文章编号：1000-3673（2021）06-2352-07 中图分类号：TM 721 文献标志码：A 学科代码：470·40

基于 Levenberg-Marquardt 算法的鞍结分岔点快速计算

林立亨[1]，董树锋[1]，唐坤杰[1]，毛航银[2]，宋永华[1,3]

(1. 浙江大学 电气工程学院，浙江省 杭州市 310027;
2. 国网浙江省电力公司，浙江省 杭州市 310007;
3. 智慧城市物联网国家重点实验室(澳门大学)，中国 澳门特别行政区 999078)

Fast Calculation of Saddle-node Bifurcation Point via Levenberg-Marquardt Algorithm

LIN Liheng[1], DONG Shufeng[1], TANG Kunjie[1], MAO Hangyin[2], SONG Yonghua[1,3]

(1. College of Electrical Engineering, Zhejiang University, Hangzhou 310027, Zhejiang Province, China;
2. State Grid Zhejiang Electric Power Company, Hangzhou 310007, Zhejiang Province, China;
3. State Key Laboratory of Internet of Things for Smart City (University of Macau), Macau SAR 999078, China)

ABSTRACT: In order to calculate static voltage stability margins quickly and accurately, this paper proposes two algorithms for searching saddle-node bifurcation points based on bisection search and parabolic approximation respectively. The Levenberg-Marquardt algorithm is applied to obtain the least-squares solution in the infeasible region of the power flow equations. The bisection search algorithm uses the least-squares solution to determine whether the current trial step is in the power flow infeasible region, and finally approaches the saddle-node bifurcation point. The parabolic approximation algorithm performs a parabolic approximation to 'the least-squares value' - 'load margin' curve in the infeasible region, and the zero point of the curve is corresponding to the required saddle-node bifurcation point. The numerical experiments on several classical cases show that compared with the traditional continuous power flow algorithm, the bisection search algorithm can significantly improve the computational efficiency while ensuring the accuracy. The parabolic approximation algorithm sacrifices some accuracy, but further improves the efficiency based on the bisection search algorithm. Also, thanks to the robustness of the Levenberg-Marquardt algorithm, these two algorithms in this paper can well converge under large ill-conditioned cases, ensuring the numerical stability of these two algorithms.

KEY WORDS: static voltage stability margin; Levenberg-Marquardt algorithm; saddle-node bifurcation point; load margin; bisection search; parabolic approximation

摘要：为了快速准确地计算静态电压稳定裕度，该文提出了 2 种鞍结分岔点快速求取算法，分别采用二分搜索和抛物线近似来进行计算。基于 Levenberg-Marquardt 算法在潮流方程的不可行域也能求得最小二乘解的特性，二分搜索算法利用解得的最小二乘值判断此算点是否处于潮流不可行域，通过二分搜索来快速逼近鞍结分岔点。抛物线近似算法对不可行域的最小二乘值−负荷裕度曲线进行抛物线近似，曲线的零点即为所求的鞍结分岔点。多个经典算例测试结果表明，相较于传统的连续潮流算法，二分搜索算法在保证计算准确地同时可以大幅度提升计算效率。而抛物线近似算法牺牲了一定的计算精度，在二分搜索算法的基础上进一步提升了效率。并且得益于 Levenberg-Marquardt 算法的强鲁棒性，2 种算法即使在面对大型病态算例时也可以收敛，保证了计算的稳定性。

关键词：静态电压稳定裕度；Levenberg-Marquardt 算法；鞍结分岔点；负荷裕度；二分搜索；抛物线近似

DOI：10.13335/j.1000-3673.pst.2020.0866

0 引言

电力系统在极端运行条件下有可能发生电压失稳现象，鞍结分岔点(saddle-node bifurcation，SNB)作为反应系统静态电压稳定性的一个重要指标，直观地反映了系统当前运行点到电压崩溃点之间的距离，可以帮助工作人员了解系统当前状态，避免电压崩溃现象的发生[1-3]。因此，快速准确地计算鞍结分岔点，一直是电压稳定领域的一个重要研究内容。

传统的鞍结分岔点计算方法主要包括以下 2 种：连续潮流法(continuous power flow，CPF)和直接法。

基金项目：国家电网公司科技项目：新能源电力系统随机稳定性分析研究(52110418000N)。

Project Supported by Science and Technology Project of State Grid Corporation of China: Analysis And Research on Stochastic Stability of New Energy Power System (52110418000N).

连续潮流法通过引入参数化方程，使扩展后的潮流方程维数等于未知数个数，同时避免了在接近鞍结分岔时潮流方程雅克比矩阵奇异的问题。随后通过预测-校正的方法逐步进行求解，直至获得所求的鞍结分岔点[4]。近年来，文献[5-8]从参数化策略、步长控制等方面对连续潮流进行了研究与改进。总的来说，连续潮流法可以比较方便地计及随负荷增长变压器分接头控制、节点无功越限等事件的发生顺序，计算得到的鞍结分岔点较为准确。但是计算量较大，耗费时间长。

直接法主要包括崩溃点法(point of collapse，PoC)和非线性规划法。崩溃点法利用鞍结分岔点处雅克比矩阵奇异的性质列写方程直接进行求解，相比连续潮流法显著降低了计算量[9-11]。但其方程维数是普通潮流方程的 2 倍，在面对大型电力系统时求解效率降低，且对于初值要求高[12]。非线性规划法通过建立非线性规划模型对鞍结分岔点进行求解[13-15]。但随着系统规模的扩大，约束方程的个数急剧增多，求解难度增加，限制了该方法的使用规模。

随着人工智能的兴起，也有许多学者基于人工智能对样本的训练，直接对鞍结分岔点进行预测[16-17]。此类方法速度快，但可能需要大量样本，调参过程也比较复杂。

除了上述方法外，近年来，还有一种从不可行域对鞍结分岔点进行求取的方法正在兴起[18-20]。文献[19]利用了 factored load flow(FLF)算法在潮流不可行域依然可以收敛到复数解的特性，在可行潮流和不可行潮流之间通过二分搜索来求取鞍结分岔点。文献[20]对[19]进行了改进，对 FLF 算法在不可行域求得的潮流方程复数解虚部进行抛物线近似，从而直接获得鞍结分岔点。相较于传统的连续潮流法，文献[19-20]的方法在大幅度提升计算效率的同时保证了计算精度，具有较高的应用价值。但其中所采用的 FLF 算法在鲁棒性上有所欠缺，面对病态程度较强的算例存在不收敛的情况。

因此，本文考虑使用鲁棒性更强的算法，从潮流不可行域对鞍结分岔点进行求取。Levenberg-Marquardt(L-M)算法在 20 世纪由 K.Levenberg 提出，并经 D.Marquardt 等学者发展完善，日益成熟。文献[21]证明了在迭代初始时，L-M 算法阻尼因子较大，具有初始下降量大、迭代迅速、鲁棒性的特点，减小了对初值的依赖性。在迭代末端，阻尼因子接近 0，则 L-M 算法具有牛顿法二阶收敛性，避免了最速下降法的锯齿形震荡。且通过引入自适应阻尼因子，L-M 算法可以在迭代过程中同时改变搜索方向与步长，较最优乘子法对病态潮流方程具有更好的适应性。文献[22-24]发现 L-M 算法可以扩大潮流方程的收敛范围，并且一定可以收敛到潮流方程的最小二乘解。文献[25]进一步对 L-M 算法进行了改进，提出了高阶的 L-M 方法。

鉴于以上分析，本文提出 2 种基于 L-M 算法的鞍结分岔点快速计算方法，从不可行域对鞍结分岔点进行求取。第 1 种采用二分法，在快速增加系统功率直至达到潮流不可行域后，根据 L-M 算法所获的最小二乘值，在可行域与不可行域之间进行二分搜索求得鞍结分岔点。通过多个算例验证，与连续潮流法相比，该方法在保证计算精度的同时大幅度提升了计算效率，鲁棒性强。第 2 种算法采用抛物线近似法，通过 L-M 算法计算 2 个不可行域潮流获得最小二乘值，对这两点进行抛物线近似获得鞍结分岔点。通过算例验证表明，该方法相比二分搜索算法，牺牲了一定计算精度的同时进一步提升了计算效率，可用于对实时性要求高的电压稳定在线实时评估与预防控制辅助决策在线计算中。

1　L-M 算法在潮流不可行域上的特性

将潮流方程写为

$$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0} \tag{1}$$

式中：$\boldsymbol{x}=[\boldsymbol{U},\boldsymbol{\theta}]^{\mathrm{T}}$ 为潮流方程中的状态变量，由电压幅值与相角组成。

L-M 算法在潮流方程求解中的应用已经成熟，具体求解过程详见附录 A[21]。L-M 算法求解潮流方程的本质为求解一个最小二乘模型：

$$\min G(\boldsymbol{x})=\frac{1}{2}\boldsymbol{F}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x}) \tag{2}$$

在潮流有解时，L-M 算法可以快速收敛到精确解，对应的最小二乘值 $G(\boldsymbol{x})=0$；在潮流无解时，L-M 算法也可以不断迭代至满足收敛条件(3)：

$$\boldsymbol{J}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x})<\varepsilon \tag{3}$$

式中：$\boldsymbol{J}(\boldsymbol{x})$为雅克比矩阵；$\varepsilon$ 为 L-M 算法预设的收敛精度。

此时算法收敛到一最小二乘解。以 IEEE14 节点系统为例，按(4)增大系统所有 PQ 节点负荷与发电机有功出力值直至超出鞍结分岔点($\lambda_{\max}$=4.06)，系统的最小二乘值 $G(\boldsymbol{x})$变化情况如图 1 所示。

$$\begin{cases} P_{\mathrm{L},i}(\lambda)=\lambda P_{\mathrm{L},i,0} \\ Q_{\mathrm{L},i}(\lambda)=\lambda Q_{\mathrm{L},i,0} \\ P_{\mathrm{G},i}(\lambda)=\lambda P_{\mathrm{G},i,0} \end{cases} \tag{4}$$

式中：λ为系统负荷参数，反应了系统功率水平；$P_{\mathrm{L},i}(\lambda)$、$Q_{\mathrm{L},i}(\lambda)$、$P_{\mathrm{G},i}(\lambda)$分别为系统中负荷有功功率、

负荷无功功率、发电机有功出力实际值；$P_{L,i,0}$、$Q_{L,i,0}$、$P_{G,i,0}$ 分别为系统中负荷有功功率、负荷无功功率、发电机有功出力初始值。

获得的 $G(x)$-λ曲线如图 1 所示(图中蓝色曲线)。在负荷参数$\lambda \le \lambda_{max}$ 即系统有解时，系统的最小二乘值 $G(x)=0$；在负荷参数$\lambda > \lambda_{max}$ 即系统无解后，系统的最小二乘值 $G(x)>0$ 并随着λ的增大单调递增。因此，如果可以从潮流的不可行域出发，快速求得 $G(x)=0$ 时对应的λ值，即是我们所需的鞍结分岔点[26]。

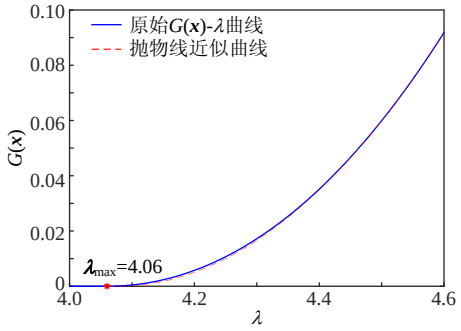


图 1 IEEE14 节点系统 $G(x)$-λ曲线

Fig. 1 $G(x)$-λ curve of the IEEE 14-bus system

2 二分搜索算法

二分搜索算法采用二分法来不断逼近 $G(x)=0$ 时对应的λ值，包括快速扫描和二分搜索两步。

1）快速扫描。

此步的目的是快速增长功率直至潮流的不可行域。具体的，可以预设一个较大的功率增长量$\Delta\lambda_1$。从基础负荷水平$\lambda_0=1$ 开始，每隔$\Delta\lambda_1$ 进行一次潮流计算直至计算得到 $G(x)>r$(r 为一足够小的值，当 $G(x)$大于该值可视为非零)，意味着此算点系统已经进入了潮流的不可行域。以 IEEE14 节点系统为例，预设$\Delta\lambda_1=0.5$，如图 2 所示，快速扫描步骤从$\lambda_0=1$ 开始，经过$\lambda=1.5, 2.0\ 2.5\cdots4.0, 4.5$ 的 7 次潮流计算，得到 $G(x)=0.0599$，说明此算点已经进入潮流不可行域，不再继续增加负荷。

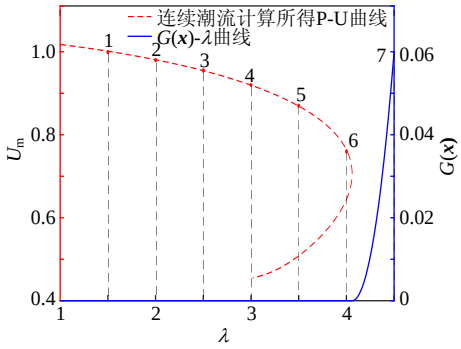


图 2 IEEE14 节点系统快速扫描步骤

Fig. 2 Fast scan step of the IEEE 14-bus system

相比传统的连续潮流算法，快速扫描步骤无需引入参数化方程和预测校正环节，只需要几步计算就能迅速达到潮流的不可行域，极大地提高了计算效率。

2）二分搜索。

上一步中最后一次计算得到的不可行点与倒数第二次计算得到的可行点一起构成了一个区间。对该区间不断进行二分搜索直至其宽度达到预设的精度ε_2，便可以求得所需的λ_{max}。具体的，如图 3 所示，IEEE14 节点系统中，对[4, 4.5]构成的区间进行二分，得到$\lambda=4.25$，利用 L-M 算法计算潮流，得到此算点对应的 $G(x)=0.0107$，意味着在此算点系统仍处于不可行域，于是将区间的右边界设为 4.25。下一步继续对[4, 4.25]进行二分搜索，如此类推，直至区间宽度达到预设的精度。理论上，只要预设的精度ε_2 足够小，二分搜索算法的计算结果将极为准确，且可达到任意精度。

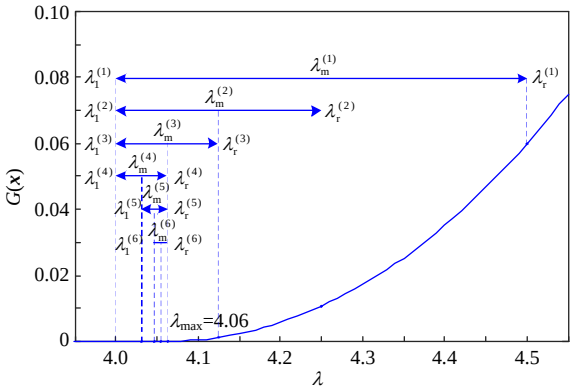


图 3 IEEE14 节点系统二分搜索步骤

Fig. 3 Bisection search step of the IEEE 14-bus system

另外，在快速扫描与二分搜索的迭代过程中，可以将上一步迭代中潮流方程的解设为下一步迭代中潮流方程的初值，可以加速方程求解，提高计算效率。

二分搜索算法的具体步骤如图 4 所示。

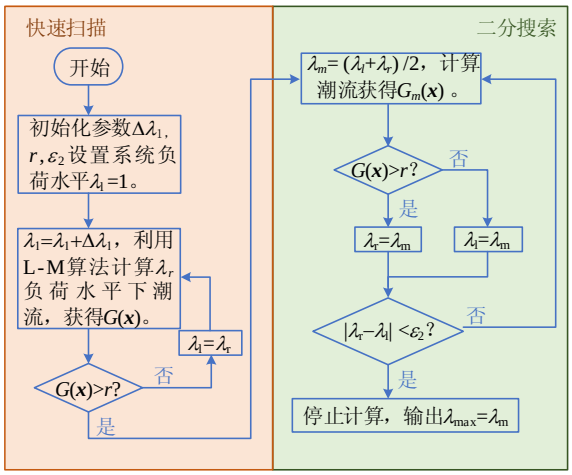


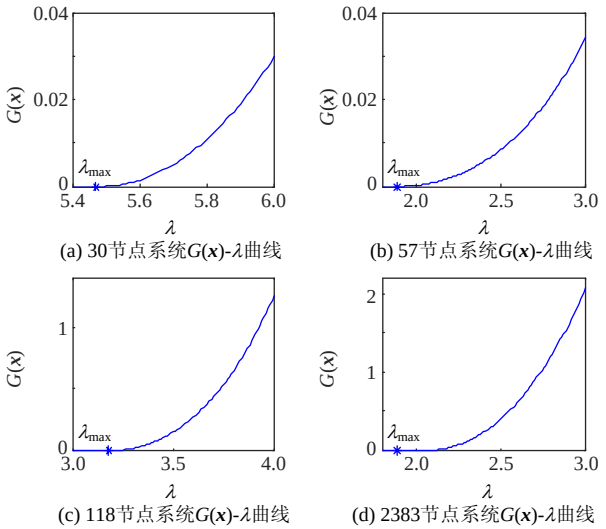
图 4 二分搜索算法流程图

Fig. 4 Bisection search algorithm flow chart

3　抛物线近似算法

二分搜索算法为了达到要求的精度，在步骤 2 中需要多次迭代，牺牲了计算效率。如果可以对不可行域的 $G(x)$-λ曲线进行多项式近似，则该多项式的零点即是所求的鞍结分岔点λ_{max}。相较二分搜索，通过求解多项式直接获得λ_{max}，可以进一步提高效率。

通过观察大量算例的 $G(x)$-λ曲线(如图 1 中蓝色曲线与图 5 所示)，发现均与抛物线非常接近。因此，本文考虑采用抛物线对不可行域的 $G(x)$-λ曲线进行近似，后续的算例分析也验证了抛物线近似在计算精度与速度上都取得了良好的效果。



(a) 30节点系统$G(x)$-λ曲线　(b) 57节点系统$G(x)$-λ曲线

(c) 118节点系统$G(x)$-λ曲线　(d) 2383节点系统$G(x)$-λ曲线

图 5　多个算例 $G(x)$ - λ曲线

Fig. 5　$G(x)$ - λ curves of multiple cases

抛物线方程可以表示为：$G(x)=a(\lambda\text{-}b)^2+c$。观察图 1 和图 5 中不可行域的 $G(x)$-λ曲线可以看出，该抛物线的顶点位于 x 轴上，因此可知 c=0。在 $G(x)$-λ曲线上取两点，通过式(5)便可以解出该抛物线的参数 a、b。

$$\begin{cases} a(\lambda_1-b)^2=G_1(x) \\ a(\lambda_2-b)^2=G_2(x) \end{cases} \tag{5}$$

式中：$(\lambda_1, G_1(x))$和$(\lambda_2, G_2(x))$为 $G(x)$-λ曲线上处于不可行域的两点，所解得的参数 b 即为所求的λ_{max}。

同样以 IEEE14 系统为例，取(4.5, 0.0599)和(4.51, 0.0628)代入式(5)中进行抛物线近似，得到 a=0.3351, b=4.0772，所得到的抛物线如图 1 中红色虚线所示。通过对比可以看出，经过近似得到的抛物线与原曲线非常相似，所得到的λ_{max}=4.0772 也与准确值比较接近，误差仅有 0.4%。因此，使用抛物线对 $G(x)$-λ曲线进行近似是可行的。

抛物线近似算法同样包括 2 步：快速扫描与抛物线近似。

1）快速扫描。

此步与二分搜索算法中第一步相似。但值得注意的是，部分大型算例的λ_{max} 值较小(如 MATPOWER 中的 case6468rte 算例λ_{max} = 1.1691)，如果预设的功率增长量$\Delta\lambda_1$ 过大，会导致用于抛物线近似的两点距离实际鞍结分岔点相对较远，降低了算法的精度。因此，可在前述快速扫描的基础上，增加一步对功率增长量的判断修正。如果在第一次增加$\Delta\lambda_1$后，系统已经进入了不可行域，说明此时取的$\Delta\lambda_1$过大，可将$\Delta\lambda_1$ 调整为原来的 1/m(m 为一缩小系数)，再重新进行快速扫描计算。

2）抛物线近似。

此步需要不可行域的 2 个点来进行抛物线近似。快速扫描步骤中的最后一个点已经进入了潮流不可行域，可用于进行抛物线近似，记作$(\lambda_1, G_1(x))$。再在负荷水平$\lambda_2=\lambda_1+\Delta\lambda_2$ 进行一次潮流计算，便可获得第 2 个点$(\lambda_2, G_2(x))$。此处的$\Delta\lambda_2$ 为一较小的功率增长量，可以让用于近似的两点离鞍结分岔点更近，提高抛物线近似的准确度。

通过抛物线近似，可以进一步提升鞍结分岔点的求解效率，适用于对计算速度要求高的电压稳定在线实时评估中。

抛物线近似算法的具体步骤如图 6 所示。

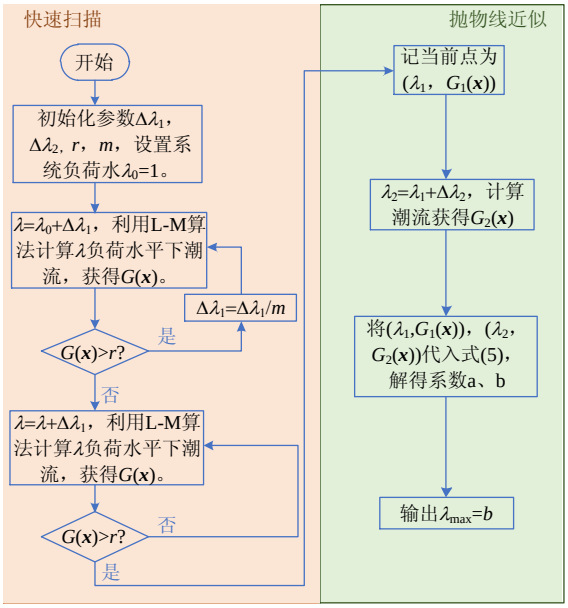


图 6　抛物线近似算法流程图

Fig. 6　Parabolic approximation algorithm flow chart

4　算例分析

本文通过 MATPOWER[27]中的多个经典算例对所提出的 2 个方法进行验证。参数设定方面，本文采用的 L-M 算法参数与文献[24]相同，最大迭代次数设为 40 次。二分搜索算法(以下记作 LM-bisection)中设置参数$\Delta\lambda_1$= 0.5[19]，$r=10^{-10}$，$\varepsilon_2=10^{-4}$。抛物

线近似算法(以下记作 LM-parabolic)中额外设置 $\Delta\lambda_2=0.01$，$m=10$。

本文提出的方法用 MATLAB 2018b 编程实现，运行在配置为 corei7-7700K 4.2 GHz CPU、32GB 内存的计算机上。

本文算法将与以下 4 种算法进行对比：

1）MATPOWER 自带的经典连续潮流算法[27](CPF)。

2）基于 FLF 的二分搜索算法[19](记作 FLF-bisection)。

3）基于 FLF 的抛物线近似算法[20](记作 FLF-parabolic)。

4）崩溃点法(PoC)，采用牛拉法求解，最大迭代次数设为 100 次。

本文在后续的计算验证中暂不考虑节点功率越限的情况，连续潮流算法采用 MATPOWER 中默认值启动，PoC 法以快速扫描步骤中进入不可行域的前一步潮流解作为初值，其余算法的初次潮流计算均采用平启动。

4.1 计算精度验证

CPF 算法作为求取鞍结分岔点的经典方法，其正确性已经得到了检验。本文将 CPF 算法求得的 λ_{max} 作为准确值，与 LM-bisection、LM-parabolic、FLF-bisection、FLF-parabolic 和 PoC 算法进行对比，计算结果如表 1 所示。

表 1 不同算法计算结果 λ_{max} 对比

Table 1 Comparison of calculation results λ_{max} of different algorithms

算例节点数	CPF	LM-bisection	LM-parabolic	FLF-bisection	FLF-parabolic	PoC
14	4.0603	4.0602 (0.002%)	4.0772 (0.416%)	4.0602 (0.002%)	4.0424 (0.439%)	4.0602 (0.002%)
30	5.4788	5.4788 (0.000%)	5.4789 (0.001%)	5.4789 (0.000%)	5.4793 (0.007%)	5.4788 (0.000%)
57	1.8921	1.8922 (0.003%)	1.8935 (0.075%)	1.8922 (0.003%)	1.9024 (0.546%)	发散
118	3.1871	3.1871 (0.000%)	3.2054 (0.574%)	$\lambda>\lambda_{max}$ 时不收敛	$\lambda>\lambda_{max}$ 时不收敛	3.1871 (0.000%)
2383	1.8937	1.8937 (0.001%)	1.8961 (0.124%)	1.8937 (0.001%)	1.9082 (0.767%)	100 次迭代后未收敛
6468	1.1691	1.1691 (0.001%)	1.1773 (0.700%)	$\lambda=1$ 时不收敛	$\lambda=1$ 时不收敛	100 次迭代后未收敛
9241	1.2432	1.2432 (0.002%)	1.2432 (0.002%)	$\lambda>\lambda_{max}$ 时不收敛	$\lambda>\lambda_{max}$ 时不收敛	100 次迭代后未收敛

表 1 括号中为不同算法计算结果的误差。从表 1 中可以看出，在计算精度方面，LM-bisection 算法获得的 λ_{max} 与 CPF 算法得到的准确值极为接近。理论上，LM-bisection 算法的计算精度取决于预设的精度 ε_2，若预设精度 ε_2 足够小，LM-bisection 算法可以达到任意精度要求。LM-parabolic 算法的计算结果同样与 CPF 算法获得的准确值非常接近，虽然准确度不及 LM-bisection 算法，但最大误差也不会超过 1%。并且相比于同样利用抛物线近似的 FLF-parabolic 算法，LM-parabolic 的计算结果也更为准确。这是因为 LM-parabolic 在进入不可行域后使用了一个小的功率增长量 $\Delta\lambda_2$，使用离 λ_{max} 更近的两点进行近似，进一步地提高了计算精度。PoC 方法虽然可以通过求解方程获得精确的鞍节分岔点，但由于其对初值的要求较高，容易出现不收敛的情况。

4.2 计算效率对比

此部分对不同算法之间的计算效率进行比较。由于 CPF 算法的计算速度受步长的影响较大，在此选用 0.01、0.05、0.1 三种步长进行对比。且为了比较的公平，CPF 算法在计算得到 λ_{max} 后即停止，不必计算出完整的 P-V 曲线。不同算法所耗费的时间如表 2 所示，表中单位为秒(s)。

表 2 不同算法计算时间对比

Table 2 Comparison of calculation time of different algorithms s

算例节点数	CPF 0.01	CPF 0.05	CPF 0.1	LM-bisection	LM-parabolic	FLF-bisection	FLF-parabolic	PoC
14	0.73	0.18	0.14	0.099	0.039	0.135	0.053	0.054
30	0.83	0.21	0.16	0.145	0.063	0.226	0.097	0.2983
57	0.47	0.14	0.14	0.121	0.027	0.195	0.060	发散
118	5.04	0.68	0.44	0.279	0.084	不收敛	不收敛	0.1943
2383	333	34.2	15.3	2.58	0.895	11.13	2.832	未收敛
6468	>10 min	484	191.5	9.68	5.28	不收敛	不收敛	未收敛
9241	>10 min	222	115.5	20.8	8.53	不收敛	不收敛	未收敛

从表 2 中可以看出，在计算效率上，LM-bisection 算法相较于步长为 0.01 与 0.05 的 CPF 算法有着显著提高。即使与步长为 0.1 的 CPF 算法相比，LM-bisection 算法同样有着显著优势。在算例规模小时，由于单次潮流计算时间过短，优势体现的不明显。随着算例规模的扩大，LM-bisection 算法的计算速度优势逐渐得以体现。在大型算例的计算中(例如 2383 节点及以上)，LM-bisection 相较于 CPF 算法节省了 80%~95%的计算时间。相较于 PoC 算法，在小算例中，LM-bisection 的计算速度与其相近，30 节点算例中 LM-bisection 计算效率更高，118 节点中 PoC 算法计算效率更高。原因在于 PoC 的计算速度受初值影响较大，30 节点系统中初值距离真实解相对 118 节点系统更远，因此需要更多次迭代才能收敛。

LM-parabolic 算法通过抛物线近似，相比 LM-bisection 算法进一步将计算效率提升了 2~3 倍。

即使是面对 9241 节点系统这样的大型算例，总共所用时间也不超过 10s。

而基于FLF的算法虽然在效率方面相比于CPF算法有所提高，但无论是 FLF-bisection 还是 FLF-parabolic 算法，其计算效率依然慢于对应的 LM-bisection 算法与 LM-parabolic 算法。

4.3　鲁棒性对比

通过表 1 和表 2 可以看出，得益于所用的 L-M 算法可以在迭代过程中同时改变搜索方向和步长，减小了对初值的依赖性，扩大了潮流方程的收敛范围[21]，LM-bisection 算法与 LM-parabolic 算法在计算中表现出强鲁棒性，即使在面对大型病态算例(如 6468 节点系统与 9241 节点系统)时依然能够收敛，保证了算法的稳定性。

而 FLF-bisection 算法与 FLF-parabolic 算法由于在 118 节点系统、6468 节点系统、9241 节点系统 3 个算例的计算中均出现了不收敛的情况。PoC 算法由于其方程维数高、受初值影响大等特点[9,11]，也在 57 节点系统、2383 节点系统等算例中出现了不收敛的情况。

5　结论

本文提出 2 种基于 L-M 算法的鞍结分岔点算法，分别采用了二分搜索与抛物线近似进行计算。算例分析表明：

1）在计算效率方面，本文提出的二分搜索算法相比连续潮流法极大的提高了计算效率，在 2383 节点及以上的大算例计算中可以节省 80%~95%的计算时间。而抛物线近似法在二分搜索算法的基础上又将计算效率进一步提升了 2~3 倍。

2）在计算精度方面，本文提出的二分搜索算法极为准确，抛物线近似算法为了计算效率一定程度上牺牲了计算精度，但误差也小于 1%。

3）在鲁棒性方面，本文算法得益于 L-M 算法的强鲁棒性，即使面对大型病态算例，也可以保证算法收敛。

本文所提的 2 个算法可以应用于不同场景，二分搜索算法可以应用于高精度要求场合(例如误差小于 0.1%)，而抛物线近似算法可以用于对计算速度要求高的电压稳定在线实时评估与预防控制辅助决策在线计算中。

在未来的研究中，将进一步考虑节点无功越限等限制因素，考虑如何利用本文算法对极限诱导分岔点进行求取。

附录见本刊网络版(http://www.dwjs.com.cn/CN/1000-3673/current.shtml)。

参考文献

[1] 万凯遥，姜彤．增补 P’Q 节点直接计算电压崩溃点的潮流方法[J]．中国电机工程学报，2018，38(12)：3507-3515．
WAN Kaiyao，JIANG Tong．A direct calculation method of voltage collapse points by supplement of a new type of P’Q bus[J]．Proceedings of the CSEE，2018，38(12)：3507-3515(in Chinese)．

[2] 范成围，陈刚，王晓茹，等．大规模电力系统静态电压安全评估[J]．电网技术，2017，41(7)：2263-2271．
FAN Chengwei，CHEN Gang，WANG Xiaoru，et al．Static voltage security assessment of large-scale power system[J]．Power System Technology，2017，41(7)：2263-2271(in Chinese)．

[3] 崔馨慧，贠志皓，刘道伟，等．大电网静态电压稳定在线防控灵敏度分析新方法[J]．电网技术，2020，44(1)：245-254．
CUI Xinhui，YUN Zhihao，LIU Daowei，et al．A new method of sensitivity analysis for static voltage stability online prevention and control of large power grids[J]．Power System Technology，2020，44(1)：245-254(in Chinese)．

[4] 赵晋泉，张伯明．连续潮流及其在电力系统静态稳定分析中的应用[J]．电力系统自动化，2005，29(11)：91-97．
ZHAO Jinquan，ZHANG Boming．Summarization of continuation power flow and its applications in static stability analysis of power system[J]．Automation of Electric Power System，2005，44(1)：91-97(in Chinese)．

[5] CHENG Xiangwen，CHEN Xiaoke，ZHAO Jinquan，et al．Probe into optimal combination of parameterization in continuous power flow[C]//Proceedings of the 2017 13th International Conference on Natural Computation，Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)．Guilin：IEEE，2017：2480-2484．

[6] 彭寒梅，曹一家，黄小庆，等．无平衡节点孤岛运行微电网的连续潮流计算[J]．中国电机工程学报，2016，36(8)：2057-2067．
PENG Hanmei，CAO Yijia，HUANG Xiaoqing，et al．Continuous power flow for islanding microgrid without balance nodes[J]．Proceedings of the CSEE，2016，36(8)：2057-2067(in Chinese)．

[7] 阮冲，刘金蝉，陈宝平，等．一种基于自适应步长控制与组合参数化的改进连续潮流计算方法[J]．电力自动化设备，2018，38(10)：184-190．
RUAN Chong，LIU Jinchan，CHEN Baoping，et al．An improved continuation power flow method based on adaptive controlled step-size and combination of parameterization[J]．Electric Power Automation Equipment，2018，38(10)：184-190(in Chinese)．

[8] PHICHAISAWAT S，SONG Y H，TAYLOR G A．Congestion management considering voltage security constraints[C]//Proceedings of International Conference on Power System Technology．Kunming：IEEE，2002：1819-1823．

[9] 江伟，王成山，余贻鑫，等．直接计算静态电压稳定临界点的新方法[J]．中国电机工程学报，2006，26(10)：1-6．
JIANG Wei，WANG Chengshan，YU Yixin，et al．A new method for direct calculating the critical point of static voltage stability[J]．Proceedings of the CSEE，2006，26(10)：1-6(in Chinese)．

[10] YAN Z，LIU Y，WU F，et al．Method for direct calculation of quadratic turning points[J]．IEE Proceedings-Generation，Transmission and Distribution，2004，151(1)：83-89．

[11] 吴浩，方鸽飞．电压稳定临界点的实用直接法[J]．电力系统及其自动化学报，1999，11(5)：18-24．
WU Hao，FANG Gefei．A practical direct method for voltage collapse point[J]．Proceedings of the CSU-EPSA，1999，11(5)：18-24(in Chinese)．

[12] 马冠雄，刘明波，王奇．一种识别静态电压稳定分岔点的混合方法[J]．电力系统自动化，2006，30(24)：17-20，43．
MA Guanxiong，LIU Mingbo，WANG Qi．A hybrid method for identifying bifurcation points in steady-state voltage stability analysis[J]．Automation of Electric Power Systems，2006，30(24)：17-20，43(in Chinese)．

[13] AYUEV B I，DAVYDOV V V，EROKHIN P M．Fast and reliable method of searching power system marginal states[J]．IEEE Transactions on Power Systems，2016，31(6)：4525-4533．

[14] 潘忠美，刘健，侯彤晖．计及相关性的含下垂控制型及间歇性电源的孤岛微电网电压稳定概率评估[J]．中国电机工程学报，2018，38(4)：1065-1074．
PAN Zhongmei，LIU Jian，HOU Tonghui．Probabilistic evaluation of voltage stability for islanded microgrids with droop-controlled and intermittent distributed generations considering correlation[J]．Proceedings of the CSEE，2018，38(4)：1065-1074(in Chinese)．

[15] 鲍海波，郭小璇．考虑新能源发电不确定性的静态电压稳定故障筛选与排序方法[J]．电力自动化设备，2019，39(7)：57-63．
BAO Haibo，GUO Xiaoxuan．Fault screening and ranking method of static voltage stability considering uncertainty of renewable energy power generation[J]．Electric Power Automation Equipment，2019，39(7)：57-63(in Chinese)．

[16] 唐滢淇，董树锋，朱承治，等．基于 Tri-Training-Lasso-BP 网络的静态电压稳定裕度在线预测方法[J]．中国电机工程学报，2020，4(12)：3824-3834．
TANG Yingqi，DONG Shufeng，ZHU Chengzhi，et al．Online prediction method of static voltage stability margin based on Tri-Training-Lasso-BP network[J]．Proceedings of the CSEE，2020，4(12)：3824-3834(in Chinese)．

[17] LI Shiyang，AJJARAPU V，DJUKANOVIC M．Adaptive online monitoring of voltage stability margin via local regression[J]．IEEE Transactions on Power Systems，2018，33(1)：701-713．

[18] HU Zechun，WANG Xifan．Efficient computation of maximum loading point by load flow method with optimal multiplier[J]．IEEE Transactions on Power Systems，2008，23(2)：804-806．

[19] GÓ MEZ-QUILES C，GÓ MEZ-EXPÓ SITO A，VARGAS W．Computation of maximum loading points via the factored load flow[J]．IEEE Transactions on Power Systems，2016，31(5)：4128-4134．

[20] GÓ MEZ-QUILES C，GÓ MEZ-EXPÓ SITO A．Fast determination of saddle-node bifurcations via parabolic approximations in the infeasible region[J]．IEEE Transactions on Power Systems，2017，32(5)：4153-4154．

[21] 严正，范翔，赵文恺，等．自适应 Levenberg-Marquardt 方法提高潮流计算收敛性[J]．中国电机工程学报，2015，35(8)：1909-1918．
YAN Zheng，FAN Xiang，ZHAO Wenkai，et al．Improving the convergence of power flow calculation by a self-adaptive Levenberg-Marquardt method[J]．Proceedings of the CSEE，2015，35(8)：1909-1918(in Chinese)．

[22] LAGECE P J．Power flow methods for improving convergence[C]//Proceedings of the 38th Annual Conference of IEEE Industrial Electronics．Montreal：IEEE，2012：1387-1392．

[23] 唐坤杰，董树锋，朱炳铨，等．大规模输配一体化系统牛顿法潮流计算性能分析及改进方法[J]．电力系统自动化，2019，43(6)：92-99．
TANG Kunjie，DONG Shufeng，ZHU Bingquan，et al．Performance analysis and improvement of newton method for power flow calculation of large-scale integrated transmission and distribution network[J]．Automation of Electric Power Systems，2019，43(6)：92-99(in Chinese)．

[24] TANG Kunjie，DONG Shufeng，SHEN Jie，et al．A robust and efficient two-stage algorithm for power flow calculation of large-scale systems[J]．IEEE Transactions on Power Systems，2019，34(6)：5012-5022．

[25] AMINI K，ROSTAMI F．Three-steps modified Levenberg-Marquardt method with a new line search for systems of nonlinear equations[J]．Journal of Computational and Applied Mathematics，2016，300：30-42．

[26] 程浩忠，利野直人．估算最接近电压稳定极限的简化方法[J]．电网技术，1996，20(4)：15-18．
CHENG Haozhong，YORINO N．A simplified method to evaluate closest voltage stability limits[J]．Power System Technology，1996，20(4)：15-18(in Chinese)．

[27] ZIMMERMAN R D，MURILLO-SANCHEZ C E．MATPOWER[Software][EB/OL]．2019．https://matpower.org．

林立亨

在线出版日期：2021-01-06。
收稿日期：2020-06-17。

作者简介：
林立亨(1995)，男，硕士研究生，研究方向为电力系统稳定性，E-mail：21810116@zju.edu.cn；
董树锋(1982)，男，通信作者，博士，副教授，研究方向为电力系统状态估计和有缘配电网分析，E-mail：dongshufeng@zju.edu.cn；
唐坤杰(1994)，男，博士研究生，研究方向为输配网协同优化分析与电力系统高性能计算方法，E-mail：tangkunjie1994@163.com。

（实习编辑　李健一）

附录 A

本文采用文献[22]中提出的自适应 L-M 算法对潮流方程式(1)进行求解，具体过程如下所示：

1）给定变量初值 $\boldsymbol{x}_1$，迭代次数 $k=1$，给定常数 m，$0<p_0<p_1<p_2<1$ 以及收敛精度 ε，设置 α_1 满足 $\alpha_1>m$。

2）根据式(A-1)计算阻尼因子 μ_k。

$$\mu_k=\alpha_k\|\boldsymbol{F}(\boldsymbol{x}_k)\|_2 \tag{A-1}$$

式中：α_k 为自适应因子。

3）根据式(A-2)计算当前迭代步 $\boldsymbol{d}_k$。

$$\boldsymbol{d}_k=-[\boldsymbol{J}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{J}(\boldsymbol{x}_k)+\mu_k\boldsymbol{I}]^{-1}\boldsymbol{J}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x}_k) \tag{A-2}$$

4）根据式(A-3)计算取舍指标 τ_k。

$$\tau_k=\frac{\|\boldsymbol{F}(\boldsymbol{x}_k)\|_2^2-\|\boldsymbol{F}(\boldsymbol{x}_k+\boldsymbol{d}_k)\|_2^2}{\|\boldsymbol{F}(\boldsymbol{x}_k)\|_2^2-\|\boldsymbol{F}(\boldsymbol{x}_k+\boldsymbol{J}_k\boldsymbol{d}_k)\|_2^2} \tag{A-3}$$

5）根据取舍指标 τ_k 选择是否接受 $\boldsymbol{d}_k$。

$$\boldsymbol{x}_{k+1}=\begin{cases}\boldsymbol{x}_k+\boldsymbol{d}_k, & \tau>p_0\\ \boldsymbol{x}_k, & \tau\le p_0\end{cases} \tag{A-4}$$

6）调整自适应因子 α_k：

$$\alpha_{k+1}=\begin{cases}10\alpha_k, & \tau_k<p_1\\ \alpha_k, & p_1\le\tau_k\le p_2\\ \max\{\alpha_k/10,\ m\}, & \tau_k>p_2\end{cases} \tag{A-5}$$

7）采用判据 $\boldsymbol{J}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x})<\varepsilon$ 来判别算法收敛与否，收敛则退出并输出结果，否则返回步骤 2）。