



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY



Комтех–Н

Мультипарадигменный метод разработки параллельных программ синтеза элементов микро-оптики

Яблоков Д.Е.

д.т.н., профессор
Востокин С.В.

д.ф.-м.н., профессор
Павельев В.С.

GRID'2025



Предмет исследования:

Анализ и моделирование информационных процессов, применяемых в компьютерной оптике.

Цель исследования:

Разработка программной технологии и реализации в виде автоматизированной системы на объектно-ориентированном языке, поддерживающем параметрический полиморфизм подтипов, для автоматизации процессов расчёта элементов микро-оптики с применением мультипарадигменного анализа.



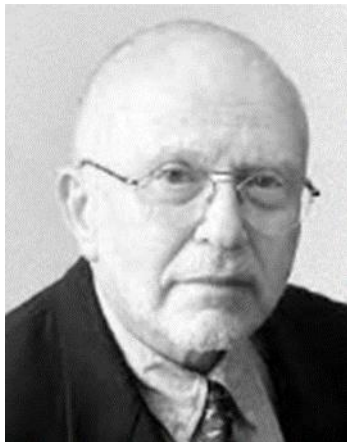
- ▶ Анализ методов и алгоритмов расчета элементов микро-оптики с целью выявления у них признаков общности и изменчивости в форме обобщённых абстракций.
- ▶ Разработка предметно-ориентированных абстракций в области расчёта элементов микро-оптики на основе спецификации перечня семантических и синтаксических требований.
- ▶ Разработка обобщённого алгоритма для расчёта элементов микро-оптики, а также построение моделирующих типов данных с применением ограничений на основе параметрического полиморфизма подтипов.
- ▶ Экспериментальное исследование эффективности предлагаемого теоретического подхода к созданию систем расчёта элементов микро-оптики и апробация полученных результатов.
- ▶ Реализация информационной технологии моделирования и расчета дифракционных элементов микро-оптики в форме программной системы.
- ▶ Исследование возможности распараллеливания расчёта элементов микро-оптики на основе разработанной программной архитектуры.



По моему опыту, даже в традиционном программировании очень важно искать возможные обобщения, которые могут оказаться весьма полезными, поскольку такие соображения дают четкое видение того, как должна быть структурирована конечная программа. Такие соображения сводятся к представлению (более или менее явно) о целых семействах подобных программ!

Edsger Dijkstra

“Notes on Structured Programming”, 1972



Мы рассматриваем набор программ, составляющий семейство, всякий раз, когда имеет смысл изучать эти программы, сначала изучив их общие свойства, а затем особые свойства отдельных членов семейства.

David Parnas

“On the Design and Development of Program Families”,
IEEE Transactions on Software Engineering,
VOL.SE-2, NO.1, MARCH 1976



Общность – это допущение, которое применяется единообразно для данного набора объектов (S). Часто такими предположениями являются атрибуты с одинаковыми значениями для всех элементов S . И наоборот, изменчивость – это предположение, верное только для некоторых элементов S , например, атрибут с разными значениями по крайней мере для двух элементов S .

Jemes Coplien, David Hoffman, David Weiss
“Commonality and variability in software engineering”
IEEE Software, vol. 15(6), pp. 37-45, 1998

Определить семейство – значит выявить требования к его потенциальным членам, охарактеризовать, что у них общего и чем они различаются.

N.L. Gupta et al.
“Auditdraw: Generating Audits the FAST Way”
IEEE International Symposium Requirements Engineering, pp. 188-197, 1997



Процедурная парадигма:

- ▶ S: Набор похожих фрагментов кода, каждый из которых может быть заменен вызовом некоторой функции F.
- ▶ C: Код, общий для всех фрагментов из S.
- ▶ V: Оригинальный код в S, не имеющий похожих фрагментов. Может быть выражен через вариацию значений параметров F или специальными инструкциями до или после каждого вызова F.

Объектно-ориентированная парадигма:

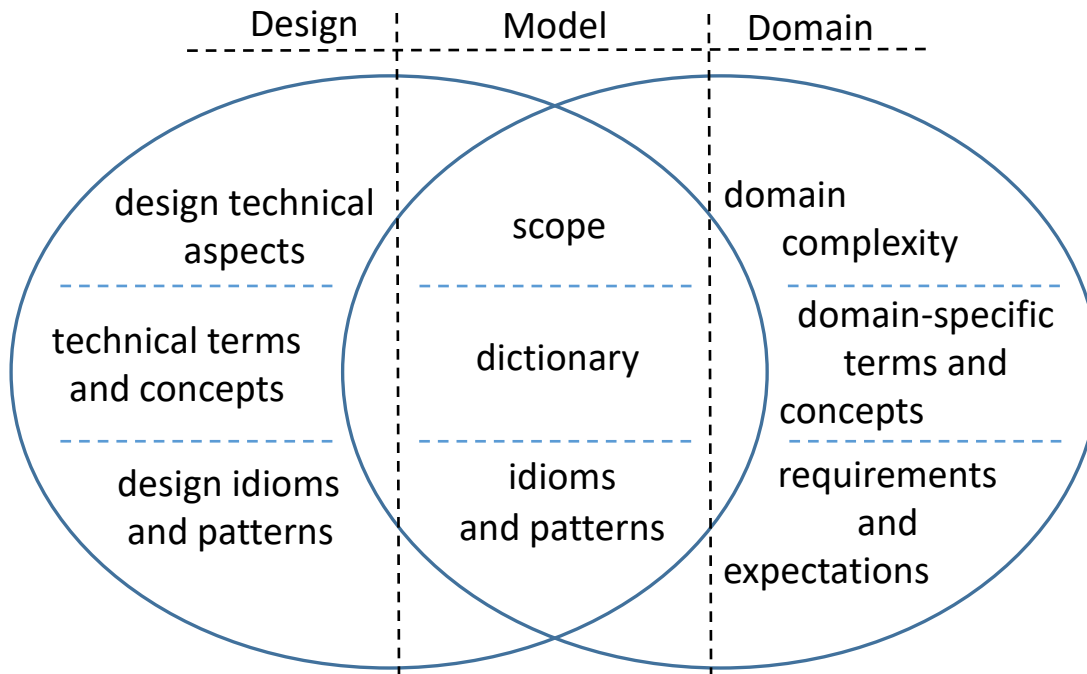
- ▶ S: Иерархия классов.
- ▶ C: Код общий для всех классов из S, который помещается в базовый класс. Общая сигнатура для всех классов в S, поддерживаемая всеми классами.
- ▶ V: Код в S, определяющий полиморфные варианты поведения; этот код помещается в подклассы.

Обобщённая парадигма:

- ▶ S: Набор процедур и/или классов, реализующих одинаковые операции, но для разных типов.
- ▶ C: Код, одинаковый для всех процедур и/или классов, отличающийся только параметрами типов для которых он выполняется.
- ▶ V: Фактические параметры шаблона, то есть разные типы.

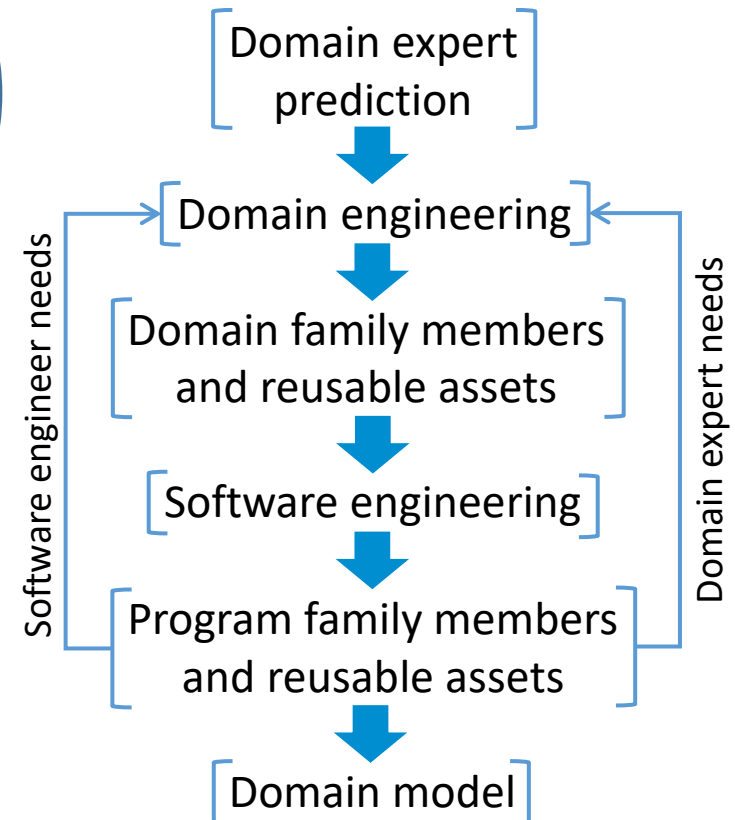


Domain-specific model distillation



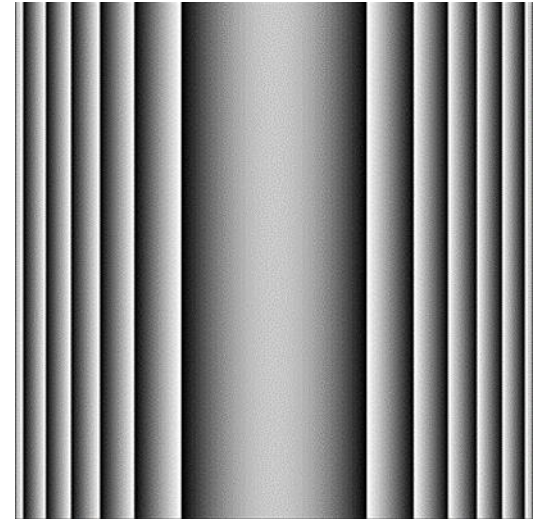
Main steps:

- ▶ Establish the scope.
- ▶ Build a common dictionary.
- ▶ Identify the commonalities and variabilities.
- ▶ Bound the variabilities for specific characteristics.





```
1 double[] Cylindrical(double x0, double dx, int m, int n,  
2     double lambda, double fLength)  
3 {  
4     var res = new double[m * n + m + n + 1];  
5     var fx = -Math.PI / (lambda * fLength);  
6     for (int j = 0, k = 0; j < n; ++j, k += m)  
7     {  
8         for (var i = 0; i < m; ++i)  
9         {  
10             var x = x0 + i * dx;  
11             res[i + k + j] = fx * Math.Pow(x, 2);  
12         }  
13     }  
14     return res;  
15 }
```



Преимущества:

- ▶ Семантика метода проста и не содержит сложных абстракций. На первый взгляд она понятна и прозрачна для построения ассоциаций.
- ▶ Возможность организации подобных методов в других единицах трансляции на той же терминологической основе и с тем же уровнем сложности.



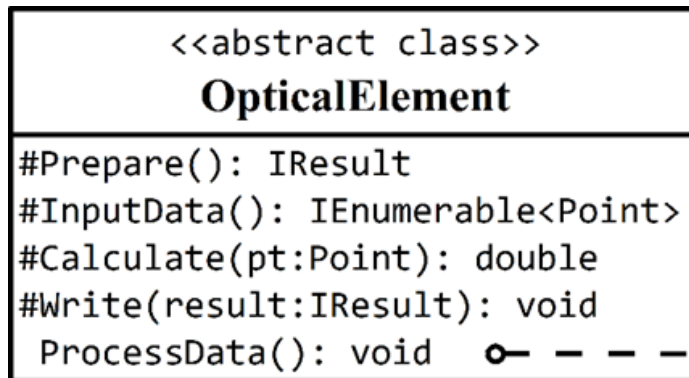
Для такого небольшого фрагмента кода сложность модификации, а также отсутствие достаточной гибкости и развитости семантики при работе с данными не является совсем плохим и недопустимым решением. Но...

Могут потребоваться изменения для:

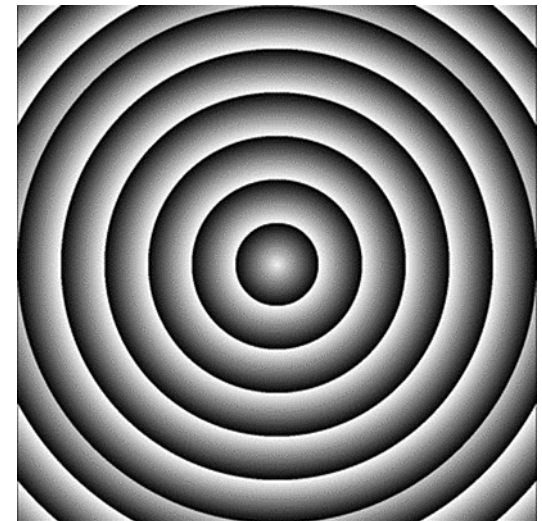
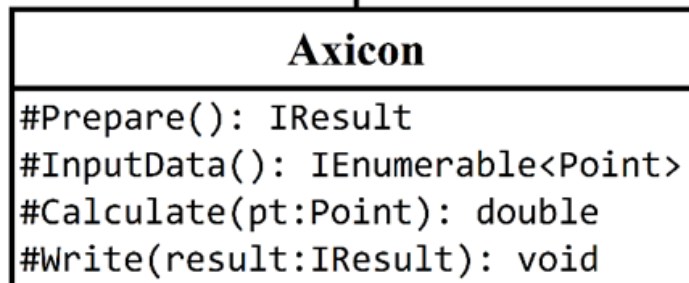
- ▶ Способа обхода передаваемой на вход алгоритма области.
- ▶ Стратегии получения координат текущей точки.
- ▶ Индексации результирующего массива.
- ▶ Типа и количества аргументов.
- ▶ Типа возвращаемого значения.
- ▶ Последовательности расчёта фазовой функции.

Выводы:

- ▶ Метод плохо подходит для алгоритмической декомпозиции.
- ▶ Метод плохо спроектирован и слабо структурирован.



```
var result = Prepare();  
foreach (var pt in InputData())  
    result.Assign(pt, Calculate(pt));  
Write(result);
```





Преимущества:

- ▶ Эффективное использование выразительных возможностей объектно-ориентированной парадигмы.
- ▶ Локализация данных и операций в виде отдельных абстракций, что повышает наглядность и удобство сопровождения.
- ▶ Стимуляция к повторному использованию. Обладает большей компактностью чем процедурный эквивалент.
- ▶ Возможность расширения каркаса понятий.
- ▶ Межкомпонентные взаимодействия, разумное разделение ответственности и обязанностей, что подталкивает к правильной стратегии принятия проектных решений.
- ▶ Параметризация поведения алгоритма.

Недостатки:

- ▶ Жесткая привязка структуры алгоритма к абстракции `OpticalElement`. Выполнение операции `ProcessData` может иметь более широкие смысловые границы.
- ▶ Способность работы только с ограниченным диапазоном типов (например, `double`).
- ▶ Возможность нарушения LSP.



Unary version

```
1 | int Phase<T, R>(T[] src, R[] res, Func<T, R> uop)
2 | {
3 |     var count = 0;
4 |     for (; count < src.Length; ++count)
5 |         res[count] = uop(src[count]);
6 |     return count;
7 | }
```

Binary version

```
1 | int Phase<T1, T2, R>(T1[] src1, T2[] src2, R[] res, Func<T1, T2, R> bop)
2 | {
3 |     var count = 0;
4 |     for (var i = 0; i < src1.Length; ++i)
5 |     {
6 |         for (var j = 0; j < src2.Length; ++j, ++count)
7 |             res[count] = bop(src1[i], src2[j]);
8 |     }
9 |     return count;
10 | }
```



Преимущества:

- ▶ Работа в наиболее общей ситуации с наибольшим количеством встроенных или пользовательских типов.
- ▶ Дополнительный уровень гибкости. Операция расчёта выделена из основного цикла. Доступ к элементам $O(1)$.
- ▶ Позволяет сконцентрироваться на основных программных абстракциях процедурной парадигмы, а затем уже на данных, которые в них обрабатываются.

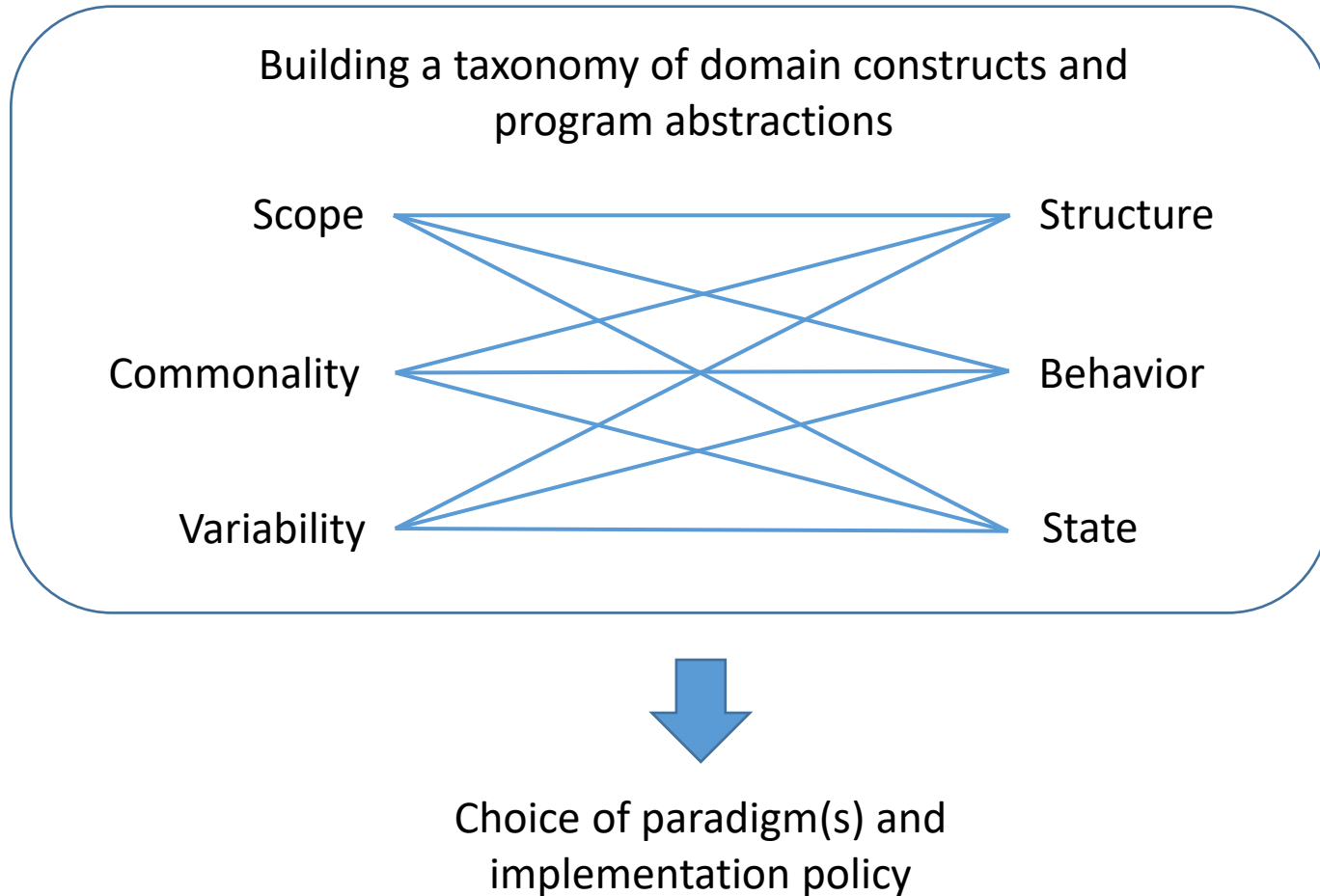
Недостатки:

- ▶ Вынужденная адаптация к форме массива. Потребуется операция копирования, которая имеет сложность $O(N)$.
- ▶ Потенциальная возможность получить `IndexOutOfRangeException`
- ▶ Массивы не обладают достаточным уровнем гибкости, чтобы использоваться в качестве основного способа представления данных
 - фиксированный размер,
 - непосредственное хранение данных,
 - поиск, вставка, удаление,
 - etc.



```
1 abstract class OpticalElement<T, R>
2 {
3     protected abstract IResult<T, R> Prepare();
4     protected abstract IEnumerable<T> InputData();
5     protected abstract R Calculate(T id);
6     protected abstract void Write(IResult<T, R> result);
7     public void ProcessData()
8     {
9         var result = Prepare();
10        foreach (var id in InputData())
11            result.Assign(id, Calculate(id));
12        Write(result);
13    }
14 };
```

- ▶ Все преимущества не обобщённой версии.
- ▶ Возможность создавать производные классы для множества различных типов.
- ▶ Жесткая привязка структуры алгоритма к абстракции `OpticalElement`.
- ▶ Возможность нарушения Liskov Substitution Principle.



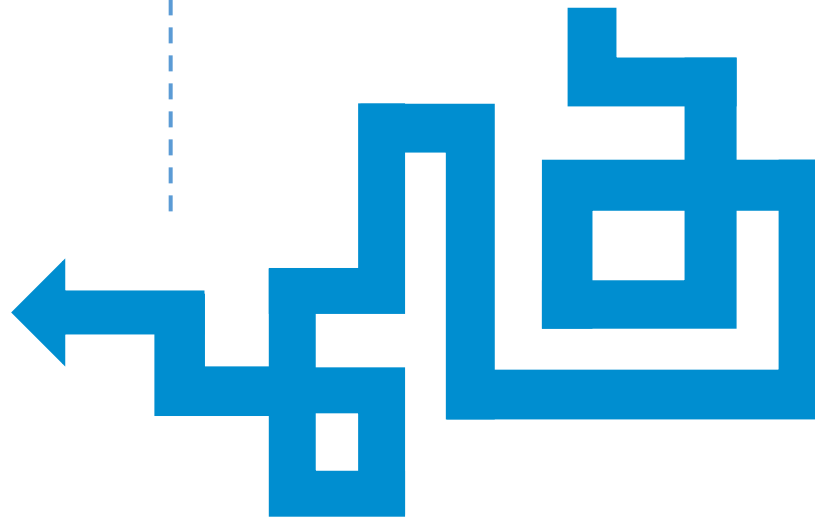


$$\varphi_1(x, y) = -\frac{\pi \left(\frac{x}{\cos \gamma} \right)^2 + y^2}{\lambda f}$$

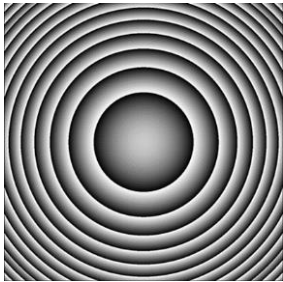


$$\varphi(x, y) = abx^2 + by^2$$
$$a = \frac{1}{\cos^2 \gamma}, b = -\frac{\pi}{\lambda f}$$

$$\varphi_2(x, y) = -\frac{\pi(x^2 - y^2 \sin^2 \gamma) + \pi y^2}{\lambda f \left(1 - \frac{1}{1 + \cot^2 \gamma} \right)}$$



- ▶ φ_1 и φ_2 алгебраически эквивалентны.
- ▶ φ_1 и φ_2 не эквивалентны структурно.
- ▶ иногда очень трудно распознать подобие φ_1 и φ_2 при неявно выраженной структурной идентичности.



1

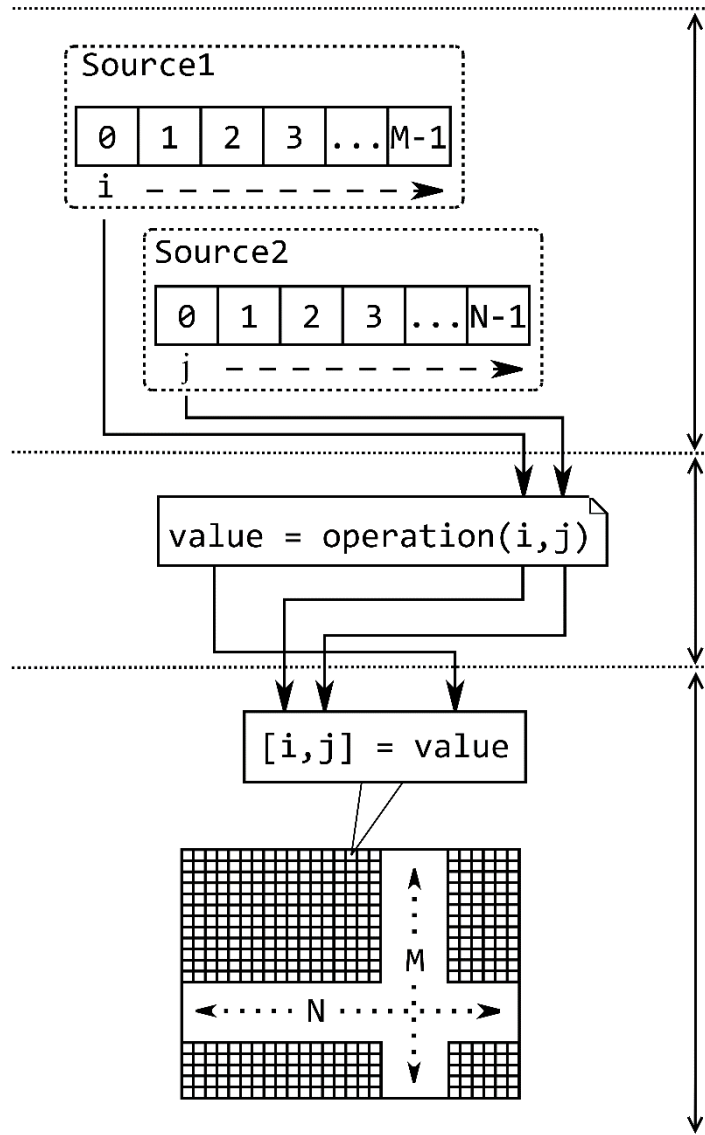
2

3

```
1 double[,] Fresnel(int m, int n, double x0, double y0,
2   double dx, double dy, double w1, double fd, double g)
3 {
4   var res = new double[m,n];
5
6   var b = -Math.PI / (w1 * fd);
7   var a = b * Math.Pow(Math.Cos(g * (Math.PI / 180)), 2);
8
9   for (var i = 0; i < m; ++i)
10  {
11    for (var j = 0; j < n; ++j)
12    {
13      var x = x0 + i * dx;
14      var y = y0 - j * dy;
15      var value = a * Math.Pow(x, 2) + b * Math.Pow(y, 2);
16      res[i,j] = value;
17    }
18  }
19 }
20 return res;
21 }
```



Program family members



- 1 input ranges traversal
- 2 calculate the current value
- 3 write the current value to the output range

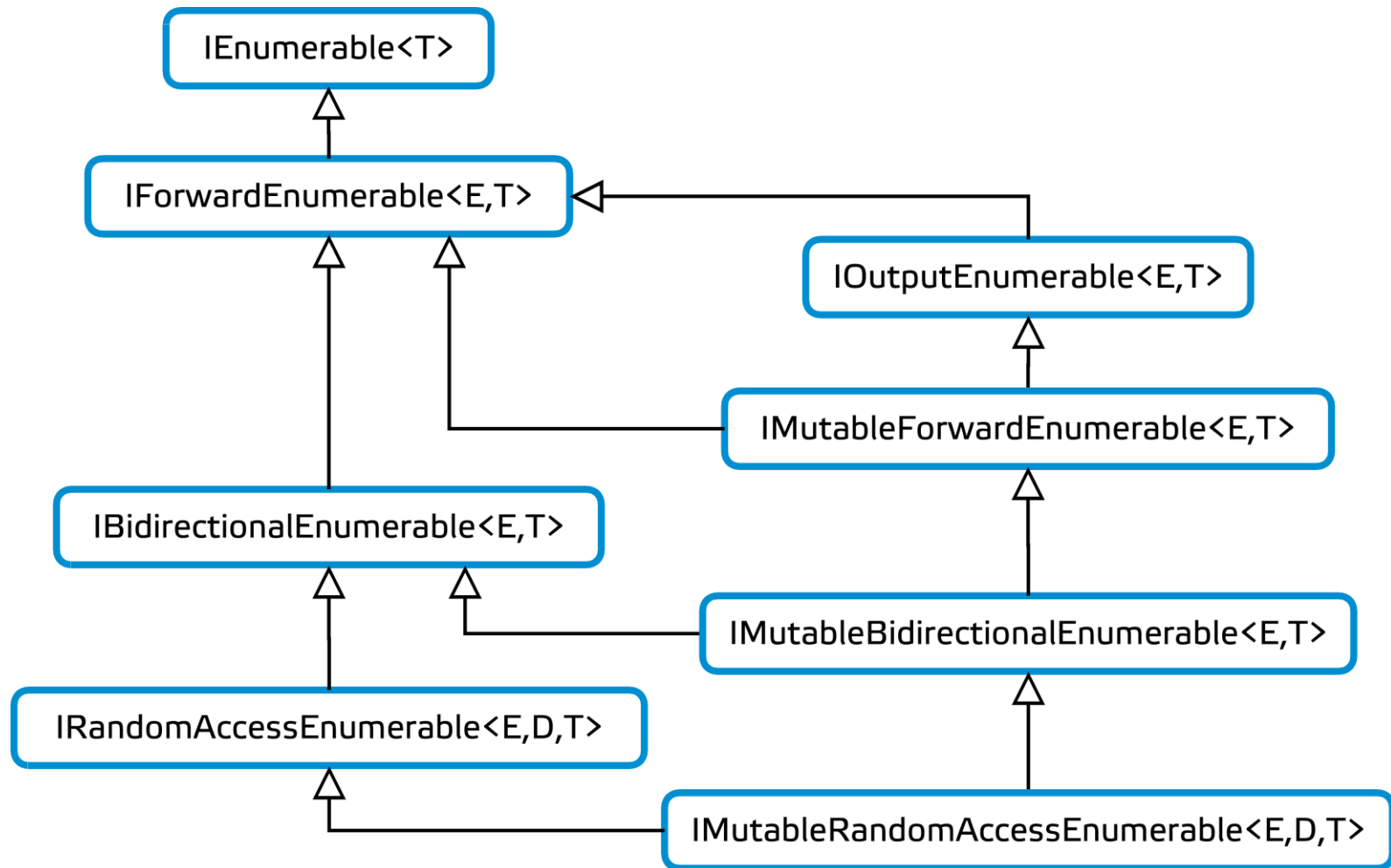


Generic algorithm (multi-paradigm approach)

```
1  OE InnerTransform<OE, T1, T2, R>(this IEnumerable<OE, R> target,  
2      IEnumerable<T1> source1, IEnumerable<T2> source2, Func<T1, T2, R> operation)  
3  where OE : IEnumerable<OE, R>  
4  {  
5      using var output = target.GetEnumerator();  
6      using var input1 = source1.GetEnumerator();  
7  
8      for (var exit = false; !exit && input1.MoveNext();) 1  
9      {  
10         foreach (var value2 in source2)  
11         {  
12             if (!output.MoveNext())  
13             {  
15                 exit = true;  
16                 break;  
17             }  
18             output.Current = operation(input1.Current, value2);  
19         }  
20     }  
21     3  
22     2  
23     return output.GetEnumerator();  
24 }
```



Refinement of IEnumerable<T> concept





Forward read-only multi-traversal enumerator

```
1 interface IForwardEnumerable<E, T> : IEnumerable<T>
2 {
3     new IForwardEnumerator<E, T> GetEnumerator();
4
5     IEnumerator IEnumerable.GetEnumerator() => GetEnumerator();
6
7     IEnumerator<T> IEnumerable<T>.GetEnumerator() => GetEnumerator();
8 };
```



```
1 interface IForwardEnumerator<E, T> : IEnumerator<T>
2 {
3     E GetEnumeratorable();
4
5     object IEnumerator.Current => Current;
6 };
```

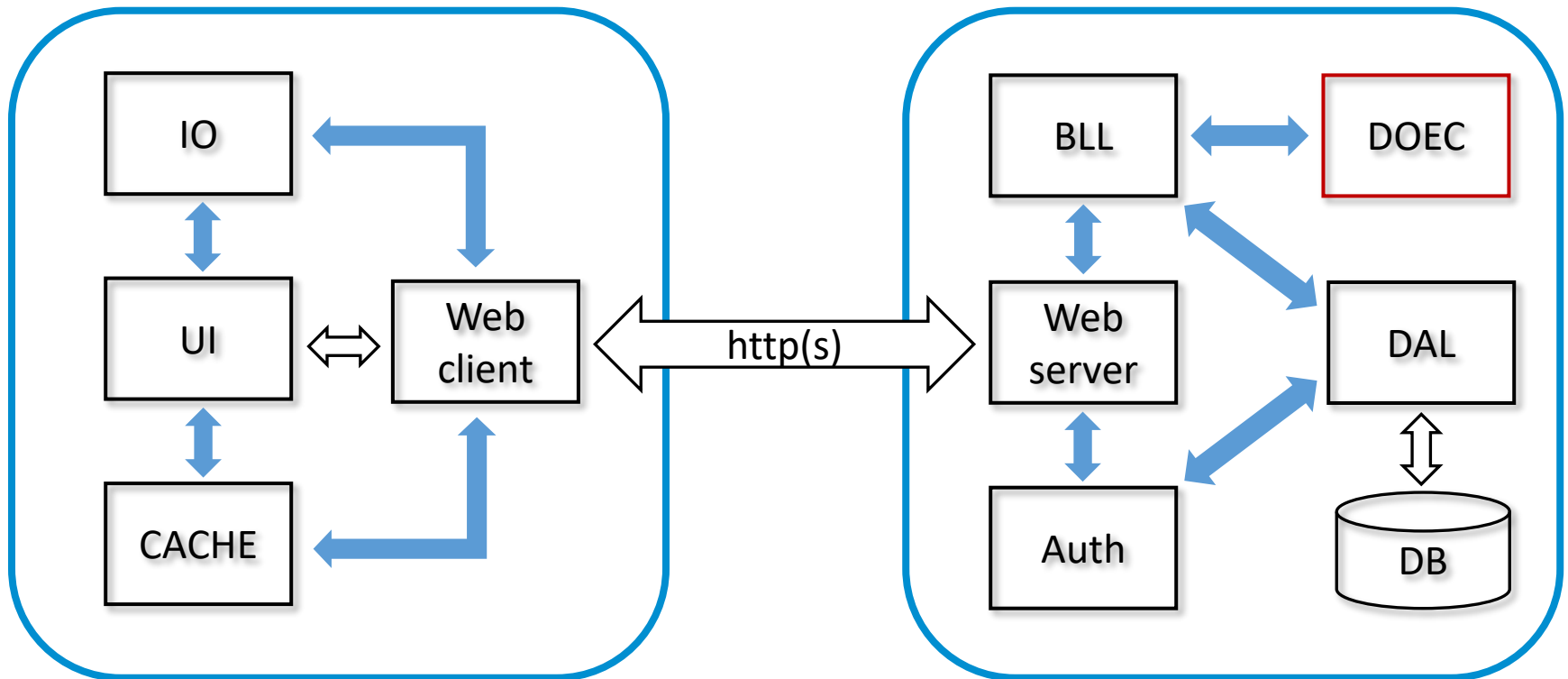


```
1 class BitmapEnumerator : IOutputEnumerator<IBitmapEnumerable, Position2d<int, byte>> {
2     private readonly int _pixelSize;
3     private readonly Bitmap _bmp;
4     private readonly BitmapData _bmpData;
5     public BitmapEnumerator(Bitmap bmp) {
6         _bmp = bmp;
7         _pixelSize = Image.GetPixelFormatSize(bmp.PixelFormat) / 8;
8         _bmpData = bmp.LockBits(new Rectangle(0, 0, bmp.Width,
9             bmp.Height), ImageLockMode.WriteOnly, bmp.PixelFormat);
10    }
11    public void Dispose() => _bmp.UnlockBits(_bmpData);
12    public IBitmapEnumerable GetEnumerable() => new BitmapEnumerable(_bmp);
13    public bool MoveNext() => return true;
14    public void Reset() {}
15    public Position2d<int, byte> Current {
16        set {
17            unsafe {
18                var currentLine = (byte*)_bmpData.Scan0 +
19                    (value.J * _bmpData.Stride);
20                currentLine[value.I * _pixelSize] = unchecked(value.Data);
21            }
22        }
23    }
24 }
```



Frontend

Backend





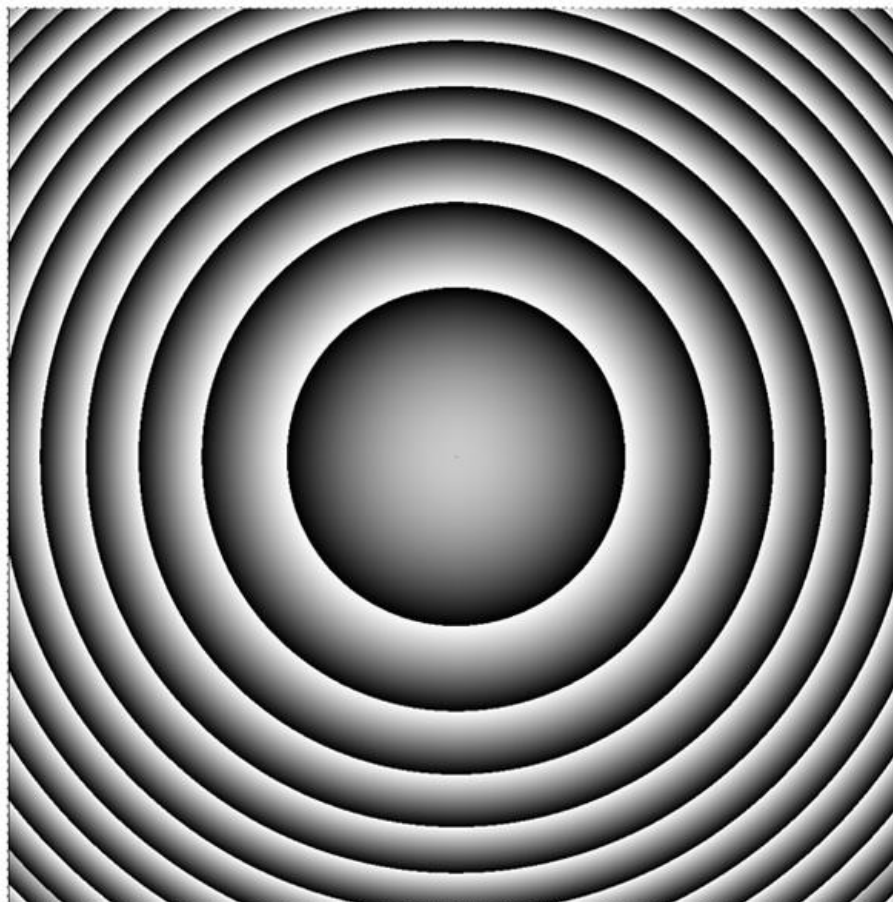
Задайте тип ДОЭ

Дифракционная сферическая линза ▼

Задайте параметры ДОЭ

Параметр	Значение
Коэффициент	0.000001
Разрешение X	1024
Разрешение Y	1024
Длина волны	1.06
Фокусное расстояние	1200000
Радиус апертуры	2000
1 row selected	1–6 из 6 < >

РАССЧИТАТЬ



Copyright © <https://doeris.com> 2025.



Дифракционная цилиндрическая линза

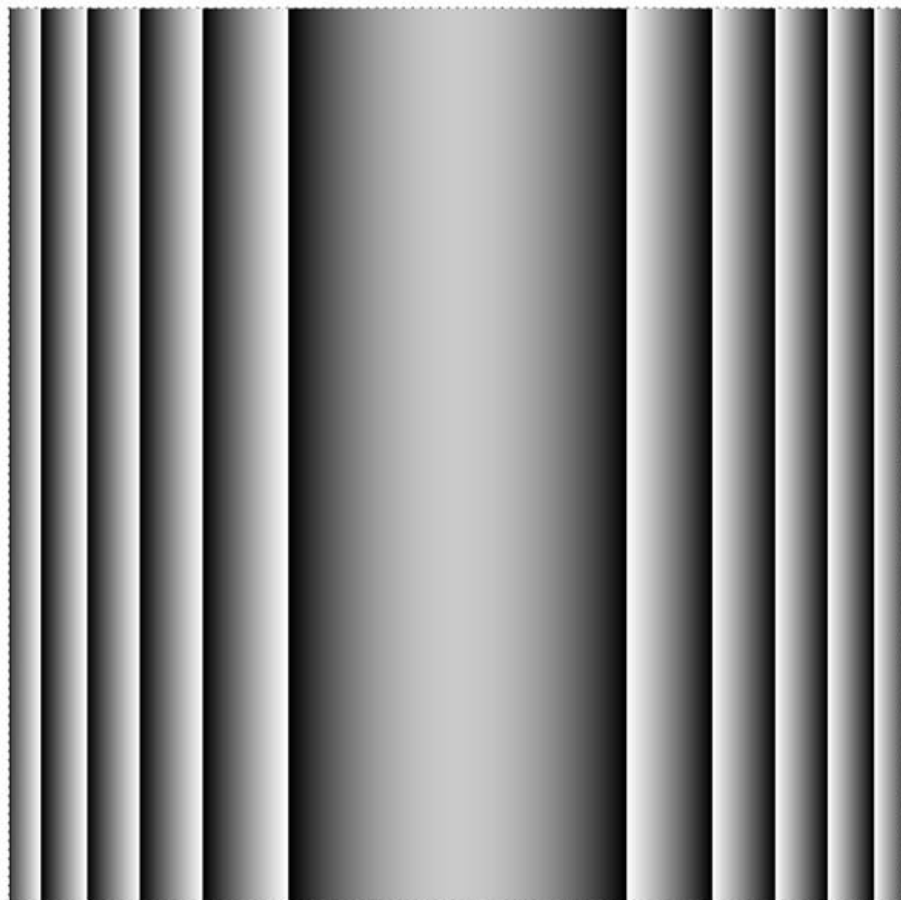
Задайте тип ДОЭ

Дифракционная цилиндрическая линза ▼

Задайте параметры ДОЭ

Параметр	Значение
Коэффициент	0.000001
Разрешение X	1024
Разрешение Y	1024
Длина волны	1.06
Фокусное расстояние	1200000
Радиус апертуры	2000
1 row selected	1–6 из 6 < >

РАССЧИТАТЬ



Copyright © <https://doeris.com> 2025.



Радиально-симметричный аксикон

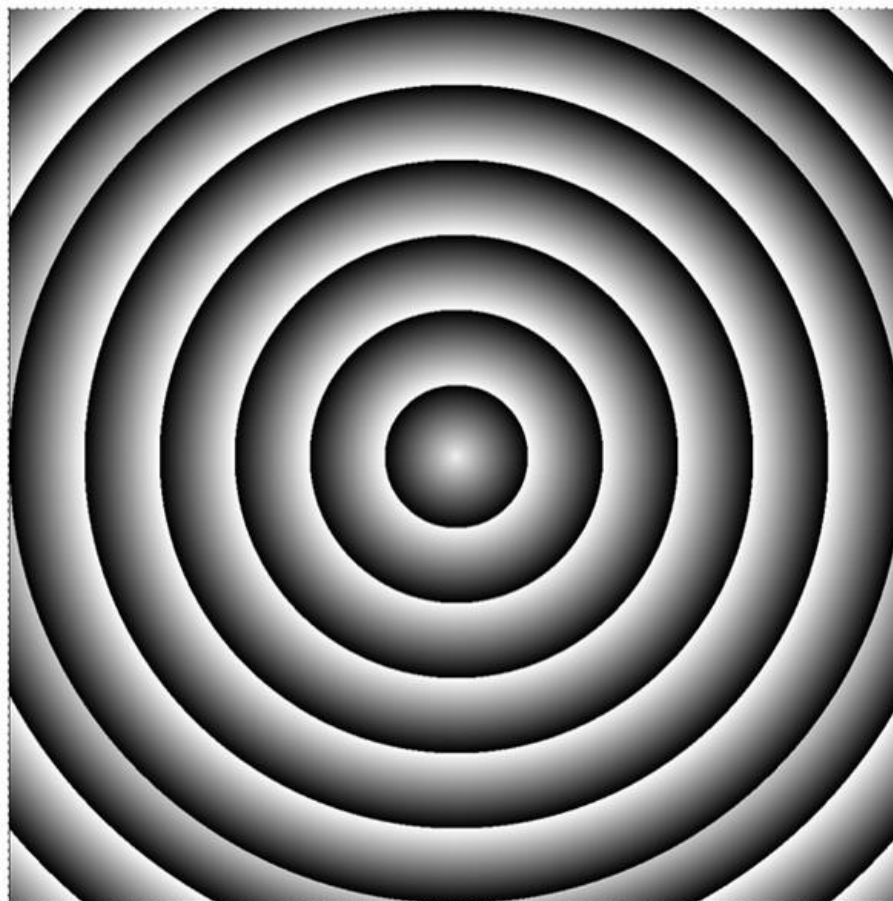
Задайте тип ДОО

Радиально-симметричный аксикон ▼

Задайте параметры ДОО

Параметр	Значение
Коэффициент	0.000001
Разрешение X	1024
Разрешение Y	1024
Полуугол при верш...	1.03
Радиус апертуры	2000
1 row selected 1–5 из 5 < >	

РАССЧИТАТЬ



Copyright © <https://doeris.com> 2025.



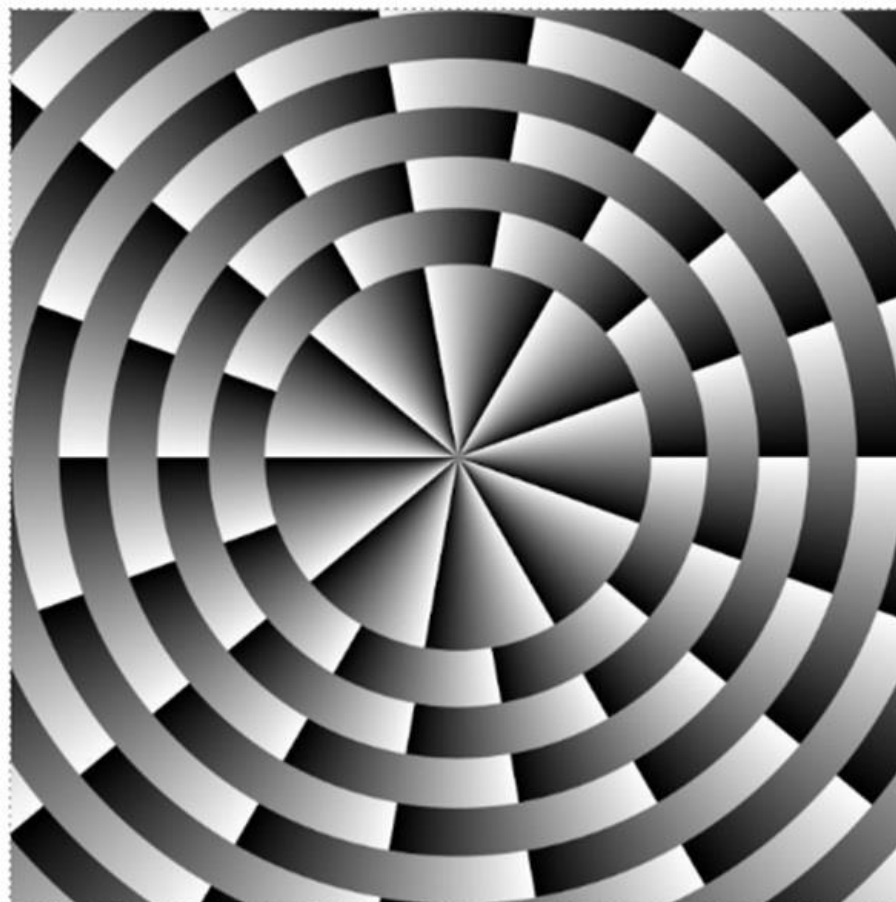
Задайте тип ДОЭ

Голографический аксикон

Задайте параметры ДОЭ

Параметр	Значение
Коэффициент	0.000001
Разрешение X	1024
Разрешение Y	1024
Параметр аксикона	3.094
Порядок	9
Радиус апертуры	2000
1-6 из 6 < >	

РАССЧИТАТЬ



Copyright © <https://doeiris.com> 2025.



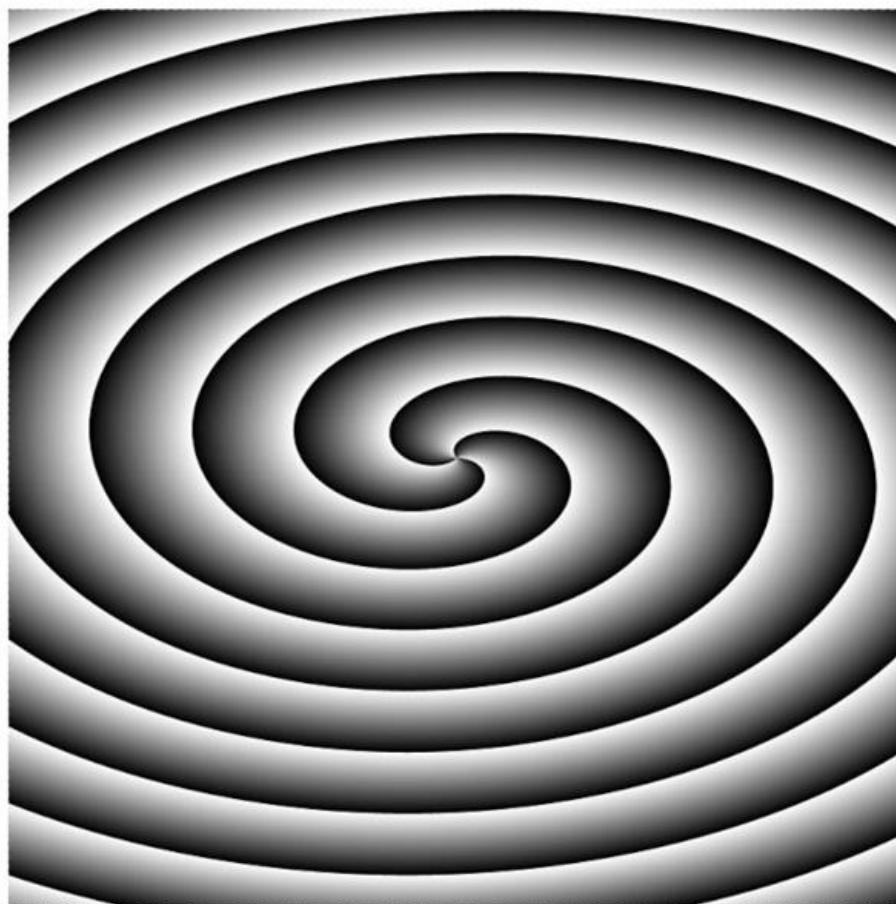
Задайте тип ДОЭ

Киноформный аксикон

Задайте параметры ДОЭ

Параметр	Значение
Коэффициент	0.000001
Разрешение X	1024
Разрешение Y	1024
Параметр аксикона	4.57
Порядок	3
Радиус аппертуры	2000
1–6 из 6 < >	

РАССЧИТАТЬ



Copyright © <https://doeris.com> 2025.



Generic parallel algorithm (multi-paradigm approach)

```
1  OE InnerTransformParallel<OE, T1, T2, R> (this IOutputEnumerable<OE, R> target,  
2      IEnumerable<T1> source1, IEnumerable<T2> source2, Func<T1, T2, R> operation)  
3  where OE : IOutputEnumerable<OE, R>  
4  {  
5      using var output = target.GetEnumerator();  
6      var exit = false;  
7      Parallel.ForEach(source1, (value1, state) =>  
8      {  
9          if (exit)  
10             state.Break();  
11         Parallel.ForEach(source2, (value2, state) =>  
12         {  
13             if (!output.MoveNext())  
14             {  
15                 exit = true;  
16                 state.Break();  
17             }  
18             output.Current = operation(value1, value2);  
19         });  
20     });  
21  
22  
23     return output.GetEnumerator();  
24 }
```

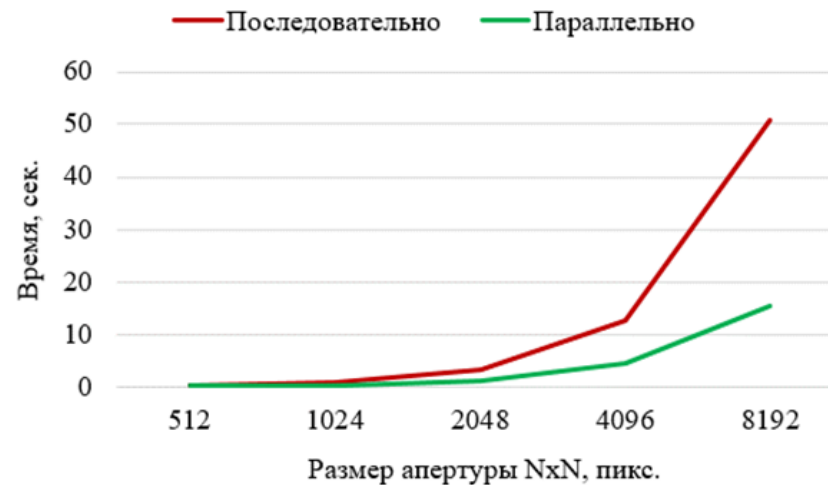
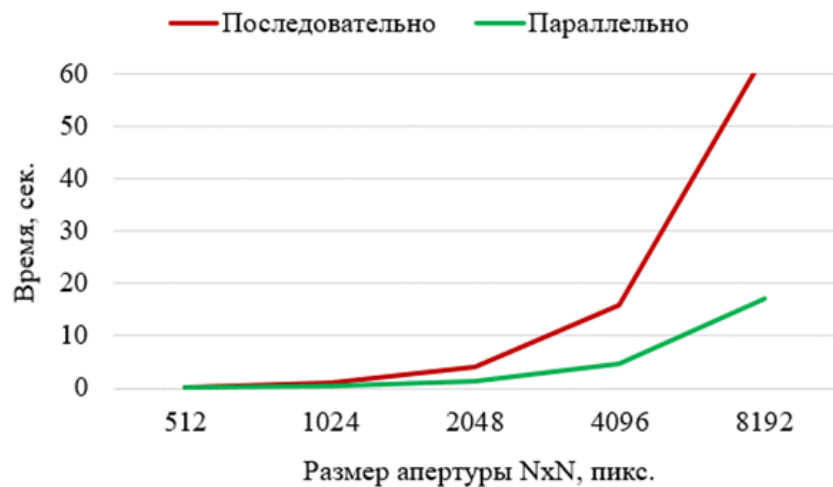
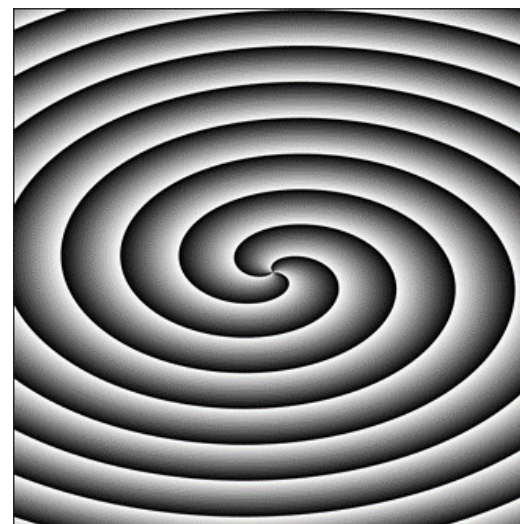
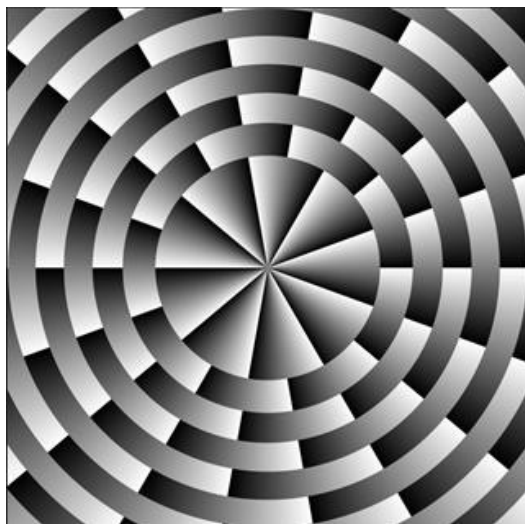
1

2

3



Результаты параллельного расчёта и рост сложности



Физико-технический институт имени А.Ф.Иоффе
Российской Академии Наук

Банк «РОССИЯ»



ДИПЛОМ

лауреата премии имени
Ю.И.Островского

за лучшие научные работы
в области оптической голографии
и интерферометрии

Вторая премия 2025 г. присуждена за работу:

**«Исследование вихревых бесселевых пучков
терагерцового излучения, сформированных
дифракционными оптическими элементами,
для задач телекоммуникации»**

Авторы: Н.Д. Осинцева, Б.А. Князев, В.С. Павельев,
В.В. Герасимов, К.Н. Тукмаков, М.С. Комленок, Ю.Ю. Чопорова,
О.Э. Камешков, А.С. Решетников, Д.Е. Яблоков

Председатель комиссии
по присуждению премии им. Ю.И. Островского

С.В. БОБАШЕВ

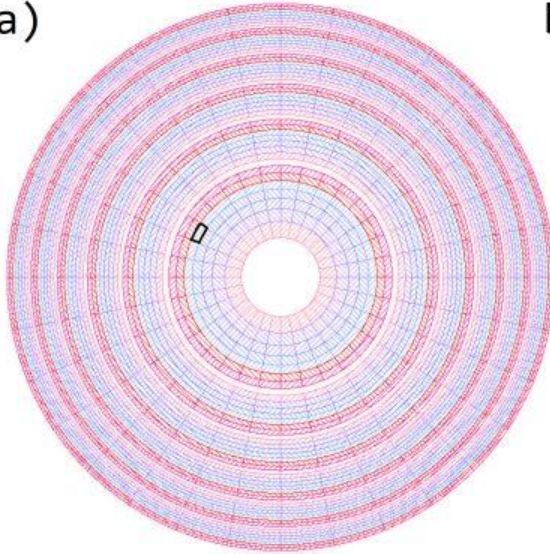
Ученый секретарь комиссии

И.В. СЕМЕНОВА



GDSII (Boundary)

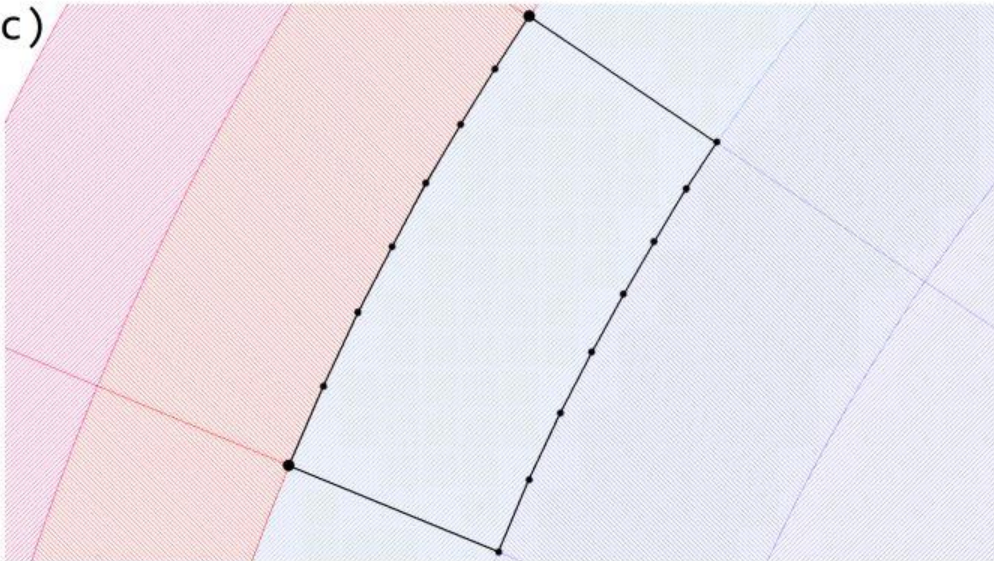
a)



b)

```
HEADER 600
BGNLIB 03/14/2021 13:39:17 03/14/2021 13:39:17
LIBNAME ITNT2021
UNITS 0.001 1e-09
BGNSTR 03/14/2021 13:39:17 03/14/2021 13:39:17
STRNAME FRESNEL
...
BOUNDARY
LAYER 5
DATATYPE 1
XY -1164967:482545 -1276158:528602 -1257923:570639
-1239687:609240 -1221452:645021 -1203217:678430
-1184982:709800 -1166747:739392 -1148512:767411
-1048443:700547 -1065089:674969 -1081735:647956
-1098382:619319 -1115028:588821 -1131675:556157
-1148321:520920 -1164967:482545
ENDEL
...
```

c)

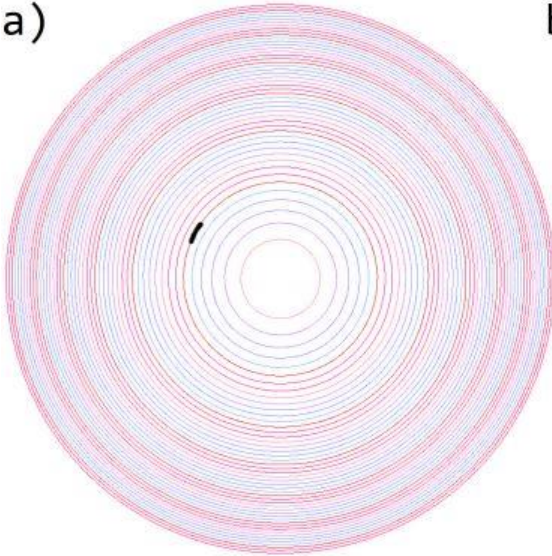


- a) Binary representation.
- b) Text representation.
- c) Selected record.



GDSII (Path)

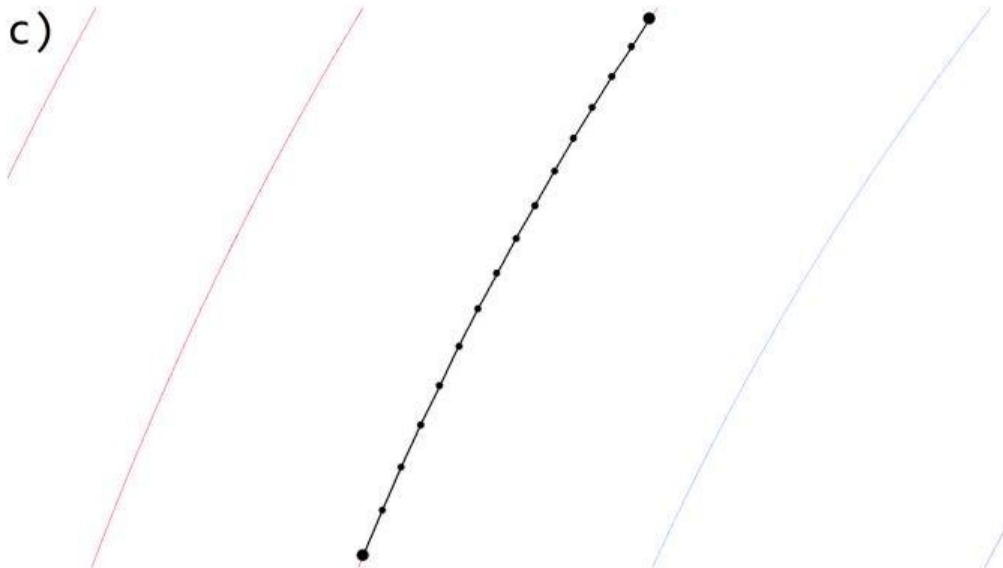
a)



b)

```
HEADER 600
BGNLIB 04/14/2021 18:32:15 04/14/2021 18:32:15
LIBNAME ITNT2021
UNITS 0.001 1e-09
BGNSTR 04/14/2021 18:32:15 04/14/2021 18:32:15
STRNAME FRESNEL
...
PATH
LAYER 6
DATATYPE 1
XY -1276158:528602 -1267648:548695 -1259138:567952
-1250629:586452 -1242119:604267 -1233609:621455
-1225099:638067 -1216590:654147 -1208080:669732
-1199570:684858 -1191060:699552 -1182551:713843
-1174041:727754 -1165531:741307 -1157021:754520
-1148512:767411
ENDEL
...
```

c)



- a) Binary representation.
- b) Text representation.
- c) Selected record.



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

THANK YOU

Sergey Vostokin
vostokin_sv@ssau.ru

Vladimir Pavelyev
nano@ssau.ru

Denis Yablokov
dyablokov.SSAU@gmail.com

GRID'2025