APPENDIX A

Formulation of failure surface for the Damaged Concrete Plasticity Model

Concrete plasticity is defined by plastic flow and yield function parameters. These parameters are related to effective stress that is defined as:

$$\bar{\sigma} = \mathrm{D}_0^{\mathrm{el}} : (\varepsilon - \varepsilon^{pl}) \tag{A1}$$

The plastic flow function and the yield surface make use of two stress invariants of the effective stress tensor, namely the hydrostatic pressure stress ($\bar{p}$) and the Mises equivalent effective stress ($\bar{q}$) which are calculated using effective stress deviator ($\bar{S}$) as shown in Eq. A2 and Eq. A3 respectively.

$$\bar{p} = -\frac{1}{3}\ \mathrm{trace}\ (\bar{\sigma}) \tag{A2}$$

$$\bar{q} = \sqrt{\frac{3}{2}(\bar{S} : \bar{S}} \tag{A3}$$

where

$$\bar{S} = \bar{\sigma} + \bar{p} I \tag{A4}$$

The plastic flow potential is given by the Drucker-Prager hyperbolic function as:

$$\mathrm{G} = \sqrt{(\varepsilon\sigma_{to} \tan\psi)^2 + \bar{q}^2} - \bar{p} \tan\psi \tag{A5}$$

where ψ is the dilation angle measured in the p-q plane at high confining pressure. A typical value of this angle for concrete equal to 15° was used in modelling the grout material of the WSRM walls.

$$\sigma_{to} = \sigma_{\mathrm{t}}\big|_{\varepsilon_t^{pl}=0} \tag{A6}$$

is the uniaxial tensile stress at failure, taken from tension stiffening data.

The plastic flow potential is of the non-associated type; however this does not make a difference in explicit analysis because the stiffness matrices are not calculated in this type of analysis.

Yield surface of concrete in plane stress space is shown in Fig. A1.

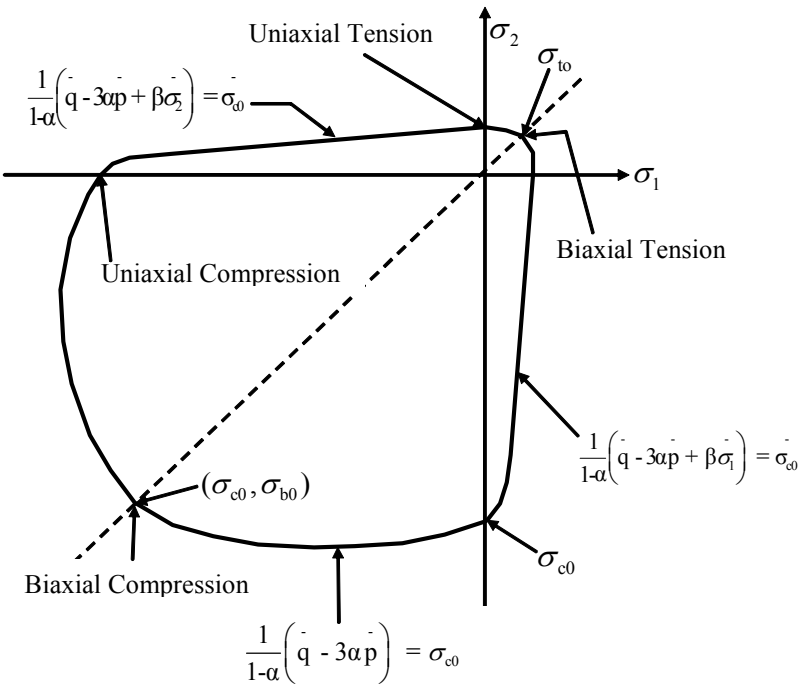


Figure A 1: Yield Surface of concrete in plane stress space

This yield surface makes use of a function provided by Lubliner et al. (1989) to account for the different evolution of strength under tension and compression. In terms of effective stresses, the yield function takes the form:

$$F = \frac{1}{1-\alpha}\left(\bar{q} - 3\alpha\bar{p} + \beta(\bar{\varepsilon}^{pl})\left\langle\bar{\sigma}_{max}\right\rangle - \gamma\left\langle-\bar{\sigma}_{max}\right\rangle\right) - \sigma_c(\bar{\varepsilon}_c^{pl}) = 0 \qquad \text{(A7)}$$

$$\alpha = \frac{(\sigma_{bo}/\sigma_{co})-1}{2(\sigma_{bo}/\sigma_{co})-1} \quad , \; 0 \le \alpha \le 0.5 \qquad \text{(A8)}$$

$$\beta = \frac{\bar{\sigma}_c(\bar{\sigma}_c^{pl})}{\bar{\sigma}_t(\bar{\sigma}_t^{pl})}(1 - \alpha) \qquad \text{(A9)}$$

$$\gamma = \frac{3(1-K_c)}{2K_c - 1} \tag{A10}$$

$\bar{\sigma}_{max}$ is the maximum principal effective stress, σ_{bo}/σ_{co} is the ratio of initial equi-biaxial compressive yield stress to initial uniaxial compressive yield stress. The default value for this parameter equal to 1.16 is used in the analysis of WSRM walls. $\bar{\sigma}_t^{pl}$ and $\bar{\sigma}_c^{pl}$ are the effective tensile cohesion stress and effective compression cohesion stress respectively.

APPENDIX B

Photographs of Masonry Prisms

Figure B 1: Samples of masonry prisms after uniaxial compression test

Figure B 2: A grouted masonry prism of wall #10 after uniaxial compression test

Figure B 3: A grouted masonry prism of wall #6 after uniaxial compression test

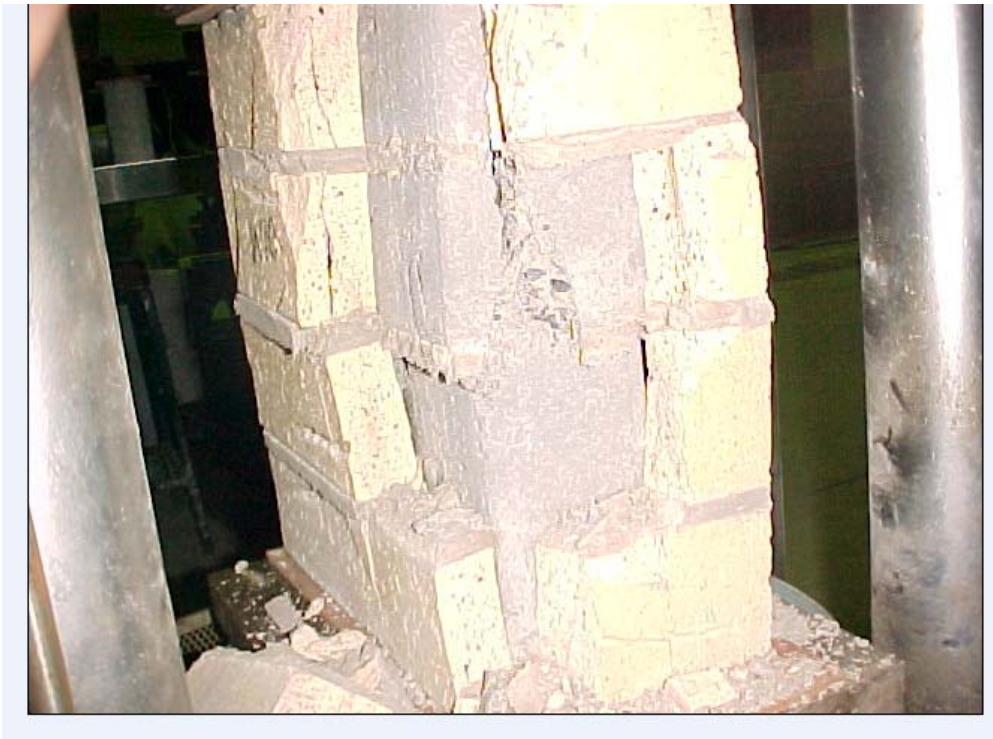


Figure B 4: Masonry shell removed from the grouted masonry prism after uniaxial compression test

Photographs of Grout Cylinders

Figure B 5: Grout cylinders after uniaxial compression test

Figure B 6: Grout cylinders of wall #5 after uniaxial compression test

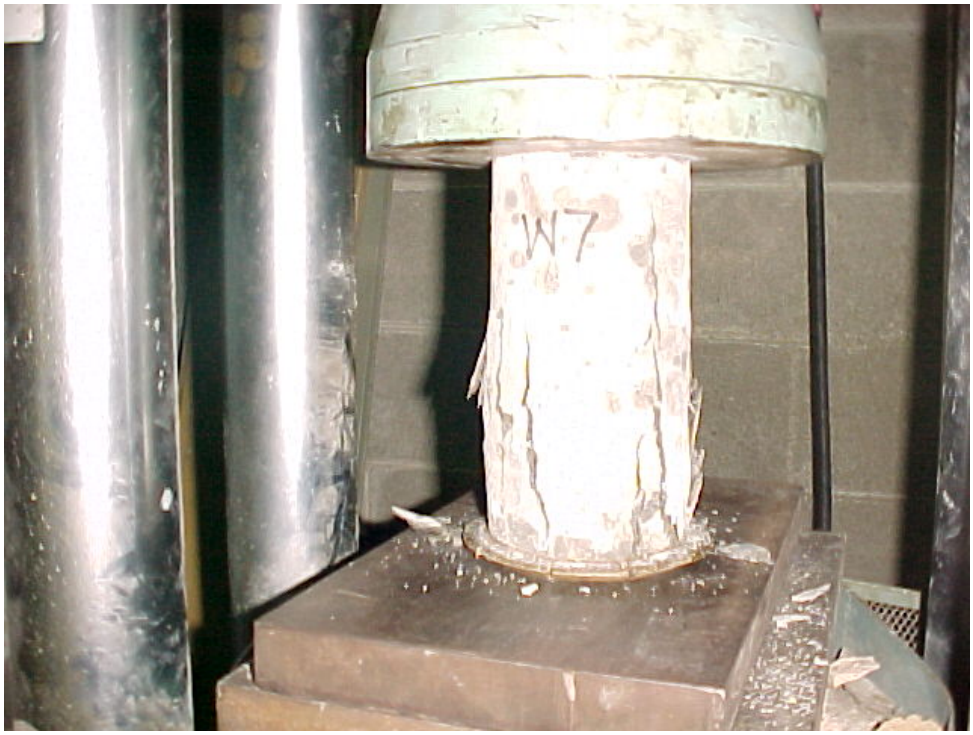


Figure B 7: Grout cylinder of wall #7 during uniaxial compression test

Figure B 8: Grout cylinders of wall #6 after uniaxial compression test

Photographs of Mortar Cubes

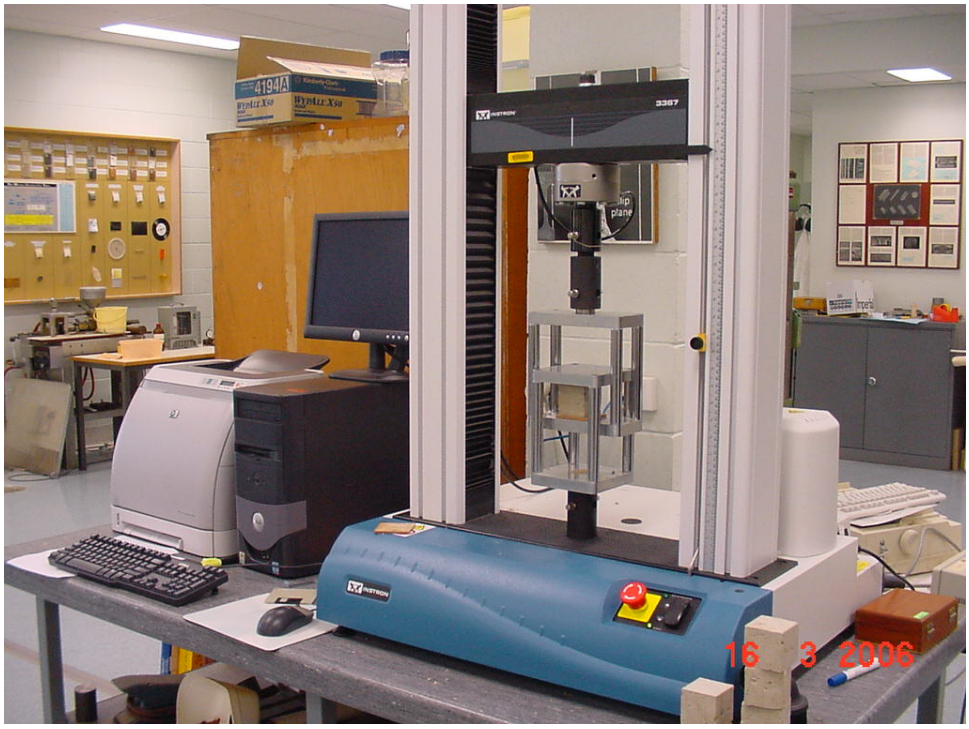


Figure B 9: Instran machine for mortar cubes testing

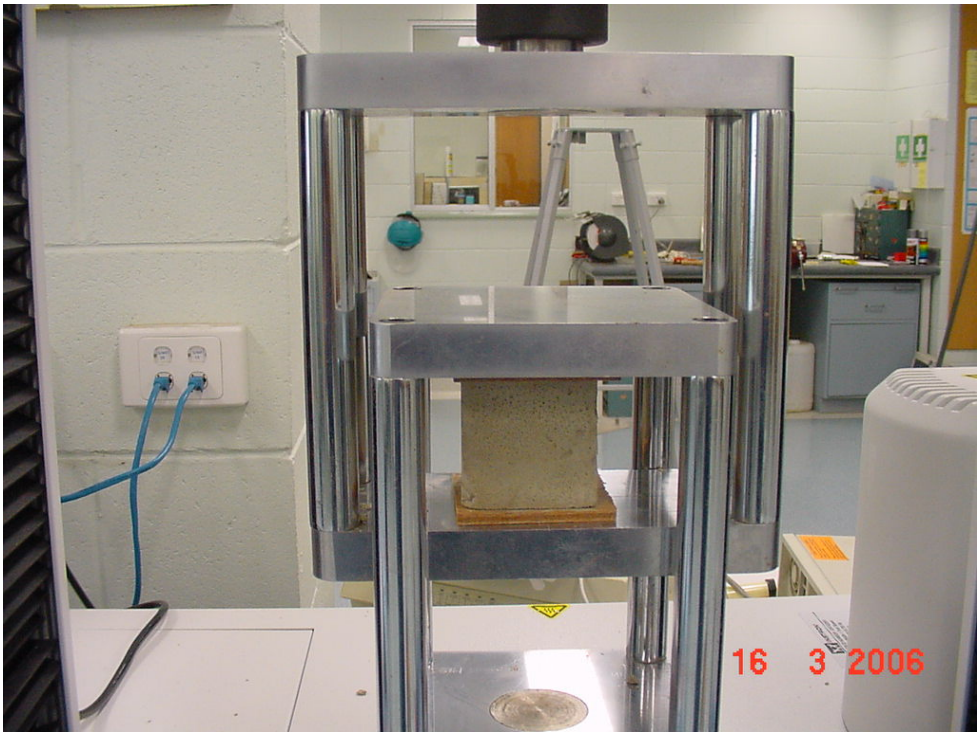


Figure B 10: Mortar cube under test

APPENDIX C

Load-displacement curves of Tested walls

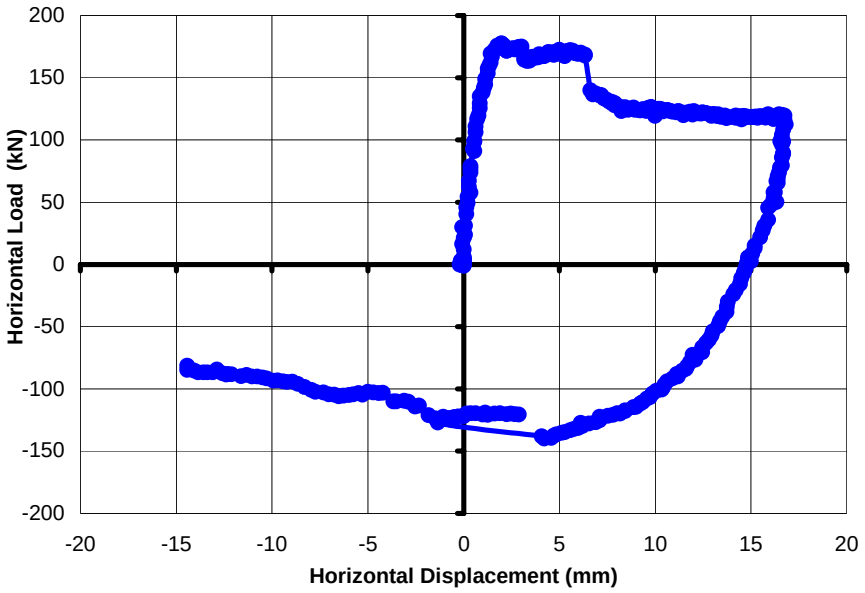


Figure C 1: Load-displacement response of Wall # 1

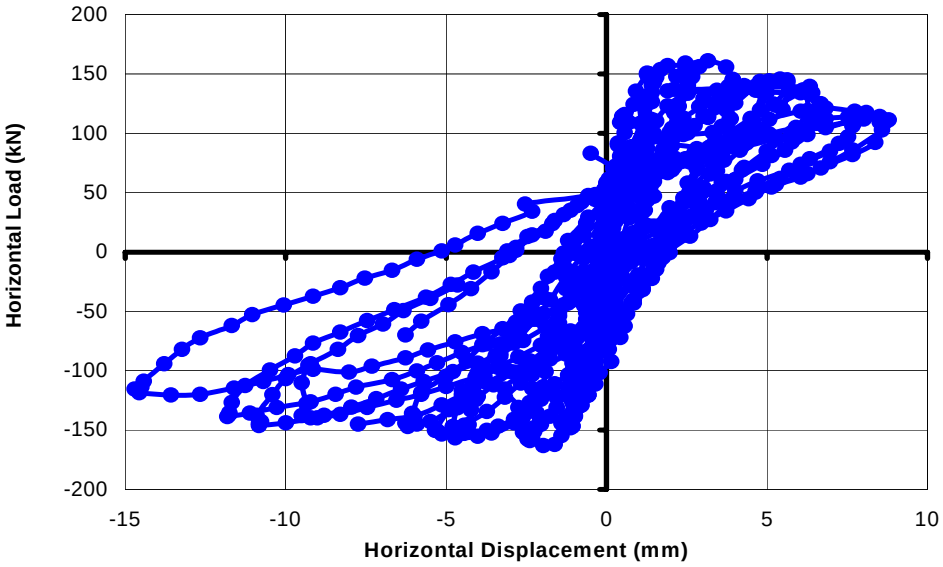


Figure C 2: Load-displacement response of Wall # 2

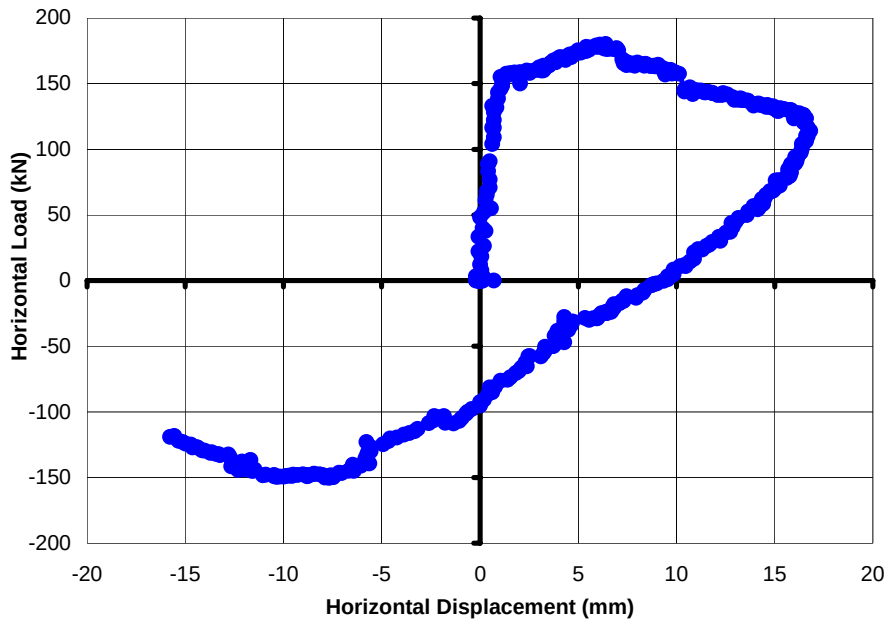


Figure C 3: Load-displacement response of Wall #3

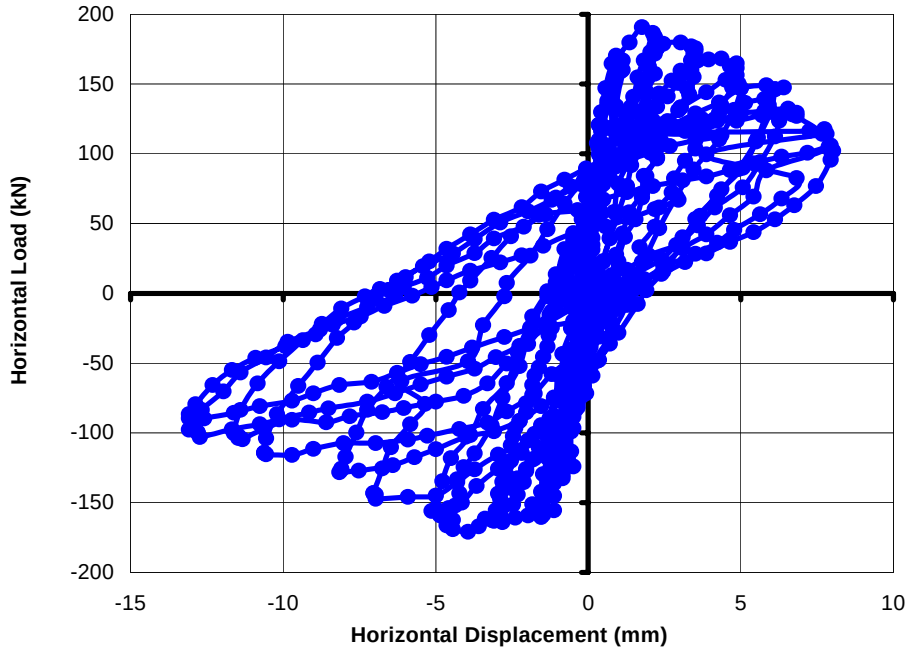


Figure C 4: Load-displacement response of Wall #4

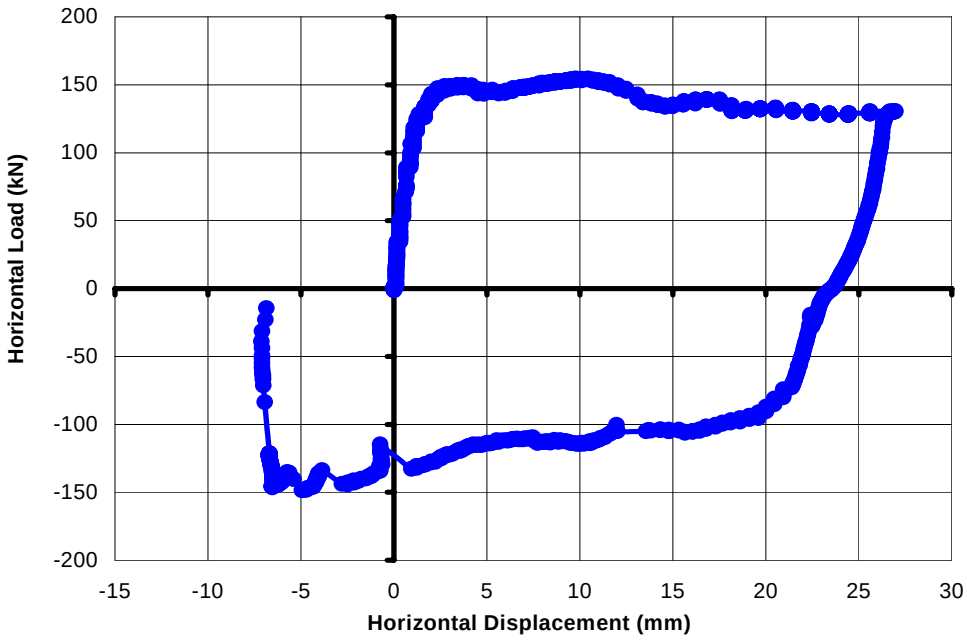


Figure C 5: Load-displacement response of Wall #5

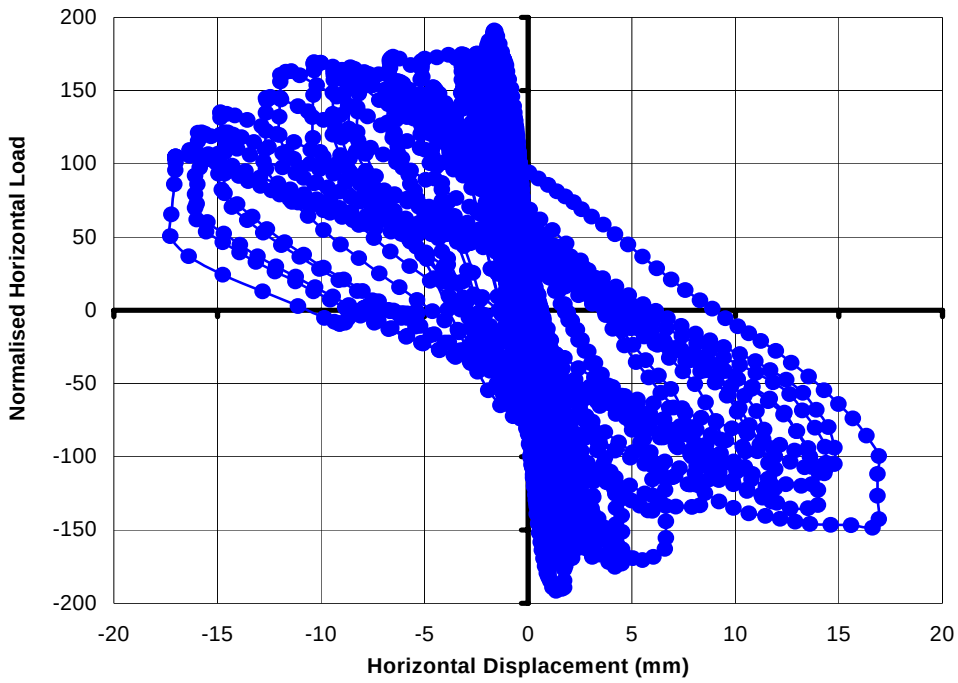


Figure C 6: Load-displacement response of Wall #6

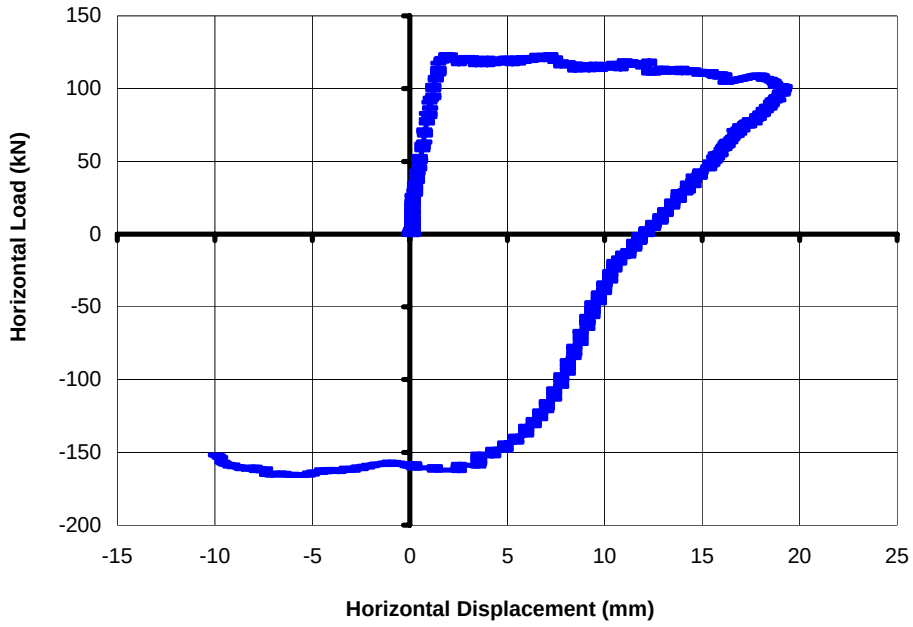


Figure C 7: Load-displacement response of Wall #7

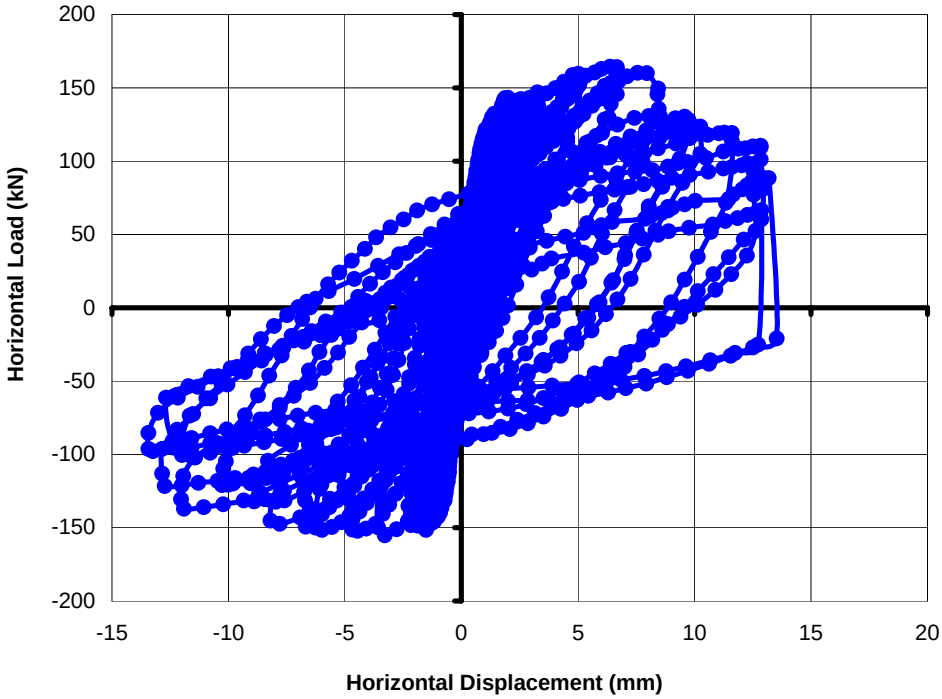


Figure C 8: Load-displacement response of Wall #8

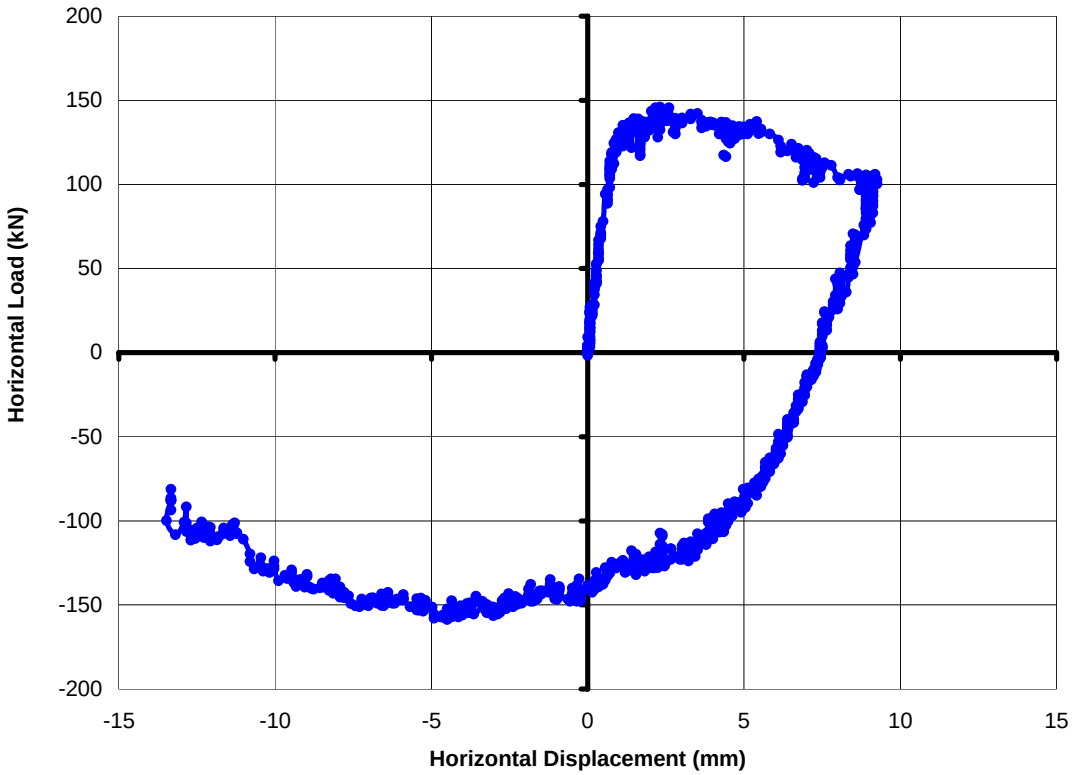


Figure C 9: Load-displacement response of Wall #9

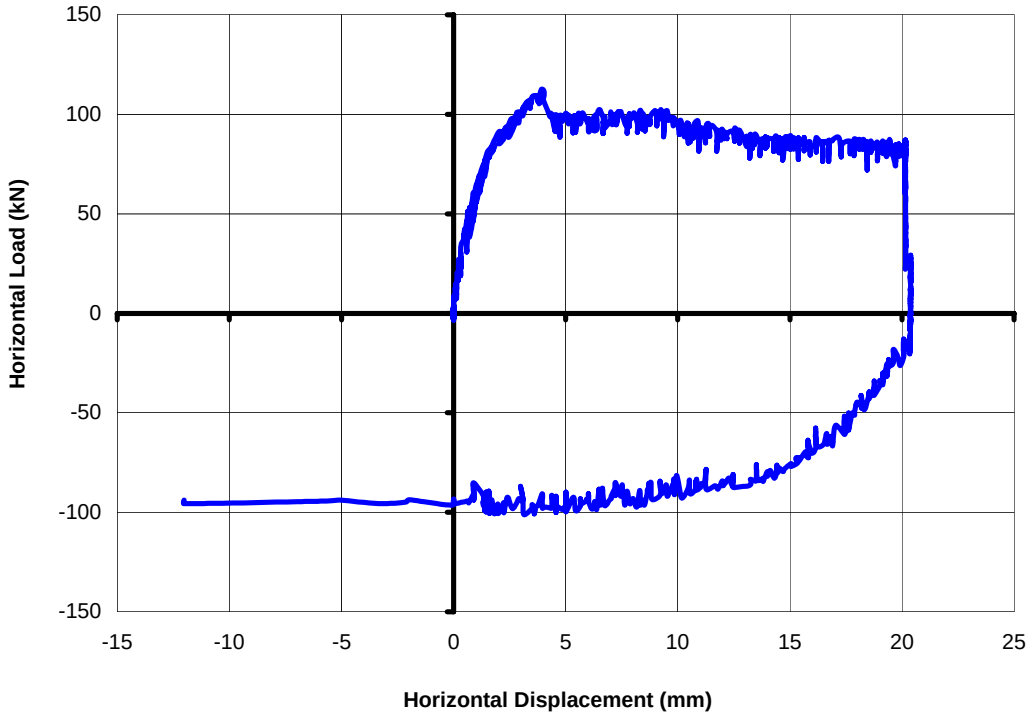


Figure C 10: Load-displacement response of Wall #10

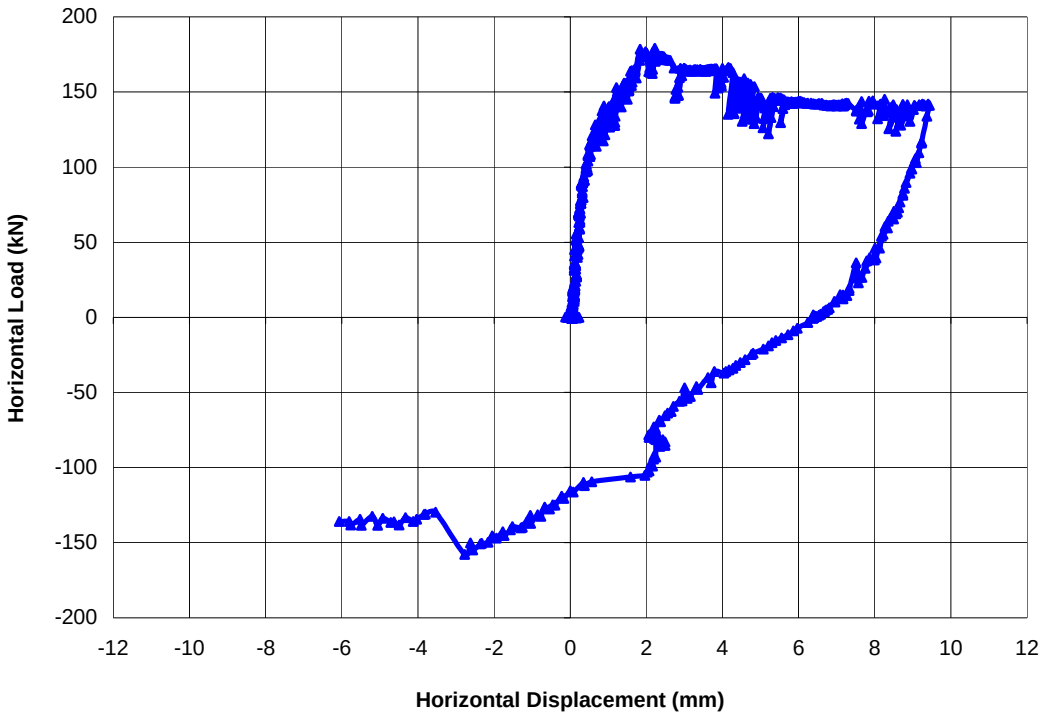


Figure C 11: Load-displacement response of Wall #11

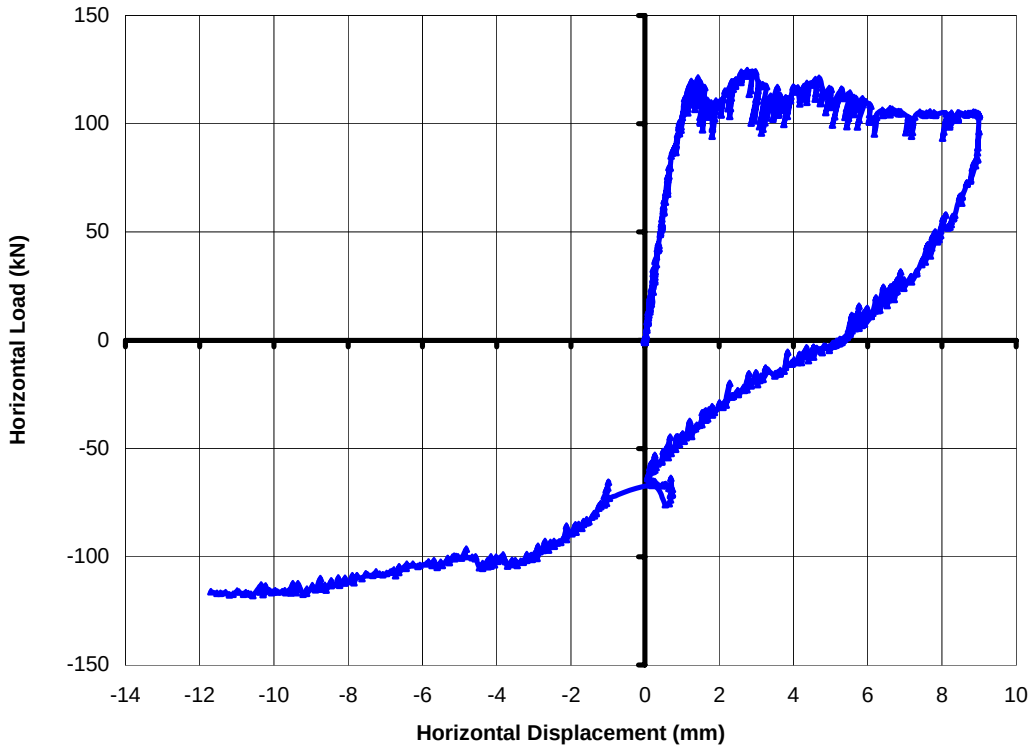


Figure C 12: Load-displacement response of Wall #12

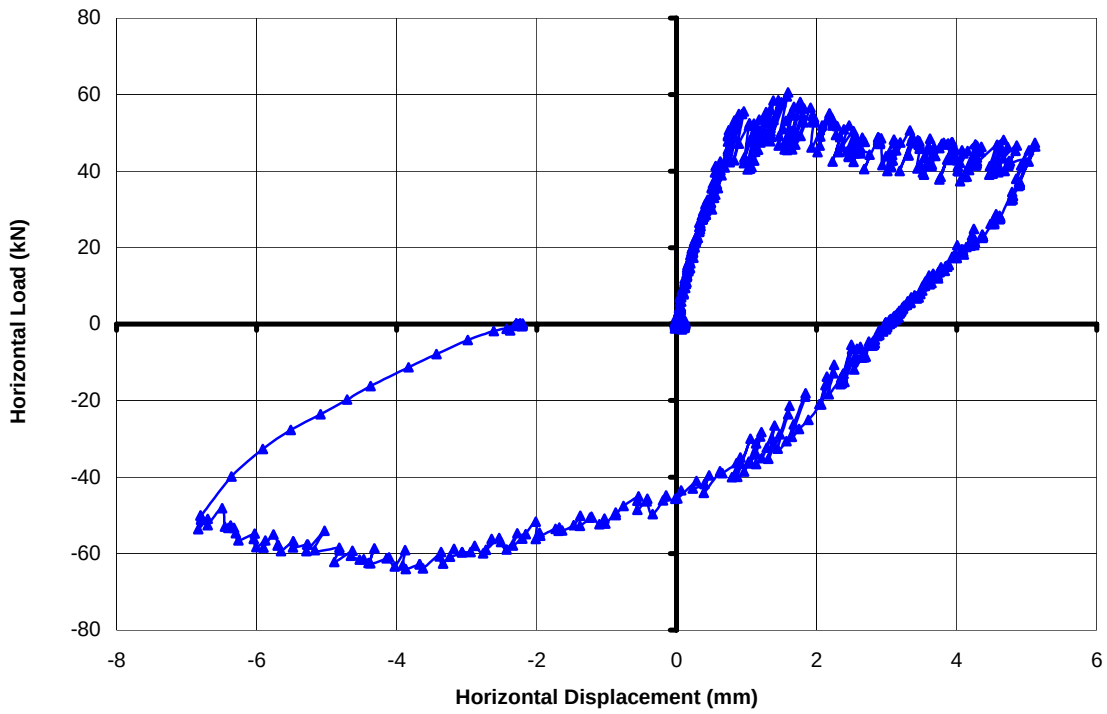


Figure C 13: Load-displacement response of Wall #13

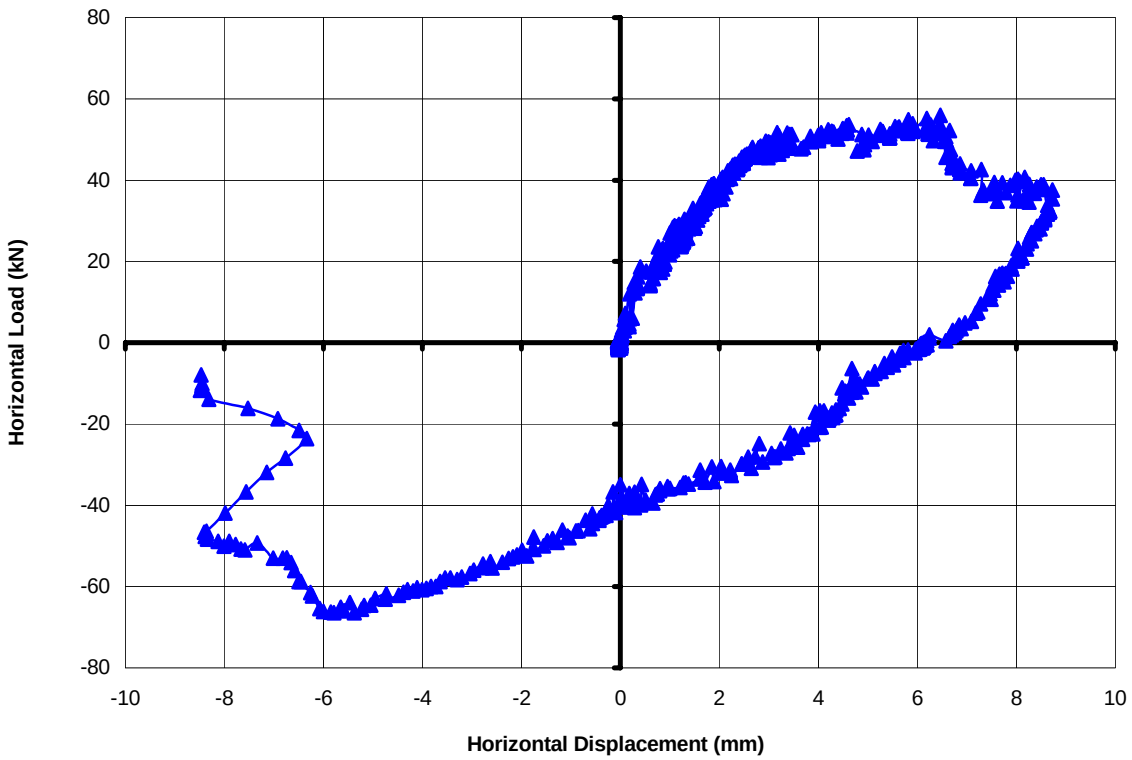


Figure C 14: Load-displacement response of Wall #14

APPENDIX D

VUMAT for Unreinforced Masonry

```
  SUBROUTINE VUMAT(
C READ ONLY (UNMODIFIABLE)VARIABLES -
  1  NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS,
        LANNEAL,
  2  STEPTIME, TOTALTIME, DT, CMNAME, COORDMP,
        CHARLENGTH,
  3  PROPS, DENSITY, STRAININC, RELSPININC,
  4  TEMPOLD, STRETCHOLD, DEFGRADOLD, FIELDOLD,
  5  STRESSOLD, STATEOLD, ENERINTERNOLD,
        ENERINELASOLD,
  6  TEMPNEW, STRETCHNEW, DEFGRADNEW,
        FIELDNEW,
C WRITE ONLY (MODIFIABLE) VARIABLES -
  7  STRESSNEW, STATENEW, ENERINTERNNEW,
        ENERINELASNEW )
C
  INCLUDE 'VABA_PARAM.INC'
C
  DIMENSION PROPS(NPROPS), DENSITY(NBLOCK),
        COORDMP(NBLOCK,*),
  1  CHARLENGTH(NBLOCK),
        STRAININC(NBLOCK,NDIR+NSHR),
  2  RELSPININC(NBLOCK,NSHR), TEMPOLD(NBLOCK),
  3  STRETCHOLD(NBLOCK,NDIR+NSHR),
  4  DEFGRADOLD(NBLOCK,NDIR+NSHR+NSHR),
  5  FIELDOLD(NBLOCK,NFIELDV),
        STRESSOLD(NBLOCK,NDIR+NSHR),
  6  STATEOLD(NBLOCK,NSTATEV),
        ENERINTERNOLD(NBLOCK),
  7  ENERINELASOLD(NBLOCK), TEMPNEW(NBLOCK),
  8  STRETCHNEW(NBLOCK,NDIR+NSHR),
  8  DEFGRADNEW(NBLOCK,NDIR+NSHR+NSHR),
  9  FIELDNEW(NBLOCK,NFIELDV),
  1  STRESSNEW(NBLOCK,NDIR+NSHR),
        STATENEW(NBLOCK,NSTATEV),
  2  ENERINTERNNEW(NBLOCK),
        ENERINELASNEW(NBLOCK)
C
  CHARACTER*80 CMNAME
C
  DOUBLE PRECISION  YOUNG(2), D(4,4),
        USRVAL(14), STATEV(2),
  &            STRESS(4), TEMP(4,4), DSTRAIN(4)
        USRVAL(1)=PROPS(1)
        USRVAL(2)=PROPS(2)
        USRVAL(3)=PROPS(3)
        USRVAL(4)=PROPS(4)
        USRVAL(5)=PROPS(5)
        USRVAL(6)=PROPS(6)
        USRVAL(7)=PROPS(7)
        USRVAL(8)=PROPS(8)
        USRVAL(9)=PROPS(9)
        USRVAL(10)=PROPS(10)
        USRVAL(11)=PROPS(11)
        USRVAL(12)=PROPS(12)
        USRVAL(13)=PROPS(13)
        USRVAL(14)=PROPS(14)

        YOUNG(1)=PROPS(15)
  YOUNG(2)=PROPS(16)

        E11=PROPS(15)
        E22=PROPS(16)
        E33=PROPS(17)
  V12=PROPS(18)
        V23=PROPS(19)
        V13=PROPS(20)
        GXY=PROPS(21)

        V21=V12/E11*E22
  V31=V13/E11*E33
  V32=V23/E22*E33
  DET=(1-V12*V21-V23*V32-V31*V13-
        2*V21*V32*V13)/E11/E22/E33
  D(1,1)=(1-V23*V32)/E22/E33/DET
  D(1,2)=(V12+V32*V13)/E11/E33/DET
  D(2,1)=D(1,2)
  D(1,3)=(V13+V12*V23)/E11/E22/DET
  D(3,1)=D(1,3)
  D(2,2)=(1-V13*V31)/E11/E33/DET
  D(2,3)=(V23+V21*V13)/E11/E22/DET
  D(3,2)=D(2,3)
  D(3,3)=(1-V12*V21)/E11/E22/DET
  D(4,4)=GXY
        D(4,1)=0.0D0
  D(4,2)=0.0D0
  D(4,3)=0.0D0
  D(1,4)=0.0D0
  D(2,4)=0.0D0
  D(3,4)=0.0D0

           DO 100 KM = 1,NBLOCK

  STRESS(1)=STRESSOLD(KM,1)
  STRESS(2)=STRESSOLD(KM,2)
  STRESS(3)=STRESSOLD(KM,3)
  STRESS(4)=STRESSOLD(KM,4)

        USRVAL(14)=CHARLENGTH(KM)

        STATEV(1)=STATEOLD(KM,1)
  STATEV(2)=STATEOLD(KM,2)

        DSTRAIN(1)=STRAININC(KM,1)
        DSTRAIN(2)=STRAININC(KM,2)
        DSTRAIN(3)=STRAININC(KM,3)
        DSTRAIN(4)=STRAININC(KM,4)
        ITER=0
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CCCCCCCCCCCCCCCCCCCCCCCCCCCC
  CALL USRMAT( DSTRAIN, 4, 0, 0, D, ITER,
        USRVAL,14,
 $           STATEV, 2, STRESS, TEMP, YOUNG )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCC

  STRESSNEW(KM,1)=STRESS(1)
  STRESSNEW(KM,2)=STRESS(2)
  STRESSNEW(KM,3)=STRESS(3)
        STRESSNEW(KM,4)=STRESS(4)

        STATENEW(KM,1)=STATEV(1)
  STATENEW(KM,2)=STATEV(2)

 100 CONTINUE

  RETURN
  END
```

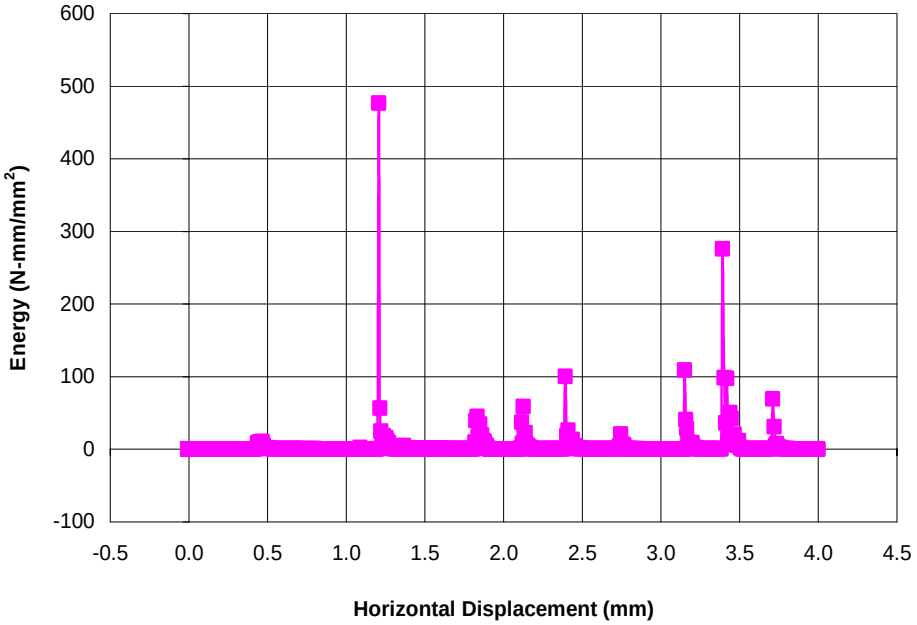
VUMAT For Reinforced Masonry

```
      SUBROUTINE VUMAT(C READ ONLY
     (UNMODIFIABLE)VARIABLES -
     1 NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS,
     LANNEAL,
     2 STEPTIME, TOTALTIME, DT, CMNAME, COORDMP,
     CHARLENGTH,
     3 PROPS, DENSITY, STRAININC, RELSPININC,
     4 TEMPOLD, STRETCHOLD, DEFGRADOLD, FIELDOLD,
     5 STRESSOLD, STATEOLD, ENERINTERNOLD,
     ENERINELASOLD,
     6 TEMPNEW, STRETCHNEW, DEFGRADNEW,
     FIELDNEW,
C WRITE ONLY (MODIFIABLE) VARIABLES -
     7 STRESSNEW, STATENEW, ENERINTERNNEW,
     ENERINELASNEW )
C
      INCLUDE 'VABA_PARAM.INC'
C
      DIMENSION PROPS(NPROPS), DENSITY(NBLOCK),
     COORDMP(NBLOCK,*),
     1 CHARLENGTH(NBLOCK),
     STRAININC(NBLOCK,NDIR+NSHR),
     2 RELSPININC(NBLOCK,NSHR), TEMPOLD(NBLOCK),
     3 STRETCHOLD(NBLOCK,NDIR+NSHR),
     4 DEFGRADOLD(NBLOCK,NDIR+NSHR+NSHR),
     5 FIELDOLD(NBLOCK,NFIELDV),
     STRESSOLD(NBLOCK,NDIR+NSHR),
     6 STATEOLD(NBLOCK,NSTATEV),
     ENERINTERNOLD(NBLOCK),
     7 ENERINELASOLD(NBLOCK), TEMPNEW(NBLOCK),
     8 STRETCHNEW(NBLOCK,NDIR+NSHR),
     8 DEFGRADNEW(NBLOCK,NDIR+NSHR+NSHR),
     9 FIELDNEW(NBLOCK,NFIELDV),
     1 STRESSNEW(NBLOCK,NDIR+NSHR),
     STATENEW(NBLOCK,NSTATEV),
     2 ENERINTERNNEW(NBLOCK),
     ENERINELASNEW(NBLOCK)
C
      CHARACTER*80 CMNAME
C
      IF (CMNAME(1:5) .EQ. 'STEEL') THEN
      CALL VUMAT_STEEL(NBLOCK, NDIR, NSHR,
     1 NSTATEV, NPROPS, TOTALTIME,
     2 PROPS, STRAININC,
     3 STRESSOLD, STATEOLD,
     4 STRESSNEW, STATENEW)
      ELSE IF (CMNAME(1:7) .EQ. 'MASONRY') THEN
      CALL VUMAT_MASONRY(NBLOCK, NDIR, NSHR,
     1 NSTATEV, NPROPS, CHARLENGTH,
     2 PROPS, STRAININC,
     3 STRESSOLD, STATEOLD,
     4 STRESSNEW, STATENEW)
      ELSE IF (CMNAME(1:14) .EQ. 'GROUTEDMASONRY')
     THEN

      CALL VUMAT_MASONRY(NBLOCK, NDIR, NSHR,
     1 NSTATEV, NPROPS, CHARLENGTH,
     2 PROPS, STRAININC,
     3 STRESSOLD, STATEOLD,
     4 STRESSNEW, STATENEW)
      END IF

      RETURN
      END

      SUBROUTINE VUMAT_MASONRY(NBLOCK, NDIR,
     NSHR,
     1 NSTATEV, NPROPS, CHARLENGTH,
     2 PROPS, STRAININC,
     3 STRESSOLD, STATEOLD,
     4 STRESSNEW, STATENEW)
      DIMENSION PROPS(NPROPS),
     1 CHARLENGTH(NBLOCK),
     STRAININC(NBLOCK,NDIR+NSHR),
     2 STRESSOLD(NBLOCK,NDIR+NSHR),
     3 STATEOLD(NBLOCK,NSTATEV),
     4 STRESSNEW(NBLOCK,NDIR+NSHR),
     5 STATENEW(NBLOCK,NSTATEV)

      DOUBLE PRECISION YOUNG(2), D(4,4),
     USRVAL(14), STATEV(2),
     &          STRESS(4), TEMP(4,4), DSTRAIN(4)

      USRVAL(1)=PROPS(1)
      USRVAL(2)=PROPS(2)
      USRVAL(3)=PROPS(3)
      USRVAL(4)=PROPS(4)
      USRVAL(5)=PROPS(5)
      USRVAL(6)=PROPS(6)
      USRVAL(7)=PROPS(7)
      USRVAL(8)=PROPS(8)
      USRVAL(9)=PROPS(9)
      USRVAL(10)=PROPS(10)
      USRVAL(11)=PROPS(11)
      USRVAL(12)=PROPS(12)
      USRVAL(13)=PROPS(13)
      USRVAL(14)=PROPS(14)


      YOUNG(1)=PROPS(15)
      YOUNG(2)=PROPS(16)

      E11=PROPS(15)
      E22=PROPS(16)
      E33=PROPS(17)
      V12=PROPS(18)
      V23=PROPS(19)
      V13=PROPS(20)
      GXY=PROPS(21)
      V21=V12/E11*E22
      V31=V13/E11*E33
      V32=V23/E22*E33
      DET=(1-V12*V21-V23*V32-V31*V13-
     2*V21*V32*V13)/E11/E22/E33
      D(1,1)=(1-V23*V32)/E22/E33/DET
      D(1,2)=(V12+V32*V13)/E11/E33/DET
      D(2,1)=D(1,2)
      D(1,3)=(V13+V12*V23)/E11/E22/DET
      D(3,1)=D(1,3)
      D(2,2)=(1-V13*V31)/E11/E33/DET
      D(2,3)=(V23+V21*V13)/E11/E22/DET
      D(3,2)=D(2,3)
      D(3,3)=(1-V12*V21)/E11/E22/DET
      D(4,4)=GXY
      D(4,1)=0.0D0
      D(4,2)=0.0D0
      D(4,3)=0.0D0
      D(1,4)=0.0D0
      D(2,4)=0.0D0
      D(3,4)=0.0D0

      DO 100 KM = 1,NBLOCK

      STRESS(1)=STRESSOLD(KM,1)
      STRESS(2)=STRESSOLD(KM,2)
      STRESS(3)=STRESSOLD(KM,3)
      STRESS(4)=STRESSOLD(KM,4)
```

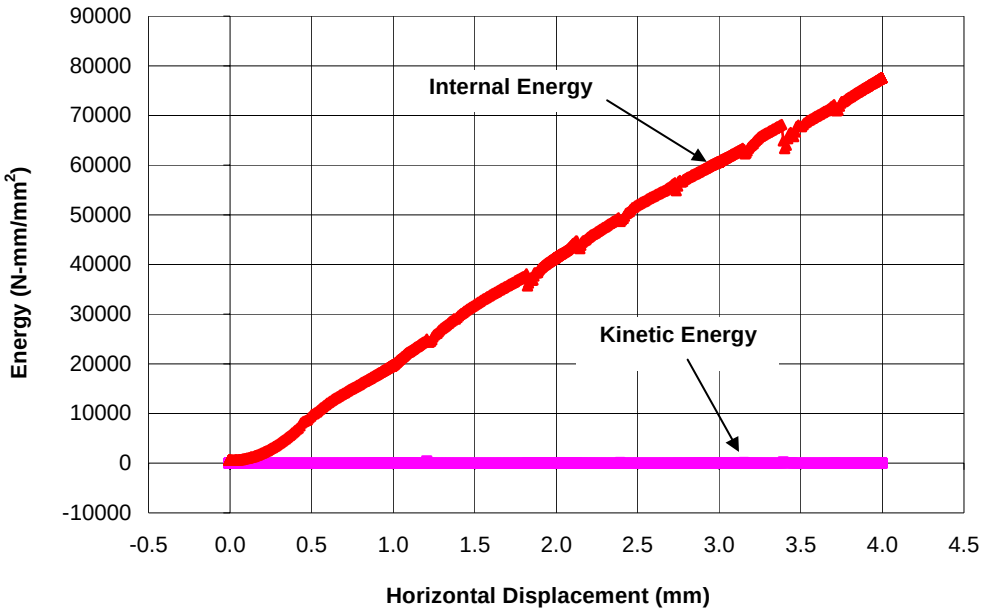
```
      USRVAL(14)=CHARLENGTH(KM)

      STATEV(1)=STATEOLD(KM,1)
   STATEV(2)=STATEOLD(KM,2)

      DSTRAIN(1)=STRAININC(KM,1)
      DSTRAIN(2)=STRAININC(KM,2)
      DSTRAIN(3)=STRAININC(KM,3)
      DSTRAIN(4)=STRAININC(KM,4)

      ITER=0

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   CALL USRMAT( DSTRAIN, 4, 0, 0, D, ITER,
      USRVAL,14,
  $         STATEV, 2, STRESS, TEMP, YOUNG )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCC

   STRESSNEW(KM,1)=STRESS(1)
   STRESSNEW(KM,2)=STRESS(2)
   STRESSNEW(KM,3)=STRESS(3)
      STRESSNEW(KM,4)=STRESS(4)

      STATENEW(KM,1)=STATEV(1)
   STATENEW(KM,2)=STATEV(2)

 100 CONTINUE

   RETURN
   END

   SUBROUTINE VUMAT_STEEL(NBLOCK, NDIR, NSHR,
  1 NSTATEV, NPROPS, TOTALTIME,
  2 PROPS,  STRAININC,
  3 STRESSOLD, STATEOLD,
  4 STRESSNEW, STATENEW)
   DIMENSION PROPS(NPROPS),
  1 CHARLENGTH(NBLOCK),
      STRAININC(NBLOCK,NDIR+NSHR),
  2 STRESSOLD(NBLOCK,NDIR+NSHR),
  3 STATEOLD(NBLOCK,NSTATEV),
  4 STRESSNEW(NBLOCK,NDIR+NSHR),
  5 STATENEW(NBLOCK,NSTATEV)

   DOUBLE PRECISION STATEV,  STRESS1,
      DSTRAIN
      DOUBLE PRECISION  E, FTY, FTU,FTF, FCU,
      KTY, KTU, KTF, A
      E  =PROPS(1)
      A  =PROPS(2)
      FTY=PROPS(3)
      FTU=PROPS(4)
      FTF=PROPS(5)
      FCU=PROPS(6)
      KTY=PROPS(7)
      KTU=PROPS(8)
      KTF=PROPS(9)

      DO 1101 KM = 1,NBLOCK

   STRESS1=STRESSOLD(KM,1)
      DSTRAIN=STRAININC(KM,1)
      STATEV=STATEOLD(KM,1)

         IF (TOTALTIME.LE.1.0D-12) THEN

   STRESS1=STRESS1+E*DSTRAIN

      ELSE

      STATEV=STATEV+DSTRAIN
      STRESS1=STRESS1+E*DSTRAIN
      IF (STRESS1<FCU) THEN
        STRESS1=FCU

      ELSEIF (STATEV> KTY) THEN

         IF (STATEV<= KTU) THEN
          STRESS1=FTY+(FTU-FTY)/(KTU-
      KTY)*(STATEV-KTY)
         ELSE
          STRESS1=FTU+(FTF-FTU)/(KTF-
      KTU)*(STATEV-KTU)
           ENDIF

      ENDIF

      ENDIF

   STRESSNEW(KM,1)=STRESS1
      STATENEW(KM,1)=STATEV
C    WRITE(*,*) STRESS1, TOTALTIME, STATEV

1101  CONTINUE

   RETURN
   END
```

APPENDIX E

Energy Plots for Validation Test Walls

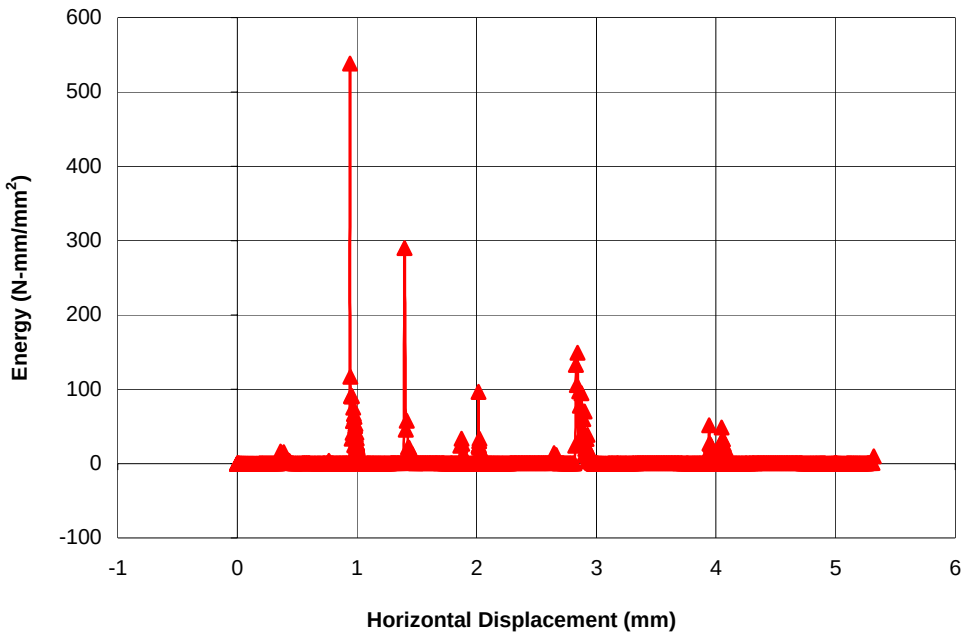


(a) Kinetic Energy

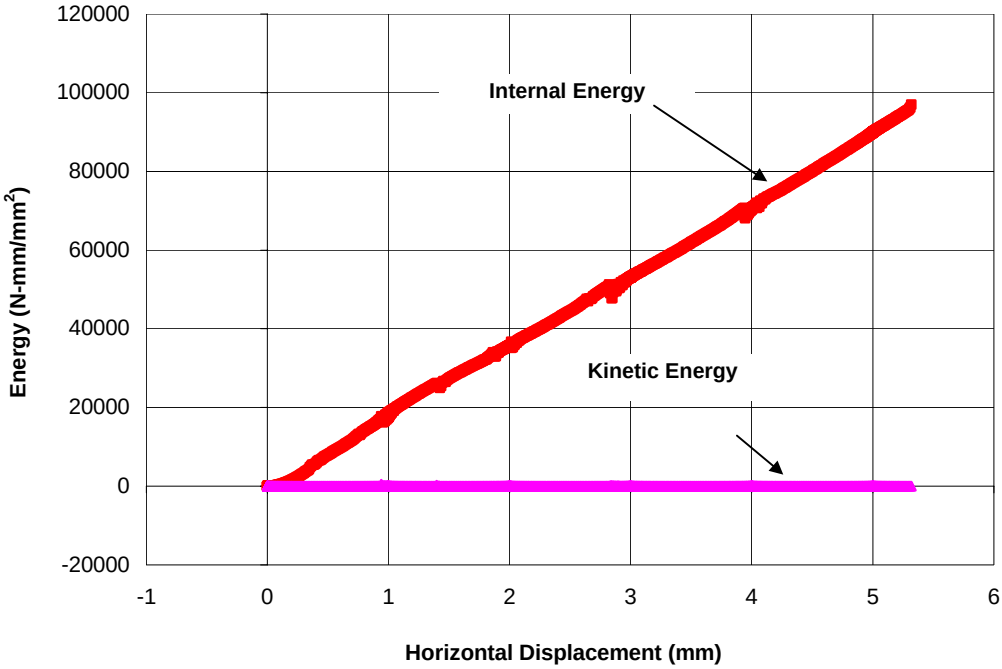


(b) Internal and Kinetic energies

Figure E 1: Energies for wall #11

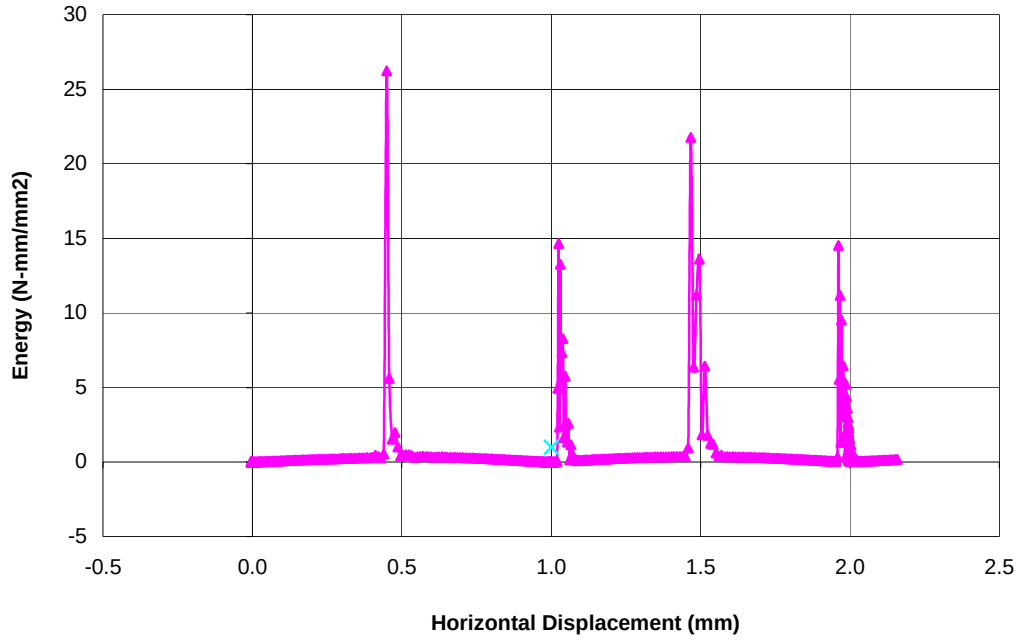


(a) Kinetic Energy

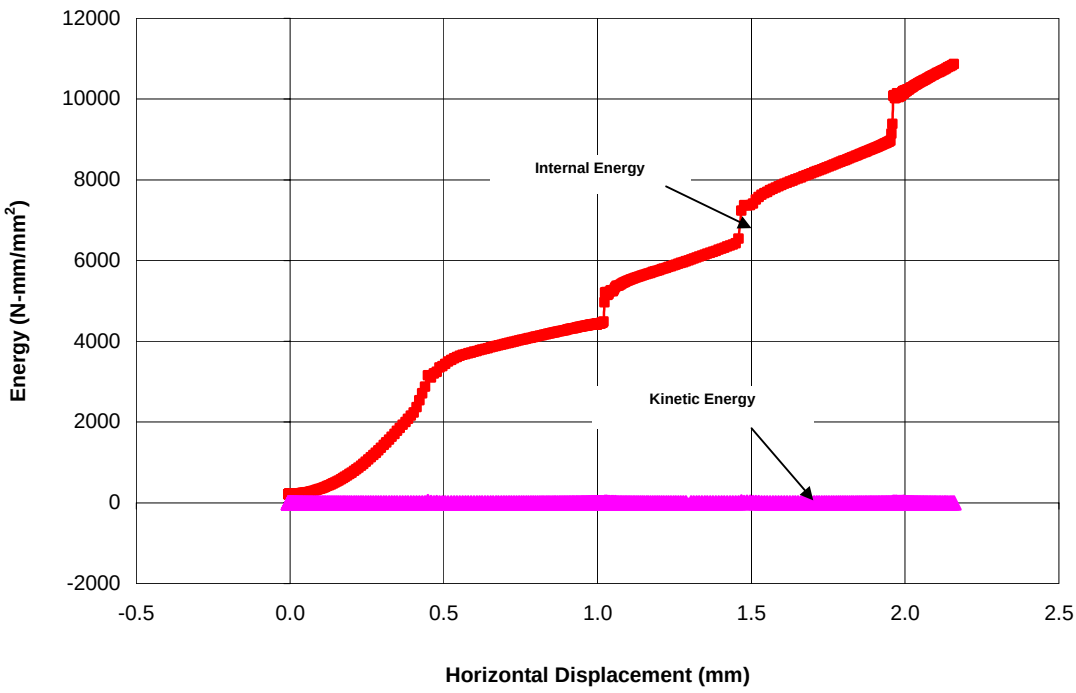


(b) Internal and Kinetic energies

Figure E 2: Energies for wall #12

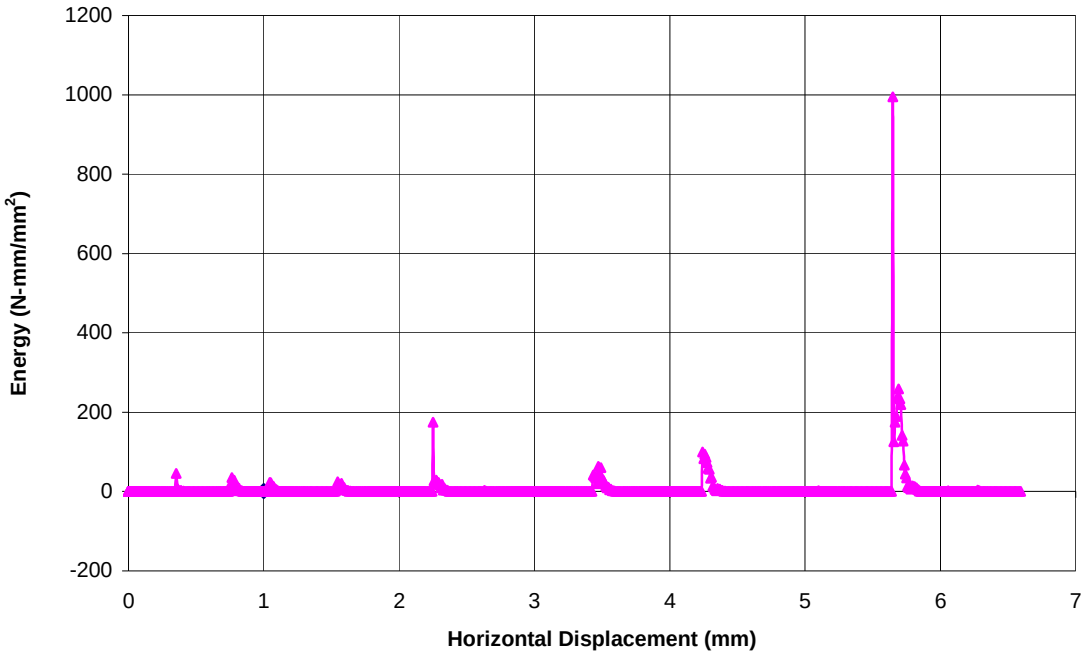


(a) Kinetic Energy

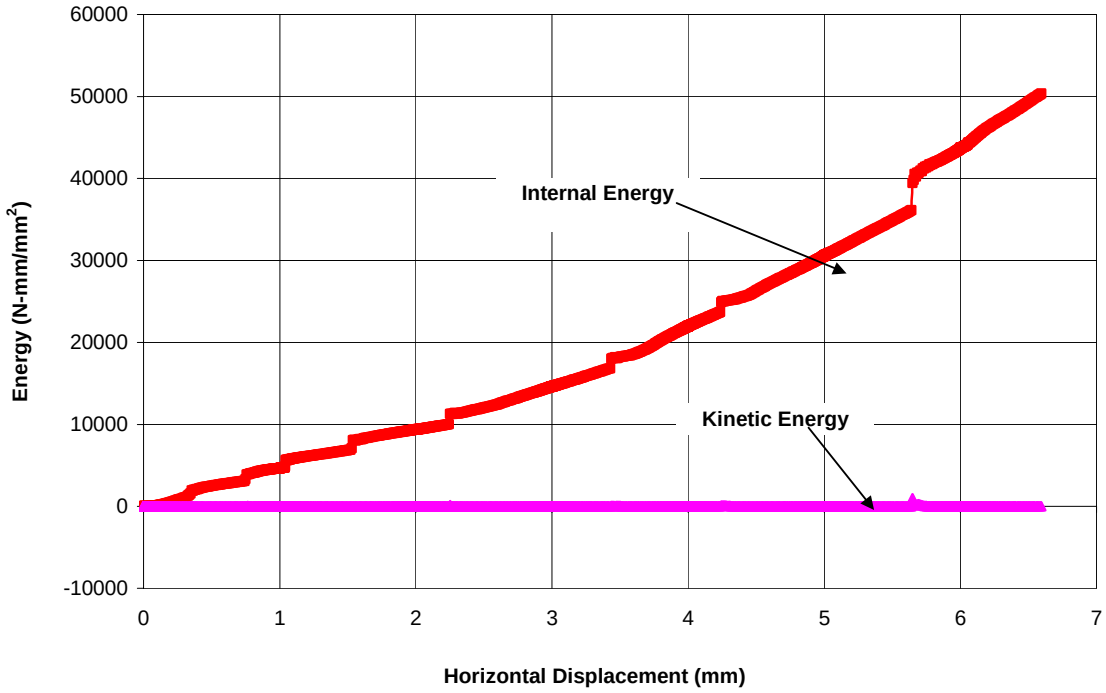


(b) Internal and Kinetic energies

Figure E 3: Energies for wall #13



(a) Kinetic Energy



(b) Internal and Kinetic energies

Figure E 4: Energies for wall #14