha\beta} S^{-1}_{\nu\beta} \chi^\dagger_\mu \chi_\nu + \frac{1}{2} S^{-1}_{\mu\alpha} S^{-1}_{\nu\beta} (\alpha\beta|\gamma\delta) S^{-1}_{\gamma\sigma} S^{-1}_{\delta\lambda} \chi^\dagger_\mu \chi^\dagger_\nu \chi_\sigma \chi_\lambda$$

$$\begin{aligned} \frac{\partial E_{HF}}{\partial \kappa} &= \langle HF | \frac{\partial \hat{H}}{\partial \kappa} | HF \rangle \\ &= \delta h_{\alpha\beta} P_{\alpha\beta} + \frac{1}{2} \delta(\mu\lambda|\nu\sigma)(P_{\mu\lambda} P_{\nu\sigma} - P_{\mu\sigma} P_{\nu\lambda}) + \delta S_{\mu\nu} S^{-1}_{\nu\lambda} F_{\lambda\sigma} P_{\sigma\mu} \end{aligned}$$

- HF gradient *straightforwardly* found by the Hellman-Feynman theorem

For non-variational wavefunctions. . . (RPA)



Work with an alternative object that **is** variational

Lagrangian Framework

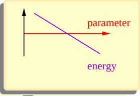
Lagrangian Framework

Remember : for a non-variational method :

Lagrangian Framework

Remember : for a non-variational method : energy $E(\mathbf{P})$

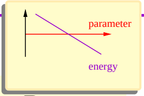
Lagrangian Framework



Remember : for a non-variational method :

energy	rules for $\mathbf{P}$
$E(\mathbf{P})$	$\mathbf{R}(\mathbf{P}) = 0$

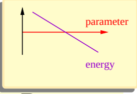
Lagrangian Framework



Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$

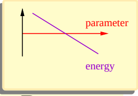
Lagrangian Framework



Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$
- stationary conditions for $\mathcal{L}$

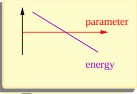
Lagrangian Framework



Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$
- stationary conditions for $\mathcal{L}$ $\frac{\partial \mathcal{L}(\mathbf{P}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{P}) = 0$

Lagrangian Framework

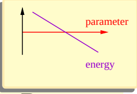


Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$
- stationary conditions for $\mathcal{L}$ $\frac{\partial \mathcal{L}(\mathbf{P},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{P}) = 0$

$$\boxed{\frac{\partial \mathcal{L}(\mathbf{P},\boldsymbol{\lambda})}{\partial \mathbf{P}} = \frac{\partial E(\mathbf{P})}{\partial \mathbf{P}} + \langle \boldsymbol{\lambda} \frac{\partial \mathbf{R}(\mathbf{P})}{\partial \mathbf{P}} \rangle = 0}$$

Lagrangian Framework



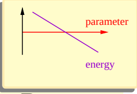
Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$
- stationary conditions for $\mathcal{L}$ $\frac{\partial \mathcal{L}(\mathbf{P},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{P}) = 0$

$$\boxed{\frac{\partial \mathcal{L}(\mathbf{P},\boldsymbol{\lambda})}{\partial \mathbf{P}} = \frac{\partial E(\mathbf{P})}{\partial \mathbf{P}} + \langle \boldsymbol{\lambda} \frac{\partial \mathbf{R}(\mathbf{P})}{\partial \mathbf{P}} \rangle = 0}$$

non zero !

Lagrangian Framework



Remember : for a non-variational method : energy $E(\mathbf{P})$ rules for $\mathbf{P}$ $\mathbf{R}(\mathbf{P}) = 0$

- introduce the Lagrangian $\mathcal{L}(\mathbf{P}, \boldsymbol{\lambda}) = E(\mathbf{P}) + \langle \boldsymbol{\lambda} \mathbf{R}(\mathbf{P}) \rangle \quad (= E(\mathbf{P}))$
- stationary conditions for $\mathcal{L}$ $\frac{\partial \mathcal{L}(\mathbf{P}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{P}) = 0$

$$\boxed{\frac{\partial \mathcal{L}(\mathbf{P}, \boldsymbol{\lambda})}{\partial \mathbf{P}} = \frac{\partial E(\mathbf{P})}{\partial \mathbf{P}} + \langle \boldsymbol{\lambda} \frac{\partial \mathbf{R}(\mathbf{P})}{\partial \mathbf{P}} \rangle = 0}$$

non zero !

RPA case

- non-variational parameters : $E(\mathbf{T}, \mathbf{C})$
- three constraints : $\mathbf{R}(\mathbf{T}, \mathbf{C}) = 0$, $(\mathbf{f})_{ai} = 0$, $\left(\mathbf{C}^\dagger \mathbf{S} \mathbf{C} - \mathbf{1}\right) = 0$
- three Lagrangian multipliers : $\boldsymbol{\lambda}$, $\mathbf{z}$ and $\mathbf{x}$

$$\mathcal{L}(\mathbf{T}, \boldsymbol{\lambda}, \mathbf{C}, \mathbf{z}, \mathbf{x}) = \langle \mathbf{I} \mathbf{T} \rangle + \langle \boldsymbol{\lambda} \mathbf{R} \rangle + \langle \mathbf{z} \mathbf{f} \rangle + \langle \mathbf{x} \left(\mathbf{C}^\dagger \mathbf{S} \mathbf{C} - \mathbf{1}\right) \rangle$$

RPA Gradient

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{I}\underline{\mathbf{T}}\rangle+\langle\boldsymbol{\lambda}\underline{\mathbf{R}}\rangle+\langle\mathbf{z}\mathbf{f}\rangle+\langle\mathbf{x}(\mathbf{C}^{\dagger}\mathbf{S}\mathbf{C}-\mathbf{1})\rangle$$

RPA Gradient

Computing $\boldsymbol{\lambda}$

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x}) = \langle \mathbf{I}\underline{\mathbf{T}} \rangle + \langle \boldsymbol{\lambda}\underline{\mathbf{R}} \rangle + \langle \mathbf{z}\mathbf{f} \rangle + \langle \mathbf{x}(\mathbf{C}^{\dagger}\mathbf{S}\mathbf{C} - \mathbf{1}) \rangle$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}} = \mathbf{I} + \boldsymbol{\lambda}\frac{\partial \mathbf{R}}{\partial \mathbf{T}} = \mathbf{0}$$

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{IT}\rangle+\langle \boldsymbol{\lambda}\mathbf{R}\rangle+\langle \mathbf{zf}\rangle+\langle \mathbf{x}(\mathbf{C}^{\dagger}\mathbf{SC}-\mathbf{1})\rangle$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}} = \mathbf{I} + \boldsymbol{\lambda}\frac{\partial \mathbf{R}}{\partial \mathbf{T}} = \mathbf{0}$$

Computing x and z

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{KM}\rangle+\langle \mathbf{K'N}\rangle+\langle \mathbf{JO}\rangle+\underbrace{\langle \boldsymbol{\epsilon}\mathbf{T}\rangle+\langle \mathbf{T}\boldsymbol{\epsilon}\rangle}_{\langle \mathbf{d}^{(2)}\mathbf{f}\rangle}+\langle \mathbf{zf}\rangle+\langle \mathbf{x}(\mathbf{C}^{\dagger}\mathbf{SC}-\mathbf{1})\rangle$$

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{IT}\rangle+\langle \boldsymbol{\lambda}\mathbf{R}\rangle+\langle \mathbf{zf}\rangle+\langle \mathbf{x}(\mathbf{C}^{\dagger}\mathbf{SC}-\mathbf{1})\rangle$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}} = \mathbf{I} + \boldsymbol{\lambda}\frac{\partial \mathbf{R}}{\partial \mathbf{T}} = \mathbf{0}$$

$$0=2(\mathbf{K}+\mathbf{KT}+\mathbf{TK}+\mathbf{TKT})+(\boldsymbol{\epsilon}\mathbf{T}+\mathbf{T}\boldsymbol{\epsilon})$$

Computing x and z

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{KM}\rangle+\langle \mathbf{K}'\mathbf{N}\rangle+\langle \mathbf{JO}\rangle+\underbrace{\langle \boldsymbol{\epsilon}\mathbf{T}\rangle+\langle \mathbf{T}\boldsymbol{\epsilon}\rangle}_{\langle \mathbf{d}^{(2)}\mathbf{f}\rangle}+\langle \mathbf{zf}\rangle+\langle \mathbf{x}(\mathbf{C}^{\dagger}\mathbf{SC}-\mathbf{1})\rangle$$

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle\mathbf{IT}\rangle+\langle\boldsymbol{\lambda}\mathbf{R}\rangle+\langle\mathbf{zf}\rangle+\langle\mathbf{x}(\mathbf{C}^\dagger\mathbf{SC}-\mathbf{1})\rangle$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{T}}=\mathbf{I}+\boldsymbol{\lambda}\frac{\partial\mathbf{R}}{\partial\mathbf{T}}=\mathbf{0}$$

$$0=2(\mathbf{K}+\mathbf{KT}+\mathbf{TK}+\mathbf{TKT})+(\boldsymbol{\epsilon}\mathbf{T}+\mathbf{T}\boldsymbol{\epsilon})$$

Computing x and z

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle\mathbf{KM}\rangle+\langle\mathbf{K}'\mathbf{N}\rangle+\langle\mathbf{JO}\rangle+\underbrace{\langle\boldsymbol{\epsilon}\mathbf{T}\rangle+\langle\mathbf{T}\boldsymbol{\epsilon}\rangle}_{\langle\mathbf{d}^{(2)}\mathbf{f}\rangle}+\langle\mathbf{zf}\rangle+\langle\mathbf{x}(\mathbf{C}^\dagger\mathbf{SC}-\mathbf{1})\rangle$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{C}}=\cdots=\boldsymbol{\Theta}+\tilde{\boldsymbol{\Theta}}(\mathbf{z})+2\mathbf{x}=\mathbf{0}$$

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\lambda,\mathbf{C},\mathbf{z},\mathbf{x})=\langle\mathbf{IT}\rangle+\langle\lambda\mathbf{R}\rangle+\langle\mathbf{zf}\rangle+\langle\mathbf{x}(\mathbf{C}^\dagger\mathbf{SC}-\mathbf{1})\rangle$$

$$0=2(\mathbf{K}+\mathbf{KT}+\mathbf{TK}+\mathbf{TKT})+(\epsilon\mathbf{T}+\mathbf{T}\epsilon)$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{T}}=\mathbf{I}+\lambda\frac{\partial\mathbf{R}}{\partial\mathbf{T}}=\mathbf{0}$$

Computing x and z

$$\mathcal{L}(\mathbf{T},\lambda,\mathbf{C},\mathbf{z},\mathbf{x})=\langle\mathbf{KM}\rangle+\langle\mathbf{K}'\mathbf{N}\rangle+\langle\mathbf{JO}\rangle+\underbrace{\langle\epsilon\mathbf{T}\rangle+\langle\mathbf{T}\epsilon\rangle}_{\langle\mathbf{d}^{(2)}\mathbf{f}\rangle}+\langle\mathbf{zf}\rangle+\langle\mathbf{x}(\mathbf{C}^\dagger\mathbf{SC}-\mathbf{1})\rangle$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{C}}=\cdots=\boldsymbol{\Theta}+\tilde{\boldsymbol{\Theta}}(\mathbf{z})+2\mathbf{x}=\mathbf{0}$$

- yields a set of two equations, solved for **z** and **x** :

$$\begin{cases}\left(\boldsymbol{\Theta}-\boldsymbol{\Theta}^\dagger+\mathbf{fz}-\mathbf{zf}+4\mathbf{g}(\mathbf{z})+4\mathbf{W}[\mathbf{z}]\right)_{ai} & =0\\ (1+\tau_{pq})\left(\boldsymbol{\Theta}+\tilde{\boldsymbol{\Theta}}(\mathbf{z})\right)_{pq} & =-4(\mathbf{x})_{pq}\end{cases}$$

RPA Gradient

Computing λ

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle \mathbf{I}\underline{\mathbf{T}}\rangle+\langle\boldsymbol{\lambda}\underline{\mathbf{R}}\rangle+\langle\mathbf{zf}\rangle+\langle\mathbf{x}(\mathbf{C}^\dagger\mathbf{SC}-\mathbf{1})\rangle$$

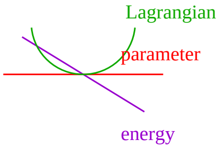
$$\frac{\partial\mathcal{L}}{\partial\mathbf{T}}=\mathbf{I}+\boldsymbol{\lambda}\frac{\partial\mathbf{R}}{\partial\mathbf{T}}=\mathbf{0}$$

$$0=2(\mathbf{K}+\mathbf{KT}+\mathbf{TK}+\mathbf{TKT})+(\boldsymbol{\epsilon}\mathbf{T}+\mathbf{T}\boldsymbol{\epsilon})$$

Computing x and z

$$\mathcal{L}(\mathbf{T},\boldsymbol{\lambda},\mathbf{C},\mathbf{z},\mathbf{x})=\langle\underline{\mathbf{K}}\mathbf{M}\rangle+\langle\underline{\mathbf{K}'}\mathbf{N}\rangle+\langle\underline{\mathbf{J}}\mathbf{O}\rangle+\underbrace{\langle\boldsymbol{\epsilon}\mathbf{T}\rangle+\langle\mathbf{T}\boldsymbol{\epsilon}\rangle}_{\langle\mathbf{d}^{(2)}\underline{\mathbf{f}}\rangle}+\langle\mathbf{z}\underline{\mathbf{f}}\rangle+\langle\mathbf{x}(\underline{\mathbf{C}^\dagger}\mathbf{S}\underline{\mathbf{C}}-\mathbf{1})\rangle$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{C}}=\cdots=\boldsymbol{\Theta}+\tilde{\boldsymbol{\Theta}}(\mathbf{z})+2\mathbf{x}=\mathbf{0}$$



- yields a set of two equations, solved for **z** and **x** :

$$\begin{cases} \left(\boldsymbol{\Theta}-\boldsymbol{\Theta}^\dagger+\mathbf{fz}-\mathbf{zf}+4\mathbf{g}(\mathbf{z})+4\mathbf{W}[\mathbf{z}]\right)_{ai} & =0 \\ (1+\tau_{pq})\left(\boldsymbol{\Theta}+\tilde{\boldsymbol{\Theta}}(\mathbf{z})\right)_{pq} & =-4(\mathbf{x})_{pq} \end{cases}$$

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of $\mathcal{L}$**

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of $\mathcal{L}$**

$$(E_c^{\mathrm{RPA}})^x = \mathcal{L}^x = \left\langle \left(\mathbf{z} + \mathbf{d}^{(2)} \right) \mathbf{f}^x \right\rangle + \langle \mathbf{M}\mathbf{K}^x \rangle + \langle \mathbf{N}\mathbf{K}'^x \rangle + \langle \mathbf{O}\mathbf{J}^x \rangle + \langle \mathbf{x}\mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \rangle$$

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of $\mathcal{L}$**

$$(E_c^{\mathrm{RPA}})^x = \mathcal{L}^x = \left\langle \left(\mathbf{z} + \mathbf{d}^{(2)} \right) \mathbf{f}^x \right\rangle + \left\langle \mathbf{M}\mathbf{K}^x \right\rangle + \left\langle \mathbf{N}\mathbf{K}'^x \right\rangle + \left\langle \mathbf{O}\mathbf{J}^x \right\rangle + \left\langle \mathbf{x}\mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \right\rangle$$

$$(E^{\mathrm{RSH}})^x = \left\langle \mathbf{d}^{(0)} \mathbf{f}^x \right\rangle - \left\langle \mathbf{d}^{(0)} \mathbf{D}\mathbf{C}^x \right\rangle - 2 \left\langle \epsilon \mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \right\rangle$$

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of** $\mathcal{L}$

$$(E_c^{\mathrm{RPA}})^x = \mathcal{L}^x = \left\langle \left(\mathbf{z} + \mathbf{d}^{(2)}\right) \mathbf{f}^x \right\rangle + \langle \mathbf{M}\mathbf{K}^x \rangle + \langle \mathbf{N}\mathbf{K}'^x \rangle + \langle \mathbf{O}\mathbf{J}^x \rangle + \langle \mathbf{x}\mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \rangle$$

$$(E^{\mathrm{RSH}})^x = \langle \mathbf{d}^{(0)} \mathbf{f}^x \rangle - \langle \mathbf{d}^{(0)} \mathbf{D}\mathbf{C}^x \rangle - 2 \langle \epsilon \mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \rangle$$

Careful derivation of $(\mathbf{f}^x - \mathbf{D}\mathbf{C}^x)$

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of** $\mathcal{L}$

$$(E_c^{\mathrm{RPA}})^x = \mathcal{L}^x = \left\langle \left(\mathbf{z} + \mathbf{d}^{(2)}\right) \mathbf{f}^x \right\rangle + \left\langle \mathbf{M}\mathbf{K}^x \right\rangle + \left\langle \mathbf{N}\mathbf{K}'^x \right\rangle + \left\langle \mathbf{O}\mathbf{J}^x \right\rangle + \left\langle \mathbf{x}\mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \right\rangle$$

$$(E^{\mathrm{RSH}})^x = \left\langle \mathbf{d}^{(0)}\mathbf{f}^x \right\rangle - \left\langle \mathbf{d}^{(0)}\mathbf{D}\mathbf{C}^x \right\rangle - 2\left\langle \epsilon \mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \right\rangle$$

Careful derivation of $(\mathbf{f}^x - \mathbf{D}\mathbf{C}^x)$

- long-range part
$$\left(\mathbf{d}^{(0)} + \mathbf{d}^{(2)} + \mathbf{z}\right)\mathbf{h}^x + \left(\frac{1}{2}\mathbf{d}^{(0)} + \mathbf{d}^{(2)} + \mathbf{z}\right)\mathbf{d}^{(0)}\mathrm{int}^{\mathrm{LR}x}$$

RPA Gradient

Once multipliers are known gradients obtained from **derivatives of** $\mathcal{L}$

$$(E_c^{\mathrm{RPA}})^x = \mathcal{L}^x = \left\langle \left(\mathbf{z} + \mathbf{d}^{(2)} \right) \mathbf{f}^x \right\rangle + \langle \mathbf{M}\mathbf{K}^x \rangle + \langle \mathbf{N}\mathbf{K}'^x \rangle + \langle \mathbf{O}\mathbf{J}^x \rangle + \langle \mathbf{x}\mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \rangle$$

$$(E^{\mathrm{RSH}})^x = \langle \mathbf{d}^{(0)} \mathbf{f}^x \rangle - \langle \mathbf{d}^{(0)} \mathbf{D}\mathbf{C}^x \rangle - 2 \langle \epsilon \mathbf{C}^\dagger \mathbf{S}^x \mathbf{C} \rangle$$

Careful derivation of $(\mathbf{f}^x - \mathbf{D}\mathbf{C}^x)$

- long-range part
  $$\left(\mathbf{d}^{(0)} + \mathbf{d}^{(2)} + \mathbf{z} \right) \mathbf{h}^x + \left(\frac{1}{2} \mathbf{d}^{(0)} + \mathbf{d}^{(2)} + \mathbf{z} \right) \mathbf{d}^{(0)} \mathrm{int}^{\mathrm{LR}x}$$

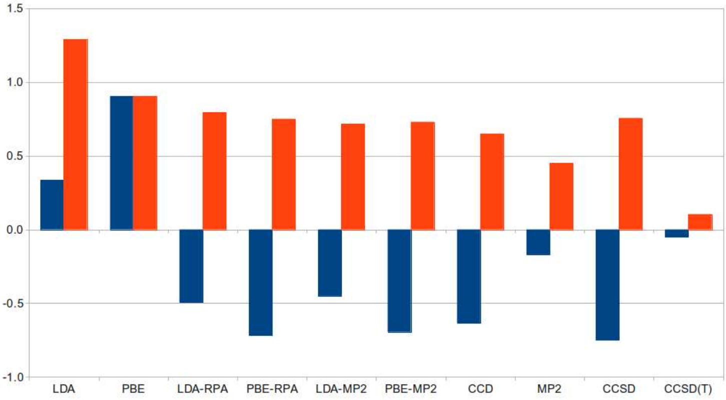
- short-range part is actually $E_{\mathrm{Hxc}}^{\mathrm{SR}}[\tilde{\rho}]^x$:
  $$E_{\mathrm{Hxc}}^{\mathrm{SR}}[\tilde{\rho}]^x = E_{\mathrm{Hxc}}^{\mathrm{SR}}[\rho^{(0)}]^x + \int \frac{\partial E_{\mathrm{Hxc}}^{\mathrm{SR}}[\rho^{(0)}]}{\partial \rho^{(0)}} (\tilde{\rho} - \rho^{(0)})$$

Preliminary Results

Test the implementation

- perfect agreement with numerical calculations
- timings are as expected (same behavior as MP2)

Geometry Optimization



Conclusion & Outlook

The **Lagrangian framework** has successfully been applied to derive **RPA gradients**

Outlook

- Implementation has been done in Molpro
  (useful parallel with **MP2 gradients**)
- Gradients of *mixed* RPA energies need further coding
  (e.g. Szabo-Ostlund variant)
- Geometry optimization seem to work
- Extension to **density fitting** seems straighforward