ΕΘΝΙΚΟ
ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

4η Σειρά Ασκήσεων - Ελλειπτικές Εξισώσεις

Author:
Θωμόπουλος Σπυρίδων, ge19042

Υπολογιστική Ρευστομηχανική

January 4, 2024

Όλοι οι κώδικες που χρησιμοποιήθηκαν για την επίλυση των ασκήσεων βρίσκονται και στην σελίδα https://github.com/spyrosthomo/Physics/tree/main/Computational_Fluid/Sets/04

Προσπάθησα να λύσω τα ερωτήματα έτσι ώστε ο κώδικας να έχει την μεγαλύτερη δυνατή ευελιξία που μπορούσα να πετύχω, διότι ενδεχομένως πολλές από τις μεθόδους να χρειαστούν στην συνέχεια του μαθήματος.

Θεωρητικά Στοιχεία

Οι εξισώσεις Navier-Stokes για την ορμή, που περιγράφουν την ροή ασυμπίεστου ρευστού σταθερής πυκνότητας είναι

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\nabla)\boldsymbol{u} - \frac{\mu}{\rho_0}\nabla^2\boldsymbol{u} = -\nabla\left(\frac{p}{\rho_0}\right) + \boldsymbol{g} \tag{1}$$

Επίσης, η εξίσωση συνέχειας για ομογενές ρευστό μόνιμης ροής δίνει

$$\begin{aligned} \frac{\cancel{\partial \rho}}{\cancel{\partial t}} + \nabla(\rho\boldsymbol{u}) &= 0 \xRightarrow{\rho=\rho_0} \\ \nabla\rho_0\boldsymbol{u} + \rho_0\nabla\boldsymbol{u} &= 0 \Rightarrow \\ \nabla\boldsymbol{u} &= 0 \end{aligned} \tag{2}$$

Το πρόβλημα που πρέπει να λύσουμε είναι η πλήρως αναπτυγμένη ροή ενός ρευστού σε αγωγό τετραγωνικής διατομής με πλευρά α. Άρα αν προσανατολίσουμε τον άξονα του αγωγού να 'ναι παράλληλα με τον άξονα z, θα έχουμε ότι $u_y = u_x = 0$ και άρα από την (2) $\partial_z u_z = 0$, συνεπώς

$$\frac{\partial u_z}{\partial z} = 0 \Rightarrow u_z = u_z(x,y) \tag{3}$$

Βάσει αυτών, οι Navier-Stokes απλοποιούνται. Εκφράζοντας τους τελεστές σε καρτεσιανές συντεταγμένες και θεωρώντας πως η ροή είναι οριζόντια, στρωτή, μόνιμη (χωρίς χρονική εξάρτηση), πλήρως αναπτυγμενη ($\partial_z u_z = 0$) κατά την διεύθυνση $\hat{\boldsymbol{z}}$, παίρνουμε

$$\begin{aligned} \frac{\cancel{\partial \boldsymbol{u}}}{\cancel{\partial t}} + \left(\cancel{u_x}\partial_x + \cancel{u_y}\partial_y + \cancel{u_z\partial_z}\right)\boldsymbol{u} - \frac{\mu}{\rho_0}\left(\nabla^2\cancel{u_x}\hat{\boldsymbol{x}} + \nabla^2\cancel{u_y}\hat{\boldsymbol{y}} + \nabla^2 u_z\hat{\boldsymbol{z}}\right) &= -\frac{1}{\rho_0}\left(\partial_x p\hat{\boldsymbol{x}} + \partial_y p\hat{\boldsymbol{y}} + \partial_z p\hat{\boldsymbol{z}}\right) \xRightarrow{\partial_z u_z = 0} \\ -\frac{\mu}{\rho_0}\left(\partial_{xx} + \partial_{yy}\right)u_z(x,y)\hat{\boldsymbol{z}} &= -\frac{1}{\rho_0}\left(\partial_x p\hat{\boldsymbol{x}} + \partial_y p\hat{\boldsymbol{y}} + \partial_z p\hat{\boldsymbol{z}}\right) \end{aligned}$$

Άρα στις τρεις διευθύνσεις γίνεται

$$\begin{cases} \hat{\boldsymbol{x}}: & \frac{\partial p}{\partial x} = 0 \\ \hat{\boldsymbol{y}}: & \frac{\partial p}{\partial y} = 0 \\ \hat{\boldsymbol{z}}: & \mu\left(\frac{\partial^2 u_z(x,y)}{\partial x^2} + \frac{\partial^2 u_z(x,y)}{\partial y^2}\right) - \frac{\partial p}{\partial z} = 0 \end{cases} \tag{4}$$

Εν τέλει, πρέπει να λύσουμε μία συνήθη διαφορική εξίσωση

$$\frac{\partial^2 u_z(x,y)}{\partial x^2} + \frac{\partial^2 u_z(x,y)}{\partial y^2} - \frac{1}{\mu}\frac{\partial p}{\partial z} = 0 \tag{5}$$

Με τις Dirichlet συνοριακές συνθήκες

$$u_z(0,y) = 0 \tag{6}$$
$$u_z(a,y) = 0 \tag{7}$$
$$u_z(x,0) = 0 \tag{8}$$
$$u_z(x,a) = 0 \tag{9}$$

Απομένει ακόμη να αδιαστατοποιήσουμε τις εξισώσεις μας. Θεωρούμε αυθαίρετα ως U μια χαρακτηριστική ταχύτητα της ροής και θέτουμε

$$\begin{aligned} x^* &= x/a \\ y^* &= y/a \\ u_z^* &= u_z/U \end{aligned}$$

Η εξίσωση με τις συνοριακές, αν επιλέξουμε $U = +a^2/(\mu\partial_z p)$ γίνονται

$$\frac{\partial^2 u_z^*(x^*, y^*)}{\partial^2 x^*} + \frac{\partial^2 u_z^*(x^*, y^*)}{\partial^2 y^*} - 1 = 0 \tag{10}$$

$$u_z^*(0, y^*) = 0 \tag{11}$$

$$u_z^*(1, y^*) = 0 \tag{12}$$

$$u_z^*(x^*, 0) = 0 \tag{13}$$

$$u_z^*(x^*, 1) = 0 \tag{14}$$

Διακριτοποίηση

Εφόσον οι άλλες συνιστώσες δεν παίζουν κάποιον ρόλο, από δω και πέρα για ευκολία θα γράφουμε $u_z^*(x^*, y^*) \equiv u$ και $x^* \equiv r, y^* \equiv y$.

Ξεκινάμε με την διακριτοποίηση του πεδίου $[x, y] \in [0, 1] \times [0, 1]$, σε ίσο αριθμό σημείων N σε κάθε διεύθυνση εφόσουν δεν υπάρχει κάποιος συγκεκριμένος λόγος (π.χ. παραλληλόγραμμη διατομή ροής) με απόσταση $\Delta x = \Delta y = h = 1/(N)$

$$x_i = y_i = \frac{i}{h},\ i = 0, \cdots, N \tag{15}$$

έτσι ώστε $r_1 = 0$ και $r_N = 1$. Τώρα διακριτοποιούμε τις παραγώγους χρησιμοποιώντας κεντρικές πεπερασμένες διαφορές, για $i = 1, \cdots, N - 1$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \tag{16}$$

$$\frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \tag{17}$$

$$\tag{18}$$

και αντικαθιστούμε στην (10), $i = 2, \cdots, N - 1$

$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + h^2 = 0 \tag{19}$$

και με τις συνοριακές συνθήκες

$$\begin{cases} \{i, j\} = 0: & u_{0,j} = u_{i,0} = 0 \\ \{i, j\} = N: & u_{N,j} = u_{j,N} = 0 \end{cases} \tag{20}$$

παίρνουμε ένα γραμμικό σύστημα $(N - 1)^2$ αλγεβρικών εξισώσεων με $(N - 1)^2$ αγνώστους. Άρα πλέον μπορούμε να εφαρμόσουμε τις μεθόδους που γνωρίζουμε για την επίλυσή του.

Το μόνο βήμα που απομένει είναι να γράφουμε το σύστημα με τρόπο που να 'ναι βολικότερος για πράξεις με πίνακες. Συγκεκριμένα θεωρούμε

$$\begin{cases} P_l = & (x_i, y_j) \\ w_l = & u_{i,j}, \qquad l = (i - 1)N + j \end{cases} \tag{21}$$

για $i = 1, \ldots, N - 1$ and $j = 1, \ldots, N - 1$. Αυτός ο τρόπος γραφής δίνει τα εσωτερικά σημεία του πλέγματος απ' τ' αριστερά στα δεξιά και από πάνω προς τα κάτω. Κατ' αυτόν τον τρόπο η εξίσωση πεπερασμένων διαφορών (19) γίνεται

$$4w_l - w_{l+N} - w_{l-N} - w_{l+1} - w_{l-1} + h^2 = 0, \qquad l = (i - 1)N + j \tag{22}$$

Τώρα είναι εκπεφρασμένη σε μορφή γραμμικού συστήματος και μπορεί να λυθεί εύκολα με οποιαδήποτε απ' τις μεθόδους που έχουν αναπτυχθεί. Για ευκολία, θα το λύσουμε απλώς με τον τελεστή του Octave "\".

Επίσης, η θεωρητική σχέση για την παροχή είναι

$$Q = -0.035144 \frac{a^4}{\mu} \frac{\mathrm{d}p}{\mathrm{d}z} \xRightarrow{E = a^2, U = a^2 \cdot (\mu\partial_z p)^{-1}}$$

$$\frac{Q}{E \cdot U} = -0.035144 \equiv \bar{u_z^*} \tag{23}$$

Ενώ υπολογιστικά η μέση αδιαστατοποιημένη ταχύτητα υπολογίζεται από την ολοκλήρωση

$$\bar{u_{z}}_{num} \equiv \frac{Q}{E} = \frac{\int u_z dE}{E} \Rightarrow$$
$$\bar{u}^*_{znum} \equiv \frac{\bar{u_z}_{num}}{U} = \frac{Q}{EU} = \frac{\int_0^1 \int_0^1 u_z^*(x^*, y^*) dx^* dy^*}{E^*}$$

Κώδικες - Αποτελέσματα

Υλοποιούμε τα παραπάνω στον παρακάτω κώδικα MATLAB. Το κυρίως πρόγραμμα που είναι στο αρχείο *EX_elliptic.m* καλεί την συνάρτηση *NS_elliptic.m* η οποία κατασκευάζει το γραμμικό σύστημα που προκύπτει από τις εξισώσεις Navier-Stokes με τις συνθήκες που απαιτεί η άσκηση. Το σύστημα επιλύεται, όπως έχει αναφερθεί, με τον τελεστή του Octave "\". Σε ξεχωριστά αρχεία, για την εύκολη μεταποίηση του προβλήματος, παρέχεται η συνάρτηση του μη ομογενούς όρου, καθώς και η συνάρτηση που απεικονίζει τα i, j σε έναν ακέραιο l και μέσω αυτής μπορούμε να καταστρώσουμε το γραμμικό σύστημά μας. Παρέχονται κατ' αντιστοιχία στα αρχεία *f1.m* και *lambda.m*.

Listing 1: /Sets/04/*Ex_elliptic.m*

```
clear; format compact
addpath("../../Linear_Systems")
%% Number of points
N       = [10 30 50 70 90 110] ;%, 10, 50, 100, 10^3];
sols    = length(N);
%-----------------------------------------------------------
%% Find & plot solutions
ms = 2; lw1 = 1.5; lw2 = 1;
maxValue = []; maxRow = []; maxCol = []; maxIndex = [];
umean   = [];
emean   = [];
for i = 1:sols
  printf(" ---------------- N = %d ----------------\n", N(i))
  h         = 1/(N(i)+1);
  x         = ((0:N(i)+1)*h)';
  y         = ((0:N(i)+1)*h)';
  [X, Y]    = meshgrid(x, y);
  % construct & find solutions
  [A, b]    = NS_elliptic(N(i), h, @f1);
  x0        = (1:N(i)*N(i)).^2';
  w         = A\b;
  %w         = Jacobi(A, b, x0, 1e-5, 2800);
  u         = reshape(w, N(i) , N(i));
  emean(i)  = trapz(trapz(X, 2), 1);
  umean(i)  = 0.5*trapz(w)/emean(i);
  %------------------------------------------------
  %% plot Solution
  % 3D-plot
  figure;
  leg = {};
  mesh(X(2:end-1, 2:end-1), Y(2:end-1, 2:end-1) , u)%, "-", "markersize", ms, "linewidth", lw1)
  hold on
  leg = sprintf("N=%d", N(i));
  legend(leg, 'location','northwest', "FontSize", 14)
  xlabel("x^*=x/a")
  ylabel("y^*=y/a")
  zlabel("u_z^*(x^*, y^*)=u_z(x, y)/U")
  title(sprintf("Solution of NS equations in square pipe for N=%d\n",N(i)), 'FontSize', 14)
  grid on
  saveas(gcf, sprintf('./plots/NS_elliptic_3sol%d.jpg', N(i)));
```

```
  hold off
  %----------------------
  % contour
  figure;
  contour(X(2:end-1, 2:end-1), Y(2:end-1, 2:end-1) , u)
  hold on
  xlabel("x^*=x/a")
  ylabel("y^*=y/a")
  title(sprintf("Contour plot for NS solution in square pipe, N=%d\n", N(i)),...
          'FontSize', 14, 'Position', [0.5, 1.05, 0])
  grid on
  % Find the maximum value and its indices
  [maxValue(i), maxIndex(i)] = max(u(:));
  [maxRow(i), maxCol(i)]     = ind2sub(size(u), maxIndex(i));
  % Add a red dot at the maximum point
  hold on
  plot(X(maxRow(i), maxCol(i)), Y(maxRow(i), maxCol(i)), 'ro', 'MarkerSize', 10, 'MarkerFaceColor', 'red');
  % Display the maximum value near the contour line
  text(X(maxRow(i), maxCol(i)), Y(maxRow(i), maxCol(i)), sprintf('Max: %.4f', maxValue(i)), ...
      'HorizontalAlignment', 'left', 'VerticalAlignment', 'bottom', 'Color', 'red', 'FontSize', 10);
  colorbar('location', 'eastoutside', 'fontsize', 10);
  fn = sprintf('./plots/NS_elliptic_Contour_sol%d.jpg', N(i));
  saveas(gcf, fn);
  hold off
  %printf("u*_max = %.8f\n", max(abs(u)))
  % sprintf("<u*>   = %.8f\n", 2*trapz(r, abs(r.*u)))
endfor

%% plot max movement
figure;
%suptitle(sprintf("Movement of maximum velocity with varying number mesh points N"))
colors = jet(length(N));

% in x direction
subplot(2, 2, 1)
for i=1:length(N)
  scatter(X(maxRow, maxCol)(1,i), maxValue(i), 30, colors(i,:), 'filled'   );
  hold on;
endfor
xlabel(sprintf("x^*=x/a"))
ylabel(sprintf("max(u)"))
%legend('location','northwest', "FontSize", 14)

%----
% in y direction
subplot(2, 2, 2)
for i=1:length(N)
  scatter(Y(maxRow, maxCol)(i,1), maxValue(i), 30, colors(i,:), 'filled'   );
  hold on;
endfor
xlabel(sprintf("y^*=y/a"))
ylabel(sprintf("max(u") )
```

```
%legend('location','northwest', "FontSize", 14)

% x-y
subplot(2, 2, 3:4)
for i=1:length(N)
  scatter(X(maxRow, maxCol)(1,i), Y(maxRow, maxCol)(i,1), 50, colors(i,:), '
      filled', ...
       'DisplayName', sprintf('N=%d',N(i)));
  hold on;
endfor
legend('location','eastoutside', "FontSize", 14)
ylim([0, 0.6])
xlim([0, 0.6])
line([0.5, 0.5], [0, 0.5], "linestyle", "-", "color", "black", '
    HandleVisibility', 'off');
line([0, 0.5], [0.5, 0.5], "linestyle", "-", "color", "black", '
    HandleVisibility', 'off');
xlabel(sprintf("x^*=x/a"))
ylabel(sprintf("y^*=y/a"))
axes( 'visible', 'off', 'title',"Movement of velocity maximum with varying
    mesh points N" );
saveas(gcf, sprintf("./plots/NS_elliptic_max_movement.jpg"));


%-----------------------------------
% figure umean
figure;
for i=1:length(N)
  scatter(i, umean(i), 30, 'filled', 'DisplayName', sprintf("N=%d",N(i)))
  hold on;
endfor
ylabel(sprintf("<u^*_z>"))
legend('location', 'eastoutside', 'FontSize', 14)
title("Mean velocity value for varying N")
saveas(gcf, sprintf("./plots/NS_elliptic_umean.jpg"))
```

Listing 2: /Sets/04/*NS_elliptic.m*

```
%% Construction of the Navier-Stokes equation matric in the case of
%          - Square pipe
%          - stationary flow
%          - incompressible fluid
%          - //z flow
%-------------------------------------------------------
function [A, b] = NS_elliptic(N, h, f)%, method)
  %% Numerical scheme's parameters
  x   = ((0:N+1)*h)';
  %x(1:N) = ((0:N-1)*h)';
  %% Define the system's matrices A*w = b: eqns (19), (21) in the text
  M = (N)*(N)        % dim of solution
  A = zeros(M, M);
  b = zeros(M, 1);
  %------------------------------
  %------------------------------
  %% construct A, b
  b = h^2*f(ones(N+2, N+2), h);
  b = reshape(b(2:end-1, 2:end-1), [], 1);
  % boundary conditions
  % y
```

```
  for i=1:N
    l1 = lambda(i, 1, N);
    lN = lambda(i, N, N);
      A((i-1)*N+1, l1) = 1;
      A((i-1)*N+N, lN) = 1;
  endfor
  % x
  for j=1:N
    l1 = lambda(1, j, N);
    lN = lambda(N, j, N);
      A(l1, j         ) = 1;
      A(lN, N*(N-1)+j) = 1;
  endfor
  % inner points
  for i=2:N-1
    %sprintf("#---------- i = %.d ---------------", i)
    for j=2:N-1
      %sprintf("#---------- j = %.d ---------------", j)
      l = lambda(i, j, N);
      if  ( l>=1)% & l<=M )       %( (l>1) & (l<M) )
        A(l, l )  = +4;
        A(l, l-1) = -1;
        A(l, l+1) = -1;
        A(l, l-N) = -1;
        A(l, l+N) = -1;
      endif

   endfor
  endfor
endfunction
```

Listing 3: /Sets/04/$f1.m$

```
function f = f1(x, h)
    f = ones(size(x));
end
```

Listing 4: /Sets/04/$lambda.m$

```
function l = lambda(i, j, N)
    %l =  i + (N-1-j)*(N-1);
    l  = (i-1) * N + j ;
endfunction
```

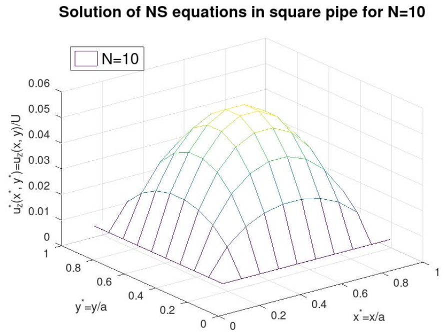
Χρησιμοποιούμε διάφορους αριθμούς πλεγματικών σημείων και συγκεκριμένα $N = \{10, 30, 50, 70, 90, 110\}$. Βλέπουμε πως η εξομαλύται η λύση για αυξανόμενο αριθμό σημείων, όπως αναμένουμε. Στις Εικόνες (1) παρατηρούμε ακριβώς αυτό στις λύσεις με $N = 10, 110$.

Ακόμη, στην Εικόνα (2) έχω σχεδιάσει τον τρόπο που κινείται το μέγιστο της κατανομής της ταχύτητας. Γνωρίζουμε απ'την θεωρία πως η ταχύητα θα πρέπει να λαμβάνει μέγιστο στο κέντρο του αγωγού, δηλαδή στο σημείο $(x^*, y^*) = (0.5, 0.5)$. Βλέπουμε πως καθώς τα πλεγματικά σημεία αυξάνονται, το εν λόγω μέγιστο συγκλίνει προς αυτό το σημείο και επίσης συγκλίνει σε μία συγκεκριμένη τιμή που είναι ίση με $max(u_z^*) \simeq 0.71$, δηλαδή $max(u_z) \simeq 0.71 \cdot U$, όπως φαίνεται στον Πίνακα (1)

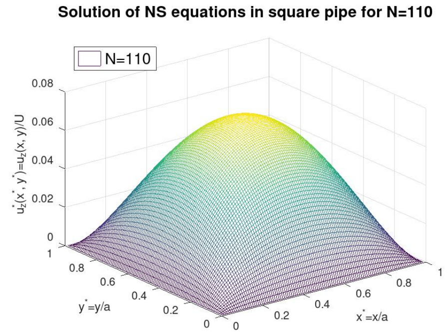
N	10	30	50	70	90	110
$max(u_z^*)$	0.056090	0.065322	0.068321	0.069741	0.070567	0.071107

Πίνακας. 1: Τιμές του μεγίστου της ταχύητας για αυξανόμενο αριθμό πλεγματικών σημείων.

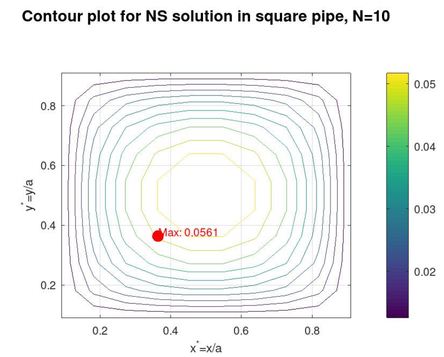
Τέλος, στην Εικόνα (3) και στον αντίστοιχο Πίνακα (2) βλέπουμε την σύγκλιση της μέσης τιμής της ταχύτητας σε μία τομή του τετραγωνικού αγωγού.



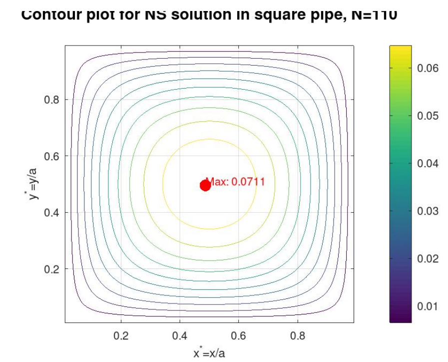
(a) Κατανομή της $u_z(x,y)$ για $N = 10$.



(b) Κατανομή της $u_z(x,y)$ για $N = 110$.

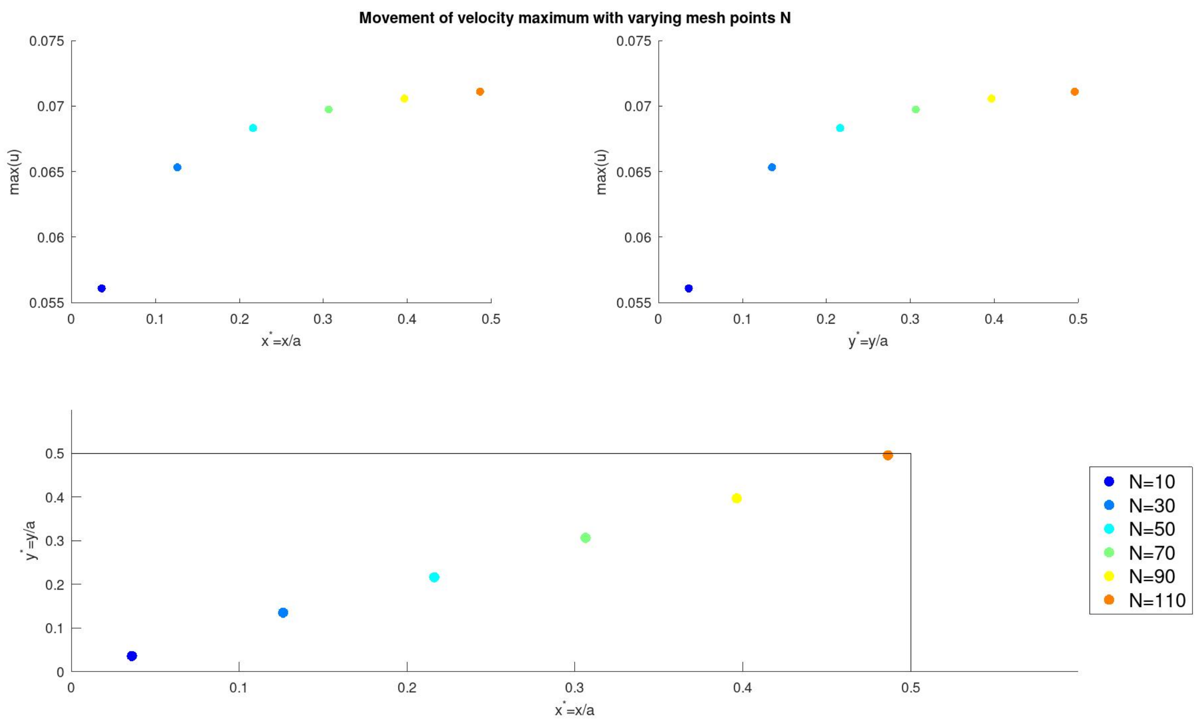


(c) 2D ισοσταθμικές καμπύλες για $N = 10$.

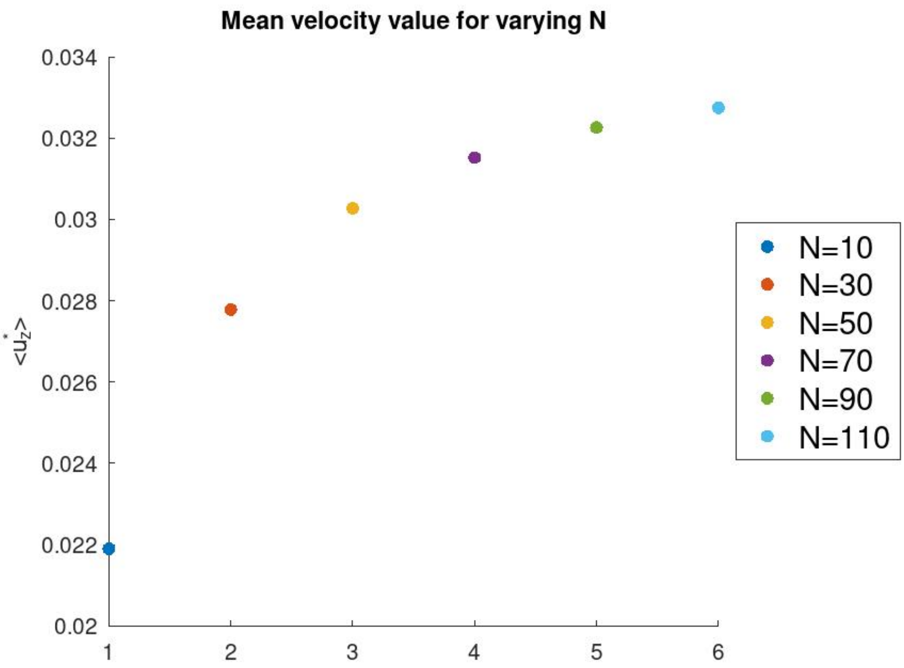


(d) 2D ισοσταθμικές καμπύλες για $N = 110$.

Εικόνα. 1: Λύση της εξίσωσης Poisson για $N = 10, 110$ με συνοριακές συνθήκες τετραγωνικού αγωγού, δηλαδή μηδενισμού της ταχύτητας στα άκρα του.



Εικόνα. 2: Σύγκλιση του μεγίστου της ταχύτητας σε μία τιμή και σε συγκεκριμένο σημείο, το κέντρο του αγωγού.



Εικόνα. 3: Παράσταση της μέσης τιμής της ταχύτητας συναρτήσει του αριθμού των πλεγματικών σημείων $N = \{10, 30, 50, 70, 90, 110\}$.

N	10	30	50	70	90	110
$\bar{u_z^*}$	0.021897	0.027785	0.030276	0.031520	0.032260	0.032750

Πίνακας. 2: Μέση τιμή της ταχύτητας για αυξανόμενο πλεγματικό αριθμό