# LineCode Kron Reduction

From OpenDSSWiki

# Kron Reduction Modifications in LINECODE

## The Question:

We had autogenerated secondary services for some of the studies and had defined the Triplex service drop by a 3x3 matrix in a LINECODE. After some study we decided we weren't really learning anything by explicitly modeling the neutral and the grounding impedances. So we decided to reduce the neutral out of the triplex model and save 1/3 of the nodes used to model secondaries (about 3000 in one case).

The original LINECODE class had a Kron reduction algorithm from the beginning, but it only eliminated the last conductor. Unfortunately, the neutral was represented by the middle (no. 2) conductor in the R, X, and C matrices. So the LINECODE object was modified on 5 Nov 2008 to accommodate this.

## Solution: New Property Added

Neutral=(no. of neutral conductor)

The Neutral property designates which conductor is the "neutral" conductor that will be eliminated by the next Kron reduction operation. The default is the last conductor (value of the "nphases" property) when the LINECODE object is defined. After the Kron reduction is executed, this value is set to 0. Subsequent issuing of Kron=Yes will not do anything until this property is set to a legal value (0 < Neutral <= nphases).
The help is up to date on this property and the **Kron** property:

```
Kron = Y/N. Default=N. Perform Kron reduction on the impedance matrix after it is formed,
reducing order by 1. Eliminates the conductor designated by the "Neutral=" property.
Do this after the R, X, and C matrices are defined.
Ignored for symmetrical components. May be issued more than once to eliminate more
than one conductor by resetting the Neutral property.
```

The Kron reduction applies only to LINECODE objects defined by R, X, and C matrices. It makes no sense for symmetrical component definitions (R1=…, X1= ..., etc)

Example
Here is an example of how we reduced the neutral (conductor 2) out of the 3x3 Triplex model matrices:

```
New Linecode.1/0 nphases=3
~ rmatrix=[0.201761667 0.018049545 0.018049545 | 0.018049545 0.201761667 0.018049545 | 0.018049545
!ohms per 1000 ft
~ xmatrix=[ 0.285770602 0.254451335 0.246486562 | 0.254451335 0.285770602 0.254451335 | 0.246486562
~ cmatrix=[15 | -12 15 | -12 -12 15]
~ Normamps=202 {202 1.25 *}
~ Neutral=2 Kron=yes ! eliminate the neutral conductor (2)
```

The result is (from "dump linecode.1/0 debug")

```
New LineCode.1/0
~ nphases=2
~ r1=0.05800
~ x1=0.12060
~ r0=0.17840
~ x0=0.40470
~ c1=3.40000
~ c0=1.60000
~ units=none
~ rmatrix="0.28652 0.10281 |0.10281 0.28652 |"
~ xmatrix="0.12019 0.08091 |0.08091 0.12019 |"
~ cmatrix="15.00 -12.00 |-12.00 15.00 |"
~ baseFreq=60
~ normamps=202
~ emergamps=202 1.25 *
~ faultrate=0.1
~ pctperm=20
~ repair=3
~ Kron=yes
~ Rg=0
~ Xg=0
~ rho=100
~ neutral=0
```

Note the revised definitions of the matrices. They are now 2x2, the Nphases property is 2, and the Neutral property has been set to 0 to block further Kron reduction.

If we, for some reason, wanted to eliminate the resulting second conductor, we would simply repeat the last line:

```
~ Neutral=2 Kron=yes ! eliminate the neutral conductor (2)
```

This statement would reset the "neutral" conductor to eliminate to a legitimate value (2). Then after this line is processed the LINECODE editor would do another Kron reduction, eliminating conductor 2, leaving a 1x1 matrix.

This feature might be useful if you have a matrix representing a line with multiple neutrals. Note that if you define a line by its geometry, the Kron reduction of the line geometry object will automatically eliminate all conductors except for the phase conductors.

If you do not reset Neutral= it will be zero following the first reduction operation and nothing will happen.

Retrieved from "http://localhost/mediawiki/index.php?title=LineCode_Kron_Reduction&oldid=338"

---