

آشنایی با محیط مسایل آزمون CUTer

نصب و راه اندازی

تألیف

مسعود فاطمی

استادیار دانشکده علوم ریاضی

دانشگاه صنعتی خواجه نصیرالدین طوسی

فهرست مطالب

سه	فهرست تصاویر
پنج	فهرست جداول
۳	فصل ۱ نصب و راه اندازی
۳	۱-۱ نصب SIFDec و CUTEr
۳۳	فصل ۲ عملکرد SIFDec
۳۳	۱-۲ بررسی عملکرد sifdecoder
۳۵	۲-۲ اندازه های SIFDec
۳۶	۳-۲ بازسازی SIFDec
۳۹	فصل ۳ عملکرد CUTEr
۳۹	۱-۳ ابزار CUTEr
۴۰	۲-۳ اندازه های CUTEr
۴۰	۳-۳ بازسازی CUTEr
۴۱	۴-۳ رابط های CUTEr
۴۲	۱-۴-۳ دستور runcuter
۴۲	۵-۳ رابط CUTEr با MATLAB
۵۱	مراجع

فهرست تصاویر

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

۱۹	۲۴-۱ نصب Sifdec ۳ (ادامه)
۲۰	۲۵-۱ خطا زدایی ۳
۲۰	۲۶-۱ کدگشایی مساله KISSING
۲۱	۲۷-۱ ویرایش فایل C.arch
۲۲	۲۸-۱ نصب CUTEr ۱
۲۳	۲۹-۱ نصب CUTEr ۲
۲۴	۳۰-۱ نصب CUTEr ۳
۲۵	۳۱-۱ حل مساله ROSENBR با بسته gen
۲۵	۳۲-۱ نصب MATLAB ۱
۲۶	۳۳-۱ خطا زدایی ۱
۲۶	۳۴-۱ خطا زدایی ۲
۲۶	۳۵-۱ خطا زدایی ۳
۲۷	۳۶-۱ خطا زدایی ۴
۲۷	۳۷-۱ خطا زدایی ۵
۲۸	۳۸-۱ ویرایش Mexopts.sh
۲۸	۳۹-۱ ویرایش Mx
۲۹	۴۰-۱ راه اندازی مجدد Mex
۲۹	۴۱-۱ آزمون CUTEr
۳۰	۴۲-۱ خطازدایی ۱
۳۰	۴۳-۱ خطازدایی ۲
۳۱	۴۴-۱ خطازدایی ۳
۳۵	۱-۲ اندازه های SIFDec
۳۶	۲-۲ کدگشایی NCB۲۰
۳۸	۳-۲ افزایش NSETVC
۳۸	۴-۲ کدگشایی مجدد NCB۲۰
۴۱	۱-۳ حل مساله KISSING۲ با بسته Gen
۴۳	۲-۳ بارگزاری مساله KISSING۲

فهرست جداول

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

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 کارا تر 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 ارائه 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.

^۱Platform

^۲Unix

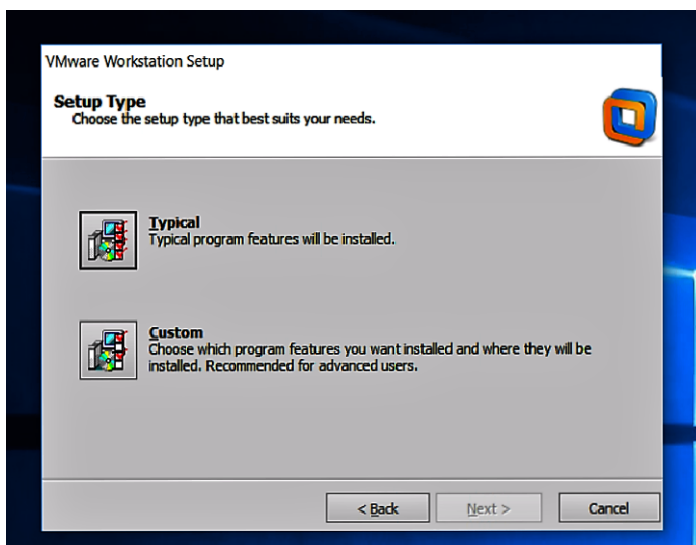
^۳Linux

^۴Mac

^۵Virtual

^۶Troubleshooting

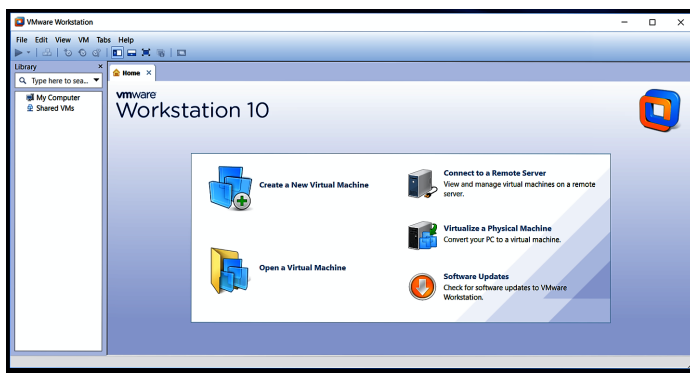
^۷Virtual Machine



شکل ۱-۱: نصب VMware Workstation گام ۱

گام ۱: 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.

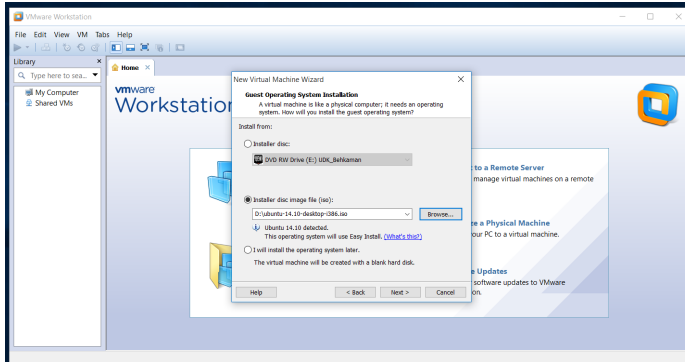
گام ۲: برنامه را اجرا کنید تا مطابق شکل ۲-۱ پنجره اصلی نرم افزار نمایان شود. سپس گزینه Create a new virtual machine را انتخاب کنید.



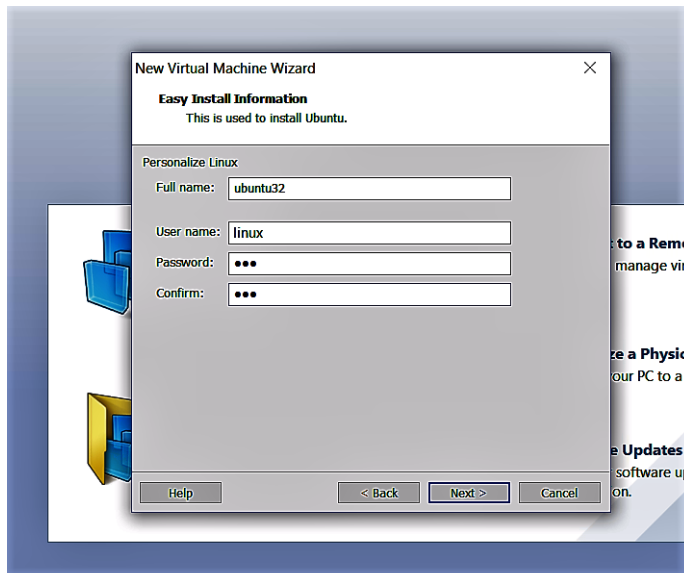
شکل ۲-۱: تولید سیستم عامل مجازی جدید

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

یعنی پنجره Guest operating system installation می رسم.



شکل ۱-۳: نصب سیستم مجازی

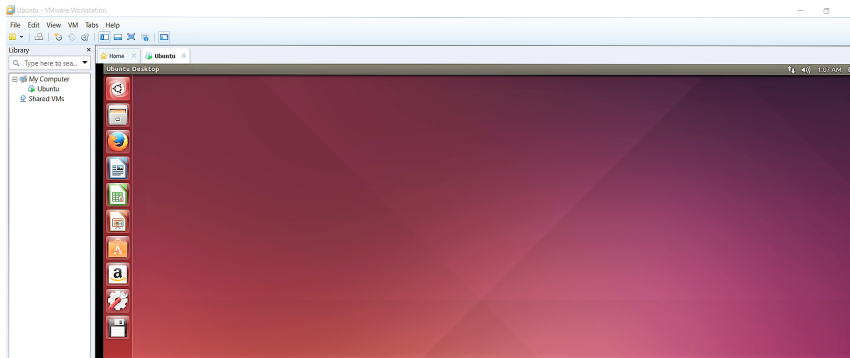


شکل ۱-۴: تعریف شناسه ورود

```
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

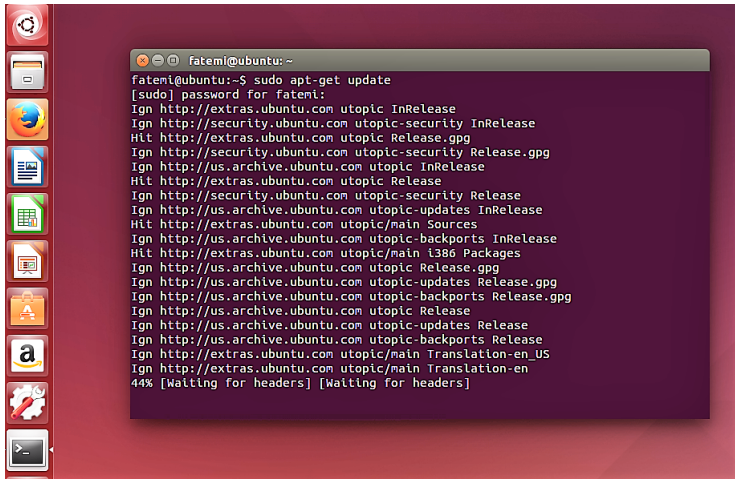
ISO حاوی این سیستم عامل را در گزینه `Installer disc image file` وارد می کنیم و `Next` را انتخاب کنید. لازم به ذکر است که ابنتو لینوکس یک نرم افزار متن باز^۴ بوده و فایل ISO آن از طریق وبگاه www.ubuntu.com قابل دریافت است.

گام ۴: 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**

^۴Open Source



```

fatemi@ubuntu:~$ sudo apt-get update
[sudo] password for fatemi:
Ign http://extras.ubuntu.com utopic InRelease
Ign http://security.ubuntu.com utopic-security InRelease
Hit http://extras.ubuntu.com utopic Release.gpg
Ign http://security.ubuntu.com utopic-security Release.gpg
Ign http://us.archive.ubuntu.com utopic InRelease
Hit http://extras.ubuntu.com utopic Release
Ign http://security.ubuntu.com utopic-security Release
Ign http://us.archive.ubuntu.com utopic-updates InRelease
Hit http://extras.ubuntu.com utopic/main Sources
Ign http://us.archive.ubuntu.com utopic-backports InRelease
Hit http://extras.ubuntu.com utopic/main i386 Packages
Ign http://us.archive.ubuntu.com utopic Release.gpg
Ign http://us.archive.ubuntu.com utopic-updates Release.gpg
Ign http://us.archive.ubuntu.com utopic-backports Release.gpg
Ign http://us.archive.ubuntu.com utopic Release
Ign http://us.archive.ubuntu.com utopic-updates Release
Ign http://us.archive.ubuntu.com utopic-backports Release
Ign http://extras.ubuntu.com utopic/main Translation-en_US
Ign http://extras.ubuntu.com utopic/main Translation-en
44% [Waiting for headers] [Waiting for headers]

```

شکل ۱-۷: بروز سازی نرم افزارها ۲

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

گام ۵: 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.

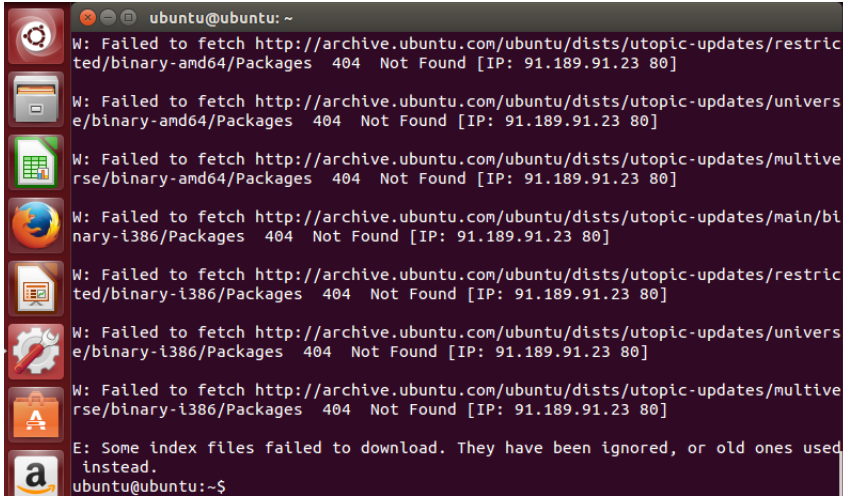
```
~$ sudo apt-get update
```

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. عبارتی 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 ۸-۱: خطا در بهنگام سازی

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

پس از تغییر سرور گام ۵ را دوباره انجام دهید.

گام ۶: در این گام باید Subversion روی سیستم نصب شود. برای اینکار پنجره Terminal را باز کرده و مطابق شکل ۱۱-۱ 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 ۱۱-۱ مراحل انجام می شود.

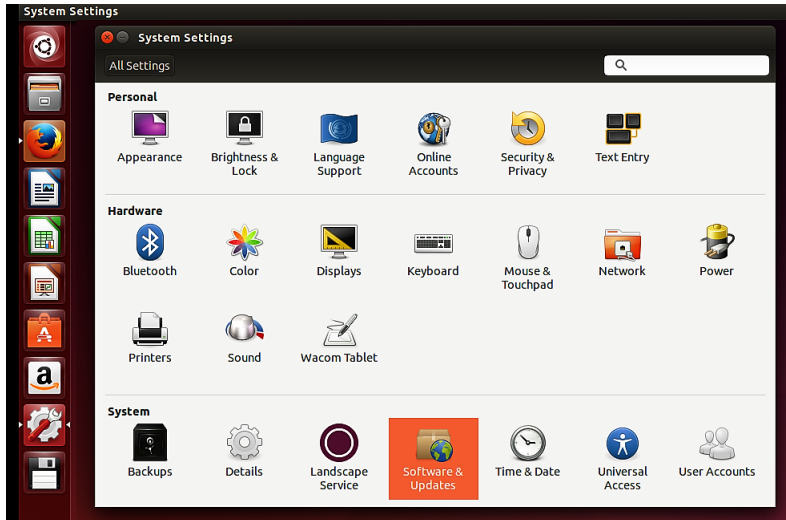
گام ۷: در این گام به ایجاد پوشه نصب می پردازیم. بعد از باز کردن ترمینال مطابق شکل ۱۲-۱ عبارات زیر را جلو اعلان لینوکس تایپ کرده اینتر کنید.

```
~$ mkdir $HOME/CUTErInstallation
```

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

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

گام ۸: پس از اجرای دستورات گام ۷ فایل bashrc بجهت ویرایش در پنجره gedit باز می شود. مطابق شکل ۱۳-۱ دستورات در کادر قرمز را با دقت در انتهای این

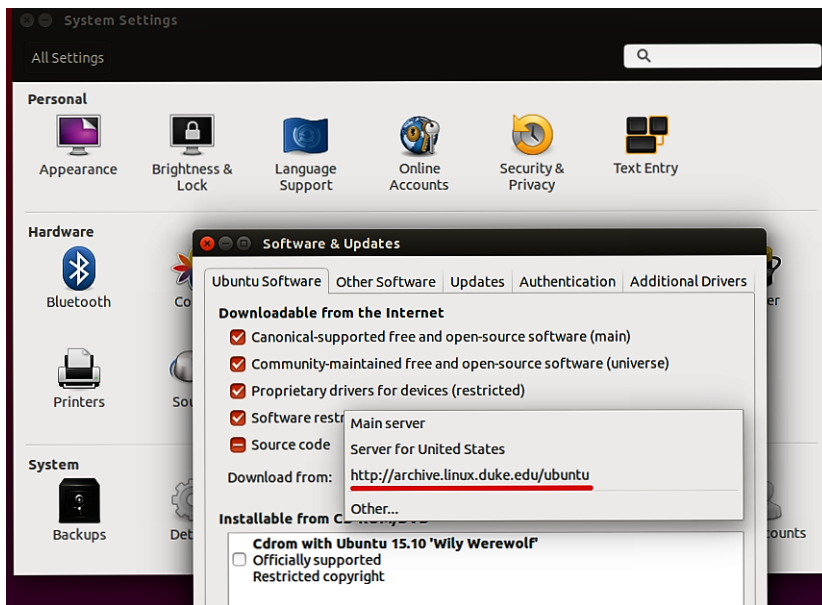


شکل ۱-۹: خطا در بهنگام سازی

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
```

گام ۹: حال به نصب کامپایلر فرتن 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/
```

گام ۹: 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.

```

linux@ubuntu: ~
linux@ubuntu:~$ sudo apt-get install subversion
[sudo] password for linux:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1
Suggested packages:
  subversion-tools db5.3-util
The following NEW packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1 subversion
0 upgraded, 5 newly installed, 0 to remove and 13 not upgraded.
Need to get 1,551 kB of archives.
After this operation, 5,490 kB of additional disk space will be used
Do you want to continue? [Y/n] y
Get:1 http://archive.linux.duke.edu/ubuntu/ wily/main libapr1 i386 1
  kB]
Get:2 http://archive.linux.duke.edu/ubuntu/ wily/main libaprutil1 i3
 83.6 kB]
Get:3 http://archive.linux.duke.edu/ubuntu/ wily/main libserf-1-1 i3
 49.4 kB]
Get:4 http://archive.linux.duke.edu/ubuntu/ wily/main libsvn1 i386 1
 039 kB]
75% [4 libsvn1 932 kB/1,039 kB 90%]
5

```

شکل ۱-۱۱: نصب Subversion

```
~$ cd $CUTERDIR
```

```
~$ svn co http://tracsvn.mathappl.polymtl.ca/SVN/cuter/cuter
/branches/CUTER2 ./cuter2
```

گام ۱۰: 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.

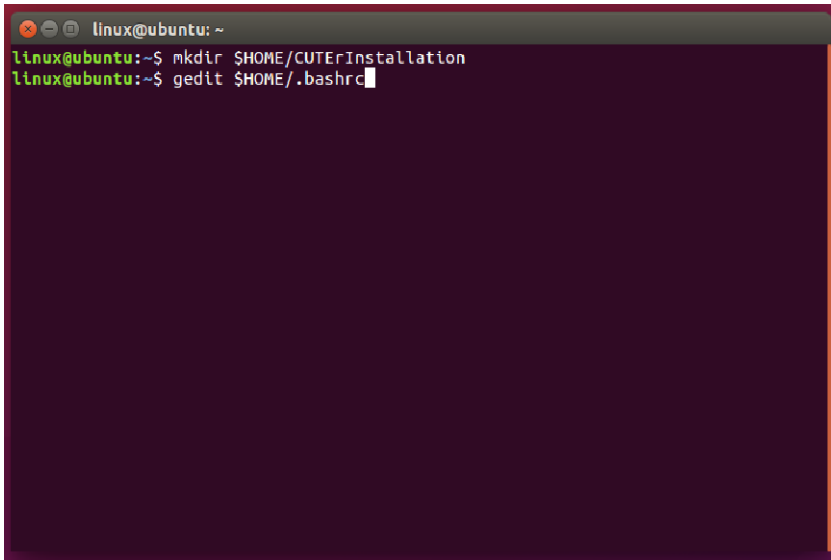
```
~$ cd $SIFDEC
```

```
~$ ./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 مجدداً 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 مجدداً Y را در باقیمانده مراحل انتخاب کنید. اگر همه چیز بدرستی انجام شود با خروجی مطابق شکل ۱-۲۴ پیروسه نصب خاتمه می یابد.

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

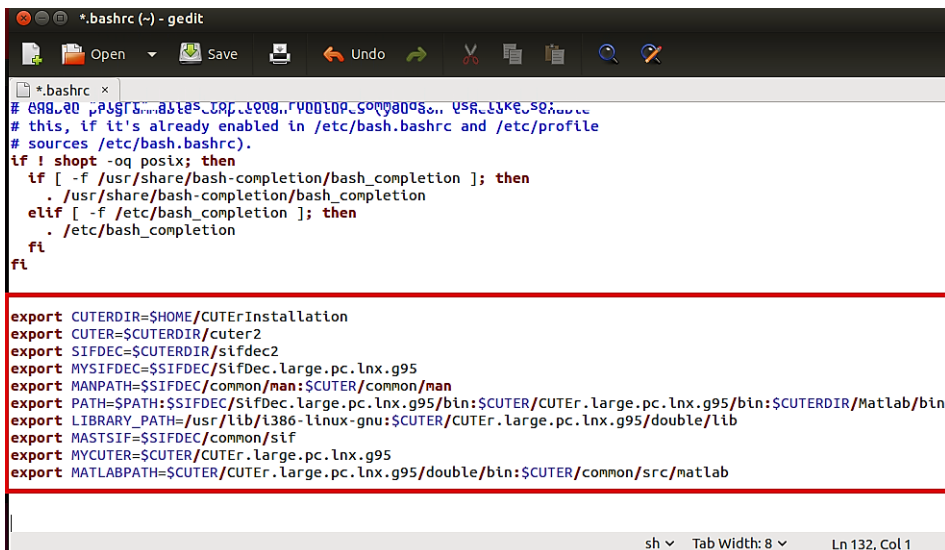
```
ld: cannot find crt1.o: no such file or directory
```

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

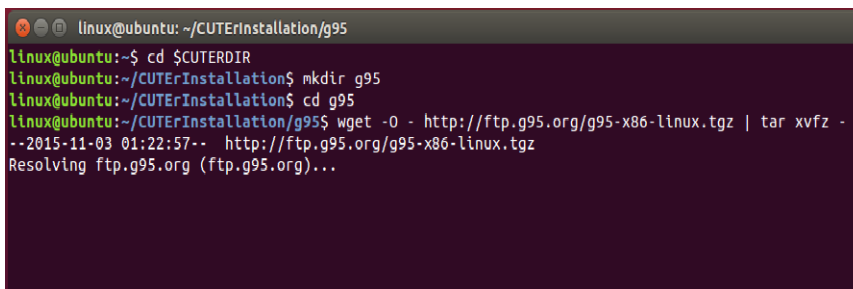


```

*.bashrc (-) - gedit
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

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
  
```

شکل ۱-۱۳: ویرایش فایل bashrc



```

linux@ubuntu: ~/CUTERInstallation/g95
linux@ubuntu:~$ cd $CUTERDIR
linux@ubuntu:~/CUTERInstallation$ mkdir g95
linux@ubuntu:~/CUTERInstallation$ cd g95
linux@ubuntu:~/CUTERInstallation/g95$ wget -O - http://ftp.g95.org/g95-x86-linux.tgz | tar xvfz -
--2015-11-03 01:22:57-- http://ftp.g95.org/g95-x86-linux.tgz
Resolving ftp.g95.org (ftp.g95.org)...
  
```

شکل ۱-۱۴: نصب کامپایلر فرتن

خروجی باید مطابق شکل ۱-۲۶ باشد.

گام ۱۱: حال نوبت به نصب CUTER رسیده است. قبل از نصب باید تغییری جزئی 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

```
linux@ubuntu: ~
linux@ubuntu:~$ sudo ln -s $CUTERDIR/g95/g95-install/bin/i686-pc-linux-gnu-g95 /usr/bin/g95
[sudo] password for linux:
linux@ubuntu:~$ sudo cp -R $CUTERDIR/g95/g95-install/lib/gcc-lib/ /usr/local/lib/
linux@ubuntu:~$
```

شکل ۱-۱۵: تعریف مسیر کامپایلر برای لینوکس

```
linux@ubuntu: ~
linux@ubuntu:~$ svn co http://tracsvn.mathappl.polytml.ca/SVN/cuter/sifdec/branches/SifDec2 ./sifdec2
A sifdec2/build
A sifdec2/build/scripts
A sifdec2/build/scripts/slash.sed
A sifdec2/build/scripts/readme.mysifdecbin
A sifdec2/build/scripts/protect.sed
A sifdec2/build/scripts/flush.sed
A sifdec2/build/scripts/umakefile.mysifdecbin
```

شکل ۱-۱۶: دانلود SifDec

all ; all ; gcc-4.1 ; -Dlsgcc ; 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 ۵، ۲، ۶، ۳، D، L، و Y 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 جزئی از

^۹Home directory

```

linux@ubuntu: ~
linux@ubuntu:~$ svn co http://tracsvn.mathappl.polytml.ca/SVN/cuter/cuter/branches/CUTEr2 ./cuter2
A   cuter2/build
A   cuter2/build/scripts
A   cuter2/build/scripts/unakefile.mycuterbin

```

شکل ۱-۱۷: دانلود CUTEr

```

linux@ubuntu: ~/CUTErInstallation/sifdec2
linux@ubuntu:~$ cd $SIFDEC
linux@ubuntu:~/CUTErInstallation/sifdec2$ ./install_sifdec

```

شکل ۱-۱۸: نصب Sifdec

پروژه نصب CUTEr نیست ولی بجهت بروز برخی خطا های رایج در هنگام نصب، این پروژه در اینجا با جزئیات 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 پروژه 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

```

linux@ubuntu: ~/CUTErInstallation/sifdec2
linux@ubuntu:~$ cd $SIFDEC
linux@ubuntu:~/CUTErInstallation/sifdec2$ ./install_sifdec
Select platform

(1) Compaq (DEC) alpha
(2) Cray
(3) HP workstation
(4) IBM RS/6000
(5) PC
(6) SGI workstation
(7) SUN workstation
(8) Mac OSX
(0) Other (user-defined)
5
Select operating system

(1) Windows 2000/XP with MinGW/Msys
(2) Linux
2
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

```

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 ۱-۳۵ مواجه 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 مواجهه 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، مجدداً 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.6 مواجه شدید دستور زیر را وارد کنید و گام ۱۲ را مجدداً تکرار کنید.

```

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

```



```
linux@ubuntu: ~
linux@ubuntu:~$ sudo ln -s /usr/bin/awk /usr/bin/gawk
```

شکل ۱-۲۲: خطا زدایی ۲

```
linux@ubuntu: ~/CUTerInstallation/sifdec2
Using directory /home/linux/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95

Logging operations [Ok]
Creating directory structure [Ok]
Copying Umakefiles [Ok]
Creating installation shortcut [Ok]
Installing README files [Ok]

Please do not forget to define the environment variable
[1] SIFDEC to be
    /home/linux/CUTerInstallation/sifdec2
[2] MYSIFDEC to be
    /home/linux/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95

as well as setting an appropriate value for MASTSIF,
which should point to your main SIF repository.

In addition, please update your MANPATH to include
    /home/linux/CUTerInstallation/sifdec2/common/man
and your PATH to include
    /home/linux/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95/bin

(see /home/linux/CUTerInstallation/sifdec2/common/doc/readme.cshrc
and /home/linux/CUTerInstallation/sifdec2/common/doc/readme.bashrc
for examples on how to do this)

cd /home/linux/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95,
read the README_FIRST and README files
then run install_mysifdec.

install_sifdec : Do you want install_mysifdec to be run in
/home/linux/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95 now (Y/n)?
Y
```

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. عبارتی FC را به 'g۹۵' 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. کافیسست 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

```
linux@ubuntu: ~/CUTerInstallation/sifdec2
cleaning in ./double/bin...
cleaning in ./double/config...
making all in ./bin...
Linking      slct                                ld: cannot find crt1.o: No such file or
directory
ld: cannot find crt1.o: No such file or directory
Makefile:177: recipe for target 'slct' failed
make[1]: *** [slct] Error 1
making all in ./double...
making all in ./double/bin...
```

شکل ۱-۲۵: خطا زدایی ۳

```
linux@ubuntu: ~
linux@ubuntu:~$ 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
```

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

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

error: expected expression 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

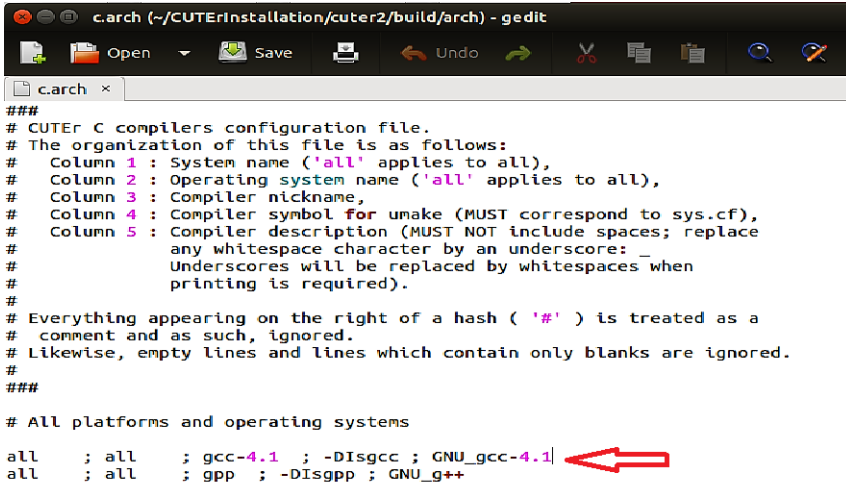
سپس فایل mcuter.c را برای ویرایش با دستور زیر باز کنید.

~\$ 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,



```

###
# CUTER C compilers configuration file.
# The organization of this file is as follows:
# Column 1 : System name ('all' applies to all),
# Column 2 : Operating system name ('all' applies to all),
# Column 3 : Compiler nickname,
# Column 4 : Compiler symbol for umake (MUST correspond to sys.cf),
# Column 5 : Compiler description (MUST NOT include spaces; replace
# any whitespace character by an underscore: _
# Underscores will be replaced by whitespaces when
# printing is required).
#
# Everything appearing on the right of a hash ( '#' ) is treated as a
# comment and as such, ignored.
# Likewise, empty lines and lines which contain only blanks are ignored.
#
###

# All platforms and operating systems

all ; all ; gcc-4.1 ; -DIsgcc ; GNU_gcc-4.1
all ; all ; gpp ; -DIsgpp ; GNU_g++

```

شکل ۱-۲۷: ویرایش فایل C.arch

”iscons:...icon);

mexErrMsgTex(msgBuf);

را غیر فعال کنید.

```
linux@ubuntu: ~/CUTerInstallation/cuter2
linux@ubuntu:~$ cd $CUTER
linux@ubuntu:~/CUTerInstallation/cuter2$ ./install_cuter
Select platform
(1) Compaq (DEC) alpha
(2) Cray
(3) HP workstation
(4) IBM RS/6000
(5) PC
(6) SGI workstation
(7) SUN workstation
(8) Mac OSX
(0) Other (user-defined)
5
Select operating system
(1) Windows 2000/XP with MinGW/Msys
(2) Linux
2
Select Fortran compiler
[1 ] Fujitsu frt
[2 ] Intel ifort
[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 C compiler
[0 ] No C extension to CUTEr
[1 ] Intel icc
[2 ] Portland Group pgcc
[3 ] GNU gcc-4.1
[4 ] GNU g++
3
```

شکل ۱-۲۸: نصب CUTEr ۱

```
linux@ubuntu: ~/CUTErInstallation/cuter2
(6) SGI workstation
(7) SUN workstation
(8) Mac OSX
(0) Other (user-defined)
5
Select operating system

(1) Windows 2000/XP with MinGW/Msys
(2) Linux
2
Select Fortran compiler
[1 ] Fujitsu frt
[2 ] Intel ifort
[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 C compiler
[0 ] No C extension to CUTEr
[1 ] Intel icc
[2 ] Portland Group pgcc
[3 ] GNU gcc-4.1
[4 ] GNU g++
3
Using C compiler: GNU_gcc-4.1
Set install precision (D=double, s=single):
D
Set install size (L=large, m=medium, s=small, c=customized):
L
By default, CUTEr with your selections will be installed in
/home/linux/CUTErInstallation/cuter2/CUTEr.large.pc.lnx.g95
Is this OK (Y/n)?
```

شکل ۱-۲۹: نصب CUTEr ۲

```

linux@ubuntu: ~/CUTerInstallation/cuter2
Building ubandh.o [Ok]
Building udh.o [Ok]
Building udimen.o [Ok]
Building udimse.o [Ok]
Building udimsh.o [Ok]
Building ueh.o [Ok]
Building ufn.o [Ok]
Building ugr.o [Ok]
Building ugrdh.o [Ok]
Building ugrsh.o [Ok]
Building unames.o [Ok]
Building uofg.o [Ok]
Building uprod.o [Ok]
Building ureprt.o [Ok]
Building useup.o [Ok]
Building ush.o [Ok]
Building uvarty.o [Ok]
Building varnames.o [Ok]
Building ccuter.o [Ok]
Building ffilesop.o [Ok]
making all in /double/cuter-specs...
cleaning in ./bin...
cleaning in ./double...
cleaning in /double/bin...
cleaning in /double/config...
cleaning in /double/lib...
cleaning in /double/cuter-specs...

If all your environment variables are properly set,
you can test your installation using the command:
  runcuter -p gen -D ROSENBR

(Note that once your PATH has been updated you need
to use the rehash command to make it active.)

[Installation complete]

-----
linux@ubuntu:~/CUTerInstallation/cuter2$

```

شکل ۱-۳۰: نصب CUTer ۳

```

linux@ubuntu:~$ runcuter -p gen -D ROSENBR
ifdecode ROSENBR

Problem name: ROSENBR

Double precision version will be formed.

The objective function uses      1 linear group
The objective function uses      1 nonlinear group

There are      2 free variables

Building      genma.o      [Ok]
*****
*
*      HELLO FROM GEN!      *
*      (DOUBLE PRECISION)  *
*
*****

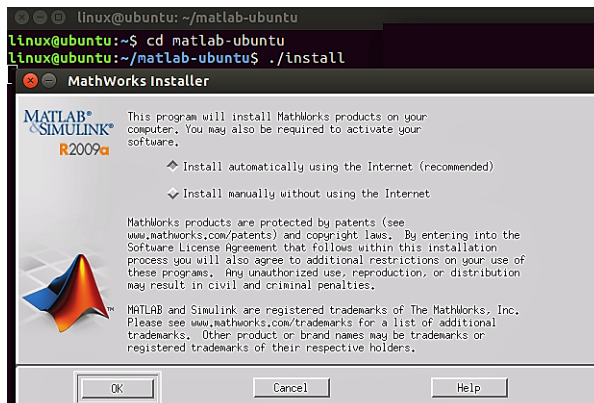
OPTIMAL SOLUTION FOUND
THE ANSWER IS  41.9999999999999

***** CUTer statistics *****

Code used      :      GEN
Variant       :      name of a variant, if needed
Problem       :      ROSENBR
# variables   =      2
# constraints =      0
# linear constraints =      0
# equality constraints =      0
# inequality constraints =      0
# bounds     =      0
# 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  =      0.00 seconds

```

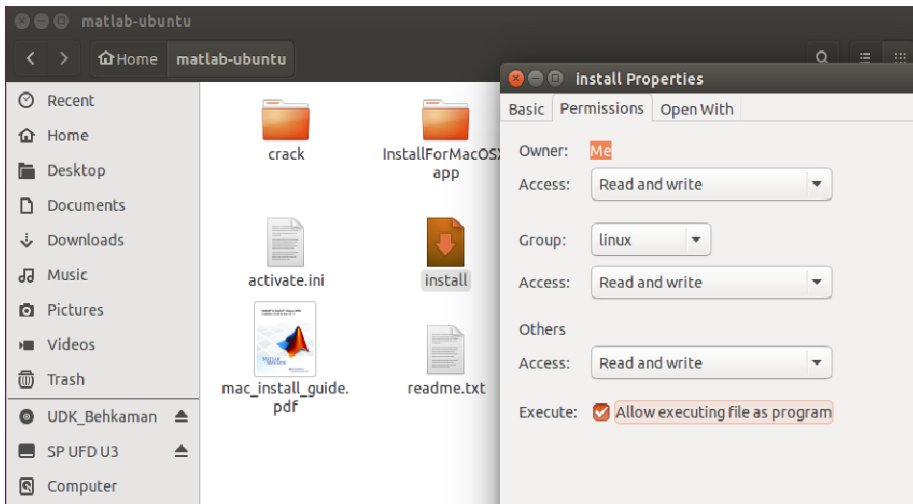
شکل ۱-۳۱: حل مساله ROSENBR با بسته جن



شکل ۱-۳۲: نصب MATLAB


```
linux@ubuntu: ~/matlab-ubuntu
linux@ubuntu:~$ cd matlab-ubuntu
linux@ubuntu:~/matlab-ubuntu$ ./install
bash: ./install: Permission denied
linux@ubuntu:~/matlab-ubuntu$
```

شکل ۱-۳۳: خطا زدایی ۱



شکل ۱-۳۴: خطا زدایی ۲

```
linux@ubuntu: ~/matlab-ubuntu
linux@ubuntu:~/matlab-ubuntu$ ./install
bash: ./install: Permission denied
linux@ubuntu:~/matlab-ubuntu$ ./install
./install: 1: ./install: /lib/libc.so.6: not found
-----
An error status was returned by the program 'xsetup',
the X Window System version of 'install'. The following
messages were written to standard error:

./install: 86: /home/linux/matlab-ubuntu/update/install/main.sh: /home/l
inux/matlab-ubuntu/update/bin/glnx86/xsetup: Permission denied

Attempt to fix the problem and try again. If X is not available
or 'xsetup' cannot be made to work then try the terminal
version of 'install' using the command:

install* -t or INSTALL* -t
-----

Sorry! Setup aborted . . .
```

شکل ۱-۳۵: خطا زدایی ۳

```

linux@ubuntu: ~/matlab-ubuntu
linux@ubuntu:~$ sudo ln -s /lib/i386-linux-gnu/libc.so.6 /lib/libc.so.6
[sudo] password for linux:
linux@ubuntu:~$ cd matlab-ubuntu
linux@ubuntu:~/matlab-ubuntu$ ./install
-----
An error status was returned by the program 'xsetup',
the X Window System version of 'install'. The following
messages were written to standard error:

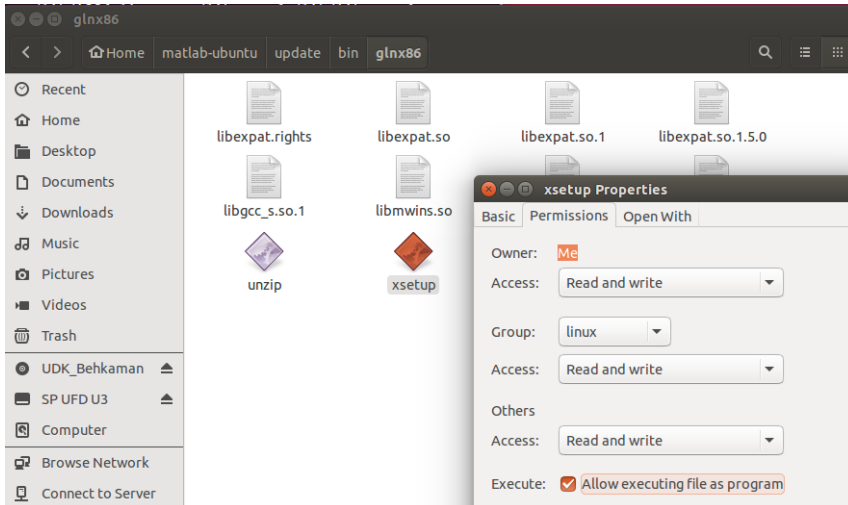
./install: 86: /home/linux/matlab-ubuntu/update/install/main.sh: /home/l
inux/matlab-ubuntu/update/bin/glnx86/xsetup: Permission denied

Attempt to fix the problem and try again. If X is not available
or 'xsetup' cannot be made to work then try the terminal
version of 'install' using the command:

install* -t or INSTALL* -t
-----

```

شکل ۱-۳۶: خطا زدایی ۴



شکل ۱-۳۷: خطا زدایی ۵

```

mexopts.sh (-/CUTErInstallation/Matlab/bin) - gedit
mx x mexopts.sh x
CDEBUGFLAGS='-g'
CLIBS="$CLIBS -lstdc++"
#
# C++keyName: GNU C++
# C++keyManufacturer: GNU
# C++keyLanguage: C++
# C++keyVersion:
CXX='g++'
CXXFLAGS='-ansi -D_GNU_SOURCE'
CXXFLAGS="$CXXFLAGS -D_FILE_OFFSET_BITS=64"
CXXFLAGS="$CXXFLAGS -fPIC -pthread"
CXXLIBS="$SRPATH $MLIBS -ln"
CXXOPTIMFLAGS='-O -DNDEBUG'
CXXDEBUGFLAGS='-g'
#
#
# FortrankeyName: g95
# FortrankeyManufacturer: GNU
# FortrankeyLanguage: Fortran
# FortrankeyVersion:
FC='g95'
FFLAGS='-fexceptions'
FFLAGS="$FFLAGS -fPIC"
FLIBS="$SRPATH $MLIBS -ln"
FOPTIMFLAGS='-O'
FDEBUGFLAGS='-g'
#
#
LD="$FC"
LDEXTENSION='.mexglx'
LDLFLAGS="-pthread -shared -m32 -Wl,--version-script,$STMW_ROOT/exte
undefined"
LDOPTIMFLAGS='-O'
LDDEBUGFLAGS='-g'
#
#
POSTLINK_CMDS=':'
#-----
;;
glnxa64)
..

```

شکل ۱-۳۸: ویرایش Mexopts.sh

```

mx x
#!/bin/bash
# Definitions for the MATLAB package
# N. Gould, D. Orban & Ph. Totnt

# Define a short acronym for the package
export PACK=mx

# Subdirectory of ${CUTER}/common/src/pkg where the package lives
export PACKAGE=mexfile

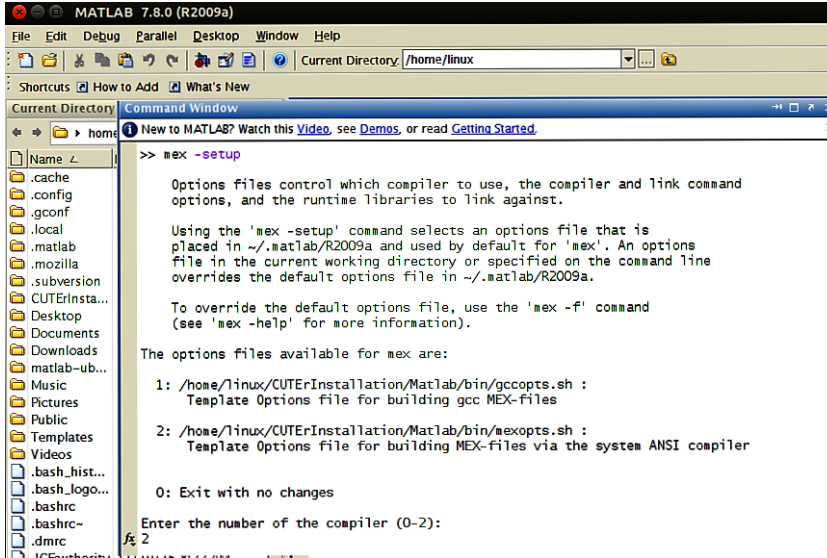
# Precision for which the package was written
# Valid values are "single", "double", "single double" and "double single"
export PACK_PRECISION="double"

# Define the name of the object files for the package which must lie in
# ${MYCUTER}/(precision)/bin
export PACKOBS=""

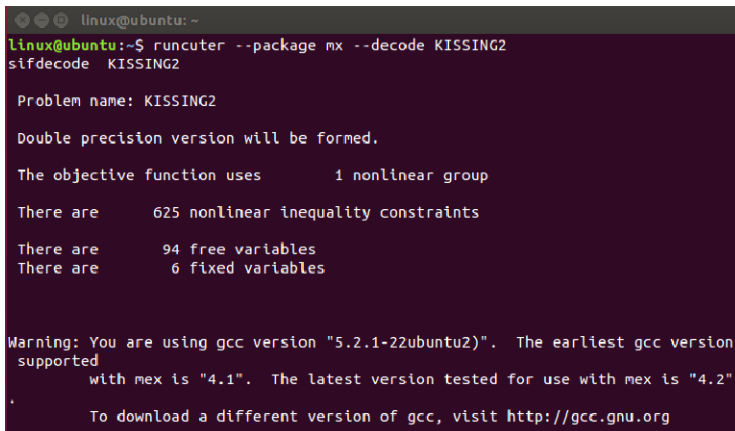
# Define package and system libraries using -lrary to include library.a or .so
export PACKLIBS="-L/usr/local/lib/gcc-lib/i686-pc-linux-gnu/4.0.3 -lf95"
# Define the name of the package specification file if any. This possibly
# precision-dependent file must either lie in the current directory or in
# ${CUTER}/common/src/pkg/${PACKAGE}/ )
export SPECS=""

```

شکل ۱-۳۹: ویرایش Mx



شکل ۴۰-۱: راه اندازی مجدد Mex



شکل ۴۱-۱: 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

```

mcuter.c (~/CUTerInstallation/cuter2/common/src/tools) - gedit
Open Save Undo
mcuter.c x
if (nrhs == 3) {
    if (! isInteger(prhs[2]) && ! mxIsDouble(prhs[2]))
        mexErrMsgTxt("cons: Constraint index must be integer\n");

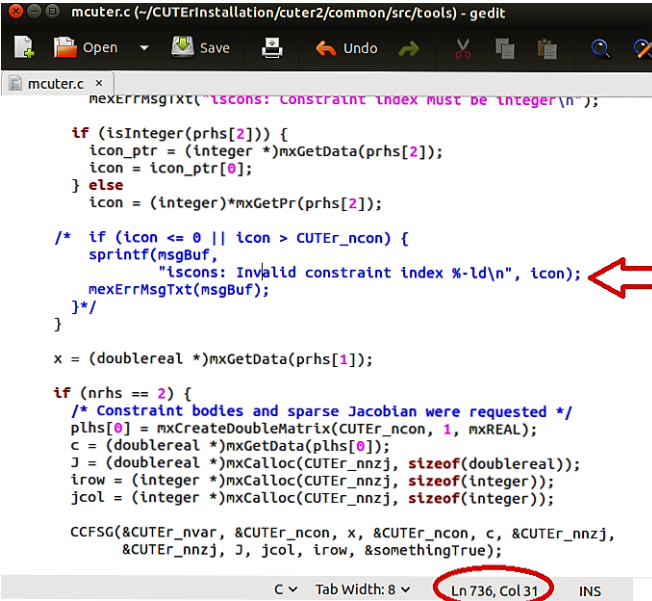
    if (isInteger(prhs[2])) {
        icon_ptr = (integer *)mxGetData(prhs[2]);
        icon = icon_ptr[0];
    } else
        icon = (integer)*mxGetPr(prhs[2]);

    /* if (icon <= 0 || icon > CUTer_ncon) {
        sprintf(msgBuf,
            "cons: Invalid constraint index %-ld\n", icon);
        mexErrMsgTxt(msgBuf);
    }*/
}

x = (doublereal *)mxGetData(prhs[1]);

if (nrhs == 2) {
    /* Constraint bodies (and Jacobian) were requested */
    plhs[0] = mxCreateDoubleMatrix(CUTer_ncon, 1, mxREAL);
    c = (doublereal *)mxGetData(plhs[0]);
    if (nlhs == 2) {
        plhs[1] = mxCreateDoubleMatrix(CUTer_ncon, CUTer_nvar, mxREAL);
        j = (doublereal *)mxGetData(plhs[1]);
    }
}
C Tab Width: 8 Ln 617, Col 43 INS
    
```

شکل ۱-۴۳: خطازدایی ۲



```

mcuter.c x
mexErrMsgTxt("iscons: Constraint index must be integer\n");

if (isInteger(prhs[2])) {
    icon_ptr = (integer *)mxGetData(prhs[2]);
    icon = icon_ptr[0];
} else
    icon = (integer)*mxGetPr(prhs[2]);

/* if (icon <= 0 || icon > CUTER_ncon) {
    sprintf(msgBuf,
           "iscons: Invalid constraint index %-ld\n", icon);
    mexErrMsgTxt(msgBuf);
}*/

x = (double real *)mxGetData(prhs[1]);

if (nrhs == 2) {
    /* Constraint bodies and sparse Jacobian were requested */
    plhs[0] = mxCreateDoubleMatrix(CUTER_ncon, 1, mxREAL);
    c = (double real *)mxGetData(plhs[0]);
    J = (double real *)mxCalloc(CUTER_nnzj, sizeof(double real));
    irow = (integer *)mxCalloc(CUTER_nnzj, sizeof(integer));
    jcol = (integer *)mxCalloc(CUTER_nnzj, sizeof(integer));

    CCFSG(&CUTER_nvar, &CUTER_ncon, x, &CUTER_ncon, c, &CUTER_nnzj,
          &CUTER_nnzj, J, jcol, irow, &somethingTrue);
}

```

C Tab Width: 8 Ln 736, Col 31 INS

شکل ۱-۴۴: خطازدایی ۳

فصل ۲

عملکرد SIFDec

همانطور که در فصل 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^۱ SIF 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 جزئیات ساختار این کدگشا به تفصیل بپردازیم.

۱-۲ 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 جزئیات آن آشنا شود. زمانی که SIFDec فراخوانی می شود مساله به فایل اجرایی این کد گشا 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 جزئیات ایتم های پر

^۱Standard Input Format

^۲Optimization Package

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

-s: با فراخوانی این ایتیم مساله با دقت ساده^۳ کدگشایی و برای حل آماده می شود. لازم به ذکر است که این ایتیم 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 ایتیم 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 ایتیم 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) -default value- uncommented
```

مشاهده می شود که پارامتر پیش فرض برای این مساله $m=25$ است.

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

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

مساله با پارامتر $m=24$ 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 $\text{THETA}=3$ و $N=100$ داشته باشد. مساله با این پارامتر ها بصورت زیر فراخوانی می شود.

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

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

^۳Single precision

^۴Double percision

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

مساله با پارامتر $THETA=2.5$ که در ایتِم show- تعریف نشده، کدگشایی می شود.

توجه: از آنجایی که پارامترهای دلخواه غیر از موارد نشان داده شده در ایتِم show- غیر استاندارد هستند معمولاً 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، 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 پارامترها داشته باشیم. در اینجا برآنیم که روش تغییر اندازه SIFDec را مورد بررسی قرار دهیم.

```

C
C MAXIMUM NUMBER OF VARIABLES.
C
C   PARAMETER      ( NMAX   = 1000000 )
C
C MAXIMUM NUMBER OF GROUPS.
C
C   PARAMETER      ( NGMAX  = 600000 )
C
C MAXIMUM NUMBER OF DIFFERENT GROUP TYPES.
C
C   PARAMETER      ( NGRMAX = 10   )
C
C MAXIMUM TOTAL NUMBER OF REAL PARAMETERS ASSOCIATED WITH GROUPS.
C
C   PARAMETER      ( NGPVMX = 1200000 )
C
C MAXIMUM NUMBER OF NONLINEAR ELEMENTS.
C
C   PARAMETER      ( NELMAX = 1000000 )
C
C MAXIMUM NUMBER OF DIFFERENT NONLINEAR ELEMENT TYPES.
C
C   PARAMETER      ( NLMAX  = 20   )
C
C MAXIMUM TOTAL NUMBER OF ELEMENTAL VARIABLES.
C
C   PARAMETER      ( NELVMX = 1000000 )

```

شکل ۲-۱: اندازه های SIFDec

پارامترهای مختلف تعیین کننده اندازه SIFDec در فایل با مسیر

\$MYSIFDEC/double/config/sifdec.siz

```

linux@ubuntu:~$ sifdecode NCB20

Problem name: NCB20

Double precision version will be formed.

The objective function uses      5001 nonlinear groups

There are      5010 free variables

** Exit from MAKEFN - insufficient space. Increase size of NSETVC

Return from MAKEFN, INFORM = -12

error exit from decoding stage. terminating execution.
linux@ubuntu:~$
    
```

شکل ۲-۲: کدگشایی NCB20

مطابق شکل ۱-۲ قرار دارند. این پارامترها در جدول ۱-۲ خلاصه شده اند. بعنوان مثال در کدگشایی مساله NCB20 خروجی مطابق شکل ۲-۲ است. همانطور که 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 NCB20 کنید مطابق شکل ۲-۴ مساله با موفقیت کدگشایی می شود.

^۵Rebuild

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

جدول ۱-۲: پارامتر های اندازه در SIFDec

```

*sifdec.siz x
C MAXIMUM TOTAL NUMBER OF REAL PARAMETERS ASSOCIATED WITH GROUPS.
C
C   PARAMETER      ( NGPVMX = 1200000 )
C
C MAXIMUM NUMBER OF NONLINEAR ELEMENTS.
C
C   PARAMETER      ( NELMAX = 1000000 )
C
C MAXIMUM NUMBER OF DIFFERENT NONLINEAR ELEMENT TYPES.
C
C   PARAMETER      ( NLMAX = 20 )
C
C MAXIMUM TOTAL NUMBER OF ELEMENTAL VARIABLES.
C
C   PARAMETER      ( NEVMAX = 1000000 )
C
C MAXIMUM TOTAL NUMBER OF INTERNAL VARIABLES.
C
C   PARAMETER      ( NINMAX = 150000 )
C
C MAXIMUM NUMBER OF ENTRIES IN AN ELEMENT HESSIAN.
C
C   PARAMETER      ( NSETVC = 500 )

```

شکل ۲-۳: افزایش NSETVC

```

linux@ubuntu: ~
Return from MAKEFN, INFORM = -12
error exit from decoding stage. terminating execution.
linux@ubuntu:~$ cd $MYSIFDEC
linux@ubuntu:~/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95$ make -s all
making all in ./bin...
making all in ./double...
making all in ./double/bin...
Building   sifdec.o           [OK]
Linking    sifdec             [OK]
making all in ./double/config...
linux@ubuntu:~/CUTerInstallation/sifdec2/SifDec.large.pc.lnx.g95$ cd
linux@ubuntu:~$ sifdecode NCB20

Problem name: NCB20

Double precision version will be formed.

The objective function uses      5001 nonlinear groups

There are      5010 free variables

```

شکل ۲-۴: کدگشایی مجدد NCB20

فصل ۳

عملکرد CUTER

پس از آن که مساله آزمون توسط SiFDec کد گشایی شد در اختیار بسته CUTER قرار می‌گیرد. این بسته با ابزاری که در اختیار دارد امکان بهره برداری از مساله را برای کاربر فراهم می‌سازد. علاوه بر این کاربر می‌تواند 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 می‌تواند مسایل آزمون را با دستور زیر دانلود کرده و در پوشه با مسیر پیش فرضی که در فایل bashrc در متغیر MASTSIF ذخیره می‌شود، مورد استفاده قرار دهد.

```
~$cd $MASTSIF
```

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

۱-۳ 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 می‌شوند علاوه بر این CUTER نیز مجهز به تعدادی کد اولیه کمکی در زبان فرترن می‌باشد که به کاربر اجازه می‌دهد که داده های موجود در مساله را در بسته های نرم افزاری مورد استفاده قرار دهد. به این کد های کمکی ابزار CUTER گفته می‌شود که در مسیر زیر قابل دسترسی هستند.

```
$CUTER/common/src/tools
```

^۱Interface

۴۰ _____ آشنایی با محیط مسایل آزمون CUTEr

فهرستی از این ابزارها در جدول های ۱-۳ و ۲-۳ 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.

۲-۳ 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. ایت 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. عبارتی دستورات زیر را بترتیب در Terminal اجرا کنید.

`~$ cd $MYCUTER`

`~$ make -s all`

۴-۳ 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 `man1` واقع در آدرس

```
$CUTeR/common/man/man1
```

مراجعه کرد. جدول ۴-۳ 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` کفایت 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 `KISSING2` با بسته باشد در اینصورت با اجرای دستور زیر در `Terminal` خروجی مطابق شکل ۱-۳ خواهد بود.

```
~$ runcuter - -package gen - -decode KISSING2
```

```
OPTIMAL SOLUTION FOUND
THE ANSWER IS 41.999999999999999

***** 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 ۱-۳: حل مساله `KISSING2` با بسته `Gen`

۱-۴-۳ 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 ایت‌های پرکاربرد معرفی می‌شود.

package - : نام بسته بهینه‌سازی مورد استفاده در حل مساله پس از این ایت‌ها قرار می‌گیرد.

decode - : نام مساله‌ای که باید حل شود پس از این ایت‌ها قرار می‌گیرد.

sifdecode منتقل می‌شود. توجه: ایت‌های که در لیست بالا نیست در صورت ورود مستقیماً به دستور

۵-۳ رابط CUTEr با MATLAB

برای استفاده از ابزار CUTEr در MATLAB یک رابط کاربری بنام mx طراحی شده که کاربر می‌تواند بعد از کد گشایی مساله از ابزار CUTEr بمنظور دسترسی به داده‌های مساله استفاده کند. برای آماده‌سازی مساله جهت استفاده در MATLAB کاربر باید ابتدا این رابط را فراخوانی کند. فرض کنید هدف استفاده از مساله KISSING2 باشد با دستور زیر یک رابط بین ابزار CUTEr و MATLAB ایجاد می‌شود.

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

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

شایان ذکر است که دستور فوق را می‌توان با عملگر (!) مستقیماً 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 ۲-۳ است.

```
» prob = cuter_ setup()
```

```

>> lruncuter --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]
      b1: [100x1 double]
      bu: [100x1 double]
       v: [625x1 double]
      c1: [625x1 double]
      cu: [625x1 double]
    equatn: [625x1 logical]
    linear: [625x1 logical]
     name: 'KISSING2 '

```

شکل ۳-۲: بارگزاری مساله KISSING2

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

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

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

```
» prob.n
```

```
ans=
```

```
100
```

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

prob.n: تعداد متغیرهای 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 متغیرها.

prob.equatin: 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_ scon : (CUTer Tools: ccfsf / ccifsf)

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 KISSING^۲ 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 متغیرهای 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 NCB^{۲۰} با بعد ۱۰۰ کد گشایی 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,
csgrsh	evaluate both the cons grad/Lag in finite element format
csgrsh	same as csgrsh, 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 ,

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

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
ipopd/sdipopd	IPOPT (Wachter)
knit/sdknit	KNITRO (Byrd, Nocedal and Waltz)
la04/sdla04	LA04 (Reid)
lbb/sdlbb	L-BFGS-B (updated) (Nocedal)
lbs/sdlbs	L-BFGS (Nocedal)
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)

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.