Φωτονική Τεχνολογία

Συνοδευτικό Υλικό για Προαιρετικό Θέμα Κεφαλαίου 3

Αλέξανδρος Πιτιλάκης / alexpiti@auth.gr / Θεσσαλονίκη / 27 Νοεμβρίου 2015

Γενικές οδηγίες

1. Οι ρουτίνες που έχουν ετοιμαστεί δοκιμάστηκαν σε **MATLAB R2014A 64bit**, οπότε να είστε επιφυλακτικοί για τη λειτουργία τους σε (πολύ) παλαιότερες διανομές. Πιστεύω πως οι διανομές μετά το 2011 θα δουλεύουν κανονικά. Κάποιες ρουτίνες (π.χ. FIDODIES) δεν δουλεύουν σε 32bit.
2. Ανοίξτε με τον MATLAB editor (ή με notepad++) τα παραδείγματα-scripts που είναι στο root-folder του ZIP που αποσυμπιέσατε και **ΔΙΑΒΑΣΤΕ ΚΑΛΑ ΟΛΑ ΤΑ ΣΧΟΛΙΑ** που βρίσκονται πάνω/δίπλα από κάθε εντολή, πριν την εκτελέσετε. Έπειτα εκτελέστε σιγά-σιγά τα διάφορα blocks, βάζοντας break-points ή σπάζοντας τα σε sections.
3. Τα scripts καλούν εξειδικευμένες ρουτίνες που βρίσκονται στους επιμέρους φακέλους (BPM, Misc, Solver) οι οποίοι φορτώνονται **στο MATLAB path**, στην αρχή του κάθε script.
4. Οι εξειδικευμένες ρουτίνες έχουν επίσης σχόλια, αλλά και συγκεκριμένες οδηγίες χρήσης στην αρχή τους. Μπορείτε να τις δείτε με "`help <mfilename>`" ή "`doc <mfilename>`" στο command-window του MATLAB (χωρίς τα quotes, όπου `<mfilename>` το αρχείο της συνάρτησης, π.χ. `MLSWG`).
5. Οι περισσότερες από τις εξειδικευμένες ρουτίνες έχουν δοκιμαστικά /ενδεικτικά ορίσματα εισόδου (συγκεκριμένα: μέσα στο "`if nargin==0`" blocks), για να πάρετε μία εικόνα του τι κάνουν. Για τη λειτουργία αυτή, καλέστε τη ρουτίνα από το command-window του MATLAB χωρίς καθόλου ορίσματα εισόδου (π.χ. σκέτο `MLSWG`) ή με το κουμπί F5 όταν την έχετε ανοικτή στον MATLAB editor.
6. Διαβάστε/εκτελέστε πρώτα το παράδειγμα-script του Solver και μετά αυτά της BPM (ξεκινώντας από τον απλό coupler), που είναι αρκετά πιο εκτενή.
7. Τα σχόλια και όλοι οι κώδικες είναι στα αγγλικά. Συγχωρέστε τα λάθη/typos!

Mode Solver

Πρόκειται για εργαλεία που χρησιμοποιούνται για τον αριθμητικό υπολογισμό των ρυθμών μίας διάταξης, εν προκειμένω κυματοδήγησης σε μονοδιάστατη διατομή.

Ο solver **MLSWG** βασίζεται στη μέθοδο **Newton-Raphson** (NR) για την επίλυση της μη-γραμμικής **χαρακτηριστικής εξίσωσης** (ΧΕ) ενός MultiLayer Slab Waveguide (κυματοδηγού πολλαπλών παραλλήλων πλακών). Λόγω της μορφής της εξίσωσης και της φύσης της μεθόδου NR, είναι πιθανό σε μία εκτέλεση να μη βρεθούν όλες οι ρίζες της ΧΕ που αντιστοιχούν στους ρυθμούς του κυματοδηγού. Στην περίπτωση αυτή, η επανάληψη του MLSWG 2 ή 3 φορές συνήθως λύνει το πρόβλημα. Αν κοιτάξετε μέσα στη ρουτίνα MLSWG.m θα βρείτε τις παραμέτρους που ελέγχουν τη σύγκλιση της μεθόδου, και μπορείτε να τις πειράξετε.

Εναλλακτικά, μπορεί να χρησιμοποιηθεί ο solver **πεπερασμένων διαφορών** (finite-difference, FD) με το όνομα **FIDODIES**. Ο solver αυτός χρειάζεται μία FD-διακριτοποίηση του προφίλ δείκτη-διάθλασης στη διατομή του κυματοδηγού πολλαπλών παραλλήλων πλακών. Για παράδειγμα, αν η διατομή είναι `x=linspace(-5,5,10000)` και το πάχος του κυματοδηγού είναι `w=2` κεντραρισμένο στο σημείο `xc=0` και οι δείκτες υποστρώματος και οδήγησης είναι `nsub=1.45` και `ngui = 3.20`, τότε η FD-διακριτοποίηση της διατομής θα είναι:

```
nFD = nsub*ones( size(x) );
nFD( abs(x-xc)<=w/2 ) = ngui;
```

Ο FIDODIES χρησιμοποιεί τη ρουτίνα SPTARN του MATLAB για εύρεση ιδιοτιμών συστήματος εξισώσεων, πάντα βρίσκει όλους τους ρυθμούς, αλλά συνήθως έχει χαμηλότερη ακρίβεια από τον MLSWG.

Ανοίξτε τις αντίστοιχες ρουτίνες για περισσότερες λεπτομέρειες.

Beam Propagation Method

Η Μέθοδος Διάδοσης Δέσμης υπολογίζει τη διάδοση μία δέσμης-διέγερσης κατά μήκος μίας διάταξης με σαφώς ορισμένο άξονα (z-axis). Χρησιμοποιείται βηματικός αλγόριθμος για τον υπολογισμό του πεδίο σε κάθε επόμενο σημείο του άξονα διάδοσης, π.χ. z1=z0+dz, με γνώση μόνο του πεδίου στο τρέχον σημείο του άξονα, π.χ. z0=0. Η συγκεκριμένη υλοποίηση της BPM βασίζεται στη Μέθοδο των Πεπερασμένων Διαφορών (FDM) για τη διακριτοποίηση της εγκάρσιας διατομής της διάταξης (x-axis). Θα περιοριστούμε στη μελέτη ρυθμών TE πόλωσης, δηλαδή με μοναδική συνιστώσα ηλεκτρικού πεδίου κατά τον y-άξονα.

Παρακάτω σχολιάζονται οι βασικές παράμετροι της FD-BPM που θα σας χρειαστούν:

- **<u>FD διακριτοποίηση διατομής (x-axis):</u>** Δημιουργείται με τη ρουτίνα LINSPACE του MATLAB και θα πρέπει να έχει ένα βήμα dx<λ/20.
- **<u>Μέγεθος βήματος διάδοσης (z-axis):</u>** Αποτελεί παράμετρο που δίνεται στον CELL πίνακα "Struct_Layout"(*) που προσδιορίζει με συστηματικό τρόπο τη κάτοψη της διάταξης που θα μελετηθεί με την FD-BPM. Για κυματοδηγούς που δεν μεταβάλλονται κατά τον z-άξονα (ομοιόμορφη διατομή) το βήμα μπορεί να είναι της τάξης του dz~λ/2. Για κυματοδηγούς υπό κάμψη ή μεταβλητού πάχους (tapered), θα πρέπει dz~λ/8.
- **<u>Δείκτης αναφοράς (nref):</u>** Θα πρέπει να είναι θετικός πραγματικός αριθμός, συνήθως ίσος με τον ενεργό δείκτη διάθλασης (n_effective) του βασικού ρυθμού ενός κυματοδηγού. Όταν ο nref αποκλίνει πολύ από τον n_effective εμφανίζονται "αριθμητικές απώλειες", δηλαδή φαίνεται πως η ένταση της δέσμης μειώνεται χωρίς αυτό να είναι φυσικά αναμενόμενο.
- **<u>Διέγερση εισόδου:</u>** Προσδιορίζεται επάνω στο x-axis που δημιουργήσαμε παραπάνω και αναφέρεται στο επίπεδο εισόδου της διάταξης (z=0). Συνήθως λαμβάνεται από κάποιον solver (MLSWG ή FIDODIES) ή μπορεί να δοθεί και προσεγγιστικά, π.χ. με χρήση "καμπάνας" γκαουσιανού σχήματος και κατάλληλου εύρους, π.χ. `E_in = exp( -(x/wid).^2 );`

(*) Για τη μορφή του CELL πίνακα "Struct_Layout", δείτε τα παραδείγματα-scripts ή τις οδηγίες στη ρουτίνα BPM\BPMFD2D_DoProp.

Troubleshooting

Η βασική πηγή λαθών θα είναι τα ορίσματα εισόδου. Οι εξειδικευμένες ρουτίνες δεν έχουν εκτενές input-argument error-checking, οπότε αν περάσετε λάθος ορίσματα εισόδου (λάθος τύπος ή λάθος πλήθος) η εκτέλεση θα σταματάει σε σφάλμα. Δείτε το HELP των ρουτίνων (πάνω-πάνω) ή εξετάστε τα δοκιμαστικά ορίσματα εισόδου που προαναφέρθηκαν για να βεβαιωθείτε ότι δίνεται τα σωστά ορίσματα.

Κάντε σαφώς τη διάκριση στο μυαλό σας μεταξύ των ΤΥΠΩΝ των μεταβλητών:

- **string:** σειρά χαρακτήρων, e.g. myStr='TE'; or myStr='This is a string';
- **scalar:** βαθμωτό μέγεθος (π.χ. ακέραιος, πραγματικός ή μιγαδικός αριθμός), e.g. x=1; y=25.5; E=7+5j;
- **boolean** : true/false ή 1/0 μόνο, e.g. bool1 = 1; bool2=false;

Οι μεταβλητές αποθηκεύονται σε πίνακες.

- **vector:** διανυσματικό μέγεθος ή πίνακας γραμμή/κολώνα, e.g A=1:1:10; B=[3.4;5j;6]; StrVect='thisIsAVector';
- **matrix:** πίνακας δύο διαστάσεων (π.χ. 3x3 ή 5-by-4), e.g. D=[2 3;4 5;6j 7j];
- **array:** πίνακας N διαστάσεων (N=2 είναι το ίδιο με matrix).

Τέλος, στις εξειδικευμένες ρουτίνες χρησιμοποιούνται συχνά πίνακες τύπου CELL, που είναι ένας τύπος που συνδυάζει κάποια πλεονεκτήματα από τους απλούς πίνακες και τα structures. Τα CELL μπορούν να αποθηκεύουν μεταβλητές ΔΙΑΦΟΡΕΤΙΚΟΥ ΤΥΠΟΥ, π.χ. myCell = { 'alex' , [7.9 4.5] , 1982; 'jon' , [5.0 6.2] , 2012 }; και ορίζονται με τις αγκύλες {}. Οπότε, ΠΡΟΣΟΧΗ να μην μπερδευτείτε με τα () ή []. Μπορούμε να τραβήξουμε μία scalar τιμή από το παραπάνω CELL με τον εξής τρόπο: myScalar=myCell{1,2}(2) που θα γυρίσει την τιμή 4.5.

Probe further

Οι εξειδικευμένες ρουτίνες που σας δόθηκαν μπορούν να εκτελέσουν αρκετά πιο σύνθετες λειτουργίες από αυτές που θα χρειαστούν για το μάθημα. Αν έχετε όρεξη να προχωρήσετε παραπέρα, μπορείτε να παίξετε με τις εσωτερικά σεταρισμένες παραμέτρους που έχουμε θέσει για τις ανάγκες των ασκήσεων. Απορίες κλπ, στο email μου.

Σημείωση

Οι κώδικες είναι copyrighted, οπότε δώστε credit αν χρησιμοποιήσετε κομμάτια τους "παραέξω" ;)