

Equivalence Check of Op 't Land's and Paul's Field-to-Line Coupling Solutions

Sjoerd Op 'T Land

► **To cite this version:**

| Sjoerd Op 'T Land. Equivalence Check of Op 't Land's and Paul's Field-to-Line Coupling Solutions.
| [Technical Report] ESEO-Tech. 2016. <hal-01265116v2>

HAL Id: hal-01265116

<https://hal.archives-ouvertes.fr/hal-01265116v2>

Submitted on 9 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Equivalence Check of Op 't Land's and Paul's Field-to-Line Coupling Solutions

Sjoerd Op 't Land

May 9, 2016

1 Introduction

To achieve electromagnetic compatibility (EMC), one can try and improve the links in the chain aggressor–coupling path–victim. Often, the coupling path comprises a coupling from an electromagnetic field to a wire-like structure. These couplings remain difficult to grasp, particularly on realistic structures. Therefore, this is a problem of continuing interest for EMC.

Paul studied the coupling of arbitrary electromagnetic fields to multi-conductor transmission lines (MTLs) in vacuum in his book on MTL theory. In Chapter 11, he starts by analysing a bifilar transmission line with arbitrary loads, which is generalised in Chapter 12 to multiconductor transmission lines. The starting point of the analysis is Taylor's representation [1], which lumps the effect of the incident field as voltage and current sources all along the line. Paul then uses chain matrices to spatially translate all the distributed sources to one end of the line. With this solutions, he calculates the terminal currents of a bifilar, generally lossy transmission line, submitted to a physical, but arbitrary electromagnetic field, terminated in arbitrary loads (11.53). He reports that this solution agrees with that obtained by Smith [2]. Paul then presents various specialisations of this general solution, amongst which the case of plane wave excitation [3, (11.65)].

Op 't Land studied the coupling of plane waves to printed circuit board (PCB) traces. In a first paper, he uses Taylor's representation and geometric reasoning to calculate the coupling of a grazing incident, vertically polarised plane wave to a lossless microstrip transmission line with characteristic loads [4]. In Chapter 2 of his thesis, it is extended to the case of multiple segments and arbitrary loads, using plain transmission line theory [5, (2.56)]. This solution agrees fairly with measurements and very well

with 3D full-wave simulations. However, no analytic equivalence with solutions from other authors (like Paul) are given.

This technical paper corroborates the equivalence of Op 't Land's and Paul's solutions.

2 Problem Definition

Op 't Land and Paul studied similar, but different problems, as schematised in Figure 1. Paul is more general in almost all aspects, except that he assumes a homogeneous medium like the vacuum. Op 't Land is much more specific, except that he considers the inhomogeneous medium of a microstrip (substrate-vacuum). Consequently, we can only expect both solutions to agree on the intersection of both problem spaces: a grazing, vertically polarised plane wave incident on a lossless line with arbitrary loads.

In order to let both solutions agree on the medium, a bifilar line in vacuum will be used. This corresponds to a microstrip on an air substrate ($\epsilon_r = 1$ and $\epsilon_{r,eff} = 1$). The distance between both wires will be double that of the substrate thickness h , because the ground plane creates an image of the trace.

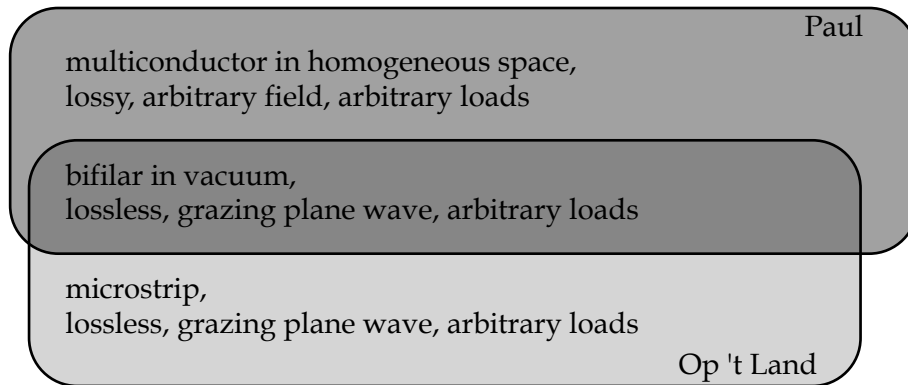
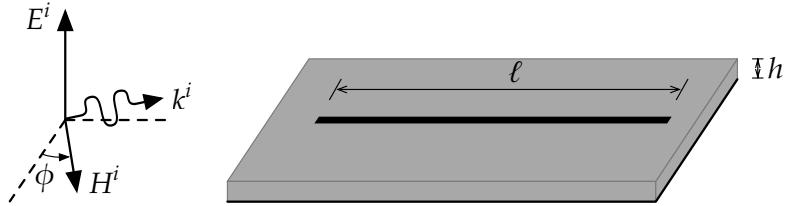


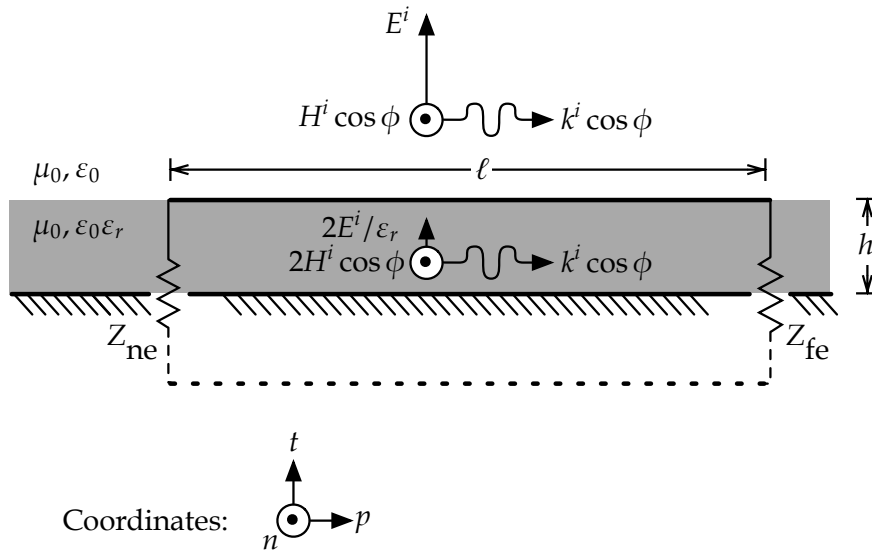
Figure 1: Problems treated by Paul [3] and Op 't Land [5].

In order to compare both equations, all symbols will be replaced to correspond to Op 't Land's definition, according to Table 1. The essential variables are defined in Figure 2.

Both Op 't Land and Paul provide solutions for the near and the far end. Without loss of generality, we will only compare the near-end solutions,



(a) Perspective on the grazing incident wave: the incident electric field is perpendicular to the substrate and the wave vector makes an angle ϕ with the transmission line axis.



(b) Cross section of the transmission line. The incident and reflected plane wave sources produce the shown substrate field.

Figure 2: Definition of the problem variables, after [5, Figure 2.16]

Op 't Land	Paul	Justification
$V_{\text{ne,final}}$	$-\hat{Z}_S \hat{I}(0)$	Ohm's law and [3, Figure 11.5]
$V_{\text{fe,final}}$	$+\hat{Z}_L \hat{I}(\mathcal{L})$	Ohm's law and [3, Figure 11.5]
$2h$	d	Op 't Land's ground plane creates a mirror wire at h below the ground plane, so it is like a bifilar line with a wire separation of $2h$
ℓ	\mathcal{L}	
Z_c	\hat{Z}_C	(Characteristic impedance)
E^i	\hat{E}_o	
$j\beta$	$\hat{\gamma}$	Specialisation to lossless lines [3, (1.33), p. 598]
ϕ	$\phi_p + \pi/2$	[3, Figure 11.6]
$\beta \cos(\phi)$	β_z	
0	β_x	Under grazing incidence, the plane wave propagates in Paul's yz -plane [3, Figure 11.6, (11.56)]
1	e_x	Under horizontally polarised, grazing incidence, the E -field always points in Paul's x direction [3, Figure 11.6, (11.55)]. . .
0	e_y, e_z	. . . and in no other direction [3, Figure 11.6, (11.55)]

Table 1: Symbolic equivalences used to compare Op 't Land's and Paul's solutions.

because they can equally serve to find the far-end solution (by swapping V_{ne} with V_{fe} and augmenting ϕ by π).

Let us first copy Paul's specialisation for plane wave excitation [3, (11.65a/c)] whilst applying Table 1:

$$\begin{aligned}
 V_{\text{ne,final}} = & -Z_{\text{ne}} \frac{2hE^i}{\cos(\beta\ell)(Z_{\text{ne}} + Z_{\text{fe}}) + j\sin(\beta\ell)\left(Z_c + \frac{Z_{\text{ne}}Z_{\text{fe}}}{Z_c}\right)} e^{-j0h} \left[\frac{\sin(0h)}{0h} \right] \left\{ \right. \\
 & -j00 \int_0^\ell \left[\cos(\beta(\ell - \tau)) + j\sin(\beta(\ell - \tau)) \frac{Z_{\text{fe}}}{Z_c} \right] e^{-j\cos(\phi)\beta\ell\tau} d\tau \\
 & \left. +1 \left[\cos(\beta\ell) + j\sin(\beta\ell) \frac{Z_{\text{fe}}}{Z_c} - e^{-j\cos(\phi)\beta\ell} \right] \right\}. \quad (1)
 \end{aligned}$$

Recognising that $\lim_{x \rightarrow 0} \sin(x)/x = 1$ and that the integral has a finite value,

we can simplify:

$$V_{\text{ne,final,Paul}} = -Z_{\text{ne}} 2hE^i \frac{\cos(\beta\ell) + j \sin(\beta\ell) \frac{Z_{\text{fe}}}{Z_c} - e^{-j \cos(\phi)\beta\ell}}{\cos(\beta\ell) (Z_{\text{ne}} + Z_{\text{fe}}) + j \sin(\beta\ell) \left(Z_c + \frac{Z_{\text{ne}} Z_{\text{fe}}}{Z_c} \right)}. \quad (2)$$

Let us also copy Op't Land's solution [5, (2.56)], whilst correcting a typo in his thesis ($Z_c \rightarrow Z_{\text{ne}} \rightarrow Z_{\text{ne}} - Z_c$):

$$\begin{aligned} V_{\text{ne,final,Op'tLand}} &= V_{\text{inc,ne}} + V_{\text{refl,ne,final}} \\ &= \left(V_{\text{ne}} + \Gamma_{\text{fe}} V_{\text{fe}} e^{-j\beta\ell} \right) \left(1 + \frac{Z_{\text{ne}} - Z_c}{Z_c} \frac{Z_{\text{fe@ne}}}{Z_{\text{ne}} + Z_{\text{fe@ne}}} \right), \end{aligned} \quad (3)$$

with the following definitions [5, (2.49), (2.53), (2.54)]:

$$\Gamma_{\text{fe}} = \frac{Z_{\text{fe}} - Z_c}{Z_{\text{fe}} + Z_c} \quad (4)$$

$$\Gamma_{\text{fe@ne}} = e^{-2j\beta\ell} \Gamma_{\text{fe}} \quad (5)$$

$$Z_{\text{fe@ne}} = Z_c \frac{1 + \Gamma_{\text{fe@ne}}}{1 - \Gamma_{\text{fe@ne}}}. \quad (6)$$

The near-end (far-end) voltage on characteristic loads V_{ne} (V_{fe}) is given by [5, (2.27),(2.28)]:

$$V_{\text{ne}} = V_{\text{LF,ne}} K_{\text{ne}} \quad (7)$$

$$V_{\text{fe}} = V_{\text{LF,fe}} K_{\text{fe}} e^{-j\beta\ell}, \quad (8)$$

where V_{LF} is the low-frequency induced voltage and K is the cross-correlation between the incident wave and the line's *eigenwave*. Variants for either end are written together; for example ' \mp ' means minus for the near end (index 'ne') and plus for the far end (index 'fe'). According to Op't Land, they are [5, (2.27-2.29)]:

$$V_{\text{LF}} = jk^i E^i \left(-\frac{\sqrt{\varepsilon_{r,\text{eff}}}}{\varepsilon_r} \mp \cos(\phi) \right) h\ell \quad (9)$$

$$K = \frac{1}{j(-k_p \mp \beta)\ell} \left(e^{j(-k_p \mp \beta)\ell} - 1 \right), \quad (10)$$

where k^i is the incident wave number, which equals the line's wave number β , because of the vacuum. Consequently, the incident wave vector's component parallel to the line k_p equals $\beta \cos(\phi)$. Filling out $\varepsilon_{r,\text{eff}} = \varepsilon_r = 1$, we

obtain:

$$V_{LF} = j\beta E^i (-1 \mp \cos(\phi)) h\ell \quad (11)$$

$$K = \frac{1}{j(-\cos(\phi) \mp 1)\beta\ell} (e^{j(-\cos(\phi) \mp 1)\beta\ell} - 1), \quad (12)$$

and, consequently:

$$V_{ne} = +E^i h (e^{j(-\cos(\phi)-1)\beta\ell} - 1), \quad (13)$$

$$V_{fe} = -E^i h (e^{j(-\cos(\phi)+1)\beta\ell} - 1) e^{-j\beta\ell}. \quad (14)$$

The challenge is to show that both solutions (i.e. (2) and (3)) are equivalent, or whether

$$V_{ne,final,Paul} - V_{ne,final,Op'tLand} = 0. \quad (15)$$

3 Demonstration

Three approaches to prove (15) will now be tried.

3.1 Manual Proof

First, I tried to solve (15) by hand. Apart from the obvious normalisation by $E^i h$, all my attempts were fruitless.

Moreover, extensive bookkeeping seemed inevitable, for example when expanding all complex exponentials to canonical sines and cosines. Assuming a significant probability of error during each manual transformation, this did not seem a robust strategy. Finally, the published proof would be lengthy and the probability of a peer being motivated to spot errors seemed low.

All in all, a manual demonstration did not seem the best strategy to advance science.

3.2 Symbolic Solver

Therefore, the normalised equation was entered into `sympy`, a free Python library for symbolic mathematics (cf. Appendix A). A call on `simplify()` was used to in the hope that the equation would reduce to 0. Various chains of explicit simplification steps were tried (e.g. `trigsimp(deep=True)`, `expand(complex=True)`, `collect(sin(beta*ell))`), but to no avail.

With Python 2.7.10 and `sympy` 0.7.6 on a 3.1 GHz Intel Core i7, `simplify()` takes 13.9 s to conclude that it cannot solve the general problem ($Z_c, \phi, \beta, \ell \in \mathbb{R}$ and $Z_{ne}, Z_{fe} \in \mathbb{C}$). That is, it does not simplify the equivalence equation of (15). That does not prove the non equivalence, it just does not prove the equivalence (and possibly my limited experience with symbolic solvers).

When the problem space is reduced by taking $Z_{ne} = Z_{fe} = Z_c$, however, `simplify()` reworks (15) to 0 within 0.5 s. This means that there is proof of equivalence for a subset of the general problem (characteristic loads).

3.3 Numerical Solver

As a last resort, the equivalence is evaluated on a finite sample of the entire problem space, using `numpy` (cf. Appendix B). Because the sample has a finite extent in the problem space, care had to be taken to define a representative sample.

For simplicity, a hyperrectangular sample was defined. The characteristic impedance Z_c was swept logarithmically from 10 m Ω to 100 Ω on 5 values. The radians per line length $\beta\ell$ was swept linearly in $[0, 4\pi)$ on 29 values; the odd number of points avoids aliasing by modulus- π effects. The azimuth of incidence ϕ was swept in $[0, 2\pi)$ on 20 values. The complex near-end (far-end) load impedance Z_{ne} (Z_{fe}) was generated by sweeping the modulus $|\Gamma_{ne}|$ ($|\Gamma_{fe}|$) in $[0.001, 1)$ (passive loads) on 10 values, and the angle $\angle Z_{ne}$ ($\angle Z_{fe}$) in $[0, 2\pi)$ on 16 values. All in all, $5 \times 29 \times 20 \times (10 \times 16)^2 = 74\,240\,000$ points.

On the aforementioned platform, the equivalence equation was evaluated in 2.0 s. Instead of a symbolic 0, a small error should be expected because of numerical imprecisions. Therefore, the maximum of all absolute errors is reported: 6.2×10^{-14} , which is only three orders of magnitude greater than the minimum `float64` value and very small with respect to the average outcome of 1.4. Hence, it can be considered a ‘numerical zero’.

4 Conclusion

I tried to show the equivalence of Paul’s and Op ’t Land’s solutions of the induced voltages at the terminals of a bifilar transmission line in vacuum under grazing, vertically polarised plane wave illumination. Manual reworking of formulæ did not lead anywhere. Basic symbolic simplification using `sympy` only proved equivalence for the special case of characteristic

loads. Numerical evaluation on the entire problem space strongly suggests that both solutions are equivalent.

References

- [1] C. D. Taylor, R. Satterwhite, and C. W. Harrison Jr., "The response of a terminated two-wire transmission line excited by a nonuniform electromagnetic field," *Antennas and Propagation, IEEE Transactions on*, vol. 13, no. 6, pp. 987–989, Nov. 1965. doi: 10.1109/TAP.1965.1138574.
- [2] A. Smith, "A more convenient form of the equations for the response of a transmission line excited by nonuniform fields," *Electromagnetic Compatibility, IEEE Transactions on*, vol. EMC-15, no. 3, pp. 151–152, 1973. doi: 10.1109/TEMC.1973.303285.
- [3] C. R. Paul, *Analysis of multiconductor transmission lines*. Wiley-Interscience, 2007.
- [4] S. Op 'T Land, R. Perdriau, M. Ramdani, M. Leone, and M. Drissi, "Simple, taylor-based worst-case model for field-to-line coupling," *Progress In Electromagnetics Research - PIER*, vol. 140, pp. 297–311, Jun. 2013. doi: 10.2528/PIER13041207. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00832250>.
- [5] S. Op 'T Land, "Integrated circuit immunity modelling beyond 1 ghz," *Theses, INSA de Rennes*, Jun. 2014. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-01165061>.

A Listing Symbolic Solution: paulVsOptLandSymbolic.py

```

1 from sympy import sin, cos, exp, I, symbols, Function
2 from time import time
3
4 ### Utility functions
5 class Gamma(Function): # Equation (4)
6     @classmethod
7     def eval(cls, Z):
8         return (Z-Zc)/(Z+Zc)
9
10 class Z(Function): # Microwave textbook formula
11     @classmethod
12     def eval(cls, Gamma):
13         return Zc*(1+Gamma)/(1-Gamma)
14
15 class Zatne(Function): # Equation (5)
16     @classmethod
17     def eval(cls, z):
18         return Z(exp(-2j*beta*ell)*Gamma(z))
19
20 class stopwatch(object):
21     def __init__(self, name='That'):
22         self.name = name
23     def __enter__(self):
24         self.startTime = time()
25     def __exit__(self, *args):
26         print '{0} took {1:.2f}s'.format(self.name,
27                                       time()-self.startTime)
28
29
30
31 ### Definition of the problem space
32 Zc = symbols('Zc', real=True, finite=True, positive=True)
33 Zne, Zfe = symbols('Zne, Zfe', finite=True)
34 phi, beta, ell = symbols('phi, beta, ell',
35                            real=True,
36                            finite=True,
37                            positive=True)
38
39 ### Sanity check: utility function unit tests
40 assert Gamma(0) == -1
41 #assert Gamma(oo) == +1 #TODO: take the limit for oo >> Zc
42 assert Gamma(Zc) == 0
43
44 assert Z(0) == Zc
45 assert Z(-1) == 0

```

```

46 #assert Z(1) == oo #TODO: take the limit for oo >> Zc
47
48 assert Zatne(Zc) == Zc
49
50 ### Reduction of the problem space by fixing one or more variables
51 #Zc = 50
52 Zne = Zc
53 Zfe = Zc
54 #phi = 0
55 #beta = 1
56 #ell = 2
57
58 ### Input of Paul's formulation
59 paulD = cos(beta*ell)*(Zne+Zfe) + I*sin(beta*ell)*(Zc+Zne*Zfe/Zc)
60 Vne_final_Paul = -Zne*2*(cos(beta*ell)+I*sin(beta*ell)*Zfe/Zc -
    exp(-1*I*cos(phi)*beta*ell)) / paulD
61
62 ### Input of Op 't Land's formulation
63 Vne = + (exp(I*(-cos(phi)-1)*beta*ell) -1)
64 Vfe = - (exp(I*(-cos(phi)+1)*beta*ell) -1) * exp(-1*I*beta*ell)
65 Zfe_ne = Zatne(Zfe)
66 Vne_final_OptLand = (Vne + Gamma(Zfe)*Vfe*exp(-1*I*beta*ell)) *
    (1+(Zne-Zc)/Zc*Zfe_ne/(Zfe_ne+Zne))
67
68 ### Attempt to prove their equivalence
69 equivalence = Vne_final_OptLand - Vne_final_Paul # Equation (15)
70 # print equivalence.evalf() #Useful when checking at one point in
    the problem space
71
72 with stopwatch('Solution_using_sympy_heuristics'):
73     print equivalence.simplify()
74 #
75 #with stopwatch('Being a bit more specific about the solution
    strategy'):
76 #     print equivalence.expand(complex=True).trigsimp()

```

B Listing Numerical Solution: paulVsOptLandNumerical.py

```

1 from numpy import *
2 from time import time
3
4 %% Utility functions
5 class stopwatch(object):
6     def __init__(self, name='That'):
7         self.name = name
8     def __enter__(self):
9         self.startTime = time()
10    def __exit__(self, *args):
11        print '{0} took {1:.2f}s'.format(self.name,
12                                       time()-self.startTime)
13
14 def s2z(Gamma):
15     return Zc * (1+Gamma)/(1-Gamma)
16 def z2s(Z):
17     return (Z-Zc)/(Z+Zc)
18
19 def alignAlong(vector, axisNumber):
20     sizes = ones(dimensions)
21     sizes[axisNumber] = -1
22     return vector.reshape(sizes)
23
24 def sweepImpedance(startDimension):
25     rho = alignAlong(linspace(0.001, 1, 10, endpoint=False),
26                      startDimension)
27     argGamma = alignAlong(linspace(0, 2*pi, 16), startDimension+1)
28     Gamma = rho * exp(1j*argGamma)
29     return s2z(Gamma)
30
31 def equals(title, equation1, equation2):
32     errors = (abs(equation2 - equation1))
33     threshold = finfo(errors.dtype).eps * 1000
34     assert len(errors.shape) == dimensions
35     print title
36     print 'Shape of the swept space:', errors.shape
37     print 'Numerical equality: {0} (Error between {1:.1e} and
38           {2:.1e} < {3:.1e}, average absolute value {4:.1e})'.format
39           (
40               all(errors < threshold),
41               errors.min(),
42               errors.max(),
43               threshold,
44               abs(equation1).mean())
45     print ''

```

```

43
44 ### Definition of the problem space
45 dimensions = 1+2+2+1+1
46
47 Zc = alignAlong(logspace(-2,2,5),0)
48 Zne = sweepImpedance(1)
49 Zfe = sweepImpedance(3)
50 betaEll = alignAlong(linspace(0,4*pi,29,endpoint=False),5)
51 phi = alignAlong(linspace(0,2*pi,20,endpoint=False),6)
52
53 ### Input of Paul's formulation
54 paulD = cos(betaEll)*(Zne+Zfe) + 1j*sin(betaEll)*(Zc + (Zne*Zfe)/
    Zc)
55 Vne_final_Paul = -Zne * 2.0 * (cos(betaEll) + 1j*sin(betaEll)*Zfe/
    Zc - exp(-1j*betaEll*cos(phi))) / paulD
56
57 ### Input of Op 't Land's formulation
58 Vne = +1 * (exp(1j*(-1.0-cos(phi))*betaEll) -1.0)
59 Vfe = -1 * (exp(1j*(+1.0-cos(phi))*betaEll) -1.0) * exp(-1j*
    betaEll)
60 Zfe_ne = s2z(z2s(Zfe)*exp(-2j*betaEll))
61 Vne_final_OptLand = (Vne + z2s(Zfe)*Vfe*exp(-1j*betaEll)) * (1 +
    ((Zne-Zc)/Zc) * Zfe_ne/(Zne+Zfe_ne))
62
63 ### Attempt to prove their equivalence
64 with stopwatch('Solution using numpy sweep over {0} samples'.
    format(Vne_final_OptLand.size)):
65     equals('Grazing incidence on bifilar line with (Zne,Zfe)',
66           Vne_final_OptLand,Vne_final_Paul)

```