

## Cable Modeling in OpenDSS

To model a generic  $n$ -conductor line in OpenDSS, as well as any other software, requires the corresponding impedance matrix  $Z^1$  (or the sequence impedances  $Z_1$  and  $Z_0$ ) as shown in Eq. 1. This is the classic approach adopted to model a line. In this case OpenDSS uses the Linecode class (see the User Manual and OpenDSS help for further details).

$$\text{Eq. 1} \quad Z = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1n} \\ \dots & Z_{22} & \dots & Z_{2n} \\ \dots & \dots & \dots & \dots \\ Z_{n1} & Z_{n2} & \dots & Z_{nn} \end{bmatrix} \quad \begin{array}{l} \text{Primitive Matrix or} \\ \text{Reduced Matrix} \end{array}$$

$$\begin{array}{l} Z_1 = R_1 + jX_1 \\ Z_0 = R_0 + jX_0 \end{array} \quad \begin{array}{l} \text{Sequence impedance} \end{array}$$

However, in case the impedance data are not directly available, OpenDSS is able to produce the impedance matrix from the physical features of cables and line if provided. For this purpose, two following data are needed:

- 1) The physical characteristics of the conductors (or cables) that constitute the line. Three classes can be adopted to input these data in OpenDSS: CNData – TSData (for cables) and WireData (for bare conductors)
- 2) Cable-conductors disposition. For this purpose, the LineGeometry class is used in OpenDSS.

This technote will focus on the 2 classes adopted to describe concentric-neutral and tape shielded cables (i.e., CNData and TSData ). WireData and LineGeometry have already been described in the User Manual.

### CNData

This class is adopted to describe concentric neutral cables. This cable owns his name by the neutral conductors that encircle the phase conductors as shown in Figure 1.

To describe a concentric neutral cable the CNData class in OpenDSS adopts the following properties:

<b>DiaCable</b>	= Diameter over cable; same units as radius; no default
<b>DiaIns</b>	= Diameter over insulation layer; same units as radius; no default. Establishes outer radius for capacitance calculation
<b>diam</b>	= Diameter of phase conductors; Alternative method for entering radius
<b>DiaStrand</b>	= Diameter of a concentric neutral strand; same units as core conductor radius; no default
<b>Emergamps</b>	= Emergency ampacity, amperes. Defaults to 1.5 * Normal Amps if not specified
<b>EpsR</b>	= Insulation layer relative permittivity; default is 2.3
<b>GMRac</b>	= Geometric Mean Radius (GMR) at 60 Hz. Defaults to .7788*radius if not specified.

<sup>1</sup> In reality the algorithm that OpenDSS adopts is based on admittance matrices (i.e.,  $Y_{prim}$ , see the “User Manual”). However, due to the general familiarity of the audience with impedance matrixes and the capability of OpenDSS in accepting them as input, only the impedance matrixes will be mentioned hereafter

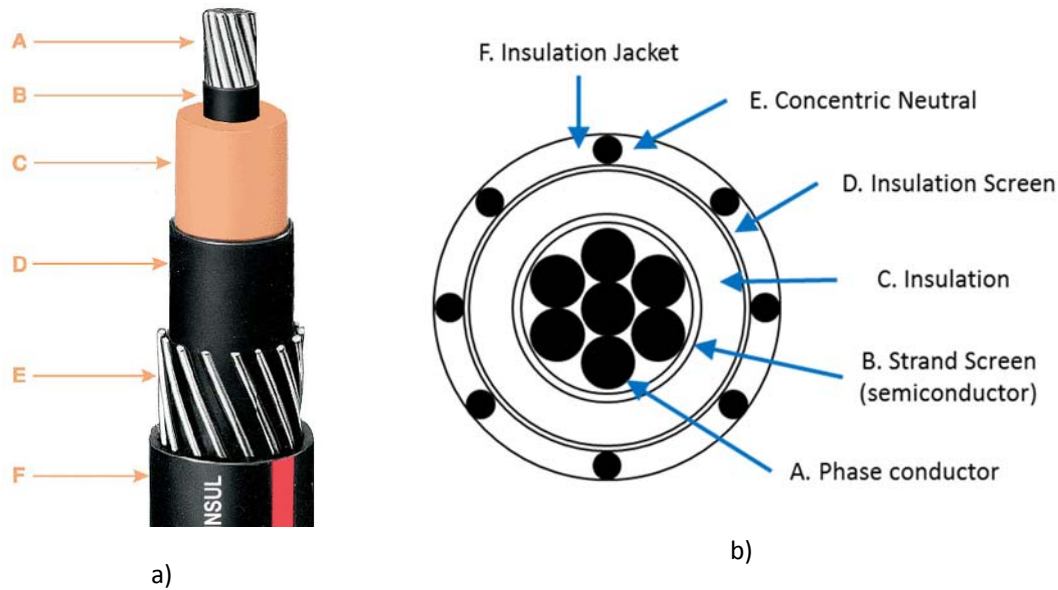


Figure 1 Structure of a Concentric Neutral Cable: a) Front [1] b) simplified cross section (not in proportion)

- GmrStrand** = Geometric mean radius (GMR) of a concentric neutral strand; same units as core conductor GMR; defaults to  $0.7788 \times \text{CN strand radius}$
- GMRunits** = Units for Geometric mean radius (GMR): {mi|kft|km|m|Ft|in|cm|mm} Default=none
- InsLayer** = Insulation layer thickness; same units as radius; no default. With Dialns, establishes inner radius for capacitance calculation
- k** = Number of concentric neutral strands; default is 2
- like** = Make like another object, e.g.: New Capacitor.C2 like=c1 ...
- normamps** = Normal ampacity, amperes. Default equal to Emergency amps/1.5 if not specified
- Rac** = Phase Conductor Resistance at 60 Hz per unit length. Defaults to  $1.02 \times \text{Rdc}$  if not specified

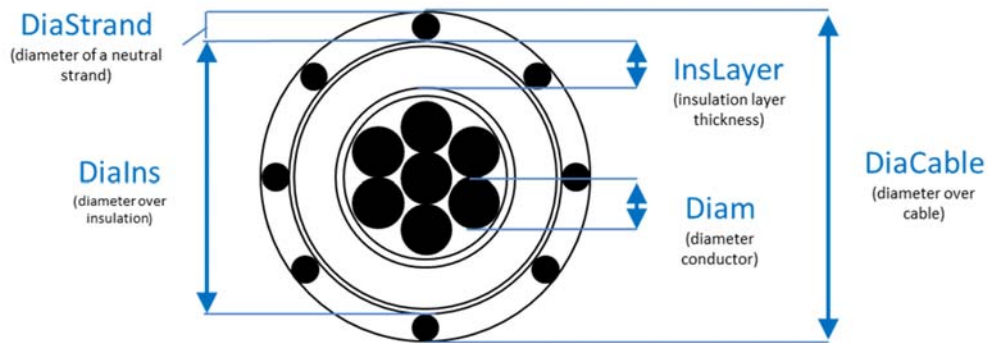


Figure 2 Main geometric properties in CNData class for a concentric neutral cable

- Radius** = Outside radius of conductor. Defaults to  $\text{GMR}/0.7788$  if not specified.
- Radunits** = Units for outside radius: {mi|kft|km|m|Ft|in|cm|mm} Default=none.
- Rdc** = dc Resistance, ohms per unit length (see Runits). Defaults to  $\text{Rac}/1.02$  if not specified
- Rstrand** = AC resistance of a concentric neutral strand ; same units as core conductor resistance ; no default.defined. Ignored for symmetrical components

**Runits** = Length units for resistance: ohms per {mi|kft|km|m|Ft|in|cm|mm} Default=*none*

Some of the previous geometric properties are also shown in Figure 2 in a cross section of a cable.

## TSDATA

This class is adopted to describe the physical features of tape shielded cable as the one shown in Figure 3. The following properties can be defined in the TSData class:

**DiaCable** = Diameter over cable; same units as radius; no default

**Dialns** = Diameter over insulation layer; same units as radius; no default. Establishes outer radius for capacitance calculation

**diam** = Diameter of phase conductors; Alternative method for entering radius

**DiaShield** = Diameter over tape shield; same units as radius; no default.

**Emergamps** = Emergency ampacity, amperes. Defaults to 1.5 \* Normal Amps if not specified

**EpsR** = Insulation layer relative permittivity; default is 2.3

**GMRac** = Geometric Mean Radius (GMR) at 60 Hz. Defaults to .7788\*radius if not specified.  
Zero-Sequence capacitance, nanofarads per unit length

**GMRunits** = Units for Geometric mean radius (GMR): {mi|kft|km|m|Ft|in|cm|mm} Default=*none*

**InsLayer** = Insulation layer thickness; same units as radius; no default. With Dialns, establishes inner radius for capacitance calculation

**like** = Make like another object, e.g.: New Capacitor.C2 like=c1 ...

**normamps** = Normal ampacity, amperes. Default equal to Emergency amps/1.5 if not specified

**Rac** = Phase Conductor Resistance at 60 Hz per unit length. Defaults to 1.02\*Rdc if not specified

**Radius** = Outside radius of conductor. Defaults to GMR/0.7788 if not specified.

**Radunits** = Units for outside radius: {mi|kft|km|m|Ft|in|cm|mm} Default=*none*.

**Rdc** = dc Resistance, ohms per unit length (see Runits). Defaults to Rac/1.02 if not specified

**Runits** = Length units for resistance: ohms per {mi|kft|km|m|Ft|in|cm|mm} Default=*none*

**TapLap** = Tape Lap in percent; default 20.0

**TapLayer** = Tape shield thickness; same units as radius; no default.

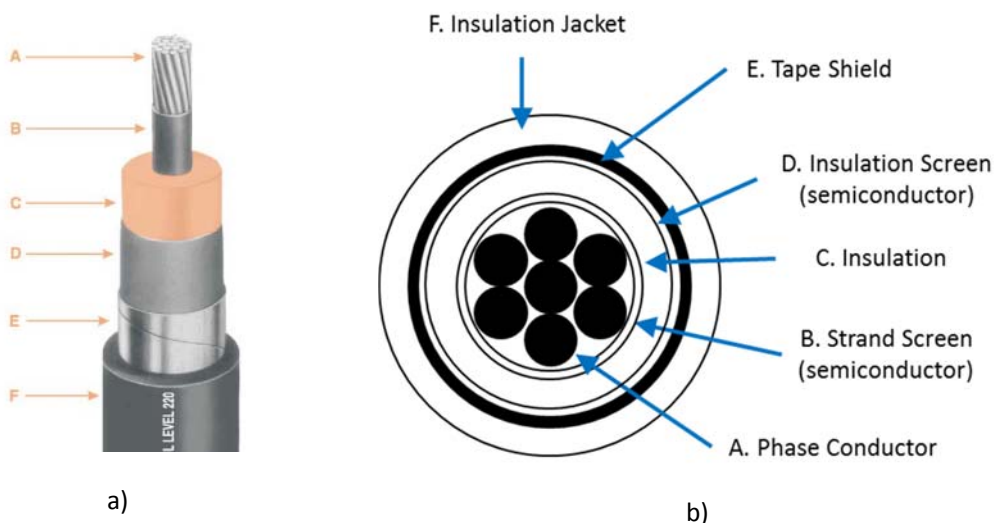


Figure 3 Structure of a Tape Shielded cable: a) Front [2] b) simplified cross section (not in proportion)

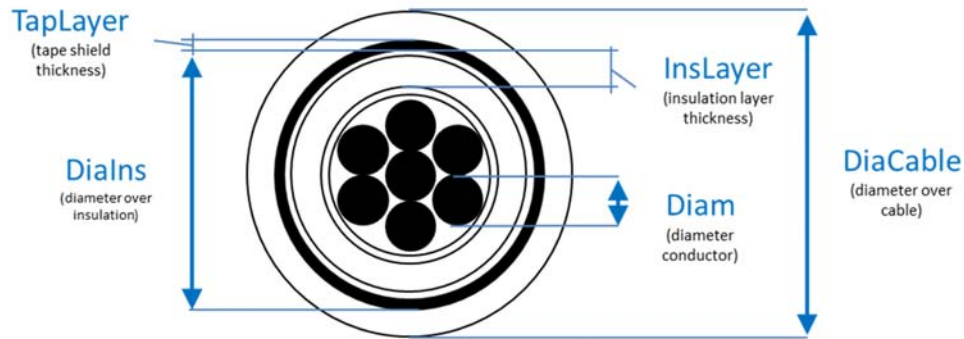


Figure 4 Main geometric properties in TSData class for a tape shielded cable

## PRACTICAL EXAMPLE: CONCENTRIC NEUTRAL CABLE

This section shows how to retrieve the data to model a concentric neutral cable. The same process can be repeated almost unchanged for tape shielded ones.

Let's assume that the following cable needs to be modeled into OpenDSS.

*Concentric neutral – Size=250 - Material=Aluminum – Neutral Size= 1/3*

The geometric and electric properties required by the CNData class can be found in any manufacturer datasheet as shown in Figure 5 and summarized as follows:

*DiaIns=1.16 in - DiaCable=1.29 in - k=13 - Insulation layer material: EPR (epsR=2.3) - InsLayer=0.22 in*

For the missing other resources such as [4] can be adopted. More precisely, for the phase conductors (250 AA with 19 strands) the following data were found (Appendix A in [4]):

Catalog Number	Conductor Size AWG/kcmil	Nominal Dia. over Insulation (in.)	Nominal Dia. over Insulation Screen (in.)	Copper Neutral, No. x AWG (1)	Nominal O.D. (in.)	Aprox. Net Weight lbs./1000'	Aprox. Ship Weight lbs./1000'	90°C Ampacity Direct Burial (2)	90°C Ampacity Duct (2)	105°C Ampacity Direct Burial (2)	105°C Ampacity Duct (2)
<b>1/3 NEUTRAL</b>											
160-23-3910	2(7x)	0.78	0.85	6 x 14	0.98	429	519	160	135	165	130
160-23-3916	1(19x)	0.81	0.89	6 x 14	1.02	464	554	180	155	190	150
160-23-3922	1/0(19x)	0.84	0.92	6 x 14	1.06	506	596	205	175	215	175
160-23-3925	2/0(19x)	0.90	0.97	7 x 14	1.10	568	658	235	200	245	195
160-23-3928	3/0(19x)	0.95	1.02	9 x 14	1.15	653	743	270	230	280	225
160-23-3931	4/0(19x)	0.99	1.06	11 x 14	1.21	752	852	300	240	315	255
160-23-3934	250(37x)	1.06	1.16	13 x 14	1.29	873	973	330	260	345	280

insulation 220 mil (0.22 inch) – Insulation material= EPR

Figure 5 Features of a Concentric neutral cable (without jacket) – 250 Aluminum – 1/3 neutral, [3]

Phase Conductors: Diam=0.567 in – GMRac=0.0171 ft (or 0.205 in) – Rac=0.41  $\Omega$ /mi (or 0.0776  $\Omega$ /kft)

Regarding the stranded neutral (Size=14, No=13), always in Appendix A in [4], it was found:

Neutral: DiaStrand=0.064 in – Rstrand=14.87  $\Omega$ /mi (or 2.55  $\Omega$ /kft)  
GmrStrand=0.00208 ft (or 0.02496 in)

Consequently as all physical features are now defined the cable can be modeled into OpenDSS adopting the code shown in Figure 6.

```
// Concentric neutral (without jacket) - 250 Aluminum - 1/3 neutral  
New CNDATA.250_Al_15kV_1/3  
~ Runits=kft Radunits=in GMRunits=in //Units of measure  
~ InsLayer=0.220 DiaIns=1.16 DiaCable=1.29 epsR=2.3 !Cable  
~ Rac=0.0776 GMRac=0.205 diam=0.567 !Phase Conductor  
~ Rstrand=2.55 GMRstrand=0.02496 DiaStrand=0.064 k=13 !Neutral
```

Figure 6 OpenDSS code to model the concentric neutral – 250 Aluminum - 1/3 Neutral

## Line Modeling: Primitive Impedance Matrix, Reduced Matrix, and Sequence Impedances

This section describes, adopting examples in OpenDSS, the differences between primitive, (Kron) reduced and sequence impedance. These are the three different “forms” of the line impedance matrix that can be adapted to model. This tech note does not have the ambition to provide a lecturer on the subject (as plenty of excellent books for the purpose) but simply to highlight some basic concepts leveraging the capability of OpenDSS in producing these models.

Let’s assume to model a three phase + neutral overhead line given by 4 AAC (i.e., All Aluminum Conductor) of American size 477. This line is one of the several modeled in the EPRI test case ckt24<sup>2</sup>. The conductors features are describes below, their disposition is shown in Figure 7 and the corresponding code in OpenDSS is shown in Figure 8.

1. Overall outside diameter=0.793 in
2. GMR=0.3 in
3. Rac=0.1968 Ohm/mile

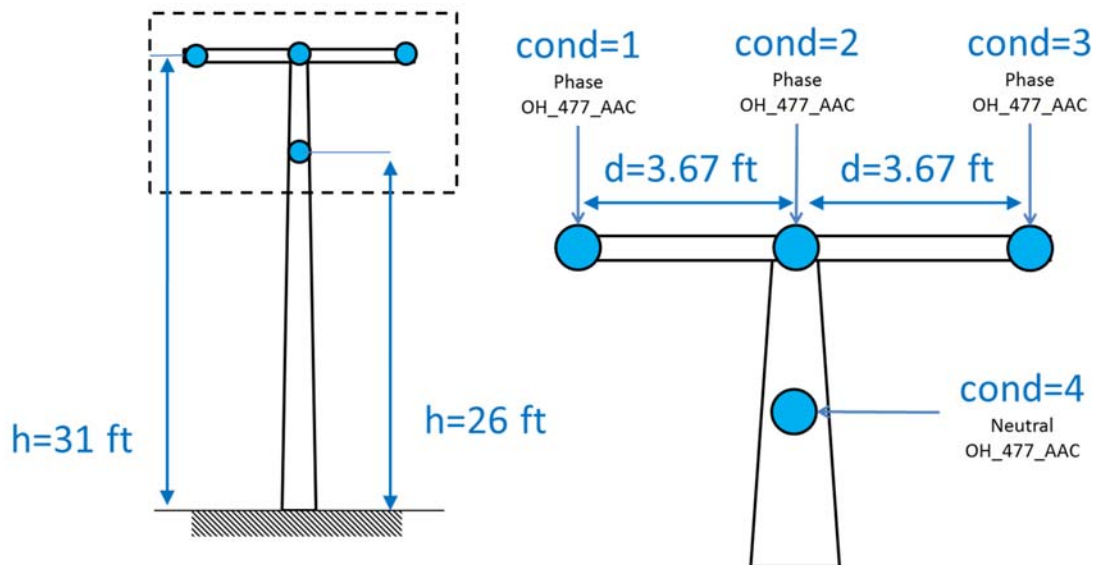


Figure 7 Modeled 3ph + neutral overhead line

```
Set DefaultBaseFrequency=60

clear

set earthmodel=Carson

New WireData.OH_477_AAC diam=0.793 radunits=in GMRac=0.3 GMRunits=in
Rac=0.1968 Runits=mi normamps=650

New LineGeometry.OH3P_FR8_N56_OH_477_AAC_OH_477_AAC_ABCN nconds=4 nphases=3
reduce=no
~ cond=1 wire =OH_477_AAC x=-3.67 h=31.33 units=ft
~ cond=2 wire =OH_477_AAC x=0 h=31.33 units=ft
~ cond=3 wire =OH_477_AAC x=3.67 h=31.33 units=ft
```

<sup>2</sup> Available also in the PC directory in which OpenDSS has been installed, in the EPRI test case folder.

```
~ cond=4 wire =OH_477_AAC x=0 h=26 units=ft

solve

// To show the impedance data calculated by OpenDSS
show lineconstant freq=60 units=mi
```

Figure 8 OpenDSS code to model the line depicted in Figure 7

The main commands adopted in Figure 8 are explained as follows:

- **earthmodel=Carson**: OpenDSS produces the line load model based on the simplified Carson approach that typically is adopted for steady state study at nominal frequency (50-60 Hz). The earth resistivity, “rho,” is generally specified with each Line definition and defaults to the commonly-assumed value of  $100 \text{ m} - \Omega$ . More details on the impact of such model (over the others) are discussed in [5].
- **WireData** and **LineGeometry**: The WireData class is adopted (see User manual and OpenDSS Help for more details) to describe the features of conductors while their placement on a pole or tower is detailed by the LineGeometry class. It is worth noticing that the WireData declaration should always be defined first. Otherwise OpenDSS cannot retrieve the conductor information referenced by the LineGeometry class.
- **nconds=4 nphases=3**: The line is defined by 4 physical conductors but only 3 phases. It means that the extra conductor (the **cond=4** in the LineGeometry declaration in Figure 8) is considered as the neutral by OpenDSS . There can be multiple neutral wires.
- **reduce=no**: This means that the line model developed by OpenDSS is NOT Kron reduced.
- **solve**: This command solves the circuit. As no loads, sourcebus or lines are present it will simply calculates the line impedances. The line impedances can also be computed without executing a Solve command by using the Show command described next.
- **show lineconstant freq=60 units=mi**: This command will produce a CSV files reporting the line model developed by OpenDSS with a frequency of 60Hz in  $\Omega/\text{ml}$ .

## PRIMITIVE MATRIX

The line model developed by OpenDSS (i.e., the primitive impedance matrix as the reduction has not been adopted) from the geometric and conductor features in Figure 8 is reported in a CSV file produced by the **show lineconstant** and reported for simplicity in Figure 9.

	Phase 1	Phase 2	Phase 3	Neutral	
$Z =$	0.292099+j1.41024				Phase1
	0.0952989+j0.804872	0.292099+j1.41024			Phase2
	0.0952989+j0.720767	0.0952989+j0.804872	0.292099+j1.41024		Phase3
	0.0952989+j0.73605	0.0952989+j0.759593	0.0952989+j0.73605	0.292099+j1.41024	Neutral

Figure 9 Primitive matrix developed by OpenDSS for the line shown in Figure 7 ( $\Omega/\text{ml}$ )

This matrix provides the mutual impedances (out of diagonal elements) as well as self-impedances (diagonal elements) of every conductor including the neutral. This is the most detailed modeling of a line as the effects of every conductor are explicitly reported. For instance, the mutual impedance between phase 2 and phase 1 (i.e.,  $Z_{21} = Z_{12} = 0.0952989 + j0.804872 \text{ } \Omega/\text{ml}$ ) and phase 3 and phase 1 (i.e.,  $Z_{31} = Z_{13} = 0.0952989 + j0.720767 \Omega/\text{ml}$ ) are different. More precisely, the imaginary part of  $Z_{21}$  is larger than the one in  $Z_{31}$ . This, in turn, implies that the influence of phase 2 on phase 1 is larger than the influence of phase 3. Indeed, as evident from Figure 7, phase 2 conductors are closest to phase 1.

## KRON REDUCED MATRIX

The Kron reduction is a standard method to mathematically embed the neutral effects in the phase conductors producing a matrix of lower dimension (i.e., of  $n_{phases} \times n_{phases}$  rather than  $n_{conds} \times n_{conds}$ ). The Kron reduction is achieved on the primitive  $Z$  matrix assuming that the voltage on both ends of the neutral conductor is zero a common assumption for multigrounded neutral systems.

OpenDSS can provide the (Kron) reduced version of the primitive matrix by simply replacing the `reduce=no` command with `reduce=yes`. The results so obtained, provided by the CSV report produced by OpenDSS, is shown in Figure 10.

	Phase 1	Phase 2	Phase 3	
$Z =$	0.271731+j1.02829			Phase1
	0.0758464+j0.410825	0.273639+j1.00372		Phase2
	0.0749315+j0.338818	0.0758464+j0.410825	0.271731+j1.02829	Phase3

Figure 10 Reduced matrix developed by OpenDSS for the line shown in Figure 7 ( $\Omega/\text{mi}$ )

It is worth highlighting that the Kron reduced matrix does not provide an approximation of the primitive matrix for the phase conductors (i.e., no information is lost). Indeed, it is possible to demonstrate that the reduced version (Figure 10) gives exactly the same results, for the phase conductors, of the primitive matrix (Figure 9) in a power flow analysis.

## SEQUENCE IMPEDANCE

Quite often a line is modeled by its corresponding homopolar, or zero-sequence, impedance ( $Z_0 = R_0 + jX_0$ ) and positive-sequence impedance ( $Z_1 = R_1 + jX_1$ )<sup>3</sup> generally known as sequence impedances. In a simplified manner the positive-sequence impedance of a line can be defined as the impedance “seen” by a positive sequence currents flowing on it. On the other hand, the homopolar impedance is the impedance seen by three homopolar (zero-sequence) currents flowing on the line. Given this simplified definition (for more details please refer to any power system book) few important key concepts can be highlighted:

- The positive-sequence impedance, as the sum of the set of positive-sequence currents is zero for definition, does not involve the return path (neutral and/or earth).
- For the homopolar impedance, the return path (neutral and or earth) plays a key role. Indeed, as the sum of a 3-phase set of homopolar currents is not zero, the resulting current has to return somewhere “seeing” the impedance of the return path as well.
- For the previous case the homopolar impedance is expected to be higher than the positive-sequence one (a reasonable assumption, in case no information is provided, is to assume  $Z_0 = 3 * Z_1$ ).
- There is no definition (at least not standard) on sequence impedances of anything other than three-phase lines. Indeed, a set of currents (either positive or homopolar) should flow on three conductors (plus the return for the homopolar). Hence, there is technically no sequence impedance definition for monophasic lines although some distribution system analysis tools have their own non-standard definition.

The sequence-impedance model involves, most of the time, simplification. Consequently, the results provided by a sequence based impedance model in a power flow analysis *are going to be different*

<sup>3</sup> In reality also the negative impedance ( $Z_2$ ) should be considered. However, for several elements (including the line but not some type of transformer) negative and positive are equal. Hence, as the focus in this technote is the modelling of lines, the negative impedance is neglected.



from those obtained by the corresponding unbalanced primitive matrix. To understand where the main differences may arise the key steps to produce the sequence impedance from the primitive matrix are summarized:

1. Produce the primitive matrix (from its geometry for instance)
2. Reduce the primitive matrix by Kron method. Indeed, the sequence impedances required a 3x3 matrix.
3. The reduced matrix is *forced to be* balanced. It means that all off diagonal elements (e.g.,  $Z_{21}$ ,  $Z_{31}$  and  $Z_{32}$ ) are averaged ( $Z_m$ ). The diagonal elements are also averaged ( $Z_s$ )
4. A conversion process is then adopted to produce, from the balanced 3x3, the sequence impedances.

$$Z_1 = Z_s - Z_m$$

$$Z_0 = Z_s + 2Z_m$$

Consequently, the third step is the most critical. Indeed, as it can be noticed in Figure 10, the off-diagonal elements of a Kron reduced matrix are not equal each other (e.g.,  $Z_{21} \neq Z_{23}$ ) as the unbalances of the primitive matrix (due to the asymmetrical geometry of the conductors) are directly reported also in the reduced one. Consequently, information is lost in this step. However, this assumption is of no impact if the Kron reduced matrix is almost balanced. This happens for symmetric disposition and/or transposed line. This is common in transmission, less in distribution.

The report produced issued by OpenDSS by the command `show lineconstant` provides also the corresponding sequence impedance that, for the specific example of Figure 7, are reported in Figure 11.

$Z_1$ , ohms per mi = 0.196826 + j 0.633275 (L1 = 1.67981 mH)

$Z_0$ , ohms per mi = 0.42345 + j1.79374 (L0 = 4.75805 mH)

**Figure 11 Sequence impedance produced by OpenDSS**

It can be noticed that in this example, for the reasons discussed previously,  $Z_0 > Z_1$  and that considering  $Z_0 \approx 3 Z_1$  is not a bad assumption.

## Further Resources

---

### REPOSITORY

In the repository other documents can found:

<https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/Doc/TechNotes/>

In addition the EPRI test circuits contains several cables-conductors modeling:

<https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/EPRI TestCircuits/>

### THE BEST OF THE FORUM

Here some of the best answers in the OpenDSS forum on line modeling issues that several users have encountered. These provide further insights and detailed information on how OpenDSS works.

#### Answer 1

It should be noted that the Kron reduction is NOT ASSUMED. You must tell OpenDSS specifically that you want it to reduce the neutral conductors (could be more than one) out if that is what you want.

The Kron reduction is done on the Z matrix (also on the inverse of the Yc matrix) so it is the **Voltage** on both ends of the conductor that is assumed to be zero, not the current.

I think one can show that the reduced model gives exactly the same answer for the phase currents as the full model. You just don't get the neutral currents explicitly. In fact Kersting defined cases with the IEEE 4-bus test system that showed precisely this fact, and you can prove this with OpenDSS.

The reduced 3x3 Z matrix will be unbalanced just as much as the original nxn, so how do you get Z1 and Z0? Obviously, you must make an assumption. Like many other analysts, OpenDSS averages the diagonals and off-diagonals (separately) of the reduced 3x3 matrix and then computes the approximate Z1 and Z0 values.

Will this give the same answer as the original unbalanced matrix? No. Whether or not this matters depends on the rest of the circuit. Bob Arritt and I wrote an IEEE conference paper 2 or 3 years ago that shows one example of the error in the solution that can be introduced. I can't recall the title right now, but you should be able to find it on IEEE Xplore.

#### Answer 2

By default, the LINE object model builds the primitive Y matrix assuming symmetrical component values (positive- and zero-sequence values). This is the most common form of input data we receive for distribution feeder data.

The program simply reconstructs a balanced 3x3 matrix based on the given values of positive- and zero-sequence impedances and capacitances or susceptances. You should be able to find this procedure defined in standard power system analysis text books. Of course, you can supply the impedance matrix values explicitly (Rmatrix, Xmatrix, and Cmatrix properties).

Link:

<https://sourceforge.net/p/electricdss/discussion/861977/thread/21bfcea3/?limit=25#efb0>

**Answer:**

There is no standard definition for sequence impedances of anything other than 3-phase lines. CYME, and other programs, define Z1 and Z0 for 1- and 2-phase lines. The basic idea is to produce the correct number in the fault current calculation. So  $(2 Z1 + Z0)/3$  would yield the single-phase impedance of a 1-phase line. In OpenDSS, you just put one number for a 1-phase line impedance. I'm going to guess that for a 2-phase line, Z1 is the line mode impedance based on the conductor spacing and Z0 is the impedance down both conductors and back through earth. In OpenDSS, you would ideally supply R, X, and C matrices just like you would for any other line using, for example, Carson's method.

You can supply Z1 and Z0 values for 1- and 2-phase lines. For a 1-phase line, specify the single-phase impedance in Z1. Z0 will default to the same value. For a two-phase line, set Z1 and Z0 the same as CYME. OpenDSS will construct a 2x2 Z matrix that should give the same answer. If not, let me know. I know we did figure that out one time.

**Link:**

<https://sourceforge.net/p/electricdss/discussion/861976/thread/51c41f39/>

**Answer:**

Since the 13-bus test feeder is quite popular, this post should be of interest:

A recent query on this forum about how to use CNDATA and CNCable led to the discovery that there is a discrepancy between the published IEEE 13-bus test feeder cable capacitance for Line Code 606 and what OpenDSS computes for the same cable.

Here are the original 606 matrices in OpenDSS format. You may have to use these to better match Kersting's results, although the cable is only 500 ft long and the differences may be down in the noise:

```
New linecode.mtx606 nphases=3 BaseFreq=60
~ rmatrix = (0.7982 | 0.3192 0.7891 | 0.2849 0.3192 0.7982 )
~ xmatrix = (0.4463 | 0.0328 0.4041 | -0.0143 0.0328 0.4463 )
~ Cmatrix = [257 | 0 257 | 0 0 257] ! <--- This is too low by 1.5
~ units=mi
```

If you let OpenDSS compute the 606 linecode using the following CNdata definition and LineGeometry definition

```
New CNDATA.250_1/3 k=13 DiaStrand=0.064 Rstrand=2.816666667 epsR=2.3
~ InsLayer=0.220 DiaIns=1.06 DiaCable=1.16 Rac=0.076705 GMRac=0.20568
diam=0.573
~ Runits=kft Radunits=in GMRunits=in
```

```
New LineGeometry.606 nconds=3 nphases=3 units=ft
~ cond=1 CNCable=250_1/3 x=-0.5 h= -4
~ cond=2 CNCable=250_1/3 x=0 h= -4
~ cond=3 CNCable=250_1/3 x=0.5 h= -4
```

... you will get:

```
New Linecode.mtx606 nphases=3 Units=mi
~ Rmatrix=[0.791721 | 0.318476 0.781649 | 0.28345 0.318476 0.791721
]
~ Xmatrix=[0.438352 | 0.0276838 0.396697 | -0.0184204 0.0276838 0.438352
]
```

```
~ Cmatrix=[383.948 | 0 383.948 | 0 0 383.948 ]
```

The Rmatrix and Xmatrix values are in substantial agreement. However, the Cmatrix values are about 1.5 times larger. This was a bit disconcerting when it was discovered. In another circuit this could make a quite significant difference.

I spent a couple of hours carefully stepping through the code and checking the OpenDSS capacitance calculations and I feel comfortable that OpenDSS is making the the correct computations.

I am not 100% sure where the original capacitance values came from and how they were computed but here is what I am guessing:

See the attached jpg file I captured from the Okonite website

[http://okonite.com/Product\\_Catalog/section2/section2-pdfs/2-39.pdf](http://okonite.com/Product_Catalog/section2/section2-pdfs/2-39.pdf).

This is the type of cable we are dealing with in the test case. The usual source of discrepancies in computing the capacitance of this kind of cable is due to ignoring the semiconducting layers around the core conductor (B) and over the insulation (D). These layers play a very important role in the durability of these cables by maintaining a uniform dielectric stress on the insulation. Thus, the main electric field is the field between these layers. So this in my opinion is the field that should be used for computing the capacitance.

So OpenDSS simply uses the cylindrical capacitance formula from the diameter over the insulation down to the diameter over the semicon layer over the core conductor. The algorithm computes the radius over the core conductor, which is not a data entry, by subtracting the thickness of the insulation (0.220 in. in this case) from the radius over the insulation. The value from this calculation agrees quite closely with the numerous actual measurements I have made.

Some analysts will use an equivalent diameter of the circle formed by the concentric neutral wires (E). This yields a smaller capacitance that does not agree with the measurements I have made. This diameter is larger by at least twice the thickness of the semicon layer. However, I didn't expect it to make a difference of a factor of almost 1.5.

I have updated the IEEE 13-bus test feeder files posted on Sourceforge. I left the original 606 matrices in the file -- commented out -- in case you want to get a better match to the published results. I doubt it will make much difference because the only LINE that uses this LINECODE is only 500 ft.

Link:

<https://sourceforge.net/p/electricdss/discussion/861976/thread/3161555d/>?

## REFERENCES:

- [1] Okonite. (2016). 15kV to 35kV Underground Primary Distribution Cable Jacketed — Red Identification Stripes. Available: [http://okonite.com/Product\\_Catalog/section2/section2-pdfs/2-42.pdf](http://okonite.com/Product_Catalog/section2/section2-pdfs/2-42.pdf)
- [2] Okonite. (2016). Okoguard® -Okoseal® Type MV-105 15kV Shielded Power Cable. Available: [http://okonite.com/Product\\_Catalog/section2/section2-pdfs/2-8.pdf](http://okonite.com/Product_Catalog/section2/section2-pdfs/2-8.pdf)
- [3] Okonite. (2016). 15kV Underground Primary Distribution Cable. Available: [http://okonite.com/Product\\_Catalog/section2/section2-pdfs/2-34.pdf](http://okonite.com/Product_Catalog/section2/section2-pdfs/2-34.pdf)
- [4] W. H. Kersting, Distribution system modeling and analysis. Boca Ratón: CRC Press, 2002.
- [5] R. Horton, W. G. Sunderman, R. F. Arritt and R. C. Dugan, "Effect of line modeling methods on neutral-to-earth voltage analysis of multi-grounded distribution feeders," Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES, Phoenix, AZ, 2011, pp. 1-6. doi: 10.1109/PSCE.2011.5772574