

UOT 532.5.013.2

R.S. QULUYEVA

MAYENİN HƏRƏKƏTİNİN İMITASIYASI MƏQSƏDİLƏ MAKROHİSSƏCİKLƏRİN VƏ MÜHİTİN PROQRAM OBYEKTİ KİMİ MODELLEŞDİRİLMƏSİ

Kompüter realizasiyası məqsədi ilə maye makrohissəciklərinin və onların hərəkət etdiyi relyefin proqram obyektı şəklində təsvir olunması məsələsinə baxılmışdır. Aparılan ədədi eksperimentlərin nəticələri analizi göstərir ki, təklif olunan yanaşmanın tətbiqi real maye selini hərəkətini kifayət qədər adekvat təsvir etməyə imkan verir.

Açar sözlər: Laqranj modeli, makrohissəcik, obyekt, mayenin axını

1. Giriş. Təhlükəli mayelərin daşınması marşrutu üzrə mümkün qəzalar zamanı maye axınının çirkləndirə biləcəyi ərazinin qiymətləndirilməsi qabaqlayıcı tədbirlər görmək baxımından çox vacib sayılır. Bu məsələ, ilk növbədə, müəyyən mənbədən başlayan axının mühitdə hərəkətinin təsvir olunmasını tələb edir. Axın oblastı əvvəlcədən məlum olmadığından, məsələnin modelinin klassik hidrodinamika tənliklərinin köməyi ilə yazılmasında çoxlu çətinliklər meydana gəlir. Ona görə də, belə hallarda mayenin hərəkətinin makrohissəciklər axını kimi modelləşdirilməsinə üstünlük verilir [1, s.459; 2; 3].

Maye hərəkətinin makrohissəciklər seli kimi təsvir olunmasına bir sıra müəlliflər tərəfindən baxılmış, müxtəlif metodlar təklif olunmuşdur. Onların sırasında hamarlanmış hissəciklər metodu [4, s.375], hərəkətdə olan hissəciklərin yarı-gizli metodu [5, s.751], Laqranj-Eyler hissəcikləri metodu [6], şəbəkədən kənar sonlu elementlər metodu [7, s.655], maye hissəciklərinin makrohissəciklər seli kimi modelləşdirilməsini [8, s.29] göstərmək olar. Məqalədə axının vizualizasiyası məsələsi qarşıya qoyulmuşdur. Bu məqsəd ilə biz əsasən [8, s.29] işinə istinad edəcəyik. Mayenin relyef üzrə hərəkətini tədqiq etmək məqsədilə, baxılan işdə onun kürə formalı makrohissəciklər seli kimi modelləşdirilməsi təklif olunmuşdur. İlk ədədi eksperimentlər göstərmişdir ki, belə kürələrin birgə hərəkəti mayenin dinamik axınının keyfiyyət mənzərəsinə uyğun gəlir. Ədəbiyyatda Laqranj yanaşması kimi tanınan bu yanaşmanın geniş miqyaslı tətbiqi üçün maye makrohissəciklərinin hərəkətini imitasiya edən proqram obyektlərinin yaradılması məqsədəuyğundur.

Beləliklə, aşağıda kompüter realizasiyası məqsədi ilə maye makrohissəciklərinin və onların hərəkət etdiyi relyefin proqram obyektı şəklində təsvir olunması məsələsinə baxılır.

2. Məsələnin qoyuluşu. Laqranj yanaşmasına uyğun olaraq, hər bir makrohissəcik baxılan anda fəzada tutduğu yerə uyğun koordinatları və hərəkət sürəti ilə xarakterizə olunur. Klassik mexanika prinsiplərinə uyğun olaraq, makrohissəcik hər bir anda ona təsir edən qüvvələrin məcmusuna uyğun olaraq aldığı təcillə sürətini dəyişir. Makrohissəciklərin birgə hərəkəti zamanı onların bir-biri ilə və mühitlə qarşılıqlı təsirdə meydana çıxan qüvvələri hesablama qaydaları məlum olmalıdır. Hesab edəcəyik ki, makrohissəciklər özlərini kürə formasında olan bərk cisim kimi aparır və ona qravitasiya qüvvəsi təsir edir, bir-biri ilə təmasda olduqda isə, onların arasında elastiklik, yapışma, daxili sürtünmə qüvvələri meydana çıxır. Qüvvələrin hesablama qaydasını [8, s.30-31]-də olduğuna analogi olaraq qəbul edəcəyik. Beləliklə, qarşılıqlı təsir qüvvələri [8, s.30-31] düsturları ilə təyin olunan "Hissəcik" və "Mustevi" obyektlərinin yaradılması tələb olunur.

3. Makrohissəciklərin proqram obyektı kimi təsviri. Məsələnin həlli C# proqram mühitində realizə olunmuşdur. Ona görə də onun şərhli zamanı proqramdan uyğun fraqmentlərə müraciət edəcəyik.

Makrohissəciyi təsvir edən əsas klass aşağıdakı kimi yazıla bilər:

```
Class Hissəcik
{
public Vektor koordinat;
```

```
public double radius = 0.005;
public int r = 1;
public Vektor suret;
public static int k = 15;
public static int mu;
public static doublesiqma = 0.147;
public static doubletetta = 8.84 * 0.001;
...
}
```

Burada koordinat və suret dəyişənləri hissəciyin (x, y, z) koordinatlarını və $v = (v_x, v_y, v_z)$ sürət vektorunu təyin edir. Vektor vektorlar və onlar üzərində əməliyyatları təyin etmək üçün yaratdığımız klassdır. Sadəlik üçün makrohissəciklərin radiusları eyni olmaqla "radius=0.005" götürülmüşdür, ümumi halda müxtəlif hissəciklər müxtəlif radiuslara malik ola bilər. Ekranda təsvir olunmuş hissəciyin radiusunu vermək üçün $r = 1$ tam ədədindən istifadə edəcəyik. Verilmiş μ , k , σ , τ ədədləri isə makrohissəciyə təsir edən qüvvələri hesablayarkən istifadə olunan sabitlərdir (əmsallardır).

Makrohissəciyi yaradarkən və ekranda təsvir edərkən aşağıdakı funksiyalardan istifadə edilir:

```
public Hissecik(double x, double y, double z)
{
    koordinat = newVektor(x, y, z);
}

public void Sekil_cek(Graphics g, Vektorkoor)
{
    Pen myPen = new Pen(Brushes.Red);
    myPen.Width = 2f;
    Rectangle rectangle = new Rectangle(Convert.ToInt32(koor.x) - r,
    Convert.ToInt32(koor.z) - r, r * 2, r * 2);
    g.DrawEllipse(myPen, rectangle);
}
```

Burada Hissecik konstrukturu verilmiş koordinatlarda yeni obyekt yaradılan zaman avtomatik olaraq çağırılan funksiyadır. Sekil_cek funksiyası g qrafikasıdan istifadə edərək monitorda verilmiş koordinatlara uyğun dairə şəklini çəkir.

Yuxarıda qeyd edildiyi kimi, makrohissəciklərin hərəkəti [8] düsturları ilə təyin ediləcək. [8, s.31]-ə əsasən hesab edirik ki, mayenin axını zamanı hər bir hissəcik qravitasiya qüvvəsinin, həmçinin, səthlə və digər hissəciklərlə toxunma zamanı onların arasında yaranan qüvvələrin təsirinə məruz qalır və bu qüvvələrin təsiri ilə öz parametrlərini dəyişir. Həmin qüvvələr aşağıdakılardır:

Makrohissəciyə təsir edən qravitasiya qüvvəsi $F = mg = \rho V g$. Burada, $m = \rho V$ makrohissəciyin kütləsi, V – onun həcmi, ρ isə sıxlığıdır. Onda, uyğun vektor funksiyanı belə vermək olar:

```
public static Vektor Gravitasiya(Hissecik h)
{
    double hecm = Math.PI * Math.Pow(h.radius, 2);
    double sixliq = 1000;
    double grtecili = 9.8;
    double gr = -(grtecili * hecm * sixliq);

    return new Vektor(0, 0, gr);
}
```

Qarşılıqlı təsir zamanı makrohissəciklər arasında meydana çıxan qüvvə elastiklik, yapışma və daxili sürtünmə qüvvələridir. Bu qüvvələrin hesablanma düsturları aşağıdakı kimidir: baxılan makrohissəciyin i -ci makrohissəciklə toxunması zamanı yaranan elastiklik qüvvəsi

$$\Phi_i^{(1)} = -\frac{k}{2} \cdot (2R - |x^0(t) - x^i(t)|) \cdot n^i(t),$$

ilə hesablanır. Burada, k – makrohissəciyin elastiklik əmsalı, n^i – makrohissəciyin i -ci makrohissəciklə toxunması zamanı toxunma səthinin normaldır.

$$\Phi_i^{(2)} = \pi\sigma\mu \cdot \sqrt{4R^2 - |x^0(t) - x^i(t)|^2} \cdot n^i(t),$$

baxılan makrohissəciyin i -ci makrohissəciklə toxunması zamanı yaranan yapışma qüvvəsini hesablayır. Burada, σ – səthi gərilmə əmsalı, n^i – makrohissəciyin i -ci makrohissəciklə toxunması zamanı toxunma səthinin normaldır, μ – normal ilə sürət vektoru arasında bucağı təyin edən əmsaldır.

$$\Phi_i^{(3)} = -\frac{1}{4}\pi\theta \cdot (4R^2 - |x^0(t) - x^i(t)|^2) \cdot q^i(t),$$

baxılan makrohissəciyin i -ci makrohissəciklə toxunması zamanı meydana çıxan daxili sürtünmə qüvvəsidir. Burada, θ – özlülük əmsalı, q^i – normal ilə i -ci hissəciyin sürət vektoru arasında bucağı təyin edən əmsaldır.

Makrohissəciklər arasında meydana çıxan qüvvə bu qüvvələrin cəminə bərabərdir:

$$\Phi = \sum_{i=1,2,\dots,N} (\Phi_i^{(1)} + \Phi_i^{(2)} + \Phi_i^{(3)}),$$

N – verilmiş anda baxılan hissəciklə qarşılıqlı təsirdə olan hissəciklərin sayıdır. Cəmləmə yalnız qarşılıqlı təsirdə olan hissəciklər üzrə aparılır. Bu funksiyanın proqram təsviri aşağıdakı şəkildədir:

```
public static Vektor Fi(Hissecik[] m, int index)
{
    Vektor[] n = new Vektor[Form1.say];
    Vektor[] q = new Vektor[Form1.say];
    double c;
    Vektor fi1 = new Vektor(0, 0, 0);
    Vektor fi2 = new Vektor(0, 0, 0);
    Vektor fi3 = new Vektor(0, 0, 0);

    for (int j = 0; j < Form1.say; j++)
    {
        if (index != j)
        {
            c = Vektor.VekUzunl(Vektor.VekFerq(m[index].koordinat,
m[j].koordinat));
            if (c != 0)
            if (c < 2 * m[index].radius)
            {
                n[j] =
Vektor.VekEdedHasil(Vektor.VekEdBol(Vektor.VekFerq(m[index].koordinat,
m[j].koordinat), c), -1);
                fi1 = Vektor.VekCem(fi1, Vektor.VekEdedHasil(n[j], (2 *
m[index].radius - c) * - k / 2));
                if (Vektor.Bucaq(n[j], m[j].suret) < 0)
                mu = 1;
                else mu = 0;
            }
        }
    }
}
```

```

        c = Math.Sqrt(4 * Math.Pow(m[index].radius, 2) -
Math.Pow(c, 2));
        fi2 = Vektor.VekCem(fi2, Vektor.VekEdedHasil(n[j], c *
Math.PI * mu * siqma));
        if (Vektor.BeraberVek(m[j].suret, 0))
            q[j] = new Vektor(0, 0, 0);
        else q[j] = Vektor.VekEdBol(Vektor.VekFerq(m[j].suret,
Vektor.VekHasil(Vektor.VekHasil(n[j], m[j].suret), n[j])),
Vektor.VekUzunl(m[j].suret));
        fi3 = Vektor.VekCem(fi3, Vektor.VekEdedHasil(q[j], c *
Math.PI * tetta * -1 / 4));
    }
}
return Vektor.VekCem(fi1, Vektor.VekCem(fi2, fi3));
}

```

Makrohissəciyin mühitlə qarşılıqlı təsir qüvvələri səthin elastiklik qüvvəsi, səthin yapışma qüvvəsi və daxili sürtünmə qüvvəsidir. Makrohissəciyin səthlə toxunması zamanı yaranan səthin elastiklik qüvvəsi

$$F^{(1)} = -2k(R - \delta) \cdot n,$$

ilə hesablanır. Burada, k – makrohissəciyin elastiklik əmsalı, n – makrohissəciyin səthlə toxunması zamanı toxunma səthinin normalıdır.

$$F^{(2)} = -2\pi\sigma\mu\sqrt{R^2 - \delta^2} \cdot n,$$

baxılan makrohissəciyin səthlə toxunması zamanı yaranan yapışma qüvvəsini hesablayır. Burada, σ – səthi gərilmə əmsalı, n – makrohissəciyin səthlə toxunması zamanı toxunma səthinin normalıdır, μ – normal ilə sürət vektoru arasında bucağı təyin edən əmsaldır.

$$F^{(3)} = -\pi\theta(R^2 - \delta^2) \cdot q,$$

toxunma zamanı meydana çıxan daxili sürtünmə qüvvəsidir. Burada, θ – özlülük əmsalı, q – normal ilə sürət vektoru arasında bucağı təyin edən əmsaldır.

Makrohissəciyin mühitlə qarşılıqlı təsir qüvvəsi bu qüvvələrin cəmi şəklində verilir:

$$F = F^{(1)} + F^{(2)} + F^{(3)},$$

Uyğun vektor funksiya aşağıdakı şəkildə olur:

```

public static Vektor F(Hissecik h, double betta, Vektor n)
{
    Vektor q = new Vektor(0, 0, 0);
    Vektor f1 = new Vektor(0, 0, 0);
    Vektor f2 = new Vektor(0, 0, 0);
    Vektor f3 = new Vektor(0, 0, 0);
    double c;
    if (h.radius > betta)
    {
        if (betta < 0) { c = 0; }
        else
            c = Math.Sqrt(Math.Pow(h.radius, 2) - Math.Pow(betta, 2));
        f1 = Vektor.VekCem(f1, Vektor.VekEdedHasil(n, 2 * k * (h.radius
- betta)));
        if (Vektor.Bucaq(n, h.suret) < 0)
            mu = 1;
        else mu = 0;
    }
}

```

```

        f2 = Vektor.VekCem(f2, Vektor.VekEdedHasil(n, c * -2 * Math.PI
* mu * siqma));
        if (!Vektor.BeraberVek(h.suret, 0))
            q = Vektor.VekEdBol(Vektor.VekFerq(h.suret,
Vektor.VekHasil(Vektor.VekHasil(h.suret, n), n)), Vektor.VekUzunl(h.suret));
            f3 = Vektor.VekCem(f3, Vektor.VekEdedHasil(q, -c * Math.PI *
tetta));
    }
    return Vektor.VekCem(f1, Vektor.VekCem(f2, f3));
}

```

Beləliklə, Gravitasiya funksiyası makrohissəciyə təsir edən qravitasiya qüvvəsini, F_1 funksiyası makrohissəciklərin bir-biri ilə qarşılıqlı təsiri zamanı yaranan qüvvələri, F funksiyası makrohissəciyin mühit ilə qarşılıqlı təsiri zamanı yaranan qüvvələri hesablayır.

4. Relyefin obyekt kimi verilməsi. Makrohissəciklərin relyefdə hərəkəti zamanı yaranan qüvvəni hesablamaq üçün həmin relyefi də obyekt kimi təyin etmək lazımdır. Sadəlik üçün bu məqalədə mayenin novda axın prosesinə baxılmışdır. Nova kəşişən iki müstəvi kimi baxılır (Şəkil 1) və bu müstəvilər aşağıdakı şəkildə proqram obyektini kimi təyin edilir:

```

class Mustevi
{
    public double a;
    public double b;
    public double c;
    public double d;
    double n, l;

    public Mustevi (double a, double b, double c, double d){
        this.a = a;
        this.b = b;
        this.c = c;
        this.d = d;
        n = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(b, 2) + Math.Pow(c, 2));
    }

    public double distance (double x, double y, double z){
        l = (a * x + b * y + c * z + d) / n;
        return l;
    }

    public Vektor getNormal()
    {
        Vektor v = new Vektor( a/n, b/n, c/n);
        return v;
    }
}

```

Mustevi klassı verilmiş a , b , c , d parametrlərinə görə müstəvi təyin edir. Burada $getNormal()$ müstəvinin normal vektorunu hesablayır, $distance(x, y, z)$ isə verilmiş nöqtədən müstəviyədək məsafəni təyin edən funksiyadır.

5. Maye axınının simulyasiyası. Maye axınının kompüterdə simulyasiyası üçün onun mənbəyi verilməlidir, bu da başlanğıc parametrlərin (mənbənin koordinatları, mənbədə maye hissəciklərinin başlanğıc sürəti) verilməsini tələb edir. Yuxarıda deyildiyi kimi, maye selinin nov üzrə

hərəkətinə baxılmışdır. Proqram modulunda novu təyin edən müstəviləri xarakterizə edən parametrlər aşağıdakı kimi verilmişdir:

```
public partial class Form1 : Form
{
    public static int a = 8;
    public static int b = 4;
    public static double alfa = (Math.PI / 4);
    public static int say = 400;
    Hissecik[] maye ;
    Mustevi omeqa1, omeqa2;
    double betta1, betta2;
    Vektor fsurface;
    Graphics g;
    int basl_say;
    System.Drawing.Pen myPen2, myPen1;
```

Burada a, b, alfa – müstəvini, say – generasiya ediləcək makrohissəciklərin sayını, maye hissəcik obyektləri yığını, omeqa1, omeqa2 – müstəvi obyektlərini, betta1, betta2 – hissəcikdən bu müstəvilərədək olan məsafələri, fsurface – bu müstəvilər ilə qarşılıqlı təsiri zamanı yaranan qüvvələrin cəmini xarakterizə etmək üçün istifadə olunur. g, myPen2, myPen1 isə təsvirləri ekrana gətirmək üçün istifadə edilən obyektlərdir. Proqram yüklənən zaman hər bir makrohissəciyin parametrlərini təyin etmək – başlanğıc sürəti və başlanğıc koordinatlarını vermək tələb olunur. Proqram modulunda bu dəyişənlər aşağıdakı kimi verilmişdir:

```
private void Form1_Load(object sender, EventArgs e)
{
    g = this.CreateGraphics();
    myPen1 = new System.Drawing.Pen(System.Drawing.Color.Green);
    myPen2 = new System.Drawing.Pen(System.Drawing.Color.Red);
    omeqa1 = new Mustevi(b, a * Math.Tan(alfa), a, -a * b);
    omeqa2 = new Mustevi(b, -a * Math.Tan(alfa), a, -a * b);
    omeqa3 = new Mustevi(0, 0, 1, 0);
    betta1 = 0; betta2 = 0;

    int rInt;
    maye = new Hissecik[say];

    Random r = new Random();
    int t = say / 10;
    for (int j = 0; j < say; j++)
    {
        maye[j] = new Hissecik(0, 0, 0);
        rInt = r.Next(-1, 2);
        if (rInt == 0) rInt = 1;
        maye[j].koordinat = new Vektor(r.NextDouble(), r.NextDouble()* rInt , b +
r.NextDouble() * Math.Tan(alfa));
        maye[j].suret = new Vektor(0, 0, 0 );
    }

    timer1.Interval = 1000;
    timer1.Enabled = true;
}
```

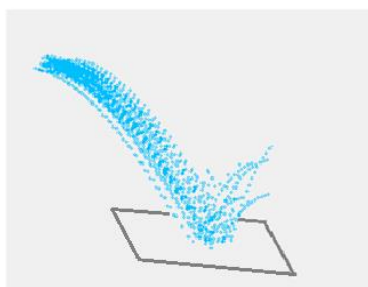
Makrohissəciyin başlanğıc koordinatlarından və hərəkət sürətindən çıxış edərək hər bir anda ona təsir edən qüvvələri hesablaya və onun hərəkəti zamanı yeni koordinatlarını və yeni sürətini təyin edə bilərik. Axın prosesini addım-ba-addım simulyasiya etmək üçün Timer1 obyektindən istifadə olunmuşdur:

```
private void timer1_Tick(object sender, EventArgs e)
{
    g = this.CreateGraphics();
    g.Clear(Form1.ActiveForm.BackColor);

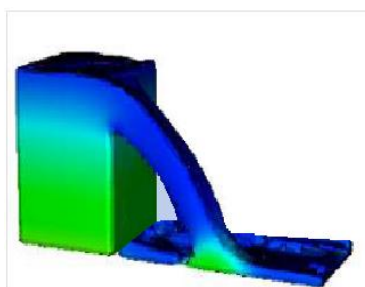
    basl_say = basl_say+15;
    if (basl_say>say)
        basl_say = say;

    for (int i = 0; i <basl_say; i++)
    {
        maye[i].Sekil_cek(g, Vektor.yeniKoor(maye[i].koordinat));
        fsurface1 = newVektor(0, 0, 0);
        betta1 = omeqa1.distance(maye[i].koordinat.x, maye[i].koordinat.y,
maye[i].koordinat.z);
        betta2 = omeqa2.distance(maye[i].koordinat.x, maye[i].koordinat.y,
maye[i].koordinat.z);
        fsurface1 = Vektor.VekCem(Hissecik.F(maye[i], betta2,
omeqa2.getNormal()), Hissecik.F(maye[i], betta1, omeqa1.getNormal()));
        maye[i].suret = Vektor.VekCem(maye[i].suret,
Vektor.VekCem(Hissecik.Fi(maye, i),
Vektor.VekCem(Hissecik.Gravitasiya(maye[i]), fsurface1)));
        maye[i].koordinat = Vektor.VekCem(maye[i].koordinat, maye[i].suret);
    }
    g.Dispose();
}
```

6. Nəticə. Proqram modullarının realizasiyası və aparılan ədədi eksperimentlər göstərdi ki, proqram obyektı şəklində modelləşdirilmiş maye makrohissəciklərin hərəkəti real maye selini təsvir etmək üçün istifadə oluna bilər. Şəkil 1-də borudan tökülən mayenin hərəkətinin tərtib olunan proqram modulu vasitəsi ilə imitasiyası verilmişdir. Modelləşdirmənin adekvatlığını təhlil etmək üçün axan mayenin təsviri 500 makrohissəcik götürməklə generasiya olunub.



Şəkil 1



Şəkil 2



Şəkil 3

Şəkil 2 də isə [9, s.101]-ə istinadən FLOW-3D proqramından istifadə etməklə alınmış kanaldan axan suyun bənd üzərindən tökülməsinin təsviri verilmişdir. Həmçinin, şəkil 3-də borudan tökülən suyun axını zamanı yaratdığı təbii mənzərə verilmişdir. Bu şəkillərin müqayisəsindən

görünür ki, proqramın işi nəticəsində alınan təsvirlər real maye axınını yaxşı imitasiya edir. Beləliklə, baxılan yanaşmanın tətbiqi real maye selini hərəkətini kifayət qədər adekvat təsvir edir.

Ədəbiyyat

1. Сивухин Д.В. Общий курс физики // В 5 т. Том I. Механика. М.: ФИЗМАТЛИТ; Изд-во МФТИ, 2005. - 560 с.
2. Зеркаль С.В., Захаров Е.В., Богомолов С.В. Моделирование движения потоков различной природы по наклонной поверхности методом частиц // Вісник Харківського національного університету Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» № 2, 2003.
3. Богомолов С.В., Захаров Е.В., Зеркаль С.В. Моделирование волн на мелкой воде методом частиц // Математическое моделирование. 2002. Т. 14., №. 3. с. 103–116.
4. Gingold R.A., Monaghan J.J. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Monthly Notices of the Royal Astronom. Society. 1977. Vol. 181, P. 375-389.
5. Koshizuka S., Nobe A., Oka Y. Numerical analysis of breaking waves using the moving particle semi-implicit method. International Journal For Numerical Methods in Fluid. 1998. P.751-769.
6. Франк А. М. Дискретные модели несжимаемой жидкости. М. Физматлит, 2001.
7. Idelsohn S., Onate E., Del Pin F. A lagrangian meshless finite element method applied to fluid structure interaction problems. Computer and Structures. 2003. Vol. 81.
8. Pashayev A.B., Sabziev E.N. Description of Fluid Flow as a System of Systems. Proc.of the 2013 8th International Conference on System of Systems Engineering, Maui, Hawaii, USA, June 2-6, 2013, pp.28-33.
9. Чижиумов С.Д. Основы гидродинамики. Комсомольск-на-Амуре. 2007. 108 с.

R.S. Guluyeva

Object modeling of macroparticles and environment for simulation of fluid flow

We consider the problem of object modeling of fluid macroparticles and the surface of their movement in order to implement it in a programming language. The analysis of the results of numerical experiments shows that the proposed approach can quite adequately describe the motion of a real fluid.

Keywords: Lagrangian model, macro-particle, object, fluid flow

УДК 532.5.013.2

Р.С. Гулуева

Моделирование макрочастиц и среды как программный объект с целью имитации движения жидкости

Рассмотрен вопрос описания в виде программного объекта макрочастиц жидкости и рельефа их перемещения с целью компьютерной реализации. Анализ результатов проведенных численных экспериментов показывает, что применение предложенного подхода позволяет адекватно описывать движение реального потока жидкости.

Ключевые слова: модель Лагранжа, макро-частица, объект, поток жидкости

AMEA İdarəetmə Sistemləri İnstitutu

Təqdim olunub 18.03.2016