Змiст

[1.Вступ 2](#_Toc532848185)

[2.Постановка задачі 4](#_Toc532848186)

[2.1 Фізична постановка задачі 4](#_Toc532848187)

[2.2 Математична постановка задачі 6](#_Toc532848188)

[3. Чисельний розв’язок задачі 11](#_Toc532848189)

[4. Результати 13](#_Toc532848190)

[5. Висновки 16](#_Toc532848191)

[6. Список використаної літератури 17](#_Toc532848192)

[7. Лістинг 18](#_Toc532848193)

# 1.Вступ

Проблеми стійкості однорідних і шаруватих необмежених тіл з дефектами їх структури і без них активно вивчалися в останні десятиліття. Часто в однорідних і шаруватих структурах, що працюють в умовах стиснення, наприклад, із-за недосконалості технологічних процесів, виникають дефекти у вигляді тріщин. В даній роботі розглянуто задачі механіки руйнування матеріалів при стиску вздовж тріщини з використанням критеріїв руйнування, побудованих на основі тривимірної лiнеаризованої теорії стійкості деформівних тіл, за умови, що руйнування реалізується у вигляді втрати стійкості стану рівноваги матеріалів біля тріщин. Моментом втрати стійкості вважається момент відкриття тріщини.

В даній дипломній роботі була досліджена втрата стійкості біматеріального тіла з міжфазним круговим розрізом при умові, що шар та пiвпростiр ідеально зчеплені між собою, а також при наявності тріщини в зоні зчеплення.

За допомогою інтегрального перетворення Ханкеля задача зведена до системи сингулярних інтегральних рівнянь з ядрами типу Коші. За допомогою квадратурної формули Гауса-Чебишева система сингулярних інтегральних рівнянь зведена до системи однорідних алгебраїчних рівнянь. Розв’язок задачі знаходиться з умови існування ненульового рішення цієї системи.

Дипломна робота складається з вступу, чотирьох параграфів, висновків, списку використаної літератури та лістингу програми.

В другому параграфі розглядається об’єкт дослідження. Формується математична модель та наведений разв’язок рівнянь стійкості.

В третьому параграфі записано розв’зок системи лінеаризованих рівнянь стійкості та граничні умови для циліндричного двошарового тіла, а також система сингулярних інтегральних рівнянь для даного випадку, записані додаткові умови для системи та представлення невідомих функцій.

В четвертому параграфі викладена методика чисельного розв’язку системи сингулярних інтегральних рівнянь з ядром типу Коші, яка використовує квадратурну формулу Гауса-Чебишева.

Основна задача дипломної роботи – ознайомлення з алгоритмом розв’язання подібних задач та створення програми на ПК для разгляненої задаі. В дипломній роботі наведені основні етапи розв’язку задачі.

# 2.Постановка задачі

## 2.1 Фізична постановка задачі

Розглядається втрата стійкості циліндричного біматеріального тіла, яке складається з ізотропного півпростору та ізотропного шару, стиснутих між двома абсолютно жорсткими гладкими плитами. Перетин тіла в площині *x0y* представлено на (Рис.1).

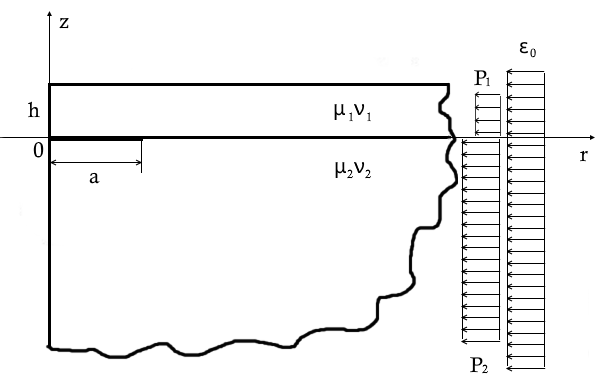


Рис.1

Передбачається, що має місце плоска деформаціяциліндрічного біматеріального тіла , ,в площині перпендикулярній осі r з дископодібною тріщиною , , в області розділу матеріалів. Конструкція піддається стискаючим навантаженням паралельним до вільної поверхні і яка призводить до однорідної поверхні та до однорідної деформації . Подальше збільшеннястискаючогонавантаженнядосягаєдеякого критичного значення, коли стаєможливийдодатковийрівноважний стан, якийхарактеризуєтьсятим, що до початкового однорідногодеформованого стану додаються нескінченно малі прирости

Вважаємо, що для шару та півпростору задані модуль пружностi**та коефіцієнт Пуассона .

## 2.2 Математична постановка задачі

Рівняння рівноваги збуреного стану за відсутності масових сил можуть бути представлені такими чином:

 (2.1)

де  (2.2)

Індекс i = 1 відноситься дошару, i = 2 до півпростору.

Розв’язок системи диференціальних рівнянь (2.1) отримано за допомогою інтегральних перетворень Фур'є і має вигляд [3]:

 (2.3)

де 

Рівняння дляприростунавантажень має вигляд:

 (2.4)

де 

Запишемо граничні умови:

,  (2.5)

, ;

, ; (2.6)

, ; (2.7)

де функції параметра: *С*2 = *D*2 = 0.

*6*невідомих функцій параметра: *A*1, *B*1, *A*2, *B*2, *C*1, *D*1 можуть бути визначені з граничних умов. Для цього, підставляючи у вирази для приросту переміщень і напружень (3.4) граничні умови, отримаємо наступну систему рівнянь:

 (2.8)

Введемо невідомі функції за формулами:

 (2.9)

Згідно з граничними умовами (2.7), бачимо:

, для  (2.10)

Підставимо у (2.9) рівняння для приросту переміщень, з умовою (2.10) застосуємо до перетворених співвідношень зворотне перетворення Ханкеля.

Тоді отримаємо:

, (2.11)

де  (2.12) Тепер, розв’язуючи систему рівнянь (3.11) і (3.8), виразимо коефіцієнти *A*1, *B*1,  *A*2, *B*2, *C*2, *D*2 через невідомі функції  та :

 (2.13)

Позначимо:



де 

Також:

  (3.14)

Kij є відомими функціями стискаючих навантажень та параметра p.

Використовуя (2.13) та (2.12) зведемо систему рівнянь (2.8) до:



Пiсля алгебраїчних перетворень отримаємо систему повних еліптичних інтегральних рівнянь:

(2.15)

К та Е - повніеліптичні інтегральні рівняння першого та другого роду, поведінку яких розглядаємо при 

Також де залежать від констант матеріалу*μ*i,*ν*i, навантаження

та мають вигляд:

 (2.16)

де  (2.17)

# 3. Чисельний розв’язок задачі

Виконавшизамінузмінних ми приходимо до системи сингулярних інтегральних рівнянь [4]:

 (3.1)

де. (3.2)

Ці невідомі функції  представимо таким чином

 (3.3)

Додаткові умови до системи системи сингулярних інтегральних рівнянь:



Для подальшого чисельного аналізу до системи (3.1) застосуємо квадратурну формулу Гауса-Чебишева [2]:

, (3.4)

де

, (3.5)

Використовуючи наступну апроксимацію:

, (3.6)

перетворимо систему сингулярних інтегральних рівнянь другого роду з ядрами типу Коші і додаткові умови (4.1) в систему 2*n* однорідних алгебраїчних рівнянь щодо 2*n* невідомих  и , (*i* = 1,2,...,*n*).

Критичне навантаження  знаходиться з умови рівності нулю визначника отриманої системи алгебраїчних рівнянь.

# 4. Результати

Залежность критичного навантаження від товщини шару, коли верхній шар жорсткіший :

= 0.3;

= 0.2;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *h* | 0.15 | 0.12 | 0.10 | 0.08 | 0.05 | 0.03 | 0.02 |
|  | .05991 | .04323 | .03151 | .02767 | .01189 | .00574 | .00265 |

= 0.2;

= 0.3;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *h* | 0.15 | 0.12 | 0.10 | 0.08 | 0.05 | 0.03 | 0.02 |
|  | .04807 | .03495 | .02239 | .01083 | .00481 | .00236 | .00124 |

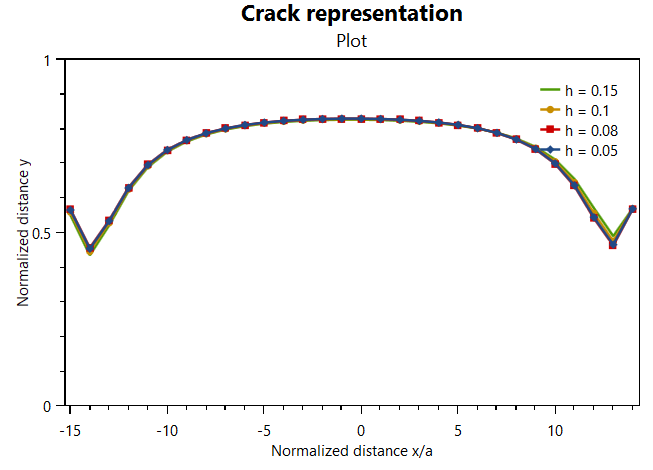
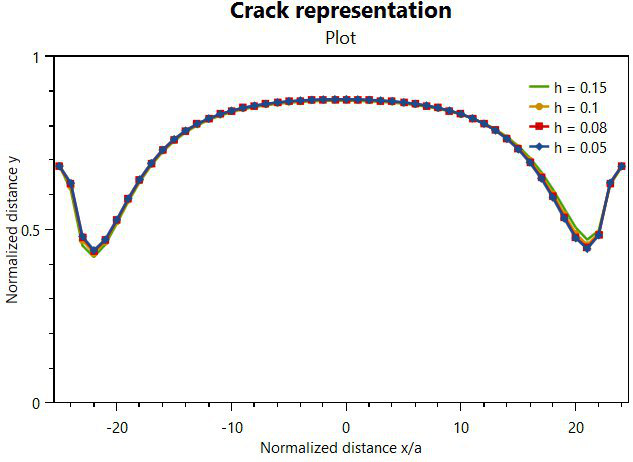
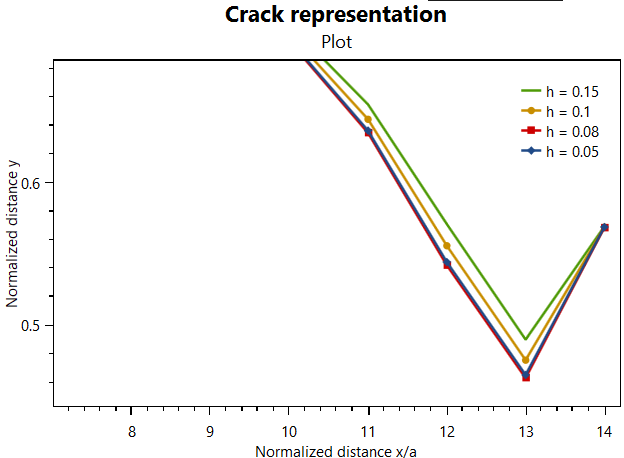


Рис.1: Розкриття тріщини = 0.3, = 0.2;

Рис. 2: Розкриття тріщини = 0.2, = 0.3;

Рис. 3: Детальнийрозбір відмінностей тріщин

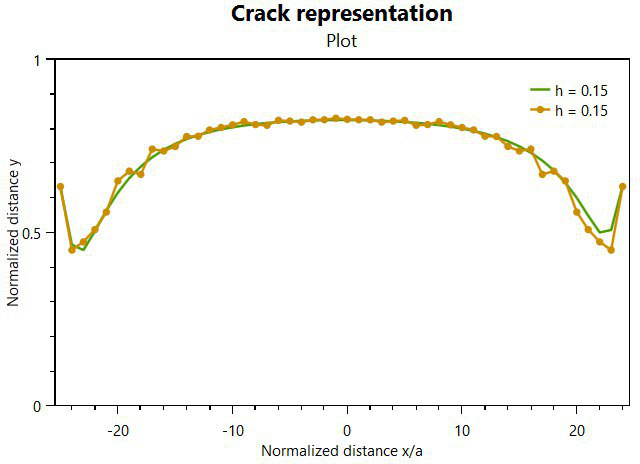


Рис. 3: Сопоставлення результатів нейронноі сітки та чисельного розв’язку

# 5. Висновки

В даній роботі була розглянута стійкість тріщини між шаром та півпросторому циліндричному тілі.

Через зворотне перетворення Ханкеля задача зведена до системи сингулярних інтегральних рівнянь за допомогою еліптичних інтегральних рівнянь першого та другого роду через . Квадратурною формулою Гауса-Чебишева система СIР перетворена в систему однорiдних алгебраїчних рiвнянь. Критичні значення навантаження знаходяться з умови рівності нулю визначника отриманої системи однорідних алгебраїчних рівнянь. Також була написана програма на мові програмування C#, за допомогоюякоїбулопобудованепредставленятріщини у момент розкриття та проаналізованазалежністькритичного навантаження від вісотиверхнього слою. Також було побудовано нейронну сітку на основі виданих результатів від програми, за для того, щоб апроксимувати дану задачу та скоротити час знаходження критичного навантаження.

# 6. Список використаної літератури

1. Амензаде Ю.А. Теория упругости. М.: Высшая школа, 1971, 286 с.

2. Лобода В.В., Шевельова А.Є., Говоруха В.Б. Чисельні методи розв’язування сингулярних інтегральних рівнянь: Дніпропетровськ, ДДУ, 1997, 52с.

3.. Трантер К.Дж. Интегральные преобразования в математической физике. М.: Гостехиздат, 1956, 204 с.

4. W.X.Wang, Y.Takao Load Bucking of a Layer Bonded to a Half-Space With an Interface Crack. ASME Journal of Applied Mechanics, vol. 62, MARCH 1995, 282 c.

# 7. Лістинг

privateMatrix\_matrixA;

constintN = 100;

doublep11, p22;

doublenu1, nu2;

doublemu1, mu2, mu, b1, b2;

doubles, p, h, a, g;

doublealpha11, alpha12, alpha21, alpha22;

doublealpha1b, alpha2b, alpha1c, alpha1d, alpha2c, alpha2d;

doubledelta1, delta2, a11, a12, a13, a21, a22, a23, a31, a32, a33, a41, a42, a43;

Func<double, double>r11, r21, r22, r12, r31, r32, r41, r42, r51, r52, r61, r62;

Func<double, double>n11, n12, n21, n22;

Func<double, double>q11, q12, q21, q22;

Func<double, double, double>K11, K12, K21, K22;

Func<double, double>TETTA, deltaMain;

double[] \_infinityT = newdouble[4];

double[] \_infinityQ = newdouble[4];

double[] \_sArray;

double[] \_tArray;

double[] p11array;

double[] detarray;

double[] Answer;

publicCrackSolution(Action<string>printer, BackgroundWorkerbackgroundWorker)

{

this.backgroundWorker = backgroundWorker;

this.printer = printer;

p11array = newdouble[N];

detarray = newdouble[N];

InvokeFunctions();

}

voidInvokeFunctions()

{

format = newNumberFormatInfo();

format.NumberGroupSeparator = ",";

format.NumberDecimalSeparator = ".";

r11 = t => (alpha1c \* s + 2 \* alpha1d \* alpha11 \* alpha11) \* Math.Exp(-2 \* t \* h \* alpha11) / delta1;

r12 = t =>2 \* alpha1d \* s \* Math.Exp(-t \* h \* (alpha11 + alpha12)) / delta1;

r21 = t => -4 \* alpha1c \* alpha11 \* alpha11 \* Math.Exp(-t \* h \* (alpha11 + alpha12)) / delta1;

r22 = t =>-(alpha1c \* s + 2 \* alpha1d \* alpha11 \* alpha11) \* Math.Exp(-2 \* t \* h \* alpha12) / delta1;

r31 = t => (a11 \* r11(t) + a12 \* r21(t) + a13) / delta2;

r32 = t => (a21 \* r12(t) + a22 \* r22(t) + a23) / delta2;

r41 = t => (a31 \* r11(t) + a32 \* r21(t) + a33) / delta2;

r42 = t => (a41 \* r12(t) + a42 \* r22(t) + a43) / delta2;

r51 = t =>a11 \* r11(t) + a12 \* r21(t) - a21 \* r31(t) - a22 \* r41(t) - a11;

r52 = t =>a11 \* r12(t) + a12 \* r22(t) - a21 \* r32(t) - a22 \* r42(t) - a12;

r61 = t =>a11 \* a11 \* r11(t) + r21(t) - a21 \* a21 \* r31(t) - r41(t) + a11 \* a11;

r62 = t =>a11 \* a11 \* r12(t) + r22(t) - a21 \* a21 \* r32(t) - r42(t) + 1;

n11 = t =>alpha1c \* r11(t) + alpha1d \* r21(t) - alpha1c;

n12 = t =>alpha1c \* r12(t) + alpha1d \* r22(t) - alpha1d;

n21 = t =>2 \* alpha11 \* alpha11 \* r11(t) + s \* r21(t) + 2 \* alpha11 \* alpha11;

n22 = t =>2 \* alpha11 \* alpha11 \* r12(t) + s \* r22(t) + s;

deltaMain = t =>r51(t) \* r62(t) - r61(t) \* r52(t);

q11 = t => (n11(t) \* r62(t) - n12(t) \* r61(t)) / deltaMain(t);

q21 = t => (n21(t) \* r62(t) - n22(t) \* r61(t)) / deltaMain(t);

q12 = t => (n12(t) \* r51(t) - n11(t) \* r52(t)) / deltaMain(t);

q22 = t => (n22(t) \* r51(t) - n21(t) \* r52(t)) / deltaMain(t);

K11 = (j, x) =>CommonTools.Integrate(t => (q11(t) - \_infinityQ[0]) \* Math.Cos(t \* j) \* Math.Cos(t \* x), \_infinityT[0]) / \_infinityQ[1];

K12 = (j, x) =>CommonTools.Integrate(t => (q12(t) - \_infinityQ[1]) \* Math.Sin(t \* j) \* Math.Cos(t \* x), \_infinityT[1]) / \_infinityQ[1];

K21 = (j, x) =>CommonTools.Integrate(t => (q21(t) - \_infinityQ[2]) \* Math.Cos(t \* j) \* Math.Sin(t \* x), \_infinityT[2]) / -\_infinityQ[2];

K22 = (j, x) =>CommonTools.Integrate(t => (q22(t) - \_infinityQ[3]) \* Math.Sin(t \* j) \* Math.Sin(t \* x), \_infinityT[3]) / -\_infinityQ[2];

TETTA = Si => - g \* Math.Log(1 - Si \* Si);

}

internaldouble[] CalculateWithNeuralNetworkTest(doubleh, doublea, doublev1, doublev2, doublemu1, doublemu2)

{

varnumber = new[] { h, v1, v2 };

double[][] w1 = newdouble[3][];

double[][] w2 = newdouble[500][];

double[][] wout = newdouble[500][];

string[] w1lines = File.ReadAllLines(@"crack\_wghts\_layer1\_4000ep.txt").ToArray();

string[] w2lines = File.ReadAllLines(@"crack\_wghts\_layer2\_4000ep.txt").ToArray();

string[] woutlines = File.ReadAllLines(@"crack\_wghts\_out\_4000ep.txt").ToArray();

// разобрать в массив

for (inti = 0; i<3; i++)

{

double[] row = w1lines[i].Split(newchar[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(v =>Double.Parse(v, format)).ToArray();

w1[i] = row;

}

for (inti = 0; i<500; i++)

{

double[] row = w2lines[i].Split(newchar[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(v =>Double.Parse(v, format)).ToArray();

w2[i] = row;

}

for (inti = 0; i<500; i++)

{

double[] row = woutlines[i].Split(newchar[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(v =>Double.Parse(v, format)).ToArray();

wout[i] = row;

}

varinputs = number;

doubletmp;

double[] outputs1 = newdouble[500];

double[] outputs2 = newdouble[500];

double[] result = newdouble[27];

for (introw = 0; row<500; row++)

{

//inputs sinapses

tmp = 0;

for (intcol = 0; col<3; col++)//(3 dendrids in 2000 neurons)

tmp += inputs[col] \* w1[col][row];

outputs1[row] = normalizeAnswer(tmp);

}

for (introw = 0; row<500; row++)

{

//inputs sinapses

tmp = 0;

for (intcol = 0; col<500; col++)////(2000 dendrids in 2000 neurons)

tmp += outputs1[col] \* w2[col][ row];

outputs2[row] = normalizeAnswer(tmp);

}

//hidden sinapses

for (introw = 0; row<27; row++)

{

tmp = 0;

for (intcol = 0; col<500; col++) //(2000 dendrids in 27 neurons)

tmp += outputs2[col] \* wout[col][row];

result[row] = normalizeAnswer(tmp);

}

printer("\*Critical P: " + result[0] + Environment.NewLine);

doubleaxys = result[2] - (result[2] - result[1]);

double[] res = newdouble[50];

for (inti = 0; i<26; i++)

{

if (i<24)

res[26 - i + 23] = result[26 - i];

res[i] = result[26 - i];

}

returnres;

}

internaldouble[] CalculateWithNeuralNetwork(doubleh, doublea, doublev1, doublev2, doublemu1, doublemu2)

{

returnanalize(new []{h, v1, v2});

}

double[] analize(double[] number)

{

double[][] w = newdouble[3][]; ;

double[][] wout = newdouble[2000][];

string[] wlines = File.ReadAllLines(@"crack\_wghts\_test.txt").ToArray();

string[] woutlines = File.ReadAllLines(@"crack\_wghts\_out\_test.txt").ToArray();

// разобрать в массив

for (inti = 0; i<3; i++)

{

double[] row = wlines[i].Split(newchar[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(v =>Double.Parse(v, format)).ToArray();

w[i] = row;

}

for (inti = 0; i<2000; i++)

{

double[] row = woutlines[i].Split(newchar[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(v =>Double.Parse(v, format)).ToArray();

wout[i] = row;

}

double[] result = newdouble[27];

varinputs = number;

doubletmp;

double[] outputs = newdouble[2000];

for (introw = 0; row<2000; row++)

{

//inputs sinapses

tmp = 0;

for (intcol = 0; col<3; col++)

tmp += inputs[col] \* w[col][row];

outputs[row] = normalizeAnswer(tmp);

}

//hidden sinapses

for (introw = 0; row<27; row++)

{

tmp = 0;

for (intcol = 0; col<2000; col++)

tmp += outputs[col] \* wout[col][row];

result[row] = normalizeAnswer(tmp);

}

printer("\*Critical P: " + result[0] + Environment.NewLine);

double[] res = newdouble[50];

for (inti = 0; i<26; i++)

{

if(i<24)

res[26 - i + 23] = result[26 - i];

res[i] = result[26 - i];

}

returnres;

}

doublenormalizeAnswer(doubleanswer)

{

returnMath.Log(answer + Math.Sqrt(answer \* answer + 1));

}

double[] globalSolution = newdouble[0];

privatedoublediv = 0.01, dec = 0;

publicdouble[] Calculate(intn, doubleh, doublea, doublenu1, doublenu2, doublemu1, doublemu2) //коефпуассона, модульсдвига

{

backgroundWorker.ReportProgress(0);

this.h = h;

this.a = a;

\_matrixA = newMatrix(2\*n);

\_sArray = newdouble[n];

\_tArray = newdouble[n - 1];

mu = mu2 / mu1;

for (inti = 1; i<N; i++)

{

p11 = i \* div + dec;

p22 = (1 - nu1) \* mu \* p11 / (1 - nu2);

alpha11 = Math.Sqrt(1 - p11 \* (1 - 2 \* nu1) / (2 \* (1 - nu1)));

alpha21 = Math.Sqrt(1 - p22 \* (1 - 2 \* nu2) / (2 \* mu \* (1 - nu2)));

alpha12 = Math.Sqrt(1 - p11);

alpha22 = Math.Sqrt(1 - p22 / mu);

//alpha1a = 2 \* ((1 - nu1) \* alpha11 - nu1 \* Math.Pow(alpha11, 3)) / (1 - 2 \* nu1);

//alpha2a = 2 \* mu \* ((1 - nu2) \* alpha21 - nu2 \* Math.Pow(alpha21, 3)) / (1 - 2 \* nu2);

alpha1b = 2 \* alpha12;

alpha2b = 2 \* mu \* alpha22;

//2

alpha1c = 2 \* (nu1 \* alpha11 - (1 - nu1) \* Math.Pow(alpha11, 3)) / (1 - 2 \* nu1);

alpha2c = 2 \* mu \* (nu2 \* alpha21 - (1 - nu2) \* Math.Pow(alpha21, 3)) / (1 - 2 \* nu2);

alpha1d = -alpha1b;

alpha2d = -alpha2b;

s = 1 + alpha12 \* alpha12; p = 1 + alpha22 \* alpha22;

delta1 = alpha1c \* s - 2 \* alpha1d \* alpha11 \* alpha11;

delta2 = alpha2c \* p - 2 \* alpha2d \* alpha21 \* alpha21;

a11 = alpha1c \* p - 2 \* alpha2d \* alpha11 \* alpha11 / mu;

a12 = alpha1d \* p - alpha2d \* s / mu;

a13 = -(alpha1c \* p + 2 \* alpha2d \* alpha11 \* alpha11 / mu);

a21 = a11;

a22 = a12;

a23 = -(alpha1d \* p + alpha2d \* s / mu);

//3

a31 = 2 \* alpha2c \* alpha11 \* alpha11 / mu - 2 \* alpha1c \* alpha21 \* alpha21;

a32 = alpha2c \* s / mu - 2 \* alpha1d \* alpha21 \* alpha21;

a33 = 2 \* alpha2c \* alpha11 \* alpha11 / mu + 2 \* alpha1c \* alpha21 \* alpha21;

a41 = a31;

a42 = a32;

a43 = alpha2c \* s / mu + 2 \* alpha1d \* alpha21 \* alpha21;

varaii = new[] {a11, a12, a13, a21, a22, a23 , a31, a32, a33 , a41, a42, a43 };

CommonTools.CalculateInfinity(new[] { q11, q12, q21, q22 }, out\_infinityQ, out\_infinityT);

b1 = \_infinityQ[0] / \_infinityQ[1];

b2 = \_infinityQ[3] / \_infinityQ[2];

g = Math.Log((1 + Math.Sqrt(b1 \* b2)) / (1 - Math.Sqrt(b1 \* b2))) / (2 \* Math.PI);

#region Filling the matrix

//Main functions

double[] CosTetta = newdouble[n];

double[] SinTetta = newdouble[n];

\_matrixA.Null();

for (intk = 0; k<n; k++)

{

\_sArray[k] = Math.Cos((2 \* k + 1) \* Math.PI / (2 \* n));

CosTetta[k] = Math.Cos(TETTA(\_sArray[k]));

SinTetta[k] = Math.Sin(TETTA(\_sArray[k]));

if (k<n - 1)

\_tArray[k] = Math.Cos((k + 1) \* Math.PI / n);

}

for (intk = 0; k<n - 1; k++)

{ //Main diaghonal (1,n-1)

\_matrixA.Body[k][k] += 0.5 \* b1 \* CosTetta[k] / Math.Sqrt(1 - Math.Pow(\_sArray[k], 2));

\_matrixA.Body[k][k + 1] += 0.5 \* b1 \* CosTetta[k + 1] / Math.Sqrt(1 - Math.Pow(\_sArray[k + 1], 2));

\_matrixA.Body[k][n + k] -= 0.5 \* b1 \* SinTetta[k] / Math.Sqrt(1 - Math.Pow(\_sArray[k], 2));

\_matrixA.Body[k][n + k + 1] -= 0.5 \* b1 \* SinTetta[k + 1] / Math.Sqrt(1 - Math.Pow(\_sArray[k + 1], 2));

//Main diaghonal (n, 2n-2)

\_matrixA.Body[n - 1 + k][k] -= 0.5 \* b2 \* SinTetta[k] / Math.Sqrt(1 - Math.Pow(\_sArray[k], 2));

\_matrixA.Body[n - 1 + k][k + 1] -= 0.5 \* b2 \* SinTetta[k + 1] / Math.Sqrt(1 - Math.Pow(\_sArray[k + 1], 2));

\_matrixA.Body[n - 1 + k][n + k] -= 0.5 \* b2 \* CosTetta[k] / Math.Sqrt(1 - Math.Pow(\_sArray[k], 2));

\_matrixA.Body[n - 1 + k][n + k + 1] -= 0.5 \* b2 \* CosTetta[k + 1] / Math.Sqrt(1 - Math.Pow(\_sArray[k + 1], 2));

}

//Direct filling 1,2n

for (intl = 0; l<n; l++)

{

//Last Rows

\_matrixA.Body[2 \* n - 2][l] += CosTetta[l] / n;

\_matrixA.Body[2 \* n - 2][l + n] += -SinTetta[l] / n;

\_matrixA.Body[2 \* n - 1][l] += SinTetta[l] / n;

\_matrixA.Body[2 \* n - 1][l + n] += CosTetta[l] / n;

for (intm = 0; m<n - 1; m++)

{

//(1, n-1)

\_matrixA.Body[m][l] += SinTetta[l] / (n \* (\_sArray[l] - \_tArray[m]))

+ K11(\_sArray[l], \_tArray[m]) \* CosTetta[l] / n

+ K12(\_sArray[l], \_tArray[m]) \* SinTetta[l] / n;

\_matrixA.Body[m][l + n] += CosTetta[l] / (n \* (\_sArray[l] - \_tArray[m]))

- K11(\_sArray[l], \_tArray[m]) \* SinTetta[l] / n

+ K12(\_sArray[l], \_tArray[m]) \* CosTetta[l] / n;

//(n, 2n-2)

\_matrixA.Body[m + n - 1][l] += CosTetta[l] / (n \* (\_sArray[l] - \_tArray[m]))

+ K21(\_sArray[l], \_tArray[m]) \* CosTetta[l] / n

+ K22(\_sArray[l], \_tArray[m]) \* SinTetta[l] / n;

\_matrixA.Body[m + n - 1][l + n] += -SinTetta[l] / (n \* (\_sArray[l] - \_tArray[m]))

- K21(\_sArray[l], \_tArray[m]) \* SinTetta[l] / n

+ K22(\_sArray[l], \_tArray[m]) \* CosTetta[l] / n;

}

}

#endregion

printer(p11 + Environment.NewLine);

p11array[i] = p11;

detarray[i] = \_matrixA.Determinant;

printer(detarray[i] + Environment.NewLine);

// when det< 0 stop

if (detarray[i] <0.0f)

{

if (div>0.00001)

{

dec = p11 - div;

div \*= 0.1;

Calculate(n, h, a, nu1, nu2, mu1, mu2);

break;

}

printer("\*---------------------------\*" + Environment.NewLine);

printer("h = " + h + Environment.NewLine);//p11 determ

printer($"Critical P {p11array[i-1]} {Environment.NewLine}");/\*{detarray[i - 1]}\*/

printer($"{p11array[i]} {detarray[i]}{Environment.NewLine}");

for (intlast = i; last<101; last++)

backgroundWorker.ReportProgress(last);

printer("\*---------------------------\*" + Environment.NewLine);

div = 0.01; dec = 0;

Printrepres(\_matrixA.AbstractResolutions);

double[] answ = newdouble[n/2];

double[] tmp = \_matrixA.AbstractResolutions;

globalSolution = newdouble[n];

globalSolution = GetPayload(tmp, n);

//globalSolution = MakePretty(tmp, n);

Printrepres(globalSolution);

returnglobalSolution;

}

backgroundWorker.ReportProgress((int)(i / dec / 2));

}

returnglobalSolution;

}

privatedouble[] GetPayload(double[] tmp, intn)

{

double[] res = newdouble[n];

if (mu1<mu2)

{

for (intv = 0; v<n; v++)

{

res[v] = tmp[n + v];

}

}

else

{

for (intv = 0; v<n; v++)

{

res[v] = tmp[n \* 2 - 1 - v];

}

}

returnres;

}

privatedouble[] MakePretty(double[] tmp, intn)

{

double[] res = newdouble[n];

double[] answ = newdouble[n/2];

if (mu1<mu2)

{

for (intv = 0; v<n / 2; v++)

{

answ[v] = tmp[n + v];

}

}

else

{

for (intv = 0; v<n / 2; v++)

{

answ[v] = tmp[n \* 2 - 1 - v];

}

}

for (intv = 0; v<n / 2; v++)

{

// Turbo mode

res[n / 2 - 1 + v] = answ[n / 2 - 1 - v];

res[v] = answ[v];

}

returnres;

}

privatevoidPrintrepres(double[] matrixAAbstractResolutions)

{

printer($"answrs = {Environment.NewLine}");

for (inti = 0; i<matrixAAbstractResolutions.Length; i++)

printer($"{matrixAAbstractResolutions[i]},{Environment.NewLine}");

}

publicstaticdoubleIntegrate(Func<double, double>func, doubleb)

{

doublesum = 0;

doubleh = b / 100;

doublex = h;

while (x<b)

{

sum += 4 \* func(x);

x += h;

if (x>= b) break;

sum += 2 \* func(x);

x += h;

}

sum = (h / 3) \* (sum + func(0) - func(b));

returnsum;

}

publicstaticvoidCalculateInfinity(Func<double, double>[] functions, outdouble[] \_infinityQ, outdouble[] \_infinityT)

{

\_infinityQ = newdouble[4];

\_infinityT = newdouble[4];

for (inti = 0; i<functions.Length; i++)

{

intstep = 0;

doubleSn = 0, Sn1 = 100;

while (Math.Abs(Sn1 - Sn) >0.000001)

{

step++;

Sn = Sn1;

Sn1 = functions[i](step);

}

\_infinityQ[i] = Sn1;

\_infinityT[i] = step;

}

}