APPENDICES

APPENDIX A

EXPERIMENTS CONDUCTED

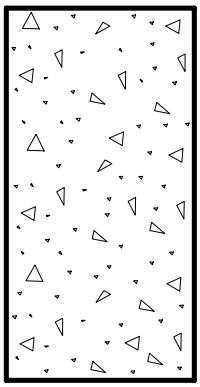
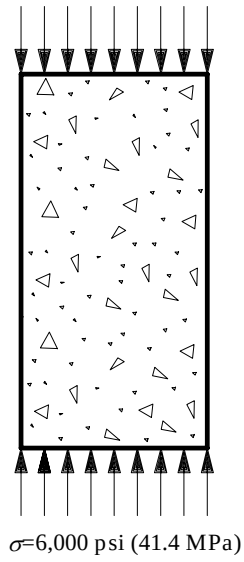
APPENDIX B

IMAGE ANALYZER COMPUTER PROGRAMS

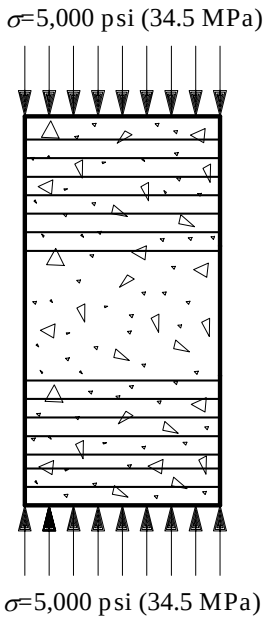
APPENDIX C

TEST CELL DESIGN

APPENDIX A: EXPERIMENTS CONDUCTED

EXPERIMENT #1	
	Loading Condition: **NO-LOAD** Applied Stresses: **0** Concrete Type: **NORMAL STRENGTH** Specimen Size: **4"Ø×8" High** Ultimate Strength: **7,500 psi (94,600 #)** Date of Cast: **November 16, 1989** Date Tested: **March 4, 1992** Confinement: **NONE** Experiment Date: **April 5, 1992**
EXPERIMENT #2	
	Loading Condition: **UNIAXIAL** Applied Stresses: **6,000 psi** Concrete Type: **NORMAL STRENGTH** Specimen Size: **4"Ø×8" High** Ultimate Strength: **7,500 psi (94,600 #)** Date of Cast: **November 16, 1989** Date Tested: **March 4, 1992** Confinement: **NONE** Experiment Date: **October 29, 1992**

EXPERIMENT #3



Loading Condition: **TRIAXIAL**

Applied Stresses: **5,000 psi**

Concrete Type: **NORMAL STRENGTH**

Specimen Size: **4"Ø×8" High**

Ultimate Strength: **7,500 psi (94,600 #)**

Date of Cast: **November 16, 1989**

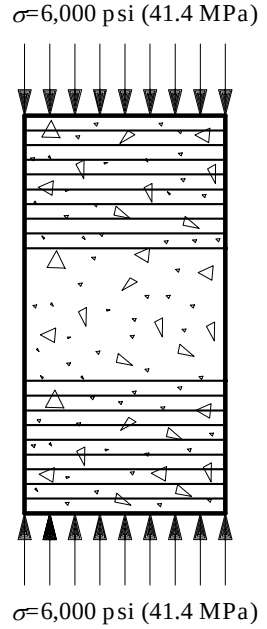
Date Tested: **March 4, 1992**

Confinement: **CONFINED 1/3 FROM EACH END**

Experiment Date: **September 17, 1992**

(Pitch of Winding: 10 Threads/Inch, Pre-Tension: 150 lbs)

EXPERIMENT #4



Loading Condition: **TRIAXIAL**

Applied Stresses: **6,000 psi**

Concrete Type: **NORMAL STRENGTH**

Specimen Size: **4"Ø×8" High**

Ultimate Strength: **7,500 psi (94,600 #)**

Date of Cast: **November 16, 1989**

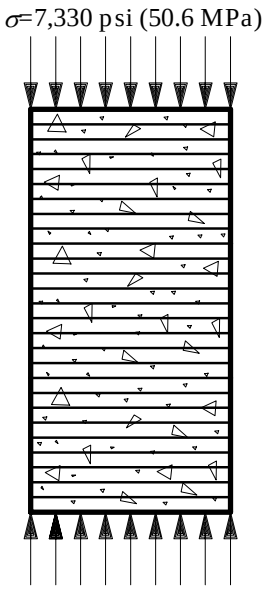
Date Tested: **March 4, 1992**

Confinement: **CONFINED 1/3 FROM EACH END**

Experiment Date: **September 22, 1992**

(Pitch of Winding: 20 Threads/Inch, Pre-Tension: 150 lbs)

EXPERIMENT #5



Loading Condition: **TRIAXIAL**

Applied Stresses: **7,330 psi**

Concrete Type: **NORMAL STRENGTH**

Specimen Size: **4"∅×8" High**

Ultimate Strength: **7,500 psi (94,600 #)**

Date of Cast: **November 16, 1989**

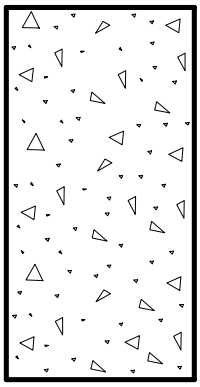
Date Tested: **March 4, 1992**

Confinement: **FULLY CONFINED**

Experiment Date: **April 5, 1992**

(Pitch of Winding: 20 Threads/Inch, Pre-Tension: 150 lbs)

EXPERIMENT #6



Loading Condition: **NO-LOAD**

Applied Stresses: **0**

Concrete Type: **HIGH STRENGTH**

Specimen Size: **4"∅×8" High**

Ultimate Strength: **11,000 psi (138,200 #)**

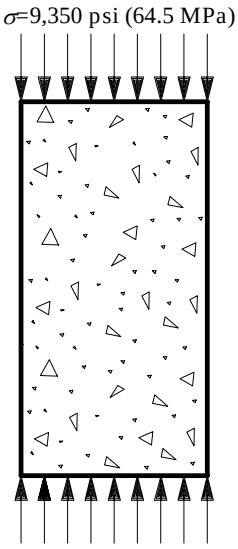
Date of Cast: **October 30, 1991**

Date Tested: **1991**

Confinement: **NONE**

Experiment Date: **September 28, 1992**

EXPERIMENT #7



Loading Condition: **UNIAXIAL**

Applied Stresses: **9,350 psi**

Concrete Type: **HIGH STRENGTH**

Specimen Size: **4"Ø×8" High**

Ultimate Strength: **11,000 psi (138,200 #)**

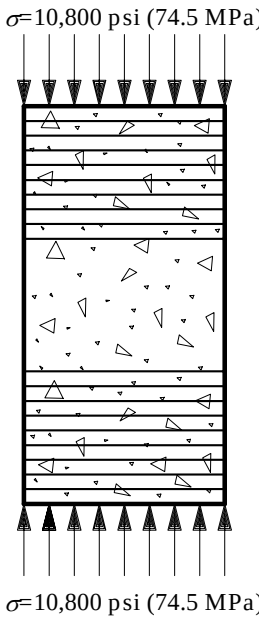
Date of Cast: **October 30, 1991**

Date Tested: **1991**

Confinement: **NONE**

Experiment Date: **March 4, 1993**

EXPERIMENT #8



Loading Condition: **TRIAXIAL**

Applied Stresses: **10,800 psi**

Concrete Type: **HIGH STRENGTH**

Specimen Size: **4"Ø×8" High**

Ultimate Strength: **11,000 psi (138,200 #)**

Date of Cast: **October 30, 1991**

Date Tested: **1991**

Confinement: **CONFINED 1/3 FROM EACH END**

Experiment Date: **March 5, 1993**

(Pitch of Winding: 20 Threads/Inch, Pre-Tension: 150 lbs)

APPENDIX B: IMAGE ANALYZER COMPUTER PROGRAMS

Macro to calculate the number of intercepts with cracks in an image by reference to intercept lines at various angles to the vertical (lines in increments of 15 degrees). Also, macro will find both #3 way and 4 way intersections in a binary thinned image.

```
loadlut "grey"
setframe "F512"
alpha
clearallio 0
Xclrmess
resetvec "aspA"
resetvec "aspB"
resetpar
setimpath "c:/images"
getim "flin0",21
getim "flin15",22
getim "flin30",23
getim "flin45",24
getim "flin60",25
getim "flin75",26
getim "flin90",27
getim "flin105",28
getim "flin120",29
getim "flin135",30
getim "flin150",31
getim "flin165",32
scalgeom 1,"x60",_OFF,_OFF
global TAREA
global CK0,CK15,CK30,CK45,CK60,CK75,CK90,CK105,CK120,CK135,CK150,CK165,TCK
global
SCK0,SCK15,SCK30,SCK45,SCK60,SCK75,SCK90,SCK105,SCK120,SCK135,SCK150,SCK165,
STCK
global SURFA,SURFASM
global NODES,SNODES
SURFA=SURFASM=TAREA=0.0
CK0=CK15=CK30=CK45=CK60=CK75=CK90=CK105=CK120=CK135=CK150=CK165=
TCK=0
SCK0=SCK15=SCK30=SCK45=SCK60=SCK75=SCK90=SCK105=SCK120=SCK135=SCK150=
SCK165=STCK=0
NODES=SNODES=0
dbA="kamran"
dbB="kamran"
rt="kamran"
m=20
```

```
InitField AREAP,TOTALAREA,FIELDCOUNT
aspA[]=TAREA,AREAP,CK0,CK15,CK30,CK45,CK60,CK75,CK90,CK105,CK120,CK135,
CK150,CK165,TCK,SURFA,NODES
aspB[]=SCK0,SCK15,SCK30,SCK45,SCK60,SCK75,SCK90,SCK105,SCK120,SCK135,SCK150,
SCK165,STCK,SURFASM,SNODES
read "Enter root name:",rt
read "Enter database name:",dbA
read "Enter database name (straightened cracks):",dbB
read "Enter number of images:",m
setimpath "f:/"+string(rt)
DBerase dbA
DBerase dbB
DBcreate dbA,"aspA"
DBcreate dbB,"aspB"

for n=1,n<=m,n=n+1
getim n,1
write "IMAGE ",n
dis2lev 1,2,170,255,_ON,_OFF,1
scrap 2,12,_OFF,0,10,_ON,_ON
scrap 2,3,_OFF,0,2,_ON,_ON
close 3,4,7,255,1
open 4,5,7,255,1
scrap 5,6,_OFF,0,200,_ON,_ON
andim 6,12,7
xorim 7,12,8
scrap 8,9,_OFF,0,10,_ON,_ON
thinbin 9,10,0,0,_OFF,_ON
dilate 10,11,7,255,2
thinbin 11,11,0,0,_OFF,_ON

morpho3x3 10,14,"nodes","nodes",1,0,1
morpho3x3 11,15,"nodes","nodes",1,0,1
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
NODES=FIELDCOUNT
Measf 15
SNODES=FIELDCOUNT

andim 10,21,14
andim 11,21,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK0=FIELDCOUNT
```

```
Measf 15
SCK0=FIELDCOUNT
andim 10,22,14
andim 11,22,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK15=FIELDCOUNT
Measf 15
SCK15=FIELDCOUNT
andim 10,23,14
andim 11,23,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK30=FIELDCOUNT
Measf 15
SCK30=FIELDCOUNT
andim 10,24,14
andim 11,24,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK45=FIELDCOUNT
Measf 15
SCK45=FIELDCOUNT
andim 10,25,14
andim 11,25,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK60=FIELDCOUNT
Measf 15
SCK60=FIELDCOUNT
andim 10,26,14
andim 11,26,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK75=FIELDCOUNT
```

```
Measf 15
SCK75=FIELDCOUNT
andim 10,27,14
andim 11,27,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK90=FIELDCOUNT
Measf 15
SCK90=FIELDCOUNT
andim 10,28,14
andim 10,28,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK105=FIELDCOUNT
Measf 15
SCK105=FIELDCOUNT
andim 10,29,14
andim 11,29,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK120=FIELDCOUNT
Measf 15
SCK120=FIELDCOUNT
andim 10,30,14
andim 11,30,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK135=FIELDCOUNT
Measf 15
SCK135=FIELDCOUNT
andim 10,31,14
andim 11,31,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK150=FIELDCOUNT
```

```
Measf 15
SCK150=FIELDCOUNT
andim 10,32,14
andim 11,32,15
dilate 14,14,5,255,1
dilate 15,15,5,255,1
identify 14,14,_OFF,_ON
identify 15,15,_OFF,_ON
Measf 14
CK165=FIELDCOUNT
Measf 15
SCK165=FIELDCOUNT
Measf 10
TAREA=TOTALAREA
TCK=CK0+CK15+CK30+CK45+CK60+CK75+CK90+CK105+CK120+CK135+CK150+CK165
STCK=SCK0+SCK15+SCK30+SCK45+SCK60+SCK75+SCK90+SCK105+SCK120+SCK135+
SCK150+SCK165
SURFA=3.0929E-6*TCK
SURFASM=3.0929E-6*STCK
Measf 9

DBopen dbA,"aspA"
DBappend dbA
DBclose dbA
DBopen dbB,"aspB"
DBappend dbB
DBclose dbB
endfor
outlist dbA,0
outlist dbB,0
graphic
```

Macro to calculate the number and length of cracks in an image

```
loadlut "grey"
setframe "F512"
alpha
clearallio 0
Xclrmess
resetvec "aspA"
resetvec "aspB"
resetpar
scalgeom 1,"x60",_OFF,_OFF
global NCRK,SNCRK,LCRK,SLCRK,ACRK,SACRK
NCRK=SNCRK=0
LCRK=SLCRK=ACRK=SACRK=0.0
dbA="kamran"
dbB="kamran"
rt="kamran"
m=20
InitField FIELDCOUNT
InitObj ANGLEDMAX,PERIM
aspA[]=NCRK,LCRK,ACRK
aspB[]=SNCRK,SLCRK,SACRK
read "Enter root name:",rt
read "Enter database name:",dbA
read "Enter database name (smoothed):",dbB
read "Enter number of images:",m
setimpath "f:/"+string(rt)
DBerase dbA
DBcreate dbA,"aspA"
DBclose dbA
DBerase dbB
DBcreate dbB,"aspB"
DBclose dbB

for n=1,n<=m,n=n+1
getim n,1
write "IMAGE ",n
dis2lev 1,2,170,255,_ON,_OFF,1
scrap 2,12,_OFF,0,10,_ON,_ON
scrap 2,3,_OFF,0,2,_ON,_ON
close 3,4,7,255,1
open 4,5,7,255,1
scrap 5,6,_OFF,0,200,_ON,_ON
andim 6,12,7
xorim 7,12,8
scrap 8,9,_OFF,0,10,_ON,_ON
thinbin 9,10,0,0,_OFF,_ON
dilate 10,11,7,255,2
```

```
thinbin 11,11,0,0,_OFF,_ON

identify 10,10,_ON,_OFF
Measf 10
NCRK=FIELDCOUNT
f=FIELDCOUNT
identify 10,10,1,0
for c=1,c<=f,c=c+1
Measo 10
LCRK=PERIM/2
ACRK=ANGLEDMAX
DBopen dbA,"aspA"
DBappend dbA
DBclose dbA
endfor

identify 11,11,_ON,_OFF
Measf 11
SNCRK=FIELDCOUNT
g=FIELDCOUNT
identify 11,11,1,0
for d=1,d<=g,d=d+1
Measo 11
SLCRK=PERIM/2
SACRK=ANGLEDMAX
DBopen dbB,"aspB"
DBappend dbB
DBclose dbB
endfor

endfor

outlist dbA,0
outlist dbB,0
graphic
```

Macro to measure interfacial cracks in an image

```
clearallio 0
Xclrmess
resetvec "asp"
resetpar
scalgeom 1,"x60",_OFF,0
global area,area2
area=area2=0.0
global INTAREA,INTA2
INTAREA=INTA2=0.0
global FINT,FINT2
FINT=FINT2=0.0
db="kamran"
rt="kamran"
m=20
InitField TOTALAREA
asp[]=area,INTAREA,FINT,area2,INTA2,FINT2
read "Enter root name:",rt
read "Enter database name:",db
read "Enter number of images:",m
setimpath "f:/"+string(rt)
DBerase db
DBcreate db,"asp"

for n=1,n<=m,n=n+1
getim n,1
write "IMAGE ",n
scalim 1,9,0,255,1,255,0,255,1
eraseinside 9,10,0
ans="y"
_STATUS=1
while 1
Xclrmess
read "Edit OK (type n to edit again, <Esc> if OK :",ans
if _STATUS==27:break
if ans=="n" : eraseinside 9,10,0
endwhile
dis2lev 10,10,0,1,_ON,_OFF,1
scrap 10,10,_OFF,0,30,_ON,_ON
dilate 10,11,7,255,2
xorim 11,10,12
dilate 12,12,7,255,1
dis2lev 1,2,170,255,_ON,0,1
scrap 2,3,_OFF,0,2,_ON,_ON
scrap 2,14,_OFF,0,10,_ON,_ON
close 3,4,7,255,1
```

```
open 4,5,7,255,1
scrap 5,6,_OFF,0,200,_ON,_ON
andim 6,14,7
xorim 7,14,8
scrap 8,9,_OFF,0,10,_ON,_ON
thinbin 9,15,0,0,_OFF,_ON
thinbin 14,4,0,0,_OFF,_ON
andim 4,12,5
Measf 4
area=TOTALAREA
if area==0 : area=10000000000
Measf 5
INTAREA=TOTALAREA
FINT=INTAREA/area
andim 15,12,6
Measf 15
area2=TOTALAREA
if area2==0 : area2=10000000000
Measf 6
INTA2=TOTALAREA
FINT2=INTA2/area2

DBopen db,"asp"
DBappend db
DBclose db
endfor

outlist db,0
```

KONTRON CODES

alpha
Sets the graphic display to alpha-numeric mode.

Parameters:
none

andim (input1, input2, output)
Andim correlates every pixel which is both in image 1 and image 2 and the result outputed to image 3. ***Input1*** and ***Input2*** are the two binary images to be correlated.

clearallio (greyvalue)
Clears the overall and contents of the entire image memory. The image memory is set to a level **GrayValue**.

Parameters:

GreyValue	image to be processed

close (input, output, shape, phase, count)
Objects in a binary or grey image are first dilated using a structuring element defined by ***shape_?***, and then eroded, using the same structuring element. This procedure merges closely adjacent objects. The closing procedure can be iterated.

Parameters:

Input	image to be processed
Output	the processed image
Shape_?	structuring element for the operation: =1 horizontal vector =2 vertical vector =3 diagonal vector, 45 degrees =4 diagonal vector, 135 degrees =5 cross =6 square =7 octagon =8 alternate squares/crosses
Phase	selects the grey level for closing
Count	number of operations

condition is a logical expression. ***Start value*** and ***iccrement*** are arithmatic or logical expressions. Example: *for i=1, i<15, i=i+1 : write.*

DBappend (fname)
Appends the currend record to the end of the file. There is no need to position the file to the last record for this operation.

Parameters:

Fname file name

DBclose (fname)
Closes a file that has been opened with DBopen.

Parameters:
Fname file name

DBcreate (fname,,fvect)
Creates a new data file. The extension .d00 will automatically be added to the specific name.

Parameters:
Fname file name
Fvect feature vector as specified in the introduction

DBerase (fname,,fvect)
Erases a data file (equivalent to the MS-DOS del command).

Parameters:
Fname file name

DBopen (fname,,fvect)
Opens an existind data file. Any access to a data file (read, write) must be preceded by this opening step (or by ***DBcreate***).

Parameters:
Fname data file name. A string of up to 8 characters, that constitutes a valid file name under MS-DOS.
Fvect a return parameter, describing the feature vector used to create the file.

dilate (input, output, shape, phase, count)
Dilates objects with the grey level ***phase*** in binary or grey images, using various structuring element ***shape***. The dilation process can be iterated.

Parameters:
Input image to be dilated
Output the dilated image
Shape_? structuring element for the operation:
=1 horizontal vector
=2 vertical vector
=3 diagonal vector, 45 degrees
=4 diagonal vector, 135 degrees
=5 cross
=6 square
=7 octagon
=8 alternate squares/crosses
Phase selects the grey level for dilation
Count number of dilations

dis2lev (Input, Output, LevelLow, LevelHigh, <I>Binary, <I>Inter, mode)
This function is used for segmentation of binary or multiphase images from grey images. 2 threshold levels discriminates objects from the background by setting two thresholds. Either the grey levels inside or those outside the selected limits will be set to grey value 0 (black, background). Depending on the variable **Binary**, the remaining pixels either keep their original grey levels or are set to white (255). The thresholds can be selected numerically or set interactively, depending on the**Mode**. The interactive setting is performed in relation to the X- and Y- position of the cursor on the tablet. Thresholds in histogram is interactive thresholding via a gray level histogram copied into the input image.

Parameters:

Input	image to be segmented
Output	the segmented image
LevelLow	lower grey level threshold
LevelHigh	upper grey level threshold These variables will carry return values, if the thresholds are defined interactively.
Binary	_ON the discriminated pixels are set to 255 (white), all other pixels are
<logical>	set to 0 (background) _OFF the discriminated pixels keep their original grey levels
Inter	_ON the threshold levels are defined interactively, by moving the cursor in horizontal (**LevelLow**) and vertical direction (**LevelHigh**). A grey scale is temporarily inserted on top of the **Input** image, to facilitate threshold setting. The values defined interactively for **LevelLow** and **LevelHigh** are returned to the parameters after execution of the function. _OFF the threshold levels are taken as defined by **LevelLow** and **LevelHigh**.
Mode_?	to select the threshold definition, if **Inter** is _ON =1 to modify **LevelLow** and **LevelHigh** Interactively =2 to choose **LevelHigh** according to the value given to the parameters and change **LevelLow** interactively =3 to choose **LevelLow** according to the value given to the parameters and change **LevelHigh** interactively

disdyn (input, output, size, threshold, <I>binary, <I>darkbackgr, <I>inter)
Dynamic threshold generates a binary image by computing a single adaptive threshold. The function is particularly useful for the discrimination of small objects and narrow linear structures on varying backgrounds.

Parameters:

Input	image to be segmented
Output	the segmented image
Size (1-255)	matrix size for the lowpass filter used for the reference image
Threshold	threshold level, for numerical input. This variable will carry a return value if the threshold is defined interactively
Binary	_ON the discriminated pixels are set to 255 (white)

<logical>	_OFF	the discriminated pixels keep their original grey levels
DarkBackgr	_ON	if the object pixels are lighter than the background
<logical>	_OFF	if the object pixels are darker than the background
Inter	_ON	threshold level is defined interactively, by moving
<logical>		the cursor horizontally. A grey scale scale is temporarily inserted on top of the **Input** image to simplify threshold setting.
	_OFF	the threshold level is defined by **Threshold**

eraseinside (input, output, auxiliary, phase)
This function copies ***Input*** to ***Output*** and sets the pixels of an image inside a closed contour into a predefined grey value.

Parameters:

Input	image input
Output	output image
Auxiliary	auxiliary image memory
Phase	grey value to fill the inside of the contour

for loop
This loop is used to repeat the commands in the loop a number of times.

Syntax: ***for (start value, condition, increment)***
.
.
endfor

Or: ***for (start value, condition, increment) : command***

getim (file, image)
Get image loads an image stored on a disk medium into the image memory.

Parameters:

File	**f**ile name of the image to be loaded
Image	image memory into which the image is to be loaded

graphic
sets the alphanumeric display to graphic display mode.

Parameters:

none

identify (input, output, <l>conn_8, <l>refarea)
Identify connects pixels in a binary image, that are part of the same object, into one entity and creats a list of the objects defined by this connectivity test. Every object is assigned its own object label in the form of a grey level common to all its pixels. The identification is a prerequisite for any object-specific or densitometric measurements. The logical parameter ***RefArea*** is set to _ON, the reference area named FAREA in the system feature vector "scaling" is set to the complete image area.

Parameters:

Input	image to be identified	
Output	the identified image	
Conn_8	_ON	to select 8-connected objects
<logical>	_OFF	to select 4-connected objects
RefArea	_ON	sets reference area to the total area
<logical>	_OFF	the reference area is not changed

InitField (parname)
This function is used to declare individual or multiple field specific parameters for measurement. ***ParName*** selects the parameter. Multiple parameter names must be separated by a comma. The parameters declared with *InitField* are inserted into the *Field* feature vector. For system-declared features one uses the *Init* commands. For user-defined elements of feature vectors the command

global <parameter>

has to be used, to allow the system global access to the variable. It is suggested that upper-case letters are used for parameter names, to indicate they are treated differently from ordinary local variables.

Parameters:

ParName	name(s) of the parameter(s) to be initialized. i.e. AREAP, TOTALAREA, FIELDCOUNT, etc.

InitObj ANGLEDMAX,PERIM
This command defines the features to be measured. By executing the above command, it will ensure that the measurement commands will provide the requested parameters.

Parameters:

ANGLEDMAX angle between the positive X-axis and the DMAX orientation of the object. DMAX is the longest diameter of an object obtained by selecting the largest of the Feret diameters measured in 32 different directions, i.e. at an angular resolution of 5.7 degrees

PERIM	the perimeter of an object in an image, calculated as: PERIM = PERIMX + PERIMY + PERIMXY.$\sqrt{2}$ PERIMX, PERIMY length of perimeter having x (PERMX) and in y (PERIMY) direction. PERIMXY Length of perimeter having direction of 45 and 135 degrees to x axis.

loadlut (file)
Loads a look-up table stored in subdirectory. ***File*** refers to the look-up table file to be loaded.

Macros
Macros are command sequences. The concept of macros is helpful to efficiently structure complex programs.

Measf (image)
Measf measures field-specific parameters of the image. The image to be measured has to be binary. If the parameter COUNT is to be measured, the input image has to be identified. Parameters measured are members of the *Field* feature vector.

Parameters:

Image	the input image

Measo (image)

Measo measures the previously defined object-specific parameters of the next object in the identified image.

Parameters:

Image	the input image

morpho 3x3 (input, output, file, operator, threshold, mode, iteration)

This function executes "hit-or-miss" transformations in binary images, according to the theory of mathematical morphology [*].
Structural elements of size 3x3 can be arbitrarily defined, large matrix sizes can be described by sequential execution of 3x3 transformations.

Parameters:

Input	Name of the input image. The image can be an arbitrary 8 bit grey image.
Output	Name of the output file. This is an 8 bit image in which only two grey values are present: 0 (logical 0) and 255 (logical 1)
File	Name of the MDL text file, without its file type .mdl.
Operator	Name of an <element>, <parallel> or <sequence> object, which is defined in MDL text file.
Threshold	Threshold for internally converting the input image into a binary image. Grey value < threshold means logical 0 Grey value > = threshold means logical 1
Mode_?	Margin definitions: = 0 The input image is embedded in a logical 0 environment. = 1 All margin pixels in the input image are internally set to 1 before morpho-operations are carried out. = 2 The input image is embedded in a logical 1 environment. = 3 Combination of **Mode** 1 and 2.
Iteration	Number of iterations A value of 0 is an exception.

open (input, output, shape, phase, count)

Objects in a binary or grey image are first eroded using a structuring element defined by ***shape_?***, and then dilated using the same structuring element. This procedure eliminates small objects without strongly affecting the larger objects. The opening can be iterated.

Parameters:

Input	image to be processed
Output	the processed image

Shape_?	structuring element for the operation: =1 horizontal vector =2 vertical vector =3 diagonal vector, 45 degrees =4 diagonal vector, 135 degrees =5 cross =6 square =7 octagon =8 alternate squares/crosses
Phase	selects the grey level for opening
Count	number of operations to be performed

outlist (database,<I>print)
A list of data measured and stored in the file **Database** is displayed on the monitor or printed out.

Parameters:

DataBase	name of the data file to be listed
Print	_ON print the data list of the whole database
<logical>	(it will not appear on the monitor)
	_OFF display the data on the monitor

resetpar
Erases **all** feature vectors defined.

resetvec (vector)
Erases a selected feature vector. This function can also be used to set all scale factors to the default unity values.

Parameters:

Vector	string variable to specify the feature vector to be deallocated.

scrap (input, output, <l>select, arealow, areahigh, <l>binaryinp, <l>binaryout)
Scrap eliminates objects on the basis of their area, in pixel units. The input and the output image can be of type "binary" or "identified". Objects in the size range **[AreaLow...AreaHigh]** are selected or rejected. The maximum value for **AreaHigh** is 65535 pixels.

Parameters:

Input	image to be processed. if it is a binary image, the variable **BinaryInp**must be set to _ON.
Output	the processed image. If an identified image is required, the value **BinaryOut** must be set to _ON.
Select	_ON objects with area in the range
<logical>	**[AreaLow...AreaHigh]** are retained
AreaLow	lower limit for the area range (in pixels)
AreaHigh	upper limit for the area range (in pixels)
BinaryInp	_ON if the **Input** image is binary
<logical>	_OFF if the **Input** image is identified

BinaryOut	_ON	if the **Output** image will be a binary image
<logical>	_OFF	if the **Output** image will be an identified image

setframe (frame)
Selects the format of the images to be processed.

Parameters:

Frame	512 * 480 pixels, image directory name "F512"

setimpath (imagepath)
Selects the path to the subdirectory for the storage and retrieval of images on disk media.

Parameters:

ImagePath	DOS directory path for the storage or retrieval of images

scalgeom (image, scale, <I>inter, <I>list)
To determine the scaling factors at a given magnification, a reference image, containing structures of known dimensions (e.g. a ruler), must be available.

Parameters:

Image		reference image or scaling
Scale		name of the scale (only active when **List** is _OFF)
Inter	_ON	interactive definition of the scaling factors
<logical>	_OFF	scaling factors are determined by scaling 2 points. The image is displayed.
List	_ON	a list of scaling parameters which have been selected or are to be edited will be displayed
	_OFF	the scaling factors in the "scale" in the file geom.def will be used.

scalim (input, output, lowin, highin, lowout, highout, phaselow, phasehigh, mode)
Scale grey value performs a linear grey level scaling of the image defined by **Input** and **Output** grey level range. The grey level ranges in the **Input** and **Output** images are defined via the various **LowIn / HighIn** and **LowOut / HighOut**, respectively. Grey levels which lie below or above the selected limits (**LowIn** and **HighIn**) can be treated in different ways, depending on the variable **Mode**.

Parameters:

Input	image to be scaled
Output	the scaled image
LowIn	lower grey level in the input range
HighIn	upper grey level for the input range
LowOut	lower grey level for the output range
HighOut	upper grey level for the output range
PhaseLow	grey level, in the output image, for those pixels in the input which are darker than **LowIn**, (depends on **Mode)**.
PhaseHigh	grey level, in the output image, for those pixels in the input image which are brighter than **HighIn**, (depends on **Mode)**.
Mode_?	defines the treatment for the grey levels outside the range **LowIn-HighIn**:

= 1 grey levels outside **LowIn** and **HighIn** are not changed
= 2 grey levels outside **LowIn** and **HighIn** are mapped to the values
LowOut or **HighOut**, respectively.
= 3 grey levels outside **LowIn** and **HighIn** are mapped to the grey
levels **PhaseLow** and **PhaseHigh**

read (arg1, arg2, arg3,...)
Read is used to read in values interactively.

Parameters:
arg1, 2, represents constants, variables or expressions, with or without brackets.

thinbin (input, output, steps, mode, <l>invert, <l>endpoints)
Binary thinning.thins white linear structures in binary images (skeletonization). For every thinning step, pixels that are not relevant for the connectivity of an object are removed from the object margins, i.e., converted into background pixels. The coonectivity of objects is thus maintained. This procedure can be continued until all objects are reduced to one-pixel-wide that approximate the skeletons. Convex objects are thinned to very short skeleton lines, or even to single pixels. Line ends generated in the thinning process are not essential for connectivity but are often necessary for the required skeleton. If **Endpoints** is switched on, pixels at those line ends are not removed by further thinning steps. **Steps** determines the number of thinning steps to be performed. Thinning stops automatically as soon as no more object pixels can be removed, regardless of the current value of **Steps**. If **Steps = 0**, thinning will be continued until the automatic stop is reached.

Parameters:

Input	image to be processed
Output	the thinned image
Steps	number of thinning steps = 0 terminates the execution automatically when all structures have been reduced to the width of one pixel
Mode_?	controls the effects of the function at the borders of the image: = 0 pixels outside the image borders are regarded as background (grey level 0), single pixes will be erased = 1 a one-pixel wide line of grey level 255 is inserted on the image borders, single pixels will be erased = 2 pixels outside the the image borders are regarded as objects (grey level = 255). Therefore structures touching the frame will be connected to this visual object, single pixels will be erased. = 3 skeleton for fibres

	= 4	same as **Mode 0**, but: single pixels will be preserved
	= 5	same as **Mode 1**, but: single pixels will be preserved
	= 6	same as **Mode 2**, but: single pixels will be preserved
Invert <logical>	_ON	the image is inverted before processing
EndPoints <logical>	_ON	line ends are not shortened

while loop

This loop is used to repeat the commands in the loop as long as the condition is valid.

Syntax:

```
while (condition)
.
.
endwhile
```

write (arg)

Write is used to output text and variables.

Parameters:

arg	represents constants, variables or expressions, with or without brackets.

xorim (input1, input2, output)

Xorim (exclusive or) correlates every pixel that is either in image 1 or image 2 but not common to image 1 and image 2 and the result outputed to image 3. ***Input1*** and ***Input2*** are the two binary images to be correlated.

Program Running Procedures for the Du Model

MCPP is written in FORTRAN 77, compiled using the Microsoft FORTRAN compiler and executed with McFortran/020. The executable code is *MCPP.apl*. MCPP can be executed by clicking on *MCPP.apl*. The program loads the input files and displays the used input parameter integer numbers.

1. ***nch-*** If nch=1, MCPP deals with at least one boundary, if nch = any other integer numbers, MCPP deals with no boundary.

2. ***number-*** is a boundary type integer. If number = 1, 2, or 3, MCPP deals with rectangular, circular, or roller cutter boundary, respectively. If the boundary type is rectangular, the user needs to input 1.

3. ***nb1-*** is total boundary element number.

4. ***nboun-*** is the number of straight sides of a rectangle, for example, nboun = 4 for a rectangle or square.

5. ***itpt-*** itpt is a user controlled program iteration number. Iteration (Itr) starts from 1 and increases by 1 only after crack(s) grows (grow) by an incremental length. The program will stop when Itr reaches the user input number. For example, if itpt = 5, program will check whether there is a growing crack; if there is, the growing crack(s) will increase incrementally by one length, and the Itr increase by 1 until Itr reaches 5. If there is no growing crack(s) during iteration, Itr will keep its prior value and program will either load to check the growing crack(s) again, or terminate running the program.

6. ***E-*** is user input Young's modulus.

7. ***gcc-*** is user input critical fracture toughness.

8. ***pxx, pyy, pxy-*** are user input far field x-direction normal stress, y-direction normal stress, and shear stress, respectively.

9. ***delpxx, delpyy, delpxy-*** are user input far field x-direction incremental normal stress, y-direction normal stress, and shear stress, respectively.

10. ***delbxx, delbyy, delbxy-*** are user input boundary x-direction incremental normal stress, y-direction incremental normal stress, and incremental shear stress, respectively.

11. ***rdl-*** is the ratio of the incremental length of a growing crack to the length of a crack tip element.

12. ***numos-*** If numos = 0, program does not calculate the stress, strain, and displacement within the body; if numos = any other number, it does.

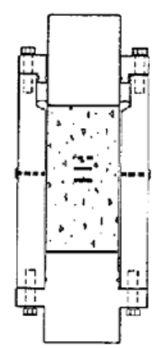
13. ***ncboun-***If ncbouns = 0, program does not calculate the stress, strain, and displacement on the boundary; if ncboun = any other number, it does.

The propagation of multiple cracks can be illustrated by a graphic executable code- *graphic.apl*, and the user only needs to input the total boundary element number after double clicking on *graphic.apl*.

APPENDIX C: TEST CELL DESIGN

UNIVERSITY OF CALIFORNIA, BERKELEY

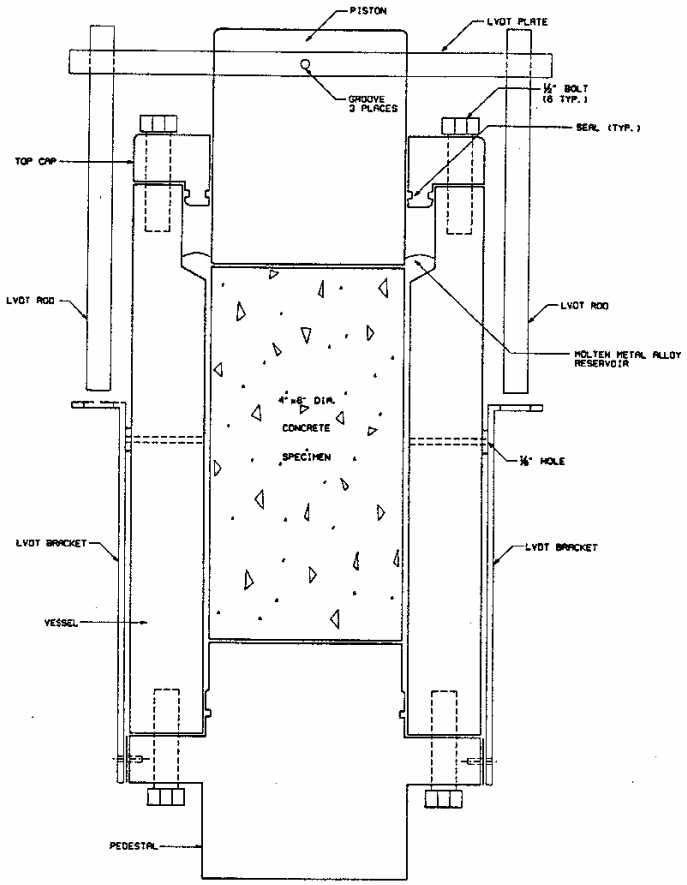
INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS



COLLEGE OF ENGINEERING
DEPARTMENTS OF CIVIL & MINING ENGINEERING

KAMRAN M. NEMATI
PROF. PAULO J. MONTEIRO
PROF. NEVILLE G. W. COOK
~ DR. LARRY MYERS

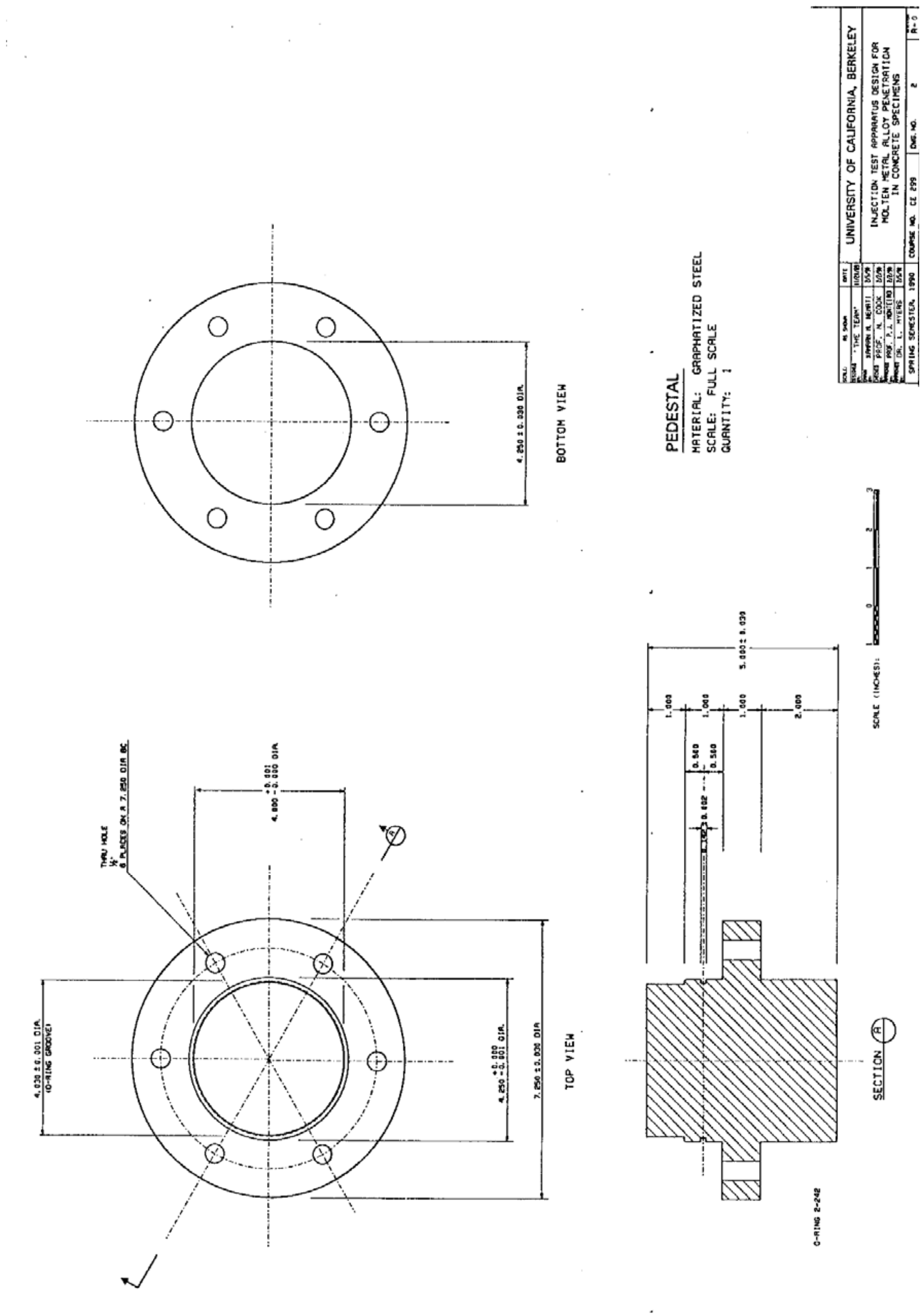
FEBRUARY, 1990

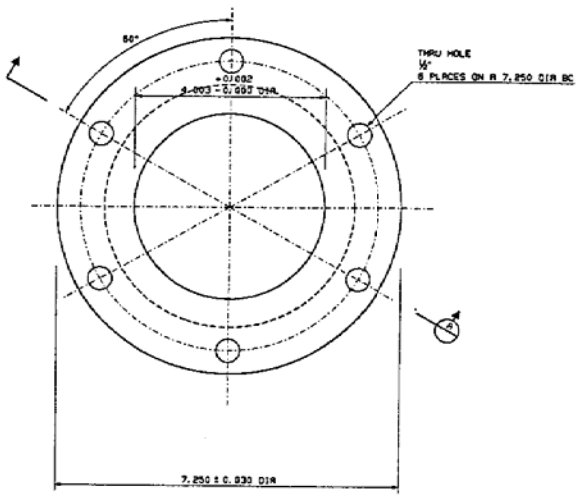


INJECTION TEST APPARATUS

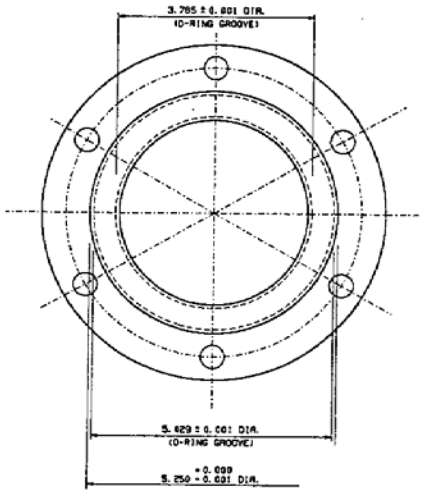
SCALE (INCHES): 1 0 1 2 3

SCALE: AS SHOWN	DATE:	UNIVERSITY OF CALIFORNIA, BERKELEY
DRAWN BY: "THE TEAM"	11/21/95	
CHKD BY: KAMRAN M. NEMATI	2/5/96	INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS
APPROVED BY: PROF. N. COOK	2/2/96	
APPROVED BY: PROF. P. J. MONTEIRO	2/2/96	
APPROVED BY: DR. L. MYERS	2/5/96	
SPRING SEMESTER, 1996	COURSE NO. CE 299	DWG. NO. 1 REVISION R-0

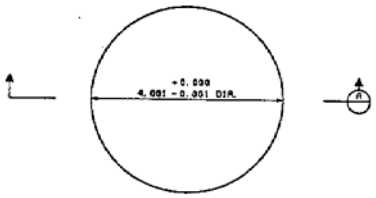

UNIVERSITY OF CALIFORNIA, BERKELEY
INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS
PEDESTAL
MATERIAL: GRAPHATIZED STEEL
SCALE: FULL SCALE
QUANTITY: 1
BOTTOM VIEW
TOP VIEW
SECTION A
7.250 ± 0.030 DIA
5.000 ± 0.030
O-RING 2-242
SCALE (INCHES):



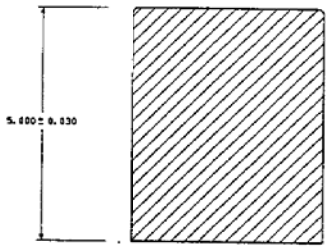
TOP VIEW



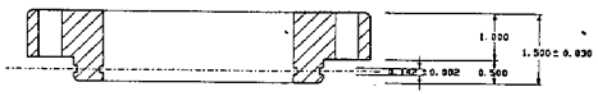
BOTTOM VIEW



TOP VIEW



SECTION (A)



SECTION (A)

O-RING 2-240
O-RING 2-250

TOP CAP

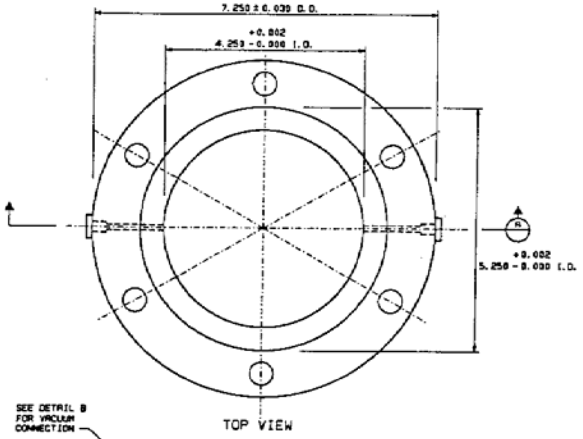
MATERIAL: GRAPHATIZED STEEL
SCALE: FULL SCALE
QUANTITY: 1

PISTON

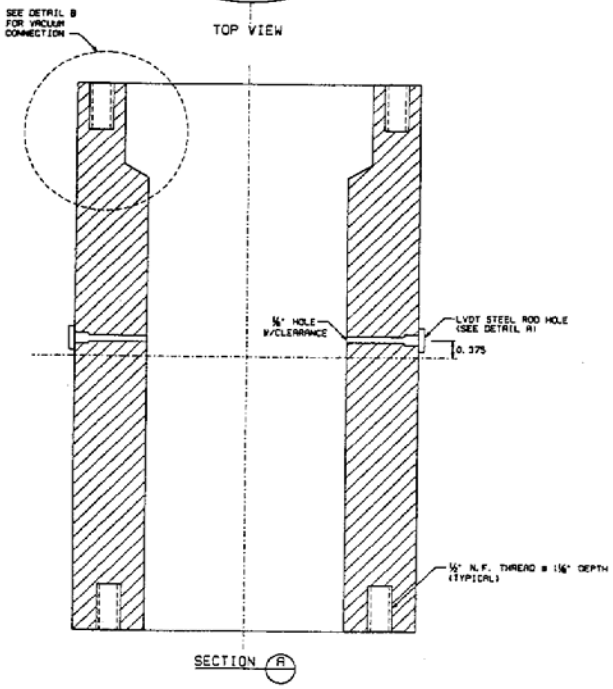
MATERIAL: HIGH STRENGTH
SCALE: FULL SCALE
QUANTITY: 1

SCALE (INCHES): 0 1 2 3

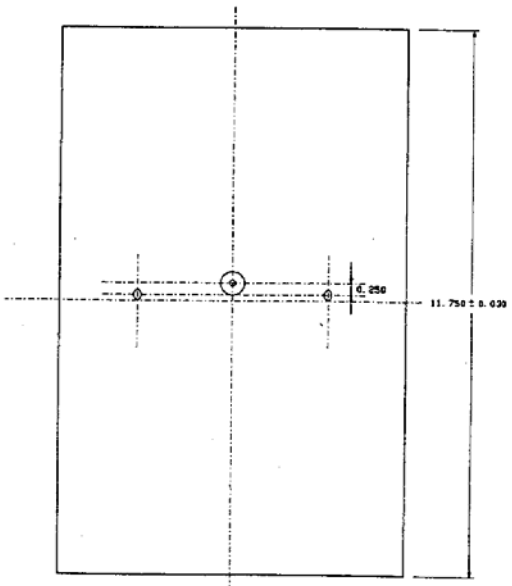
AS SHOWN	DATE	UNIVERSITY OF CALIFORNIA, BERKELEY	
"THE TEAM"	11/21/89		
KIERAN N. HEARTY	5/5/90	INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS	
PROF. N. COOK	5/5/90		
PROF. P.J. MONTEIRO	5/5/90		
DR. L. MYERS	5/5/90		
SPRING SEMESTER, 1990		COURSE NO. CE 299	DWG. NO. 3 R-0



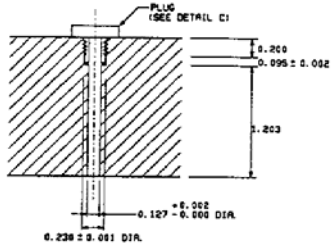
TOP VIEW



SECTION A

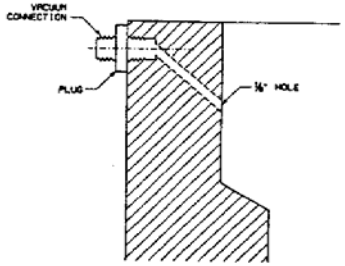


SIDE VIEW



EXPANDED SECTION VIEW
SCALE: DOUBLE SCALE
DETAIL A

PLUG
SCALE: DOUBLE SCALE
DETAIL C



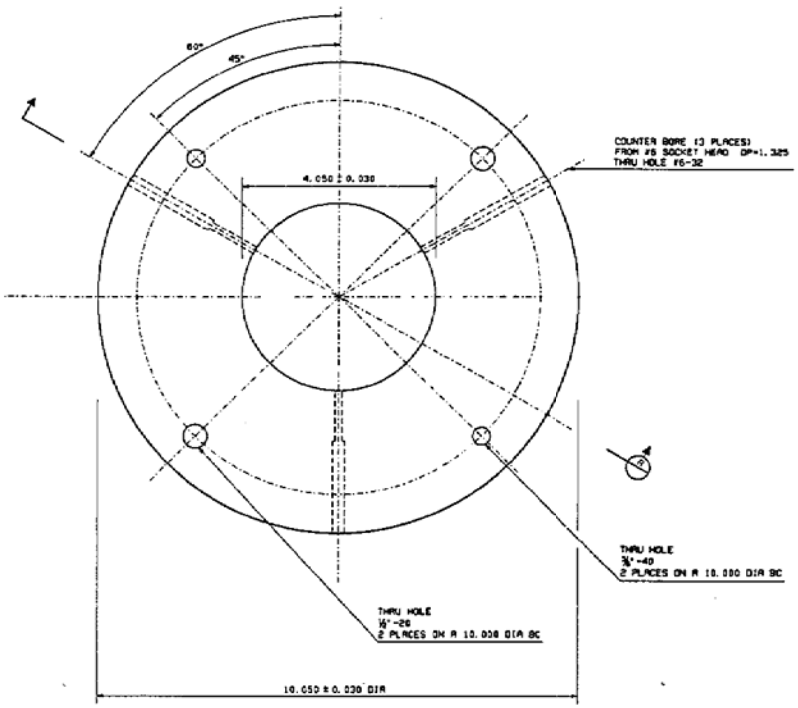
VACUUM CONNECTION
SCALE: DOUBLE SCALE
DETAIL B

VESSEL

MATERIAL: HIGH STRENGTH
SCALE: FULL SCALE
QUANTITY: 1

SCALE (INCHES): 1 0 1 2 3

	"THE TEAM"	12/2/89	UNIVERSITY OF CALIFORNIA, BERKELEY
	KAMRAN M. NEMATI	2/5/90	INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS
	PROF. N. COOK	2/5/90	
	PROF. P. J. MONTEIRO	2/5/90	
	DR. L. MYERS	2/5/90	
SPRING SEMESTER, 1990	COURSE NO. CE 299	DWG. NO. 4	A-3

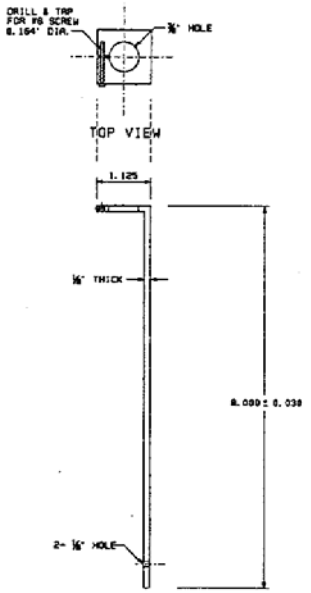


TOP VIEW

SECTION A

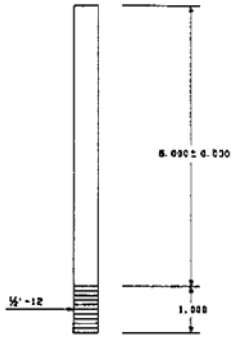
LVDT PLATE (TOP)

MATERIAL: ALUMINIUM
SCALE: FULL SCALE
QUANTITY: 1



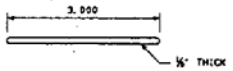
LVDT BRACKET

MATERIAL: MILD STEEL
SCALE: FULL SCALE
QUANTITY: 2
(TO BE ATTACHED TO PEDESTAL)



LVDT ROD

MATERIAL: MILD STEEL
SCALE: FULL SCALE
QUANTITY: 2



STEEL ROD

MATERIAL: MILD STEEL
SCALE: FULL SCALE
QUANTITY: 2

SCALE (INCHES): 0 1 2 3

SCALE: AS SHOWN	DATE:	UNIVERSITY OF CALIFORNIA, BERKELEY	
DESIGNED BY: "THE TEAM"	11/20/98		
DRAWN BY: KARRAN A. NEMATI	05/98	INJECTION TEST APPARATUS DESIGN FOR MOLTEN METAL ALLOY PENETRATION IN CONCRETE SPECIMENS	
CHECKED BY: PROF. N. COOK	05/98		
APPROVED BY: PROF. P.J. MONTEIRO	05/98		
DRAWING BY: DR. L. MYERS	05/98		
SPRING SEMESTER, 1999	COURSE NO. CE 299	DWG. NO. 9	REV. R-0