УДК 004.94, 517.9
DOI https://doi.org/10.32782/pet-2024-1-10

Дмитро ШВАЛІКОВСЬКИЙ
інженер кафедри теоретичної та комп'ютерної фізики імені А. В. Свідзинського, Волинський національний університет імені Лесі Українки, просп. Волі, 13, м. Луцьк, Волинська обл., Україна, 43025
ORCID ID: *https://orcid.org/0009-0009-8860-0329*

Павло ШИГОРІН
доцент кафедри теоретичної та комп'ютерної фізики імені А. В. Свідзинського, Волинський національний університет імені Лесі Українки, просп. Волі, 13, м. Луцьк, Волинська обл., Україна, 43025
ORCID ID: *https://orcid.org/0000-0003-2396-8041*

Бібліографічний опис статті: Шваліковський, Д., Шигорін, П. (2024). Числовий розрахунок задачі плеяд у середовищі CAS Maxima. *Фізика та освітні технології*, 1, 80–86, doi: https://doi.org/10.32782/pet-2024-1-10

ЧИСЛОВИЙ РОЗРАХУНОК ЗАДАЧІ ПЛЕЯД У СЕРЕДОВИЩІ CAS MAXIMA

Модельні задачі небесної механіки описуються за допомогою диференціальних рівнянь або їх систем. Добре відомо, що задача багатьох тіл в аналітичному вигляді є нерозв'язною, тому для симуляції руху зірок або їх скупчень необхідно послуговуватись числовими методами. Для гравітаційно зв'язаних об'єктів виникають завдання, що складно моделюються навіть за допомогою комп'ютера, оскільки часто розглядувана модель є жорсткою, тобто є значна залежність еволюції розв'язку від початкових умов. В роботі запропонована схема розрахунку задачі семи тіл зоряного кластеру Плеяд. Розрахунки провадяться у відкритій системі комп'ютерної алгебри Maxima, котра надає всі необхідні інструменти як для безпосередніх обчислень, так і для візуалізації отриманих результатів. Використано метод розв'язання диференціальних рівнянь Рунге-Кутта-Фельберга 5-го порядку, побудовані графіки зоряних траєкторій та залежності їх видимих координат на небесній сфері від часу.
Ключові слова: *CAS Maxima, чисельний розрахунок, метод Рунге-Кутта-Фельберга, зоряні скупчення, Плеяди.*

Dmytro SHVALIKOVSKYI
Engineer of the Computing Physics Laboratory at the Department of Theoretical and Computing Physics of the Educational and Scientific Institute of Physics and Technology, Lesya Ukrainka Volyn National University, 13 Volya Ave., Lutsk, Volyn region, Ukraine, 43025
ORCID ID: *https://orcid.org/0009-0009-8860-0329*

Pavlo SHYGORIN
Associate Professor at the Department of Theoretical and Computing Physics of the Educational and Scientific Institute of Physics and Technology, Lesya Ukrainka Volyn National University, 13 Volya Ave., Lutsk, Volyn region, Ukraine, 43025
ORCID ID: *https://orcid.org/0000-0003-2396-8041*

To cite this article: Shvalikovskyi, D., Shygorin, P. (2024). Chyslovyi rozrakhunok zadachi pleiad v seredovyshchi CAS Maxima [Numerical calculation of the Pleiades problem in CAS Maxima Environment]. *Physics and Educational Technology*, 1, 80–86, doi: https://doi.org/10.32782/pet-2024-1-10

NUMERICAL CALCULATION OF THE PLEIADES PROBLEM IN CAS MAXIMA ENVIRONMENT

Model problems of celestial mechanics are described by means of differential equations or their systems. It is well known that the many-body problem is intractable in an analytical form, so numerical methods must be used to simulate the motion of stars or their clusters. For gravitationally bound objects, problems arise that are difficult to model even with a computer, since the model under consideration is often rigid, i.e., there is a significant dependence of the solution evolution on the initial conditions. This paper proposes a scheme for solving the seven-body problem of the Pleiades star cluster. The calculations are carried out in the open system of computer algebra Maxima, which provides all the necessary *tools for both direct calculations and visualization of the results. The method of solving the Runge-Kutta-Felberg differential equations of the 5th order is used, and graphs of stellar trajectories and the dependence of their apparent coordinates on the celestial sphere on time are plotted.*

Key words: *CAS Maxima, numerical calculation, Runge-Kutta-Fehlberg method, star clusters, Pleiades.*

Вступні зауваги. Зоряний кластер Плеяд (українська назва Стожари) має діаметр в 12 світлових років. Від нас Плеяди віддалені на відстань в 410 світлових років, вони вважаються порівняно молодим скупченням зірок, вік яких знаходиться в діапазоні 75-100 мільйонів років. Плеяди давно відомі як фізично пов’язана група об’єктів, а не результат випадкової проєкції різновіддалених зір. Велику цікавість цей кластер викликає в першу чергу тим, що він є другим після групи Ґіад найближчим до Землі зірковим скупченням. Плеяди та Ґіади є ідеальним ґрунтом для моделювання еволюції зір, оскільки всі їхні зірки мають однаковий вік і склад, але демонструють широкий діапазон мас [1, 2]. Ці два скупчення єдині, відстань до яких можна виміряти на поверхні Землі безпосередньо за їхнім паралактичним зміщенням. Виміряні значення лежать в основі діаграми Герцшпрунга – Рассела, за якою визначають відстані до всіх інших зоряних скупчень. Науковці екстраполюють отриману шкалу від розсіяних зіркових скупчень до галактик і галактичних скупчень, побудувавши шкалу космічних відстаней [4]. Зрештою, знання астрономів про вік і розвиток Всесвіту у великій мірі залежать від знання відстані до зіркового скупчення Плеяд, тому вони є своєрідним еталоном, котрий добре вивчений та змодельований (рис. 1) [7].

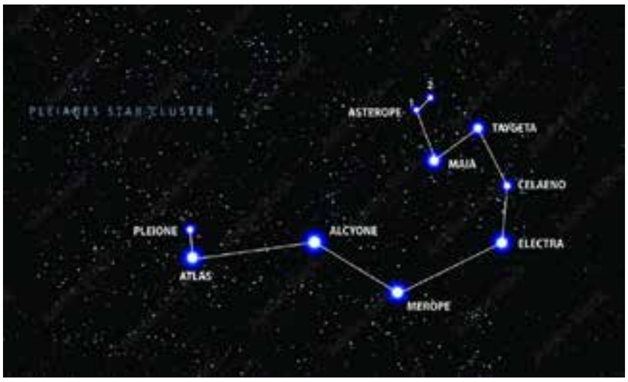

Рис. 1. Сузір'я Плеяд

Для моделювання задачі використаємо програмний пакет CAS Maxima, який є найбільш розробленим вільним аналогом відомих математичних застосунків Maple чи Mathematica, та у певних класах розрахунків ні в чому їм не поступається [6]. Цей пакет може успішно використовуватись для розв’язання цілої низки проблем математичного аналізу, лінійної алгебри, чисельного та візуального моделювання. Переваги його наступні: можливість вільного використання (Maxima відноситься до класу вільних програм та поширюється на основі ліцензії GNU); можливість функціонування під управлінням різних ОС (зокрема Linux та Windows); невеликий розмір програми; має зручний та інтуїтивно зрозумілий інтерфейс.

Постановка проблеми та аналіз досліджень. Задача Плеяд – це модельна задача небесної механіки семи зірок, що рухаються в одній площині. Математично вона описується системою з 14 диференціальних рівнянь другого порядку разом з 28 початковими умовами, що визначають початкове положення та швидкість для кожного небесного тіла. Задача не жорстка, але досить складна, її можна легко розширити на будь-яку кількість об’єктів, не обов’язково в одній площині. Звісно, реальний рух зірок відбувається у тривимірному просторі, але нас в першу чергу цікавитиме проєкція цього руху на площину небесної сфери.

Нехай маємо рух кількох масивних тіл на площині. Запишемо другий закон Ньютона для i-го тіла та вираз для сили притягання між двома тілами i та j, що взаємодіють гравітаційно:

$$\vec{F}_i = \frac{d\vec{p}_i}{dt} = m_i \frac{d^2\vec{r}_i}{dt^2}; \quad \vec{F}_{ij} = G\frac{m_i m_j}{\left|\vec{r}_j - \vec{r}_i\right|^3}\left(\vec{r}_j - \vec{r}_i\right).$$

У системі багатьох тіл сила, що діє на окремий об’єкт, є векторною сумою всіх сил, що діють на нього від усіх інших об’єктів. Тому можемо записати $\vec{F}_i = \sum_{i \neq j} \vec{F}_{ij}$, що після покладання $G = 1$ та розділення змінних нам дає

$$\begin{cases} \ddot{x}_i = \sum\limits_{i \neq j} \dfrac{m_j\left(x_j - x_i\right)}{\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)^{\frac{3}{2}}}; \\ \ddot{y}_i = \sum\limits_{i \neq j} \dfrac{m_j\left(y_j - y_i\right)}{\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)^{\frac{3}{2}}}. \end{cases}$$

Початкові дані для цього руху беруться із роботи [3]. Числа означають положення кожного тіла відносно певної умовної нерухомої точки всередині скупчення (у світлових роках) та їх швидкості в момент часу $t=0$. Маси розглядуваних зірок є фактично однаковими, вони міняються від 3 до 6 мас Сонця. Ми покладемо масу кожної зірки рівною її номеру $m_j = j$, на вигляд розв'язку це вплине мінімально.

$$\begin{cases} x_i\big|_{t=0} = [3,3,-1,-3,2,-2,2]^T; \\ y_i\big|_{t=0} = [3,-3,2,0,0,-4,4]^T; \\ \dot{x}_i\big|_{t=0} = [0,0,0,0,0,1.75,-1.5]^T; \\ \dot{y}_i\big|_{t=0} = [0,0,0,-1.25,1,0,0]^T. \end{cases}$$

Щоб перейти від 14 рівнянь другого порядку до системи з 28 рівнянь першого порядку, запишемо їх у наступній формі [5]:

$$\frac{d}{dt}\begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} w \\ f(z) \end{pmatrix},$$

де z позначає набір всіх координат x_i та y_i, $f(z)$ позначає набір правих частин рівнянь для $\ddot{x}_i$ та $\ddot{y}_i$, w проміжна змінна, яка є відповідною проєкцією лінійної швидкості кожного об'єкту.

Виклад основного матеріалу й обґрунтування отриманих результатів. Переведемо ці вхідні дані на мову Maxima та побудуємо розв'язки. Задаємо вирази для гравітаційних рівнянь руху (команда concat з'єднує будь-які списки чи символи в один):

```
(%i1) fx(i):=sum(if i#j then j·(concat(x,j)-concat(x,i)) /
                        ((concat(x,i)-concat(x,j))^2+
                         (concat(y,i)-concat(y,j))^2)^1.5
                 else 0 ,j,1,7);
```

$$\text{(\%o1) } fx(i) := \sum_{j=1}^{7} \text{if } i\#j \text{ then } \frac{j\,(concat(x,j)-concat(x,i))}{\left((concat(x,i)-concat(x,j))^2+(concat(y,i)-concat(y,j))^2\right)^{1.5}} \text{ else } 0$$

```
(%i2) fy(i):=sum(if i#j then j·(concat(y,j)-concat(y,i)) /
                        ((concat(x,i)-concat(x,j))^2+
                         (concat(y,i)-concat(y,j))^2)^1.5
                 else 0 ,j,1,7);
```

$$\text{(\%o2) } fy(i) := \sum_{j=1}^{7} \text{if } i\#j \text{ then } \frac{j\,(concat(y,j)-concat(y,i))}{\left((concat(x,i)-concat(x,j))^2+(concat(y,i)-concat(y,j))^2\right)^{1.5}} \text{ else } 0$$

Формуємо список з 28 правих частин рівнянь руху, запровадивши додаткові змінні w_x, w_y (команда flatten робить із списків різних рівнів вкладеності один великий список першого рівня, де перелічені всі елементи):

```
(%i3) equs:flatten([makelist(concat(wx,i),i,1,7),
                    makelist(concat(wy,i),i,1,7),
                    makelist(fx(i),i,1,7),
                    makelist(fy(i),i,1,7)])$
```

Формуємо список із 28 невідомих функцій, відносно яких потрібно розв'язати рівняння руху:

```
(%i4) funcs:flatten([makelist(concat(x,i),i,1,7),
                     makelist(concat(y,i),i,1,7),
                     makelist(concat(wx,i),i,1,7),
                     makelist(concat(wy,i),i,1,7)]);
(%o4) [x1,x2,x3,x4,x5,x6,x7,y1,y2,y3,y4,y5,y6,y7,wx1,wx2,wx3,wx4,wx5,wx6,wx7,wy1,wy2
,wy3,wy4,wy5,wy6,wy7]
```

Можна помітити, що завершення команди значком «$» не виводить результату здійснення операції на екран, тоді як значок «;» задає виведення вихідної лінії для перегляду. Вибір кінцевого значка залежить лише від вподобань програміста, та побіжної перевірки того, чи всі команди вдало записались. Відображені результати обчислень можуть бути досить об'ємними, наприклад задана команда (%i3) продукує кілька сторінок тексту.

Тепер задаємо початкові умови у момент часу $t=0$ та проміжок інтегрування, оберемо його рівним $t \in [0;3]$, де одиниця часу рівна 100 тис. років.

```
(%i5) init:[3,3,-1,-3,2,-2,2,3,-3,2,0,0,-4,4,0,0,0,0,0,1.75,-1.5,0,0,0,-1.25,1,0,0];
(%o5) [3,3,-1,-3,2,-2,2,3,-3,2,0,0,-4,4,0,0,0,0,0,1.75,-1.5,0,0,0,-1.25,1,0,0]

(%i7) t_start:0; t_end:3;
(%o6) 0
(%o7) 3
```

Пакет Maxima має кілька вбудованих команд для чисельного інтегрування диференціальних рівнянь. Крім того, цілком можливо написати свою власну команду, яка буде інтегрувати ЗДР з будь-якою наперед заданою точністю (вона буде обмежуватись лише технічними характеристиками комп'ютера та вільним часом дослідника, який хоче дочекатись результатів). Існує стандартна функція rk:

rk(deqn,y,y0,[x,x0,x1,h]),

що реалізує метод Рунге-Кутта 4-го порядку. Тут deqn – диференціальне рівняння, y – залежна змінна, y0 – початкове значення у точці x0, [x,x0,x1,h] – незалежна змінна, початкова, кінцева точки та крок інтегрування. Цей метод фактично повсюдно використовується і вважається досить надійним для задач, що не містять порогових ефектів (threshold effects) або жорстких умов (stiff problems).

Існує, однак, більш вдала реалізація цього методу, що носить назву метод Рунге-Кутта-Фельберга 4-5 порядку. Особливість його полягає в тому, що розбиття проміжку інтегрування на малі частини залежить не від заздалегідь заданого кроку по осі x, а від величини зміщення обчисленої інтегральної кривої по осі y. Якщо ця величина при деяких x_i буде перевищувати певне вказане значення (за замовчуванням 10^{-6}), система повернеться назад, збільшить кількість кроків розбиття та проведе перерахунок, тим самим підвищивши порядок точності інтегрування від 4 до 5. Тому цей метод використовують при вказаних вище проблемних ситуаціях.

Метод Рунге-Кутта-Фельберга 4-5 порядку реалізує команда rkf45. Цей спосіб застосовний лише до ЗДР першого порядку, що і зумовило введення нами проміжної змінної w. Розглянемо докладно синтаксис та застосування команди.

Для побудови чисельного розв'язку диференціального рівняння воно має бути записане у вигляді $y'(x)=f(x,y)$, у випадку автономної системи – має бути у вигляді $\{\dot{x}=G(x,y); \dot{y}=H(x,y)\}$. Синтаксис наступний:

rkf45(f(x,y),y,y0,[x,x0,x1],opts) – для одиночного рівняння;

rkf45([G(x,y),H(x,y)],[x,y],[x0,y0],[t,t0,t1], opts) – для системи рівнянь.

Тут f, G, H – права частина ЗДР або їх список, y або [x,y] – залежна змінна або їх список, y0 або [x0,y0] – початкові значення залежної змінної, [x,x0,x1] або [t,t0,t1] – оголошення незалежної змінної, її початкове та кінцеве значення, opts – додаткові опції.

Завантажуємо пакет rkf45 та застосовуємо його:

```
(%i8) load(rkf45);
(%o8) C:/Maxima-gcl-5.37.3/share/maxima/5.37.3/share/contrib/rkf45/rkf45.mac
```

```
(%i9) sol1:rkf45(equs,funcs,init,[t,t_start,t_end],report=true)$
------------------------------------------------------
Info: rkf45:
   Integration points selected: 1143
    Total number of iterations: 1144
           Bad steps corrected: 2
       Minimum estimated error: 3.896190579368019 10^-8
       Maximum estimated error: 9.82861310778899 10^-7
Minimum integration step taken: 2.515209236707204 10^-5
Maximum integration step taken: 0.0650739568467315
------------------------------------------------------
```

Із відгуку системи дізнаємось, що було обрано 1143 точки інтегрування, з максимальною похибкою $\sim 10^{-6}$ на будь-якому кроці. Результати повернулись у вигляді списку із 1143 елементів, кожен з яких є списком із 29 елементів (час t, сім просторових координат x_i, сім просторових координат y_i, сім x-компонент швидкості w_{xi}, сім y-компонент швидкості w_{yi}). Для опрацювання даних виділимо окремо списки для часу та всіх просторових координат.

```
(%i10) t_list:makelist(sol1[k][1],k,1,length(sol1))$
```

```
(%i17) x1_list:makelist(sol1[k][2],k,1,length(sol1))$
       x2_list:makelist(sol1[k][3],k,1,length(sol1))$
       x3_list:makelist(sol1[k][4],k,1,length(sol1))$
       x4_list:makelist(sol1[k][5],k,1,length(sol1))$
       x5_list:makelist(sol1[k][6],k,1,length(sol1))$
       x6_list:makelist(sol1[k][7],k,1,length(sol1))$
       x7_list:makelist(sol1[k][8],k,1,length(sol1))$
```

```
(%i24) y1_list:makelist(sol1[k][9],k,1,length(sol1))$
       y2_list:makelist(sol1[k][10],k,1,length(sol1))$
       y3_list:makelist(sol1[k][11],k,1,length(sol1))$
       y4_list:makelist(sol1[k][12],k,1,length(sol1))$
       y5_list:makelist(sol1[k][13],k,1,length(sol1))$
       y6_list:makelist(sol1[k][14],k,1,length(sol1))$
       y7_list:makelist(sol1[k][15],k,1,length(sol1))$
```

Позначимо окремо початкові точки, з яких стартував рух кожної зірки:

```
(%i25) init_pnts:[discrete,makelist([init[i],init[i+7]],i,1,7)];
(%o25) [discrete,[[3,3],[3,-3],[-1,2],[-3,0],[2,0],[-2,-4],[2,4]]]
```

Тепер можемо зобразити траєкторії руху кожної зорі (рис. 2 a). Як бачимо, на графіку присутня область, на якій траєкторії дуже переплітаються. Клацнувши правою кнопкою миші на відповідній частині малюнка, можна виділити цю область, і тоді вона автоматично збільшиться (рис. 2 b).

```
(%i26) plot2d([init_pnts,[discrete,x1_list,y1_list],[discrete,x2_list,y2_list],
          [discrete,x3_list,y3_list],[discrete,x4_list,y4_list],
          [discrete,x5_list,y5_list],[discrete,x6_list,y6_list],
          [discrete,x7_list,y7_list]],  [x,-4,4],[y,-6,6],
          [style,[points,2,5,7],[lines,2,1],[lines,2,2],
          [lines,2,3],[lines,2,4],[lines,2,5],[lines,2,6],[lines,2,2]],
          [legend,"Стартові точки","Зоря 1","Зоря 2","Зоря 3","Зоря 4","Зоря 5","Зоря 6","Зоря 7"]
           );
(%o26) [C:/Users/orreginal-pc/maxout.gnuplot]
```

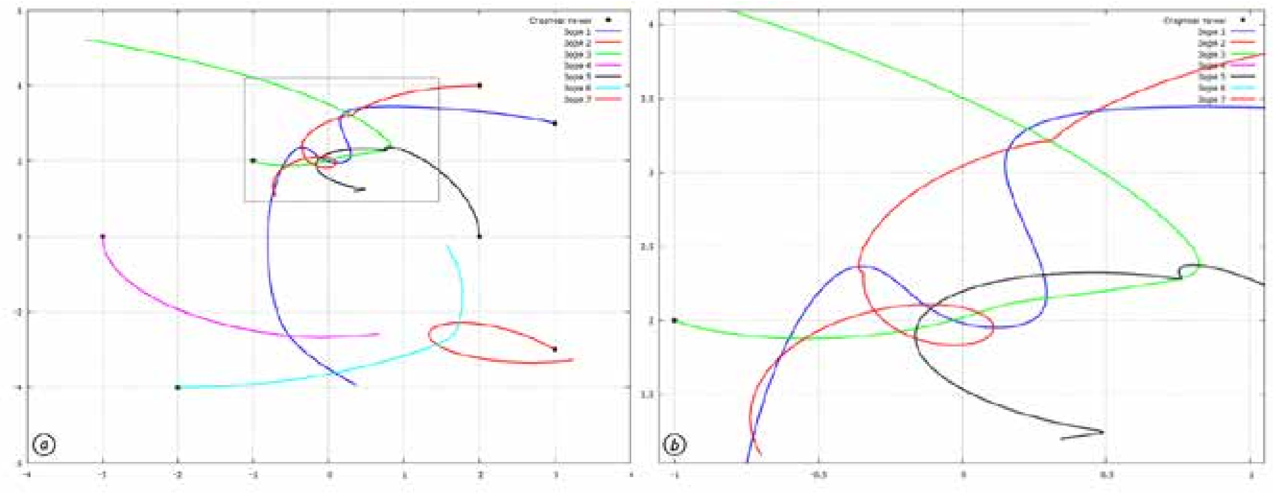

Рис. 2. Траєкторії окремих зір зоряного скупчення Плеяд (a), збільшена частина області з переплетеними траєкторіями (b)

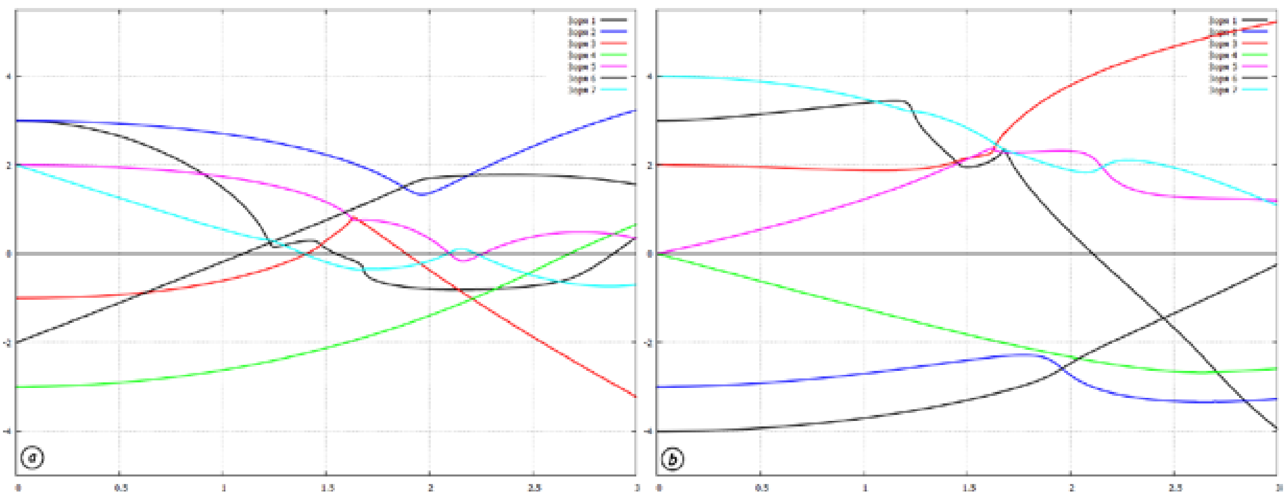

Рис. 3. Зміна з часом координат $x(t)$ (a) та $y(t)$ (b)

Задавши подібну команду plot2d, де у парах замість x чи y стоять значення часових точок, можна зобразити часову еволюцію відповідних просторових компонент $x(t)$, $y(t)$ (рис. 3).

Використавши додаткову опцію absolute_tolerance, можна змінювати величину похибки та підвищувати точність інтегрування, однак платою за це буде значно більше навантаження на процесор та збільшення часу обчислень. Наприклад, точність 10^{-10} замість 10^{-6} призведе до того, що час обробки зміниться від 5 с до 10 год. Втім, задачу чисельного інтегрування руху кількох тіл для реальних потреб космічних розрахунків обчислюють з точністю 10^{-16}, що для звичайних комп'ютерів вимагало б років безперервної роботи.

Висновки. За допомогою пакету математичних розрахунків CAS Maxima було чисельно проінтегровано задачу семи тіл у небесній механіці, побудовані точки розв'язків та відповідні траєкторії руху зір у зоряному скупченні Плеяд. Отримані дані моделювання узгоджуються з експериментальними фактами та попередньо зробленими розрахунками. Продемонстровано, що розроблений підхід є гнучким та легко масштабується, у ньому можна керувати точністю обчислень та застосовувати до іншого числа небесних тіл.

ЛІТЕРАТУРА:

1. Dennis Stello, Poul Erik Nissen. The problem of the Pleiades distance. Constraints from Stromgren photometry of nearby field stars. arXiv:astro-ph/0105222v1.
2. Guillermo Abramson. Around the Pleiades. Centro Atómico Bariloche, CONICET and Instituto Balseiro. URL: https://www.arxiv-vanity.com/papers/1808.02968/.
3. Francesca Mazzia and Felice Iavernaro. The Set for Initial Value Problem Solvers. Department of Mathematics University of Bari. Report 40, 2003.
4. J. Kelly Beatty. Resolving the Pleiades Distance Problem. Sky & Telescope. August 28, 2014.
5. MathWorks Help Center. Solve Celestial Mechanics Problem with High-Order Solvers. URL: https://de.mathworks.com/help/matlab/math/solve-celestial-mechanics-problem-with-high-order-ode-solvers.html.
6. Шваліковський Д. М. CAS Maxima: основи роботи. Луцьк: Вежа-Друк, 2022. 106 с.
7. Шигорін П. П. Вибрані питання астрономії та астрофізики: Навчальний посібник. Луцьк, 2020. 136 с.

REFERENCES:

1. Dennis Stello, Poul Erik Nissen. The problem of the Pleiades distance. Constraints from Stromgren photometry of nearby field stars. arXiv:astro-ph/0105222v1.
2. Guillermo Abramson. Around the Pleiades. Centro Atómico Bariloche, CONICET and Instituto Balseiro. Retrieved from https://www.arxiv-vanity.com/papers/1808.02968/.
3. Francesca Mazzia and Felice Iavernaro. (2003). The Set for Initial Value Problem Solvers. Department of Mathematics University of Bari. Report 40.
4. J. Kelly Beatty. (2014).Resolving the Pleiades Distance Problem. Sky & Telescope. August 28.
5. MathWorks Help Center. Solve Celestial Mechanics Problem with High-Order Solvers. Retrieved from https://de.mathworks.com/help/matlab/math/solve-celestial-mechanics-problem-with-high-order-ode-solvers.html.
6. Shvalikovskyi, D. M. (2022). CAS Maxima: osnovy roboty [CAS Maxima: the Basics]. Lutsk: Vezha-Print, 106 p [in Ukrainian].
7. Shygorin, P. P. (2020). Vybrani pytannia astronomii ta astrofizyky: Navchalnyi posibnyk [Selected Issues of Astronomy and Astrophysics: a Study Guide]. Lutsk, 136 p [in Ukrainian].