Εργασία Υπολογιστικές Μέθοδοι 2025
Κατέβας Χρήστος
p300320

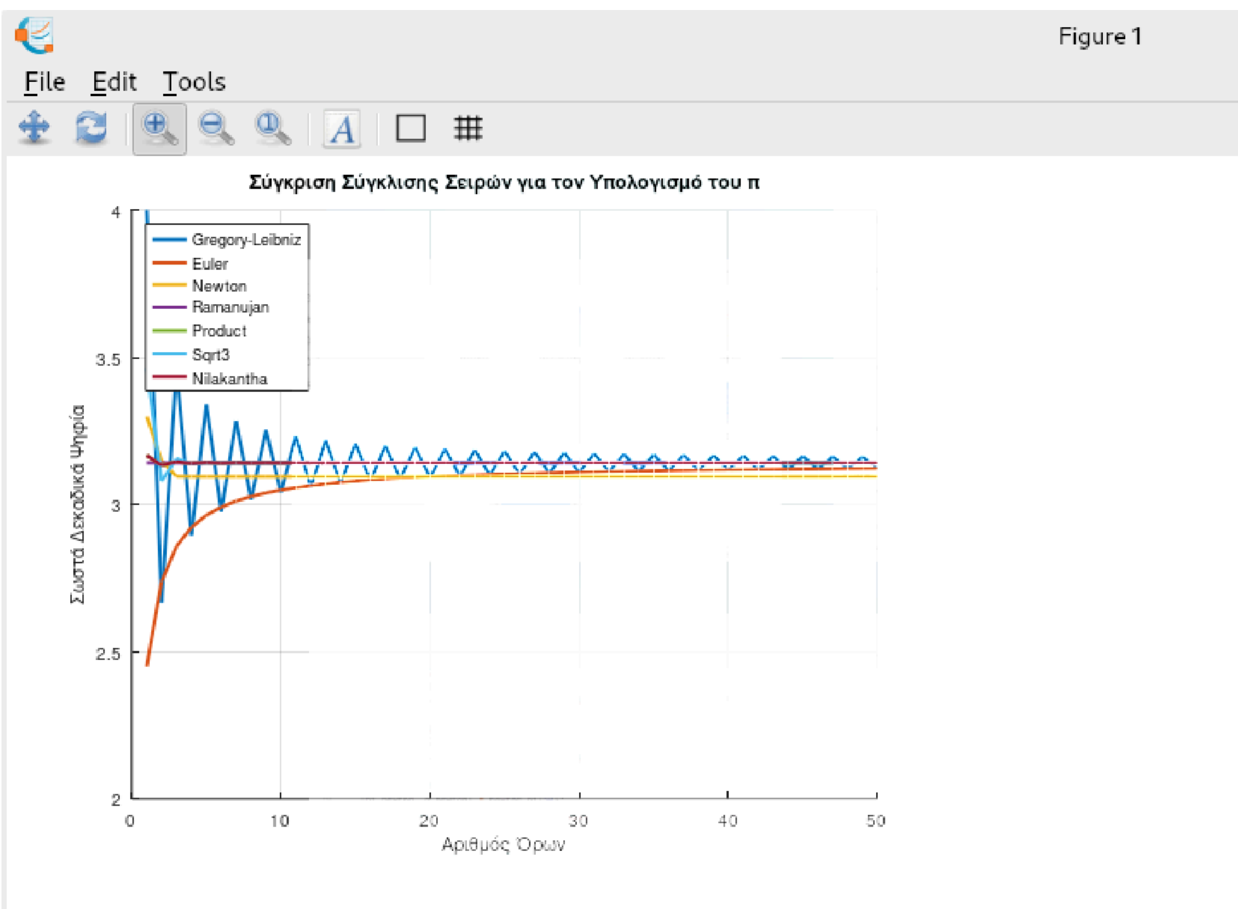
Υπολογισμός αριθμού π

Η παρούσα εργασία στοχεύει στον υπολογισμό του αριθμού π μέσω γνωστών άπειρων σειρών που έχουν προταθεί από μαθηματικούς όπως ο Leibniz, ο Newton, ο Euler, ο Ramanujan και άλλοι. Για κάθε μία από αυτές τις σειρές υλοποιήθηκαν αντίστοιχες συναρτήσεις σε περιβάλλον MATLAB, με σκοπό να εξεταστεί η ακρίβεια της προσέγγισης του π και η ταχύτητα σύγκλισης κάθε μεθόδου.

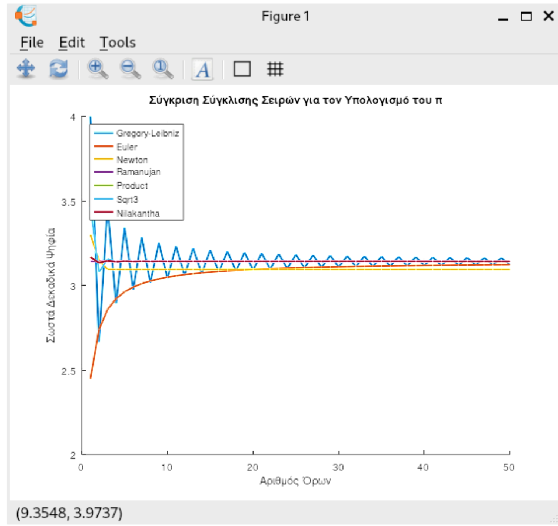
Για κάθε μέθοδο έγινε ο υπολογισμός:
- Της προσέγγισης του π μετά από έναν αριθμό όρων.
- Τον αριθμό των σωστών δεκαδικών ψηφίων σε σχέση με την πραγματική τιμή του π.
- Τη γραφική σύγκριση της σύγκλισης όλων των μεθόδων για έως 50 όρους.

Παρατηρώντας τα αποτελέσματα και το γράφημα διαπιστώνεται ότι ορισμένες μέθοδοι, όπως αυτή του Ramanujan και του Newton, παρέχουν εξαιρετικά ακριβή προσέγγιση με λίγους όρους, σε αντίθεση με μεθόδους όπως του Leibniz που συγκλίνουν πολύ αργά.

(Το πρόγραμμα εκτελέστηκε με Octave σε Debian 12)



```
--- Gregory-Leibniz ---
Όροι: 1000, Προσέγγιση: 3.140592653839794, Σωστά δεκαδικά: 3
--- Euler ---
Όροι: 10000, Προσέγγιση: 3.141497163947215, Σωστά δεκαδικά: 4
--- Ramanujan ---
Όροι: 2, Προσέγγιση: 3.141592653589793, Σωστά δεκαδικά: Inf
--- Product Series ---
Όροι: 100, Προσέγγιση: 3.141592410971982, Σωστά δεκαδικά: 6
--- Sqrt(3) Series ---
Όροι: 100, Προσέγγιση: 3.141592653589794, Σωστά δεκαδικά: 15
--- Nilakantha ---
Όροι: 100, Προσέγγιση: 3.141592410971982, Σωστά δεκαδικά: 6
--- Newton Series ---
Όροι: 22, Προσέγγιση: 3.094717668363173, Σωστά δεκαδικά: 1
\nΠάτησε Enter για έξοδο...
```




```
function main
    clc; clear;
    disp('--- Gregory-Leibniz ---')
    [pi_greg, d_greg] = gregory_leibniz(1000);
    fprintf('Όροι: 1000, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_greg, d_greg);

    disp('--- Euler ---')
    [pi_euler, d_euler] = euler_pi(10000);
    fprintf('Όροι: 10000, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_euler, d_euler);

    disp('--- Ramanujan ---')
    [pi_raman, d_raman] = ramanujan_pi(3);
    fprintf('Όροι: 2, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_raman, d_raman);

    disp('--- Product Series ---')
    [pi_prod, d_prod] = product_series(100);
```

```
    fprintf('Όροι: 100, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_prod, d_prod);

    disp('--- Sqrt(3) Series ---')
    [pi_sqrt3, d_sqrt3] = sqrt3_series(100);
    fprintf('Όροι: 100, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_sqrt3, d_sqrt3);

    disp('--- Nilakantha ---')
    [pi_nil, d_nil] = nilakantha(100);
    fprintf('Όροι: 100, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_nil, d_nil);

    disp('--- Newton Series ---')
    [pi_newton, d_newton] = newton_pi(50);
    fprintf('Όροι: 22, Προσέγγιση: %.15f, Σωστά δεκαδικά: %d\n', pi_newton, d_newton);

    plot_convergence();
    input('\nΠάτησε Enter για έξοδο...');
end

function [pi_approx, digits] = gregory_leibniz(N)
    pi_approx = 0;
    for k = 0:N-1
        pi_approx = pi_approx + ((-1)^k)/(2*k + 1);
    end
    pi_approx = 4 * pi_approx;
    digits = floor(-log10(abs(pi_approx - pi)));
end
```

```
function [pi_approx, digits] = euler_pi(N)
    sum_val = 0;
    for n = 1:N
        sum_val = sum_val + 1/n^2;
    end
    pi_approx = sqrt(6 * sum_val);
    digits = floor(-log10(abs(pi_approx - pi)));
end

function [pi_approx, digits] = ramanujan_pi(N)
    sum_val = 0;
    for n = 0:N-1
        numerator = factorial(4*n) * (1103 + 26390*n);
        denominator = (factorial(n))^4 * 396^(4*n);
        sum_val = sum_val + numerator/denominator;
    end
    pi_approx = 1 / (2*sqrt(2)/9801 * sum_val);
    digits = floor(-log10(abs(pi_approx - pi)));
end

function [pi_approx, digits] = product_series(N)
    pi_approx = 3;
    for n = 1:N
        denominator = 2*n * (2*n+1) * (2*n+2);
        term = 4 / denominator;
        if mod(n,2) == 1
            pi_approx = pi_approx + term;
```

```
        else
            pi_approx = pi_approx - term;
        end
    end
    digits = floor(-log10(abs(pi_approx - pi)));
end

function [pi_approx, digits] = sqrt3_series(N)
    sum_val = 0;
    for n = 0:N-1
        term = (-1)^n / (3^n * (2*n + 1));
        sum_val = sum_val + term;
    end
    pi_approx = 6 * sum_val / sqrt(3);
    digits = floor(-log10(abs(pi_approx - pi)));
end

function [pi_approx, digits] = nilakantha(N)
    pi_approx = 3;
    for n = 1:N
        term = 4 / ((2*n)*(2*n+1)*(2*n+2));
        if mod(n,2) == 1
            pi_approx = pi_approx + term;
        else
            pi_approx = pi_approx - term;
        end
    end
```

```
    digits = floor(-log10(abs(pi_approx - pi)));
end


function [pi_approx, digits] = newton_pi(N)
    x = 1/4;
    sum_val = 0;
    for k = 0:N-1
        switch k
            case 0
                coeff = 2/3;
            case 1
                coeff = -1/5;
            case 2
                coeff = -2/7;
            case 3
                coeff = -1/16 * 2/9;
            case 4
                coeff = -5/128 * 2/11;
            case 5
                coeff = -7/256 * 2/13;
            case 6
                coeff = -21/1024 * 2/15;
            case 7
                coeff = -33/2048 * 2/17;
            case 8
                coeff = -429/32768 * 2/19;
            otherwise
```

```
            coeff = 0;
        end
        sum_val = sum_val + coeff * x^(k + 1.5);
    end
    pi_approx = 3*sqrt(3)/4 + 24 * sum_val;
    digits = floor(-log10(abs(pi_approx - pi)));
end




function plot_convergence()
    max_terms = 50;
    methods = {@gregory_leibniz, @euler_pi, @newton_pi, @ramanujan_pi, @product_series,
@sqrt3_series, @nilakantha};
    names = {'Gregory-Leibniz', 'Euler', 'Newton', 'Ramanujan', 'Product', 'Sqrt3', 'Nilakantha'};

    figure;
    hold on;
    for i = 1:length(methods)
        terms = 1:max_terms;
        digits = arrayfun(@(n) methods{i}(n), terms);
        plot(terms, digits, 'LineWidth', 1.5);
    end
    legend(names, 'Location', 'northwest');
    xlabel('Αριθμός Όρων');
    ylabel('Σωστά Δεκαδικά Ψηφία');
    title('Σύγκριση Σύγκλισης Σειρών για τον Υπολογισμό του π');
    grid on;
```

end