به نام خدا

عنوان:

آنالیز مودال ساختارهای اتمی بوسیله دینامیک مولکولی

توسعه دهندگان:	حامد امینی: کارشناسی ارشد، مهندسی مواد	
	روح الله دهقانی فیروزآبادی: دکتری، هیئت علمی	
تهیه کننده مستند:	حامد امینی	
تاریخ تنظیم سند:	۹۳/ ۰۴ / ۰۴	

Market Code

فهرست مطالب

۱- چکیده ۳

۲- مقدمه ای بر موازی سازی : ۴

۳- مستندات کاربردی ۷

۳-۱- شناسنامه کد ۷

۳-۲- معرفی ورودی و خروجیهای کد مهندسی ۷

۳-۳- الزامات اجرای برنامه ۸

۴- مستندات آموزشی ۱۴

۴-۱- MAIN: ۱۵

۴-۲- LmpRun: ۱۷

۴-۳- SlepcSolver: ۲۰

۴-۴- Utils: ۲۱

۵- اعتبارسنجی کد ۲۲

Market Code

۱-چکیده

هدف از کد حاضر آنالیز مودال ساختارهای انمی بوسیله دینامیک موکولی است. این محاسبات بر اساس تغییرات کم اتم های ساختار از محل تعادلی خود و محاسبه تغییر نیرو ها است. با استفاده از تغییرات نیرو و مقدار تغییر مکان می توان ماتریس سختی ساختار مورد نظر را محاسبه کرد. این تغییرات نیرو توسط کد LAMMPS که به صورت کتابخانه خارجی فراخوانی می شود، محاسبه می شود. ماترسی سختی به دست آمده با فرمت sparse توسط کتابخانه PETSC در آمده و توسط کتابخانه SLEPC فرکانس ها و شکل مدهای ساختار به دست می ایند. این کد و تمام کتابخانه های مورد استفاده آن به صورت موازی با کتابخانه MPICH اجرا می شوند.

کلید واژه: آنالیز مودال، ساختار اتمی، دینامیک مولکولی، موازی سازی

Market Code

۲-مقدمه ای بر موازی سازی :

موازی سازی را می توان به دو دسته تقسیم کرد: موازی سازی با حافظه مشترک و حافظه جدا. در حالت اول که به آن multi thread اطلاق می شد با ایجادچندین thread یک برنامه یاprocess با شماره شناسه خاص (ID) می تواند با طوری موازی برنامه را روی چندین node اجرا کند. در عین حال تمام آنها از یک قسمت از حافظه RAM استفاده می کنند.

در حالت موازی سازی با حافظه جدایا تفکیک شده برای اجرای یک برنامه خاص چندینprocess با شماره شناسه های خاص ایجاد می شوند. هر کدان از process ها حافظه مخصوص خود را روی RAM در اختیار میگیرند. هر کدام از process ها قسمتی از عملیات برنامه را بر عهده می گیرند. به منظور ارتباط بین process های مختلف از جمله پخش کردن اطلاعات از process اصلی روی دیگر process ها و جمع آوری اطلاعات نیاز به کتابخانه های message parallel interface یا mpi است. این process ها می توانند روی هسته های یک CPU چندین هسته ای اجرا شود یا هسته های سیستم های جدا از هم که در درون یه Network قرار دارند اجرا شوند.

یکی از کتابخا نه های معروف موازی سازی mpich است. در زیر نمونه ساده ای از برنامه آورده شده است:

برای شروع mpi ابتدا باید ماژول MPI_Init که آرگومان های command line را به عنوان ورودی می گیرد را فراخوانی کرد. ماژوا های MPI_Comm_rank و MPI_Comm_size تعداد هستهای اختصاص داده شده به برنامه (size) و وشماره پروسس فراخوانی شده (rank) را در اختیار کاربر قرار می هند. Mpi را باید با ماژل MPI_Finalize اتمام کرد.

Market Code

```
#include <stdio.h>
#include "mpi.h"
int main( int argc, char *argv[] )
{
   int rank,size;
   MPI_Init(&argc,&argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);
   printf( "Hello world from process %d of %d\n", rank, size );
   MPI_Finalize();
   return 0 ;
}
```

خروجی این برنامه برای سیستم ۴ هسته ای به صورت زیر است:

```
Hello world from process 0 of 4
Hello world from process 1 of 4
Hello world from process 2 of 4
Hello world from process 3 of 4
```

این برنامه به طور هر زمان روی تمام هسته ها اجرا شده و هر هسته با شماره خاص خود (size) ماژول printf را اجرا می کند. در حالتی که تمایل به اجرای برنامه روی هسته یا process خاصی را داشته باشیم از ا فرمان if استفاده می کنیم:

```
if(rank == WhateEverYouWant):
    do somting;
```

برای مثال در کد حاضر قسمت مقداردهی اولیه بر اساس شرایط مرزی را روی process اول (یا MASTER) انجام داده و قسمت حل معدلات خطی را روی process اولیه و مابقی به صورت موازی انجام می دهیم(MASTER و SLAVE). به تعریف متغییر ها در کد های موازی باید توجه شود برای مثال:

```
if(rank == 0):
    double x  = SomeValue;
double y = SomeValue;
```

متغیر x فقط برای process صفر تعریف شده و برای بقیه process ها تعریف نشده و به آن دسترسی ندارند. اما متغیر y برای تمام process ها تعریف شده و هر process متغیر y خاص خود را دارد یعنی هر process مقدار خاصی از متغیر y را می توانند داشته باشند. در کد حاضر به طور مستقیم از کتابخانه های mpi استفاده نشده است.

Market Co

۳- مستندات کاربردی

۳-۱- شناسنامه کد

شناسنامه کدمهندسی تهیه شده برای آنالیز مودال ساختار اتمی بوسیله دینامیک مولکولی به شرح زیر است.

جدول۱: شناسنامه کد تحلیل آیروالاستیک بالک ضخیم

نام برنامه	Lmp_hess
نویسندگان	حامد امینی –روح الله دهقانی فیروزآبادی
تاریخ ارسال	۱۳۹۲/۱۱/۰۷

۳-۲- معرفی ورودی و خروجی‌های کد مهندسی

۳-۲-۱- ورودی‌های کد

ورودی های کد حاضر شامل:

۱- فایل باینری حاصل از تعادل ساختار اتمی پس از به تعادل رساندن با برنامه LAMMPS که با نام (equ.restart) ذخیره می شود. برای به دست آوردت این فایل باید ساختار اتمی را در فرمت data file لمپس را ابتدا آماده کرده و آن را توسط فایل اجرایی لمئس به تعادل برسانید. نمونه ای از این فرایند در قسمت ارزیابی اورده شده است.

. ۲- فایل با نام (potential.mod)که نوع پتانسیل و ضرایب پتانسیل های بین اتمی ساختار مورد نظر در آن ذخیره می شود.

۳- فایل مقادیر کنترل کننده برنامه. این فایل که با نام دلخواه ذخیره می شودشامل موارد زیر است:

خط ۱- exclude_type : تایپ اتم خارج شده از محاسبات.(اتم هایی که از محاسبات خارج می شوند قسمت ثابت ساختار را تشکیل می دهند)

خط ۲-displacement : مقدار تغییر مکان اتم در محاسبات از مکان تعادلی خود

خط۳-FlagUnits : اگر ۱ واحدهای metal و در غیر این صورت real برای محاسبات به کار می

Market Code

رود.

خط ۴–NrequestEigen : تعداد فرکانس هایی (مدها) که باید محاسبه شوند

خط ۵–Nprint : تعداد فرکانس هایی که اطلاعات باید چاپ شود

خط ۶–FlagEPSType : نوع ماتریس در حین محاسبات مقادیر وژه. اگر مقدار ۱ باشد ماتریس Hermitian و در غیر این صورت non-Hermitian

خط ۷–Scalse : مقدار بزرگ نمایی شکل مدها

۳–۲–۲– خروجی برنامه

خروجی ها شامل:

۱– hessian_matrix.bin : ضرایب غیر صفر ماتریس سختی در حالت باینری

۲– EigenVectors.dat : بردار مقادیر ویژه که برای محاسبه شکل مد ها استفاده می شود

۳–EigenValues.dat : مقادر ویژه یا فرکانس های ساختار

۴–Dump.dat : مختصات اتمی برای ایجاد شکل مدها

۵– Modes.lammpstrj : شکل مدهای ساختار در فرمت دامپ LAMMPS . برای نمایش گرافیکی این فایل می توان از برنامه های ovito یا vmd استفاده کرد.

۳-۳- الزامات اجرای برنامه

منظور اجری کد حاضر نیاز به سیستم عامل unix/linux یا محیط های شبیه ساز ان مانند Cygwin است. کد حاضر از کتابخانه های petsc و slepc به منظورذخیره سازی ماتریس–بردارها و به دست اوردن مقدار ویژه یا فرکانس طبیعی های ساختار و بردار مقدار ویژه یا شکل مد های ساختار اتمی استفاده می کند. ماتریس سختی ساختار اتمی با فراخوانی LAMMPS به صورت کتابخانه خارجی

Market Code

محاسبه می شود. کتابخه های مذکور خود نیاز به کامپیلر های C++/C و کتابخانه موازی ساز MPICH دارند.

نحوه نصب کتابخانه های petsc و slepc و همچنین کامپایل LAMMPS به صورت کتابخانه استاتیک در ادامه توضیح داده شده است.

۳-۳-۱- PETSC:

ابتدا با مراجعه به سایت آخرین نسخه برنامه را دانلود کنید و در پوشه معینی قرار دهید. برای مثال فایل دانلود شده را درون پوشه با آدرس زیر کپی میکنیم:

```
/home/username/Packages
```

ترمینال را باز کرده و وارد مسیر فایل دانلوده میشویم و فایل را extract میکنیم و وارد پوشه آن میشویم:

```
Cd /home/username/Packages
tar xvfz petsc-3.4.2.tar.gz
cd petsc-3.4.2
```

برای ساخت Makefile که کامپایل کدهای petsc را بر عهده دارد باید اسکریپت configure اجرا شود.

آپشن ها و مشخصات کامپایلرهای مختلف و نحوه استفاده از بسته های مورد نیاز petsc را میتوان در حین اجرای این اسکریپت مشخص کرد. در زیر نمونه ای از آن دیده میشود که کامپایلرهای سی و فرترن و سی پلاس پلاس مشخص شده و به برنامه گفته شده برنامه های مورد نیاز , blas , lapack mpich را دانلود و کامپایل کند:

```
./configure --with-fc=gfortran --with-cc=gcc
–with-cxx=g++--with-clanguage=cxx
--download-f-blas-lapack --download-mpich
```

بعد از ساخت Makefile با دستور زیر برنامه کامپایل میشود و با فرمان test از صحت کامپایل

مطمن میشویم:

```
make
// test compile
make test
```

در انتهای کامپایل مقایر دو متغیر PETSC_DIR و PETSC_ARCH پرینت میشودکه باید به

متغیرهای محیطی سیستم عامل اضافه شود.

برای این منظور:

```
//open .bashrc file in /home/username path
cd /home/username
nano .bashrc

// add these two line to bottom of file
export PETSC_DIR=[IT'S VALUE]
export PETSC_ARCH=[IT'S VALUE]
```

برای اطلاع بیشتر نصب Petsc به داکیومنت های ان مراجعه نمایید.

۳-۳-۲- SLEPC:

ابتدا با مراجعه به سایت آخرین نسخه برنامه را دانلود کنید و در پوشه معینی قرار دهید. برای مثال فایل

دانلود شده را درون پوشه با آدرس زیر کپی میکنیم:

```
/home/username/Packages
```

ترمینال را باز کرده و وارد مسیر فایل دانلوده میشویم و فایل را extract میکنیم و وارد پوشه آن