GENERIERUNG UND PARALLELE LÖSUNG VON SIMULATIONSMODELLEN FÜR NETZOBJEKTE MIT VERTEILTEN PARAMETERN

Svjatnyj V.A., Moldovanova O.V., Cheptsov O.O.
Fakultät für Rechentechnik und Informatik (FRTI)
Nationale Technische Universität Donezk, Artemstr. 58, 83000, Ukraine
svjatnyj@cs.dgtu.donetsk.ua

Zeitz M.
Institut für Systemdynamik und Regelungstechnik (ISR), Universität Stuttgart,
Postfach 801140, 70511 Stuttgart
zeitz@isr.uni-stuttgart.de

Rothermel K.
Institut für Parallele und Verteilte Systeme (IPVS), Universität Stuttgart,
Universitätsstraße 38, 70569 Stuttgart
rothermel@informatik.uni-stuttgart.de

Kurzfassung

Svjatnyj V.A., Moldovanova O.V., Cheptsov O.O., Zeitz M., Rothermel K. Generierung und parallele Lösung von Simulationsmodellen für Netzobjekte mit verteilten Parametern. *Die Entwicklung von parallelen Simulationsmodellen für komplexe dynamische Netzobjekte mit verteilten Parametern verlangt eine umfangreiche Rechnerunterstützung. Zur rechnerunterstützten Generierung der Simulationsgleichungen werden verschiedene Approximationsansätze zur Diskretisierung der Ortskoordinaten betrachtet und die Algorithmen für die Linienmethode in dem Gleichungsgenerator implementiert. Der parallele Gleichungslöser für die differential-algebraischen Simulationsgleichungen ist wie der Gleichungsgenerator ein wichtiger Bestandteil der an der TU Donezk entwickelten parallelen Simulationsumgebung für dynamische Systeme.*

1. Einführung

Beim Aufstellen von Simulationsmodellen für Netzobjekte mit verteilten Parametern (NOVP) treten folgende Probleme auf: fehlerfreie Erstellung von hochdimensionalen Gleichungssystemen; aufwändige Vorverarbeitung auf die numerisch lösbaren Modellformen; operative Arbeit mit multidimensionalen Arrays der NOVP-Parameter. In diesem Beitrag werden ein Gleichungsgenerator und ein Gleichungslöser für die Behandlung der obigen Probleme dargestellt.

2. Gleichungsgenerator

Die Approximationsverfahren der ursprünglichen Gleichungen und Randbedingungen bilden die Basis für die Entwicklung von geeigneten Simulationsmodelle. Hierzu werden die folgenden drei Approximationsansätze [1], [2], [3] verwendet: das Linien-, das Frequenz- und das Charakteristikenverfahren.

Die Topologie des aerodynamischen Netzobjektes wird als Graph G(U, V) mit einer Menge von Knoten U = n und Zweigen V = m dargestellt. Aerodynamische Prozesse werden durch die Luftstrom- und Druckvektoren Qi bzw. Pi charakterisiert und im i-ten Zweig durch das Gleichungssystem

$$\begin{cases} -\dfrac{\partial P_i}{\partial \xi} = r_i Q_i^2 + \dfrac{\rho}{F_i}\dfrac{\partial Q_i}{\partial t} \\ -\dfrac{\partial P_i}{\partial t} = \dfrac{\rho a^2}{F_i}\dfrac{\partial Q_i}{\partial \xi} \end{cases}, \tag{1}$$

beschrieben. Hier sind r_i – spezifischer aerodynamischer Wiederstand, ρ – Luftdichte, F_i – Schnittfläche, a – Schallgeschwiendigkeit, ξ – laufende Koordinate.

Die Approximation des Gleichungssystems (1) mit dem Linienverfahren liefert für das k-te Element des i-ten Zweiges mit einer Länge $\Delta\xi_{ik}$ die folgenden Differenzengleichungen [1]:

$$\begin{cases} \dfrac{P_{ik} - P_{i,k+1}}{\Delta\xi_{ik}} = r_{ik} Q_{ik}^2 + \dfrac{\rho}{F_{ik}}\dfrac{dQ_{ik}}{dt} \\ \dfrac{dP_{i,k+1}}{dt} = \dfrac{\rho a^2}{F_{ik}}\dfrac{Q_{ik} - Q_{i,k+1}}{\Delta\xi_{ik}} \end{cases} \tag{2}$$

Für das Netzobjekt mit verteilten Parametern werden die Vektoren Q(X,Y), P(PX, PY) entsprechend (2) ins Multivektorsystem umgewandelt, d. h. jeder Zweig des Graphenbaums X_j (j = 1…n-1) und Antibaums Y_q (q = 1…γ) wird durch zwei Vektoren dargestellt: $X_j = (X_{j1}, X_{j2}, \ldots, X_{jM})^T$ – Luftstrom, $XP_j = (XP_{j1}, XP_{j2}, \ldots, XP_{jM+1})^T$ – Druck im Baumzweig j; $Y_q = (Y_{q1}, Y_{q2}, \ldots, Y_{qM})^T$ – Luftstrom, $YP_q = (YP_{q1}, YP_{q2}, \ldots, YP_{qM+1})^T$ – Druck im Antibaumzweig q, wobei M die Menge der Approximierungselemente in den Zweigen X_j, Y_q ist und $\gamma = m - n + 1$ eine zyklomatische Zahl des Graphen bedeutet . Für das gesamte Netzobjekt werden die Vektoren X, Y, XP, YP als Matrizen betrachtet und deren Elemente aus folgenden Gleichungen berechnet:

$$\begin{cases} \dot{X}_{jk} = \alpha_{X_j}(XP_{jk} - XP_{j,k+1}) - \beta_{X_j} X_{jk}\left|X_{jk}\right| \\ \dot{XP}_{j,k+1} = g_{X_j}(X_{jk} - X_{j,jk+1}) \end{cases}, \tag{3}$$

$$\begin{cases} \dot{Y}_{qs} = \alpha_{Y_q}(YP_{qs} - YP_{q,s+1}) - \beta_{Y_q} Y_{qs}\left|Y_{qs}\right| \\ \dot{YP}_{q,s+1} = g_{Y_q}(Y_{qs} - Y_{q,s+1}) \end{cases}. \tag{4}$$

Hier sind $\alpha_{Xj}, \alpha_{Yq}, \beta_{Xj}, \beta_{Yq}, g_{Xj}, g_{Yq}$ die aerodynamischen Parameter, die durch folgende Formeln definiert werden:

$$\alpha_{ik} = \frac{F_{ik}}{\rho \Delta\xi_{ik}}, \ \beta_{ik} = \frac{r_{ik} F_{ik}}{\rho}, \ g_{ik} = \frac{\rho a^2}{F_{ik} \Delta\xi_{ik}}. \quad (5)$$

Eine kompakte Darstellung der Differentialgleichungssysteme (3), (4) hat die Vektor-Form:

$$\begin{cases} \dot{X} = \alpha_X \Delta XP - \beta_X ZX \\ \dot{XP} = g_X \Delta X \\ \dot{Y} = \alpha_Y \Delta YP - \beta_Y ZY \\ \dot{YP} = g_Y \Delta Y \end{cases}, \quad (6)$$

wobei gilt: $\Delta X_{jk} = X_{jk} - X_{j,k+1}$, $\Delta Y_{qs} = Y_{qs} - Y_{q,s+1}$, $\Delta XP_{jk} = XP_{jk} - XP_{j,k+1}$, $\Delta YP_{qs} = YP_{qs} - YP_{q,s+1}$, $ZX_{jk} = X_{jk} \mid X_{jk} \mid$, $ZY_{qs} = Y_{qs} \mid Y_{qs} \mid$; α_X, α_Y, β_X, β_Y, g_X, g_Y sind Diagonalmatrizen mit den aerodynamischen Parametern.

Bei dem Frequenzverfahren wird ein simuliertes Netz als regelbares Netzobjekt betrachtet. Die Zweige des Netzes werden in einer genügend großen Raumdimension angesetzt, weshalb die Übergangsprozesse den Luftfluss in jedem Punkt beschreiben. Das erhaltene System von gewöhnlichen Differentialgleichungen beschreibt die Frequenzeigenschaften der diskreten Abschnitte der realen Regelungszweige mit einer Länge bis zu 400 m in einer für die Praxis ausreichenden Genauigkeit [2].

Die Idee der Charakteristikenverfahren ist an den physikalischen Prozess der Wellenausbreitung angelehnt [3]. Die Strömungsvorgänge stellen Wellenprozesse dar, bei denen sich die stetigen oder unstetigen Störungen auf charakteristischen Kurven ausbreiten. Entlang dieser Kurven, welche als Charakteristiken bezeichnet werden, können die Strömungszustände des zu untersuchenden Systems bestimmt werden. Mit den Charakteristiken und durch die Verträglichkeitsbedingungen sind alle notwendigen Gleichungen gegeben, auf denen die numerische Lösung von Differentialgleichungssystemen beruhen. Aus mathematischer Sicht verkörpert diese Äquivalenz den wesentlichen Inhalt der Charakteristikentheorie, die eine Lösungstheorie für hyperbolische partielle Differentialgleichungen darstellt.

Der Gleichungsgenerator für das parallele Simulationsmodell von Netzobjekten stellt ein Programm dar, das auf der Grundlage von gegebenen Daten und Angaben des Topologieanalysators sowie Linien-Approximationsverfahrens ein Gleichungssystem (6) mit den notwendigen Randbedingungen generiert. Auf Wunsch des Modellentwicklers wird das Ergebnis der Generierung visualisiert und dokumentiert. Der Gleichungsgenerator ist in C-Code und in MATLAB implementiert.

3. Gleichungslöser

Die Approximation der ursprünglichen Gleichungen durch das Linienverfahren liefert 2*m*M gewöhnliche Differentialgleichungen. Der Gleichungsgenerator stellt alle Gleichungen in einer für die numerische Lösung geeigneten Form [4] dar. Für die Entwicklung der parallelen MIMD-Gleichungslöser werden die folgenden Definitionen eingeführt:

Definition1: Virtueller MIMD-Prozess ist ein relativ autonomes Programm, das dem Lösungsalgorithmus des Gleichungssytemteils entspricht und über eine Schnittstelle mit den benachbarten Prozessen kommuniziert.

Definition 2: Virtuelles paralleles Simulationsmodell ist eine Abstraktion, die aus dem von dem Gleichungsgenerator gelieferten Gleichungssystem und einer nach einem Parallelisierungsansatz entwickelten Struktur der MIMD-Prozesse für parallele Gleichungslösung besteht.

Definition 3: Die durch einen Graph oder eine Matrix dargestellten logischen Verbindungen zwischen den MIMD-Prozessen stellen ein virtuelles Verbindungsnetzwerk (Kommunikationsgraph oder Kommunikationsmatrix) dar.

Definition 4: Virtuelle parallele Rechnerarchitektur ist eine lose gekoppelte, nach dem MIMD-Prinzip funktionierende und nicht beschränkte heterogene Menge der vollfunktionellen Prozessoren mit lokalem Speicher, die gemeinsam eine Lösung des Simulationsproblems mit dem algorithmisch bedingten Datenaustausch über ein programmgesteuertes virtuellesVerbindungsnetzwerk durchführen.

Definition 5: Zielrechnerarchitektur (ZRA) ist die dem Modellentwickler zur Verfügung stehende lose gekoppelte, nach dem MIMD-Prinzip funktionierende, beschränkte heterogene Menge der installierten vollfunktionellen Prozessoren mit lokalem Speicher und verfügbarem vordefinierten programmgesteuerten Verbindungsnetzwerk.

Definition 6: Devirtualisierung des virtuellen parallelen Simulationsmodells ist eine Umwandlung, die zur Modellrealisierung eindeutig auf gegebenen ZRA führt und aus folgenden Grundfunktionen besteht: Darstellung der vorhandenen ZRA-Ressourcen; Zuordnung "virtuelle Prozesse – reale Prozessoren", "virtuelle Schnittstellen – Kommunikationsgraph (Kommunikationsmatrix) – ZRA - Verbindungsnetzwerk" unter Berücksichtigung von Kriterien der Lastbalancierung, Netzwerkanpassung, Minimierung des Datenaustauschaufwandes; Formierung der Spezifikation von den Prozessoren zugeordneten MIMD-Prozessen für den parallelen Code-Generator.

Für die Entwicklung der virtuellen parallelen Simulationsmodelle des Netzobjektes mit verteilten Parametern werden die möglichen Parallelisierungsansätze in der Reihenfolge, die der Granularität der virtuellen MIMD-Prozesse entspricht, betrachtet:

Ansatz 1. Jede Gleichung für die Berechnung der Luftströme und der Druckwerte im k-ten Stützpunkt des j-ten Zweiges des Netzobjektes ($1 \le k \le M$, $1 \le j \le m$) wird in einem ihr zugeordneten MIMD-Prozess gelöst. Dabei unterscheidet man zwischen den Q(X, Y)- und P(XP, YP)-Prozesse, die entsprechend dem ausgewählten numerischen Verfahren die Cauchy-Aufgabe (3), (4) bezüglich der Variablen $X_{jk}(i+1)$, $Y_{jk}(i+1)$ und $XP_{jk}(i+1)$, $YP_{jk}(i+1)$ lösen. Hier ist i die laufende Iterationsnummer des numerischen Verfahrens. Die Topologie des parallelen Simulationsmodells spiegelt den gesamten Graph des Netzobjektes wider und besteht aus für den j-ten Zweig zugeordneten Doppelketten der X-, XP- bzw. Y-, YP-Prozesse. Für die Abbildung des virtuellen parallelen Simulationsmodells auf virtuelle parallele Rechnerarchitektur werden der benötigte virtuelle Datenaustausch und die darauf folgende Synchronisation der parallelen Abläufe betrachtet. Die Schnittstelle des X- bzw.Y-Prozesses beinhaltet drei Eingänge – als äußere: XP_k, $-XP_{k+1}$ (YP_k, $-YP_{k+1}$) und als innere: $-X_k*|X_k|$ ($-Y_k*|Y_k|$) – und den Ausgang X_k (Y_k). Die Schnittstelle des XP- bzw. YP- Prozesses hat zwei äußere Eingänge X_k, $-X_{k+1}$ (Y_k, $-Y_{k+1}$) und einen Ausgang XP_{k+1} (YP_{k+1}). Am Ende der Berechnungsschritte wird folgender Datenaustausch zwischen den Nachbarprozessen durchgeführt:

- Prozess X_k sendet seinen Wert $X_k(i+1)$ dem Prozess XP_{k+1} und empfängt die Werte $XP_k(i+1)$ vom Prozess XP_k und $-XP_{k+1}(i+1)$ vom Prozess XP_{k+1};
- Prozess XP_{k+1} sendet seinen Wert $-XP_{k+1}(i+1)$ dem Prozess X_k, Wert $XP_{k+1}(i+1)$ – dem Prozess X_{k+1} und empfängt die Werte $X_k(i+1)$ vom Prozess X_k und $-X_{k+1}(i+1)$ vom Prozess X_{k+1}.

Die virtuelle parallele Rechnerarchitektur, die dem 1.Ansatz entspricht, hat folgende Charakteristiken:

- Anzahl der Prozessoren (Abbildung Prozess – Prozessor ist 1 zu 1) ist 2mM;
- Verbindungsnetzwerk hat eine Gitterstruktur, die den Datenaustausch zwischen den Nachbarprozessoren erleichtert; man kann auch ein spezielles VNW in Betracht ziehen;
- Synchronisationsmechanismus erlaubt den Datenaustausch nach dem Ende des längsten Prozesses.

Ansatz 2. Die Gleichungen (3), (4) für k-ten Stützpunkt des j-ten Zweiges des Netzobjektes ($1 \le k \le M$, $1 \le j \le m$) werden in dem ihnen zugeordneten k-ten MIMD-Prozess gelöst. Die Topologie des virtuellen parallelen Simulationsmodells spiegelt den gesamten Graph des Netzobjektes wider und besteht aus für den j-ten Zweigen zugeordneten Ketten der X-, XP- bzw. Y-, YP-Prozesse , d. h. aus insgesamt m*M Prozessen.

Ansatz 3. Die M Gleichungen (3) und (4) für jeden j-ten Zweig des Netzobjektes ($1 \le j \le m$) werden in dem ihnen zugeordnetem j-ten MIMD-Prozess gelöst. Die Topologie des virtuellen parallelen Simulationsmodells spiegelt den gesamten Graph des Netzobjektes wider und besteht aus m Prozessen.

Ansatz 4. Das Gleichungssystem (6) des Netzobjektes für j = 1…n-1, q = 1…γ wird entsprechend der Partitionierung des Netzgraphen [3] in Gs Subsysteme zerlegt und in den Gs zugeordneten MIMD-Prozessen gelöst. Die Topologie des virtuellen parallelen Simulationsmodells spiegelt ein in Gs Teilgraphen zerlegtes Netzobjekt wider und besteht aus Gs Prozessen, die minimierte Verbindungen haben.

Ein definierter Devirtualisierungsvorgang führt zur Implementierung des Simulationsmodells auf der Zielrechnerarchitektur. Der entwickelte Gleichungslöser stellt ein MPI-Programm dar, das zyklisch einen dem Parallelisierungsansatz entsprechenden Algorithmus der numerischen Lösung des Matrix-Vektor-Gleichungssystems realisiert. Für die Parallelisierung wird die SPMD-Organisation für den Löser ausgewählt. Das Programm wird auf mehreren MIMD-Prozessoren verteilt. Im Hauptteil des Programms werden die notwendigen Variablen deklariert und die MPI-Initialisation durchgeführt. Mit Hilfe von MPI-Funktionen wird vom Benutzer die angeforderte Prozessoranzahl zugewiesen. Jeder Algorithmus der obigen Parallelisierungsansätze sieht im Löser den Austausch zwischen Prozessoren vor. Die erhaltenen Lösungsdaten werden für die nachfolgende Analyse und Visualisierung in Dateien gespeichert.

Bei der parallelen Simulation von NOVP wird die höchste Produktivität erreicht, wenn der Simulationsalgorithmus mit der Struktur des Parallelrechners und dessen Implementierungsmöglichkeiten vereinbar ist. Die gebräuchlichen Effizienzfaktoren für parallele Simulationsalgorithmen sind der Geschwindigkeits-Gewinnfaktor (Speedup) Sp und die Effizienz Ep. Dabei werden für die Lösung von n Aufgaben die Ausführungszeiten T1(n) und Tp(n) mit dem besten sequentiellen Algorithmus bzw. einem parallelen Algorithmus auf p Prozessoren betrachtet. Die Effizienzverluste werden von der nicht idealen Parallelität der Algorithmen sowie durch die Datenaustauschzeiten verursacht.

4. Modelluntersuchungen

Gleichungsgenerator und Gleichungslöser werden für ein dynamisches Netzobjekt mit verteilten Parametern in einer Grubenbewetterungsanlage getestet. Der Graph des Objektes enthält 4 Baumzweige und 4 Antibaumzweige. Ein Knoten PA ist mit der Atmosphäre und ein Knoten PV mit einem Ventilator verbunden. Die Untersuchungen zeigen, dass der vorgeschlagene Algorithmus des Gleichungsgenerators eine fehlerfreie und schnelle Formierung von Matrix-Vektor-Gleichungen ermöglicht. Die Simulationsexperimente lassen die vorgeschlagenen Parallelisierungsansätze vergleichen.

5. *Zusammenfassung und Ausblick*

Die vorgestellten Untersuchungen und Ergebnisse liefern einen Beitrag zur parallelen Modellierung und Simulation von komplexen Netzobjekten mit verteilten Parametern. Geplante weitere Entwicklungen konzentrieren sich auf weitere industrielle Anwendungen der entwickelten Werkzeuge, die in der parallelen problemorientierten Simulationsumgebung für dynamische Netzobjekte mit verteilten Parametern an der TU Donezk implementiert sind.

LITERATUR

1. *Svjatnyj, V.A., Moldovanova, O.V.*: Gleichungsgenerator von parallelen Modellen für dynamische Netzobjekte mit verteilten Parametern - Problems of Simulation and Computer-Aided Design of Dynamic Systems. Collected Volume of Scientific Papers. Donetsk State Technical University, Donetsk, 1999, S. 135 – 141 (in Russisch).
2. *Lapko, V.V.*: Ein Modellierungsansatz für aerodynamische Netze mit verteilten Parametern. 13. Symposium ASIM´1999, Tagungsband, 1999, S. 325 – 330.
3. *Hanf, G.*: Modellierung und Simulation der instationären Grubenbewetterung auf verteilten Rechnerarchitekturen. Fortschr.-Ber. VDI Reihe 10, Nr. 699. Düsseldorf: VDI Verlag 2002.
4. *Zeitz, M.*: Simulationstechnik. In: Chemie-Ingenieur-Technik, 59, 1987, S. 464 – 469.

Дата надходження до редколегії: 13.11.2003 р.