



УНИВЕРСИТЕТ
ЛОБАЧЕВСКОГО



Исследование явления локализации в решетках нелинейных осцилляторов с пространственной неоднородностью

Магистерская диссертация
студентки группы 381406м2:

Игнашевой Е.В.

Научный руководитель:

Лаптева Т.В.

Рецензент:

Конаков О.И.

Введение

Круг задач:

- Численное интегрирование решётчатых систем на больших временных и пространственных масштабах

Актуальность:

- Программные средства, разработанные и используемые ранее, требуют объемных вычислительных ресурсов, поэтому для проведения исследований необходимы компьютерные кластеры, которые не всегда доступны

Постановка задачи

Основные решённые задачи:

- Изучение симплектического метода интегрирования семейства *SABA* второго порядка с корректором
- Реализация интегратора и подтверждение корректности вычислений
- Проведение оптимизации, получение максимального ускорения вычислений
- Применение на практике, демонстрация эффективности полученного инструмента интегрирования

FSW модель

$$H_{FSW} = \sum_l \left[\frac{p_l^2}{2} + \frac{e_l u_l^2}{2} + \alpha \frac{(u_m - u_l)^4}{\gamma} \right]$$

Модель FSW описывает динамику цепи связанных осцилляторов, обладает свойством гамильтоновости.

Применима к схеме интегрирования SABA

SABA₂C для FSW

Общие формулы интегрирования для *SABA₂C*:

$$SABA_2 = e^{c_1 \tau L_A} e^{d_1 \tau L_B} e^{c_2 \tau L_A} e^{d_1 \tau L_B} e^{c_1 \tau L_A}$$

$$SABA_2C = e^{-(\tau^3 g/2)L_c} \cdot SABA_2 \cdot e^{-(\tau^3 g/2)L_c}$$

Схема *SABA₂C* для FSW модели:

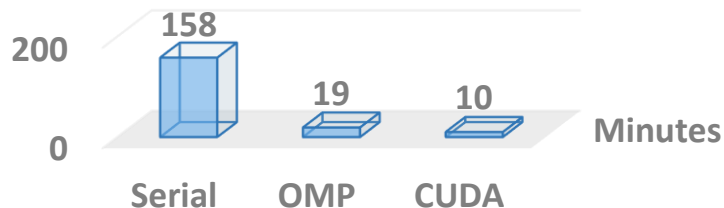
$$e^{\tau L_A} = \begin{cases} u'_l = u_l + \tau p_l \\ p'_l = p_l, \end{cases} \quad e^{\tau L_B} = \begin{cases} u'_l = u_l \\ p'_l = p_l - \tau Q_l \end{cases}$$

$$e^{\tau L_c} = \begin{cases} u'_l = u_l \\ p'_l = p_l - 2\tau \{ [\tilde{e}_l + 3(u_{l-1} - u_l)^2 + 3(u_{l+1} - u_l)^2] Q_l - \\ - 3(u_{l-1} - u_l)^2 Q_{l-1} - 3(u_{l+1} - u_l)^2 Q_{l+1} \} \end{cases}$$

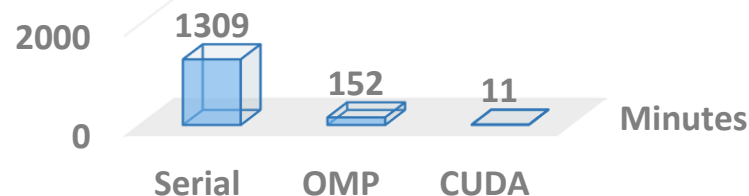
$$Q_x = \tilde{e}_x u_x - (u_{x-1} - u_x)^3 - (u_{x+1} - u_x)^3$$

Численный интегратор

16 Осцилляторов,
tend = 10e6



128 Осцилляторов,
tend = 10e6



$$\frac{Time_{Serial}}{Time_{OMP}} \cong 8, \quad \frac{Time_{Serial}}{Time_{CUDA}} \cong 16$$

$$\frac{Time_{Serial}}{Time_{OMP}} \cong 8, \quad \frac{Time_{Serial}}{Time_{CUDA}} \cong 128$$

Тип	Модель	Количество ядер	Максимальная полоса пропускания памяти	Частота
CPU	Intel Core i7 6770k	8	34,1 GB/s	4.2 GHz
GPU	NVidia GTX 960	1024	112 GB/s	1.2 MHz

Численный интегратор

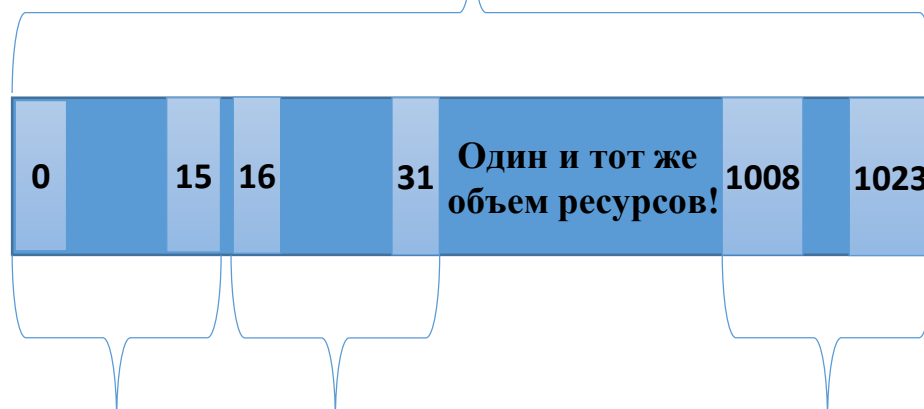
Распределение вычислительной нагрузки:

Необходимо распределить нагрузку таким образом, чтобы всегда производились только полезные вычисления

1 эксперимент для цепи из 1024 осц.

1024

ПЛОХО



64 эксперимента для цепей из 16 осц.

16

16

16

ХОРОШО

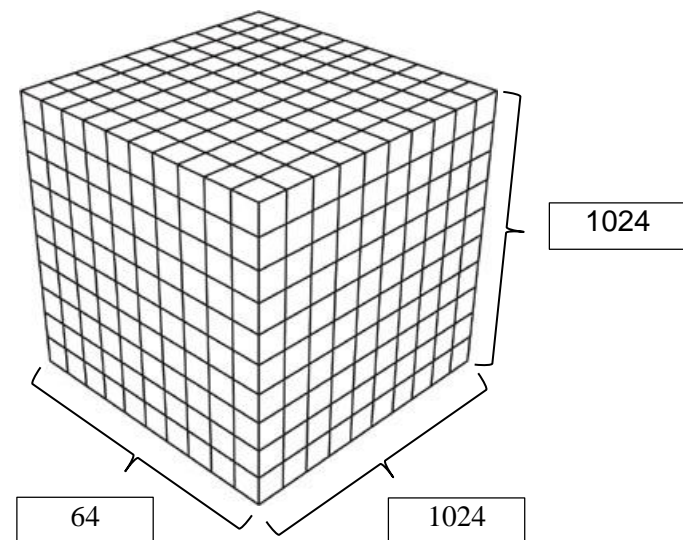
Численный интегратор

$$Time_{GPU} = \frac{Time_{CPU}}{N_{OSC} \times N_{RUN}}$$

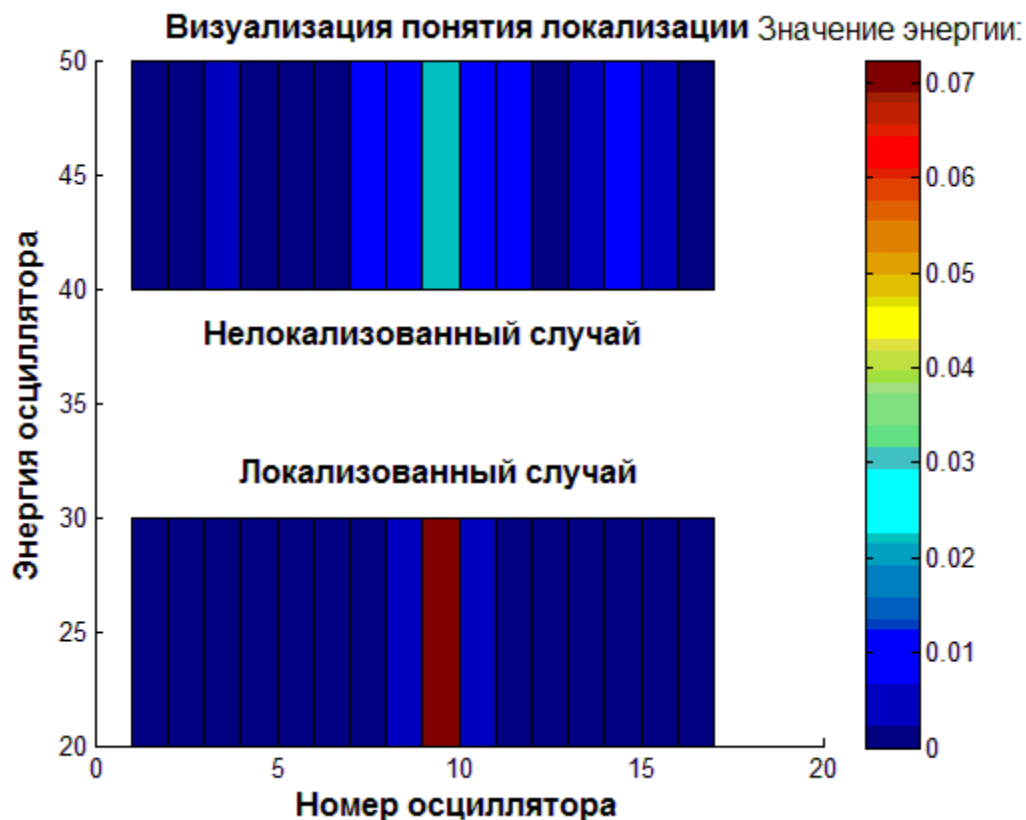
Полученное ускорение:

Время работы интегратора зависит от величины сетки потоков и производительности графического процессора. N_{OSC} и N_{RUN} - должны быть не больше максимальных значений сетки, поддерживаемой на конкретном GPU.

NVidia GTX 960



Понятие локализации



Локализация

Для локализованного случая характерно сохранение динамики только на возбужденном в начальный момент осцилляторе, либо в малой его окрестности, а для нелокализованного случая — распространение энергии по всей цепи.

Применение на практике

Утверждение:

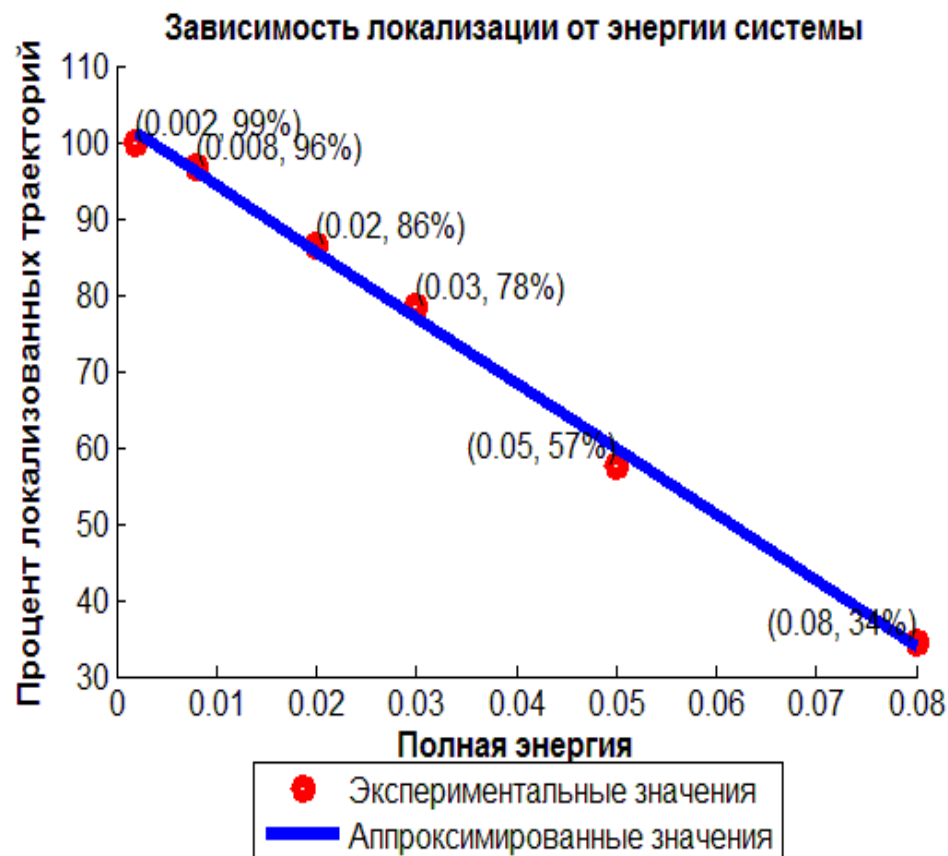
Если E_{ini} – полная энергия системы, P_L – вероятность локализации, то:

при $0 < E_{ini} < \frac{1}{3}$: $P_L = 1 - 3E_{ini}$,

при $E_{ini} > \frac{1}{3}$: $P_L \approx 0$

Результат:

На основе объемной экспериментальной выборки была получена статистическая вероятность локализации при заданных условиях. Линейный характер локализации виден на рисунке



Заключение

- Исследовано семейство симплектических интеграторов SABA
- Разработан производительный инструмент численного интегрирования
- Эффективность применения продемонстрирована в решении практической задачи
- Достигнут результат, способный в дальнейшем оказать значительную помощь в проведении численных экспериментов



Спасибо! Вопросы?

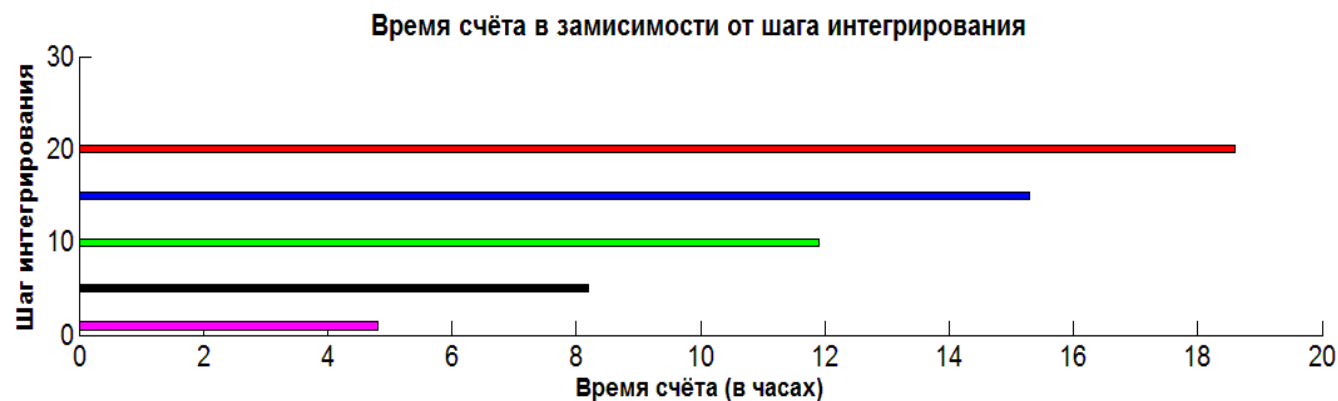
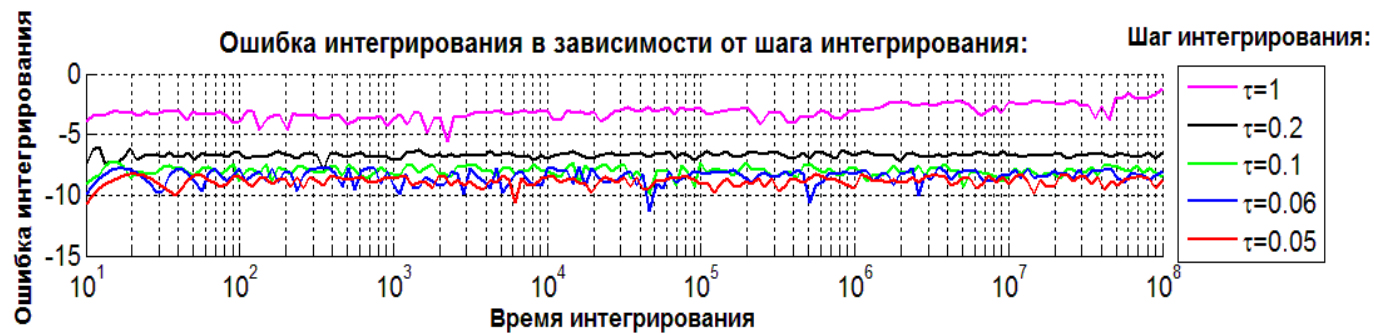
Магистерская диссертация
студентки группы 381406м2:

Игнашевой Е.В.

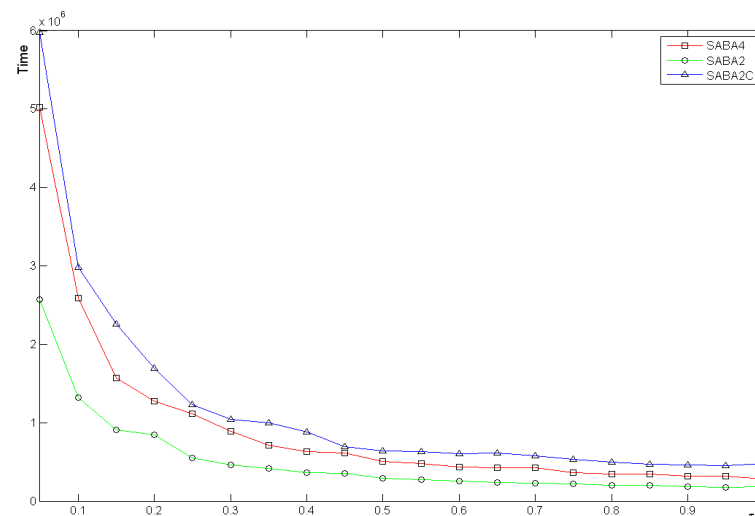
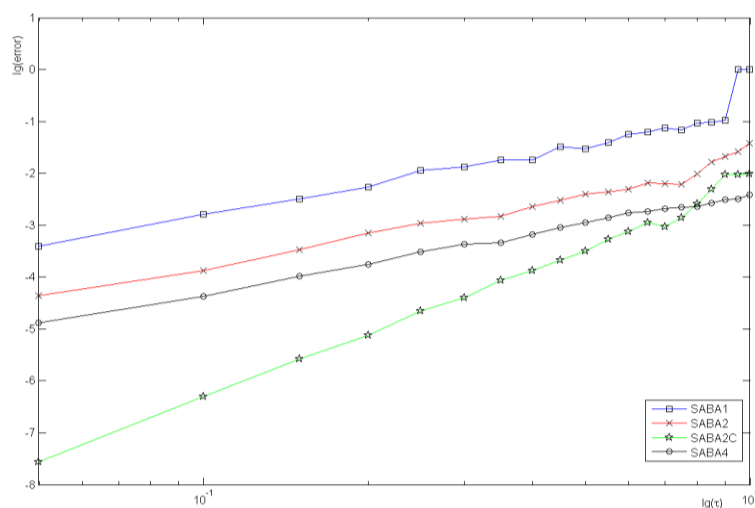
Научный руководитель:
Рецензент:

Лаптева Т.В.

Приложение: Точность интегрирования



Приложение: Выбор SABA



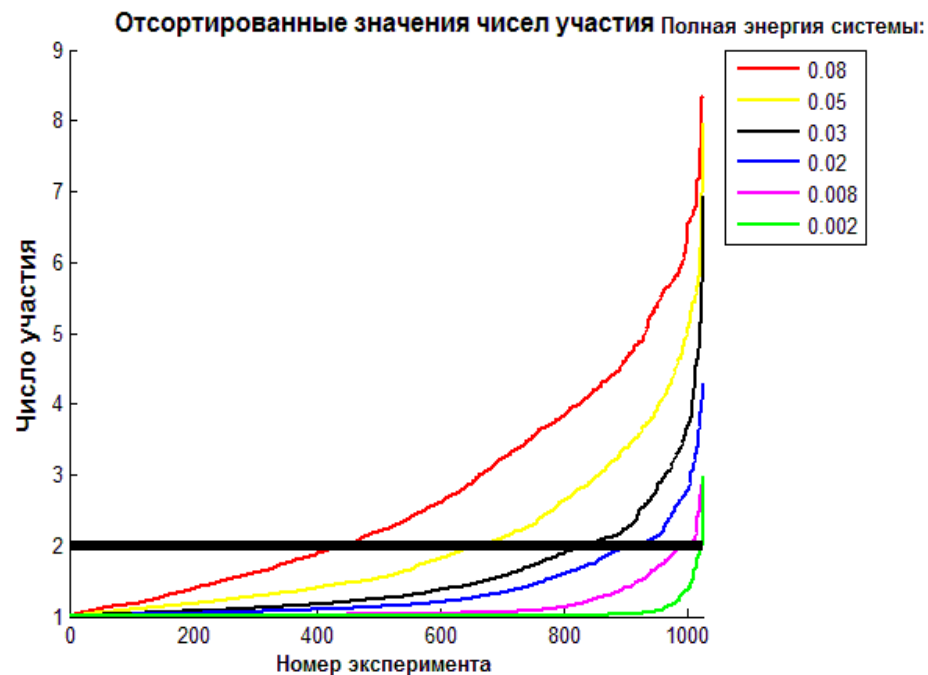
- При малых шагах интегрирования наименьшую ошибку даёт схема с корректором SABA₂C
- Время счёта для различных конфигураций схемы SABA существенно не отличается

Приложение: Критерии локализации

**Критерий локализации по
числу участия: $P \leq 2$,**

$$\text{где } P = \frac{(\sum_{i=0}^N E_i)^2}{\sum_{i=0}^N E_i^2},$$

$E_{\text{exp}}^2 = (\sum_{i=0}^N E_i)^2$ – полное
мгновенное значение энергии
(экспериментальное).



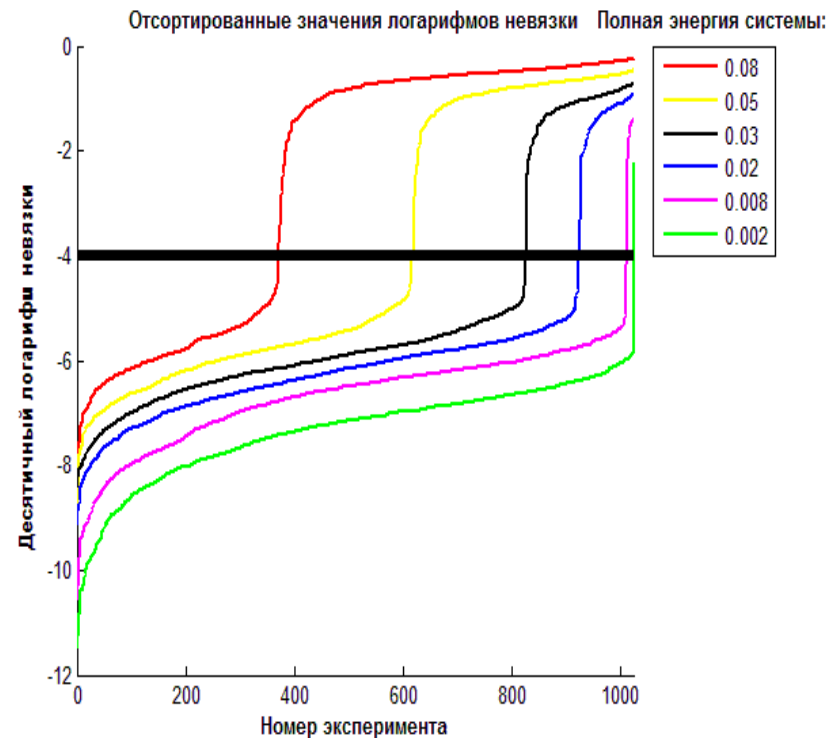
Приложение: Критерии локализации

**Критерий локализации по
невязке: $\log_{10} \varepsilon \leq -4$,**

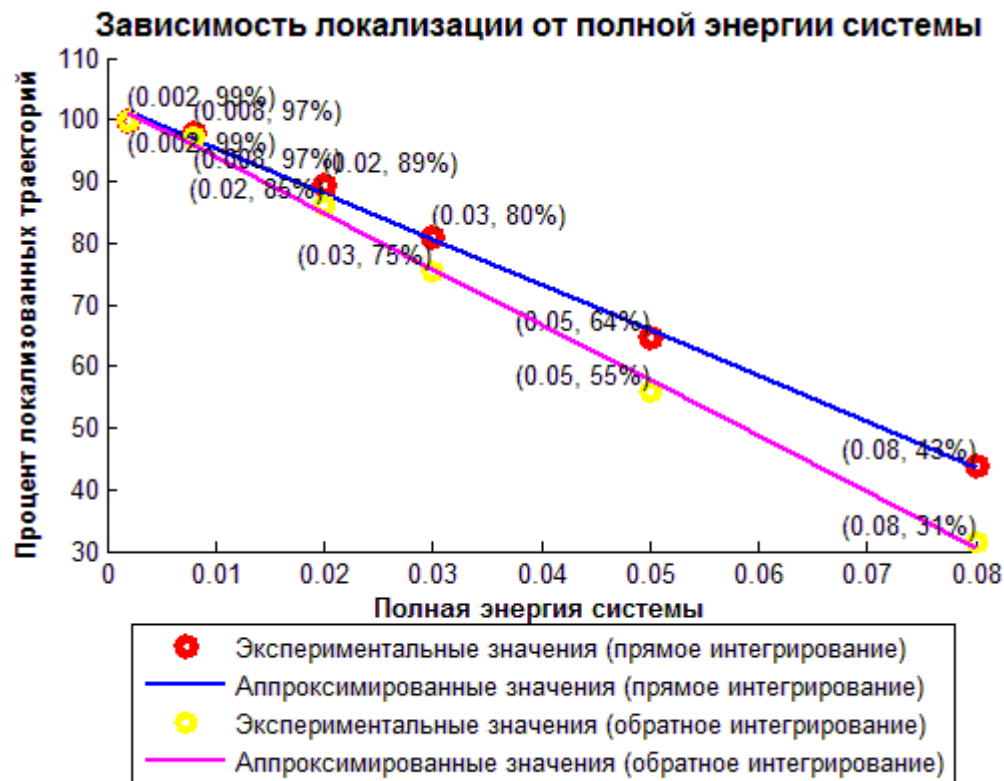
где $\varepsilon = |X_0 - \bar{X}_0| =$

$$= \sqrt{\sum_{i=0}^N [(u_i - \bar{u}_i)^2 + (p_i - \bar{p}_i)^2]} -$$

если евклидова норма между
состояниями X_0 и \bar{X}_0 .



Приложение: Критерии локализации



Приложение: Код оператора L_A

```
__device__ void LA(float tau, int ID, float *X0, float *TempPoint, float *EPS)
{
    TempPoint[ID*2] = X0[ID*2] + tau * X0[ID*2+1];
    TempPoint[ID*2+1] = X0[ID*2+1];
}
```

Приложение: Код оператора L_B

```
__device__ void LB(float tau, int ID, float *X0, float *TempPoint, float *EPS)
{
    float uLM1_ul_3=0, uLP1_ul_3=0, Ql;
    if (ID!=0)
        uLM1_ul_3 = (X0[(ID-1)*2]-X0[ID*2])*(X0[(ID-1)*2]-
            - X0[ID*2])*(X0[(ID-1)*2]-X0[ID*2]);
    else
        uLM1_ul_3 = 0;
    if (ID!=N-1)
        uLP1_ul_3 = (X0[(ID+1)*2]-X0[ID*2])*(X0[(ID+1)*2]-
            - X0[ID*2])*(X0[(ID+1)*2]-X0[ID*2]);
    else
        uLP1_ul_3 = 0;
    Ql=EPS[ID]*X0[ID*2] - uLM1_ul_3 - uLP1_ul_3;
    TempPoint[ID*2] = X0[ID*2];
    TempPoint[ID*2+1] = X0[ID*2+1] - tau * Ql;
}
```

Приложение: Код оператора L_c

```
__device__ void LC(float tau, int ID, float *X0, float *TempPoint, float *EPS)
{
    float ul_minus_1, el_minus_1, ul_minus_2, ulM1_ul_2, ulM1_ul_3,
    ulM2_ulM1_3, ul_plus_1, el_plus_1, ul_plus_2, ulP1_ul_2,
    ulP1_ul_3, ulP2_ulP1_3, ul, pl, el, Ql, Ql_minus_1, Ql_plus_1;
    ul = X0[ID*2];
    pl = X0[ID*2+1];
    el = EPS[ID];
    fill_bounds(ul_minus_1, el_minus_1, ul_minus_2, ulM1_ul_2, ulM1_ul_3,
                ulM2_ulM1_3, ul, ul_plus_1, el_plus_1, ul_plus_2,
    ulP1_ul_2, ulP1_ul_3, ulP2_ulP1_3= 0);
    Ql = el*ul - ulM1_ul_3 - ulP1_ul_3;
    Ql_minus_1 = el_minus_1*ul_minus_1 - ulM2_ulM1_3 - (-1)*ulM1_ul_3;
    Ql_plus_1 = el_plus_1*ul_plus_1 - (-1)*ulP1_ul_3 - ulP2_ulP1_3;
    TempPoint[ID*2] = ul;
    TempPoint[ID*2+1] = pl - 2 * tau * ( (el + 3*ulM1_ul_2 + 3*ulP1_ul_2 ) *
        * Ql - 3 * Ql_minus_1 * ulM1_ul_2 - 3 * Ql_plus_1 * ulP1_ul_2 );
}
```