Minimum Energy Conical Intersection Optimization

안희선

문서 버전 1.0

**1. 개요**

본 프로그램은 penalty constrained optimization 방법[JPCB, 112, 405 (2008)]으로 특정 분자의 두 전자 상태들 간에 Minimum Energy Conical Intersections (MECIs) 의 구조를 최적화 한다. 필요한 두 전자 상태들의 potential energy와 energy gradient를 계산하기 위해 Multi-reference 방법 중 Complete Active Space Self Consistent Field (CASSCF) 또는 Multi Reference Configuration Interaction (MRCI)를 사용한다. 기존에 발표된 H2S, ethylene의 MECIs와 작성된 프로그램을 사용하여 계산한 MECIs를 비교 한다.

**2. 구동 환경**

* 운영체제: 리눅스, 테스트: SUSE Linux Enterprise Server (SLES) 10
* Potential energy 계산: 외부 프로그램 MOLPRO 2015
* Python 2.7
* Third party library: numpy (Anaconda 2 사용)
* 구성 파일: meciopt (main), molprolib.py, optimize.py, penaltyfun.py

**3. 개발 목적**

* 기존에 발표된 여러 Penalty functions에 대한 성능 평가 및 비교
* 새로운 페널티 함수의 설계 및 평가
* MRCI 수준의 MECIOpt 계산 및 정확도 평가, 그리고 새로운 시스템에 대한 MECI 찾기
* 본 연구실에서 새로 개발한 non-adiabatic coupling을 추정하는 수식 적용 및 장단점 파악

**4. 사용 방법**

**4-1. help 옵션**

기본적으로 main script인 meciopt 모듈에서 리눅스 Usage 형태의 옵션을 처리하도록 작성되어 있다. 아래와 같이 --help 옵션을 볼 수 있다.

Prompt> meciopt -h

Usage: meciopt inputfile [options]...

Example: meciopt -h # option help

meciopt -t # print inputfile templet

meciopt -t > inputfile # save inputfile templet

meciopt inputfile # run inputfile in default workdir

meciopt inputfile -w temp # run inputfile in workdir is temp/

Options:

--version show program's version number and exit

-h, --help show this help message and exit

-t, --templet print templet of meciopt input. If this option is

activated, all other options are ignored.

-w WORKDIR, --workdir=WORKDIR

set work dir. If the specified WORKDIR does not exist,

it is created.If it already exists, the existing

workdir is changed toWORKDIR[\_1[\_2...]] and a new

WORKDIR is created.

-m MOLPRO\_INPUTFILE, --molprorun=MOLPRO\_INPUTFILE

run molpro calculation

-s MECIOPTOUT, --summary=MECIOPTOUT

print geometries in g09 format and summary table

**4-2. input 템플릿**

meciopt를 실행하기 위해서는 형식화된 input이 필요하다. 프로그램 내부에 기본적인 input 템플릿을 출력하도록 작성하였다. meciopt의 input은 python variable을 정의하는 방식과 동일하며 주석도 python 방식으로 작성할 수 있다. 자세한 input 옵션에 대한 설명은 input 옵션 부분을 참고하라.

Prompt> meciopt -t

# MECIOPT INPUT TEMPLET

# This input file is run in meciopt script via the 'exec' command,

# so the variables defined in this input file are equivalent

# the variables in meciopt.

# The option values are case-sensitive.

# ---------------------------------------------------------------------

# Optimization options (str): 'PF'

OPTMTD = 'PF'

PFUN = 'jpcb2008'

LINESEARCH = 'HALFSTEP'

PF\_ALPHA = 3.50

PF\_BETA = 0.02

PF\_MAXCYC = 30

THREH\_STEP = 1.0e-6

THRESH\_GRAD = 0.005

THRESH\_EGAP = 0.001

NEWG = 'OLDG' # IDENTITY, OLDG, NEWCALC

# update method of G matrix (str): 'BFGS', 'MS'

UPDATEG = 'BFGS'

# Optimization coordinates (str): 'XYZ', 'Z-matrix'

OPTCOORD = 'XYZ'

# ---------------------------------------------------------------------

# molpro single point energy input (str)

INPSTR = '''\*\*\*,h2s

memory,128,m

basis=6-31g\*

symmetry,x

geometry={

S

Q 1 LR

H 2 HSR 1 GM

H 2 HSR 1 GMM 3 DD}

LR = 1.9989 bohr

SR = 4.2904 bohr

HSR = SR/2

GM = 90.0 degree

GMM = 180 - GM

DD = 180.0 degree

{hf;occ,7,2;wf,18,1,0}

{multi;

closed,4,1;

occ,10,2;

wf,18,2,2;state,2;weight,0.55,0.45;

wf,18,2,0;state,2;weight,0.55,0.45;

wf,18,1,0;state,1;weight,0.55}

---

'''

# ---------------------------------------------------------------------

# IDXUPP = index of Upper-state, IDXLOW = index of Lower-state (int)

IDXLOW = 3

IDXUPP = 4

# numerical difference delta R in bohr (float)

DIFDR = 0.01

# print option (int)

# 1 : print until vectors (ex. F, R ...)

# 2 : print until matries (ex. G, P ...)

# 3 : print until system massage

PRINT = 1

**4-3. 계산 시작**

먼저 Input templet 을 파일로 저장하고 적절히 수정한다.

Prompt> meciopt –t > h2s\_meciopt.inp

계산을 시작하려면 아래와 같이 input 파일을 인수로 넘기면 된다.

Prompt> meciopt h2s\_meciopt.inp

작업 디렉토리를 따로 지정하지 않으면 input 파일에서 확장자를 제외한 이름으로h2s\_meciopt/ 디렉토리가 생성되며 작업 디렉토리에 각 iteration step에 해당되는 에너지 및 gradient 계산의 MOLPRO input, output이 저장되며 linesearch에 관련된 input, output이 저장된다.

meciopt에 관련된 output은 화면에 출력되기 때문에 파일에 작성하기 위해 아래와 같이 redirection 형식으로 계산을 지정하며 background job으로 수행하도록 &를 추가한다.

Prompt> meciopt h2s\_meciopt.inp > h2s\_meciopt.out &

**4-4. meciopt output 요약**

meciopt 계산이 완료되면 매 step에서 구조가 변화되는 과정의 데이터를 GaussView로 확인하도록 형식화된 요약 output을 만들고, 중요 스칼라들을 표 형식으로 정리해주는 기능이 추가되어 있다.

Prompt> meciopt –-summary=h2s\_meciopt.out > h2s\_meciopt\_summary.out

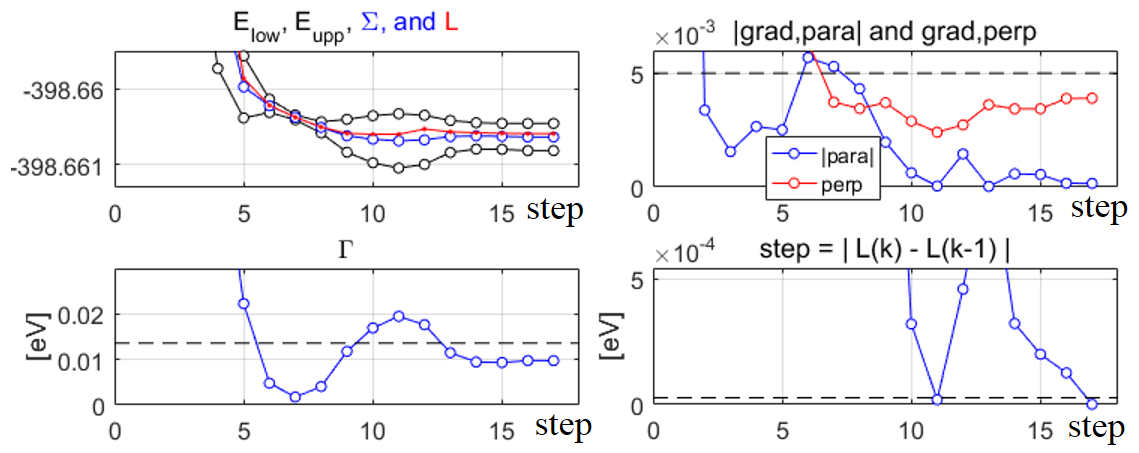
**5. meciopt input 옵션**

|  |  |  |
| --- | --- | --- |
| 옵션명 | 가능한 값 | 설명 |
| OPTMTD | (str) PF | 최적화 방법 설정  현재 페널티 함수 방법만 존재 |
| PFUN | (str) jpcb2008, jpcl2011,  cej2004, gamma | 페널티 함수 선택 |
| LINESEARCH | (str) HALFSTEP, ARMIJO | Line search 방법 선택 |
| PF\_ALPHA  PF\_BETA | (float) [default] | 페널티 함수의 α, β 초기값 선택  선택하지 않으면 각 페널티 함수의 디폴트 값이 자동 설정 |
| PF\_MAXCYC | (int) | Main iteration의 회수 지정 |
| THRESH\_STEP  THRESH\_GRAD  THRESH\_EGAP | (float) | Convergence threshold를 설정 |
| NEWG | (str) IDENTITY, OLDG,  NEWCALC | Iteration이 반복될 때, inverse Hessian G의 갱신 방법에 대한 지정. IDENTITY는 매 iteration에서 단위 행렬 사용. OLDG 사용 추천 |
| UPDATEG | (str) BFGS, MS | Inverse Hessian G의 계산 방법 지정 |
| OPTCOORD | (str) XYZ, Z-matrix | 최적화 좌표 시스템 설정.  현재 XYZ 만 구현되어 있음. |
| DIFDR | (float) | 외부 프로그램 MOLPRO를 사용하여 energy gradient를 계산할 때 numerical difference를 지정. 단위는 Bohr. |
| IDXLOW  IDXUPP | (int) | INPSTR에 지정된 MOLPRO input의 마지막 single point 계산의 전자 상태들 중 고려 대상인 두 전자 상태의 idnex를 지정.  Symmetry를 고려하지 않은 리스트 되는 순서로 선택하기 때문에 신중히 지정 할 것 |
| PRINT | (int) 1-3 | =1 : gradient 같은 vector 정보를 output에 출력  =2 : G 행렬 같은 matrix 정보를 output에 출력  =3 : debug print |
| INPSTR | (str) | MOLPRO의 single point input 문자열을 지정.  meciopt 첫 step 전에 initial input으로 계산된 후, 설정된 XYZ 구조를 바탕으로 모든 계산이 진행된다. 특히 casscf 또는 mrci 계산이 반드시 설정되어야 하며, 마지막으로 지정된 에너지 계산을 바탕으로 meciopt를 진행한다. |

**6. 테스트 계산**

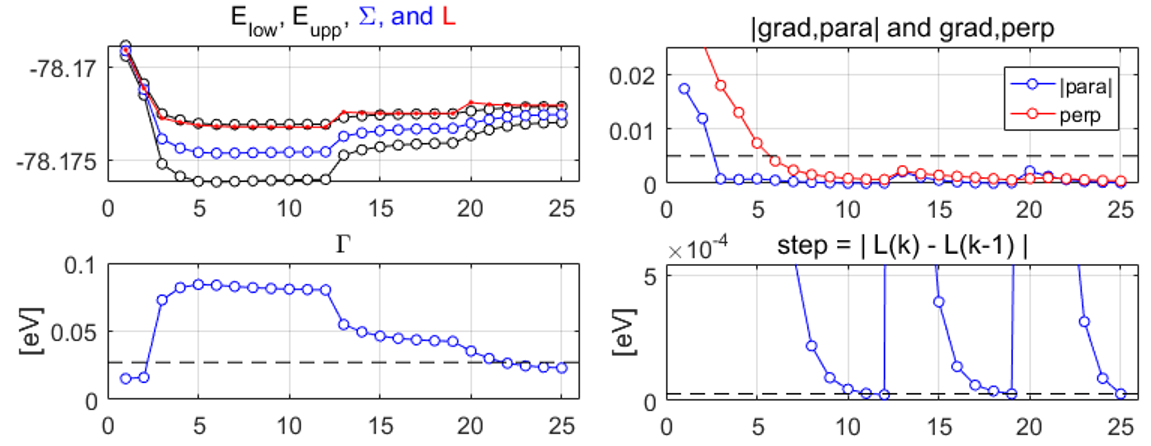
**6-1. H2S, MECI=11A”-21A”**

~test/test01\_h2s/ 에는 H2S 분자에 대한 11A”-21A” Conical Intersection 최적화 결과가 있다. 아래 그림과 같이 본 테스트 계산은 step=17에 수렴이 완료되며, 모든 수렴 조건이 threshold보다 낮아진 것을 확인 할 수 있다.



**6-2. Ethylene, MECI=Ethylidene (11A-21A)**

~test/test02\_ethylene/ 에는 Ethylene 분자에 대한 11A-21A Conical Intersection (Ethylidene 구조) 최적화 결과가 있다. H2S와 마찬가지로 아래 그림과 같이 본 테스트 계산은 step=25에 수렴이 완료되며, 모든 수렴 조건이 만족되었음을 확인할 수 있다.



**7. 관련 수식**

**7-1. 목적 함수**

특정 구조 에서 두 들뜬 전자 상태의 에너지를 , 라 할 때 최적화하는 목적 함수 은 다음과 같다.

 (1)

이 때 는 아래와 같다.

 (2)

Penalty function 는 여러 형태로 정의할 수 있다. [JPCB, 112, 405 (2008)]에서 정의한 Penalty function은 다음과 같다.

 (3)

이 때 는 아래와 같다.

 (4)

따라서 목적 함수 을 다시 쓰면 다음과 같다.

 (5)

목적 함수에서 , 는 MECI를 효율적으로 찾도록 적절히 선택되는 Parameter이며 두 에너지 Gap에 관련된 Parameter이다. [JPCB, 112, 405 (2008)]에서 로 설정하며 에너지 Gap을 더 줄이기 위해 의 값을 적절한 방식으로 올려야 한다.

**7-2. 구조 최적화 방법**

Quasi-Newton 알고리즘 중 하나인 Broyden, Fletcher, Goldfarb, and Shanno(BFGS) 방법을 사용하여 최적화 한다. 이 방법을 사용하기 위해 특정 iteration k의 구조에서 목적 함수의 1차 미분이 필요하다. 아래와 같이 식 (5)가 목적 함수일 때 함수적 1차 미분이 정의된다.

 (6)

 (7)

식 (7)과 같이 를 계산하기 위해 가 필요하며 아래와 같다.

 (8)

식 (8)을 계산하기 위해 두 전자 상태의 에너지 gradient가 필요하다.

 (9)

본 프로그램에서 식 (9)를 계산하기 위해 XYZ 좌표 시스템을 사용한다.

Inverse Hessian approximation [J. Nocedal and S. J. Wright, “Numerical Optimization,” Springer-Verlag, New York, 1999.] 은 아래와 같이 계산된다.

 (10)

식 (10)에서 , 은 아래와 같고,

 (11)

 (12)

k=1 에서 를 사용한다. Search direction은 아래와 같이 계산된다.

 (13)

적절한 Line search 방법으로 step length를 계산하면 (k+1) step의 구조가 아래와 같이 갱신된다.

 (14)

**7-3. 페널티 함수**

여러 Penalty function의 효과를 비교 분석하기 위해  페널티 함수 뿐 아니라 다음 두 Penalty function 또한 구현하였다.

 (15)

 (16)

**7-4. 수렴 조건**

수렴 조건은 [JPCB, 112, 405 (2008)]를 바탕으로 GAMESS, QChem 등에서 사용하는 방식을 참고하여 다음과 같은 네 가지를 설정한다.

 (17)

 (18)

 (19)

본 프로그램에서는 각 thresholds를 다음과 같이 설정한다.

 (20)

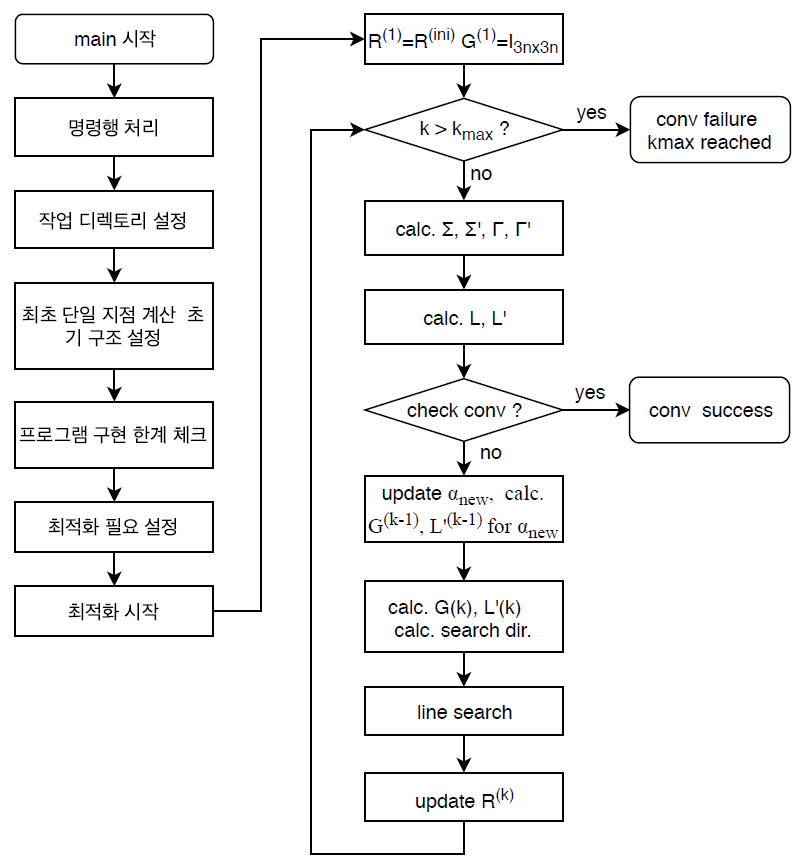
**8. 소스 파일 세부사항**

**8-1. meciopt 모듈**

본 프로그램의 시작 루틴이며, 아래와 같은 것들이 포함되어 있다.

|  |  |
| --- | --- |
| (a) 모듈 변수 |  |
| AU2EV, AU2KCAL,  AU2ANG, AU2CM | 단위 변환 인자 |
| ATOMS | 원자 리스트 |
| INPUT\_PARAMS | 입력 파일 유효성 검사를 위한 유효 옵션 목록 |
| INPUT\_TEMPLET | 입력 파일 템플릿 문자열 |
|  |  |
| (b) 유틸리티 함수 |  |
| get\_nonexistence\_dirname | 작업 디렉토리명 생성 |
| process\_cmdline | 명령행 인수 처리 |
| get\_input\_params | 입력 파일에서 입력 옵션 반입 |
| run\_molpro | 외부 프로그램 molpro를 간단히 실행 |
| print\_summary | meciopt의 결과 파일로부터 매 step의 중요 스칼라 데이터를 표로 출력하고, 각 step의 구조를 gaussian09의 output format으로 출력하여 GaussView로 구조변화를 확인하는 데이터를 출력 |
| make\_input\_and\_run | 이전 molpro 계산의 wfu 파일을 연동하면서 molpro 계산을 하고, 그 결과를 처리하는 객체들을 리턴 |
| check\_implement\_limit | 아직 구현되지 않는 기능에 대한 예외처리 |
|  |  |
| (c) meciopt output 출력 관련 함수들 | |
| msg\_exit | 메시지 출력 후 meciopt 종료 |
| println | 문자열 출력 후 줄 넘기기 |
| print\_section\_title | 섹션 제목 출력 |
| print\_underline\_title | 밑 줄 쳐진 제목 출력 |
| print\_params | 딕셔너리 형태의 파라메터 출력 |
| print\_geometry | 구조 파라메터 출력 |
| print\_energy | 계산된 전자상태들의 에너지 및 여러 에너지 정보 출력 |
| print\_sp\_summary | 단일 지점 계산의 결과 정보를 출력 |
| print\_vectors | 에너지 gradient와 같이 각 원자에 대한 xyz축에 따른 기울기 형태의 데이터를 출력 |
| print\_3n3n\_matrix | Inverse Hessian G 행렬과 같은 3n x 3n 행렬 출력 |
|  |  |
| (d) optimization 함수 |  |
| setup | Optimization을 위한 필요 설정 함수 |
| runopt\_penaltyfunction | Optimization 알고리즘 구현 함수 |
| compute\_G\_bfgs | BFGS방법으로 Inverse Hessian G 계산 함수 |
| compute\_G\_ms | MS방법으로 Inverse Hessian G 계산 함수 |
| calc | 외부 프로그램 Molpro와 연동되는 interface 함수이면서 필요한 energy 및 numerical gradient를 계산 |
|  |  |

(e) FLOW CHART: main → runopt\_penaltyfunction



**8-2. molprolib.py 모듈**

외부 프로그램 MOLPRO 관련 모듈이다. 관련 클래스 및 문자열 처리를 위한 정규 표현식 등이 정의되어 있다.

* class Input: MOLPRO input의 keywords 분석 및 입력된 input의 실행
* class Output: MOLPRO output에서 주요 데이터 parsing
* class InputGenerator: MOLPRO input 파일의 생성, wfu 파일의 sync 처리, single point 계산 및 numerical gradient 계산을 위한 input 문자열 생성

**8-3. optimize.py 모듈**

line search의 step length를 결정하는 half-step 방법과 Armijo 방법을 구현한 함수가 작성되어 있다. 본 최적화 문제에서 line search 1회에 MOLPRO single point 계산이 한 번씩 수행되기 때문에 되도록이면 상황에 맞게 적절한 line search 방법을 선택해야 한다.

**8-4. penaltyfun.py 모듈**

현재 세 penalty function 를 정의해 두었고, 등이 변경되면 update 메서드에 의해 목적 함수의 값과 정보가 쉽게 갱신되도록 작성되어 있다.

(끝)