Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра математичного та комп'ютерного моделювання

Дипломна робота

на здобуття ступеня вищої освіти «магістр»

на тему: **«Ефективні методи розв'язування частинних випадків айконального рівняння»**

«Effective methods for solving partial cases of eikonal equation»

Виконала: студентка денної форми навчання
спеціальності 113 Прикладна математика
Прокоф'єва Софія Валентинівна

Керівник: канд. фіз.-мат. наук, доц. Таірова М. С.
Рецензент: канд. фіз.-мат. наук, доц. Васильєв О. Б.

Рекомендовано до захисту:

Протокол засідання кафедри

№ ___ від _____ 2021 р.

Завідувач кафедри

_____ _____

Захищено на засіданні ЕК № _____

Протокол № ___ від _____ 2021 р.

Оцінка _____ / _____ / _____

Голова ЕК

_____ _____

Одеса — 2021 р.

Odesa I. I. Mechnikov National University
Faculty of Mathematics, Physics and Information Technology
Department of Mathematical and Computer Modeling

Master thesis

«Effective methods for solving partial cases of eikonal equation»

Fulfilled by: full-time student
specialty 113 Applied Mathematics
Sofiia Prokofieva

Supervisor: Ph.D. Mariia Tairova
Reviewer: Ph.D. Oleksandr Vasyliev

Odesa — 2021

ЗМІСТ

ВСТУП

Айкональне рівняння - це нелінійне диференціальне рівняння, що виводиться із рівнянь Максвелла і пов'язує хвильову оптику з геометричною оптикою. Це рівняння має декілька фізичних інтерпретацій, серед яких задача пошуку найкоротшого шляху та електромагнітний потенціал.

Частковий випадок айконального рівняння можна розглядати з точки зору оптимального керування - за найкоротший час пройти відстань від однієї точки до іншої. Для цього рівняння можна розв'язати геометрично. Такий підхід потребує відносно низьку обчислювальну потужність, тому застосовується в багатьох сферах, наприклад, у комп'ютерній графіці. Як правило, айкональне рівняння використовується для 3-вимірного простору.

4-вимірний простір використовується в багатьох галузях та набуває все більшого розвитку. Не так давно стала популярна 3-вимірна печать, а вже зараз активно розвивають та починають використовувати 4-вимірну печать [1]. У таких галузях як комп'ютерна томографія [2] або геологія, де досліджують активність вулкану [3], для моделі використовують час як четвертий простір. Бази даних також використовують 4-вимірну модель та час для опису об'єкту [4]. Візуалізація 4-вимірного простору необхідна для дослідження фракталів та кватерніонів [5]. Також вона використовується для розробки ігор.

У цій роботі було розглянуто частковий випадок айконального рівняння, що виникає із спеціального набору функцій, що описують граничні умови та виділення комплексного варіанта нелінійних перетворень у багатовимірному просторі. Реалізація методу була виконана для 4-вимірного простору. Цей метод можна також використовувати у многовимірному просторі.

Матеріали даної роботи були представлені на IX-ій міжнародній науково-практичній конференції «Results of modern scientific research and development», 14-16 листопада 2021 р., Мадрид [6].

INTRODUCTION

The eikonal equation is a non-linear partial differential equation. It is derivable from Maxwell's equations of electromagnetics, and provides a link between physical optics and geometric optics. This equation has several physical interpretations, including the shortest-path problems and the electromagnetic potential.

The partial case of eikonal equation can be considered from the optimal control point of view of - to pass the distance from one point to another in the shortest time. Eikonal equation can be solved geometrically. This approach requires relatively low computing power, so it is used in many areas, such as computer graphics. Typically, eikonal equation is used for 3-dimensional space.

4D is used in many industries and it is becoming more and more widespread. 3D printing has recently become popular, and 4D printing is already being actively developed and used [1]. Time is used as the fourth space for the model in such areas as computed tomography [2] or geology, which studies the activity of volcano [3]. Databases visualization also may use a multidimensional models to describe the information objects in data mining [4]. Visualization of 4-dimensional space is usefull for the study complex and quaternion functions and fractals [5]. It is also used in developing games.

In this work was considered a special case of the eikonal equation that arises from a special set of functions that describe the boundary conditions and the selection of comprehensive variant of nonlinear transforms in a multidimensional space. In addition, the implementation of the method was carried out only for 4D, although it could be used for spaces of other dimensions.

The results of the work were presented at the conference Proceedings of IX international scientific and practical conference "Results of modern scientific research and development", Madrid, Spain, 2021 [6].

results showed that the average number of frames decreases more slowly with increasing nonlinearity than in the first test.

ВИСНОВКИ

У рамках цієї роботи був розглянутий частковий випадок айконального рівняння та проаналізовані існуючі методи його розв'язання. Також був розглянутий розширений метод трасування сфер (sphere tracing) для нелінійних трансформацій. У роботі запропоновано спосіб використання розширеного способу для 4-мірного простору та продемонстровано на практиці.

Для розширення методу трасування сфер до 4-мірного використовуються відповідні вдосконалені версії неявних функцій, які описують примітивні границі.

Для нелінійного трасування сфер на кожному кроці ітерації розглядається крайова задача, де кожен крок є звичайним диференціальним рівнянням першого порядку з початковою умовою результату попереднього кроку. Диференціальне рівняння виглядає як функція від точки, в якій перераховується значення нормального вектора. Для цього використовується добуток зворотньої матриці Якобі та вектора, який задає направлення променю. Універсальний підхід не є доцільним, оскільки виконує забагато обчислень у випадку, коли промінь знаходиться доволі близько до границі або не може відтворити викривлення променю достатньо точно. Тому було обрано змішаний підхід – за замовчуванням використовувати будь-який доволі точний метод розв'язання звичайних диференціальних рівнянь та у випадку, коли відстань між границею та променем є доволі малою – використовувати метод Ейлера.

Розширений метод трасування сфер був реалізований як шейдер на мові GLSL. Для матриці Якобі використовувався спеціальний параметр, який задає вплив нелінійної трансформації на промінь. Для різних значень цього параметра було визначено середню та максимальну кількість кадрів за секунду під час роботи шейдеру. Під час тесту було виявлено, що навіть незначний вплив нелінійної трансформації значно зменшує швидкість роботи шейдеру. Чим більше значення параметру, тем менша середня кількість кадрів за секунду. При значенні параметру більше ніж 0.65 шейдер працює дуже повільно.

Також був реалізований шейдер для нескінченних періодичних границь. Швидкість його роботи, якщо значення параметра нелінійності дорівнює нулю, вдвоє менша, ніж при відсутності нескінченних періодичних границь. Результати тесту показали, що середня кількість кадрів при збільшенні параметру нелінійності падає повільніше, ніж у першому тесті.

СПИСОК ЛІТЕРАТУРИ

1. Javaid M. 4D printing applications in medical field: A brief review / M. Javaid, A. Haleem // — Clinical Epidemiology and Global Health. — 2019. — V.7, №3. — P. 317—321.
2. Kwong Y. Four-dimensional computed tomography (4DCT): A review of the current status and applications / Y Kwong., A. O. Mel, G. Wheele, J. M. Troupis // —Journal of Medical Imaging and Radiation Oncology. — 2015. — P. 545—554.
3. Polacci M. Crystallisation in basaltic magmas revealed via in situ 4D synchrotron X-ray microtomography / M. Polacci, F. Arzilli, G. La Spina, N. Le Gall, B. Cai, M. E. Hartley, D. Di Genova, N. T. Vo, S. Nonni, R. C. Atwood, E. W. Llewellin, P. D. Lee, M. R.Burton // — Scientific Reports. — 2018.
4. Irmak E. Concept of 4th Dimension for Databases / Irmak E., Kurtuldu Ö. // — 14th International Conference on Machine Learning and Applications (ICMLA). — 2015. — P. 1159—1162.
5. Raytracing 4D fractals, visualizing the four dimensional properties of the Julia set [Електронний ресурс]. — 2008. — Режим доступу до ресурсу: `http://www.codinginstinct.com/2008/11/raytracing-4d-fractals-visualizing-four.html`
6. Prokofieva S. Methods for solving partial types of the eikonal equation / S. Prokofieva // — Madrid: Barca Academy Publishing, Madrid, Spain, 2021. — P. 192—194.
7. Sethian, J. A. A fast marching level set method for monotonically advancing fronts / J. A.Sethian // — Proceesings of the National Academy of Sciences. — 1969. — V.93, №4. — P. 1591—1595.
8. Gómez J. V. Fast Methods for Eikonal Equations: An Experimental Survey / J. V.Gómez, S. Garrido, D. Alvarez, L. E. Moreno // — IEEE Access. — 2019.
9. Dijkstra E. A note on two problems in connection with graphs / E. Dijkstra // — Numerische Mathematik — 1959. — V.1 — P. 269-271.
10. Zhao H. Fast sweeping method for eikonal equations / H. Zhao // —

Mathematics of Computation — 2005. — V.74 — P. 603-627.

11. Seyb D. Non-linear sphere tracing for rendering deformed signed distance fields / D. Seyb, A. Jacobson, D. Nowrouzezahrai, W. Jarosz // — ACM Trans. Graph. — 2019. — V.38, №6. — P. 12
12. Barr A. H. Global and Local Deformations of Solid Primitives / A. H. Barr // — Computer Graphics (Proceedings of SIGGRAPH). — 1984. — V.18, №3. — P. 21–30.
13. Signed distance function [Електронний ресурс]. — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Signed_distance_function
14. Tomczak L. J. GPU Ray Marching of Distance Fields / L. J. Tomczak // — Technical University of Denmark. — 2012.
15. Perlin K. Hypertexture / K. Perlin, E. M. Hoffert // — In ACM SIGGRAPH Computer Graphics. — 1989. — V23, №3. — P. 253-262.
16. Wright D. Dynamic Occlusion with Signed Distance Fields. Advances in Real-Time Rendering / D. Wright // — SIGGRAPH. — 2015.
17. Distance functions [Електронний ресурс]. — Режим доступу до ресурсу: https://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm
18. Taubin G. Distance Approximations for Rasterizing Implicit Curves / G. Taubin // — ACM TransactIons on Graphics. — 1994. — V.13, 1. — P. 3—42.
19. Euler's method [Електронний ресурс]. — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Euler_method
20. Heun's method [Електронний ресурс]. — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Heun
21. Runge-Kutta's method [Електронний ресурс]. — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods

APPENDIX A

Code fragments for shader, written in GLSL:

```
#define ANTIALIASING 3

#define RAY_STEP 200
float minIn(in vec4 r){
    float m=r.x;
    if(m>r.y)m=r.y;
    if(m>r.z)m=r.z;
    if(m>r.w)m=r.w;
    return m;
}

float maxIn(in vec4 r){
    float M=r.x;
    if(M<r.y)M=r.y;
    if(M<r.z)M=r.z;
    if(M<r.w)M=r.w;
    return M;
}

float sdPlane( vec4 p ) {
   return p.y;
}

float sdSphere( vec4 p, float r ) {
    return length(p)-r;
}

float sdEllipsoid( in vec4 p, in vec4 r) {
    return (length(p/r) - 1.) * minIn(r);
}

float sdTesseract( vec4 p, vec4 b ) {
    vec4 d = abs(p) - b;
    return min(maxIn(d),0.0) + length(max(d,0.0));
}
```

```
float udRoundTesseract( vec4 p, vec4 b, float r) {
    return length(max(abs(p)-b,0.0))-r;
}

float sdCapsule( vec4 p, vec4 a, vec4 b, float r ) {
    vec4 pa = p - a;
    vec4 ba = b - a;
    float h = clamp( dot(pa,ba)/dot(ba,ba), 0.0, 1.0 );
    return length( pa - ba*h ) - r;
}

float sdCylinder(in vec4 p,in vec2 h ) {
    return max(length(p.xzw)-h.x, abs(p.y)-h.y );
}

float sdCone(in vec4 p,in vec2 h ) {
    return max( length(p.xzw)-h.x, abs(p.y)-h.y) - h.x*p.y;
}

float sdCubicalCylinder(vec4 p, vec3 rh1h2) {
    vec3 d = abs(vec3(length(p.xz), p.y, p.w)) - rh1h2;
    return min(max(d.x,max(d.y,d.z)),0.) + length(max(d,0.));
}

float sdDuoCylinder( vec4 p, vec2 r1r2) {
    vec2 d = abs(vec2(length(p.xz),length(p.yw))) - r1r2;
    return min(max(d.x,d.y),0.) + length(max(d,0.));
}

float nonlinear_param = 0.65;

vec4 getNormVector(in vec4 rd, in vec4 point){
    mat4 jacobian = mat4(1.0, nonlinear_param*cos(point.y), 0., 0.,
                         nonlinear_param*cos(point.x), 1.0, 0., 0.,
                         0., 0., 1.0, 0.,
                         0., 0., 0., 1.0);

    mat4 inverse_jacobian = inverse(jacobian);
    vec4 norm_vector = inverse_jacobian*rd;
    norm_vector = normalize(norm_vector);
```

```
    return norm_vector;
}

vec4 heun (in vec4 init, in float dist, in vec4 rd, in int maxstep){
    vec4 res = init;
    float h = dist/float(maxstep);
    vec4 k1, k2;
    for (int i=0; i<maxstep; i++)
    {
        k1 = getNormVector(rd, res);
        k2 = getNormVector(rd, res+h*k1);
        res = res + h*(k1/2.0+k2/2.0);
    }
    return res;
}

vec4 rungeKutta4 (in vec4 init, in float dist, in vec4 rd, in int maxstep){
    vec4 res = init;
    float h = dist/float(maxstep);
    vec4 k1, k2, k3, k4;
    for (int i=0; i<maxstep; i++)
    {
        k1 = getNormVector(rd, res);
        k2 = getNormVector(rd, res+h*k1/2.0);
        k3 = getNormVector(rd, res+h*k2/2.0);
        k4 = getNormVector(rd, res+h*k3);
        res = res + (k1+2.0*k2+2.0*k3+k4)*(h/6.0);
    }
    return res;
}

vec4 euler(in vec4 init, in float dist, in vec4 rd, in int maxstep){
    vec4 res = init;
    float h = dist/float(maxstep);
    for (int i=0; i<maxstep; i++)
    {
        res = res + h*getNormVector(rd, res);
    }
    return res;
}
```

```
float epsilon = 0.0005;
float a = 3.0;

vec2 castRay( in vec4 ro, in vec4 rd, in float maxd ) {
    float t = 2.0;
    float h=epsilon*2.0;
    vec2 res;
    vec4 point = ro+rd*t;
    for( int i=0; i<RAY_STEP; i++ ) {
        if (abs(h)<epsilon || t>maxd ) break;
        res = map(point);
        h = res.x;
        if (epsilon*a > abs(h))
        {
            point = euler(point, h, rd, 10);
        }
         else
        {
            point = heun(point, h, rd, 10);
        }
         t+=h;
    }
    return vec2( t, t>=maxd ? -1. : res.y );
}
```