SIZE SCALING RELATIONSHIPS IN FRACTURE NETWORKS

Quarterly Report

June 1st 1998 through June 30th 1998

Tom Wilson

Issued July 27, 1998

DOE Award Number DE-FG26-98FT40385--01

DISCLAIMER

This report was prepared as an account of work sponsored by and agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, express or implied, or assumes any legal liability responsibility for the accuracy , completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

SIZE SCALING RELATIONSHIPS IN FRACTURE NETWORKS

Abstract

Work focused on the program development phase, which has been successfully completed. A revised timetable has been prepared that reflects the June start date and other modifications associated with internal budgeting problems and the actual versus proposed start dates.

Table of Contents

Title Page	1
Disclaimer	2
Abstract	3
Table of Contents	4
Executive Summary	5
Results and Discussion	5
Conclusion	10
References	12
Appendix 1	13

SIZE SCALING RELATIONSHIPS IN FRACTURE NETWORKS

EXECUTIVE SUMMARY

A formal project start date of June 1 1998 was assigned to this grant/contract. The actual papers were received around the middle of June. Internal budgeting numbers were not assigned till July 24th making it difficult to get project work underway. Changes in the actual start date along with internal budgeting delays required some revision of the research timetable.

Development of a modified box-counting algorithm referred to as the box-flex by Barton (1995) was identified in the proposal as the initial task to be undertaken. The main reason for the program development effort was to allow counting to take place at non-base 2 subdivisions of the analysis area to provide increased resolution of space-filling characteristics of the enclosed pattern. In application, the box-flex method is not believed to be the best procedure to undertake this. As an alternative, high resolution counting has been obtained through use of fractional box sizes. An inherent problem with the box flex method is that the outer bounding area of analysis expands and contracts to incorporate non-base 2 subdivisions of the area. Thus the actual pattern being analyzed changes from one box size to the next. The region of analysis in the program developed for use in this project remains fixed. There is no need for the outer box to expand or contract in size. The algorithm will be thoroughly tested and compared to Barton's using the fracture patterns analyzed in his study.

Given the lush vegetation characteristics of the Appalachian region this time of year, field visits will be scheduled for the spring of 99 and possibly fall of 98. The pre-award field schedule assumed a winter start date followed by spring field visits. Schedule adjustments have been incorporated into a revised task-referenced timetable submitted with this report. The objectives and tasks of the proposed research will be met within the context of the actual project timeframe.

RESULTS AND DISCUSSION

The initial work conducted under this grant has been focused on the development of a detailed box counting analysis algorithm. In the traditional approach, box counting begins with a single box that covers the entire area. The length of the box sides is decreased by a factor of two so that in the second iteration a total of four boxes are required to cover the area. Again the number of boxes containing part of the pattern is counted and the process repeated. If the pattern is fractal then the number of occupied boxes (N) varies as the length of the box side (r) raised to some power (-D) as shown in Equation (1).

$$N = r^{-D} \quad (1)$$

D is referred to as the fractal dimension, and C is a constant of proportionality.

Analysis of box counting data is illustrated using fracture patterns from the Valley and Ridge study area (Figure 1A). Number of occupied boxes (N) versus length of box side (r) are plotted in a log-log format (Figure 1B). As noted by Wilson et al. (1997) and

in the current contract proposal from which this example is taken, box counting data are

typically non-linear. The slope varies gradually from a steeply sloping region ($D = 2$)

Figure 1: A) Fracture patterns and B) traditional log-log display of box counting data.

associated with the larger boxes to a region of 0-slope for the smaller boxes. This non-linear behavior suggests that the fracture pattern (Figure 1A) is not fractal. Walsh and

Watterson (1993) note similar behavior in their reevaluation of Pavement 1000 from Barton and Hsieh (1989) in the Yucca Mountain area.

The features encountered in the box plot (Figure 1B) are, however, anticipated, and the reasons for this non-linearity are briefly reviewed. Recall (Wilson et al 1997) that the initial box is always occupied since it covers the entire area. In the next covering, which consists in this example of four 79x79 inch boxes, all boxes continue to be filled. When boxes at successively smaller scales are all occupied the number of occupied boxes increases as the square of the number of boxes along the map edge, and the number of boxes covering the map edge increases as $(1/r)^2$ yielding the log-log slope of -2 and fractal dimension of 2. This result is usually encountered in the first 2 to 3 coverings of the area.

The region of Figure 1B where $D = 0$ results from the discrete nature of the fracture trace database. The fracture patterns are digitized at a certain sampling interval; there are a finite number of samples in the database used for box counting. The data shown in Figure 1A were sampled at approximately 0.3 inch intervals. For box sizes less than the sampling interval the number of occupied boxes will equal the total number of samples in the data set. Further decreases of box size yield no further increase in the number of occupied boxes and the slope of the curve in this region is 0 (Figure 1B).

The $D=0$ region and the transition to it is an artifact of the digital format of the data. If the data were comprised of continuous line segments, a reduction in box size by $1/2$ would double the number of occupied boxes and we would eventually end up in a non-fractal region with D approaching 1. The scale at which this occurs may represent a resolution limit below which finer details are not represented, or are, in fact, not present.

Regions of the box plot where $D \approx 2$ and D approaches 0 provide no useful information about the data. They are associated with artifacts of the initial box covering and the digital form of the data. The data must be examined between these transitional regions in the box plot. Walsh and Watterson (1993) suggest that fractal analysis should be confined to the range of scales between the largest and smallest fracture spacings in the data set. Spacing of a few inches or so represent an approximate minimum spacing for the fractures shown in Figure 1A. Maximum spacing of about 40 to 50 inches or so appears in the pattern (Figure 1A). However, the log-log box plot between this range of scales is non-linear (Figure 2B). The box plot may be expected to exhibit roughly linear behavior over the range from a few inches to a few tens of inches, but as shown in Figure 1B slope decrease continues to occur through this region. Visually these changes appear minor, but in terms of point-to-point variations in D they are significant dropping from 1.59 to 1.25. The fractal model does not appear viable. This problem was also identified in the data presented by Barton (1995) in the project proposal. Linearity in the box-plot although visually plausible over a certain range consistently exhibits slope reduction.

The box counting response over the approximately linear range (Figure 1B) is examined in greater detail using fractional logarithmically spaced reductions in box size. This required use of fractional boxes to cover the area. Inclusion of fractional boxes in the count is preferred over the approach taken by Barton (1995) where the area is expanded or contracted to prevent occurrence of fractional size boxes. The drawback in the approach taken by Barton (1995) is that the pattern being analyzed actually varies from

one box size to the next as the region is expanded or contracted to accommodate only full sized boxes.

Detailed box counting over the 3.9 to 39 inch range was undertaken of the fracture pattern shown in Figure 1A. The initial count was made for a 16 box covering and the final count for a 1600 box covering. The 3.9 to 39 inch range was divided into 20 equally spaced logarithmic intervals. The point-to-point slopes are shown in Figure 2A and the detailed N vs. r plot is shown in Figure 2B. The point-to-point slopes suggest that there are two regions in the data: one above 13 inch scales and one below. The data on either side of the 13inch-break form roughly linear regions in Figure 2B with D of 1.58 and 1.21.

Figure 2: A) Point-to-point variations in the fractal dimension of the box counting results shown in B. B) Detailed box counting of the fracture pattern shown in Figure 1A.

The bi-linear character of box counting data suggests the presence of range-limited fractal response. Interestingly, this bi-linearity appears in patterns observed over much

larger areas. As an example, the active fault patterns from a 70km by 70km area of central Japan are shown in Figure 3A. The traditional base-2 box counting results are shown in Figure 3B. The results of detailed box counting are shown in Figure 3C. Bilinear behavior is noted. While these results have not demonstrated, nor do they suggest, a continuum, they do illustrate bi-linear behavior in fracture patterns separated in scale by 5 orders of magnitude. The response represents something either fundamental about fracture patterns and rock mechanics or a bias in the interpreter to under-represent smaller scale features in the portrayal of a pattern. This break tends to occur at scales of about one-tenth the scale of the view, map or image. The observed bi-linearity may be actual, or simply a case of forcing lines through a response that is actually curvilinear.

Figure 3. A Active fault patterns from a region in central Japan. B) Log-log plot of base-2 box counting for the active fault pattern. C) Log-log display of detailed, non-base-2 coverings of the pattern.

CONCLUSIONS

An algorithm was developed which undertakes box counting for non-base-2 variations in box size. Box size is decreased logarithmically. Fractional boxes are used

around the edges when necessary so that the pattern can be completely covered without expanding or contracting the initial covering box. The algorithm is included as Appendix 1.

Bilinear or curvilinear behavior has been observed in detailed box counting over scales at which the pattern is usually assumed to be well represented. In the next quarter, patterns analyzed by Barton (1995) will be digitized and evaluated using the revised box counting algorithm. Behavior in the region below the slope break will be examined by revisiting individual station locations and re-photographing patterns at smaller scale. Analysis of these detailed photographs will be undertaken to evaluate the reliability of the prediction made from larger scale.

During the coming quarter program development work will be undertaken to sort out fracture length and orientation data from the digitized image files. A statistical analysis of these data will be undertaken in subsequent quarters.

REVISED TIMETABLE

1																			
2																			
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			

1. Revise the box-counting program to incorporate the box-flex method (Barton 1995) of box counting for estimation of fractal dimension. Software development also required to analyze fracture length and orientation distributions. This task includes testing of new algorithms using data collected in the previous contract.

2. Re-photograph fracture patterns from close (1-3m) and distant (10m) views.

3. Locate and photograph additional fracture patterns

4. Reinterpret and redigitize all fracture images, clearly differentiating between individual sets by establishing abutting relationships.

5. Recompute fractal dimensions of all data using the box-flex method and compare to previous analysis.

6. Compute the fractal dimensions from multiple scale observations using the box-flex method.

7. Examine box-pots for existence of a true fractal regime.

8. Compare results from different scales.

9. Compute length distributions for individual sets and the total pattern observed on digitized fracture patterns images (includes software development).

10. Compute the fractal dimension of fracture lengths at each exposure.

11. Evaluate the stochastic properties of fracture length in 2D images, and spacing in 1D data sets.

12. Compute fracture orientations from digitized images (includes software development)

13. Write final report

REFERENCES

- Barton, C. C., 1995, Fractal analysis of scaling and spatial clustering of fractures: in Barton and LaPointe (eds.) *Fractals in the Earth Sciences*, Plenum Press, 265p.
- Barton, C. C., and Hsieh, P. A., 1989, Physical and hydrologic- flow properties of fractures; 28th International Geological Congress Field Trip Guidebook T385, *American Geophysical Union*, Washington, D. C., p. 36.
- Walsh, J. J., and Watterson, J., 1993, Fractal analysis of fracture patterns using the standard box-counting technique: valid and invalid methodologies; Brevia, in *Journal of Structural Geology*, vol. 15, no. 12, pp. 1509-1512.
- Wilson, T. H., Dominic, J., and Halverson, J., 1997, Fractal Interrelationships in Field and Seismic Data: Final Research Report of Work Conducted on U. S. Department of Energy Contract DE-FG21-95MC32158, 162 p.

```

!*****logbox.for*****
! This program determines whether boxes of a given size
! used to cover a pattern are occupied or not. Box size
! is varied from some a specified maximum size to minimum
! size, and the variations occur in fractional logarithmic
! intervals. Point-to-point slope is computed, and the
! power-law slope over a given range can also be computed.
!


---


dimension x(50000),y(50000),ax(100),ay(100)
character in*16
character ans*1
write(*,1)
1 format(' Enter outline filename:',$)
  read(*,2) in
2 format(a16)
  open (1,file=in,status='old')
  open (3,file='logbox.dat',status='unknown')
  axmin=1e32
  aymin=1e32
  axmax=-1e32
  aymax=-1e32
  do 3 i=1,50000
    read(1,*,end=100) x(i),y(i)
    if(x(i).lt.axmin) axmin=x(i)
    if(x(i).gt.axmax) axmax=x(i)
    if(y(i).lt.aymin) aymin=y(i)
    if(y(i).gt.aymax) aymax=y(i)
3  continue
    100  max=i-1
    write (*,4)max
4  format(' Number of data points = ',I10)
400 write(*,*)'*****'
    write(*,*)'min x = ',axmin
    write(*,*)'max x = ',axmax
    write(*,*)'min y = ',aymin
    write(*,*)'max y = ',aymax
    write(*,*)'*****'
    write(*,*)'Specify the X and Y extents of the '
    write(*,*)'area you wish to box count.'
    write(*,5)
5  format(' Enter Xmin and Xmax:',$)
    read (*,*)xmin,xmax
    write(*,6)
6  format(' Enter Ymin and Ymax:',$)

```

```

read (*,*)ymin,ymax
write(*,*)'*****'
write(*,*)
write(*,7)
7   format(' Enter starting number of boxes along x axis:',$)
   read(*,*)aminb
   write(*,8)
8   format(' Enter ending number of boxes along x axis:',$)
   read(*,*)amaxb
   write(*,9)
9   format(' How many steps do you want to cover this range?',$)
   read (*,*) nint

   write(*,10)
10  format(' Number of boxes along x-axis will decrease in')
   write(*,11)
11  format(' size logarithmically between the max and min values')
   write(*,*)

!initialize individual point locations for point-point
!computation of fractal dimension.

J=1
ax(1)=0.0
an(1)=0.0
ax(2)=0.0
an(2)=0.0
fd=0.0
deld=0.0

!compute base 10 log of the min and max numbers of boxes

sizemax=(xmax-xmin)/aminb
sizemin=(xmax-xmin)/amaxb

write(*,*)
write(*,*)'The length of the largest box is ',sizemax
write(*,*)'The length of the smallest box is ',sizemin
write(*,*)
write(*,12)
12  format(' (x length)', ' (# occupied boxes)',
1   ' point-point d')

almin = alog10(aminb)

```

```

amax = alog10(amaxb)

! Compute the logarithmic decrement in box size
aldif = (amax-almin)/(nint-1)

! initial parameter settings
d = almin
rnb = aminb
n = aminb
rleng=(xmax-xmin)/amaxb
fracwin = 0.0

!Compute length of starting box sides
dx = (xmax-xmin)/rnb
frac=dx/100
dy=dx

! initialize number of occupied boxes before box counting
15 fno=0.0

!exit program when box size becomes too small
if(j.gt.nint) go to 99

!run through x
do 16 nx=1,n
!run through y for each x
do 16 ny=1,n

!test to see if this box is a fractional box
if (nx.eq.n.and.fracwin.ne.0.0) then
    xnx=xmax
    xnn=xnx-(fracwin*dx)
else
    xnx=xmin+(nx*dx)
    xnn=xnx-dx
endif

if (ny.eq.n.and.fracwin.ne.0.0) then
    ynx=ymax
    ynn=ynx-(fracwin*dx)
else
    ynx=ymin+(ny*dy)
    ynn=ynx-dy
endif

```

```

!see if any data are located within the box.
do 17 m=1,max
  if(x(m).gt.xnn.and.x(m).lt.xnx) then
    if(y(m).gt.ynn.and.y(m).lt.ynx) go to 19
  endif
17 continue

  go to 16

! After a given box has been examined adjust total number
! of occupied boxes.

19  if (ny.eq.n.and.nx.ne.n)then
      if (fracwin.ne.0.0) then
        fno = fno+fracwin
      else
        fno = fno+1.0
      endif
      go to 16
    endif
  if (ny.ne.n.and.nx.eq.n)then
      if (fracwin.ne.0.0) then
        fno = fno+fracwin
      else
        fno = fno+1.0
      endif
      go to 16
    endif
  if(ny.eq.n.and.nx.eq.n) then
      if (fracwin.ne.0.0) then
        fno = fno+(fracwin*fracwin)
      else
        fno = fno+1.0
      endif
      go to 16
    endif
  fno=fno+1.0
16  continue

! output results of counting a particular box size
j=j+1
ax(j)=dx

```



```

an(j)=fno

if (ax(j-1).eq.0)then
    go to 101
else
    call fracd(ax(j-1),an(j-1),ax(j),an(j),deld)
endif
101 if(deld.eq.0.0)then
    write (*,*)dx,fno
    write(3,21)dx,fno
else
    write(*,*)dx,fno,deld
    write(3,22)dx,fno,deld
endif

21  format(1x,f12.6,1x,f10.4)
22  format(1x,f12.6,1x,f10.4,2x,f9.5)

! get ready for the next box count. The area has been
! covered by boxes of the preceding size and number of
! occupied boxes counted.
! Now initialize parameters for next box size.
d = d+aldif
rnb = 10**d
n = rnb

! Compute x and y lengths of next box size.
! Also, Test whether the next box size will include
! fractional boxes. If fractional boxes are present increment n
dx=(xmax-xmin)/rnb
frac=dx/100.0
dy=dx
fracwin = rnb-n
if (fracwin.gt.frac) then
    n=n+1
else
    fracwin = 0.0
endif

go to 15
99  continue

write(*,*)'_____'
```

```

write(*,*)

call fit(sizemax,sizemin,ax,an,nint,fd)

write(*,300)
300  format(' Do you want to box count another subdivision:',$)
    read (*,301)ans
301  format(a1)
    if (ans.eq.'y'.or.ans.eq.'Y') goto 400

stop
end

```

```

subroutine fracd(ax1,an1,ax2,an2,deld)

  alx1=log10(ax1)
  alx2=log10(ax2)
  aln1=log10(an1)
  aln2=log10(an2)

  deld=-1.0*((aln2-aln1)/(alx2-alx1))

4    continue

  return
end

subroutine fit(smax,smin,ax,an,nint,fd)
  dimension ax(100),an(100),alx(100),aln(100),f(40)
  character ans*1
  real*8 fk
  f(1)=12.7
  f(2)=4.3
  f(3)=3.18
  f(4)=2.78
  f(5)=2.57
  f(6)=2.45
  f(7)=2.37
  f(8)=2.31

```

f(9)=2.26
 f(10)=2.23
 f(11)=2.2
 f(12)=2.18
 f(13)=2.16
 f(14)=2.15
 f(16)=2.13
 f(17)=2.12
 f(18)=2.11
 f(19)=2.1
 f(20)=2.09
 f(21)=2.08
 f(22)=2.07
 f(23)=2.075
 f(24)=2.068
 f(25)=2.063
 f(26)=2.06
 f(27)=2.055
 f(28)=2.05
 f(29)=2.045
 f(30)=2.04

```

1  write(*,2)
2  format(' Do you wish to specify the range over which D')
   write(*,3)
3  format(' is calculated?',$)
   read(*,4)ans
4  format (A1)
   write(*,*)
41 IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') then
      write(*,5)
5  format(' Enter max and min box sizes:',$)
   read (*,*)s1,s2
   else
      s1=smax
      s2=smin
   endif

   sumx=0.0
   sumy=0.0
   sumxy=0.0
   sumx2=0.0
   sumy2=0.0
   j=0
  
```

```

do 6 i=2,nint+1
if(ax(i).le.s1.and.ax(i).ge.s2)then
    aln(i)=log10(an(i))
    alx(i)=log10(ax(i))
    sumx=sumx+alx(i)
    sumy=sumy+aln(i)
    sumxy=sumxy+alx(i)*aln(i)
    sumx2=sumx2+alx(i)*alx(i)
    sumy2=sumy2+aln(i)*aln(i)
    j=j+1
endif
6 continue
fk=j
s12=(1/(fk-1))*(sumx2-(1/fk)*sumx*sumx)
s22=(1/(fk-1))*(sumy2-(1/fk)*sumy*sumy)
sxy=(1/(fk-1))*(sumxy-(1/fk)*sumx*sumy)
H=sxy/s12
d=-h
qnot=(fk-1)*(s22-(h**2)*s12)
k=fk-2
c=f(k)
el=c*(qnot**0.5)/(s12*(fk-1)*(fk-2))**0.5
r=sxy/((s12*s22)**0.5)
write(*,7)s1,s2,d
7 format(' Over the range ',f10.4,' to ',f10.4,' d = ',f8.6)
write(*,72)r
72 format(' The correlation coefficient is ',f10.7)
write(*,71)el
71 format(' Width of the 95% confidence band is ',f10.7)

write(*,*)

do 8 i=1,nint
aln(i)=0.0
alx(i)=0.0
8 continue

write(*,9)
9 format(' Do you want to select another range of')
write(*,10)
10 format(' box sizes?', $)
read (*,11)ans
11 format(a1)

```

```
IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') goto 41
```

```
return  
end
```