

文章编号:1001-9081(2010)11-2921-03

## 双曲平衡点的二维流形计算方法

陈燕<sup>1</sup>, 陈晓丽<sup>2</sup>, 王凤琴<sup>1</sup>

(1. 郑州轻工业学院 计算机与通信工程学院, 郑州 450002; 2. 南阳理工学院 艺术设计系, 河南 南阳 473004)

(ychen2009@sohu.com)

**摘要:**提出了一种改进的向量场二维流形计算方法。新算法通过解初值问题来计算轨道,然后等轨道弧长向外扩展来增长流形,保证了计算的快速性;应用曲率控制技术实现了轨道上的离散网格点的优化分布;在进行网格点插值时综合运用了距离控制和曲率控制,用重新计算轨道的方法来确定插值点的位置,一定程度上克服了原二维流形计算方法的不能保证插值点精度的弱点。仿真结果也表明,新算法能够很好地应用于二维稳定流形的计算。

**关键词:**动力系统;不变流形;双曲;洛伦兹流形

**中图分类号:** TP391.9; TP181 **文献标志码:** A

## Two-dimensional hyperbolic equilibrium manifold method

CHEN Yan<sup>1</sup>, CHEN Xiao-li<sup>2</sup>, WANG Feng-qin<sup>1</sup>

(1. School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou Henan 450002, China;

2. Department of Art, Nanyang Institute of Technology, Nanyang Henan 473004, China)

**Abstract:** The computation of manifold plays an important role in understanding the dynamics of a nonlinear system. This paper presented a revised algorithm for computing two dimensional manifolds of vector fields. The trajectory was grown by solving appropriate initial value problem, then the manifold was grown step by step with equal arc-length along the trajectory and points on the trajectory were selected according to the local curvature. For the first time, both curvature-control technique and distance constraints were employed when interpolation between mesh points was needed. And in this way, the accuracy of interpolated points was guaranteed and it was the main advantage over the existing methods. The simulation results show that the proposed algorithm is effective.

**Key words:** dynamical system; invariant manifold; hyperbolic; Lorenz manifold

### 0 引言

动力系统中的很多问题都可以近似用微分方程来描述,通过对微分方程的定性分析,直接研究其积分曲线或轨迹的几何性质及稳定性,就能够了解原动力系统的相关性质。流形计算属于研究动力系统的一种几何方法。通过对系统稳定和不变流形的刻画,有利于我们更深入地了解动力系统的性能,从而为解释系统的复杂动力特性提供依据。研究表明,流形计算对于解释混沌以及同宿、异宿现象都有一定的帮助,此外,流形计算在最优控制理论、电力系统稳定边界的研究中也有一定的应用。但稳定流形和不变流形一般不能够通过解析表达式直接求解,所以其数值计算方法就变得非常必要。

### 1 向量场双曲平衡点的二维流形常用计算方法

连续动力系统(即向量场)以常微分方程的形式给出:

$$\dot{x} = f(x) \quad (1)$$

其中:  $x \in \mathbb{R}^n$ ,  $f$  充分光滑。在研究过程中若  $f$  是非线性的,则要先对  $f$  在系统的平衡点处进行线性化处理,则式(1)可变形为:

$$\dot{x} = Ax \quad (2)$$

其中  $A = Df(x_0) = [\partial f_i / \partial x_j](x_0)$  为  $f$  在系统平衡点  $x_0$  处的雅可比矩阵。若  $A$  的特征值全部都不等于 0, 称  $x_0$  为系统的一个双曲平衡点。系统在  $x_0$  处的稳定和不变流形可定义为:

$$W^s(x_0) = \{x \in \mathbb{R}^n : \lim_{t \rightarrow -\infty} \varphi^t(x) = x_0\} = \bigcup_{t \leq 0} \varphi^t(W_{loc}^s(x_0)) \quad (3)$$

$$W^u(x_0) = \{x \in \mathbb{R}^n : \lim_{t \rightarrow \infty} \varphi^t(x) = x_0\} = \bigcup_{t \geq 0} \varphi^t(W_{loc}^u(x_0)) \quad (4)$$

其中:  $\varphi$  为微分方程的解,又叫做流(flow)。  $A$  的小于 0 的特征值对应的特征向量张成稳定特征值空间  $E^s(x_0)$ , 大于 0 的特征值对应的特征向量张成不稳定特征值空间  $E^u(x_0)$ 。稳定和不变流形定理<sup>[1]</sup>证明了在  $x_0$  的邻域内存在局部稳定流形  $W_{loc}^s(x_0)$  和局部不稳定流形  $W_{loc}^u(x_0)$ , 它们在  $x_0$  处分别与  $E^s(x_0)$  和  $E^u(x_0)$  相切。在本文只考虑稳定特征值或不变特征值个数为 2 的情况,也就是二维流形的计算。

关于二维流形的计算,国际上提出了很多不同思路的算法。其中轨道弧长法<sup>[2]</sup>计算速度较快,它通过改变独立变量,使得流形在各个方向上都能均匀增长,每经过一定的积分步数,它都要对环上的网格点进行调整,当网格点间的距离过大时,使用线性插值进行点的插入;由于流形上的动力学特性决定了网格点之间的距离,所以无法保证插值点的精度。测地线法<sup>[3]</sup>通过求解边值问题来确定网格点的位置,环与环之间的距离由环上流形变化最快的一点来决定,否则会漏掉流形细节,而且算法的总趋势是用小的环间距离来保证计算的精度,所以在环上其他流形变化相对较慢的部分就增加了不必要的计算量;此外,因为要对环间距离进行检查,当环上有

收稿日期:2010-05-19;修回日期:2010-07-20。 基金项目:河南省教育厅自然科学基金资助项目(2010B510023)。

作者简介:陈燕(1979-),女,河南南阳人,助教,硕士,主要研究方向:数字图像处理、流形计算; 陈晓丽(1977-),女,河南南阳人,助教,主要研究方向:信号处理; 王凤琴(1980-),女,河南开封人,讲师,博士,主要研究方向:数字图像处理。

个别离散点不满足角度约束条件时,就要调整距离,然后重新计算环上的所有离散点,这样的话,为了得到一个满足条件的环可能就要反复计算好几次,这也增加了额外的计算量;它的优势在于精度较高。本文算法主要就是基于这两种算法提出的。

其他方法还包括使用 AUTO<sup>[4]</sup> 进行轨道延拓, PDE 方法<sup>[5]</sup> 和肥胖轨道法<sup>[6]</sup> 以及细分法<sup>[7]</sup>, 但是这些方法计算比较复杂,而且在形式上不够直观<sup>[8]</sup>。

## 2 算法介绍

新算法结合了轨道弧长法和测地线法的优点。以不稳定流形  $W^u(x_0)$  的计算为例(此时流  $\varphi$  的迭代值  $t$  应取正值, 计算稳定流形  $W^s(x_0)$  时只需将  $t$  取为负值即可), 步骤如下:

1) 初始化。在不稳定特征值空间  $E^u(x_0)$  上以  $x_0$  为圆心,  $\delta$  为半径作圆( $\delta$  不宜取得过小, 否则流形增长速度过慢, 增加了积分时间; 另一方面, 适当减小  $\delta$  可以增加计算精度)。将该初始圆  $S_\delta(x_0)$  作为局部不稳定流形  $W_{loc}^u(x_0)$ , 在  $S_\delta(x_0)$  上取  $N$  个均匀分布的点, 点的集合记做  $C_0$ ; 将相邻的点用线段连起来, 就构成了流形的第一个环。弧长  $ARC = \delta$ 。

2) 增长流形。假定我们现在已经计算得到了第  $i$  个环  $C_i$ , 现在要从当前环  $C_i$  出发计算  $C_{i+1}$ 。对于每个点  $p \in C_i$ , 对式(1)使用四阶 Runge-Kutta 方法求解初值问题, 计算一段弧长为  $\Delta$  的轨道, 轨道的弧长可由轨道上相邻离散点间的距离之和近似代替。

得到轨道后, 要对轨道上的离散点进行优化选取, 从而能够用最少的点有效地表示轨道, 本文采用了文献[3, 9]中提出的曲率控制的方法。

本文方法与文献[3]中的方法应用曲率约束条件的不同之处在于, 文献[3]中的方法相邻两个环上对应的网格点之间不存在其他网格点, 曲率约束条件是用来检查环之间的距离是否合适, 而距离受制于环上流形变化最快的一点; 而本文方法则是先计算一段弧长一定的轨道, 然后在轨道上用曲率约束条件来选取一些优化分布的点来表示轨道, 相邻两个环对应网格点之间可能存在多个网格点, 环与环之间的距离不受曲率约束条件的限制, 所以即使提高曲率控制的精度, 也不会使流形增长过慢。

考虑轨道上相邻3个离散点之间的角度<sup>[10]</sup>, 如图1所示。

由图易得:

$$\sin\left(\frac{\alpha}{2}\right) = \frac{\|\bar{p} - p_{k-1}\|}{2\|p_k - p_{k-1}\|} \quad (5)$$

其中:

$$\bar{p} = p_k + \frac{\|p_k - p_{k-1}\|}{\|p_k - p_{k+1}\|}(p_k - p_{k+1}) \quad (6)$$

$\bar{p}$  与线段  $p_k p_{k+1}$  共线, 且  $\|\bar{p} - p_k\| = \|p_{k-1} - p_k\|$ , 即点  $p_k$  到点  $p_{k-1}$  和到点  $\bar{p}$  的距离相等。

当  $\alpha$  较小时,  $\sin\left(\frac{\alpha}{2}\right) \approx \frac{\alpha}{2}$ , 式(5)化简为:

$$\alpha = \frac{\|\bar{p} - p_{k-1}\|}{\|p_k - p_{k-1}\|} \quad (7)$$

现在检查  $\alpha$  是否满足下列约束条件:

$$\begin{cases} \alpha_{\min} < \alpha < \alpha_{\max} \\ (\Delta\alpha)_{\min} < \Delta_k\alpha < (\Delta\alpha)_{\max} \end{cases} \quad (8)$$

其中  $\Delta_k = \|p_{k+1} - p_k\|$ 。第一个条件确保  $\alpha$  保持较小值, 防止漏掉流形细节, 第二个条件用来控制局部插值误差。由于我们

是对轨道上的点进行优化选取, 所以当  $p_{k-1}$  和  $p_k$  确定后, 只要沿着轨道向前搜索, 并使  $p_{k+1}$  满足式(8)所示的条件即可。这点与文献[3]是不同的。各条轨道的末点就构成了新环  $C_{i+1}$ 。

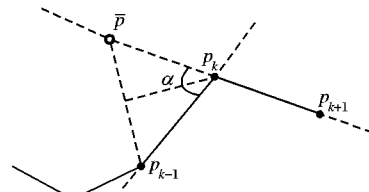


图1 曲率约束条件

当对轨道上第1和第2两个点之间的曲率约束检查时, 如果严格按照上面所述的步骤, 则必须提供它们之前的点的位置信息, 以及相应的距离条件  $\Delta_i$ ; 由于每条轨道在起始部分都需要进行与此相同的操作, 所以我们就必须保存大量的从上一环出发的轨道的信息, 这样使得数据操作变得复杂。一般情况下, 轨道上第1和第2两个点之间的距离  $\Delta_{12}$  总能满足条件  $\Delta_{12} < \Delta_{\min}$ , 它本身能够满足精度条件, 所以就直接将前两点作为轨道上的点, 然后从第3个点开始严格执行上述控制条件。这样做虽然可能产生一些多余的网格点, 但是就不必再存储上一环中的点的信息, 减少了内存的使用量, 而且使得所采用的数据结构得到了简化, 操作起来也更简便。

3) 调整网格点。网格点的调整分为两个步骤来完成。

距离检查。这一步主要用来对网格点进行粗调整, 保证同一环上相邻的网格点的距离保持适中。检查  $C_{i+1}$  中相邻网格点间的距离  $d$ , 当距离  $d < \Delta_{\min}$  时, 说明两个网格点靠得太紧, 就需要删掉一个点。相应的若距离  $d > \Delta_{\max}$ , 就需要在两个网格点之间插入一个新的网格点, 在这里使用第2)步的方法从  $C_i$  上对应的区间出发重新计算一条轨道<sup>[11]</sup>, 轨道末点就是要插入的网格点的候选点。为了快速找到满足条件的轨道, 使用对分法在环  $C_i$  的对应段上进行搜索。找到满足距离条件的轨道末点后, 插入到  $C_{i+1}$  中。由于插值只在精度可以保证的  $C_i$  上进行, 而没有在精度无法预知的  $C_{i+1}$  上直接进行, 所以使用重新计算轨道的方法确定插值点的位置必然要比直接线性插值的精度要高。

曲率约束。经过距离检查后, 为了进一步提高精度, 在环  $C_{i+1}$  上对其相邻离散点采用曲率控制技术, 这一步相当于对网格点进行细调整, 防止漏掉流形细节。与第2)步类似, 顺次计算新环  $C_{i+1}$  中各相邻点所处线段所夹的角度, 并要求它们满足条件:

$$\begin{cases} \beta_k < \beta_{\max} \\ \Delta_k\beta_k < (\Delta\beta)_{\max} \end{cases} \quad (9)$$

若不满足上述条件, 说明相邻网格点之间的距离过大, 则采用上面所说的重新计算轨道的方法计算一条新的轨道曲线, 插入一个满足条件的网格点。注意这里我们只进行网格点的插入, 而不再进行删除, 这样有利于保证流形的质量。

4) 画出流形环并进行环间加网。首先将经过调整的  $C_{i+1}$  中的相邻的网格点用线段连起来, 然后用第2)步中计算所得的优化轨道将  $C_{i+1}$  中的离散网格点与它在  $C_i$  中对应的点连起来完成环间加网(在此基础上, 还可进行网内的三角剖分, 这样便于将流形表示成为一个面, 本文在仿真过程中没有加入这一步)。弧长  $ARC = ARC + \Delta$ , 判断弧长是否已经达到预设弧长  $L_{ARC}$ , 若  $ARC < L_{ARC}$ , 则将  $C_{i+1}$  作为新的当前环, 并转到2), 否则计算结束。

### 3 仿真实例

以洛伦兹系统为例。该系统最初由气象学家 E. N. Lorenz 在 1963 年提出,用来描述大气对流现象,它具有著名的“奇怪吸引子”和“蝶形吸引子”,而且有丰富的混沌行为,常常被当做例子来测试流形计算方法的性能。洛伦兹系统方程组为:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = \rho x - y - xz \\ \dot{z} = xy - \beta z \end{cases} \quad (10)$$

已知原点  $x_0 = (0, 0, 0)$  为双曲平衡点,系统在  $x_0$  处的雅可比矩阵  $A$ :

$$A = Df(x_0) = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - z & -1 & -x \\ y & x & -\beta \end{bmatrix}_{x_0} = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix} \quad (11)$$

矩阵  $A$  有实特征值  $-\beta, -\frac{\sigma+1}{2} \pm \frac{\sqrt{(\sigma+1)^2 + 4\sigma(\rho-1)}}{2}$ ,

将参数设置为  $\sigma = 10, \rho = 28, \beta = 8/3$ , 特征值约为  $-22.828, -2.667, 11.828$ 。所以该系统具有一维不稳定流形  $W^u$  和二维稳定流形  $W^s$ 。本次仿真所采用的精度参数为:  $\delta = 1, N = 20, \Delta = 2, \alpha_{\max} = 0.4, \alpha_{\min} = 0.3, (\Delta\alpha)_{\max} = 1, (\Delta\alpha)_{\min} = 0.1, \Delta_{\max} = 1, \Delta_{\min} = 0.25, \beta_{\max} = 1, (\Delta\beta)_{\max} = 2$ , 分  $L_{ARC} = 100$  和  $L_{ARC} = 120$  两种情况分别进行计算。

得到的二维稳定流形如图 2 所示。

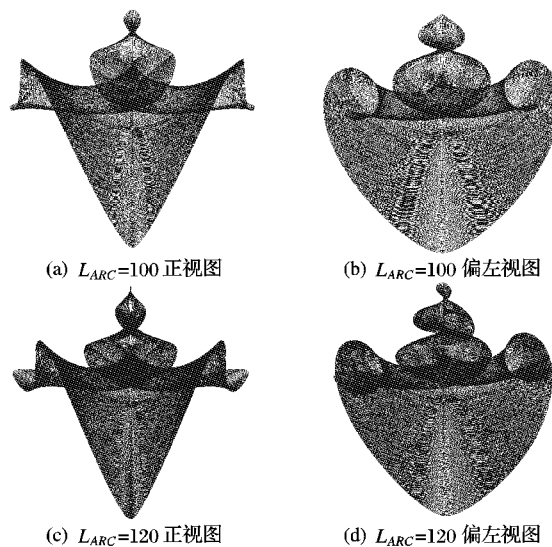


图 2 洛伦兹系统二维稳定流形

实验机配置为 Intel Celeron 1.73 GHz, 内存 1 GB。用 Matlab 编程计算, 图 2(a) 和 (b) 耗时仅为 3 min 10 s, 图 2(c) 和 (d) 耗时为 3 min 40 s。

本文算法与轨道弧长法<sup>[2]</sup>和测地线法<sup>[3]</sup>的对比分析见表 1。

表 1 几种算法对比分析

算法	轨线增长方式	轨道上点的分布	离散点插值方法	能否保证插值点精度	每一环的计算量
本文算法	求解初值问题	根据流形曲率优化分布	重新计算轨道法	能	$N \cdot T_{IVP}$
轨道弧长法	求解初值问题	平均分布	线性插值法	不能	$N \cdot T_{IVP}$
测地线法	求解边值问题	根据流形曲率优化分布	重新计算轨道法	能	$N^2 \cdot T_{IVP}$

注:  $N$  代表环上离散点的个数,  $T_{IVP}$  为计算一个初值问题的耗时。

因为边值问题通常使用初值问题的打靶法来求解, 所以每个边值问题的求解都相当于多个初值问题的求解, 因而其计算量近似为  $N^2 \cdot T_{IVP}$ 。

由表 1 可知, 本文提出的新算法继承了原有两种算法的优势, 既保证了计算的精度, 同时计算量也比较小。

### 4 结语

本文提出的二维流形计算方法兼顾了精度和计算速度两个方面的优势。使用等轨道弧长向外扩展来增长流形, 继承了轨道弧长法<sup>[2]</sup>计算量小、速度快的长处; 放宽了测地线法<sup>[3]</sup>等环间距离条件, 避开了边值问题的求解, 曲率控制条件的运用, 保持了测地线法精度高的优势; 由于轨道向外扩展过程中同时起到了环间加网的作用, 所以环间距离就不必像测地线法一样为了保证精度而迁就于环上流形变化最快的一点, 减少了重复计算。在对环上网格点进行调整时, 使用重新计算轨道的方法确定插值点的位置, 并首次综合使用了距离约束进行粗调整、曲率控制技术进行细检查两个步骤, 提高了插值点的精度。

另外, 在实际操作过程中, 可以将第 2) 和 3) 步结合起来并行向前计算, 即每计算得到新环上的一个点就马上检查它是否满足曲率约束条件, 这样更便于对网格点进行插入和删

除操作。

新算法能够适用于一般二维流形的计算, 但是要把该算法推广到高维流形的计算比较困难, 因为只有在二维情况下, 流形才能被看做是轨道的集合。高维流形的算法研究, 是作者今后的努力方向。

### 参考文献:

- [1] PALIS J, MELO W D. Geometric theory of dynamical systems [M]. Berlin: Springer, 1982.
- [2] JOHNSON M E, JOLLY M S, KEVREKIDIS I G. Two-dimensional invariant manifolds and global bifurcations: Some approximation and visualization studies [J]. Numerical Algorithms, 1997, 14: 125 - 140.
- [3] KRAUSKOPF B, OSINGA H M. Two-dimensional global manifolds of vector fields [J]. Chaos, 1999, 9(3): 768 - 774.
- [4] DOEDEL E J. AUTO, A program for the automatic bifurcation analysis of autonomous systems [J]. Congressus Numerantium, 1981, 30: 265 - 384.
- [5] GUCKENHEIMER J, VLADIMIRSKY A. A fast method for approximating invariant manifolds [J]. Journal on Applied Dynamical Systems, 2004, 3(3): 232 - 260.
- [6] HENDERSON M E. Computing invariant manifolds by integrating fattened trajectories [J]. Journal on Applied Dynamical Systems, 2005, 4(4): 832 - 882.

(下转第 2936 页)



的规则体的正文中出现一次,且该正文不是比较谓词。因此,形如“ $F1(X_1, \dots, X_n) :- \text{not } F2(X_1, \dots, X_n).$ ”的规则在 dlw 语言中是不安全的。

在 dlw<sup>[11]</sup>上运行该程序,可得两个4-皇后问题的解:

```
{hasqueen(3,1),hasqueen(1,2),hasqueen(4,3),hasqueen(2,4)}
{hasqueen(2,1),hasqueen(4,2),hasqueen(1,3),hasqueen(3,4)}
```

利用下面的命令:

```
C:\>dlw.mingw -silent -N=16 -pfilter=hasqueen nqueen.txt
```

运行该程序,dlw 返回 92 个 8-皇后问题的解,其中前 8 个解如下所示:

```
{hasqueen(3,1),hasqueen(7,2),hasqueen(2,3),hasqueen(8,4),hasqueen(5,5),hasqueen(1,6),hasqueen(4,7),hasqueen(6,8)}
{hasqueen(3,1),hasqueen(6,2),hasqueen(2,3),hasqueen(7,4),hasqueen(5,5),hasqueen(1,6),hasqueen(8,7),hasqueen(4,8)}
{hasqueen(6,1),hasqueen(3,2),hasqueen(1,3),hasqueen(7,4),hasqueen(5,5),hasqueen(8,6),hasqueen(2,7),hasqueen(4,8)}
{hasqueen(3,1),hasqueen(6,2),hasqueen(8,3),hasqueen(1,4),hasqueen(5,5),hasqueen(7,6),hasqueen(2,7),hasqueen(4,8)}
{hasqueen(7,1),hasqueen(3,2),hasqueen(8,3),hasqueen(2,4),hasqueen(5,5),hasqueen(1,6),hasqueen(6,7),hasqueen(4,8)}
{hasqueen(6,1),hasqueen(3,2),hasqueen(1,3),hasqueen(8,4),hasqueen(5,5),hasqueen(2,6),hasqueen(4,7),hasqueen(7,8)}
{hasqueen(4,1),hasqueen(7,2),hasqueen(1,3),hasqueen(8,4),hasqueen(5,5),hasqueen(2,6),hasqueen(6,7),hasqueen(3,8)}
{hasqueen(6,1),hasqueen(4,2),hasqueen(2,3),hasqueen(8,4),hasqueen(5,5),hasqueen(7,6),hasqueen(1,7),hasqueen(3,8)}
```

在上面的命令行中,“-N=max”规定了算术运算中遇到最大合法整数值,dlw 只能正确识别在该范围内的整数。对 N-皇后问题,max 应为 2N。

#### 4 结语

本文给出了一种把以谓词逻辑公式表示的约束型知识和定义型知识转化为 ASP 程序的方法。该方法允许首先利用约束型知识和定义型知识对待求解问题进行规格,然后利用现有的回答集求解器来求得问题的部分或者全部解。虽然本文采用的回答集求解器是 dlw,但只需对相关算法进行简单的修改便可生成能够在其他回答集求解器<sup>[12-13]</sup>上运行的逻辑程序。在构造以 ASP 为知识表示语言的非单调知识库的过程中,本文方法可作为一个把现有以谓词逻辑公式表示的知识自动转化为 ASP 规则和约束的工具。未来,将把本文方法应用于大规模工业应用领域,如机械装配序列规划问题的求解<sup>[14]</sup>,以测试本文方法的时空效率和其他性能;并利用该方法建立用于求解机械装配序列规划问题的专用 ASP 知识库<sup>[15]</sup>。

#### 参考文献:

- [1] ACOSTA GUADARRAMA J C. Maintaining knowledge bases at the object level [C]// MICAI 2007: the 2007 6th Mexican International Conference on Artificial Intelligence, Special Session. Washington, DC: IEEE Computer Society, 2008: 3-13.
- [2] SCHAUB T. Model-based knowledge representation and reasoning via answer set programming [M]. Berlin, Heidelberg: Springer-Verlag, 2008: 1-2.
- [3] GELFOND M, LIFSCHITZ V. Classic negation in logic programs and disjunctive databases [J]. Next Generation Computing, 1991, 9(3/4): 365-385.
- [4] GELFOND M, LIFSCHITZ V. The stable model semantics for logic programming [C]// Proceedings of the ICLP'88. Cambridge, MA: The MIT Press, 1988: 1070-1080.
- [5] BARAL C, GELFOND M. Logic programming and knowledge representation [J]. Journal of Logic Programming, 1994, 19: 73-148.
- [6] LIFSCHITZ V. Answer set programming and plan generation [J]. Artificial Intelligence, 2002, 138(1/2): 39-54.
- [7] NIEMELÄ I. Logic programs with stable model semantics as a constraint programming paradigm [J]. Annals of Mathematics and Artificial Intelligence, 1999, 25(3/4): 241-273.
- [8] YOU J, CARTWRIGHT R, LI M. Iterative belief revision in extended logic programs [J]. Theoretical Computer Science, 1996, 170(1/2): 383-406.
- [9] PETTOROSSO A, PROietti M. Perfect model checking via unfold/fold transformations [C]// Proceedings of the First International Conference on Computational Logic. Berlin: Springer-Verlag, 2000: 613-628.
- [10] LIN F, ZHAO Y. ASSAT: Computing answer sets of a logic program by SAT solvers [J]. Artificial Intelligence, 2004, 157(1/2): 115-137.
- [11] LEONE N, PFEIFER G, FABER W, et al. The DLV system for knowledge representation and reasoning [J]. ACM Transactions on Computational Logic, 2006, 7(3): 499-562.
- [12] LIN Z, ZHANG Y, HERNANDEZ H. Fast SAT-based answer set solver [C]// Proceedings of AAAI '06. Cambridge, MA: The MIT Press, 2006: 92-97.
- [13] GIUNCHIGLIA E, LIERLER Y, MARATEA M. Cmodels - 2: SAT-based answer set programming [EB/OL]. [2009-12-12]. <http://www.star.dist.unige.it/~enrico/ftp/04aaai.pdf>.
- [14] GOTTIPOLU R B, GHOSH K. A simplified and efficient representation for evaluation and selection of assembly sequences [J]. Computer in Industry, 2003, 50(3): 251-264.
- [15] ZHAO L Z, QIAN J Y, CHANG L, et al. Using ASP for knowledge management with user authorization [J]. Data and Knowledge Engineering, 2010, 69(8): 737-762.

(上接第2923页)

- [7] DELLNITZ M, HOBMAN A. A subdivision algorithm for the computation of unstable manifolds and global attractors [J]. Numerische Mathematik, 1997, 75: 293-317.
- [8] KRAUSKOPF B, OSINGA H M, DOEDEL E J, et al. A survey of methods for computing (un) stable manifolds of vector fields [J]. International Journal of Bifurcation and Chaos, 2005, 15(3): 763-791.
- [9] KRAUSKOPF B, OSINGA H M. Growing unstable manifolds of planar maps [EB/OL]. [2009-12-12]. <http://www.ima.umn.edu/preprints/OCT97/1517.pdf>.
- [10] KRAUSKOPF B, OSINGA H M. Computing geodesic level sets on

global (un) stable manifolds of vector fields [J]. Journal on Applied Dynamical Systems, 2003, 2(4): 546-569.

- [11] HOBSON D. An efficient method for computing invariant manifolds of planar maps [J]. Journal of Computational Physics, 1993, 104(1): 14-22.
- [12] KRAUSKOPF B, OSINGA H M. Growing 1D and quasi 2D unstable manifolds of maps [J]. Journal of Computational Physics, 1998, 146(1): 406-419.
- [13] GUCKENHEIMER J, WOLFOLK P. Dynamical systems: Some computational problems [M]// Bifurcations and Periodic Orbits of Vector Fields. Dordrecht, Netherlands: Kluwer Academic Publishers, 1993: 241-277.