

ГЕНЕРАТОР И РЕШАТЕЛЬ УРАВНЕНИЙ ПРОБЛЕМНО ОРИЕНТИРОВАННОЙ ПАРАЛЛЕЛЬНОЙ МОДЕЛИРУЮЩЕЙ СРЕДЫ ДЛЯ СЕТЕВЫХ ОБЪЕКТОВ С СОСРЕДОТОЧЕННЫМИ ПАРАМЕТРАМИ

Перерва А.А.
Кафедра ЭВМ ДГТУ
svjatnyj@cs.dgtu.donetsk.ua

Abstract

Pererva A.A. Generator and decider of equations of domain oriented parallel simulation environment for net objects with lumped parameters. There is considered the methods of parallel model building in this article. Equations generator and their solving by means of numerical methods are described.

1. Введение

Проблемно ориентированная параллельная моделирующая среда (ПОПМС) [1] должна иметь средства топологического анализа сложных объектов моделирования и формирования параллельных моделей. В работе [2] представлен топологический анализатор, который формирует топологическую информацию о сетевом объекте. На её основе можно решить основную задачу формирования параллельной модели – автоматическое генерирование уравнений объекта в виде, удобном для моделирования и решение уравнений численными методами.

Рассмотрим вопросы построения генератора и решателя уравнений ПОПМС.

2. Алгоритм генерирования уравнений (ГУ) сетевого объекта

Сетевой объект с сосредоточенными параметрами имеет ветви, динамические процессы в которых описываются дифференциальными уравнениями вида :

$$K_i \frac{dQ_i}{dt} + R_i Z_i = H_i \quad (1)$$

где Q_i – поток в i -ой ветви (электрический ток, расход жидкости, воздуха, газа); $i=1,2,\dots,m$ – номера ветвей; K_i, R_i – физические параметры ветви, H_i – разность потенциалов (давлений) в начальном и конечном узлах ветви, $Z_i=Q_i \cdot |Q_i|$ – вспомогательная переменная.

Введем вектор потоков $Q=(Q_1, Q_2, \dots, Q_m)^T$, диагональные матрицы параметров $R=(R_1, R_2, \dots, R_m)^{DM}$, $K=(K_1, K_2, \dots, K_m)^{DM}$ и векторы $H=(H_1, H_2, \dots, H_m)^T$, $Z=(Z_1, Z_2, \dots, Z_m)^T$. Пользуясь матрицами инцидентий A и контуров S , которые формируются топологическим анализатором [2], запишем систему уравнений сетевого объекта в векторно-матричной форме [3] :

$$AQ = 0 \quad (2)$$

$$SK \frac{dQ}{dt} + SRZ = SH \quad (3)$$

Генератор уравнений сетевого объекта должен представлять собой программу, которая преобразует систему (2), (3) к виду, удобному для численного решения. Алгоритмы этих преобразований сводятся к следующему.

Топологический анализатор [2] формирует : дерево и антидерво графа сетевого объекта с подвекторами потоков $X=(X_1, X_2, \dots, X_{n-1})^T$, $Y=(Y_1, Y_2, \dots, Y_r)^T$, которые принадлежат вектору Q

$$Q = \begin{pmatrix} X \\ Y \end{pmatrix}; \tag{4}$$

матрицы инцидентий и контуров, упорядоченные относительно подвекторов X, Y

$$A=(A_X \ A_Y), \tag{5}$$

$$S=(S_X \ S_Y); \tag{6}$$

таблицу исходных данных, в которой матрицы R, K, вектор H упорядочиваются по X, Y:

$$R = \begin{pmatrix} R_X & 0 \\ 0 & R_Y \end{pmatrix}, \quad K = \begin{pmatrix} K_X & 0 \\ 0 & K_Y \end{pmatrix}, \quad H = (H_X, H_Y)^T.$$

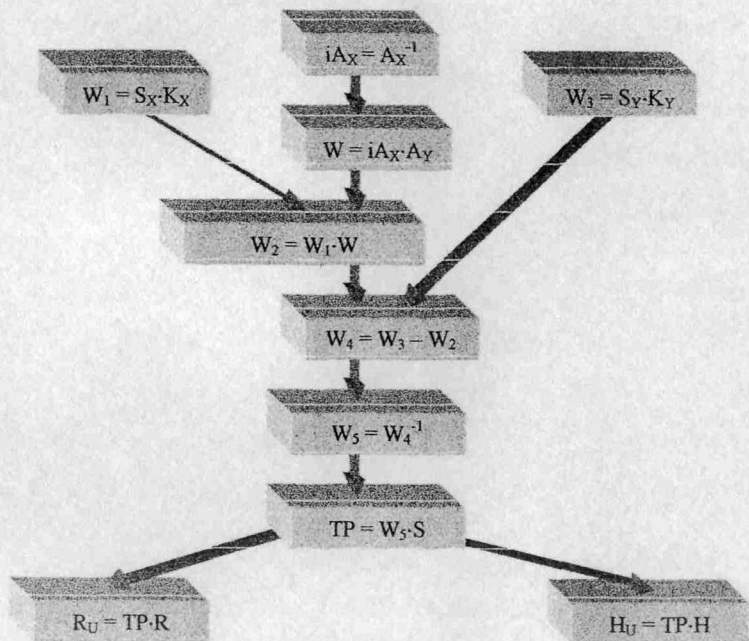


Рис.1. Алгоритм ГУ.

Подставив (4), (5) в (2), получим уравнение потоков в узлах

$$A_X X + A_Y Y = 0,$$

которое разрешается относительно потоков в ветвях дерева

$$X = -WY$$

с помощью матричной операции

(7)

$$W = A_X^{-1} A_Y. \quad (8)$$

Приняв во внимание, что $Z=(Z_X, Z_Y)^T$, подставим (4), (6) в (3):

$$S_X K_X \frac{dX}{dt} + S_Y K_Y \frac{dY}{dt} + SRZ = SH. \quad (9)$$

Подставив значение вектора X из (7) в (9), разрешим последнее относительно вектора производных

$$\frac{dY}{dt} = H_U - R_U Z \quad (10)$$

путем выполнения следующих матрично-векторных операций

$$H_U = (S_Y K_Y - S_X K_X W)^{-1} \cdot SH, \quad R_U = (S_Y K_Y - S_X K_X W)^{-1} \cdot SR.$$

Блок-схема алгоритма генерирования уравнений представлена на рис.1. Здесь выделена топологическая матрица

$$TP = (S_Y K_Y - S_X K_X W)^{-1} \cdot S \quad (11)$$

Операция $H_U = TP \cdot H$ при $H = \text{const}$ (тестовые варианты моделей) выполняется в генераторе уравнений, а в реальных объектах с $H = (H_X(X), H_Y(Y))^T$ она переносится в решатель уравнений.

3. Алгоритмы решателя уравнений (РУ)

Сгенерированные уравнения (7), (10) представлены в форме, которая требуется для численного решения. Построим алгоритмы функционирования решателя уравнений так, чтобы пользователь мог выбрать параллельную реализацию методов Эйлера, Адамса-Башфорта второго и Рунге-Кутты четвертого порядков. Эти методы реализованы в современных языках моделирования (SIMULINK, ACSL, ISRSIM и др.). Их использование в ПОПМС соответствует принципу преемственности параллельных сред с последовательными средствами моделирования динамических систем [1]. Анализ системы уравнений (7), (10) показывает, что в алгоритме РУ можно выделить следующую группу векторно-матричных операций, которые являются общими для всех трех указанных методов решения дифференциальных уравнений, а именно:

- 1) задание вектора начальных условий $Y(i) = Y(0)$, вычисление зависящего от него вектора

$$X(i) = -WY(i) \quad (12)$$

и формирование начального значения вектора $Q(i) = (X(i), Y(i))^T$; при организации i -го цикла вычислений эти операции соответствуют $i=0$;

- 2) вычисление вектора правой части уравнения (10) на i -ом шаге ($i=0,1, \dots$)

$$V(i) = H_U(i) - V_1(i), \quad (13)$$

где $H_U(i) = TP \cdot H(i)$ – вектор преобразованных характеристик вентиляторов; $V_1(i) = R_U \cdot Z(i)$ – вектор преобразованных потерь давлений на аэродинамических сопротивлениях;

$$Z(i) = (Z_X(i), Z_Y(i));$$

$$Z_X(i) = (X_1(i) \cdot | X_1(i)|, \dots, X_{n-1}(i) \cdot | X_{n-1}(i)|);$$

$$Z_Y(i) = (Y_1(i) \cdot | Y_1(i)|, \dots, Y_\gamma(i) \cdot | Y_\gamma(i)|), \gamma = m-n+1;$$

$$H(i) = (H_{X_1}(i), \dots, H_{X_{n-1}}(i), H_{Y_1}(i), \dots, H_{Y_\gamma}(i))^T,$$

$$H_{X_j}(i) = H_{0X} - aX_j^2(i) - bX_j(i), j = 1, 2, \dots, n-1;$$

$$H_{Y_k}(i) = H_{0Y} - aY_k^2(i) - bY_k(i), k = 1, 2, \dots, m-n+1;$$

Здесь a, b – коэффициенты аппроксимации характеристик вентиляторов; в общем случае задаются диагональные матрицы этих коэффициентов.

Вычисление $Y(i+1)$ для каждого шага итераций различными методами представлено на рис.2. Для реализации метода Рунге-Кутты дополнительно вычисляются $K_2(i), K_3(i), K_4(i)$.

Число повторений цикла определяется заданной точностью решения

$$\Delta Q(i) = |Q(i) - Q(i-1)| \leq \delta$$

и требуемой длительностью моделирования.

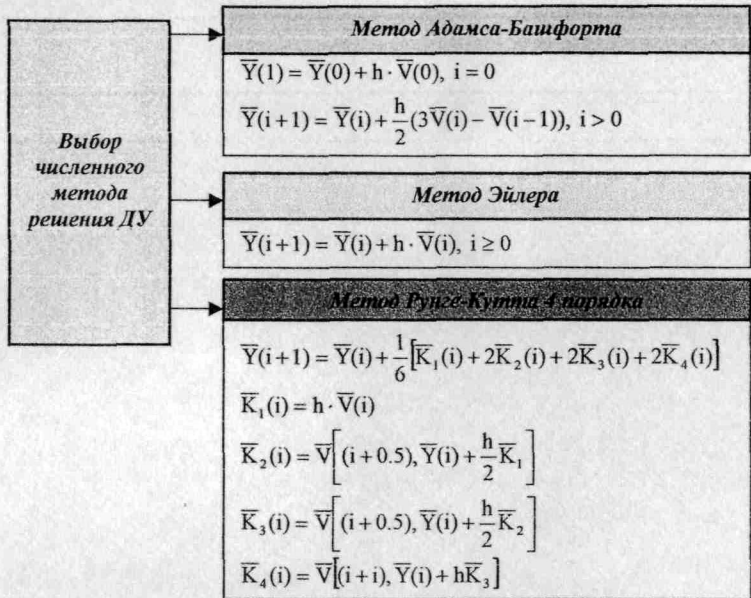


Рис.2. Алгоритмы решения дифференциальных уравнений.

4. Имплементация алгоритмов

ГУ и РУ сетевого объекта реализуются в параллельных SIMD- и MIMD-системах. При SIMD-имплементации алгоритмов использован язык параллельного программирования PARALLAXIS. В MIMD-компоненте моделирующей среды алгоритмы реализованы на языке С при MPI-организации обмена между процессорами.

При имплементации на языке PARALLAXIS используются параллельные алгоритмы умножения, инверсии и транспонирования матриц и векторов. Требуемое поле процессорных элементов (ПЭ) представляется двумерной решеткой с соответствующими связями, которые описываются выражениями

```
CONFIGURATION grid[1..m_max],[1..m_max];
CONNECTION right: grid[i, j] <-> grid[i, j+1].left;
           down: grid[i, j] <-> grid[i+1, j].up;
multi1:    grid[i, j] -> grid[i, 1..m_max].multi1;
multi2:    grid[i, j] -> grid[1..m_max, j].multi2;
trans:     grid[i, j] -> grid[j, i].trans;
```

Матрицы A , S , векторы R и H распределяются на поле $\Pi\Xi$ таким образом, чтобы индексы процессоров DIM1 и DIM2 совпадали соответственно с индексами локальных компонент; процессорный элемент $\Pi\Xi_{ij}$ хранит значения A_{ij} , S_{ij} в своей локальной памяти. Векторы рассматриваются как специальные матрицы, которые являются одномерными.

Ниже даны фрагменты PARALLAXIS-программ генератора и решателя уравнений.

Фрагмент ГУ:

PARALLEL

```

...
iAx := InverseMatrix(Ax, d);
W := MatrixProdukt(iAx, Ay, d, d, g);
W1 := MatrixProdukt(Sx, Kx, g, d, d);
W2 := MatrixProdukt(W1, W, g, d, g);
W3 := MatrixProdukt(Sy, Ky, g, g, g);
W4 := W3 - W2;
W5 := InverseMatrix(W4, g);
TP := MatrixProdukt(W5, S, g, g, m);
Ru := MatrixProdukt(TP, R, g, m, m);
Hu := MatrixProdukt(TP, H, g, m, m);

```

ENDPARALLEL;

Фрагмент PY:

PARALLEL

```

t := 0.0;
LOOP
  Q_B := MatrixProdukt(-W, Q_Y, d, g, l);
  (* Вектор Q_B подставляется в Q *)
  IF (dim2 = 1) AND (dim1 <= d) THEN
    Q := Q_B;
  END;
  Z := Q*ABS(Q);
  H := R*Z + K*(Q-Q_alt) / tstep; Q_alt := 0;
  Ausgabe(H, Q, t);
  IF t > tsim THEN EXIT; END;
  delta_Q_Y := MatrixProdukt(TP, Ho - R*Z, g, m, l);
  (* Вычисление нового состояния Q_Y *)
  Q_Y := Q_Y + Integral(delta_Q_Y, delta_Q_Y_alt, t);
  delta_Q_Y_alt := delta_Q_Y;
  (* Вектор Q_Y подставляется в Q *)
  PROPAGATE.down^d(Q_Y, Q_hilf);
  IF (dim1 > d) AND (dim2 = 1) THEN Q := Q_hilf; END;
END;
ENDPARALLEL;

```

Процедура вычисления интеграла по методу Адамса-Башфорта 2 порядка :

```
PROCEDURE Integral(VECTOR delta_f, delta_f_alt : REAL;  
                   SCALAR t : REAL) : VECTOR REAL;  
  
  BEGIN  
    IF t = 0.0 THEN  
      RETURN h*delta_f;  
    ELSE  
      RETURN h1*delta_f - h2*delta_f_alt;  
    END;  
  END Integral;
```

5. Заключение

Разработанные генератор и решатель уравнений являются центральной составной частью проблемно ориентированной параллельной моделирующей среды для сетевых объектов с сосредоточенными параметрами. Эксперименты с тестовой моделью сети подтвердили работоспособность ГУ и РУ. Дальнейшие работы направлены на реализацию всех методов решения в SISD-, SIMD- и MIMD-системах, на разработку подсистем диалога, визуализации, на решение практических задач в моделирующей среде.

Литература

- [1] Святный В.А. Проблеми паралельного моделювання складних динамічних систем. Наукові праці ДонДТУ, Серія ІКОТ, вип.6., 1999, с.6-14.
- [2] Перерва А.А. Топологический анализатор параллельной модели сетевого объекта. Наукові праці ДонДТУ, Серія ІКОТ, вип.6., 1999, с.73-78.
- [3] Абрамов Ф.А., Фельдман Л.П., Святный В.А. Моделирование динамических процессов рудничной аэрологии. – Киев : Наукова думка, 1981.
- [4] Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране, М.: Мир, 1977 г.
- [5] G. Eberhardt. Entwicklung massiv paralleler Algorithmen zur Simulation dynamischer Systeme. Studienarbeit, IPVR, Universität Stuttgart, 1992.