1996

Running NEC4 on the Cray at N.P.S.

Neta, Beny; Knorr, J. B.

Running NEC4 on the Cray at N.P.S.

B. Neta
Naval Postgraduate School
Monterey, California

J. B. Knorr
Naval Postgraduate School
Monterey, California

September 19, 1996

NEC4 is the latest version of Numerical Electromagnetic Code developed at Lawrence Livermore National Laboratory to analyze electromagnetic responses of antennas and scatterers. The code is based on the method of moments to solve integral equations.

In order to run the program on a Cray computer one must modify the subroutine PARSIT by replacing the two read statements by decode statements as follows:

```
      READ(BUFFER(1:LENGTH),*,ERR=9000) INTFLD(I)
```

by

```
      DECODE(LENGTH,9998,BUFFER) INTFLD(I)
9998  FORMAT(i40)
```

and the statement

```
      READ(BUFFER(1:LENGTH),*,ERR=9000) REFLD(I-MAXINT)
```

by

```
      DECODE(LENGTH,9997,BUFFER) REFLD(I-MAXINT)
9997  FORMAT(G40.0)
```

The subroutine SECOND must be replaced also since Cray has its own SECOND function. We renamed the subroutine sSECOND.

Profiling a run on a Cray shows that for a case with 6931 segments, we have:

Subroutine	percentage
SEGXCT (tests pair of segments for intersection)	31.0%
SEGCHK (calls SEGXCT to check)	30.8%
SQRT	20.0%
CONECT (sets up segment connection data)	14.8%
sort	1.7%
all others	1.7%

Table 1: Total run time on Sirius (8 processor Cray) is 11.9 hours

We have concluded that we can save time by avoiding the processing of geometry. This is useful when one wants to run several cases for the same geometry configuration. To this end we modified the subroutine DATAGN.

We have added in the beginning of the program the following statements:

```
c
c  do we read geometry data from GW (and other) cards or from a file
c
      write(*,100)
100   format(' do we have geometry cards to process'/
     &       ' or all cards were processed before'/
     &       ' please answer y if geometry cards are to be processed')
       read(*,102,err=104) ny
102    format(a)
       print *,'  ny ',ny
c       if(ny.eq.'y') go to 110
c
c      the answer is NO
c      geometry cards were already processed
c      ask for geometry file name
c
       write(*,106)
106    format(' what is the geometry file name?')
       read(*,91,err=107) geom
91     format(a)
       if(geom.ne.' ') open(unit=19,file=geom,status='unknown',err=335)
       if(ny.eq.'y') go to 110
       go to 200
335    write(*,109)
109    format(' open error for geometry file')
       stop 109
```

```
104    write (*,105) ny
105    format(' read error y/n  answer was ',a1)
       stop 105
107    write(*,108) geom
108    format(' error in reading name of geometry file '/
     &        ' name was ',a)
       stop 107
110    continue
```

At the end we add the writing to the geometry file

```
       write(19,70) x
       write(19,70) y
       write(19,70) z
       write(19,70) si
       write(19,70) bi
       write(19,70) alp
       write(19,70) bet
       write(19,70) salp
       write(19,70) t2x
       write(19,70) t2y
       write(19,70) t2z
70     format(5f11.5)
       write(19,71) icon1
       write(19,71) icon2
       write(19,71) itag
       write(19,71) iconx
       write(19,71) ipsym,ld,n1,n2,n,np,m1,m2,m,mp,nwire,isct,iphd
71     format(10i5)

375    WRITE(3,72)
72     format(5x,'error writing to file 19 geom')
       STOP
c
c      read previously processed geometry file
c
200    continue
       read(19,70) x
       read(19,70) y
       read(19,70) z
       read(19,70) si
       read(19,70) bi
       read(19,70) alp
       read(19,70) bet
```

```
read(19,70) salp
read(19,70) t2x
read(19,70) t2y
read(19,70) t2z
read(19,71) icon1
read(19,71) icon2
read(19,71) itag
read(19,71) iconx
read(19,71) ipsym,ld,n1,n2,n,np,m1,m2,m,mp,nwire,isct,iphd
return
```

The timing on the Jedi (4 processor Cray computer) is given in table 2.

Subroutine	Create	Use Geometry	Do Not Check Geometry
FACTR	68.10%	70.30%	69.90%
CMWW	6.08%	6.19%	6.14%
EKSCLR	4.57%	4.70%	4.65%
EKSCSZ	3.70%	4.00%	3.90%
EFLDSC	3.40%	3.40%	3.40%
all others	14.10%	11.40%	11.90%
Total Time (secs)	13548.20	13074.75	13184.75
Total Time (hours)	3.76	3.63	3.66

Table 2: Timing comparison for geometry check on Jedi

We conclude that we save 473 seconds by using previous geometry file and 360 second by not checking the geometry at all. This is not much in comparison to almost 4 hours of run time.

As one can see in the previous table, the most time consuming routine is FACTR. Thus, to save computer time we decided to parallelize the LU factorization algorithm. The subroutine FACTR was replaced by a parallel algorithm on the Cray. The results are given in table 3.

Subroutine	3 processors	4 processors
FACTR	50.4%	50.0%
CMWW	10.5%	10.6%
EKSCLR	7.0%	7.1%
EKSCSZ	6.5%	6.7%
EFLDSG	5.5%	5.4%
all others	20.0%	20.2%
Total Time (secs)	8845.09	8844.86
Total Time (hours)	2.46	2.46

Table 3: Timing comparison for parallelization of LU factorization on Jedi

We save 1.2 hours (one third of the computer time) by factoring the matrix in parallel. The next time consuming part of the code is the process to fill in the matrix. In table 4, we give the time for fill-in and factor in serial and parallel versions of NEC4. Remember that the only part we parallelized is FACTR subroutine.

Subroutine	Serial	3 processors	4 processors
fill-in	3866.87	4112.59	4103.92
factor	9191.70	4644.98	4653.19

Table 4: CPU time in seconds on Jedi

Notice that now the factorization in parallel takes as much time as the fill-in process. Certainly the next step in parallelization is the fill-in.

Anyone interested in such a version can send a request to bneta@nps.navy.mil attaching the routines PARSIT, SECOND, DATAGN and FACTR.