ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

Πληροφορική

Ενότητα 12: Υπολογιστική και Μηχανολογία: το «παιχνίδι της ζωής» και η προσομοίωση δομικών μεταβολών χάλυβα και δασικών πυρκαγιών

Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Υπολογιστική & και Μηχανολογία: το «παιχνίδι της ζωής» και η προσομοίωση δομικών μεταβολών χάλυβα και δασικών πυρκαγιών

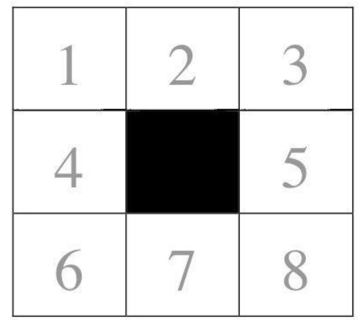
Το παιχνίδι της ζωής του Conway

- Σύλληψη του άγγλου μαθηματικού John Conway (1970)
  - Αποτελείται από πολύ απλούς κανόνες σε σχέση με «οργανισμούς» και τη γέννηση, την επιβίωση και το θάνατό τους
  - Θεωρείται ένα από τα πιο κλασσικά παραδείγματα «κυτταρικών αυτομάτων»
  - Συνδέεται με ποικίλες άλλες περιοχές της τεχνητής νοημοσύνης, όπως οι μηχανές Turing και οι ιοί υπολογιστών.

“Το παιχνίδι της ζωής”

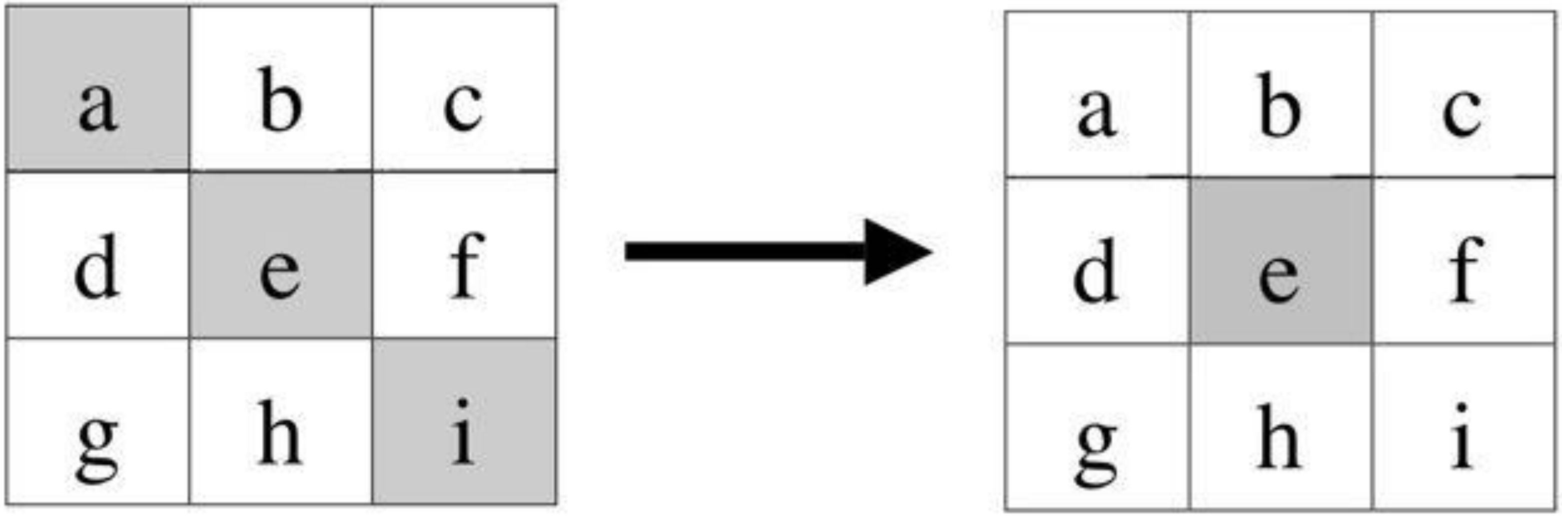
- Οι οργανισμοί “ζουν” σε κάνναβο

- Κάθε κελί έχει 8 “γείτονες”



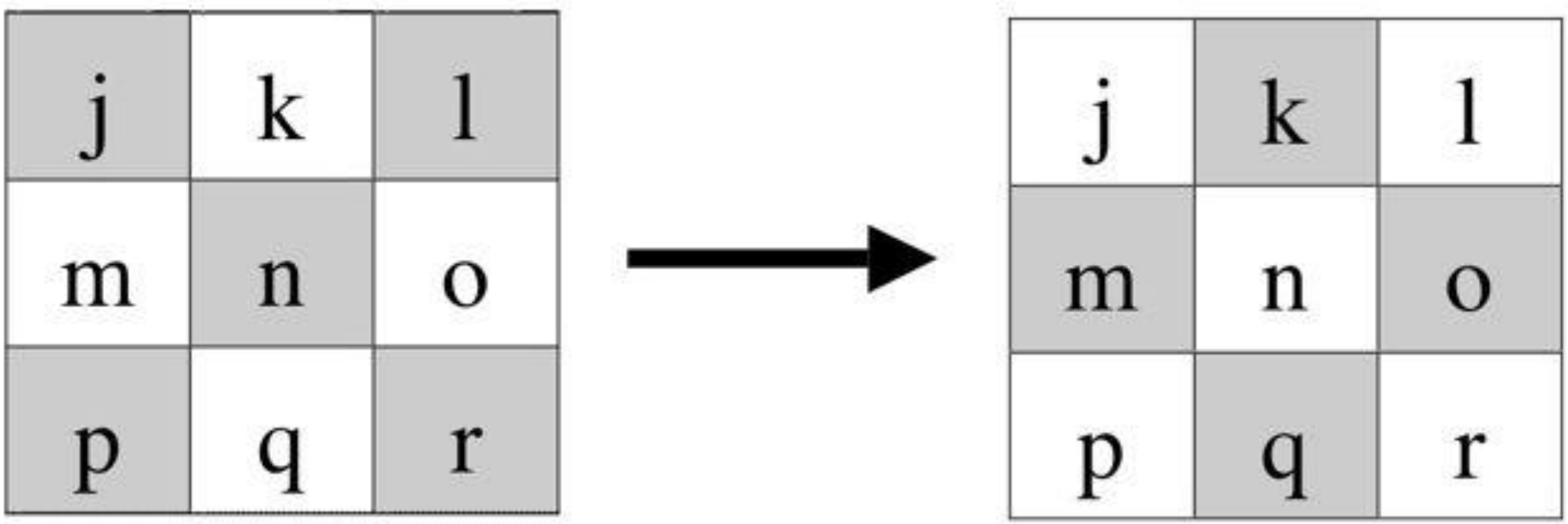
ΠτΖ: κανόνες

- Βασικός κανόνας: σε κάθε κελί υπάρχει (ή όχι) ένας ζωντανός οργανισμός. Σε κάθε κύκλο του “παιχνιδιού” οργανισμοί επιβιώνουν, γεννιούνται και πεθαίνουν ως ακολούθως:
  - Υφιστάμενος οργανισμός επιβιώνει εάν έχει 2 ή 3 ζωντανούς γείτονες
  - Ένας οργανισμός πεθαίνει από απομόνωση εάν έχει λιγότερους από 2 γείτονες
  - Ένας οργανισμός πεθαίνει λόγω “υπερπληθυσμού” εάν έχει περισσότερους από 3 γείτονες
  - Εάν ένα ακατοίκητο κελί έχει ακριβώς 3 γείτονες, θα κατοικηθεί στην επόμενη γενιά



Initial state

After one step



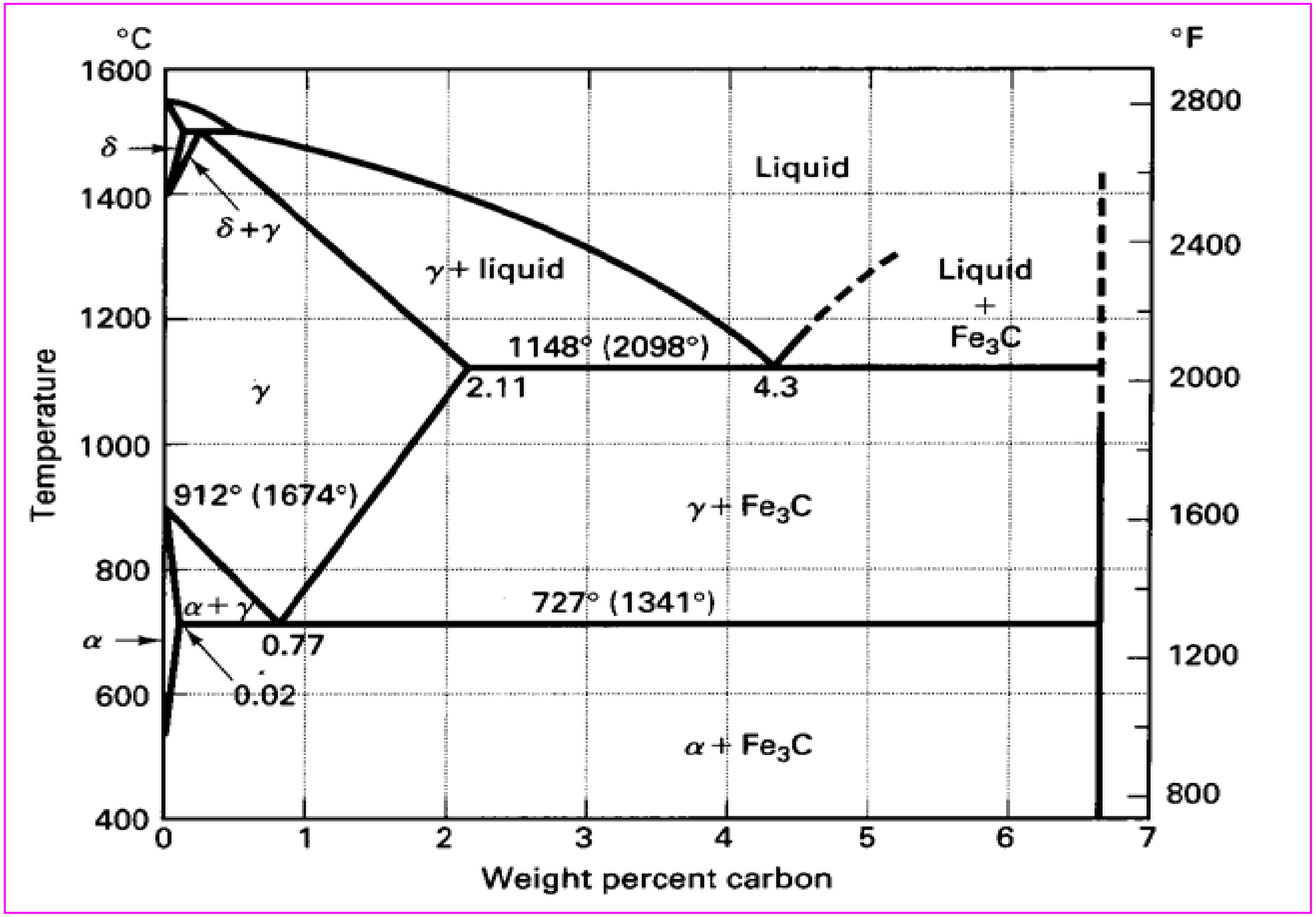
Initial state

After one step

Αυτή η αλλαγή “κατάστασης” μας θυμίζει κάτι….

Βασικά στοιχεία θεωρίας

- Τι συμβαίνει στον χάλυβα όταν ψυχθεί;



α, *ferrite* (*BCC*)

γ, *austenite* (*FCC*)

$\delta, \delta -$ *ferrite* (*BCC*)

Fe_3C, *cementite* (6.67%*C*)

Curie po. nonmagnetic to

magnetic transition

Ο καθαρός σίδηρος πρίν λειώσει στους 1538°C, αλλάζει 2 φορές

- ***Φερρίτης*** ή ***α σίδηρος***: δομή BCC, μέχρι 912°C.
- ***Ωστενίτης*** ή ***γ σίδηρος***: δομή FCC, μέχρι 1394°C
- ***δ φερρίτης***: δομή BCC μέχρι τους 1538°C

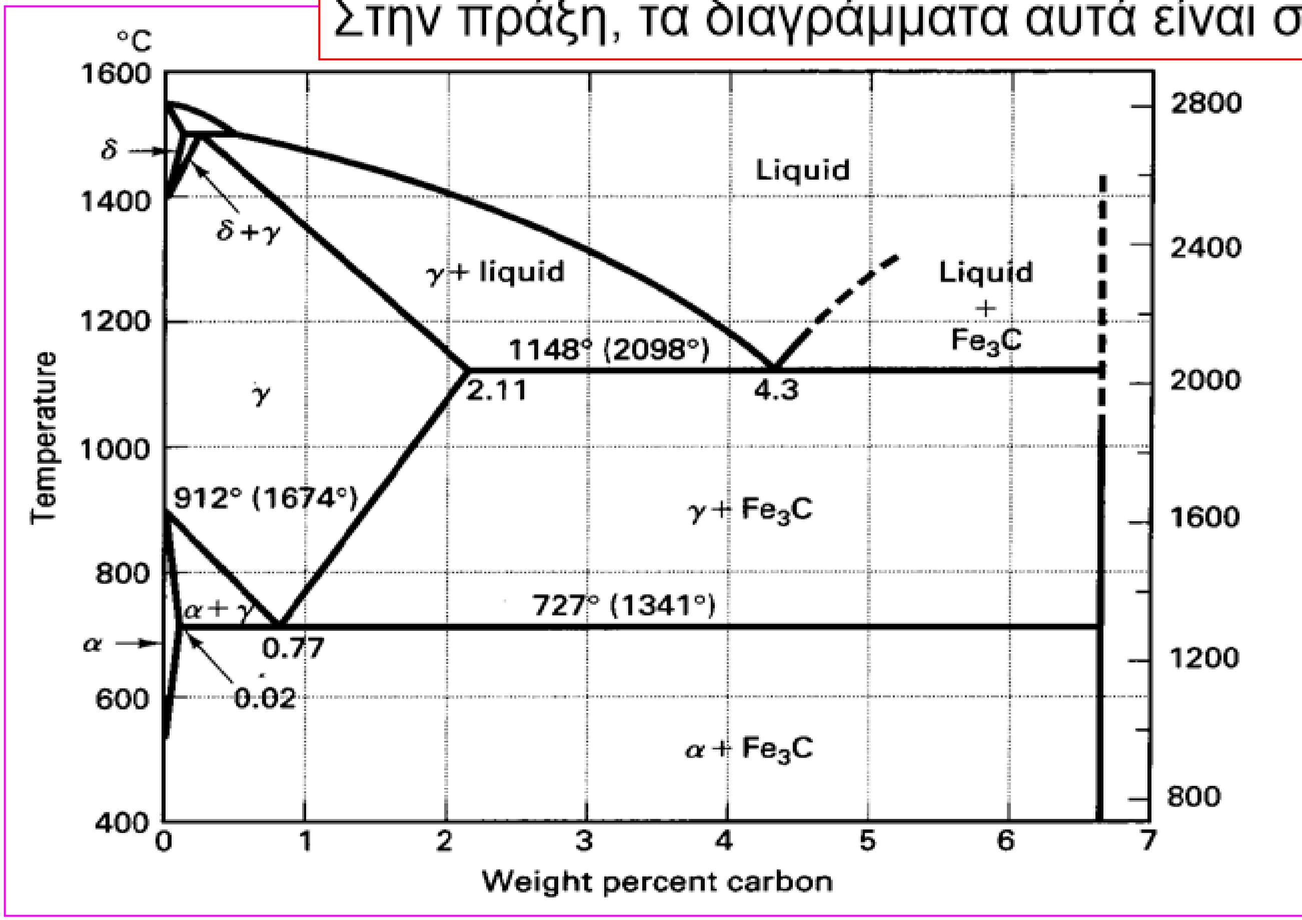
Σε περιεκτικότητα 6.7 wt% C, έχουμε τον ***Σεμεντίτη*** (Fe_3C)

Στην πράξη, τα διαγράμματα αυτά είναι σιδήρου-σεμεντίτη

Bo

- Τι συμβαίνε

Σύσταση (wt% C)



α, ferrite (BCC)

γ, austenite (FCC)

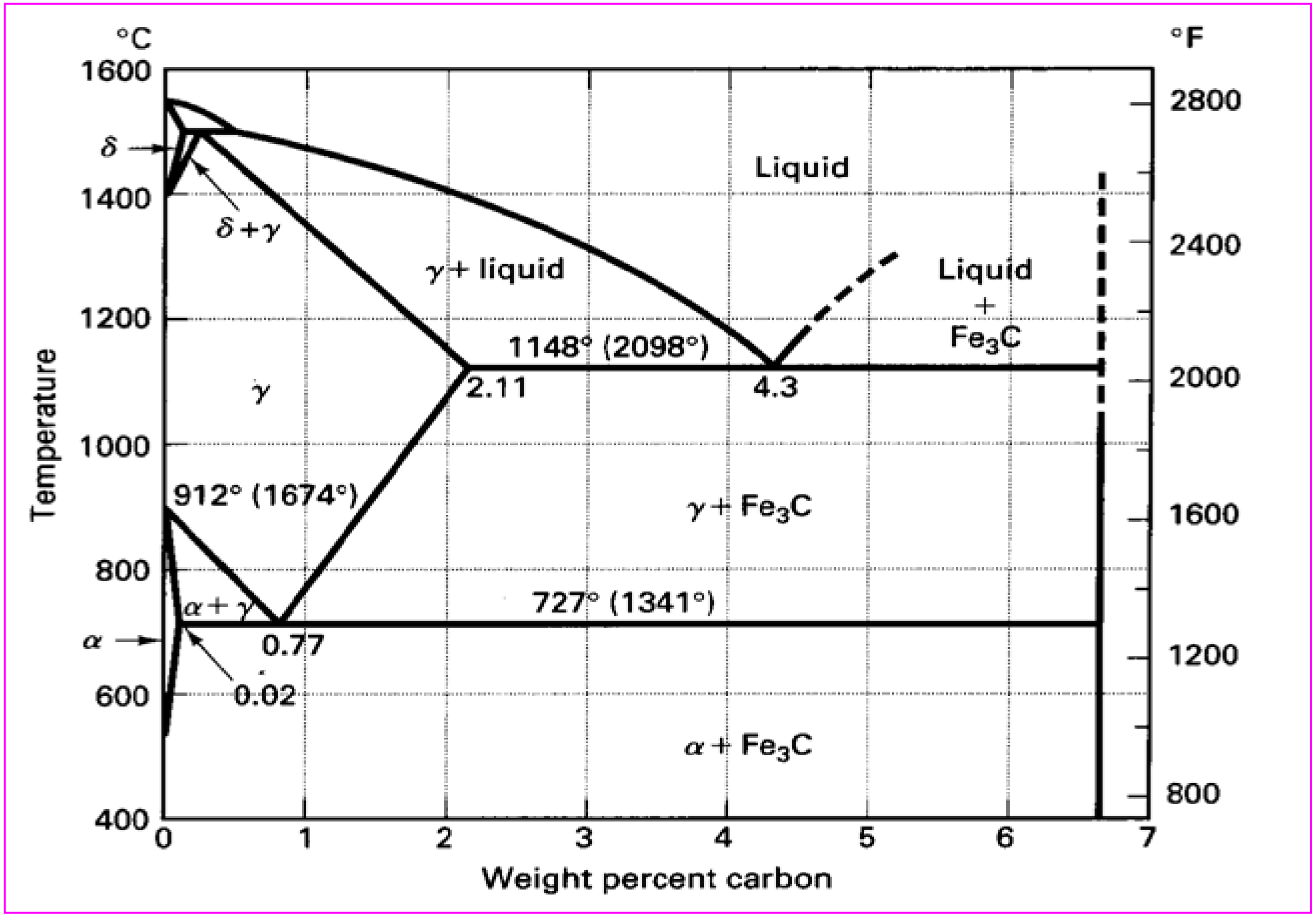
δ, δ – ferrite (BCC)

Fe_3C, cementite (6.67%C

Curie po. nonmagnetic to magnetic transition

Βασικά στοιχεία θεωρίας

- Τι συμβαίνει στον χάλυβα όταν ψυχθεί;



α, *ferrite* (BCC)

γ, *austenite* (FCC)

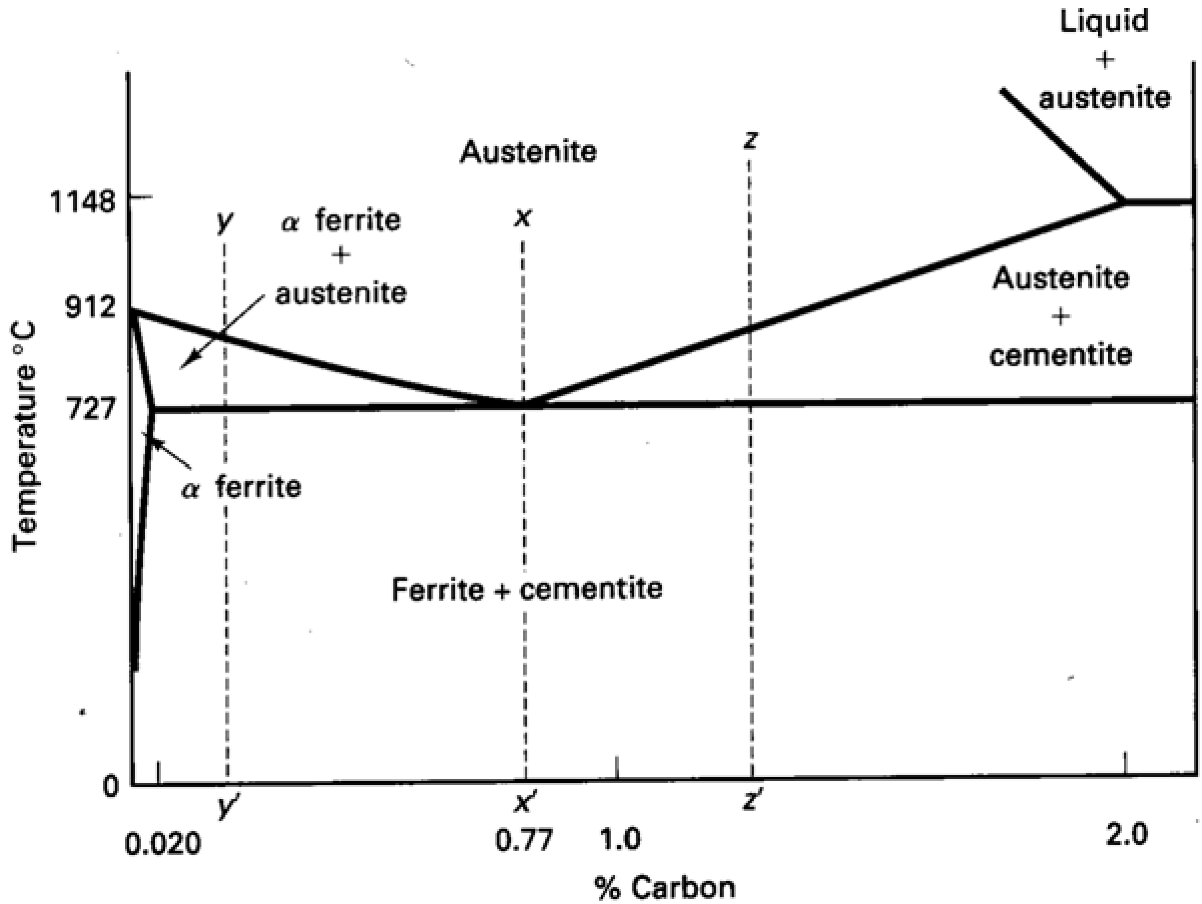
$\delta, \delta -$ *ferrite* (BCC)

Fe_3C, *cementite* $(6.67\%C$

Curie po. nonmagnetic to

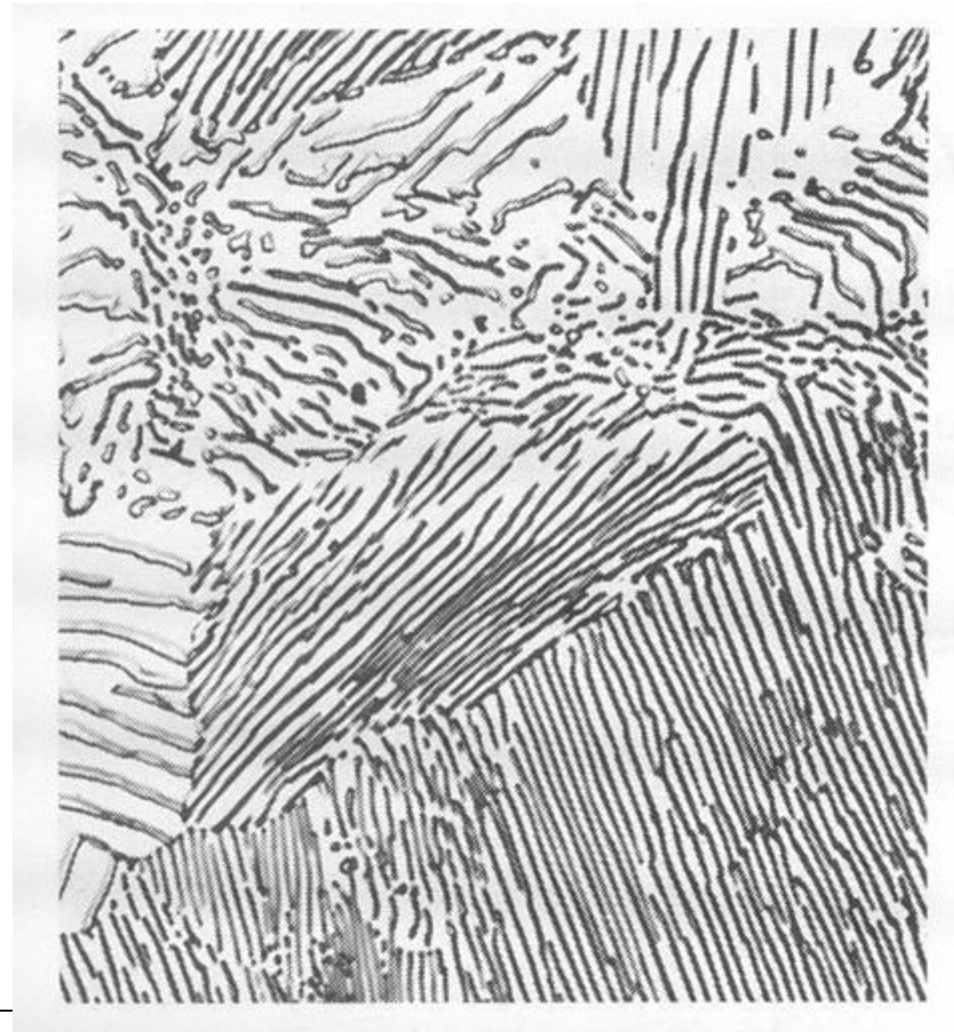
magnetic transition

Βασικά στοιχεία θεωρίας

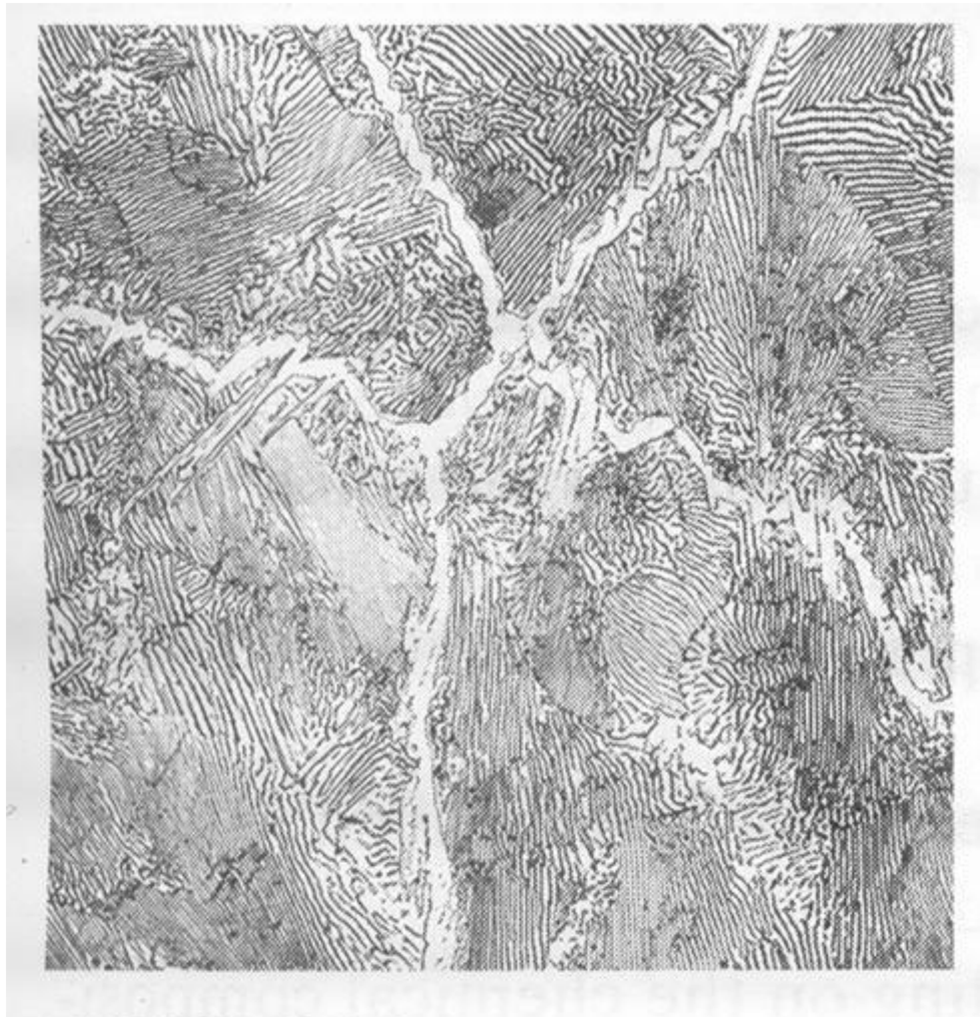


$$Austenite_{0.77\%C;FCC} \rightarrow Ferrite_{0.02\%C;BCC} + Cementite_{6.67\%C}$$

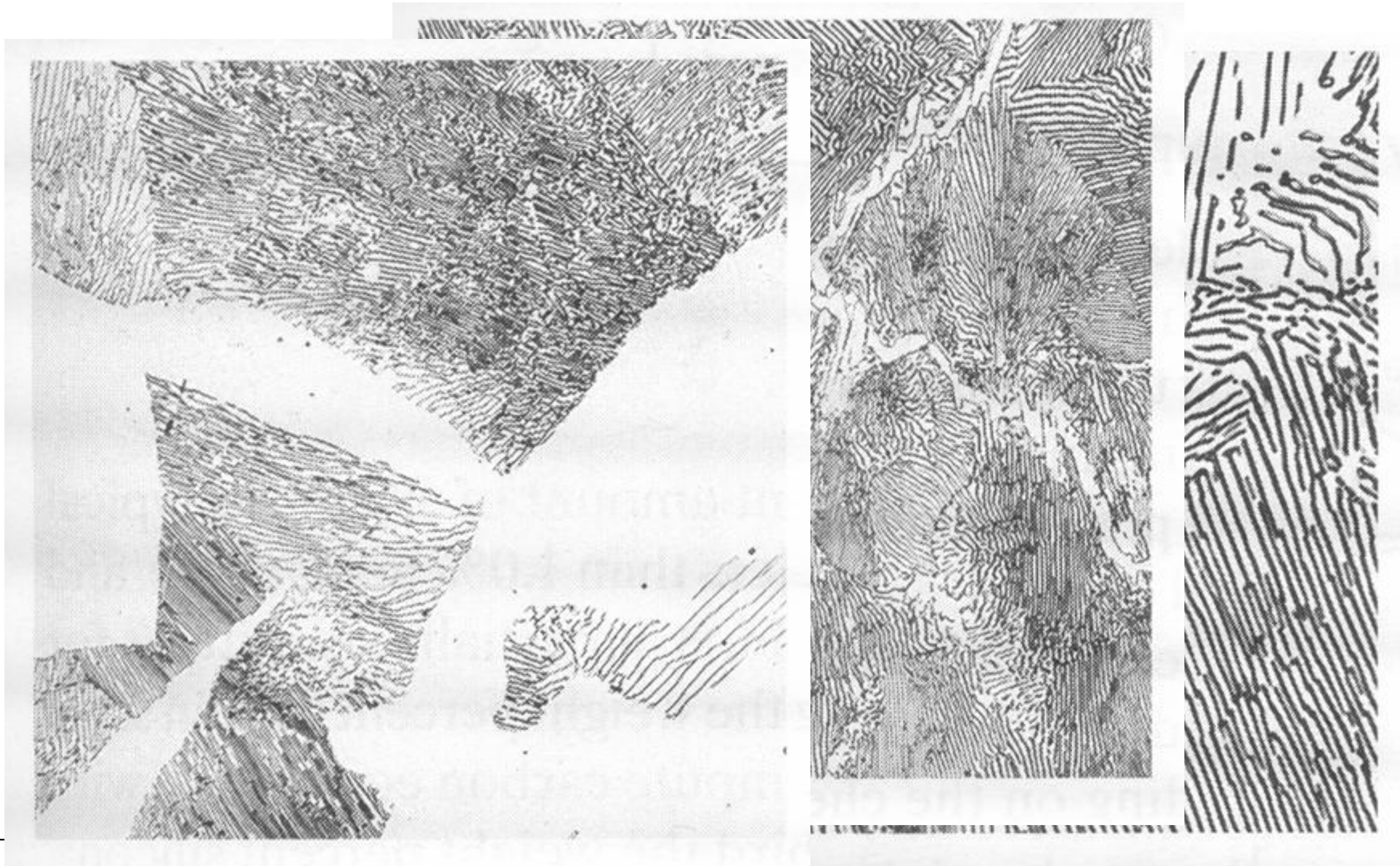
Κάποιες δομές συναρτήσει % C



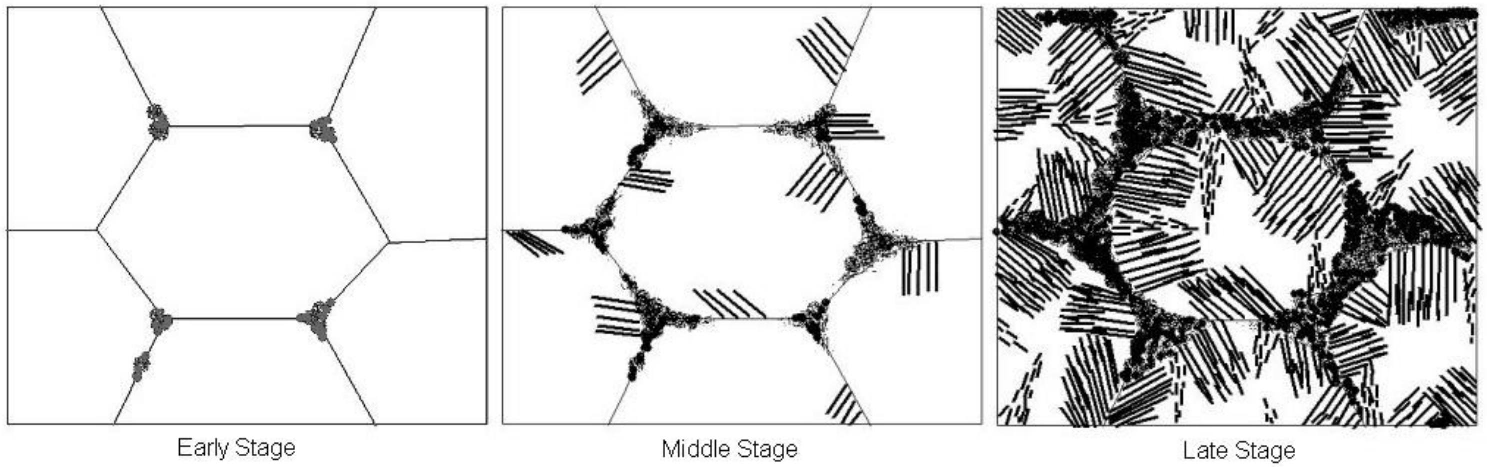
Κάποιες δομές συναρτήσει % C



Κάποιες δομές συναρτήσει % C



Μικροδομές χάλυβα & ΠτΖ



- Θεωρούμε αρχική εικόνα μικροδομής χάλυβα που απαρτίζεται από ωστενίτη και μερικούς μόνο πυρήνες φερρίτη. Κάθε κυψελίδα της εικόνας (pixel) που γειτονεύει με φερρίτη, λαμβάνει έναν βαθμό πιθανότητας μετατροπής σε φερρίτη. Η αλλαγή γίνεται στη βάση αυτών των βαθμών, και ακολουθώντας συγκεκριμένους κανόνες.

Κανόνες για υλοποίηση

- Εάν μία κυψελίδα φερρίτη γειτνιάζει με 0,1,4,5,6,7 ή 8 κυψελίδες φερρίτη, τότε καταστρέφεται (0,1 από «μοναξιά», 4-8 λόγω «υπερπληθυσμού»)
- Εάν μία κυψελίδα φερρίτη γειτνιάζει με 2 ή 3 κυψελίδες φερρίτη, τότε επιβιώνει και στο επόμενο χρονικό βήμα
- Εάν μία κυψελίδα ωστενίτη γειτνιάζει με 3 κυψελίδες φερρίτη, τότε μετατρέπεται σε φερρίτη.

Literature

- http://dierk-raabe.com/cellular-automaton-model/

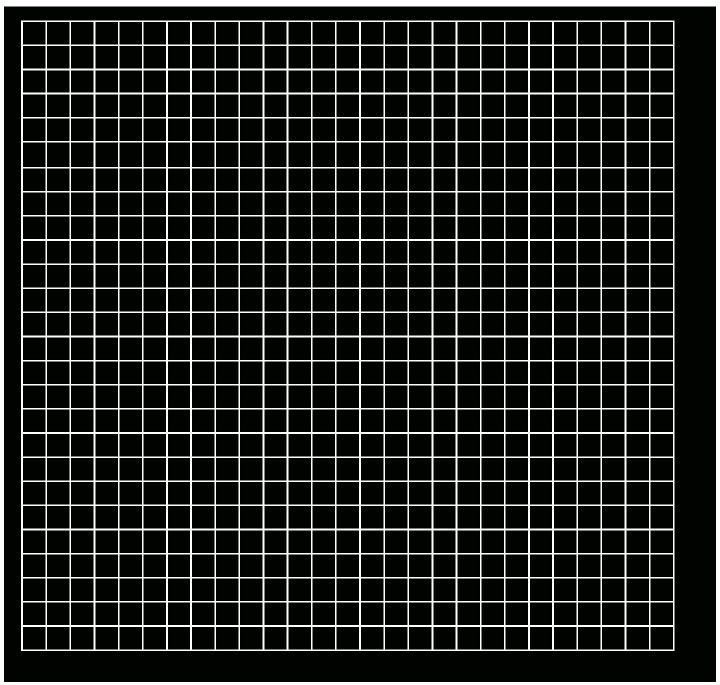
Πηγαίνοντας παρακάτω….

GoL και διάχυση

- Διάχυση: διαδικασία βάσει της οποίας μία ιδιότητα ή μια ποσότητα «μεταφέρεται» από μία «περιοχή» του «χώρου» σε άλλη.
  - Θερμότητα
  - Βαφή σε υγρό
  - Κίνηση μυρμηγκιών
  - ….

GoL και διάχυση

- *Δάσος = πίνακας nxn*
- *Κατάσταση κελιών δάσους = 0, 1, 2 (άδειο, δένδρο, καιγόμενο δένδρο)*



- *probTree* = πιθανότητα ύπαρξης δένδρου στο cell (μπορεί να προκύψει από χάρτη χρήσεων γής, μέσω δορυφορικής επισκόπησης)
- *probBurning* = πιθανότητα ύπαρξης καιγόμενου δένδρου στο cell (μπορεί να προκύψει από είδος βλάστησης και μετεωρολογικές συνθήκες)

Αρχικοποίηση (ψευδοκώδικας)

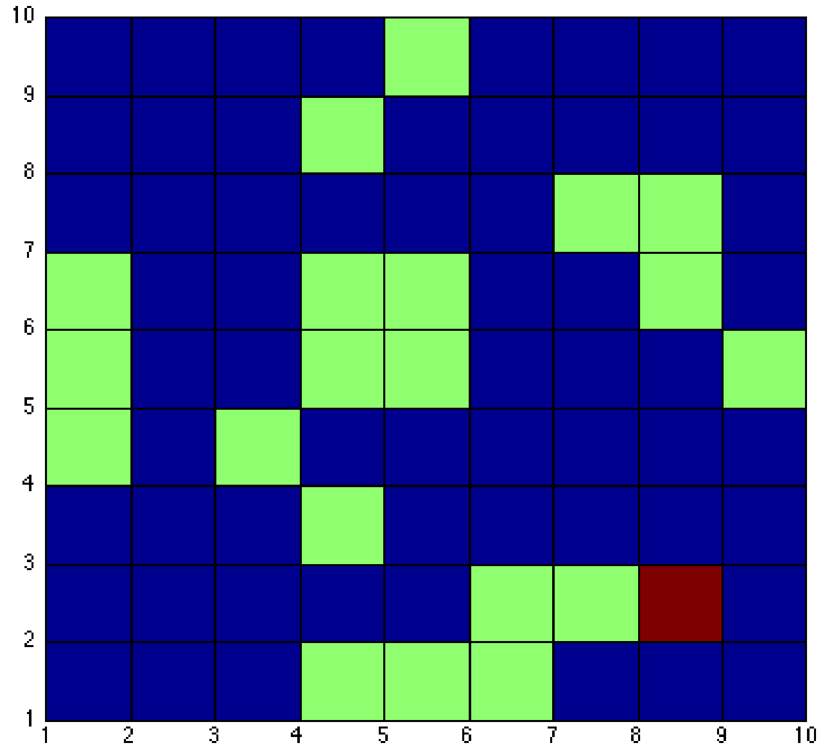
```
if ένας τυχαίος > probTree
  if άλλος τυχαίος > probBurning
     τότε cell=2     // το δένδρο καίγεται
  else
     τότε cell=1     // το δένδρο δεν καίγεται
else
  cell=0        // δεν υπάρχει δένδρο
```

Αρχικοποίηση (Matlab)

```
grid = zeros(n); % N x N grid με μηδενικά

rand1 = rand(n);  % N x N τυχαίοι στο (0,1) (ύπαρξη δένδρου)

rand2 = rand(n);  %(για έλεγχο κάυσης δένδρου)

grid(rand1 > probTree) = 1;

grid(rand1 > probTree &  rand2 > probBurning) = 2;
```

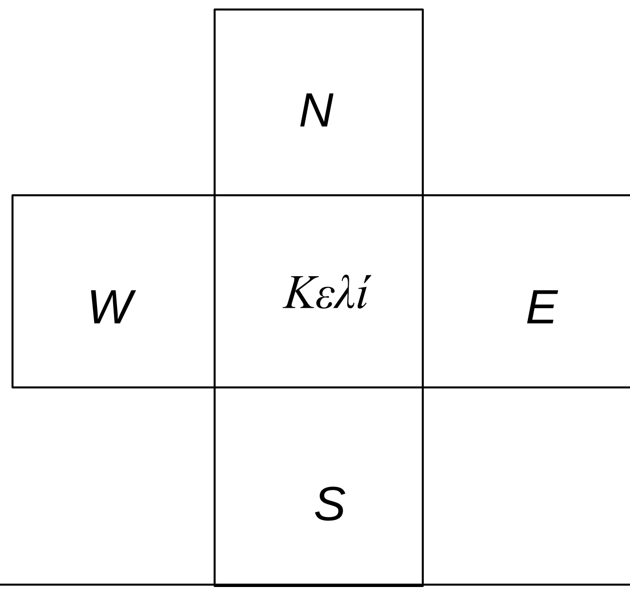
Αρχικοποίηση(Matlab)

```
pcolor(grid)

axis square
```



Κανόνες ανανέωσης

Προσδιορίζουμε την κατάσταση των κελιών στο χρόνο $t + 1$ στη βάση της τρέχουσας κατάστασης (χρόνος t) και την κατάσταση των γειτόνων του στην γειτονιά von Neumann (N, S, E, W):

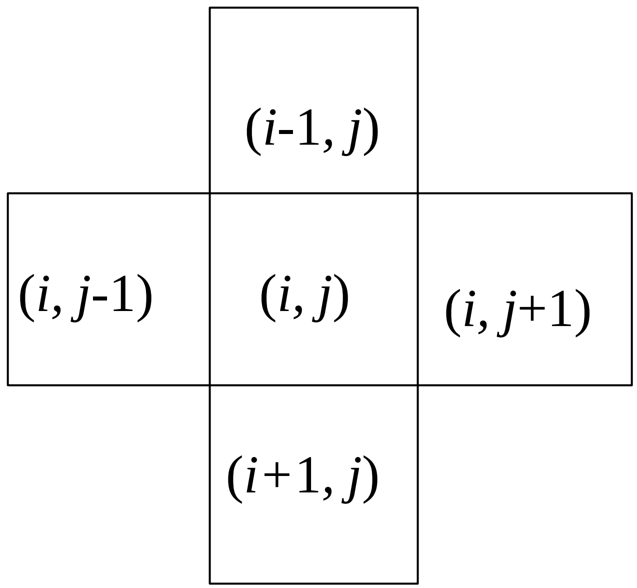


Κανόνες ανανέωσης

- Εάν το κελί είναι κενό,παραμένει κενό για $t + 1$
- Εάν το κελί περιέχει καιγόμενο δένδρο, το δένδρο εξαφανίζεται για $t + 1$
- Εάν το κελί περιέχει δένδρο
  - Και βρίσκεται δίπλα σε κελί με καιγόμενο δένδρο, τότε αναφλέγεται με πιθανότητα (1 – *prob_fire_resistance*)
- Ένα δένδρο μπορεί να αναφλεγεί από τυχαία γεγονότα (κεραυνός, ατύχημα, κλπ): *probRandomFire*

Εφαρμογή των κανόνων σε κάθε κελί

- ***i*** γραμμές; ***j*** στήλες
- Πρόβλημα: τι κάνουμε στα άκρα (boundaries)?

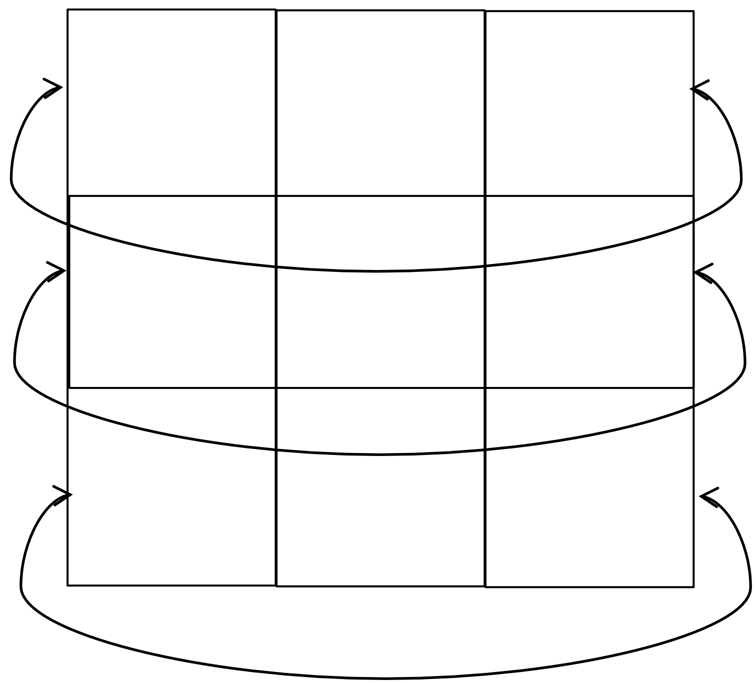


Χειρισμός οριακών συνθηκών

Κυψελίδες εκτός της περιμέτρου είναι ανενεργές (τίποτα δεν υπάρχει δυτικά της στήλης 1 και ανατολικά της N; βόρεια της γραμμής 1 και νότια της N)

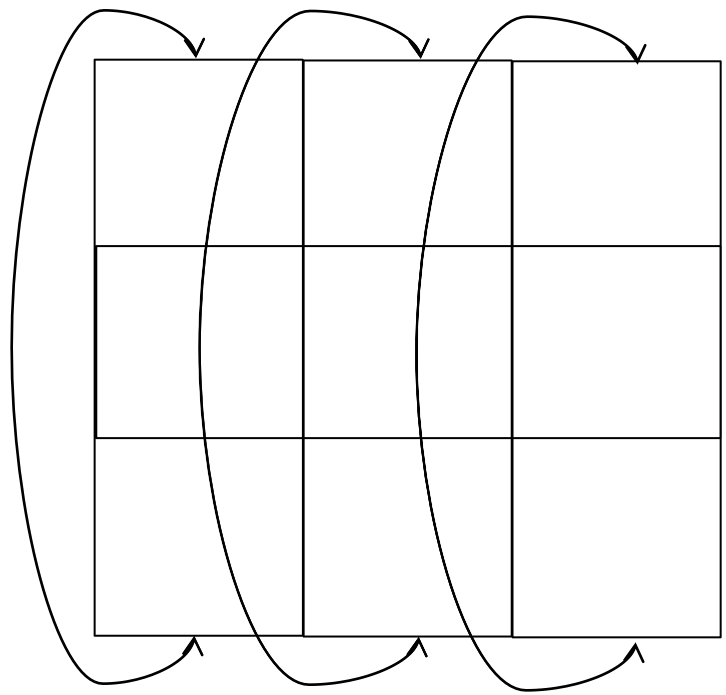
Periodic Boundary Conditions

Έτσι οι στήλες 1 και n γειτονεύουν!!



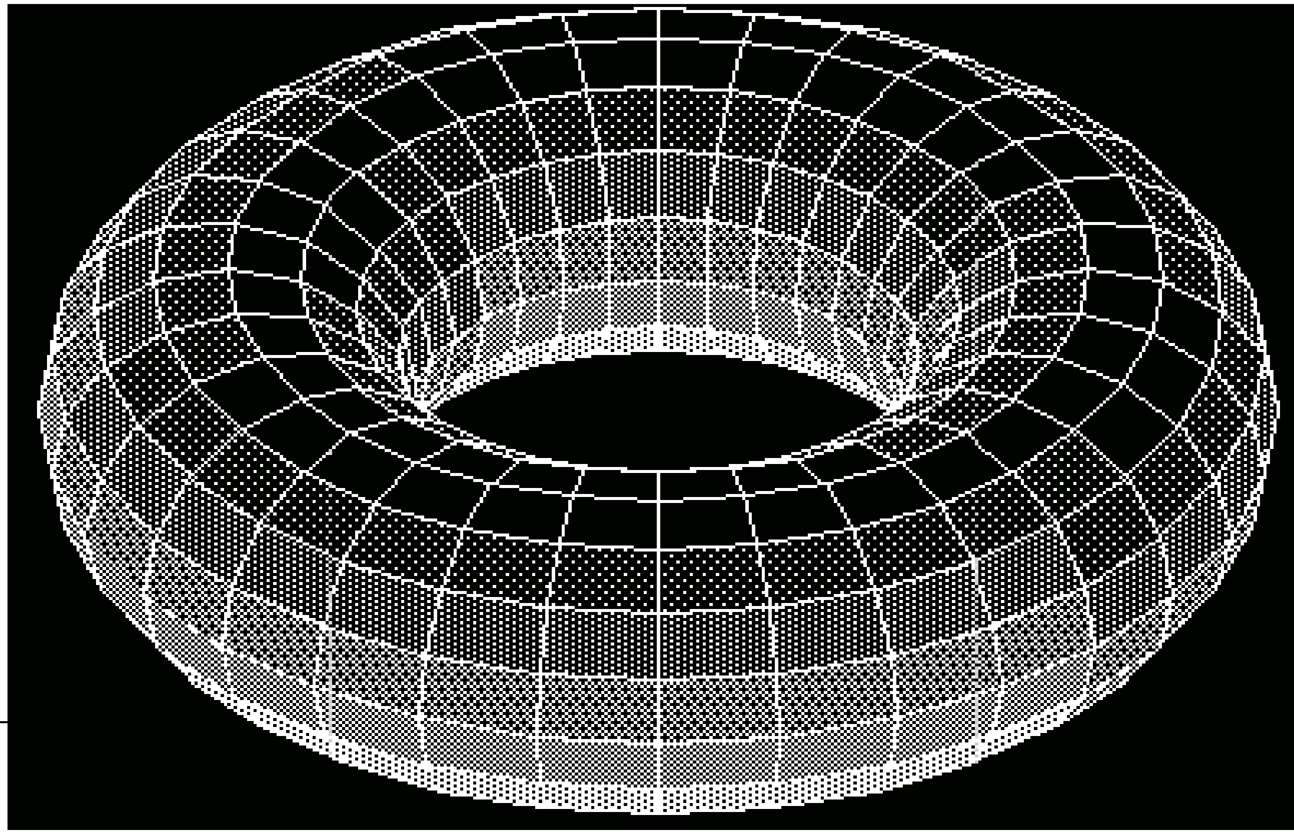
Periodic Boundary Conditions

Όμοια οι γραμμές 1 και n είναι “γείτονες”!



Periodic Boundary Conditions

Οπότε έτσι προκύπτει ότι ο «χώρος» μας είναι ένας τόρος!.

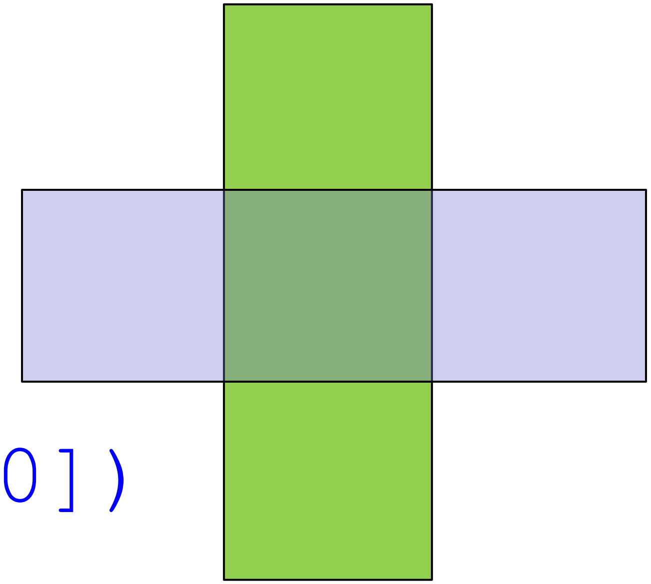


Periodic Boundary Conditions στο Matlab

Η συνάρτηση **`circshift`** το κάνει αυτό για εμάς!

```
>> a = [1 2 3; 4 5 6; 7 8 9]

a =

     1     2     3
     4     5     6
     7     8     9

>> circshift(a, [0 1]) % W neighbors

ans =

     3     1     2
     6     4     5
     9     7     8
```

Οπότε για κάθε κυψελίδα ισχύει ότι



- Η βόρεια κυψελίδα της είναι η `circshift(oldgrid,[1 0]`
- Η νότια κυψελίδα της είναι η `circshift(oldgrid, [-1,0])`
- Η ανατολική κυψελίδα της είναι η `circshift(oldgrid, [0 -1])`
- Η δυτική κυψελίδα της είναι η `circshift(oldgrid, [0 1])`

Μεταφράζοντας τους κανόνες σε Matlab

- Εάν το κελί είναι κενό,παραμένει κενό για $t+1$

```
% δεν χρειάζεται να κάνουμε τίποτα
```

Μεταφράζοντας τους κανόνες σε Matlab

- Εάν το κελί περιέχει καιγόμενο δένδρο, το δένδρο εξαφανίζεται $t+1$

```
newgrid(oldgrid == 2) = 0;
```

Μεταφράζοντας τους κανόνες σε Matlab

- Εάν το κελί περιέχει δένδρο
  - Και βρίσκεται δίπλα σε κελί με καιγόμενο δένδρο, τότε αναφλέγεται με πιθανότητα (1 – *prob_fire_resistance*)

```
n = length(oldgrid); % το «μήκος» μιας διάστασης
havetree = oldgrid == 1     ;
susceptible = rand(n) < (1 - prob_fire_resistance);
nfire = circshift(oldgrid, [1 0])  == 2;
sfire = circshift(oldgrid, [-1 0]) == 2;
efire = circshift(oldgrid, [0 -1]) == 2;
wfire = circshift(oldgrid, [0  1]) == 2;
adjacent = nfire | sfire | efire | wfire;
newgrid(havetree & adjacent & susceptible) = 2;
```

Μεταφράζοντας τους κανόνες σε Matlab

- Ένα δένδρο μπορεί να αναφλεγεί από τυχαία γεγονότα (κεραυνός, ατύχημα, κλπ): *probRandomFire*

```
newgrid(havetree & (rand(n) < probRandomFire)) = 2;
```

- Δοκιμάστε το δικό σας πρόγραμμα!

Συνοψίζοντας………

Α κύκλος

Στοχεύει

(α) στη γνωριμία με τις βασικές αλγοριθμικές διαδικασίες επίλυσης προβλημάτων και

(β) στην κατανόηση της λειτουργίας ενός προγράμματος και της "παραγωγής" αποτελεσμάτων με τη βοήθεια Η/Υ.

"Στο τέλος του πρώτου κύκλου θα μπορείτε να αναπτύσσετε μικρά προγράμματα και συναρτήσεις και θα γνωρίζετε τα περισσότερα στοιχεία σύνταξης του Matlab"

Α κύκλος

- Δομές
  - Ακολουθίας
  - Επανάληψης
  - Ελέγχου

Α κύκλος

- Συναρτήσεις
  - Ορίσματα εισόδου
  - Διάνυσμα εξόδου

Α κύκλος

- «Διανυσματικός» - αποδοτικός προγραμματισμός
  - Χρησιμοποιώ αποτέλεσμα λογικών ελέγχων πίνακα ως δείκτες για να επιδράσω επί συγκεκριμένων στοιχείων πίνακα
    - X(X>5)=5

Β κύκλος

Ανάλυση & αποτελεσματικότητα αλγορίθμων

"Στο τέλος του δεύτερου κύκλου θα είσαστε σε θέση να συγκρίνετε την αποτελεσματικότητα δύο αλγορίθμων και να σχεδιάσετε το δικό σας αλγόριθμο επίλυσης ενός προβλήματος".

Β κύκλος

- Αναδρομικές συναρτήσεις
  - Συναρτήσεις που καλούν & χρησιμοποιούν τον εαυτό τους
- Υπολογιστική πολυπλοκότητα
  - Στοιχειώδεις πράξεις
  - Υπολογιστικά βήματα
- Γραφικές παραστάσεις

Γ κύκλος

Προσομοιώσεις, μαθηματικές εφαρμογές Matlab

“Στο τέλος του τρίτου κύκλου θα είσαστε σε θέση να εφαρμόσετε τεχνικές προσομοίωσης για την επίλυση ενός πραγματικού προβλήματος, και να επιλύσετε βασικά προβλήματα με μαθηματικές διαδικασίες μέσω Matlab”.

Γ κύκλος

- Προσομοιώσεις
  – Τυχαίοι αριθμοί

Γ κύκλος

- Μαθηματικά κλπ με Matlab
  - Συμβολική & αναλυτική
    - Επίλυση εξισώσεων
    - Παραγώγιση
    - Ολοκλήρωση
  - Επίλυση συστημάτων
  - Διαφορικές

Γ κύκλος

- Simulink

Στόχοι μαθήματος

- Ανάλυση ενός προβλήματος και σχεδίαση μιας λύσης (αλγορίθμου).
- Ανάπτυξη δομημένων, αποδοτικών, κατανοητών και επαναχρησιμοποιήσιμων προγραμμάτων που επιλύουν συγκεκριμένα προβλήματα ή οικογένειες προβλημάτων.
- Κατανόηση πολύπλοκων προγραμμάτων.
- Εκτίμηση της αποδοτικότητας της λύσης που δημιουργήσατε (αλγορίθμου).

Το μάθημα της Πληροφορικής και ο διδάσκων θα είναι πάντα δίπλα σας…

Καλή συνέχεια στις σπουδές σας!

Σημείωμα Αναφοράς

- Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Κωνσταντίνος Καρατζάς. «Πληροφορική. Ενότητα 12: Υπολογιστική και Μηχανολογία: το “παιχνίδι της ζωής” και η προσομοίωση δομικών μεταβολών χάλυβα και δασικών πυρκαγιών». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: https://opencourses.auth.gr/courses/OCRS328/

Σημείωμα Αδειοδότησης

Τέλος ενότητας

Θεσσαλονίκη, Εαρινό Εξάμηνο 2014-2015

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.