آشنایی با محیط مسایل آزمون CUTEr

نصب و راه اندازی

تألیف

مسعود فاطمی

استادیار دانشکده علوم ریاضی

دانشگاه صنعتی خواجه نصیر الدین طوسی

فهرست مطالب

فهرست تصاویر سه

فهرست جداول پنج

فصل ۱ نصب و راه اندازی ۳

۱-۱ نصب SIFDec و CUTEr ۳

فصل ۲ عملکرد SIFDec ۳۳

۲-۱ بررسی عملکرد sifdecoder ۳۳

۲-۲ اندازه های SIFDec ۳۵

۲-۳ بازسازی SIFDec ۳۶

فصل ۳ عملکرد CUTEr ۳۹

۳-۱ ابزار CUTEr ۳۹

۳-۲ اندازه های CUTEr ۴۰

۳-۳ بازسازی CUTEr ۴۰

۳-۴ رابط های CUTEr ۴۱

۳-۴-۱ دستور runcuter ۴۲

۳-۵ رابط CUTEr با MATLAB ۴۲

مراجع ۵۱

فهرست تصاویر

۱-۱ نصب Vmware Workstation گام ۱ . . . . . ۴
۱-۲ تولید سیستم عامل مجازی جدید ۴
۱-۳ نصب سیستم مجازی ۵
۱-۴ تعریف شناسه ورود ۵
۱-۵ محیط کاربری لینوکس ۶
۱-۶ بروز سازی نرم افزار ها ۶
۱-۷ بروز سازی نرم افزار ها ۷
۱-۸ خطا در بهنگام سازی ۸
۱-۹ خطا در بهنگام سازی ۹
۱-۱۰ تغییر سرور دانلود ۱۰
۱-۱۱ نصب Subversion ۱۱
۱-۱۲ ایجاد پوشه نصب ۱۲
۱-۱۳ ویرایش فایل bashrc ۱۳
۱-۱۴ نصب کامپایلر فرترن ۱۳
۱-۱۵ تعریف مسیر کامپایلر برای لینوکس ۱۴
۱-۱۶ دانلود SifDec ۱۴
۱-۱۷ دانلود CUTEr ۱۵
۱-۱۸ نصب Sifdec ۱ . . . . . ۱۵
۱-۱۹ نصب Sifdec ۲ . . . . . ۱۶
۱-۲۰ نصب Sifdec ۲ (ادامه) ۱۷
۱-۲۱ خطا زدایی ۱ . . . . . ۱۷
۱-۲۲ خطا زدایی ۲ . . . . . ۱۸
۱-۲۳ نصب Sifdec ۲ (ادامه) ۱۸

۱-۲۴ نصب Sifdec ۳ (ادامه) ۱۹
۱-۲۵ خطا زدایی ۳ . . . . . ۲۰
۱-۲۶ کد گشایی مساله KISSING ۲۰
۱-۲۷ ویرایش فایل C.arch ۲۱
۱-۲۸ نصب CUTEr ۱ . . . . . ۲۲
۱-۲۹ نصب CUTEr ۲ . . . . . ۲۳
۱-۳۰ نصب CUTEr ۳ . . . . . ۲۴
۱-۳۱ حل مساله ROSENBR با بسته gen ۲۵
۱-۳۲ نصب MATLAB ۱ . . . . . ۲۵
۱-۳۳ خطا زدایی ۱ . . . . . ۲۶
۱-۳۴ خطا زدایی ۲ . . . . . ۲۶
۱-۳۵ خطا زدایی ۳ . . . . . ۲۶
۱-۳۶ خطا زدایی ۴ . . . . . ۲۷
۱-۳۷ خطا زدایی ۵ . . . . . ۲۷
۱-۳۸ ویرایش Mexopts.sh ۲۸
۱-۳۹ ویرایش Mx ۲۸
۱-۴۰ راه اندازی مجدد Mex ۲۹
۱-۴۱ آزمون CUTEr ۲۹
۱-۴۲ خطازدایی ۱ . . . . . ۳۰
۱-۴۳ خطازدایی ۲ . . . . . ۳۰
۱-۴۴ خطازدایی ۳ . . . . . ۳۱

۲-۱ اندازه های SIFDec ۳۵
۲-۲ کدگشایی NCB۲۰ . . . . . ۳۶
۲-۳ افزایش NSETVC ۳۸
۲-۴ کدگشایی مجدد NCB۲۰ . . . . . ۳۸

۳-۱ حل مساله KISSING۲ با بسته Gen ۴۱
۳-۲ بارگزاری مساله KISSING۲ . . . . . ۴۳

فهرست جداول

۲-۱ پارامتر های اندازه در SIFDec ۳۷

۳-۱ ابزار برای مسایل نامقید ۴۷
۳-۲ ابزار برای مسایل مقید ۴۸
۳-۳ اندازه های CUTEr ۴۹
۳-۴ رابط های CUTEr ۵۰

پیشگفتار

CUTEr یک محیط مسایل آزمون برای بسته های نرم افزاری متن باز بهینه سازی و جبر خطی است که با ایجاد یک رابط بین این بسته ها و مجموعه مسایل آزمون، امکان مقایسه بسته ها را بمنظور توسعه و تولید بسته های کاراتر فراهم می سازد. این محیط شامل مجموعه ای از مسایل آزمون کد شده در فرمت استاندارد (SIF) است. این مسایل ابتدا توسط یک کدگشا (SIFDec) به فرمت های شناخته شده در فرترن تبدیل می شود و سپس توسط محیط CUTEr در اختیار بسته های نرم افزاری متن بازی مانند KNITRO ، SNOPT ، MINOS و FilterSQP قرار می گیرد. این محیط همچنین با ایجاد یک رابط با MATLAB امکان استفاده از مسایل آزمون را برای کاربران مبتدی فراهم ساخته است. در این نوشتار برآنیم تا خواننده را با برخی قابلیت های این بسته شامل نصب و راه اندازی و همچنین رابط کاربری آشنا سازیم.

فصل ۱

نصب و راه اندازی

برای نصب محیط مسایل آزمون CUTEr نیاز به یک سکوی[۱] مبتنی بر یونیکس[۲] مانند لینوکس[۳] یا مک[۴] OS/X داریم. بنابراین در گام اول نیاز به نصب یک سیستم عامل با مشخصات فوق است. این سیستم عامل را می توان یا بصورت مجزا یا بصورت مجازی[۵] در یک سیستم عامل مبنا نصب کرد.

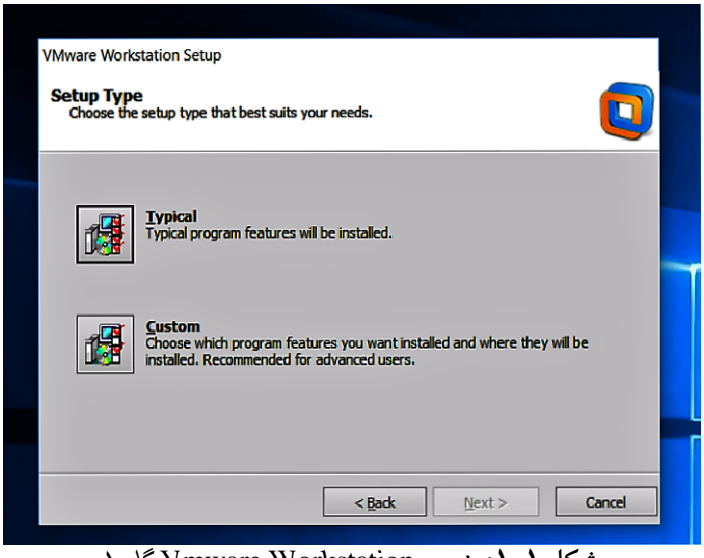
در این فصل روش نصب CUTEr در سیستم عامل مجازی لینوکس با اختیار سیستم عامل مبنای ویندوز بصورت تصویری آموزش داده می شود. کلیه مراحل خطا زدایی[۶] به تفصیل و بصورت تصویری ارایه می شود.

۱-۱ نصب SIFDec و CUTEr

برای نصب مجازی لینوکس در ویندوز نیاز به یک نرم افزار ماشین مجازی[۷] داریم در اینجا از نرم افزار VMware Workstation نسخه ۱۰.۰.۴ استفاده شده است. در ادامه روش نصب این نرم افزار در سیستم عامل ویندوز ۱۰ آموزش داده می شود.

---

[۱] Platform
[۲] Unix
[۳] Linux
[۴] Mac
[۵] Virtual
[۶] Troubleshooting
[۷] Virtual Machine



شکل ۱-۱: نصب Vmware Workstation گام ۱

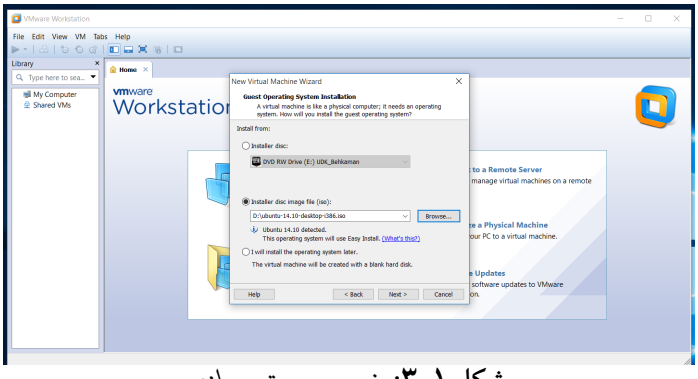
گام۱: ابتدا این نرم افزار را تهیه کرده بر روی فایل Setup کلیک کرده تا مطابق شکل ۱-۱ گام اولیه نصب آغاز شود. در تمام مراحل نصب با انتخاب گزینه های پیش فرض کلمه Next را انتخاب کنید تا نرم افزار نصب شود.

گام۲: برنامه را اجرا کنید تا مطابق شکل ۱-۲ پنجره اصلی نرم افزار نمایان شود. سپس گزینه Create a new virtual machine را انتخاب کنید.

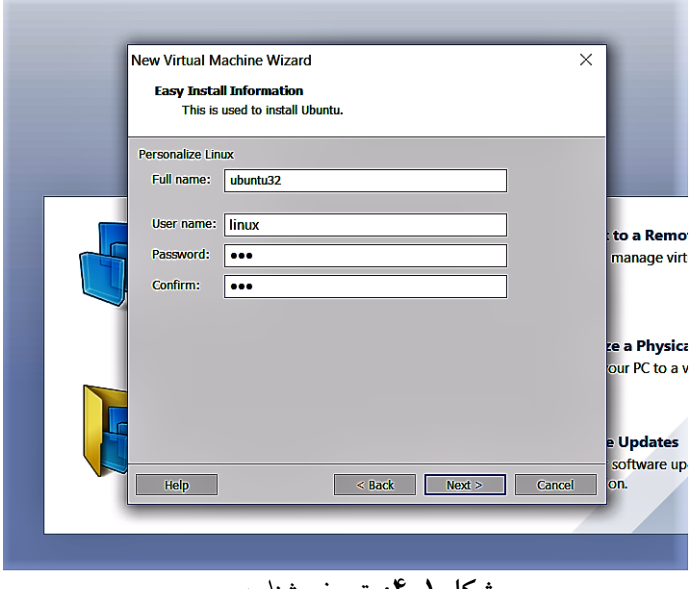
شکل ۱-۲: تولید سیستم عامل مجازی جدید

گام۳: با انتخاب گزینه های پیش فرض مطابق شکل ۱-۳ به گام نصب سیستم مجازی

یعنی پنجره Guest operating system installation می رسیم.



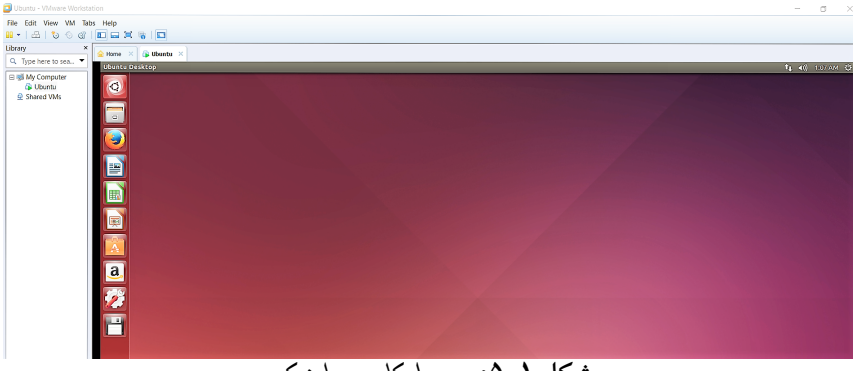
شکل ۱-۳: نصب سیستم مجازی



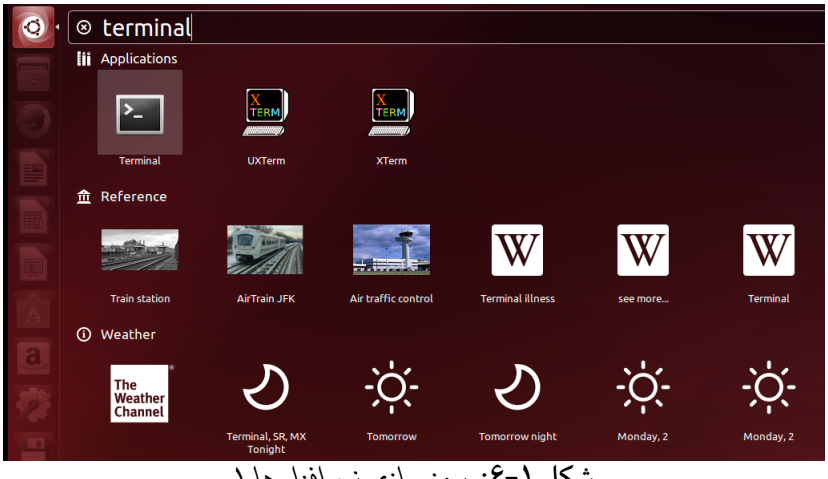
شکل ۱-۴: تعریف شناسه ورود

```
sudo ln -s $CUTERDIR/g95/g95-install/bin/i686-pc-linux-gnu-g95
/usr/bin/g95
sudo cp -R $CUTERDIR/g95/g95-install/lib/gcc-lib/ /usr/local/lib/
```

در اینجا آدرس فایل نصب سیستم عامل مجازی وارد می شود. از آنجایی که سیستم عامل مجازی مورد انتخاب ما لینوکس اوبنتو نسخه ۱۵.۱۰ است لذا آدرس فایل



شکل ۱-۵: محیط کاربری لینوکس



شکل ۱-۶: بروز سازی نرم افزار ها ۱

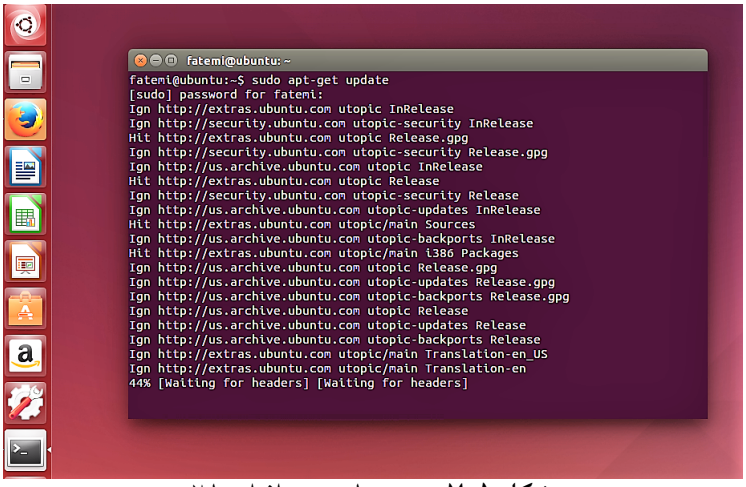
ISO حاوی این سیستم عامل را در گزینه Installer disc image file وارد می کنیم و Next را انتخاب کنید. لازم به ذکر است که ابنتو لینوکس یک نرم افزار متن باز[۸] بوده و فایل ISO آن از طریق وبگاه WWW.ubuntu.com قابل دریافت است.

گام۴: بجهت هماهنگی مراحل نصب با این راهنما توصیه می شود گزینه ها در پنجره Easy install information مطابق شکل ۱-۴ تکمیل شود. بعبارتی:

Full name=**ubuntu32**

Username=**linux**

[۸] Open Source



شکل ۱-۷: بروز سازی نرم افزار ها ۲

با انتخاب Next سیستم عامل نصب و مطابق شکل ۱-۵ وارد محیط کاربری لینوکس می شوید.

گام۵: قبل از هر چیز از اتصال به شبکه اینترنت اطمینان حاصل کنید. سپس مطابق شکل ۱-۶ از منوی سمت چپ گزینه اول را انتخاب و در کادر مربوطه کلمه Terminal را جستجو کرده روی آن کلیک کنید. پنجره Terminal باز می شود. در این پنجره مطابق شکل ۱-۷ جلوی اعلان "~$" عبارت زیر را تایپ کنید.

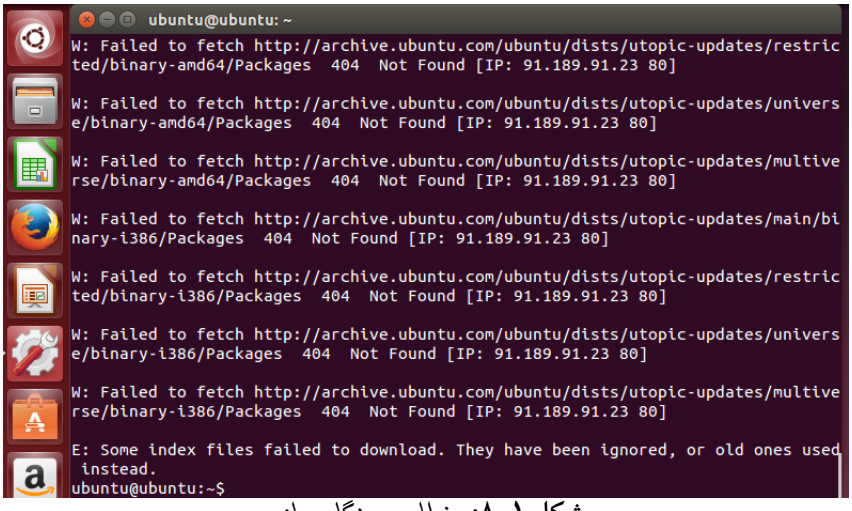
~$ sudo apt-get update

و با وارد کردن رمز خود وارد مراحل بهنگام سازی شوید.

خطا زدایی: در این مرحله ممکن است با خطایی مطابق شکل ۱-۸ مواجه شوید. بعبارتی خروجی بصورت زیر است:

E: Some index files failed to downloads. They have been ignored or old ones used instead

در این موارد مشکل از سرور ارسال کننده بوده و شما می توانید با تغییر سرور خطا را بر طرف سازید. با مشاهده این پیام وارد پنجره System setting از طریق منوی سمت چپ تصویر در شکل ۱-۹ شده و Software & Updates را کلیک کنید تا وارد این پنجره شوید در این حال از لیست تاشو در شکل ۱-۱۰ می توانید سرور را تغییر دهید. در این حالت سرور مشخص شده در شکل (خط قرمز) پیشنهاد می شود. یعنی



شکل ۱-۸: خطا در بهنگام سازی

http://archive.linux.duke.edu/ubuntu

پس از تغییر سرور گام ۵ را دوباره انجام دهید.

گام۶: در این گام باید Subversion روی سیستم نصب شود. برای اینکار پنجره Terminal را باز کرده و مطابق شکل ۱-۱۱ دستور

```
~$ sudo apt-get install subversion
```

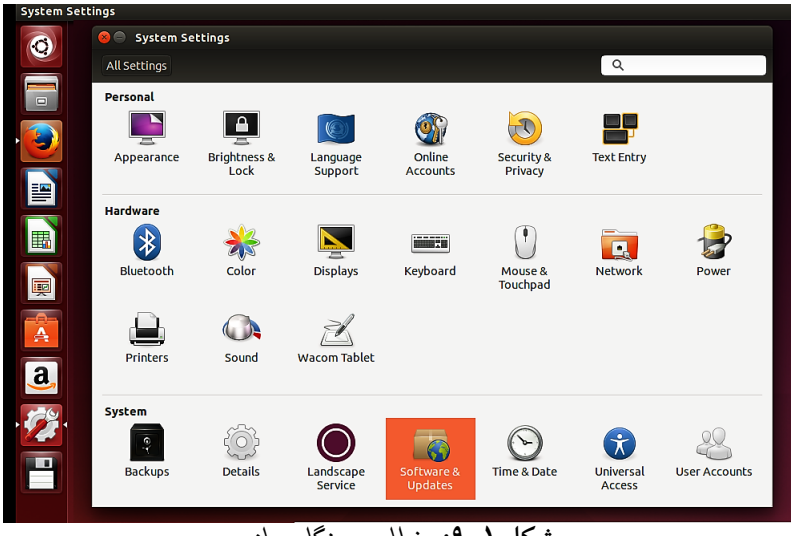
را جلو اعلان لینوکس تایپ کرده کلید اینتر را فشار دهید. پس از ورود رمز عبور اگر همه چیز درست باشد مطابق شکل ۱-۱۱ مراحل انجام می شود.

گام۷: در این گام به ایجاد پوشه نصب می پردازیم. بعد از باز کردن ترمینال مطابق شکل ۱-۱۲ عبارات زیر را جلو اعلان لینوکس تایپ کرده اینتر کنید.

```
~$ mkdir $HOME/CUTErInstallation
~$ gedit $HOME/.bashrc
```

توجه: دقت کنید که لینوکس اوبنتو به حروف کوچک و بزرگ حساس است. لذا سعی کنید دقیقا دستورات را با رعایت کوچکی و بزرگی کلمات مطابق شکل ها وارد کنید.

گام۸: پس از اجرای دستورات گام ۷ فایل bashrc بجهت ویرایش در پنجره gedit باز می شود. مطابق شکل ۱-۱۳ دستورات در کادر قرمز را با دقت در انتهای این

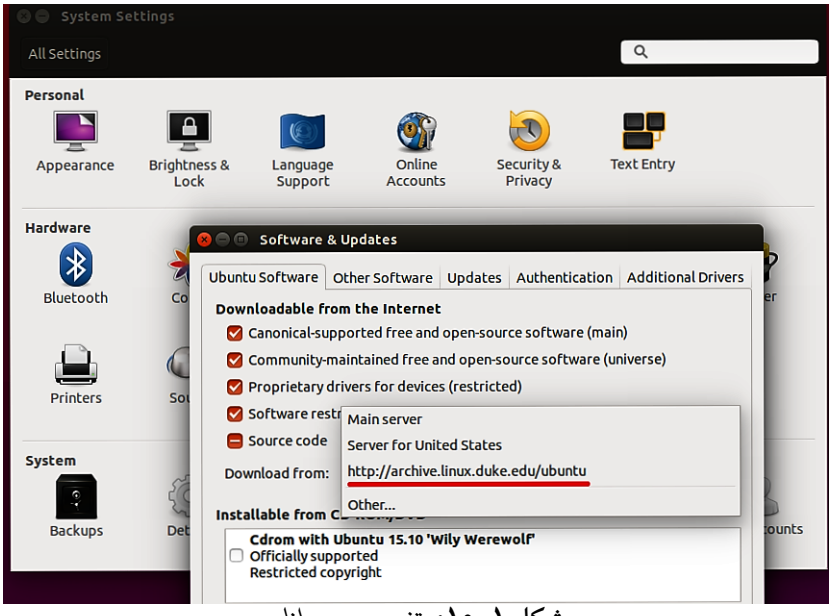


شکل ۱-۹: خطا در بهنگام سازی

فایل وارد کنید.

```
export CUTERDIR=$HOME/CUTErInstallation
export CUTER=$CUTERDIR/cuter2
export SIFDEC=$CUTERDIR/sifdec2
export MYSIFDEC=$SIFDEC/SifDec.large.pc.lnx.g95
export MANPATH=$SIFDEC/common/man:$CUTER/common/man
export PATH=$PATH:$SIFDEC/SifDec.large.pc.lnx.g95/bin:
$CUTER/CUTEr.large.pc.lnx.g95/bin:$CUTERDIR/Matlab/bin
export LIBRARY_ PATH=/usr/lib/i386-linux-gnu:
$CUTER/CUTEr.large.pc.lnx.g95/double/lib
export MASTSIF=$SIFDEC/common/sif
export MYCUTER=$CUTER/CUTEr.large.pc.lnx.g95
export MATLABPATH=$CUTER/CUTEr.large.pc.lnx.g95/double/bin:
$CUTER/common/src/matlab
```

گام۹: حال به نصب کامپایلر فرترن ۹۵ می پردازیم. این کامپایلر را در پوشه نصب CUTEr مطابق شکل ۱-۱۴ نصب می کنیم. یک ترمینال جدید باز کرده دستورات زیر را بترتیب و خط به خط جلو اعلان لینوکس "$~" وارد کنید.



شکل ۱-۱۰: تغییر سرور دانلود

```
~$ cd $CUTERDIR
~$ mkdir g95
~$ cd g95
~$ wget -O - http://ftp.g95.org/g95-x86-linux.tgz | tar xvfz -
```

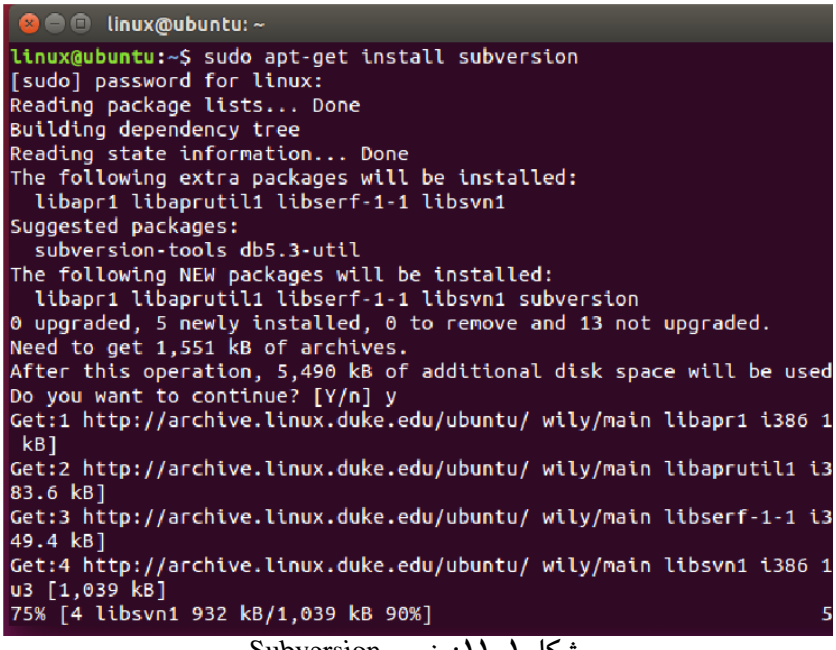
برای تعریف این کامپایلر در لینوکس مطابق شکل ۱-۱۵ دستورات زیر را جلو اعلان لینوکس تایپ کنید.

```
~$ sudo ln -s $CUTERDIR/g95/g95-install/bin/i686-pc-linux-gnu-g95 /usr/bin/g95
~$ sudo cp -R $CUTERDIR/g95/g95-install/lib/gcc-lib/ /usr/local/lib/
```

گام۹: حال به نصب CUTEr می پردازیم. ابتدا باید ملزومات آن یعنی Sifdec نصب شود. به این منظور مطابق شکل ۱-۱۶ آن را با دستور زیر دانلود می کنیم.

```
~$ cd $CUTERDIR
~$ svn co http://tracsvn.mathappl.polymtl.ca/SVN/cuter/sifdec/branches/SifDec2 ./sifdec2
```

سپس مطابق شکل ۱-۱۷ بطور مشابه CUTEr را دانلود می کنیم.



شکل ۱-۱۱: نصب Subversion

~$ cd $CUTERDIR
~$ svn co http://tracsvn.mathappl.polymtl.ca/SVN/cuter/cuter
/branches/CUTEr2 ./cuter2

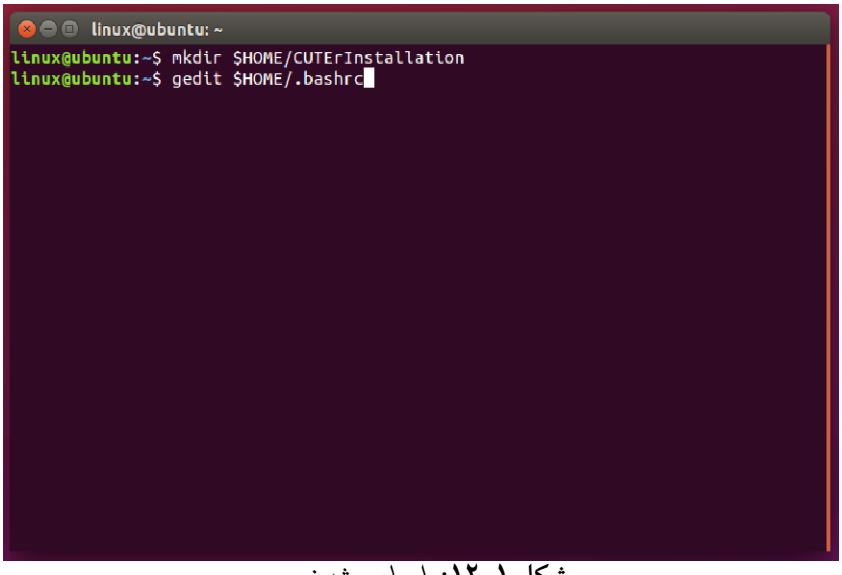
گام۱۰: ابتدا باید Sifdec نصب شود. بدین منظور مطابق شکل ۱-۱۸ توسط دستورات زیر وارد پوشه Sifdec۲ که در فولدر نصب CUTEr قرار دارد شده و سپس فایل نصب را اجرا می کنیم.

~$ cd $SIFDEC
~$ ./install_ sifdec

مطابق شکل های ۱-۱۹ و ۱-۲۰ ابتدا گزینه Select platform ظاهر می شود که عدد ۵ انتخاب کرده اینتر کنید. سپس گزینه Select operating system ظاهر شده که ۲ را انتخاب کنید. مقدار ۶ را در گزینه Select Fortran compiler انتخاب کنید.

خطا زدایی: اگر پس از انتخاب گزینه ۲ مطابق شکل ۱-۲۱ خطای زیر

gawk: no such file or directory



شکل ۱-۱۲: ایجاد پوشه نصب

ملاحظه شد آنگاه مطابق شکل ۱-۲۲ دستورات زیر را در یک Terminal جدید وارد کرده اینتر کنید تا یک پیوند بین awk و gawk در سیستم ایجاد شود. سپس گام ۱۰ را مجددا تکرار کنید.

```
~$ sudo ln -s /usr/bin/awk    /usr/bin/gawk
```

برای گزینه های بعدی نیز بترتیب D و L و Y را انتخاب کنید تا خروجی مطابق شکل ۱-۲۳ حاصل شود. در ادامه مراحل نصب مجددا Y را در باقیمانده مراحل انتخاب کنید. اگر همه چیز بدرستی انجام شود با خروجی مطابق شکل ۱-۲۴ پروسه نصب خاتمه می یابد.

خطا زدایی: اگر در مراحل نصب با خطایی مطابق شکل ۱-۲۵ بصورت

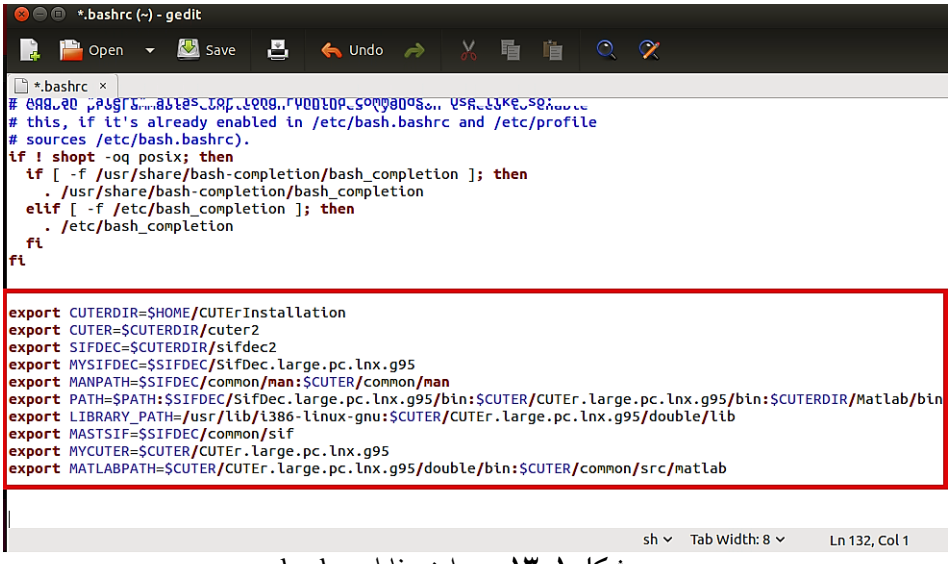
```
ld: cannot find crt1.o: no such file or directory
```

مواجه شدید ابتدا یک Terminal جدید باز کرده دستور زیر را در آن اجرا کنید. سپس گام ۱۰ را مجددا اجرا کنید.

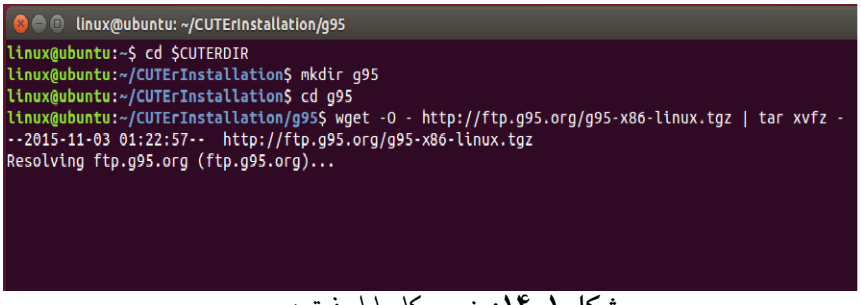
```
~$ sudo apt-get install gcc-multilib
```

برای اطمینان از صحت نصب Sifdec یک Terminal جدید باز کرده و دستور زیر را در آن وارد کنید.

```
~$ sifdecode KISSING2
```



شکل ۱-۱۳: ویرایش فایل bashrc



شکل ۱-۱۴: نصب کامپایلر فرتن

خروجی باید مطابق شکل ۱-۲۶ باشد.

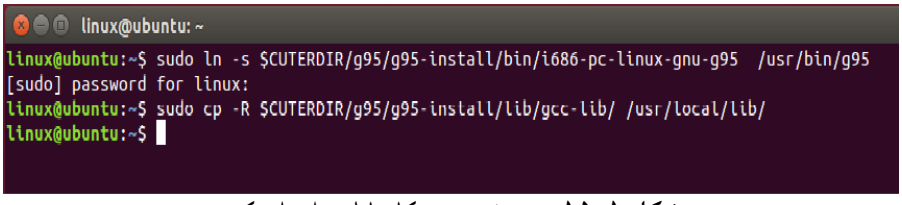
گام۱۱: حال نوبت به نصب CUTEr رسیده است. قبل از نصب باید تغییری جزیی در یکی از فایل های نصبی دهیم. بدین منظور یک Terminal جدید باز کرده و دستور زیر را در آن وارد کنید.

~$ gedit $CUTER/build/arch/c.arch

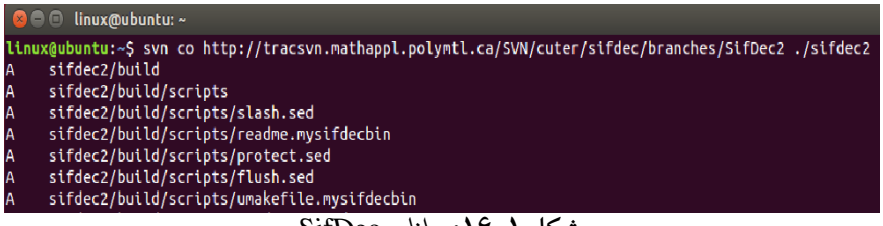
در فایل باز شده مطابق شکل ۱-۲۷ خط

all ; all ; gcc ; -DIsgcc ; GNU_ gcc

را یافته و به



شکل ۱-۱۵: تعریف مسیر کامپایلر برای لینوکس



شکل ۱-۱۶: دانلود SifDec

all ; all ; gcc-4.1 ; -DIsgcc ; GNU_ gcc-4.1

تبدیل کنید. سپس دستورات زیر را جلو اعلان لینوکس وارد کرده تا پروسه نصب آغاز شود.

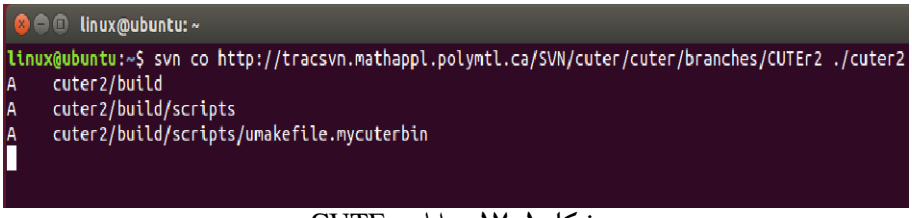
~$ cd $CUTER
~$ ./install_ cuter

مطابق شکل های ۱-۲۸ و ۱-۲۹ بترتیب مقادیر L،D،۳،۶،۲،۵ وY را برای گزینه های ظاهر شده انتخاب کنید. در سایر موارد نیز گزینه Y را اختیار کنید در نهایت با خروجی مطابق شکل ۱-۳۰ نصب خاتمه می یابد. دقت کنید که در همه موارد مطابق شکل باید عبارت [ok] ظاهر شود. برای آگاهی از صحت نصب CUTEr دستور زیر را اجرا کنید خروجی باید مطابق شکل ۱-۳۱ باشد.

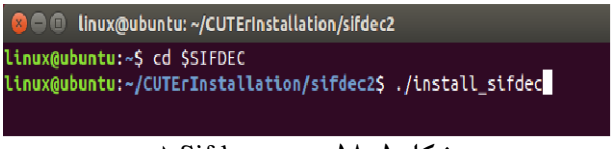
~$ runcuter -p gen -D ROSENBR

گام ۱۲: برای استفاده از CUTEr در MATLAB ابتدا باید MATLAB نصب شود. بدین منظور به توزیعی از آن سازگار با لینوکس نیاز دارید. در این راهنما ما ابتدا فایل های اجرایی را در یک دایرکتوری در بنام matlab-ubuntu در پوشه خانه[۹] کپی کرده و اقدام به نصب می کنیم لذا دستوراتی که در ادامه مشاهده می کنید با این پیش فرض معرفی شده است. دقت کنید که نصب MATLAB جزیی از

[۹] Home directory



شکل ۱-۱۷: دانلود CUTEr



شکل ۱-۱۸: نصب Sifdec ۱

پروسه نصب CUTEr نیست ولی بجهت بروز برخی خطا های رایج در هنگام نصب، این پروسه در اینجا با جزییات دنبال می شود.

پس از کپی فایل های اجرای MATLAB در پوشه matlab-ubuntu یک Terminal جدید باز کرده دستورات زیر را وارد کنید تا پروسه نصب مطابق شکل ۱-۳۲ آغاز شود.

```
~$ cd matlab-ubuntu
~$ ./install
```

توجه: پوشه نصب MATLAB را در پوشه نصب CUTEr ایجاد کرده سپس MATLAB را در آن نصب کنید. با دستورات زیر این پوشه ایجاد می شود.

```
~$ cd $CUTERDIR
~$ mkdir Matlab
```

سپس در پنجره Mathworks Installer در شکل ۱-۳۲ پوشه نصب را روی پوشه ایجاد شده در قسمت قبل تنظیم کنید.

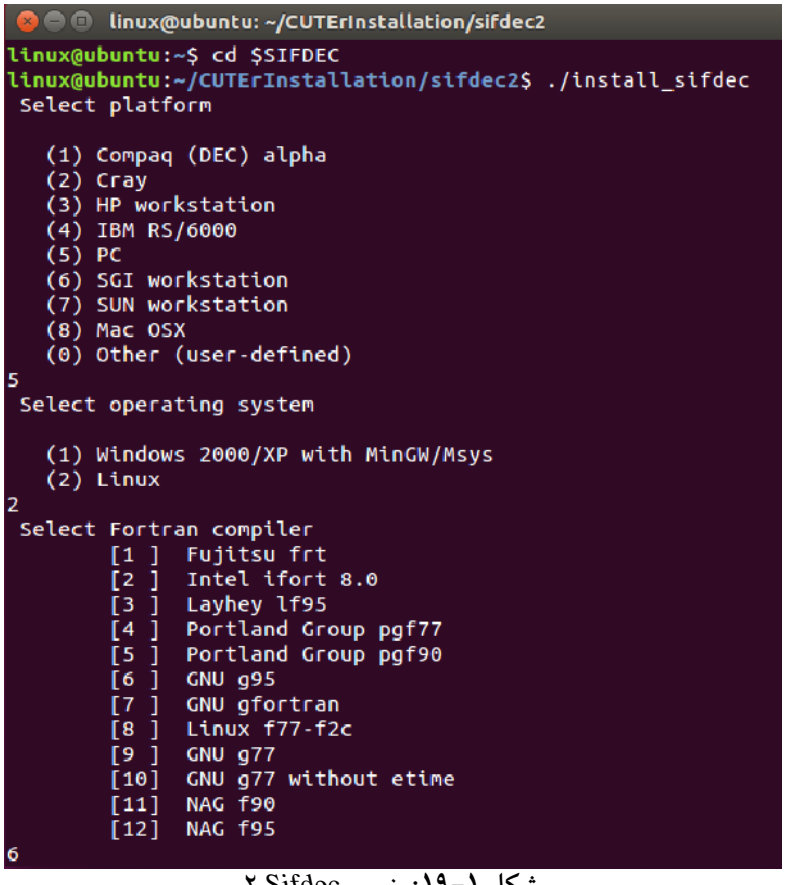
خطا زدایی: اگر پس از اجرای دستور

```
~$ ./install
```

با خطایی مطابق شکل ۱-۳۳ بصورت

```
bash: ./install : Permission denied
```

مواجه شدید. در این حالت باید به پوشه نصب MATLAB مراجعه کرده مطابق شکل ۱-۳۴ فایل install را یافته با راست کلیک بروی آن وارد



شکل ۱-۱۹: نصب Sifdec ۲

Properties شده و لیست تاشو ACCESS را به حالت and Wright Read تغییر دهید. بعلاوه گزینه Execute را نیز تیک بزنید. پس از تغییر، مجددا گام ۱۲ را اجرا کنید.

اگر با خطای

./install : 1: ./install: /lib/libc.so.6: not found

مطابق شکل ۱-۳۵مواجه شدید دستور زیر را اجرا کرده و مجددا گام ۱۲ را تکرار کنید.

~$ sudo ln -s /lib/i386-linux-gnu/libc.so.6 /lib/libc.so.6

اگر با خطای

./install : 86: ...xsetup: Permission denied

```
 Select Fortran compiler
        [1 ]  Fujitsu frt
        [2 ]  Intel ifort 8.0
        [3 ]  Layhey lf95
        [4 ]  Portland Group pgf77
        [5 ]  Portland Group pgf90
        [6 ]  GNU g95
        [7 ]  GNU gfortran
        [8 ]  Linux f77-f2c
        [9 ]  GNU g77
        [10]  GNU g77 without etime
        [11]  NAG f90
        [12]  NAG f95
6
 Select install precision (D=double, s=single):
D
 Select install size (L=large, M=medium, S=small, C=customized):
L

 By default, SifDec with your selections will be installed in
  /home/linux/CUTErInstallation/sifdec2/SifDec.large.pc.lnx.g95
 Is this OK (Y/n)?
```

شکل ۱-۲۰: نصب Sifdec ۲ (ادامه)

```
linux@ubuntu: ~/CUTErInstallation/sifdec2
   (2) Linux
2
 Select Fortran compiler
/home/linux/CUTErInstallation/sifdec2/bin/install_sifdec.sh: line 290: /usr/bin/
gawk: No such file or directory
/home/linux/CUTErInstallation/sifdec2/bin/install_sifdec.sh: line 291: /usr/bin/
gawk: No such file or directory
```

شکل ۱-۲۱: خطا زدایی ۱

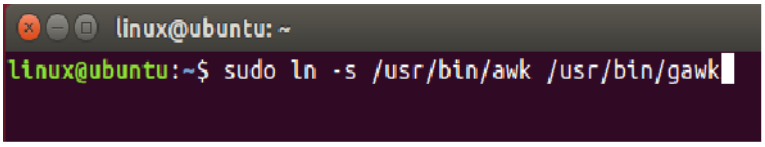
مطابق شکل ۱-۳۶ مواجه شدید آنگاه مطابق شکل ۱-۳۷ به پوشه حاوی فایل Xsetup مراجعه کرده در پنجره Properties لیست تاشو ACCESS را به حالت Read and Wright تغییر دهید. بعلاوه گزینه Execute را نیز تیک بزنید. پس از تغییر، مجددا گام ۱۲ را اجرا کنید. این فایل در مسیر

$HOME/matlab-ubuntu/update/bin/glnx86

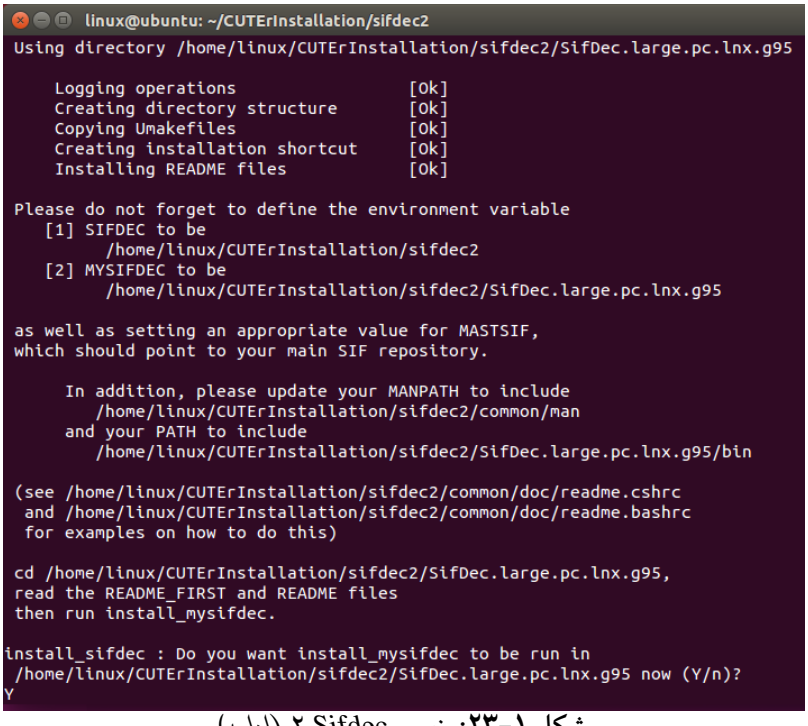
قرار دارد.

اگر با خطایی مبنی بر عدم وجود فایل libxp.so.۶ مواجه شدید دستور زیر را وارد کنید و گام ۱۲ را مجددا تکرار کنید.

~$ sudo ln - /usr/lib/i386-linux-gnu/libXpm.so.4 /usr/lib/i386-linux-gnu/libXp.so.6



شکل ۱-۲۲: خطا زدایی ۲



شکل ۱-۲۳: نصب Sifdec ۲ (ادامه)

پس از نصب MATLAB باید برخی فایل ها ویرایش شود در اینجا به این موضوع پرداخته می شود.

اولین فایل Mexopts.sh است که در پوشه نصب MATLAB و در زیرپوشه Bin قرار دارد. این فایل را با دستور

```
~$ gedit $CUTERDIR/Matlab/bin/mexopts.sh
```

برای ویرایش باز کرده مطابق شکل ۱-۳۸ خط مربوطه را در فایل پیدا کرده و تغییرات زیر را در آن اعمال کنید. بعبارتی FC را به 'g۹۵'=FC و LD را به LD='$FC' تغییر دهید. دومین فایل Mx است که در پوشه نصب CUTEr قرار دارد. با دستور زیر فایل را باز کنید.

```
make[1]: Leaving directory '/home/linux/CUTErInstallation/sifdec2/SifDec.lar

install_mysifdec : Do you want to 'make all' in
 /home/linux/CUTErInstallation/sifdec2/SifDec.large.pc.lnx.g95 now (Y/n)?
y
cleaning in ./bin...
cleaning in ./double...
cleaning in ./double/bin...
cleaning in ./double/config...
 Casting      local.f                  [Ok]
 Building     local.o                  [Ok]
making all in ./bin...
 Linking      slct                     [Ok]
 Linking      clsf                     [Ok]
 Casting      classall                 [Ok]
 Casting      classify                 [Ok]
 Casting      helpmsg                  [Ok]
 Casting      select                   [Ok]
 Casting      sifdecode                [Ok]
 Installing   show.awk                 [Ok]
 Installing   param.awk                [Ok]
making all in ./double...
making all in ./double/bin...
 Building     sifdec.o                 [Ok]
 Building     decode.o                 [Ok]
 Building     gps.o                    [Ok]
 Building     inlanc.o                 [Ok]
 Building     makefn.o                 [Ok]
 Building     mafnad.o                 [Ok]
 Building     makegr.o                 [Ok]
 Building     magrad.o                 [Ok]
 Building     printp.o                 [Ok]
 Building     rest.o                   [Ok]
 Building     trans.o                  [Ok]
 Building     utils.o                  [Ok]
 Linking      sifdec                   [Ok]
making all in ./double/config...
cleaning in ./bin...
cleaning in ./double...
cleaning in ./double/bin...
```

شکل ۱-۲۴: نصب Sifdec ۳ (ادامه)

~$ gedit $CUTER/CUTEr.large.pc.lnx.g95/bin/mx

و مطابق شکل ۱-۳۹ عبارت زیر را جلو PACKLIBS وارد کنید.

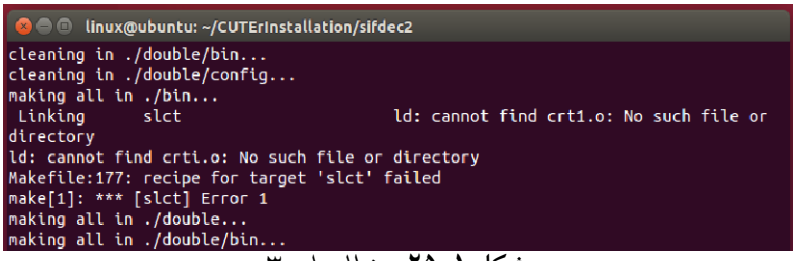
export PACKLIBS=”-L/usr/local/lib/gcc-lib/i686-pc-linux-gnu/4.0.3 -lf95 ”

در آخرین گام MATLAB را باز کنید. کافیست Terminal را باز کرده و دستور زیر را در آن وارد کنید.

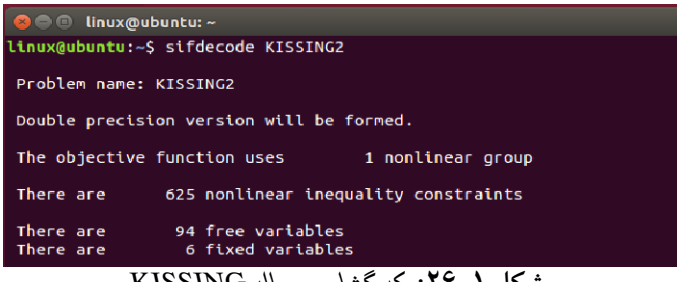
~$ matlab

سپس در پنجره Command window مطابق شکل ۱-۴۰ عبارت زیر را تایپ کرده اینتر کنید. در انتها با ورود عدد ۲ مجددا اینتر کنید.

» mex -setup



شکل ۱-۲۵: خطا زدایی ۳



شکل ۱-۲۶: کد گشایی مساله KISSING

گام ۱۳: در این گام صحت ارتباط بین CUTEr و MATLAB مورد آزمون قرار می گیرد. بدین منظور یک Terminal جدید باز کرده دستور زیر را در آن وارد کنید. خروجی باید مطابق شکل ۱-۴۱ باشد.

~$ runcuter - -package mx - -decode KISSING2

خطازدایی: اگر در گام ۱۳ با خطایی مشابه شکل ۱-۴۲ مواجه شدید، یعنی

error: expected expresion before '**long**'...long(icon));

مراحل زیر را دنبال کنید.

ابتدا دستور زیر را در Terminal وارد کرده اینتر کنید.

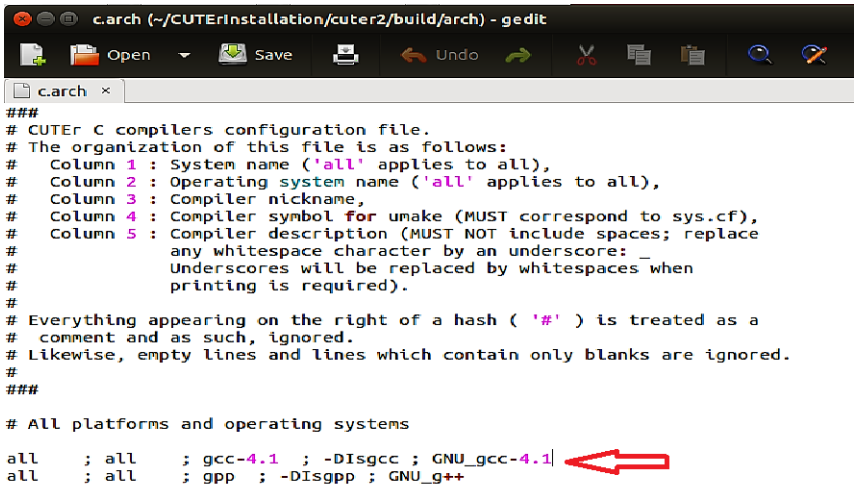
~$ sudo ln -s /usr/lib/gcc/i686-linux-gnu/5/libstdc++.so /usr/lib/libstdc++.so

سپس فایل mcuter.c را برای ویرایش با دستور زیر باز کنید.

~$ gedit $CUTER/common/src/tools/mcuter.c

مطابق شکل های ۱-۴۳ و ۱-۴۴ مکان نما را ابتدا به خط ln۶۱۷،col۴۳ و سپس به خط ln۷۳۶،col۳۱ برده و در هر دو سطر عبارت

if (icon<=0 || icon > CUTEr_ ncon)
sprintf(msgBuf,

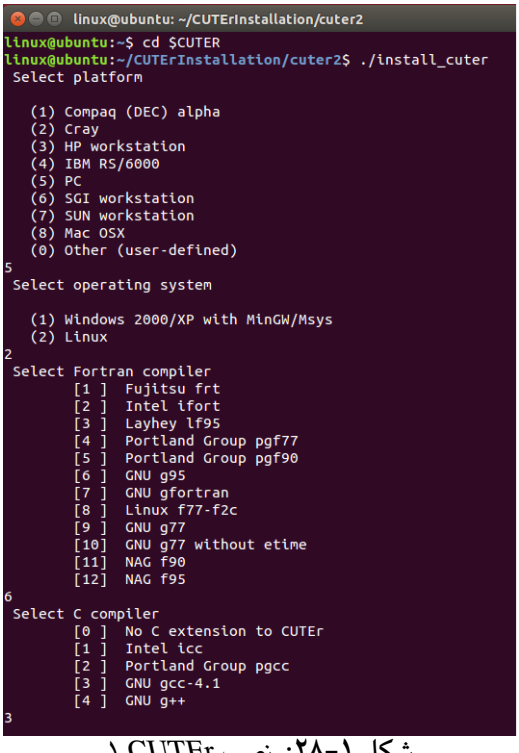


شکل ۱-۲۷: ویرایش فایل C.arch

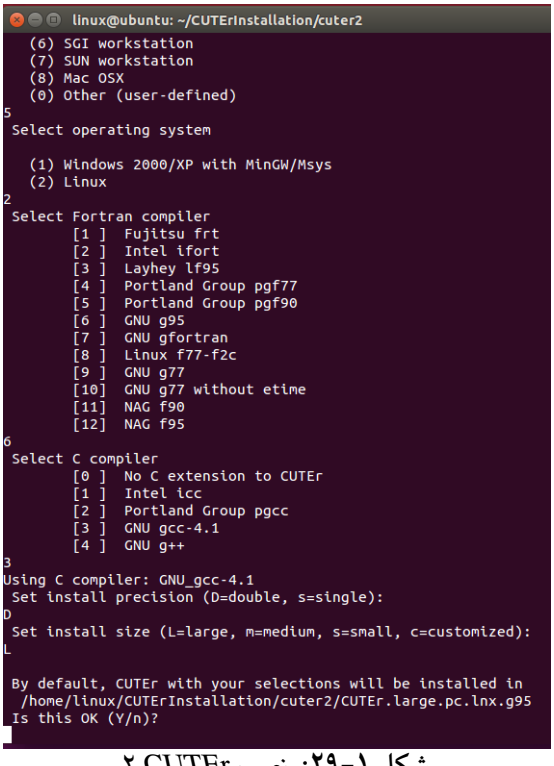
”iscons:...icon);

mexErrMsgTex(msgBuf);

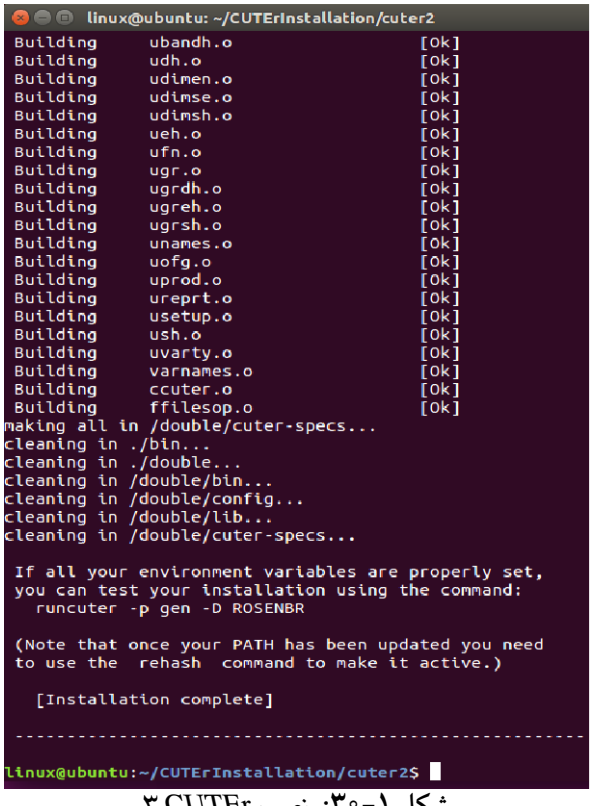
را غیر فعال کنید.



شکل ۱-۲۸: نصب CUTEr ۱

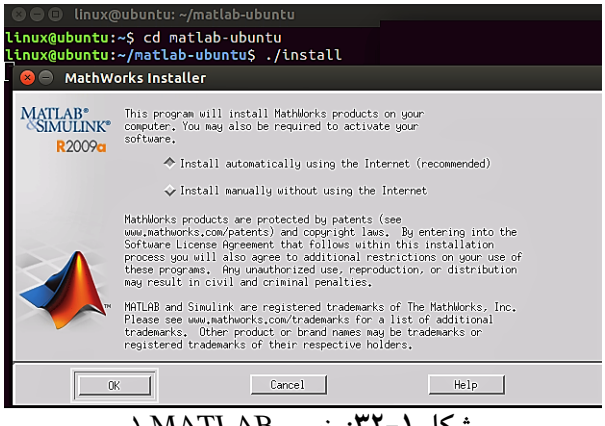


شکل ۱-۲۹: نصب CUTEr ۲

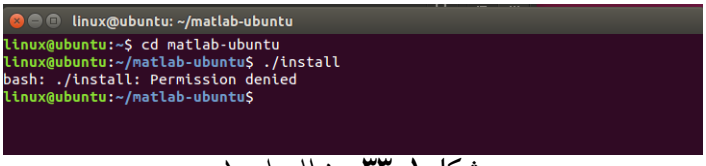


شکل ۱-۳۰: نصب CUTEr ۳

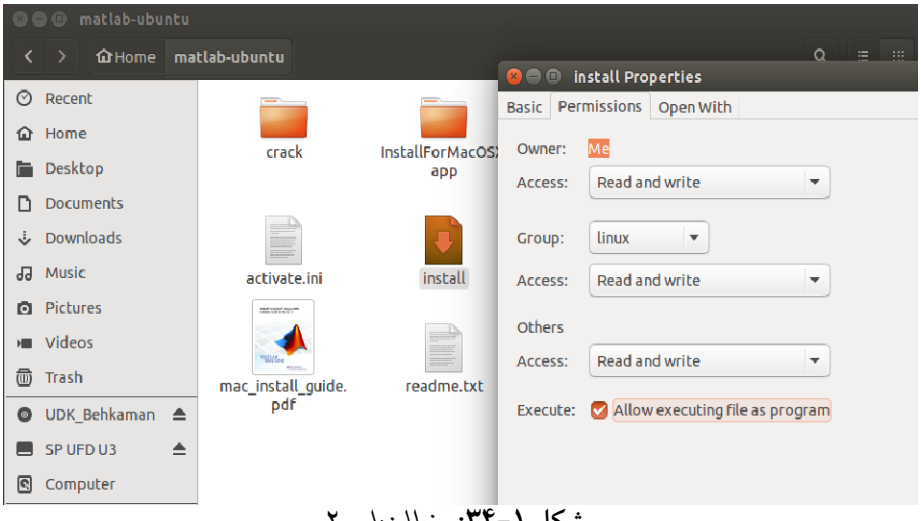
شکل ۱-۳۱: حل مساله ROSENBR با بسته gen



شکل ۱-۳۲: نصب MATLAB ۱

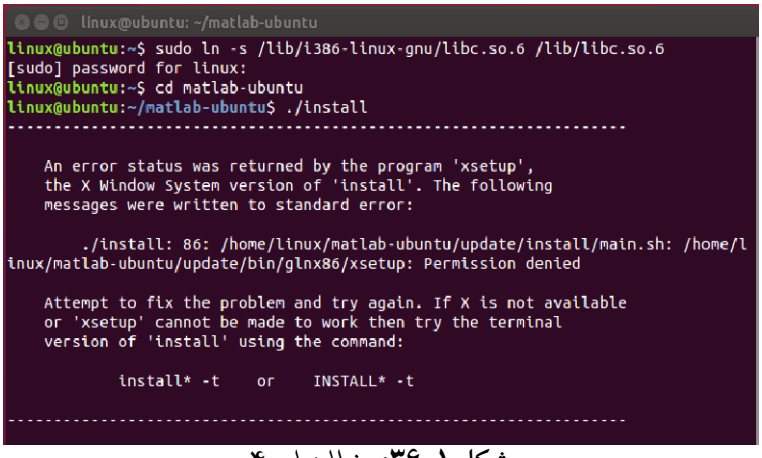


شکل ۱-۳۳: خطا زدایی ۱

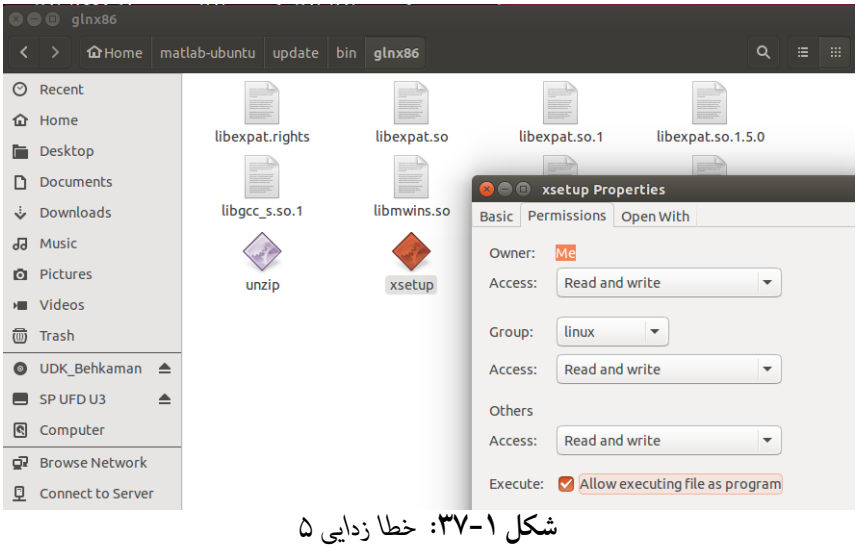


شکل ۱-۳۴: خطا زدایی ۲

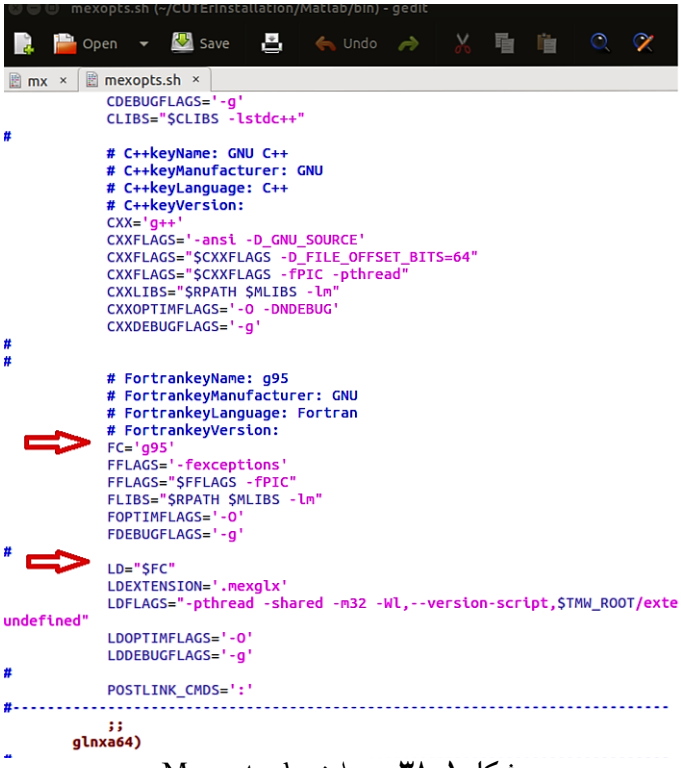
شکل ۱-۳۵: خطا زدایی ۳



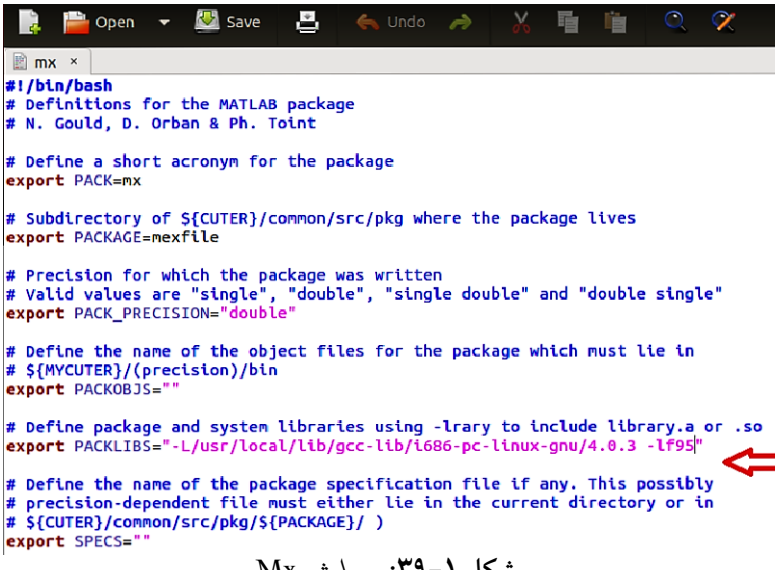
شکل ۱-۳۶: خطا زدایی ۴



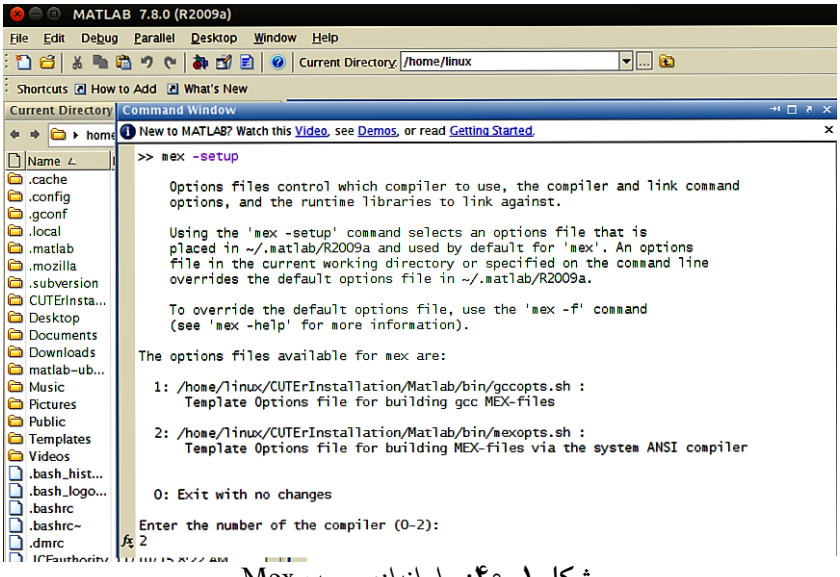
شکل ۱-۳۷: خطا زدایی ۵



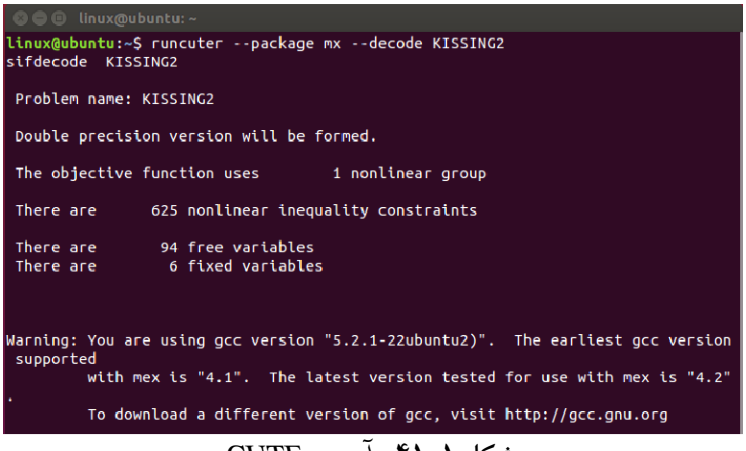
شکل ۱-۳۸: ویرایش Mexopts.sh



شکل ۱-۳۹: ویرایش Mx



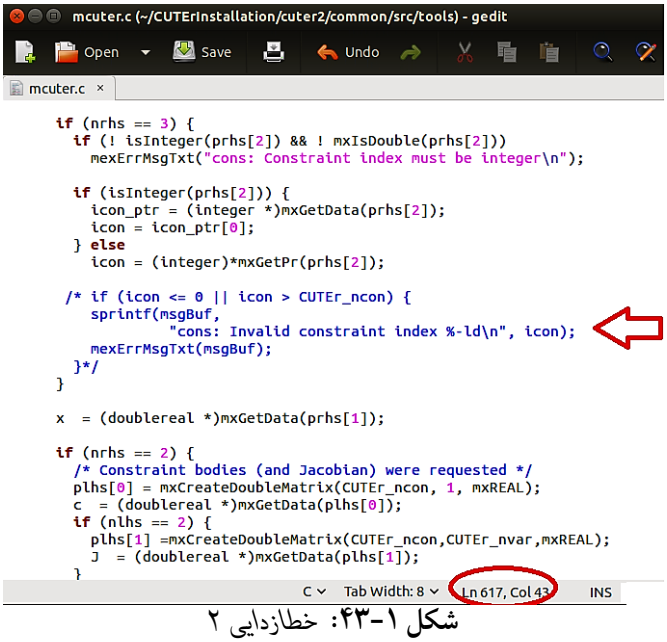
شکل ۱-۴۰: راه اندازی مجدد Mex



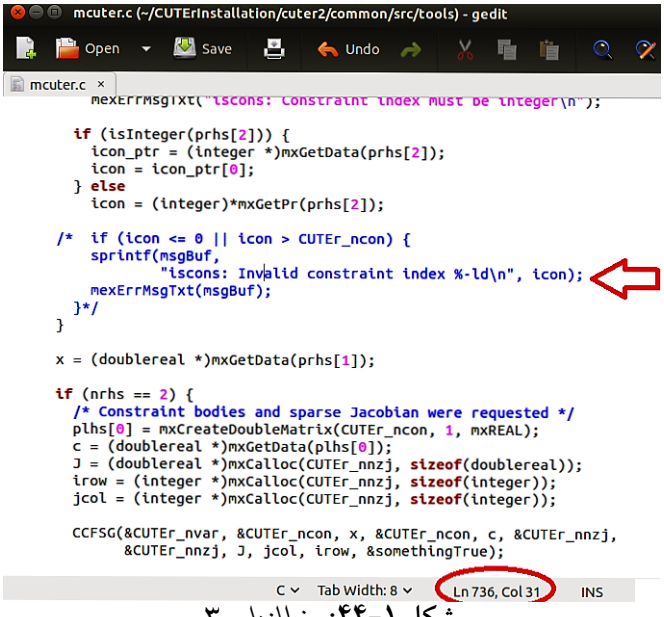
شکل ۱-۴۱: آزمون CUTEr

```
linux@ubuntu:~$ runcuter --package mx --decode KISSING2
sifdecode  KISSING2

 Problem name: KISSING2

 Double precision version will be formed.

 The objective function uses        1 nonlinear group

 There are      625 nonlinear inequality constraints

 There are       94 free variables
 There are        6 fixed variables



Warning: You are using gcc version "5.2.1-22ubuntu2)".  The earliest gcc version supported
         with mex is "4.1".  The latest version tested for use with mex is "4.2".
         To download a different version of gcc, visit http://gcc.gnu.org

/home/linux/CUTErInstallation/cuter2/common/src/tools/mcuter.c: In function ‘mexFunction’:
/home/linux/CUTErInstallation/cuter2/common/src/tools/mcuter.c:617:61: error: expected expression before ‘long
’
                    "cons: Invalid constraint index %-ld\n", long(icon));
                                                            ^
/home/linux/CUTErInstallation/cuter2/common/src/tools/mcuter.c:736:62: error: expected expression before ‘long
’
                   "iscons: Invalid constraint index %-ld\n", long(icon));
                                                             ^

    mex: compile of ' "/home/linux/CUTErInstallation/cuter2/common/src/tools/mcuter.c"' failed.
```

شکل ۱-۴۲: خطازدایی ۱



شکل ۱-۴۳: خطازدایی ۲



شکل ۱-۴۴: خطازدایی ۳

فصل ۲

عملکرد SIFDec

همانطور که درفصل قبل دیدیم اولین گام در نصب CUTEr ، نصب SIFDec است. در حقیقت SIFDec یک کد گشاست، بدین معنی که مسایل آزمون کد شده در فرمت SIF[۱] را به داده ها و فایل های سازگار در فرترن ۷۷ ترجمه می کند. این فایل ها سپس قابل دسترسی توسط بسته های بهینه سازی[۲] خواهند بود. لازم به ذکر است که برخی بسته های بهینه سازی مانند LANCELOT-B و GALAHAD نیز مشابه CUTEr از SIFDec برای کد گشایی مسایل استفاده می کنند، از این رو مجزا بودن نصب CUTEr از SIFDec طبیعی بنظر می رسد. در این بخش قصد داریم به بررسی جزییات ساختار این کدگشا به تفصیل بپردازیم.

۲-۱ بررسی عملکرد sifdecoder

اگرچه بسته های نرم افزاری در زمان حل یک مساله SIFDec را بصورت خودکار فراخوانی می کنند، ولی ضروریست که خواننده با جزییات آن آشنا شود.

زمانی که SIFDec فراخوانی می شود مساله به فایل اجرایی این کد کشا یعنی sifdecode سپرده می شود. شکل کلی این دستور بصورت زیر است:

sifdecode [-s] [-h] [-k] [-o j] [-l sec] [-s] [-f] [-b] [-a j] [-show] [-param name=value [,name=value...]] [-force] [-debug] probname[.SIF]

ایتم هایی که در [.] قرار دارند اختیاری هستند. در ادامه به برخی جزییات ایتم های پر

---

[۱] Standard Input Format

[۲] Optimization Package

کاربرد اختیاری پرداخته می شود و خواننده را به مرجع [] برای ایتم های دیگر ارجاع می دهیم.

-s: با فراخوانی این ایتم مساله با دقت ساده[۳] کد گشایی و برای حل آماده می شود. لازم به ذکر است که این ایتم زمانی موثر است فراخوانی شود که SIFDec در حالت دقت ساده نیز نصب شده باشد. دقت کنید که در بخش قبل SIFDec تنها در حالت دقت مضاعف[۴] نصب شد.

-h: با فراخوانی این ایتم راهنمایی که مشاهده می کنید ظاهر می شود. این را می توانید با اجرای دستور زیر در Terminal تحقیق کنید.

```
~$ sifdecode -h
```

-show: با فراخوانی این ایتم همه پارامتر های قابل تغییر در مساله، بعنوان مثال ابعاد مساله، نمایش داده می شود. بطور نمونه با ورود دستور

```
~$ sifdecode -show KISSING2
```

خروجی بصورت زیر است.

```
m=24 (IE)                uncommented
m=25 (IE) -defult value- uncommented
```

مشاهده می شود که پارامتر پیش فرض برای این مساله m=۲۵ است.

-param: با فراخوانی این ایتم می توان مساله را با پارامتر های غیر پیش فرض نمایش داده شده در قسمت قبل کد گشایی کرد. بطور نمونه با ورود دستور

```
~$ sifdecode -param m=24 KISSING2
```

مساله با پارامتر m=۲۴ کدگشایی می شود. دقت کنید اگر مساله ای بیش از یک پارامتر قابل تغییر داشت مقادیر با کاما از هم جدا شوند. فرض کنید مساله SIMPLE دو پارامتر قابل تغییر THETA=۳ و N=۱۰۰ داشته باشد. مساله با این پارامتر ها بصورت زیر فراخوانی می شود.

```
~$ sifdecode -param N=100, THETA=3 SIMPLE
```

-force: با این ایتم می توان مساله را با پارامتر های دلخواه کد گشایی کرد حتی پارامتر هایی که در ایتم show- تعریف نشده اند. مثلا برای مثال قبل با وارد کردن

[۳] Single precision

[۴] Double percision

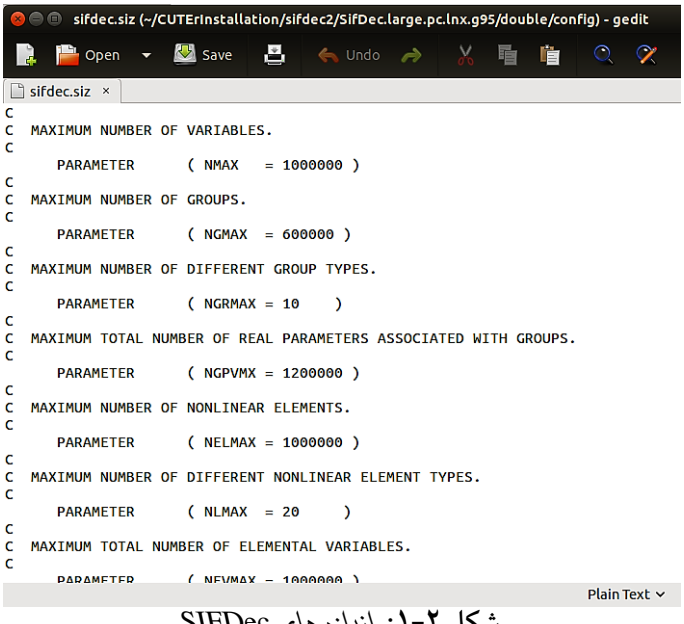
```
~$ sifdecode -param N=100, THETA=2.5 -force SIMPLE
```

مساله با پارامتر THETA=۲.۵ که در ایتم show- تعریف نشده، کدگشایی می شود.

توجه: از آنجایی که پارامتر های دلخواه غیر از موارد نشان داده شده در ایتم show- غیر استاندارد هستند معمولا این حالت با پیغام خطا همراه است. لذا استفاده از این دستور توصیه نمی شود.

۲-۲ اندازه های SIFDec

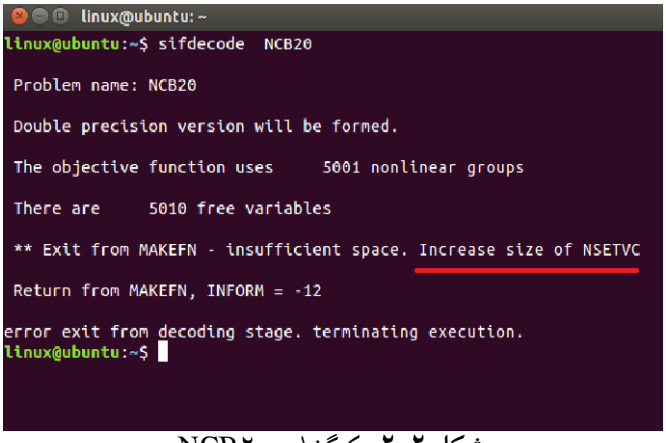
بسته SIFDec در سه اندازه Small، Medium و Large قابل نصب است. علاوه بر این موارد اندازه Custom نیز قابل دسترس است که کاربر می تواند براساس انتخاب های دلخواه SIFDec را نصب کند. این اندازه ها میزان حافظه اختصاص داده شده برای کد گشایی را تعیین می کنند. همانطور که در ادامه میبینیم ممکن است برای برخی مسایل نیاز به افزایش حافظه اختصاص داده شده به برخی پارامتر ها داشته باشیم. در اینجا برآنیم که روش تغییر اندازه SIFDec را مورد بررسی قرار دهیم.



شکل ۲-۱: اندازه های SIFDec

پارامتر های مختلف تعیین کننده اندازه SIFDec در فایل با مسیر

$MYSIFDEC/double/config/sifdec.siz



شکل ۲-۲: کدگشایی NCB۲۰

مطابق شکل ۲-۱ قرار دارند. این پارامتر ها در جدول ۲-۱ خلاصه شده اند. بعنوان مثال در کدگشایی مساله NCB۲۰ خروجی مطابق شکل ۲-۲ است. همانطورکه ملاحظه می شود اندازه پارامتر NSETVC باید افزایش یابد. بدین منظور فایل sifdec.size در آدرس داده شده را با دستور زیر که در ترمینال وارد می کنید باز کرده و

~$ gedit $MYSIFDEC/double/config/sifdec.siz

مطابق شکل ۲-۳ اندازه NSETVC را افزایش می دهیم. بعبارتی ۱۰۰ را به ۵۰۰ تغییر می دهیم. بعد از ذخیره سازی می توانید از پنجره gedit خارج شوید و SIFDec را بازسازی[۵] کنید.

۲-۳ بازسازی SIFDec

پس از هرگونه ویرایش در فایل ها و جزییات SIFDec باید بمنظور اعمال تغییرات آن را بازسازی کرد. برای اینکار می توان از دستور all Make در مسیر $MYSIFDEC استفاده کرد. بعبارتی دستورات زیر را بترتیب در Terminal اجرا کنید.

~$ cd $MYSIFDEC

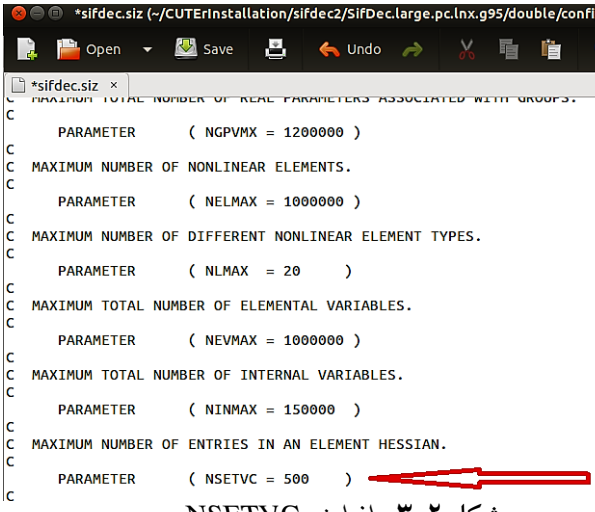
~$ make -s all

حال اگر دوباره اقدام به کدگشایی مساله NCB۲۰ کنید مطابق شکل ۲-۴ مساله با موفقیت کدگشایی می شود.

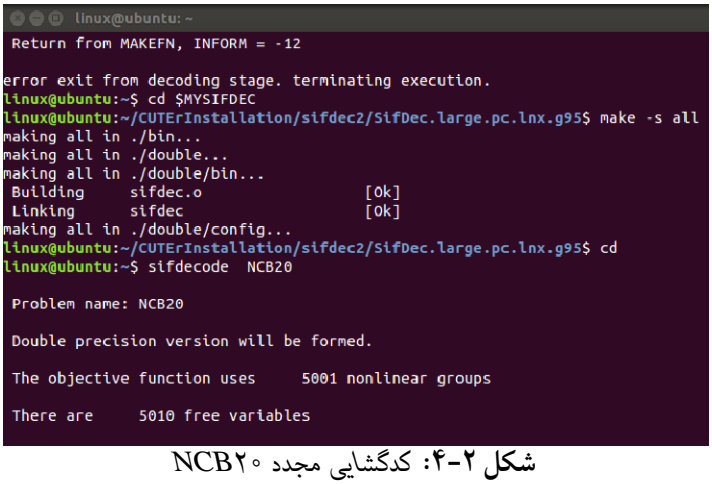
[۵] Rebuild

پارامتر	شرح پارامتر
NMAX	حداکثر تعداد متغیر های قابل تعریف در مساله
NGMAX	حداکثر تعداد گروه های قابل تعریف در مساله
NGRMAX	حداکثر تعداد قابل تعریف برای نوع گروها در مساله
NGPVMX	حداکثر تعداد پارامتر های ثابت در گروه ها
NELMAX	حداکثر تعداد واحد های غیر خطی قابل تعریف در مساله
NLMAX	حداکثر تعداد قابل تعریف برای نوع واحد های غیر خطی در مساله
NEVMAX	حداکثر تعداد متغیر ها قابل تعریف در واحد های غیر خطی
NINMAX	حداکثر تعداد متغیر ها ی اینترنال قابل تعریف در مساله
NSETVC	حداکثر تعداد عناصر ناصفر در ماتریس هسی
NEPVMX	حداکثر تعداد پارامتر های ثابت قابل تعریف در واحد های غیر خطی
LA	حداکثر تعداد عناصر ناصفر در واحد های خطی
NINDEX	حداکثر تعداد پارامتر های صحیح قابل تعریف در مساله
NRLNDX	حداکثر تعداد پارامتر های حقیقی قابل تعریف در مساله
NBMAX	حداکثر تعداد کران های بالا و پایین برای متغییر ها
NSMAX	حداکثر تعداد نقاط شروع قابل تعریف در مساله
NOBMAX	حداکثر تعداد کران های بالا و پایین برای تابع هدف در مساله

جدول ۲-۱: پارامتر های اندازه در SIFDec



شکل ۲-۳: افزایش NSETVC



شکل ۲-۴: کدگشایی مجدد NCB۲۰

فصل ۳

عملکرد CUTEr

پس از آن که مساله آزمون توسط SiFDec کد گشایی شد در اختیار بسته CUTEr قرار می گیرد. این بسته با ابزاری که در اختیار دارد امکان بهره برداری از مساله را برای کاربر فراهم می سازد. علاوه بر این کاربر می تواند از این بسته بعنوان یک رابط [۱] جهت اتصال به مسایل آزمون و استفاده از آنها در کد ها و نرم افزار های بهینه سازی بهره گیرد.

کاربر پس از نصب CUTEr می تواند مسایل آزمون را با دستور زیر دانلود کرده و در پوشه با مسیر پیش فرضی که در فایل bashrc در متغیر MASTSIF ذخیره می شود، مورد استفاده قرار دهد.

~$cd $MASTSIF
~$svn co http://tracsvn.mathappl.polymtl.ca/svn/sif/trunk

۳-۱ ابزار CUTEr

همانطور که در ابتدا نیز اشاره کردیم مسایل آزمون موجود در فرمت (SIF) پس از کد گشایی توسط SIFDec به فایل های قابل اجرا در فرترن تبدیل می شوند علاوه بر این CUTEr نیز مجهز به تعدادی کد اولیه کمکی در زبان فرترن می باشد که به کاربر اجازه می دهد که داده های موجود در مساله را در بسته های نرم افزاری مورد استفاده قرار دهد. به این کد های کمکی ابزار CUTEr گفته می شود که در مسیر زیر قابل دسترسی هستند.

$CUTER/common/src/tools

[۱] Interface

فهرستی از این ابزارها در جدول های ۳-۱ و ۳-۲ گرد آوری شده است. کاربر علاقه مند می تواند اطلاعات جامعی از هر یک از این ابزارها را در پوشه man به آدرس

$CUTER/common/man/man3

مطالعه کند.

بعنوان مثال با وارد کردن دستور زیر در Terminal

~$man cdh

اطلاعات جامعی از فایل cdh که بمنظور محاسبه هسی تابع لاگرانژی مورد استفاده قرار می گیرد، نمایش داده می شود.

۳-۲ اندازه های CUTEr

مشابه SIFDec بسته CUTEr نیز در سه اندازه کوچک، متوسط و بزرگ عرضه می شود که کاربر می تواند بر اساس نیاز هر اندازه دلخواه را با انتخاب بترتیب گزینه های S ، M و L در زمان نصب در اختیار داشته باشد. البته یک اندازه Custom نیز در نظر گرفته می شود که براساس انتخاب کاربر شکل می گیرد. ایتم های مشخص کننده اندازه در جدول ۳-۳ گرد آوری شده است. این اندازه ها از طریق فایل به آدرس

$MYCUTER/double/config/tools.size

قابل تغییرند.

مشابه SIFDec برای برخی مسایل ممکن است نیاز به تغییر برخی اندازه ها داشته باشیم. معمولا در صورت ناکافی بودن یک اندازه خاص CUTEr پیغامی مبنی بر افزایش آن نمایش می دهد. در صورت مشاهده چنین پیغامی اندازه مورد نظر را در فایل tools.size تغییر داده و سپس CUTEr را بازسازی کنید.

۳-۳ بازسازی CUTEr

پس از هرگونه ویرایش در فایل ها و جزییات CUTEr باید بمنظور اعمال تغییرات آن را بازسازی کرد. برای اینکار می توان از دستور Make all در مسیر $MYCUTER استفاده کرد. بعبارتی دستورات زیر را بترتیب در Terminal اجرا کنید.

~$ cd $MYCUTER

~$ make -s all

۳-۴ رابط های CUTEr

بسته CUTEr با ایجاد رابط با برخی نرم افزار های بهینه سازی و جبر خطی امکان استفاده از ابزارش را برای این نرم افزار ها فراهم ساخته است. برای دسترسی به اطلاعات مربوط به این رابط ها می توان به پوشه man۱ واقع در آدرس

$CUTER/common/man/man1

مراجعه کرد. جدول ۳-۴ نام بسته های نرم افزاری بهینه سازی به همراه نام رابط نظیر را نشان می دهد. بعنوان مثال برای نمایش اطلاعات مربوط به بسته CG-DESCENT کافیست دستور زیر را در Terminal وارد کنید.

~$ man cgd

توجه: برای استفاده از بسته های نرم افزاری در CUTEr باید به پوشه نظیر به آن در مسیر

$CUTER/common/src/pkg

مراجعه کرده و مطابق فایل راهنمای موجود، داده های مورد نیاز بسته را تهیه کنید. برای استفاده از بسته مورد نظر دز حل یک مساله می توانید از دستور runcuter استفاده کنید. فرض کنید هدف حل مساله KISSING۲ با بسته باشد در اینصورت با اجرای دستور زیر در Terminal خروجی مطابق شکل ۳-۱ خواهد بود.

~$ runcuter - -package gen - -decode KISSING2

```
 OPTIMAL SOLUTION FOUND
 THE ANSWER IS  41.9999999999999

********************** CUTEr statistics ***********************

Code used                :  GEN
Variant                  :  name of a variant, if needed
Problem                  :  KISSING2
# variables              =            100
# constraints            =            625
# linear constraints     =              0
# equality constraints   =              0
# inequality constraints =            625
# bounds                 =              6
# objective functions    =           0.00
# objective gradients    =           0.00
# objective Hessians     =           0.00
# constraints functions  =           0.00
# constraints gradients  =           0.00
# constraints Hessians   =           0.00
Exit code                =              0
Final f                  =   0.4200000E+02
Set up time              =           0.01 seconds
Solve time               =           0.00 seconds

***************************************************************
```

شکل ۳-۱: حل مساله KISSING۲ با بسته Gen

۳-۴-۱ دستور runcuter

اطلاعات مربوط به این دستور از طریق صفحه man قابل دسترس است.

```
~$ man runcuter
```

شکل کلی دستور بصورت

```
runcuter - -package package [- -single] [- -help] [- -keep] [- -rebuild] [- -output 0|1] [- -limit secs] [- -cfortran] [- -debug] [- -uncons] [-Lpath/to/lib] [- -blas keyword] [- -lapack keyword] [- -decode problem[.SIF]
```

در اینجا تنها ایتم های پرکاربرد معرفی می شود.

package – -: نام بسته بهینه سازی مورد استفاده در حل مساله پس از این ایتم قرار می گیرد.

decode – -: نام مساله ای که باید حل شود پس از این ایتم قرار می گیرد.

توجه: ایتم های که در لیست بالا نیست در صورت ورود مستقیما به دستور sifdecode منتقل می شود.

۳-۵ رابط CUTEr با MATLAB

برای استفاده از ابزار CUTEr در MATLAB یک رابط کاربری بنام mx طراحی شده که کاربر می تواند بعد از کد گشایی مساله از ابزار CUTEr بمنظور دسترسی به داده های مساله استفاده کند. برای آماده سازی مساله جهت استفاده در MATLAB کاربر باید ابتدا این رابط را فراخوانی کند. فرض کنید هدف استفاده از مساله KISSING۲ باشد با دستور زیر یک رابط بین ابزار CUTEr و MATLAB ایجاد می شود.

```
~$ runcuter –package mx –decode KISSING2
```

این رابط یک فایل بنام mcuter.mexglx در پوشه Home ایجاد می کند که واسط ارتباط کاربر با CUTEr در MATLAB است.

شایان ذکر است که دستور فوق را می توان با عملگر (!) مستقیما در پنجره window Command در MATLAB نیز وارد کرد.

```
» !runcuter –package mx –decode KISSING2
```

بمنظور بارگزاری مساله در حافظه از دستور زیر استفاده کنید. خروجی مطابق شکل ۳-۲ است.

```
» prob = cuter_ setup()
```

```
>> !runcuter --package mx --decode KISSING2
sifdecode  KISSING2

 Problem name: KISSING2

 Double precision version will be formed.

 The objective function uses        1 nonlinear group

 There are      625 nonlinear inequality constraints

 There are       94 free variables
 There are        6 fixed variables



Warning: You are using gcc version "5.2.1-22ubuntu2)".  Th
         with mex is "4.1".  The latest version tested for
         To download a different version of gcc, visit htt

>> prob=cuter_setup()

prob =

         n: 100
         m: 625
      nnzh: 1300
      nnzj: 4900
         x: [100x1 double]
        bl: [100x1 double]
        bu: [100x1 double]
         v: [625x1 double]
        cl: [625x1 double]
        cu: [625x1 double]
    equatn: [625x1 logical]
    linear: [625x1 logical]
      name: 'KISSING2  '
```

شکل ۳-۲: بارگزاری مساله KISSING۲

اطلاعات نمایش داده شده براساس شکل کلی مساله بهینه سازی بصورت زیر است.

$$\min f(x) \qquad s.t.\ \ c_l \leq c(x) \leq c_u,\ \ b_l \leq x \leq b_u.$$

متغیر prob یک structure است که خصوصیات مساله را در خود ذخیره کرده است. برای دسترسی به هر ویژگی کافیست از عملگر (.) استفاده شود. به مثال زیر توجه کنید.

» prob.n

ans=

100

بعبارتی تعداد متغییر های مساله که برابر ۱۰۰ است، نمایش داده می شود. جزییات موارد نمایش داده شده به شرح زیر است.

prob.n: تعداد متغیر های مساله.

prob.m: تعداد قیود مساله.

prob.nnzh: تعداد عناصر ناصفر در مثلث هسین تابع.

prob.nnzj: تعداد عناصر ناصفر در ژاکوبین.

prob.x: نقطه شروع اولیه پیش فرض در مساله.

prob.v: نقطه شروع اولیه پیش فرض برای ضرایب لاگرانژ در مساله.

prob.cl(u): بردار کران های پایین (بالا) روی قیود.

prob.bl(u): بردار کران های پایین (بالا) روی متغییرها.

prob.equatn: بردار منطقی نمایش دهنده قیود تساوی (True) و نامساوی (False) .

prob.linear: بردار منطقی نمایش دهنده قیود خطی (True) و غیر خطی (False) .

prob.name: نمایش نام مساله.

بعد از بارگزاری مساله کاربر می تواند از ابزار CUTEr برای بهره برداری از مساله استفاده کند. این ابزار بهمراه معادل تابع MATLAB آنها در ادامه گردآوری شده است.

cuter_ dims: (CUTEr Tools: udimen / cdimen)
Discover problem dimensions (may be called before cuter_ setup()) .

cuter_ setup: (CUTEr Tools: usetup / csetup)
Setup problem data structure.

cuter_ varnames : (CUTEr Tools: varnames)
Discover variable names.

cuter_ connames : (CUTEr Tools: connames)
Discover constraint names

cuter_ obj : (CUTEr Tools: uofg / cofg)
Evaluate objective function value and its gradient if requested

cuter_ objcons : (CUTEr Tools: cfn)
Evaluate objective and constraints

cuter_ cons : (CUTEr Tools: ccfg / ccifg)
Evaluate constraint bodies and their gradients if requested. Evaluate a single constraint value and its gradient if requested.

cuter_ scons : (CUTEr Tools: ccfsg / ccifsg)

Evaluate constraint bodies and Jacobian in sparse format. Evaluate a single constraint value and its gradient as a sparse vector.

cuter_ lagjac : (CUTEr Tools: cgr)

Evaluate Jacobian and gradient of either objective or Lagrangian.

cuter_ slagjac : (CUTEr Tools: csgr)

Evaluate Jacobian in sparse format and gradient of either objective or Lagrangian as a sparse vector.

cuter_ jprod : (CUTEr Tools: cjprod)

Evaluate the matrix-vector product between the Jacobian and a vector.

cuter_ jtprod : (CUTEr Tools: cjprod)

Evaluate the matrix-vector product between the transpose Jacobian and a vector.

cuter_ hess : (CUTEr Tools: udh / cdh)

Evaluate the Hessian matrix of the Lagrangian, or of the objective if the problem is unconstrained. The matrix is stored as a dense array.

cuter_ ihess : (CUTEr Tools: udh / cdh)

Evaluate the Hessian matrix of the i-th problem function (i=0 is the objective function), or of the objective if problem is unconstrained.

cuter_ hprod : (CUTEr Tools: uprod / cprod)

Evaluate the matrix-vector product between the Hessian of the Lagrangian (or the objective if unconstrained) and a vector.

cuter_ gradhess : (CUTEr Tools: ugrdh / cgrdh)

Evaluate the gradient of either the objective or the Lagrangian, the Jacobian (or its transpose) and the Hessian of the Lagrangian in dense format.

cuter_ sphess : (CUTEr Tools: ush / csh)

Evaluate the Hessian matrix of the Lagrangian, or of the objective if the problem is unconstrained, in sparse format.

cuter_ spihess : (CUTEr Tools: ush / csh)

Evaluate the Hessian matrix of the i-th problem function (i=0 is the objective function), or of the objective if problem is unconstrained, in

sparse format.

برای آشنایی از روش بکارگیری هر تابع در MATLAB می توانید از دستور help استفاده کنید. بعنوان مثال

» help cuter_ cons

Return constraint bodies and Jacobian if requested.

or return a single constraint value and its gradient if requested

Usage: c = cuter_ cons(x) or [c,J] = cuter_ cons(x)

ci = cuter_ cons(x,i) or [ci,gi] = cuter_ cons(x,i)

فرض کنید هدف محاسبه مقدار تابع و گرادیان مساله KISSING۲ در نقطه پیش فرض prob.x باشد. با ورود دستور زیر

» [f,g]= prob_ obj(prob.x)

مقدار تابع هدف و گرادیانش در نقطه پیش فرض، بترتیب در متغییرهای f و g ذخیره می شود.

از دستور runcuter می توان بطور هم زمان برای تغییر بعد مساله و ایجاد رابط با بسته های نرم افزاری استفاده کرد. در مثال زیر مساله NCB۲۰ با بعد ۱۰۰ کدگشایی و سپس رابط MATLAB آن فراخوانی می شود.

~$ runcuter - -package mx -param N=100 - -decode NCB20

Tool	name Brief description
ubandh	extract a banded matrix out of the Hessian matrix,
udh	evaluate the Hessian matrix,
udimen	get the number of variables involved,
udimse	determine the number of nonzeros required to store the sparse Hessian matrix in finite element format,
udimsh	same as udimse, in coordinate format,
ueh	evaluate the sparse Hessian matrix in finite element format,
ufn	evaluate function value,
ugr	evaluate gradient,
ugrdh	evaluate the gradient and Hessian matrix,
ugreh	evaluate the gradient and Hessian matrix in finite element format,
ugrsh	evaluate the gradient and Hessian matrix in coordinate format,
unames	obtain the names of the problem and its variables,
uofg	evaluate function value and possibly gradient,
uprod	form the matrix-vector product of a vector with the Hessian matrix,
usetup	set up the data structures for unconstrained minimization,
ush	evaluate the sparse Hessian matrix,
uvarty	determine the type of each variable.
ureprt	obtain statistics concerning function evaluation and CPU time used,

جدول ۳-۱: ابزار برای مسایل نامقید

Tool	name Brief description
ccfg	evaluate constraint functions values and gradients,
ccfsg	same as ccfg, in sparse format,
ccifg	evaluate a single constraint function value and gradient,
ccifsg	same as ccifg, in sparse format,
cdh	evaluate the Hessian of the Lagrangian,
cdimen	get the number of variables and constraints ,
cdimse	determine number of nonzeros to store the Lagrangian Hessian in finite element format,
cdimsh	same as cdimse but in coordinate format,
cdimsj	determine number of nonzeros to store the matrix of gradients and constraints, in sparse format,
ceh	evaluate the sparse Lagrangian in finite element format,
cfn	evaluate function and constraints values,
cgr	evaluate constraints gradients and Lagrangian gradient,
cgrdh	same as cgr, plus Lagrangian Hessian,
cidh	evaluate the Hessian of a problem function,
cish	same as cidh, in sparse format,
cnames	obtain the names of the problem and its variables,
cofg	evaluate function value and possibly gradient,
cprod	form the matrix-vector product of a vector with the Lagrangian Hessian,
cscfg	evaluate constraint functions values and possibly gradients in sparse format,
cscifg	same as cscfg, for a single constraint,
csetup	set up the data structures for cons minimization,
csgr	evaluate cons and obje/Lag function gradients,
csgreh	evaluate both the cons grad/Lag in finite element format
csgrsh	same as csgreh, in sparse format,
csh	evaluate the Hessian of the Lagrangian, in sparse format,
cvarty	determine the type of each variable,
creprt	obtain statistics about func eval/CPU time ,

جدول ۳-۲: ابزار برای مسایل مقید

پارامتر	شرح
LIWK	اندازه فضای کار صحیح مورد استفاده توسط الگوریتم
LWK	اندازه فضای کار مورد استفاده در فرمت ساده یا مضاعف
LLOGIC	اندازه فضای کار منطقی مورد استفاده توسط الگوریتم
LCHARA	تعداد رشته های مورد استفاده در الگوریتم
LFUVAL	اندازه بردار جهت زخیره سازی تابع هدف و مشتقات.

جدول ۳-۳: اندازه های CUTEr

Interface	Package
cgd/sdcgd	CG Descent (Hager and Zhang)
cgp/sdcgp	CG+ (Liu, Nocedal and Waltz)
cob/sdcob	COBYLA (Powell)
fil/sdfil	FilterSQP (Fletcher and Leyffer)
gen/sdgen	Generic Fortran 77 interface
gen90/sdgen90	Generic Fortran 90 interface
genc/sdgenc	Generic C/C++ interface
hrb/sdhrb	SIF–Harwell- or Rutherford-Boeing
ipopt/sdipopt	IPOPT (Wachter)
knit/sdknit	KNITRO (Byrd, Nocedal and Waltz)
la04/sdla04	LA04 (Reid)
lbb/sdlbb	L-BFGS-B (updated) (Nocedal)
lbs/sdlbs	L-BFGS (Nodedal)
lmb/sdlmb	L-BFGS-B (Nocedal)
lqo/sdlqo	LOQO (Benson, Shanno and Vanderbei)
mns/sdmns	MINOS (Murtagh and Saunders)
nits/sdnits	NITSOL (Pernice and Walker)
nps/sdnps	NPSOL (Gill, Murray, Saunders and Wright)
osl/sdosl	OSL (IBM)
pds/sdpds	PDS (Torczon)
prx/sdprx	PRAXIS (Brent and Chandler)
snp/sdsnp	SNOPT (Gill, Murray and Saunders)
stn/sdstn	Stenmin (Bouaricha)
tao/sdtao	TAO (Benson, Curfman McInnes, More and Sarich)
ten/sdten	Tenmin (Schnabel and Chow)
trn/sdtrn	TRON (Lin and Mor´e)
unc/sdunc	Uncmin (Koontz, Schnable and Weiss)
va15/sdva15	VA15 (Nocedal)
ve09/sdve09	VE09 (Gould)
ve12/sdve12	HSL VE12 (Gould)
ve14/sdve14	VE14 (Gould)
vf13/sdvf13	VF13 (Powell)

جدول ۳-۴: رابط های CUTEr

مراجع

[1] N. I. M. Gould , D. Orban and Ph.L. Toint, *General CUTEr documentation*, CERFACS Technical Report TR/PA/02/13, 2005.

[2] N. I. M. Gould , D. Orban and Ph.L. Toint, *General SifDec documentation*, CERFACS Technical Report TR/PA/02/14, 2003.