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:

\[ \sigma = D_0^\text{pl} \left( \varepsilon - \varepsilon^\text{pl} \right) \]  \hspace{1cm} (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 (\( p \)) and the Mises equivalent effective stress (\( q \)) which are calculated using effective stress deviator (\( \tilde{S} \)) as shown in Eq. A2 and Eq. A3 respectively.

\[ p = \frac{1}{3} \text{trace} (\dot{\sigma}) \]  \hspace{1cm} (A2)

\[ q = \sqrt{\frac{3}{2} (\tilde{S} : \tilde{S})} \]  \hspace{1cm} (A3)

where

\[ \tilde{S} = \dot{\sigma} + \dot{p} I \]  \hspace{1cm} (A4)

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

\[ G = \sqrt{(\varepsilon \sigma_{\text{ho}} \tan \psi)^2 + q^2 - p \tan \psi} \]  \hspace{1cm} (A5)

where \( \psi \) 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.

\[ \sigma_{\text{ho}} = \sigma_{\text{t}|_{\varepsilon^\text{pl} \rightarrow 0}} \]  \hspace{1cm} (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.

\[
\frac{1}{1-\alpha} \left( q - 3\alpha p + \beta \sigma \right) = \sigma_{\text{cr}}
\]

\[
\frac{1}{1-\alpha} \left( q - 3\alpha p + \beta \sigma \right) = \sigma_{\text{co}}
\]

*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:

\[
F = \frac{1}{1-\alpha} \left( q - 3\alpha p + \beta (\varepsilon) \left( \sigma_{\text{max}} \right) - \gamma \left( -\sigma_{\text{max}} \right) \right) - \sigma_{\varepsilon} (\varepsilon_{\varepsilon}) = 0 \quad (A7)
\]

\[
\alpha = \frac{(\sigma_{\text{bo}}/\sigma_{\text{co}}) - 1}{2(\sigma_{\text{bo}}/\sigma_{\text{co}}) - 1}, \quad 0 \leq \alpha \leq 0.5 \quad (A8)
\]

\[
\beta = \frac{\sigma_{\varepsilon} (\varepsilon_{\varepsilon})}{\sigma_{\varepsilon} (\sigma_{\varepsilon})} (1 - \alpha) \quad (A9)
\]
\( \gamma = \frac{3(1-K_c)}{2K_c - 1} \) \hspace{1cm} (A10)

\( \sigma_{\text{max}} \) is the maximum principal effective stress, \( \sigma_{\text{bo}}/\sigma_{\text{co}} \) 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. \( \sigma_{t}^{\text{-pl}} \) and \( \sigma_{c}^{\text{-pl}} \) 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
SUBROUTINE VUMAT(C
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,
ENERELASOLD,
6  TEMPNW, STRETCHNEW, DEFGRADNEW, FIELDNEW,
C WRITE ONLY (MODIFIABLE) VARIABLES -
7  STRESSNEW, STATENEW, ENERINTERNNEW,
ENERELASNEW )
C
INCLUDE 'VABA_PARAM.INC'
C
DIMENSION PROPS(NPROPS), DENSITY(NBLOCK),
1  COORDMP(NBLOCK,*),
2  CHARLENGTH(NBLOCK),
3  STRAININC(NBLOCK,NDIR+NSHR),
4  RELSPININC(NBLOCK,NSHR),
5  TEMPOLD(NBLOCK,NDIR+NSHR),
6  STRETCHOLD(NBLOCK,NDIR+NSHR),
7  DEFGRADOLD(NBLOCK,NDIR+NSHR+NSHR),
8  FIELDOLD(NBLOCK,NFIELDV),
9  STRESSOLD(NBLOCK,NDIR+NSHR),
10  STATEOLD(NBLOCK,NSTATEV),
11  ENERINTERNOLD(NBLOCK),
12  ENERELASOLD(NBLOCK),
13  TEMPNW(NBLOCK,NDIR+NSHR),
14  STRETCHNEW(NBLOCK,NDIR+NSHR+NSHR),
15  DEFGRADNEW(NBLOCK,NDIR+NSHR+NSHR+NSHR),
16  FIELDNEW(NBLOCK,NFIELDV),
17  STRESSNEW(NBLOCK,NDIR+NSHR),
18  STATENEW(NBLOCK,NSTATEV),
19  ENERINTERNNEW(NBLOCK),
C
CHARACTER*80 CMNAME
C
DOUBLE PRECISION YOUNG(2), D(4,4),
USRVAL(14), STATEV(2),
& STRESS(4), TEMP(4,4), DSRRAIN(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)
11=PROPS(15)
E22=PROPS(16)
E33=PROPS(17)

RETURN
END
VUMAT For Reinforced Masonry

SUBROUTINE VUMAT(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, STRETCHOOLD, DEFGRADOOLD, FIELDOLD,
5 STRESSOLD, STATEOLD, ENERINTERNOLD,
ENERINELASOLD,
6 TEMPNW, STRETCHNEW, DEFGRADNEW,
FIELDNEW,
C WRITE ONLY (MODIFIABLE) VARIABLES -
7 STRESSNEW, STATENEW, ENERINTERNNEW,
ENERINELASNEW)
C
INCLUDE 'VABA_PARAM.INC'
C
DIMENSION PROPS(NPROPS), DENSITY(NBLOCK),
1 CHARLENGTH(NBLOCK),
2 STRAININC(NBLOCK,NDIR+NSHR),
3 RELSPININC(NBLOCK,NSHR), TEMPOLD(NBLOCK),
4 STRETCHOOLD(NBLOCK,NSHR), DEFGRADOOLD(NBLOCK,NSHR),
5 FIELDOLD(NBLOCK,NFIELDV),
6 STRESSOLD(NBLOCK,NDIR+NSHR), STATEOLD(NBLOCK,NSTATEV),
7 ENERINTERNOLD(NBLOCK),
8 STRETCHNEW(NBLOCK,NDIR+NSHR),
9 DEFGRADNEW(NBLOCK,NDIR+NSHR),
10 STRESSNEW(NBLOCK,NDIR+NSHR), STATENEW(NBLOCK,NSTATEV),
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,
1 NSHR, NSTATEV, NPROPS, CHARLENGTH,
2 PROPS, STRAININC,
3 STRESSOLD, STATEOLD,
4 STRESSNEW, STATENEW)
DOUBLE PRECISION YOUNG(2), D(4,4),
USRVAL(14), STATE(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)
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)
END IF
RETURN
END
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

DO 1101 KM = 1,NBLOCK

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

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),
2  STRAININC(NBLOCK,NDIR+NSHR),
3  STATEOLD(NBLOCK,NSTATEV),
4  STRESSNEW(NBLOCK,NDIR+NSHR),
5  STATENEW(NBLOCK,NSTATEV)

DOUBLE PRECISION STATEV, STRESS,
DSTRAIN
DOUBLE PRECISION E, FTY, FTU, FTCU, FCCU,
KTY, KTCU, KTF, A
E =PROPS(1)
A =PROPS(2)
FTY=PROPS(3)
FTU=PROPS(4)
FTC=PROPS(5)
FCCU=PROPS(6)
KTY=PROPS(7)
KTC=PROPS(8)
KTF=PROPS(9)

DO 1101 KM = 1,NBLOCK

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

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),
2  STRAININC(NBLOCK,NDIR+NSHR),
3  STATEOLD(NBLOCK,NSTATEV),
4  STRESSNEW(NBLOCK,NDIR+NSHR),
5  STATENEW(NBLOCK,NSTATEV)

DOUBLE PRECISION STATEV, STRESS,
DSTRAIN
DOUBLE PRECISION E, FTY, FTU, FTCU, FCCU,
KTY, KTCU, KTF, A
E =PROPS(1)
A =PROPS(2)
FTY=PROPS(3)
FTU=PROPS(4)
FTC=PROPS(5)
FCCU=PROPS(6)
KTY=PROPS(7)
KTC=PROPS(8)
KTF=PROPS(9)

DO 1101 KM = 1,NBLOCK

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

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
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
Figure E 3: Energies for wall #13

(a) Kinetic Energy

(b) Internal and Kinetic energies
Appendix

(a) Kinetic Energy

(b) Internal and Kinetic energies

Figure E 4: Energies for wall #14