Locating Branch Points in Power Flow Problems using the Integral Approximation Method

Rastko Živanović[1,2]

[1]School of Engineering
University of Applied Sciences Upper Austria
Wels, Austria

[2]School of Electrical and Mechanical Engineering,
The University of Adelaide,
Australia

***Abstract*— In this paper, we present an efficient continuation method that can reconstruct solution branches and locate branch points in power flow problems. For this purpose, we propose the construction of an integral approximation using points from a solution branch interval obtained using conventional power flow software. This is in contrast to single-point Padé approximation methods, where the power flow problem must be embedded in a complex plane and therefore requires a new type of software. The paper presents simulation examples designed to explain how the proposed algorithm works, as well as to demonstrate the accuracy and computational efficiency of the method.**

Index Terms—analytic continuation, static voltage stability, power flow problem

I. INTRODUCTION

The system of polynomial equations is used to mathematically model steady-state conditions of power grids. Solution values of these equations are magnitude and phase angle of voltages on all network buses. This is the power flow problem, and it has multiple solutions [1]. If we vary some parameters of the model (e.g., the state of network loading), multiple solutions of bus voltage magnitudes move along algebraic curves (i.e., solution branches). Apart from stable solution branches, there are some branches that are practically infeasible because of constraints in system operation (e.g., low voltage branches) and others that represent a set of locally unstable points. It is possible, especially for heavy loaded grids, that different solution branches come close to each other. In this case, iterative Newton's method as well as the semidefinite relaxation method, might switch from stable branches to some other neighboring branches, as illustrated in [2]. Solution branches are terminated with singular points of branching type (branch points). It is of great practical value to determine location of branch points on the stable solution branches. These are points where voltage collapse phenomenon occurs, and distance to these points is used to determine a static voltage stability margin [3].

When approximating a solution branch with a rational function, it was noticed that poles and zeros lie on a branch cut and they are exponentially clustered at branch points. This is used to locate branch points [4]. The Padé method is commonly used to find poles and zeros of the rational approximation. In the Padé numerical method, a rational function is constructed from the approximation by Taylor series expansion at a point. The Taylor series coefficients are calculated by embedding the power flow problem in the complex plane. The first technique proposed in this framework is the holomorphic embedding load flow method (HELM) [5]. An alternative approach is based on discrete Fourier transform (DFT), where Newton iterations are used to solve the power flow problem for a varying complex parameter (e.g. load parameter) on a circular contour around a center point which is a point of approximation with Taylor series [6,7]. While HELM requires new coding, DFT-based method can reuse existing software by adding complex-parameter computing capabilities. In the case when the solution branches are rational functions, the Padé method is a tool for obtaining the exact analytical continuation and accurate location of branch points; but the power flow equations are quadratic, and thus the quadratic approximation is expected to give an improved continuation result, as discussed in our previous work [7].

In this paper, we investigate another possibility: the approximation of solution branches using values obtained by conventional power flow program (not embedded in the complex plane) for variation of load parameter within a certain interval, away from singularities. We will show that an analytic continuation based on such an approximation is possible and that the accurate location of branch points can be achieved. Staying within this framework, the paper will introduce the integral approximation method and compare its performance with "adaptive Antoulas-Anderson" (AAA) rational approximation [8] and DLog-transformed AAA rational approximation.

First, in section II we present the fundamental idea of integral approximation method. This approximation has a form of initial value solution of a linear inhomogeneous differential equation of order one with coefficients represented by polynomials. This is the closed-form solution expressed by using integrals. Singularities, including branch points, are roots of a polynomial coefficient next to the first derivative in the differential equation. In part B of section II, we explain greedy

Submitted to the 23rd Power Systems Computation Conference (PSCC 2024).

iteration algorithm that is used to construct integral approximation from data obtained through solving a sequence of power flow problems using a conventional software (MATPOWER in our case). Finally, in part C of section II, we show how to find roots of a polynomial coefficient next to the first derivative, and to obtain locations of branch points. Section III presets simulation examples that illustrate performance of the proposed method. We use a simple 2-bus power system to compare accuracy of the integral approximation with direct and DLog-transformed AAA rational approximation. A comparison with MATPOWER continuation power flow, using 9-bus network, reveals the potential for computation acceleration, which is especially important for large-size systems.

II. The Proposed Method

In this section, the integral approximation formula is devised as a solution of inhomogeneous linear differential equation for a specific initial value. Greedy-type iterative algorithm is used to calculate parameters of integral approximation formula using a small number of power flow problem solutions. Finally, we show how to get branch points from this approximation.

A. Integral Approximation

When the quadratic algebraic equation defines a node voltage v as an implicit function of the load parameter λ [7],

$$A(\lambda)v^2 + B(\lambda)v + C(\lambda) = 0\,, \tag{1}$$

then in [9] it is shown that the following inhomogeneous linear differential equation is a valid alternative to that implicit function:

$$P(\lambda)v' + Q(\lambda)v + R(\lambda) = 0\,. \tag{2}$$

The terms $A(\lambda)$, $B(\lambda)$, and $C(\lambda)$ in (1) are polynomials. When (1) and (2) both represent implicitly the same relation between v and λ, the following expression relates the polynomial $P(\lambda)$ in (2) to polynomials $A(\lambda)$, $B(\lambda)$, and $C(\lambda)$ in (1):

$$P(\lambda) = A(\lambda)D(\lambda), \tag{3}$$

where the polynomial

$$D(\lambda) = B(\lambda)^2 - 4A(\lambda)C(\lambda) \tag{4}$$

is the discriminant of the quadratic algebraic equation (1). This expression together with the expressions for $Q(\lambda)$ and $R(\lambda)$ in terms of $A(\lambda)$, $B(\lambda)$, and $C(\lambda)$ are derived previously in [9]. The real roots of $P(\lambda)$ correspond to the equation singularities including the branch points. The branch points are roots of (4).

The polynomials $P(\lambda)$, $Q(\lambda)$ and $R(\lambda)$ can be estimated by fitting the implicit equation (2) to data generated using a standard power flow software. The integral approximation is the solution of (2) for a specified initial value $v(\lambda_0) = v_0$ (usually $\lambda_0 = 0$ corresponds to the base load) [10],

$$\hat{v}(\lambda) = \exp\left[-\int_{\lambda_0}^{\lambda} \frac{Q(t)}{P(t)}\,dt\right]\left\{v_0 - \int_{\lambda_0}^{\lambda} \frac{R(t)}{P(t)} \exp\left[\int_{\lambda_0}^{t} \frac{Q(\mu)}{P(\mu)}\,d\mu\right] dt\right\}. \tag{5}$$

B. The Greedy Algorithm

The input data are a bus voltage values v_ℓ for a load parameter sequence λ_ℓ of length m, within a specified interval. These input values are obtained using a conventional power flow program without any numerical difficulties (interval chosen is not near singularity). For load parameter sequence λ_ℓ we use Chebyshev nodes and the barycentric Lagrange interpolation is applied to represent $v(\lambda)$ within the chosen interval [11]. The derivative v' is obtained directly from the barycentric Lagrange interpolant of v, as described in [11]. To summarize: the input data consists of three vectors

$\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_m]^T$ (m Chebyshev nodes),

$\mathbf{v} = [v_1 \dots v_m]^T$, and

$\mathbf{v}' = [v'_1 \dots v'_m]^T$.

The output data are n support points stored in the vector

$\mathbf{s} = [s_1 \dots s_n]^T$, where $s_1 = \lambda_1$,

and the weights α_i, β_i, γ_i, $i = 1,2,\dots,n$, (stored in the vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$) of the polynomials

$$P(\lambda_\ell) = \sum_{i=1}^{n} \alpha_i\, C_i(\lambda_\ell),$$
$$Q(\lambda_\ell) = \sum_{i=1}^{n} \beta_i\, C_i(\lambda_\ell),$$
$$R(\lambda_\ell) = \sum_{i=1}^{n} \gamma_i\, C_i(\lambda_\ell). \tag{6}$$

The basis functions of these polynomials are

$$C_i(\lambda_\ell) = C_{\ell,i} = 1/(\lambda_\ell - s_i), \tag{7}$$

and they are stored in the Cauchy matrix $\mathbf{C}$.

The greedy algorithm chooses support points one by one; a new support point s_i is selected that has the largest absolute approximation error, i.e., difference between the interpolant $v(\lambda)$ and the integral approximant $\hat{v}(\lambda)$ that is based on previously selected support points. We start these iterations with the support point $s_1 = \lambda_1$. Integral approximants are obtained via numerical solutions of (5). We use Clenshaw-Curtis quadrature: the FFT transforms data on Chebyshev nodes to coefficients of Chebyshev expansion and then the procedure integrates the expansion term by term [12]. Implementation of interpolation and integration algorithms are available in the Chebfun toolbox for MATLAB [12,13].

For each new support point chosen, the weights are recalculated by fitting (2). Initially selected, the first support point $s_1 = \lambda_1$ is excluded from the data used in fitting (2). The weights are calculated by solving a homogenous overdetermined system of equations:

$$[diag(\mathbf{v}')\mathbf{C} \;\; diag(\mathbf{v})\mathbf{C} \;\; \mathbf{C}\,] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = 0\,. \tag{8}$$

According to [14], the least squares solution of (8) is the last column of the matrix $\mathbf{V}$ obtained via the singular value decomposition: $[diag(\mathbf{v}')\mathbf{C} \;\; diag(\mathbf{v})\mathbf{C} \;\; \mathbf{C}\,] = \mathbf{U\Sigma V}^T$.

We stop the greedy iterations when the maximum difference between the bus voltage interpolant and its integral approximant ($max|v(\lambda) - \hat{v}(\lambda)|$) is below a specified threshold (e.g. 10^{-12}). Then the roots of $P(\lambda)$ are calculated and the branch points are obtained.

C. Branch Points

To find roots of $P(\lambda)$ and determine branch points we need to solve the following equation:

$$\textstyle\sum_{i=1}^{n} \alpha_i/(\lambda - s_i) = 0\,. \tag{9}$$

Here we use a method based on the generalized eigenvalue problem [15]. The roots of (9) are the eigenvalues of the following companion matrix pencil:

$$\left(\begin{bmatrix} 0 & -\mathbf{1}^T \\ \boldsymbol{\alpha} & \mathrm{diag}(\mathbf{s}) \end{bmatrix}, \begin{bmatrix} 0 & \\ & \mathbf{I} \end{bmatrix}\right), \tag{10}$$

where $\mathbf{1}$ is a vector of ones and $\mathbf{I}$ is an identity matrix.

We use Schur's complement to prove that the eigenvalues of (10) are solutions of (9). The characteristic equation of the pencil is

$$\det\left\{\lambda \begin{bmatrix} 0 & \\ & \mathbf{I} \end{bmatrix} - \begin{bmatrix} 0 & -\mathbf{1}^T \\ \boldsymbol{\alpha} & \mathrm{diag}(\mathbf{s}) \end{bmatrix}\right\} =$$

$$= \det\begin{bmatrix} 0 & \mathbf{1}^T \\ -\boldsymbol{\alpha} & \lambda\mathbf{I} - \mathrm{diag}(\mathbf{s}) \end{bmatrix} = 0\,. \tag{11}$$

Applying the $\mathbf{LDU}$ matrix block decomposition (using Schur's complement), (11) is modified as

$$\det\left\{\mathbf{L}\underbrace{\begin{bmatrix} 0 + \left(\lambda\mathbf{I} - \mathrm{diag}(\mathbf{s})\right)^{-1}\boldsymbol{\alpha} & \mathbf{0} \\ \mathbf{0} & \lambda\mathbf{I} - \mathrm{diag}(\mathbf{s}) \end{bmatrix}}_{\mathbf{D}}\mathbf{U}\right\} = 0 \tag{12}$$

The determinants of lower $\mathbf{L}$ and upper $\mathbf{U}$ triangular matrices in (12) are equal to 1, and the determinant of the block diagonal matrix $\mathbf{D}$ is the characteristic polynomial, and hence,

$$\det\left[\left(\lambda\mathbf{I} - \mathrm{diag}(\mathbf{s})\right)^{-1}\boldsymbol{\alpha}\right]\det[\lambda\mathbf{I} - \mathrm{diag}(\mathbf{s})] = 0, \tag{13}$$

or in different form, this is equivalent to (9).

The matrices in the companion matrix pencil (10) have dimension $(n+1)\times(n+1)$, whereas the degree of the polynomial $P(\lambda)$ is $n-1$. In this case we have two infinite eigenvalues that are rejected.

III. Examples

A. 2-bus Example

First, we use a simple 2-bus power flow example [7] to illustrate potential of the integral approximation method and to compare its performance with the AAA rational approximation [8] and DLog-transformed rational approximation. There are only two possible branches of real solution points (high- and low-voltage) in this problem, and the closed form solution is available. In this case, accuracy of locating branch points can be evaluated exactly.

In this example two buses are connected via a single line of impedance $Z = (0.001 + j0.1)pu$. Bus 1 is a PV-type bus and a reference bus where the voltage is kept at $1pu$. Power factor is specified for bus 2: $pf = 0.97$ lagging; and for active power p, voltage magnitude v of bus 2 is unknown. Equation relating bus 2 voltage magnitude and its active power is [7],

$$v^4 + [2\mathcal{R}e(\sigma)p - 1]v^2 + (|\sigma|p)^2 = 0\,, \tag{14}$$

where $\sigma = \left(1 \pm j\sqrt{1/pf^2 - 1}\right)Z^*$ (+ indicates lagging, and $-$ leading power factor). $\mathcal{R}e(\sigma)$ is the real part of σ and $|\sigma|$ is its absolute value. The branch point of interest in this example is $p^* = 3.8712$, and it is directly calculated from the equation formulated by equating discriminant of (14) with 0.

The barycentric Lagrange interpolation based on 13 Chebyshev nodes is used to represent pv curve within selected interval ($0 \le p \le 1$). We convert values at Chebyshev nodes to Chebyshev expansion coefficients using FFT [12]. Fig. 1 shows fast convergence of these coefficients. In Fig. 2 values of the interpolant at Chebyshev nodes in the selected interval are shown as red dots. In the same figure the support points calculated using the proposed greedy algorithm are denoted using the symbol 'o'. The initial support point is $s_1 = p_1 = 0$, and the additional two points found by the algorithm are: $s_2 = 0.6343$ and $s_3 = 0.9459$. The branch point found using the integral approximation method is $p^* = 3.8748$. It is shown in Fig. 2 using the symbol '$*$'. Error of the integral approximation result, in this example, is 3.7e-3.

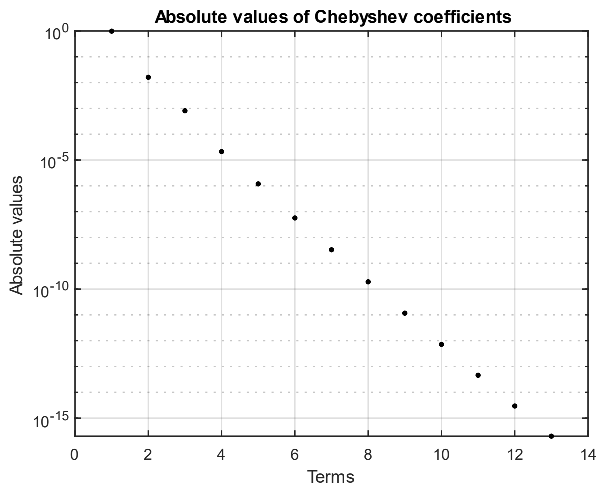

Figure 1. Absolute values of Chebyshev expansion coefficients; interpolation of $v(p)$ in 2-bus example

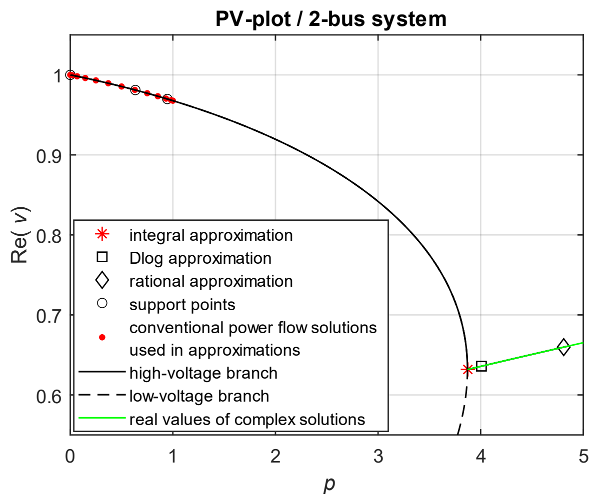

Figure 2. Locating a branch point in 2-bus example

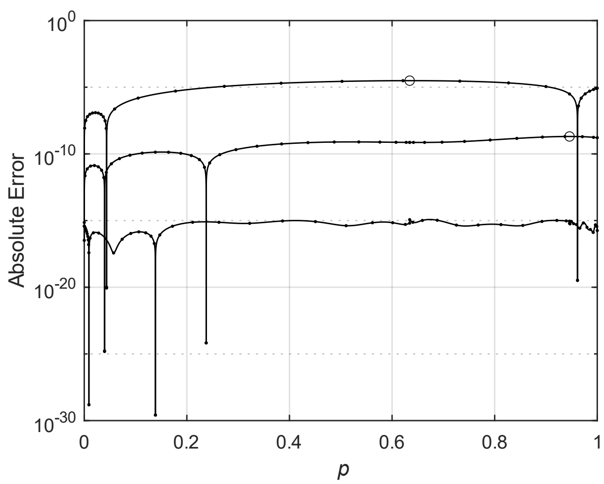

Figure 3. Accuracy achieved in the integral approximation of $v(p)$ in 2-bus example after each of the greedy iterations (accuracy curves are interpolants based on Chebyshev nodes shown as dots); support points are shown as 'o'

Fig. 3 illustrates how the approximation accuracy is improved through greedy iterations. Based on the initial support point $s_1 = p_1 = 0$, the first approximation is constructed and the error is plotted as a top line in Fig. 3. The largest error is indicated using the symbol 'o', and that point is a new support point s_2. New approximation error is the line in the middle and the largest error (symbol 'o') indicates another support point s_3. The bottom curve in Fig. 3 shows error of the integral approximation based on three support points. Iterations stop when the largest error is around 10^{-15} (below prescribed tolerance).

In Fig. 2 we also show results obtained using direct and DLog-transformed AAA rational approximation of the pv curve (interval $0 \le p \le 1$). Implementation of the AAA algorithm is available in the Chebfun toolbox [13]. The smallest real pole of direct rational approximation gives the estimated location of a branch point (diamond symbol in Fig. 2). Error of the rational approximation is 0.9346. In the DLog-transformed rational approximation we use again the AAA algorithm to approximate,

$$\frac{d}{dp}\log\big(v(p)\big) = \frac{Q(p)}{P(p)}. \tag{15}$$

This is equivalent to the representation based on the homogeneous differential equation,

$$P(p)v' - Q(p)v = 0\,. \tag{16}$$

Roots of the polynomial $P(p)$ are singularities including branch points. The branch point location estimated with DLog approach is shown in Fig. 2 using symbol '□'. Error of the DLog approximation is 0.1327. The polynomials $P(p)$ and $Q(p)$ have the form (6), and the greedy type algorithm is used to determine support points and weights in the AAA rational approximation [8]. Roots of $P(p)$ are calculated using the method explained in section II.C.

B. Comparison with MATPOWER Continuation Power Flow

In this section we compare the proposed method with the continuation power flow that is implemented in MATPOWER [16]. In this way, we can get an idea about the effectiveness of the proposed method. For that purpose, we show how many Newton iterations (solutions of linear system of equations) are required to solve the problem by either using the proposed method or continuation power flow algorithm. Solution of linear system in each iteration is by far the most time-consuming for large networks. Data for the new proposed method are generated using Newton iterations in MATPOWER. In both algorithms the tolerance for Newton iterations is 10^{-8}. The example we use here is the 9-bus test system. The base case data are provided in MATPOWER under the name 'case9'. This system has 3 generators, 3 loads and 9 branches. The load parameter λ is scaling up active and reactive power of loads and generation.

The following are relevant setting parameters in the MATPOWER continuation power flow:

- pseudo arc length parametrization [17],
- adaptive step size to balance between speed and robustness, minimum step size is 10^{-4} and maximum step size is 0.2, adapt step damping is 0.7,
- tolerance for nose point detection is 10^{-5},
- active and reactive power generator limits as well as bus voltage and branch MVA limits are not enforced.

To reach the branch point (nose point), continuation power flow requires 24 prediction-correction steps, as shown in Fig. 4 Voltage at bus 9 is critical. In total, 73 solutions of the linear system of equations are required:

24 solutions with augmented Jacobian matrix in prediction steps, and 49 solutions in correction steps.

The branch point found is $\lambda^* = 1.0942$.

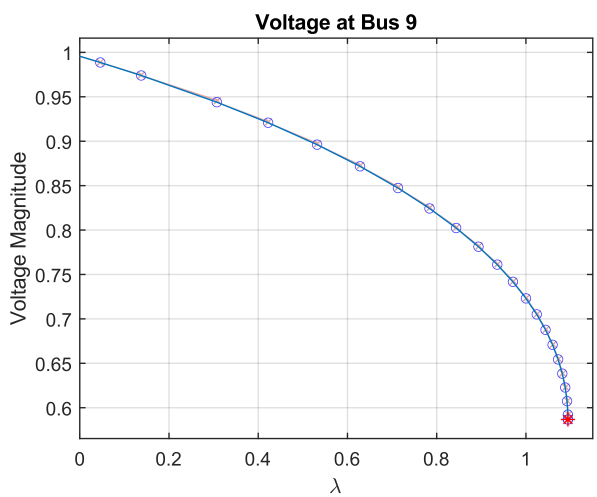

Figure 4. MATPOWER continuation power flow result; 'o' indicates steps and '∗' shows location of the branch point

The first step in the integral approximation algorithm is to interpolate λv curves within selected interval (in this example $0 \leq \lambda \leq 0.5$). Using 16 Chebyshev nodes we construct 6 interpolants for bus voltages (excluding PV buses) having the machine precision. To check the interpolation accuracy, we convert values at Chebyshev nodes to Chebyshev expansion coefficients [12]. The last coefficient is below 10^{-14} for all interpolations. Coefficients of the interpolant of bus-9 voltage can be seen in Fig. 5. Derivatives of voltages at all buses except generator buses are found, and bus 9 with the largest gradient is selected as the critical bus. In Fig. 6 values at Chebyshev nodes in the selected interval are shown as red dots. The greedy algorithm has found 2 support points in addition to the initial one at $s_1 = \lambda_1 = 0$. The support points are shown as 'o' in Fig. 6. The branch point found is $\lambda^* = 1.0957$, shown in Fig. 6 using the symbol '∗'. In this example, difference between continuation power flow result and integral approximation result is 1.5e-3.

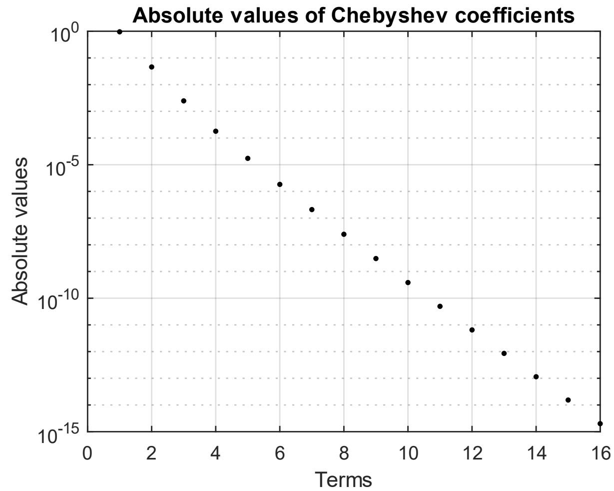

Figure 5. Absolute values of Chebyshev expansion coefficients (bus 9 interpolant)

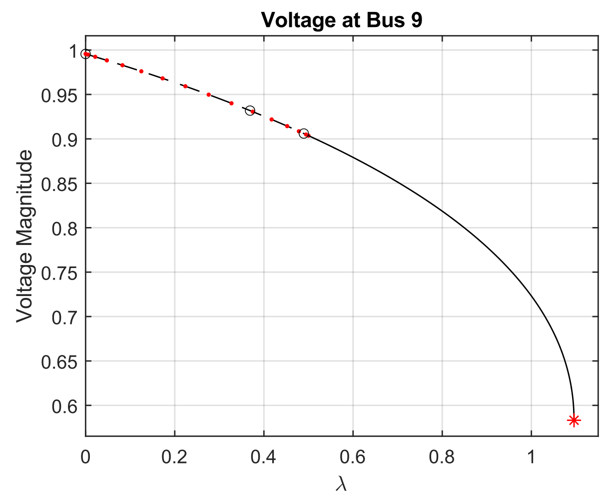

Figure 6. Locating a branch point in 9-bus example (symbol '∗'); red dots are values at Chebyshev nodes (interpolant), 'o' indicates support points.

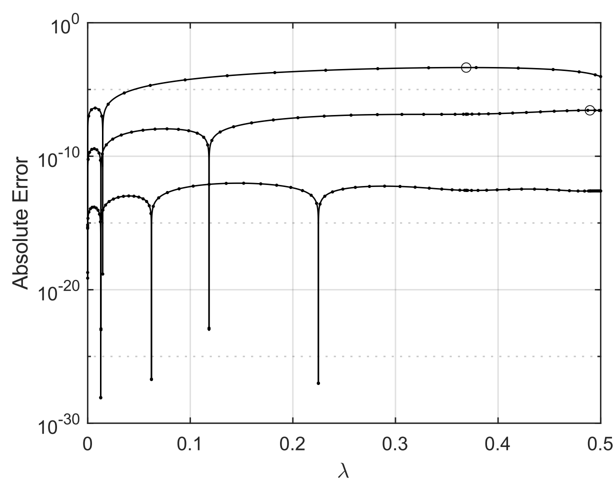

Figure 7. Accuracy achieved in the integral approximation of bus 9 voltage as a function of λ in the 9-bus example after each of the greedy iterations (accuracy curves are interpolants based on Chebyshev nodes shown as dots); support points are shown as 'o'

Fig. 7 illustrates how the approximation accuracy is improved through greedy iterations. Based on the initial support point $s_1 = \lambda_1 = 0$, the approximation is constructed and the error is plotted as a top line in Fig. 7. The largest error is indicated using 'o' and that point is a new support point s_2. New approximation error is the line in the middle and the largest error (symbol 'o') indicates another support point s_3. Finally, based on three support points we have integral approximation with error shown in the bottom curve. The largest error is below 10^{-12} (i.e. below prescribed tolerance) and we stop with iterations.

The integral approximation method requires 64 solutions of the linear system of equations (iterations) if at each Chebyshev node we start with the flat start and use 4 iterations to reach Newton iteration tolerance. This can be reduced to 49 iterations if prediction step is incorporated as in the continuation method:

> starting point requires 4 iterations, plus 15 prediction steps and 2 iterations in each node when starting with predicted values.

The presented example demonstrates potential of the integral approximation to improve computation speed in locating branch points compared to the classical continuation method. This is especially important for large size power systems.

In addition, it is possible to use the integral approximation to pass through any solution branch in one large step, i.e., to jump all the way near a branch point. At a branch point the differential equation (2) is singular and we should switch to the continuation power flow. This approach is similar to the hybrid method proposed in [18]. By using this strategy, we can find multiple power flow solutions.

IV. Conclusions

The proposed integral approximation algorithm can be a valuable tool for tracking solution branches of the power flow problems and locating singular points (i.e., branch points). The advantage of this technique compared to the Padé-type approximation is that it relies on a conventional power flow software (e.g., MATPOWER) and does not require a new software to solve the power flow problem embedded in the complex plane. Here we summarize the important contributions of the paper:

a) A novel iterative greedy-type algorithm is proposed for constructing integral approximations. Order of approximation is automatically determined. Interpolation, differentiation, and integration steps in this algorithm are implemented using the barycentric Lagrange interpolation method. High numerical robustness and accuracy are achieved thanks to this interpolation method, as well as due to the representation of the polynomials in (2) using the pole/residue form (6), (7).

b) Numerically robust calculation of polynomial roots (and locating branching points) is achieved using a new method based on the solution of the generalized eigenvalue problem.

c) Using simulation examples, we demonstrate the computational efficiency and accuracy of the proposed method in comparison with direct and DLog-transformed AAA rational approximations, as well as with MATPOWER continuation power flow software.

References

[1] W. Ma and J. Thorp, "An efficient algorithm to locate all the load flow solutions," *IEEE Trans. Power Systems*, vol. 8, no. 3, pp. 1077–1083, Aug. 1993.

[2] S.S. Baghsorkhi, S.P. Suetin, "Embedding AC power flow in the complex plane part II: A reliable framework for voltage collapse analysis," https://doi.org/10.48550/arXiv.1609.01211

[3] T Van Cutsem, C Vournas, *Voltage stability of electric power systems*, Springer Science & Business Media, 2007.

[4] S.S. Baghsorkhi, S.P. Suetin, "Embedding AC power flow in the complex plane part I: modelling and mathematical foundation, https://doi.org/10.48550/arXiv.1604.03425

[5] A. Trias, "The Holomorphic Embedding Load Flow method," *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 22-26, San Diego, California, July 2012.

[6] A. Sarnari, R. Živanović, Said Al-Sarawi, "Augmenting load flow software for reliable steady-state voltage stability studies," *International Trans. Electrical Energy Systems*, vol. 29, issue 9, Sept. 2019.

[7] R. Živanović, "Continuation via quadratic approximation to reconstruct solution branches and locate singularities in the power flow problem," in *Proc. 24th IEEE Mediterranean Conference on Control and Automation*, pp. 866-870, Jun 2016.

[8] Y. Nakatsukasa, O. Sète, and L.N. Trefethen, "The AAA algorithm for rational approximation," *SIAM Journal on* Scientific *Computing*, vol. 40 no. 3, pp. A1494-A1522, 2018.

[9] D.L. Hunter, G.A. Baker, Jr., "Methods of series analysis. III. Integral approximant methods," *Phys. Rev., Sect. B. Condens. Matter*, vol. 19, no. 7, April 1979.

[10] J. Goldberg, M.C. Potter, *Differential equations: a systems approach*, Prentice-Hall, 1998, p. 73.

[11] J.P. Berrut, L.N. Trefethen, " Barycentric Lagrange interpolation," *SIAM Review*, vol. 46, no. 3, pp. 501-517, 2004.

[12] Z. Battles, L.N. Trefethen, "An extension of Matlab to continuous functions and operators", *SIAM J. Sci. Comput*., vol. 25, no. 5, pp. 1743-1770, 2004.

[13] T. A. Driscoll, N. Hale, and L. N. Trefethen, editors, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.

[14] K. Inkilä, "Homogeneous least squares problem," *Photogrammetric Journal of Finland*, vol. 19, no. 2, pp. 34-42, 2005.

[15] A. Amiraslani, R. M. Corless, L. Gonzalez-Vega, and A. Shakoori, "Polynomial algebra by values," Tech. Rep. TR-04-01, Ontario Research Centre for Computer Algebra, January 2004.

[16] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," IEEE Trans. Power Systems, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[17] H. Mori and S. Yamada, "Continuation power flow with the nonlinear predictor of the Lagrange's polynomial interpolation formula," *Proceedings of the IEEE Transmission and Distribution Conference and Exhibition: Asia Pacific*, vol. 2, pp. 1133-1138, Oct. 2002.

[18] Wu Dan and Bin Wang, "Holomorphic embedding based continuation method for identifying multiple power flow solutions" *IEEE Access* vol. 7, pp. 86843-86853, 2019.