Matematicko-fyzikální fakulta
Univerzity Karlovy, Praha

Disertační práce

Kontinuace invariantních podprostorů

Ondřej Liberda

Školitel: Doc. RNDr. Vladimír Janovský, DrSc.
Obor: Vědecko-technické výpočty

e-mail: oliber@volny.cz

Prohlašuji, že jsem svou disertační práci vypracoval samostatně a výhradně s použitím citované literatury. Souhlasím se zapůjčováním práce.

V Praze 11.6.2006 Ondřej Liberda

Obsah

Předmluva

Řadu fyzikálních jevů lze modelovat pomocí dynamických systémů. Z matematického hlediska se jedná o soustavy obyčejných diferenciálních rovnic s parametrem,

$$u\dot{(}t) = G(u(t), \lambda), \qquad u(0, \lambda) = u_0(\lambda),$$

kde $t \mapsto u(t) \in \mathbb{R}^n$ je hledaná neznámá funkce (resp. vektor funkcí), $\frac{\mathrm{d}u(t)}{\mathrm{d}t}$ její derivace, $\lambda \in \mathbb{R}$ reálný parametr a $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ nelineární zobrazení. Nezávislá proměnná t má obvykle fyzikální význam času. V numerické praxi se často setkáváme s případem, kdy G vznikne prostorovou diskretizací parametrického systému parciálních diferenciálních rovnic.

Důležitou třídou problémů při studiu dynamických systémů je určení stability stacionárních (tj. v čase konstantních) řešení v závislosti na parametru λ a nalezení takových hodnot parametru λ, při nichž dochází ke změně stability stacionárních řešení.

O stabilitě příslušného stacionárního řešení pro danou hodnotu parametru λ lze rozhodnout na základě spektra linearizovaného operátoru G_u. Splňují-li všechna vlastní čísla linearizace podmínku $\Re(\mu) < 0$, je řešení stabilní. Existuje-li vlastní číslo μ_1 linearizace takové, že $\Re(\mu_1) > 0$, je řešení nestabilní. Není-li splněna ani jedna z obou výše uvedených podmínek (tj. pokud pro všechna vlastní čísla platí $\Re(\mu) \leq 0$ a pokud navíc existuje takové vlastní číslo μ_1, pro něž $\Re(\mu_1) = 0$), nelze o stabilitě příslušného řešení rozhodnout.

Naše pozornost se tedy bude soustředit na invariantní podprostory příslušející vlastním číslům z "pravé části" spektra a na algoritmy jejich numerického výpočtu. V praxi můžeme předpokládat, že dimenze invariantních podprostorů m bude vždy malá ve srovnání s celkovou dimenzí n. Metody pro výpočet invariantního prostoru lineárního operátoru jsou dobře známy. Naším cílem je ovšem výpočet invariantního prostoru pro celou třídu matic v závislosti na parametru λ, tj. *kontinuace invariantních podporostorů.*

Druhým cílem je detekovat změnu stability stacionárních řešení. Pro ten účel je zapotřebí zahrnout do invariantního podprostoru i vektory příslušné vlastním číslům, které jsou sice nalevo od imaginární osy, ale které v půběhu kontinuace tuto osu překročí. Součástí kontinuace invariantních podprostorů jsou proto algoritmy změny dimenze těchto podprostorů.

Disertační práce sleduje oba vytýčené cíle. V úvodní kapitole je formulována úloha kontinuace invariantních podprostorů. V kapitolách 2-4 následuje popis tří základních přístupů ke kontinuaci, jimiž jsou metody typu RPM (Recursive Projection Method), metody typu prediktor-korektor založené na řešení vroubené Sylvesterovy rovnice a metody typu prediktor-korektor založené na řešení Riccatiovy rovnice.

Disetační práce je souhrnem výsledků získaných během postgraduálního studia, které byly publikovány v odborné časopisecké literatuře. Součástí práce jsou proto přílohy, jejichž obsahem jsou právě zmiňované publikace.

Přínosem práce je srovnání všech tří typů metod kontinuace invariantních podprostorů. Původním výsledkem v oblasti metod typu RPM je modifikace základního algoritmu RPM, tzv. Projected RPM. Metodu Projected RPM jsme podrobili řadě numerických testů a provedli důkaz její lokální konvergence.

U metod typu prediktor-korektor založených na řešení vroubené Sylvesterovy rovnice jsme navrhli použití předpodmínění mocninných iterací pomocí Caylayovy transformace. Cílem bylo umožnit detekci bodů změny stability. Numerické experimenty potvrdily opodstatněnost tohoto kroku.

Pro metody typu prediktor-korektor založených na řešení Riccatiovy rovnice jsme ukázali těsnou souvislost s metodami založenými na řešení vroubené Sylvesterovy rovnice.

Kapitola 1

Úvod

Uvažujme čtvercovou matici $A \in \mathbb{R}^{n\times n}$. Sloupce matice $Z \in \mathbb{R}^{n\times m}$, $m < n$, generují bázi m-dimensionálního invariantního podprostoru, jsou-li splněny podmínky

$$AZ = Z\Lambda, \tag{1.1}$$

$$rank(Z) = m. \tag{1.2}$$

Matice $\Lambda \in \mathbb{R}^{m\times m}$ z definiční rovnice (1.1) představuje restrikci lineárního operátoru A na podprostor $span(Z)$. Mezi spektry obou matic platí vztah

$$\sigma(\Lambda) \subset \sigma(A). \tag{1.3}$$

Numerické metody výpočtu invariantních podprostorů čtvercové matice jsou popsány např. v [8].

V dalším textu se zaměřujeme na metody kontinuace invariantních podprostorů (dále uváděné pod zkratkou CIS z anglického Continuation of Invariant Subspaces). K danému zobrazení intervalu do prostoru čtvercových matic

$$s \mapsto A(s), \qquad s \in I \subset \mathbb{R}, \quad A \in \mathbb{R}^{n\times n}, \tag{1.4}$$

hledáme příslušná zobrazení

$$s \mapsto (Z(s), \Lambda(s))), \qquad s \in I \subset \mathbb{R}, \quad Z \in \mathbb{R}^{n\times m}, \quad \Lambda \in \mathbb{R}^{m\times m}, \tag{1.5}$$

kde matice $Z(s), \Lambda(s)$ splňují (1.1), (1.2).

Motivací ke studiu invariantních podprostorů je následující třída problémů: Uvažujme dynamický systém

$$\dot{u} = G(u, \lambda), \tag{1.6}$$

kde $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n\times n}$ je dostatečně hladké vektorové pole závislé na parametru λ. V aplikacích získáváme nelineární zobrazení G z prostorové diskretizace parciálních diferenciálních rovnic (např. metodou konečných prvků, konečných objemů nebo konečných diferencí). Při studiu vlastností dynamických systémů (1.6) hrají klíčovou roli stacionární řešení, tj. kořeny soustavy nelineárních rovnic

$$G(u, \lambda) = 0. \tag{1.7}$$

Genericky jsou stacionární řešení parametrizovatelná skalárním parametrem

$$s \mapsto (u(s), \lambda(s)), \qquad s \in I \subset \mathbb{R}, \quad (u(s), \lambda(s)) \in \mathbb{R}^n \times \mathbb{R}.$$

K výpočtu kořenů rovnice (1.7) používáme standardní kontinuační metody typu prediktor–korektor, viz [1], [12], které generují posloupnost bodů $\{s_i\} \subset I$ spolu s kořeny $(u_i, \lambda_i) \equiv (u(s_i), \lambda(s_i))$.

Položme $A(s) \equiv G_u(u(s), \lambda(s))$. Označme dále symbolem $n_-(s)$ počet vlastních čísel matice $A(s)$ (včetně násobnosti) se zápornou reálnou částí. Analogicky nechť $n_+(s)$ značí počet vlastních čísel s kladnou reálnou částí a $n_0(s)$ počet vlastních čísel s reálnou částí rovnou 0. Typicky je $n_-(s) \gg n_0(s) + n_+(s)$. V bodech nespojitosti funkce $s \mapsto n_+(s)$ může dojít ke změnám topologické klasifikace příslušných stacionárních řešení. Tyto body nazýváme *body změny stability*.

K nalezení bodů změny stability není zapotřebí znát celé spektrum matice $A(s)$. Zvolme $\gamma < 0$. Očíslujme vlastní čísla matice $A(s)$ tak, aby platilo

$$\Re\mu_1(s) \geq \cdots \geq \Re\mu_m(s) \geq \gamma > \Re\mu_{m+1}(s) \geq \cdots \geq \Re\mu_n(s). \tag{1.8}$$

Příslušné vlastní vektory (popř. zobecněné vlastní vektory) označme $v_1(s), \ldots, v_n(s)$. Prvních m vektorů $v_1(s), \ldots, v_m(s)$ tvoří bázi invariantního podprostoru, matice $Z(s) \equiv (v_1(s), \ldots, v_m(s))$ tedy splňuje rovnice (1.1), (1.2). Dimenze m invariantního podprostoru je malá v porovnání s celkovou dimenzí n.

Vlastní analýza změny stability je nyní redukována na nalezení spektra matice $\Lambda(s) \in \mathbb{R}^{m \times m}$. Hodnotu $n_+(s)$ získáme jako počet vlastních čísel matice $\Lambda(s)$ s kladnou reálnou částí. Obdobně obdržíme $n_0(s)$. Hodnotu $n_-(s)$ vypočteme ze vztahu $n_+(s) + n_0(s) + n_-(s) = n$.

Dimenze m ve vzorci (1.8) je závislá na parametru s, funkce $s \mapsto m(s)$ je po částech konstantní. V jejích bodech nespojitosti je genericky

$$\lim_{s \to s_0+} m(s) - \lim_{s \to s_0-} m(s) \in \{0, 1, -1, 2, -2\}, \quad s_0 \in I.$$

V dalších kapitolách se soustředíme na dva cíle :

1. Kontinuace m-dimensionálních invariantních podprostorů

2. Indikace bodů nespojitosti funkce $s \mapsto m(s)$

V literatuře můžeme nalézt tři varianty CIS ve spojení s analýzou změny stability stacionárních řešení dynamického systému (1.6) :

- Metoda rekurzivních projekcí [15]
- Metoda prediktor–korektor založená na řešení vroubené Sylvesterovy rovnice [2]
- Metoda prediktor–korektor založená na řešení Riccatiovy rovnice [5]

V kapitole (2) stručně popíšeme metodu rekurzivních projekcí (RPM), kterou lze použít též ke kontinuaci invariantních podprostorů, viz [15]. Numerické experimenty s původní RPM metodou nepřinesly uspokojivé výsledky, viz [13], [11]. V [10] je navržena metoda *Projected RPM*, která řeší některé nedostatky původní RPM metody. Analýza Projected RPM je provedena v [9].

Kapitola (3) pojednává o metodách založených na přímém řešení definiční rovnice invariantního podprostoru, jak bylo navrženo v [2]. Změnou dimenze m na základě indikace bodů nespojitosti funkce $s \mapsto m(s)$ se zabývá [14]. Indikátor nespojitosti je založen na mocninných iteracích s Cayleyovou transformací $\mathcal{C}(\mathcal{A})$, viz [6].

Kontinuace podle [5] založená na řešení Riccatiovy rovnice je popsána v kapitole (4), v [3] lze nalézt modifikaci této metody. V kapitole (4) rovněž poukážeme na vztah mezi metodami [2] a [5].

Kapitola 2

Metoda rekurzivních projekcí

Metoda rekurzivních projekcí (RPM, z anglického Recursive Projection Method) definovaná v [15] převádí (1.7) na problém nalezení pevného bodu nelineárního zobrazení závislého na parametru,

$$F(u, \lambda) = u. \tag{2.1}$$

Přirozeným způsobem, jak převést (1.7) na (2.1), je použití dynamické simulace

$$F(u, \lambda) = u \ + \varphi(\Delta t, u, \lambda), \tag{2.2}$$

kde $\varphi(t, u, \lambda)$ je tok dynamického systému (1.6). Aproximujeme-li tento tok pomocí metod numerické integrace soustavy obyčejných diferenciálních rovnic, získáme konkrétní předpis pro zobrazení F. V nejjednodušším případě Eulerovy metody obdržíme $F(u, \lambda) = u + \Delta t G(u, \lambda)$ pro pevně zvolené Δt. V [15] je dále popsána metoda pro kontinuaci křivky řešení Γ úlohy (2.1) : Pro $(u, \lambda) \in \Gamma$ a pro dané $\delta > 0$ očíslujme vlastní čísla matice $F_u(u, \lambda)$ tak, aby platilo

$$|\mu_1| \geq \cdots \geq |\mu_m| > 1 - \delta \geq |\mu_{m+1}| \geq \cdots \geq |\mu_n|.$$

Dále definujme prostor $\mathbb{P}$ jako invariantní podprostor příslušný vlastním číslům $\mu_1, \cdots, \mu_m$ a prostor $\mathbb{Q}$ buď jeho ortogonálním doplňkem, takže $\mathbb{R}^n = \mathbb{P} \oplus \mathbb{Q}$. Je-li konečně P ortogonální projekce F na podprostor $\mathbb{P}$ a Q ortogonální projekce F na podprostor $\mathbb{Q}$, je (2.1) ekvivalentní soustavě

$$\begin{aligned} p \equiv Pu &= PF(p+q, \lambda) && (2.3) \\ q \equiv Qu &= QF(p+q, \lambda). && (2.4) \end{aligned}$$

Lze ukázat, že zobrazení QF je kontraktivní a rovnici (2.4) lze tedy řešit metodou postupných aproximací. K řešení rovnice (2.3) na prostoru malé dimenze použijeme

Newtonovu metodu. Výsledná metoda je za jistých dodatečných předpokladů na zobrazení F lokálně konvergentní.

Důležitou součástí RPM je aproximace invariantního prostoru $\mathbb{P}$. V [15] jsou popsány dvě procedury :

1. Změna dimenze m. Zjistíme-li v průběhu kontinuace, že výše uvedená metoda nekonverguje, popř. konverguje pomalu, je potřeba zvýšit dimenzi prostoru $\mathbb{P}$. To se děje v případě, kdy některá vlastní čísla překročila hranici kruhu $\{z \in \mathbb{C}, |z| < 1 - \delta\}$ v komplexní rovině. Prostor $\mathbb{P}$ je pak nutno rozšířit o příslušné vlastní vektory. V generickém případě se jedná o jedno reálné vlastní číslo nebo o komplexně sdružený pár vlastních čísel. Aproximace příslušných vlastních vektorů lze přitom získat na základě diferencí $q^{(k+1)} - q^{(k)}$.

2. Korekce prostoru $\mathbb{P}$. V průběhu kontinuace je zapotřebí získávat v každém bodě křivky řešení Γ novou aproximaci prostoru $\mathbb{P}$. V [15] je za tímto účelem použita metoda mocninných iterací s maticí F_u aplikovaných na vektory báze podprodtoru $\mathbb{P}$.

V [10] poukazujeme na některé nedostatky původní metody RPM :

- Vztah mezi spektry $\sigma(G_u)$ a $\sigma(F_u)$ může být velice komplikovaný v případě použití složitějších metod aproximace toku φ. Tyto komplikace se projeví zejména u detekce Hopfových bifurkačních bodů.

- Projekce na podprostor $\mathbb{Q}$ se provádí až poté, co proběhla dynamická simulace na intervalu $\langle 0, \Delta t\rangle$. V případě $\Re\mu_1 \gg 0$ může díky konečné aritmetice dojít ke znehodnocení $\mathbb{Q}$-složky $\varphi(\Delta t, u, \lambda)$ i pro relativně malé hodnoty Δt.

- Metoda RPM nezohledňuje skutečnost, že v praktických příkladech jsou často k dispozici hodnoty diferenciálů G_u. Hodnoty F_u je naproti tomu nutno aproximovat, např. pomocí diferencí.

Navržena je proto modifikace algoritmu pod jménem *Projected RPM.* Metoda vychází z původní formulace problému (1.7). Uvažujme očíslování vlastních čísel matice G_u jako v (1.8). Dále postupujeme analogicky s [15] : Buď $\mathbb{P}$ invariantní podprostor příslušný vlastním číslům $\mu_1, \cdots, \mu_m$ a $\mathbb{Q}$ jeho ortogonální doplněk. P a Q buďte příslušné ortogonální projekce. Rovnice (1.7) je ekvivalentní systému rovnic

$$PG(p+q,\lambda) = 0 \tag{2.5}$$

$$QG(p+q,\lambda) = 0, \tag{2.6}$$

kde jsme stejně jako v (2.3) a (2.4) položili $p = Pu$ a $q = Qu$. Vlastními čísly linearizace zobrazení QG jsou právě čísla $\mu_{m+1}, \ldots, \mu_n$. Rovnici (2.6) můžeme proto s výhodou řešit dynamickou simulací systému

$$\dot{q}(t) = QG(p, q(t), \lambda), \tag{2.7}$$

tj. definujeme $(k+1)$-ní aproximaci pomocí k-té předpisem $q^{(k+1)} = \psi(\Delta t, q^{(k)})$, kde ψ je tok dynamického systému (2.7). Rovnici (2.5) řešíme opět pomocí Newtonovy metody. V [9] dokazujeme, že výše popsaný algoritmus je za jistých dodatečných předpokladů na zobrazení G lokálně konvergentní.

Podobně jako v původní RPM, i v její modifikaci potřebujeme znát podprostor $\mathbb{P}$. Narozdíl od původní RPM však nemůžeme použít metodu mocninných iterací. Ty totiž aproximují vlastní podprostor příslušný vlastním číslům s velkou absolutní hodnotou. My však potřebujeme, aby prostor $\mathbb{P}$ obsahoval vlastní vektory příslušné vlastním číslům z pravé části spektra, tj. vlastním číslům $\mu_1, \ldots, \mu_m$ z (1.8). Používáme proto metodu mocninných iterací s vhodnou funkcí matice $A \equiv G_u$. Touto funkcí je tzv. Cayleyva transformace, viz [6]. Výsledná metoda, tj. Projected RPM s použitím Cayleyvy transformace, představená v [9], umožňuje kontinuaci stacionárních řešení spolu s analýzou průchodů vlastních čísel imaginární osou. Cayleyva transformace je též zmiňována v následující kapitole v souvislosti s řešením vroubené Sylvesterovy rovnice.

Kapitola 3

Vroubená Sylvesterova rovnice

Definujme zobrazení $T : \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times m} \times \mathbb{R} \to \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times m}$

$$T(Z, \Lambda, s) = \begin{pmatrix} A(s)Z - Z\Lambda \\ S^{\mathrm{T}}Z - \mathrm{I}_m \end{pmatrix}, \tag{3.1}$$

kde $S \in \mathbb{R}^{n\times m}$ je pevná matice s plnou hodností. Hledáme kořeny rovnice $T(Z, \Lambda, s) = 0$. Předpokládejme, že tyto kořeny lze parametrizovat pomocí s, tj. existuje zobrazení $s \mapsto (Z(s), \Lambda(s))$, $s \in I \subset \mathbb{R}$ takové, že

$$T(Z(s), \Lambda(s), s) = 0. \tag{3.2}$$

V kapitole 3.1 je popsána metoda typu prediktor-korektor pro kontinuaci křivky řešení rovnice (3.1), založená na řešení vroubené Sylvestrovy rovnice (viz [8]) pomocí tzv. *Bartelsova–Stewartova* algoritmu. Jeho rozbor je obsahem kapitoly 3.2. V obou kapitolách přitom postupujeme podle [2].

Kontinuační algoritmus produkuje posloupnost bazí $Z(s) \in \mathbb{R}^{n\times m}$, které generují invariantní podprostory matic $A(s)$ s konstantní velikostí m. Je zřejmé, že je zapotřebí do báze $Z(s)$ přidávat, resp. z ní odebírat vektory tak, aby byla splněna podmínka (1.8). Touto problematikou se zabýváme v kapitole (3.3).

3.1 Kontinuace invariantních podprostorů

Derivováním rovnice (3.2) podle s v bodě $s_0 \in I$ získáme rovnice pro $\dot{Z} \equiv Z'(s_0)$, $\dot{\Lambda} \equiv \Lambda'(s_0)$:

$$\begin{pmatrix} A(s_0)\dot{Z} \;\; -\dot{Z}\,\Lambda_0 - Z_0\dot{\Lambda} \\ S^{\mathrm{T}}\dot{Z} \end{pmatrix} = \begin{pmatrix} -A'(s_0)Z_0 \\ 0 \end{pmatrix}, \tag{3.3}$$

kde $Z_0 \equiv Z(s_0)$ a $\Lambda_0 \equiv \Lambda(s_0)$. Vyřešíme-li (3.3), můžeme definovat prediktorový krok předpisem

$$Z^{(0)} = Z_0 + \dot{Z}\;\delta s, \quad \Lambda^{(0)} = \Lambda_0 + \dot{\Lambda}\delta s. \tag{3.4}$$

Alternativou k (3.4) je

$$Z^{(0)} = Z_0 + \dot{Z}\ \delta s, \quad Z^{(0)} = orth(Z^{(0)}), \quad \Lambda^{(0)} = (Z^{(0)})^{\mathrm{T}} A(s) Z^{(0)}. \tag{3.5}$$

Zápisem $orth(Z)$ rozumíme ortonormalizaci báze Z pomocí modifikovaného Gramm-Schmidtova ortonormalizačního procesu, viz např. [8]. Takto definované $\Lambda^{(0)}$ splňuje nutnou podmínku k tomu, aby mohlo být řešením rovnice (1.1).

Korektorový krok je založen na standardní Newtonově metodě aplikované na rovnici (3.2) pro $s \equiv s_0 + \delta s$. Aproximaci $(Z^{(k+1)}, \Lambda^{(k+1)})$ získáme ze $(Z^{(k)}, \Lambda^{(k)})$ řešením rovnice

$$\left(\begin{array}{c} A(s)Z^{(k+1)} - Z^{(k+1)}\Lambda^{(k)} - Z^{(k)}\Lambda^{(k+1)} \\ S^{\mathrm{T}} Z^{(k+1)} \end{array} \right) = \left(\begin{array}{c} -Z^{(k)}\Lambda^{(k)} \\ \mathrm{I}_m \end{array} \right). \tag{3.6}$$

Iterujeme přitom tak dlouho, dokud nedosáhneme požadované numerické přesnosti. Označme $(Z^{(last)}, \Lambda^{(last)})$ výsledek posledního kroku prediktoru.

3.2 Bartelsův–Stewartův algoritmus

Obě rovnice (3.3) i (3.6) lze zapsat ve tvaru

$$\begin{aligned} AH - H\Lambda - Z\Delta &= B && (3.7) \\ S^{\mathrm{T}} H &= C, && (3.8) \end{aligned}$$

což je tzv. vroubená Sylvesterova rovnice pro neznámé $H \in \mathbb{R}^{n\times m}$ a $\Delta \in \mathbb{R}^{m\times m}$. Pro její řešení lze s úspěchem použít Bartelsův-Stewartův algoritmus. Jeho hlavní myšlenka je založena na ortonormální transformaci matice Λ na horní trolúhelníkovou matici $\tilde{\Lambda}$,

$$\tilde{\Lambda} = Q^{\mathrm{T}} \Lambda Q, \quad Q^{\mathrm{H}} Q = \mathrm{I}_m. \tag{3.9}$$

Vzhledem k $m \ll n$ je tato netriviální transformace časově nenáročná (ve smyslu procesorového času). Pomocí Q definujeme transformace jednotlivých matic :

$$\tilde{B} = BQ, \quad \tilde{C} = CQ, \quad \tilde{H} = HQ, \quad \tilde{\Delta} = \Delta Q. \tag{3.10}$$

Konečně, položme $\lambda_j = \tilde{\Lambda}_{jj}$, $\tilde{H}_j$ buď j-tý sloupec matice $\tilde{H}$ a $\tilde{\Delta}_j$ buď j-tý sloupec matice $\tilde{\Delta}$. S použitím výše zavedených definic sestavíme soustavy lineárních rovnic :

$$\left(\begin{array}{cc} A \ - \lambda_j \mathrm{I}_n & -Z \\ S^{\mathrm{T}} & 0 \end{array} \right) \left(\begin{array}{c} \tilde{H}_j \\ \tilde{\Delta}_j \end{array} \right) = \left(\begin{array}{c} \tilde{B}_j + \sum_{k=1}^{j-1} \tilde{\Lambda}_{kj} \tilde{H}_k \\ \tilde{C}_j \end{array} \right) \tag{3.11}$$

pro neznámé $\tilde{H}_j$, $\tilde{\Delta}_j$ (s použitím běžné konvence $\sum_{k=1}^{0} \equiv 0$). Postupným řešením (3.11) pro $j = 1, \ldots, m$ obdržíme matice $\tilde{H}$, $\tilde{\Delta}$, potažmo hledané matice H, Λ.

Pro řešení soustav typu (3.11) se v [2] doporučuje použít algoritmus BEMW (Block Elimination Mixed for Wider-bordered systems), viz [7]. Matice soustavy je složena z bloků $A - \lambda_j \mathrm{I}_n$, $-Z$ a S^{T}. V typických aplikacích je prvním blokem řídká matice z $\mathbb{R}^{(n-m)\times(n-m)}$. Řídkost celkové matice je tedy „pokažena“ vroubícími maticemi s malým počtem řádek, resp. sloupců. Konstrukce algoritmu BEMW umožňuje použít pro řešení soustavy s (téměř singulární) maticí $A - \lambda_j \mathrm{I}_n$ libovolný „black box“ řešič. Výsledné stability metody BEMW je dosaženo vyrušením chyb v „téměř singulárních“ směrech. V [3] je jako alternativa navrženo použít pro řešení (3.11) metody uvedené v [17].

3.3 Modifikovaná Cayleyova transformace

Zjistíme-li na základě analýzy spektra $\sigma(\Lambda^{(last)})$, že některá vlastní čísla matice $\Lambda^{(last)}$ nesplňují podmínku $\Re\mu > \gamma$, odstraníme příslušné vektory z báze $Z^{(last)}$ a snížíme její dimenzi m. V generickém případě odebereme jeden nebo dva vektory. Naopak je nutno přidávat do báze $Z^{(last)}$ vlastní vektory příslušné vlastním číslům, která procházejí zleva doprava přímkou $\Re\mu = \gamma$ v komplexní rovině.

K detekci těchto průchodů používáme algoritmus, který nazýváme modifikovaná Cayleyova transformace. Položme pro jednoduchost zápisu $A = A(s)$, $Z = Z^{(last)}$. Buďte α_1, α_2 dvě reálná čísla, $\alpha_2 < \alpha_1$, $\alpha_1 \notin \sigma(A)$. Definujme

$$\mathcal{C}(A) = (A \ - \alpha_1 \mathrm{I})^{-1}(A - \alpha_2 \mathrm{I}). \tag{3.12}$$

Zobrazení $A \mapsto \mathcal{C}(A)$ se nazývá Cayleyova transformace, viz [6]. Toto zobrazení transformuje vlastní čísla $\Re\mu < \frac{1}{2}(\alpha_1 + \alpha_2)$ dovnitř a vlastní čísla $\Re\mu > \frac{1}{2}(\alpha_1 + \alpha_2)$ vně jednotkového kruhu v komplexní rovině.

Pro dané $\gamma < 0$ položme

$$\alpha_1 = \gamma + \omega, \quad \alpha_2 = \gamma - \omega, \qquad \omega > 0. \tag{3.13}$$

Zvolme počáteční bázi $Q^{(0)} \in \mathbb{R}^{n\times r}$ libovolně. Dimenze r buď malé přirozené číslo. Uvažujme následující iterační algoritmus pro $Q^{(k)} \in \mathbb{R}^{n\times r}$, který generuje posloupnosti bazí $V^{(k)}$, $Q^{(k)}$:

```
do
    $\mathtt{V}^{(\mathtt{k})} = \mathcal{C}(\mathtt{A})\mathtt{Q}^{(\mathtt{k}-1)}$
    $\mathtt{V}^{(\mathtt{k})} = (\mathtt{I} - \mathtt{ZZ}^{\mathtt{T}})\mathtt{V}^{(\mathtt{k})}$
    $\mathtt{Q}^{(\mathtt{k})} = \mathtt{orth}(\mathtt{V}^{(\mathtt{k})})$
    $\mathtt{k} = \mathtt{k} + 1$
    **until** (convergence or $\mathtt{k} = \mathtt{k_{max}}$),
```

tj. mocninné iterace předpodmíněné Cayleyovou transformací. Násobení maticí $\mathcal{C}(A)$ zleva představuje řešení soustavy lineárních rovnic s maticí $A - \alpha_1 I$. V každém kroku provádíme projekci pomocí zobrazení $\mathrm{I} - ZZ^{\mathrm{T}}$, čímž ořezáváme složky vektorů z $V^{(k)}$ obsažené v podprostoru $span(Z)$. Iterace $Q^{(k)}$ proto konvergují k bázi r-dimenzionálního podprostoru příslušného vlastním číslům $\mu_{m+1}, \ldots, \mu_{m+r}$.

Položme $Q = Q^{(k_{last})}$, $M = (Z, Q) \in \mathbb{R}^{n\times(m+r)}$. Pokračování algoritmu je následující :

```
    $\mathtt{H} = \mathtt{M^T A M}$

    **compute** $\sigma(\mathtt{H}) = \{\mu_1, \ldots, \mu_{\mathtt{m+r}}\}$

    **find** $\mathtt{z_j} : \mathtt{H z_j} = \mu_\mathtt{j} \mathtt{z_j}, \|\mathtt{z_j}\| = 1, \quad \mathtt{j} = 1, \ldots, \mathtt{m} + \mathtt{r}$

    $\mathtt{m_{new}} = 0$,, **empty** Z

    **for** $\mathtt{j} = 1, \ldots, \mathtt{m} + \mathtt{r}$ **do**

        **if** $\Re\mu_\mathtt{j} > \gamma$ **then**

            $\mathtt{Z_j} = \mathtt{M z_j}$

            $\mathtt{m_{new}} = \mathtt{m_{new}} + 1$

        **end if**

    **end for**

    $\mathtt{m} = \mathtt{m_{new}}$

    $\mathtt{Z} = \mathtt{orth(Z)}$,
```

kde Z_j značí j-tý sloupec nově konstruované báze Z. Výše uvedený algoritmus v kombinaci s kontinuací invariantních podprostorů označujeme jako CIS + ProjectedCayley, viz [14]. V [14] také demonstrujeme úspěšné použití CIS + Projected-Cayley v numerických testech detekce Hopfových bifurkačních bodů.

Kapitola 4

Riccatiova rovnice

Nechť je rovnice (1.1) splněna pro nějaká A, Z a Λ. Předpokládejme navíc $Z^{\mathrm{T}}Z = \mathrm{I}_m$. Buďte dále $X \in \mathbb{R}^{n\times m}$, $X^{\mathrm{T}}X = \mathrm{I}_m$ aproximace matice Z, $Y \in \mathbb{R}^{n\times(n-m)}$ ortogonální doplněk X, tj. $X^{\mathrm{T}}Y = 0$, $Y^{\mathrm{T}}Y = \mathrm{I}_{n-m}$. Zřejmě platí $XX^{\mathrm{T}} + YY^{\mathrm{T}} = \mathrm{I}_n$. Matici Z lze nyní vyjádřit jako součet

$$Z \;= X + YP, \tag{4.1}$$

kde $P \in \mathbb{R}^{(n-m)\times m}$. Vyjádřeme matici A v nových souřadnicích (X, Y). Definujeme tedy $L = (X, Y)^{\mathrm{T}}A(X, Y)$. Zapsáno blokově,

$$L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix}.$$

kde $L_{11} \in \mathbb{R}^{m\times m}$, $L_{12} \in \mathbb{R}^{m\times(n-m)}$, $L_{21} \in \mathbb{R}^{(n-m)\times m}$ a $L_{22} \in \mathbb{R}^{(n-m)\times(n-m)}$. Definujme operátor $\mathcal{R} : P \in \mathbb{R}^{(n-m)\times m} \mapsto \mathcal{R}(P) \in \mathbb{R}^{(n-m)\times m}$ předpisem

$$\mathcal{R}(P) \equiv L_{22}P - PL_{11} + L_{21} - PL_{12}P. \tag{4.2}$$

Následující tvrzení poukazuje na vztah mezi kořeny tzv. *Riccatiovy* rovnice

$$\mathcal{R}(P) = 0 \tag{4.3}$$

a kořeny rovnice (1.1):

Tvrzení. *Buďte $A \in \mathbb{R}^{n\times n}$, $X \in \mathbb{R}^{n\times m}$ a $Y \in \mathbb{R}^{n\times(n-m)}$ definovány jako výše, buď $P \in \mathbb{R}^{(n-m)\times m}$. Položíme-li dále $Z = X + YP$, jsou následující podmínky ekvivalentní :*

(i) $AZ - Z\Lambda = 0$

(ii) $\mathcal{R}(P) = 0$.

Přitom pro Λ *platí* $\Lambda = L_{11} + L_{12}$.

Důkaz. Nechť je splněna podmínka (i). Vynásobením této rovnice maticí X^{T} zleva obdržíme rovnost

$$X^{\mathrm{T}}A(X+YP) - X^{\mathrm{T}}(X+YP)\Lambda = 0.$$

Po roznásobení získáme vyjádření pro Λ:

$$\Lambda = L_{11} + L_{12}P.$$

Dosaďme tuto hodnotu zpět do (i) a vynásobme rovnici zleva maticí Y^{T}:

$$Y^{\mathrm{T}}A(X+YP) - Y^{\mathrm{T}}(X+YP)(L_{11}+L_{12}P) = 0.$$

Po úpravě dostaneme hledaný vztah (ii).

Předpokládejme na druhou stranu, že je splněna podmínka (ii). Z definice matice L a z identity $XX^{\mathrm{T}} + YY^{\mathrm{T}} = I_n$ plynou následující rovnosti:

$$AX = XL_{11} + YL_{21}, \quad AY = XL_{12} + YL_{22}. \tag{4.4}$$

S jejich použitím obdržíme

$$AZ = A(X+YP) = X(L_{11}+L_{12}P) + Y(L_{21}+L_{22}P).$$

Ve druhém sčítanci nahradíme výraz v závorce výrazem $PL_{11} + PL_{12}P$ na základě (ii) a obdržíme rovnost

$$AZ = Z(L_{11}+L_{12}P),$$

což je rovnice (i) pro $\Lambda = L_{11} + L_{12}P$. ♢

K řešení Riccatiovy rovnice použijeme Newtonovu metodu. Derivováním (4.2) získáme vztah pro diferenciál

$$D\mathcal{R}(P)\Delta = (L_{22} - PL_{12})\Delta - \Delta(L_{11} + L_{12}P).$$

Je-li $\{P^{(k)}\}_{k=0}^{\infty}$ posloupnost aproximací počínaje $P^{(0)} = 0$, získáme $(k+1)$-ní iteraci předpisem $P^{(k+1)} = P^{(k)} + \Delta$, kde $\Delta \in \mathbb{R}^{(n-m)\times m}$ je řešením Sylvesterovy rovnice

$$(L_{22} - P^{(k)}L_{12})\Delta - \Delta(L_{11} + L_{12}P^{(k)}) = -\mathcal{R}(P^{(k)}). \tag{4.5}$$

V kontextu kontinuace invariantních podprostorů lze použít výše uvedeného postupu jako korektorového kroku, viz [5]. Matice $L_{22} - P^{(k)}L_{12} \in \mathbb{R}^{(n-m)\times(n-m)}$ jsou obecně plné i v případě, kdy matice A je řídká. Použití této metody proto nemusí být vhodné pro úlohy velké dimenze. V následujícím tvrzení ukazujeme, že posloupnost $\{P^{(k)}\}_{k=0}^{\infty}$ má vzájemně jednoznačný vztah k posloupnostem $\{Z^{(k)}\}_{k=0}^{\infty}$, $\{\Lambda^{(k)}\}_{k=0}^{\infty}$, které jsou generovány na základě (3.7), (3.8). Vycházíme přitom z [3].

Oba přístupy jsou tedy zcela ekvivalentní a můžeme s výhodou aplikovat postupy popsané v kapitole (3) s využitím řídkosti matic $A(s)$.

Tvrzení. *Položme v (3.1)* $S = X = Z^{(0)}$. *Předpokládejme navíc* $P^{(0)} = 0$. *Mezi posloupností* $\{P^{(k)}\}_{k=0}^{\infty}$ *definovanou rovnicí (4.5) a posloupnostmi* $\{Z^{(k)}\}_{k=0}^{\infty}$, $\{\Lambda^{(k)}\}_{k=0}^{\infty}$ *definovanými rovnicí (3.6) platí vztahy*

$$\begin{aligned} Z^{(k)} &= X + YP^{(k)} && (4.6)\\ \Lambda^{(k)} &= X^{\mathrm{T}}AZ^{(k)}. && (4.7) \end{aligned}$$

Důkaz. Předně, rovnost (4.7) je přímým důsledkem rovnice (3.6). Skutečně, vynásobíme-li první řádek (3.6) zleva maticí X^{T} a využijeme-li druhého řádku (3.6) pro hodnoty $k, k+1$, obdržíme rovnost

$$X^{\mathrm{T}}AZ^{(k+1)} - \Lambda^{(k)} - \Lambda^{(k+1)} = -\Lambda^{(k)},$$

odkud již okamžitě plyne (4.7).

Násobíme-li v rovnici (3.6) zleva maticí Y^{T}, dostáváme

$$Y^{\mathrm{T}}AZ^{(k+1)} - Y^{\mathrm{T}}Z^{(k+1)}\Lambda^{(k)} - Y^{\mathrm{T}}Z^{(k)}\Lambda^{(k+1)} + Y^{\mathrm{T}}Z^{(k)}\Lambda^{(k)} = 0.$$

Po dosazení z rovnice (4.6) a úpravách dotaneme

$$(L_{22} - P^{(k)}L_{12})P^{(k+1)} - P^{(k+1)}(L_{11} + L_{12}P^{(k)}) = -L_{21} - P^{(k))}L_{12}P^{(k)}.$$

Ke stejnému výsledku ovšem vede i dosazení $\Delta = P^{(k+1)} - P^{(k)}$ a následné úpravy rovnice (4.5). Protože Sylvesterova rovnice (4.5) má jednoznačné řešení, je dokázán indukční krok $k \to k+1$. Pro hodnotu $k = 0$ tvrzení platí díky volbě $P^{(0)} = 0$, $X = Z^{(0)}$.

$\diamondsuit$

Dosažené výsledky, závěry a shrnutí

V předchozích kapitolách jsme podali přehled numerických metod pro kontinuaci invariantních podprostorů (CIS) a detekci bodů změny stability. Známé metody jsme klasifikovali do tří hlavních typů. Metody typu RPM jsou založeny na dynamické simulaci, tj. na numerické integraci relevantních dynamických systémů. Metody typu prediktor–korektor aplikované na definiční rovnici invariantního podprostoru vedou k řešení vroubené Sylvesterovy rovnice. Výhodou tohoto přístupu je zachování řídkosti matice soustavy. Třetí skupinu tvoří metody typu prediktor–korektor aplikované na Riccatiovu rovnici. Ukázali jsme, že z teoretického hlediska jsou oba přístupy ekvivalentní.

Na základě numerických experimentů s metodou RPM jsme navrhli její modifikaci, kterou jsme pojmenovali Projected RPM. Zatímco původní RPM provádí stabilizaci výsledku numerické integrace nestabilního dynamického systému, Projected RPM používá numerickou integraci k získání stacionárního řešení stabilizovaného dynamického systému. Výsledkem je metoda vyznačující se větší robustností. S výhodou také můžeme v konkrétních aplikacích využít znalosti diferenciálů G_u. K detekci bodů změny stability jsme použili mocninné iterace aplikované na Cayleyovu transformaci diferenciálu G_u.

Na Cayleyově transformaci je založena i detekce bodů změny stability u metod typu prediktor–korektor. Její použití nám umožnilo provádět změny dimenze sledovaného invariantního podprostoru tak, abychom včas zachytili vlastní čísla překračující imaginární osu.

Poděkování

Chci na tomto místě poděkovat Vladimíru Janovskému za vedení mého doktorského studia, za cenné rady a připomínky, které přispěly k dokončení disertační práce.

Můj dík patří také současným i bývalým zaměstnancům KNM MFF UK, zejména pak Petru Majerovi za cenné konzultace a podnětné diskuse.

Paní Plandorové z KNM a zaměstnankyním studijního oddělení pak děkuji za podporu při řešení administrativních úkonů po dobu mého doktorského studia.

Všem přátelům, blízkým a zejména rodičům děkuji za podporu a povzbuzení, které jsem při psaní disertační práce potřeboval a kterých se mně v hojné míře dostávalo.

Literatura

[1] E. L. Allgower and K. Georg, *Numerical continuation methods*, Springer Verlag, New York, 1990.

[2] W-J. Beyn, W. Kless and V. Thümmler, *Continuation of low-dimensional subspaces*, in Ergodic Theory, Analysis and Efficient Simulation of Dynamical Systems, B. Fiedler, ed, Springer Verlag, Berlin, 2001, pp. 47–73.

[3] J. Bošek, *Continuation of invariant subspaces in bifurcation problems*, PhD. thesis, Department of Mathmatics and Informatics, Phillips University, Marburg/Lahn, 2002.

[4] B. D. Davidson, *Large-scale continuation and numerical bifurcation for partial differential equations*, SIAM J. Numer. Anal., 34(5) (1997), pp. 2001–2027

[5] L. Dieci and M. J. Friedman, *Continuation of invariant subspaces*, Lin. Alg. Appl., 8 (2001), pp. 317–327.

[6] T. J. Garatt, G. Moore and A. Spence, *A generalized Cayley transform for the numerical detection of Hopf bifurcation points in large systems*, In Contributions in numerical mathematics, World Sci. Publ., 1993, pp. 177–195.

[7] W. Goaverts and J. D. Pryce, *Mixed block elimination for linear systems with wider borders*, IMA J. Numer. Anal. 13 (1993), pp. 469–483).

[8] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed, The Johns Hopkins University Press, Baltimore, 1996.

[9] V. Janovský and O. Liberda, *Continuation of invariant subspaces via Recursive Projection Method*, Applications of Mathematics 48 (2003), pp. 241–255.

[10] V. Janovský and O. Liberda, *Projected version of the RPM algorihtm*, in Proceedings of The 3^{rd} Scientific Colloqium, Institute of Chemical Technology, Prague, 2001, pp. 89–100.

[11] V. Janovský and O. Liberda, *Recursive Projection Method for detecting bifurcation points*, in Proceedings of SANM'99, Union of Czech Mathematicians and Physicists, 1999, pp. 121–134.

[12] M. Kubíček, M. Marek *Computational Methods in Bifurcation Theory and Dissipative Structures*, Springer, 1983

[13] O. Liberda *Numerical experiments with the Recursive Projection Method*, in Proceedings of Contributed Papers, WDS'99, pp. 84–89 in WDS'99 Proceedings of Contributed Papers, J. Šafránková, ed., matfyzpress, 1999, pp. 84–89.

[14] O. Liberda and V. Janovský, *Indication of a stability loss in the Continuation of Invariant subspaces*, Mathematics and Computers in Simulation 61 (2003), pp. 517–524.

[15] G. M. Shroff and H. B. Keller, *Stabilization of unstable procedures: the Recursive Projection Method*, SIAM J. Numer. Anal., 30 (1993), pp. 1099–1120.

[16] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Reviews., 15 (1973), pp. 727–763.

[17] P. Y. Yalamov and M. Paprzycki, *Stability and performance analysis of block elimination solver for bordered linear systems*, IMA J. Numer. Anal. 19 (1999), pp. 335–348.

Přílohovou část disertační práce tvoří publikované články k problematice kontinuace invariantních podprostorů.

Příloha A

Numerical experiments with the Recursive Projection Method

O. Liberda

Publikováno v *WDS'99 Proceedings of Contributed Papers, J. Šafránková, ed., matfyzpress, 1999, pp. 84–89.*

Příloha B

Recursive Projection Method for detecting bifurcation points

V. Janovský, O. Liberda

Publikováno v *Proceedings of SANM'99, Union of Czech Mathematicians and Physicists, 1999, pp. 121–134.*

Příloha C

Projected version of the Recursive Projection Method algorithm

V. Janovský, O. Liberda

Publikováno v *Proceedings of The* 3^rd^ *Scientific Colloqium, Institute of Chemical Technology, Prague (2001), pp. 89–100.*

Příloha D

Continuation of invariant subspaces via the Recursive Projection Method

V. Janovský, O. Liberda

Publikováno v *Applications of Mathematics 48 (2003), pp. 241–255.*

Příloha E

Indication of a stability loss in the Continuation of Invariant Subspaces

O. Liberda, V. Janovský

Publikováno v *Mathematics and Computers in Simulation 61 (2003), pp. 517–524.*

Numerical experiments with the Recursive Projection Method

Ondřej Liberda[1]

Department of Numerical Mathematic, Faculty of Mathematics and Physics, Charles University, Malostr. nám. 25, 118 00 Prague, Czech Republic

Abstract: A modification of the Recursive Projection Method for finding steady states of parameter dependent vector fields is introduced. Both original and modified procedures are compared.

1 Introduction

To find zeroes of the system of nonlinear equations

$$G(u, \lambda) = 0, \tag{1}$$

where $G : \mathcal{R}^N \times \mathcal{R} \to \mathcal{R}^N$ is a sufficiently smooth mapping dependent on the bifurcation parameter λ, we use path–following algorithms. Let us suppose for a moment that the solution path Γ,

$$\Gamma = \{(u, \lambda) \in \mathcal{R}^{N+1} : G(u, \lambda) = 0\},$$

can be parametrized with respect to the parameter $\lambda \in \mathcal{R}$. Given two points $(u_{-1}, \lambda_{-1}) \in \Gamma$ and $(u_0, \lambda_0) \in \Gamma$ on the solution path we perform two steps to find a new point on Γ:

1. secant predictor:
$$\lambda_1 = \lambda_0 + \delta\lambda,$$

[1] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

$$u^{(0)} = u_0 + \frac{\delta\lambda}{\lambda_0 - \lambda_{-1}}(u_0 - u_{-1}),$$

where $\delta\lambda$ is the step size. It can be changed during the continuation by means of adaptive methods.

2. corrector step: we try to find a point (u_1, λ_1) on Γ, i.e. we solve equation (1) for unknowns $u_1 \in \mathcal{R}^N$ while λ_1 remains fixed. Since $G(u, \lambda)$ is nonlinear we only find an approximation of (u_1, λ_1) by applying an iterative procedure on the system (1). These iterations are represented by a mapping $F : \mathcal{R}^N \times \mathcal{R} \to \mathcal{R}^N$, i.e. we have got a fixed-point scheme
$$u^{(k+1)} = F(u^{(k)}, \lambda_1). \tag{2}$$

We can use for example a Newton's method:

$$u^{(k+1)} = u^{(k)} - \left[G_u(u^{(k)}, \lambda_1)\right]^{-1} G(u^{(k)}, \lambda_1),$$

where G_u denotes the Jacobian matrix of G. The main disadvantage of the Newton's method is that we have no information about the spectrum of G_u which is essential for detecting of special points on the solution path such as folds, pitchfork bifurcations, Hopf bifurcations etc.

In the following we will consider the case when G is a spatial discretization of the right-hand side of the partial differential equation

$$\frac{\partial u}{\partial t} = \mathcal{G}(u, \lambda).$$

After the dicratization we obtain an autonomous system of ODEs:

$$\frac{\partial u}{\partial t} = G(u, \lambda). \tag{3}$$

We are mainly concerned about steady states of (3) which are precisely the solutions of (1). To find steady states we can use dynamical simulation. Let $\varphi(t, u^0)$ denote the flow of (3), i.e. the solution of (3) at time t with the initial condition $u(0) = u^0$. Then we can define mapping

$$F(u, \lambda) = \varphi(\Delta t, u), \tag{4}$$

where Δt is a time step. Of course we do not know the exact values of flow in practise. Instead a numerical integrator is employed to approximate the flow. However such a fixed-point procedure will fail to converge during the continuation due to the eigenvalues from spectrum $\sigma(G_u)$ crossing the imaginary axis.

2 Coupled itearations

It was suggested in [1] to use the following scheme:

Let us denote $\mu_1, \ldots, \mu_N$ the eigenvalues of the Jacobian $F_u(u, \lambda)$. Generally the iterations (4) will fail to converge or the convergence rate will be slow due to m eigenvalues $\mu_1, \ldots, \mu_m$ lying outside the disk

$$K_\delta = \{z \in \mathcal{C} : \|z\| \leq 1 - \delta\}$$

for some $0 < \delta < 1$. The main idea of coupled iterations is to decompose the state space:

$$\mathcal{R}^N = \mathcal{P} \oplus \mathcal{Q},$$

where $\mathcal{P}$ is the maximal invariant subspace of F_u belonging to the eigenvalues $\mu_1, \ldots, \mu_m$ and $\mathcal{Q}$ is the orthogonal complement of $\mathcal{P}$. Subspaces $\mathcal{P}$ and $\mathcal{Q}$ define orthogonal projectors P and Q respectively. Now we can replace the system

$$u = F(u, \lambda)$$

by couple of systems

$$\begin{aligned} p = f(p, q, \lambda) &= PF(p + q, \lambda), && (5) \\ q = g(p, q, \lambda) &= QF(p + q, \lambda), && (6) \end{aligned}$$

where $u = p + q$ is the unique decomposition of u with $p = Pu$ and $q = Qu$. It is shown in [1] that all the eigenvalues of

$$g_q(p, q, \lambda) = QF_u(p + q, \lambda)Q$$

lie in K_δ. Therefore we can perform Newton's iterations only on a small subpace $\mathcal{P}$ while continuing to use fixed-point iterations on $\mathcal{Q}$. The resulting scheme is then of the form:

$$\begin{aligned} [I - f_p(p^{(k)}, q^{(k)}, \lambda)](p^{(k+1)} - p^{(k)}) &= f(p^{(k)}, q^{(k)}, \lambda) - p^{(k)}, && (7) \\ q^{(k+1)} &= g(p^{(k)}, q^{(k)}, \lambda). && (8) \end{aligned}$$

2.1 Numerical realization of the RPM

To implement the scheme in practice we have to solve several problems:

1. Increasing the dimension of the unstable subspace $\mathcal{P}$ when the scheme is not convergent.

2. Updating the basis during the continuation.

3. Decreasing the dimension of the basis.

ad 1: We start the algorithm with $m = 0$. Generally only a single real eigenvalue μ_{m+1} or a pair of complex conjugate eigenvalues approach the boundary of the unit disk in complex plain. Hence one or two new vectors should be adjoined to the subspace $\mathcal{P}$. This can be done automatically during the continuation without evaluating the spectrum of F_u. Let us denote

$$\Delta q^{(k)} = q^{(k+1)} - q^{(k)}.$$

According to [1] the vector $\Delta q^{(k)}$ is approximately the k-th power iteration with matrix g_q applied to the vector $\Delta q^{(0)}$ provided F_u is Lipschitz continuous and the prediction $u^{(0)}$ is close enough to the exact solution $u_1 \equiv u_1(\lambda_1)$. This gives us possibility to compute new unstable directions from the Gram-Schmidt factorization of the two last difference vectors:

$$\left[\Delta q^{(k)}, \Delta q^{(k-1)}\right] = \hat{D}T,$$

where $T \in \mathcal{R}^{2\times 2}$ is upper triangular and $\hat{D} \in \mathcal{R}^{N\times 2}$ is orthogonal. Now we add one vector (the first column of $\hat{D}$) if $T_{11} \gg T_{22}$ or both columns of $\hat{D}$ otherwise.

ad 2: Let Z be the basis of the unstable subspace $\mathcal{P}$. Clearly Z depends on the parameter λ. This means that after each continuation step we should correct it by performing one or more steps of the so called subspace iterations:

$$Z(\lambda_1) = orth(F_u(u_1, \lambda_1)Z(\lambda_0)), \tag{9}$$

where "$orth$" denotes computing an orthonormal basis for $F_u Z$.

ad 3: After each successful continuation step we compute the eigenvalues of the matrix

$$H = Z^T F_u Z.$$

These should be the dominant eigenvalues of the Jacobian F_u. If only $\tilde{m} < m$ of them lie outside K_δ we decrease the dimension m of $\mathcal{P}$ by computing a real basis $V \in \mathcal{R}^{m\times\tilde{m}}$ for the dominant eigenvectors of H and replacing Z by $orth(ZV)$.

3 Modified RPM

For steady state problem (3) we suggested a slightly modified algorithm. Let

$$\alpha_k = \rho_k + i\tau_k, \quad k = 1, \ldots, N$$

be the eigenvalues of G_u and let us sort them in such a way that

$$\rho_1 \geq \ldots \geq \rho_m > 0 \geq \rho_{m+1} \geq \ldots \geq \rho_N.$$

If $m > 0$ then the steady state $u(\lambda)$ is not stable. Again we can decompose the state space

$$\mathcal{R}^N = \mathcal{P} \oplus \mathcal{Q},$$

where $\mathcal{P}$ is the invariant subspace corresponding to $\alpha_1, \ldots, \alpha_m$ and $\mathcal{Q}$ is the orthogonal complement of $\mathcal{P}$. Now let us decompose the system of ODEs:

$$\dot{q} = QG(p + q, \lambda), \tag{10}$$

$$\dot{p} = PG(p + q, \lambda). \tag{11}$$

The Jacobian of the right-hand side of (10) is QG_uQ. It can be shown that its eigenvalues are

$$0, \ldots, 0, \alpha_{m+1}, \ldots, \alpha_N. \tag{12}$$

There are m zeroes at the beginning of the list (12). Therefore we can use dynamical simulation for solving (10). The remaining equations (11) have to be solved with the help of Newton's method since the eigenvalues of the Jacobian of the right-hand side PG_uP are

$$\alpha_1, \ldots, \alpha_m, 0, \ldots, 0.$$

The difference between RPM and our method is that we use Newton's method on the original problem (1). Another difference is that we compute the flow of dynamical system (10) with projected right-hand side while in the original RPM we first evaluate the flow and then we project it by Q to the stable subspace $\mathcal{Q}$.

3.1 Numerical realization of the modified RPM

To implement the modified RPM algorithm we have to solve the same problems that we have mentioned in the section (2.1).

ad 1: As for increasing the dimension of $\mathcal{P}$ we can use the same procedure with Gram-Schmidt factorization of

$$\left[\Delta q^{(k)}, \Delta q^{(k+1)}\right]$$

since the eigenvectors of G_u are identical with those of F_u (even though it is not true for the eigenvalues).

ad 2: It seems to be the most difficult problem to maintain the accuracy of the basis Z. We could use the subspace iterations (9) but it would require computing $F(u, \lambda)$, i.e. we would have to use a time integrator on the original system (3). Unfortunately it is precisely one of the things we wanted to avoid by modifying the original RPM algorithm. However the numerical experiments indicate that it is not necessary to update subspace $\mathcal{P}$ after each continuation step. We decided only to increase the basis size when the process was not convergent and to decrease the basis size from time to time (once per 20 continuation steps).

ad 3: Analogically to the original version of RPM we compute the matrix

$$\hat{H} = Z^T G_u Z$$

and test its spectrum (this is not expensive as the dimension of $\hat{H}$ is m). Only the eigenvectors corresponding to the eigenvalues α_k with $Re\,\alpha_k > -\hat{\delta}$ are taken to the new basis of the unstable subspace $\mathcal{P}$. Here $\hat{\delta}$ is a positive constant.

3.2 Advanteges of the modified RPM

1. We use a time integration procedure (such as Euler's method, Runge-Kutta methods, implicit stiff solvers etc.) on a dynamical system (10). The Jacobian of its right-hand side has no eigenvalues with positive real parts. Therefore the time integration process should be more stable even for larger time steps and in the cases when we are not very close to the solution path.

2. We perform Newton's steps on the original problem (1). Hence we can get information about bifurcation points by testing the spectrum of the matrix $\hat{H}$.

3.3 Numerical algorithm

Let Z be an orthonormal basis in $\mathcal{P}$. Then the projectors P and Q are of the form

$$P = ZZ^T, \quad Q = I - ZZ^T.$$

In order to perform Newton's steps in $\mathcal{R}^m$ we introduce new variable $z \in \mathcal{R}^m$:

$$z = Z^T p = Z^T u, \quad p = Zz \qquad \text{and} \qquad u = Zz + q.$$

Moreover to enable continuation past folds we add a constraint on variables z and λ. The resulting system in $\mathcal{R}^m$ is then

$$\begin{aligned} Z^T G(Zz + q, \lambda) &= 0, \\ (z - z^{(0)})^T (z^{(0)} - z_0) + (\lambda - \lambda^{(0)})(\lambda^{(0)} - \lambda_0) &= 0, \end{aligned}$$

where $z_0 = Z^T u_0$ and $z^{(0)} = Z^T z^{(0)}$. The modified RPM algorithm is listed bellow:

Modified RPM Corrector

```
    $\tau := (z^{(0)} - z_0,\ \lambda^{(0)} - \lambda_0)^T$;
    while ($\|G(u)\| + constraint > tol$);
        $M := \begin{pmatrix} Z^T G_u Z & Z^T G_\lambda \\ & \tau^T \end{pmatrix}$;
        $constraint := (z - z^{(0)},\ \lambda - \lambda^{(0)})^T \tau$;
        $b := \begin{pmatrix} Z^T G \\ constraint \end{pmatrix}$;
        $\begin{pmatrix} z \\ \lambda \end{pmatrix} := \begin{pmatrix} z \\ \lambda \end{pmatrix} - M^{-1} b$;
        $q :=$ ODESolver $(QG, q, \lambda, \Delta t)$;
        $u := Zz + q$;
        if (not convergence) then $increase\_basis\_size(Z)$;
    endwhile;
    $decrease\_basis\_size(Z)$;
    $subspace\_iteration(Z)$;
end;
```

4 Numerical results

Both examples bellow are the same sample problems as in [1]. We used Runge-Kutta fourth order procedure with steplength adaptation to compute the flow. The local tolerance was chosen $5 \cdot 10^{-5}$. All the derivatives were replaced by differences with steplength $\epsilon = 10^{-5}$. Paremeter *tol* from algorithm above was set to $5 \cdot 10^{-3}$. The length of the predictor step h was held constant during the continuation with $h = 0.15$. In the first example we examined steady states of one-dimensional Bratu problem:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + \lambda e^u \\ u(0) &= u(1) = 0.\end{aligned}$$

We used standard continuous piecewise linear finite elements on a grid with 40 points to discretize the differential operator. Figure 1 shows computed solution path. In the figure 2 we can see solution path of the non-symmetric problem arising from the spatial discretization of the following couple of PDEs:

$$\frac{\partial u_1}{\partial t} = \frac{1}{5}\frac{\partial^2 u_1}{\partial x_1^2} + \lambda(u_1 - u_2) + \frac{1}{5}\lambda^2 e^{u_1} \tag{13}$$

$$\frac{\partial u_2}{\partial t} = \frac{1}{5}\frac{\partial^2 u_2}{\partial x_1^2} + \lambda(u_1 + u_2) + \frac{1}{5}\lambda^2 e^{u_1} \tag{14}$$

with boundary conditions

$$u_1(0) = u_1(1) = u_2(0) = u_2(1) = 0.$$

References

[1] G. M. Shroff, H. B. Keller, *Stabilization of unstable procedures: the Recursive Projection Method*, SIAM J. Numer. Anal., 30(4)(1993), pp. 1099-1120

Figure 1: Solution path of Bratu problem

Figure 2: Solution path of problem (13), (14)

Recursive Projection Method for detecting bifurcation points

Vladimír Janovský, Ondřej Liberda

Faculty of Mathemetics and Physics, Charles University, Prague, Czech Republic, Malostr. nám. 25, 118 00

Abstract: The Recursive Projection Method is tested on two model problems. The aim of the experiments is to investigate the relationship between the actual bifurcation scenario and the performance of the method. The theoretical analysis suggests that such a link may exist. The performed test are rather pesimistic.

1 Introduction

Let us consider a system of nonlinear equations:

$$G(u, \lambda) = 0, \tag{1}$$

where $G : \mathcal{R}^N \times \mathcal{R} \rightarrow \mathcal{R}^N$ is a sufficiently smooth mapping dependent on the bifurcation parameter λ. There are two main tasks:

1. seeking for the solution path

   $$\Gamma = \{(u, \lambda) \in \mathcal{R}^{n+1} : G(u, \lambda) = 0\},$$

   i.e. the set of points solving the system (1)

2. detecting bifurcation points along the solution path

For the principles of pathfollowing algorithms we refer e.g. the [10], [5] or [1]. In the large–scale computations the implementation of iterative techniques, e.g. [11], seems to be inevitable.

As the latter problem is concerned, we can generically encounter folds (turning points) and Hopf bifurcation points along the path Γ. Detecting folds is comparatively easy, see e.g. [3]. Computation of Hopf points represents a serious problem since a good initial guess is needed, see [12], [4]. Recent developments e.g. using bialternate product (see [4]) or Cayley transformation (see [9]) are certainly some alternatives but they suffer serious drawbacks.

The Recursive Projection Method (RPM), see [6], was designed to solve the above formulated problem. Its variants for the continuation of periodic solutions, see [7] and [8], were reported to be very successful. In this paper, we will analyse the performance of RPM. We have deliberately chosen the same examples as in [6].

The outline is as follows: In the next section we will briefly describe the continuation procedure we have chosen for a reference. The particular implementation of RPM is described in Section 3. Two model problems are solved in Section 4. The performence of RPM is evaluated in Conclusions.

2 Pathfollowing via predictor corrector

We recall the idea of a general pathfollowing algorithm, see [5]. Given two points $(u_{-1}, \lambda_{-1}) \in \Gamma$ and $(u_0, \lambda_0) \in \Gamma$ on the solution path we perform two steps to find a new point on Γ:

1. predictor step:

$$\lambda_1 = \lambda_0 + \delta\lambda,$$

$$u^{(0)} = u_0 + h_C \tau,$$

   where τ is a unit vector, e.g. the tangent vector to the solution path in the point (u_0, λ_0) or its approximation by secant vector. Scalars $\delta\lambda$ and h_C act like step sizes that can be changed during the continuation by means of adaptive methods.

2. corrector step: we try to find a point (u_1, λ_1) on Γ, i.e. we solve equation (1) for unknowns $u_1 \in \mathcal{R}^n$ while $\lambda_1 = \lambda^{(0)}$ remains fixed. Since $G(u, \lambda)$ is nonlinear we only find an approximation of (u_1, λ_1) by applying an iterative procedure on the system (1). These iterations are represented by a mapping $F : \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}^n$, i.e. we have got a fixed-point scheme

$$u^{(k+1)} = F(u^{(k)}, \lambda_1). \tag{2}$$

Moreover to enable continuation past turning points we introduce one scalar equation (the so called constraint):

$$N(u, \lambda) = 0. \tag{3}$$

According to [10] we use the perpendicularity condition

$$(w - w^{(0)})^T (w^{(0)} - w_0) = 0, \tag{4}$$

where

$$w = (u, \lambda), \qquad w_0 = (u_0, \lambda_0), \qquad w^{(0)} = (u^{(0)}, \lambda^{(0)}) \tag{5}$$

are vectors from $\mathcal{R}^{n+1}$.

As the corrector step, we have used Newton's method:

$$u^{(k+1)} = u^{(k)} - \left[G_u(u^{(k)}, \lambda_1)\right]^{-1} G(u^{(k)}, \lambda_1).$$

In fact we solve an augmented system

$$\hat{G}(u, \lambda) = (G(u, \lambda), N(u, \lambda)) = 0, \tag{6}$$

where $\hat{G}$ maps $\mathcal{R}^{n+1}$ on $\mathcal{R}^{n+1}$. Newton's method for this augmented system has the form

$$\begin{pmatrix} u^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} = \begin{pmatrix} u^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \hat{G}'(u^{(k)}, \lambda^{(k)}) \hat{G}(u^{(k)}, \lambda^{(k)}). \tag{7}$$

Here

$$\hat{G}' = \begin{pmatrix} G_u & G_\lambda \\ N_u & N_\lambda. \end{pmatrix}$$

In applications the mapping G often arises from discretization of physical models. Jacobian matrix G_u is then sparse. Therefore it is advantageous to implement iterative solvers for solving the system of linear equations in (7). We used a variant of Biconjugate Gradients method. The resulting method is pretty robust.

3 The Recursive Projection Method

In the following text we will restrict ourselves to the case when mapping G is obtained from spatial discretization of the right-hand side of the partial differential equation

$$\frac{\partial u}{\partial t} = \mathcal{G}(u, \lambda).$$

After the discretization we obtain an autonomous system of ODEs:

$$\frac{\partial u}{\partial t} = G(u, \lambda). \tag{8}$$

We are mainly concerned about steady states of (8) which are precisely the solutions of (1).

In [6] qualitatively different approach is used for solving the bifurcation problem (1). It is based on seeking for fixed point of the mapping (2) i.e. we solve the equation

$$u = F(u, \lambda) \tag{9}$$

instead of solving (1). The particular choice of F will be specified later.

3.1 Dynamical simulation

Let

$$\varphi(t; u, \lambda) \tag{10}$$

denote flow of the dynamical system (8) i.e. the solution of (8) with initial condition

$$\varphi(0; u, \lambda) = u.$$

For stable steady states of (8) it holds

$$\lim_{t \to \infty} \varphi(t; u, \lambda) = u. \tag{11}$$

Hence we arrive at a method for finding stable steady states:
Let us define

$$F(u, \lambda) = \varphi(\Delta t; u, \lambda) \tag{12}$$

where Δt is fixed. Now we employ fixed–point iterations (2).

3.2 Coupled iterations

Fixed-point procedure (2) will fail to converge during the continuation due to eigenvalues from spectrum $\sigma(G_u)$ crossing the imaginary axis. The steady states are then no more stable. We will follow the steps described in [6]. Let us denote $\mu_1, \ldots, \mu_N$ the eigenvalues of the Jacobian $F_u(u, \lambda)$. Generally the iterations (2) will fail to converge or the convergence rate will be slow due to m eigenvalues $\mu_1, \ldots, \mu_m$ lying outside the disk

$$K_\delta = \{z \in \mathcal{C} : \|z\| \leq 1 - \delta\}$$

for some $0 < \delta < 1$. The main idea of coupled iterations is to decompose the state space:

$$\mathcal{R}^n = \mathcal{P} \oplus \mathcal{Q},$$

where $\mathcal{P}$ is the maximal invariant subspace of F_u belonging to the eigenvalues $\mu_1, \dots, \mu_m$ and $\mathcal{Q}$ is the orthogonal complement of $\mathcal{P}$. Subspaces $\mathcal{P}$ and $\mathcal{Q}$ define orthogonal projectors P and Q respectively. Now we can replace the system

$$u = F(u, \lambda)$$

by couple of systems

$$\begin{aligned} p = f(p, q, \lambda) &= PF(p + q, \lambda), & (13) \\ q = g(p, q, \lambda) &= QF(p + q, \lambda), & (14) \end{aligned}$$

where $u = p + q$ is the unique decomposition of u with $p = Pu$ and $q = Qu$. It is shown in [6] that all the eigenvalues of

$$g_q(p, q, \lambda) = QF_u(p + q, \lambda)Q$$

lie in K_δ. Therefore we can perform Newton's iterations only on a small subpace $\mathcal{P}$ while continuing to use fixed-point iterations on $\mathcal{Q}$. The resulting scheme is then of the form:

$$\begin{aligned} [I - f_p(p^{(k)}, q^{(k)}, \lambda)](p^{(k+1)} - p^{(k)}) &= f(p^{(k)}, q^{(k)}, \lambda) - p^{(k)}, & (15) \\ q^{(k+1)} &= g(p^{(k)}, q^{(k)}, \lambda). & (16) \end{aligned}$$

3.3 Numerical realization of the RPM

To implement the scheme in practice we have to solve several problems:

1. Increasing the dimension of the unstable subspace $\mathcal{P}$ when the scheme is not convergent.

2. Updating the basis during the continuation.

3. Decreasing the dimension of the basis.

ad 1: We start the algorithm with $m = 0$. Generally only a single real eigenvalue μ_{m+1} or a pair of complex conjugate eigenvalues approach the boundary of the unit disk in complex plain. Hence one or two new vectors should be adjoined to the subspace $\mathcal{P}$. This can be done automatically during the continuation without evaluating the spectrum of F_u. Let us denote

$$\Delta q^{(k)} = q^{(k+1)} - q^{(k)}.$$

According to [6] the vector $\Delta q^{(k)}$ is approximately the k-th power iteration with matrix g_q applied to the vector $\Delta q^{(0)}$ provided F_u is Lipschitz continuous and the prediction $u^{(0)}$ is close enough to the exact solution $u_1 \equiv u_1(\lambda_1)$. This gives us possibility to compute new unstable directions from the Gram-Schmidt factorization of the two last difference vectors:

$$\left[\Delta q^{(k)}, \Delta q^{(k-1)}\right] = \hat{D}T,$$

where $T \in \mathcal{R}^{2\times 2}$ is upper triangular and $\hat{D} \in \mathcal{R}^{N\times 2}$ is orthogonal. Now we add one vector (the first column of $\hat{D}$) if $T_{11} \gg T_{22}$ or both columns of $\hat{D}$ otherwise to the basis.

ad 2: Let $Z \in \mathcal{R}^{n\times m}$ be the basis of the unstable subspace $\mathcal{P}$. Clearly Z depends on the parameter λ. This means that after each continuation step we should correct it by performing one or more steps of the so called subspace iterations:

$$Z(\lambda_1) = orth(F_u(u_1, \lambda_1)Z(\lambda_0)), \tag{17}$$

where "$orth$" denotes computing an orthonormal basis for F_uZ.

ad 3: After each successful continuation step we compute the eigenvalues of the matrix

$$H = Z^T F_u Z.$$

These should be the dominant eigenvalues of the Jacobian F_u. If only $\tilde{m} < m$ of them lie outside K_δ we decrease the dimension m of $\mathcal{P}$ by computing a real basis $V \in \mathcal{R}^{m\times\tilde{m}}$ for the dominant eigenvectors of H and replacing Z by $orth(ZV)$.

To benefit from the fact that dimension m is generally small we introduce new variables $z \in \mathcal{R}^m$:

$$z = Z^T u = Z^T p, \qquad p = Zz, \qquad q = u - Zz.$$

Together with orthogonality constraint (3) applied only on the subspace $\mathcal{P}$ we arrive at a $(m+1)$-dimensional system for z and λ:

$$\begin{aligned} z - Z^T F(Zz + q, \lambda) &= 0 \qquad (18) \\ N(z, \lambda) &= 0. \qquad (19) \end{aligned}$$

The orthogonality constraint on $\mathcal{R}^{m+1}$ takes the form

$$N(z, \lambda) = (z - z^{(0)})^T (z^{(0)} - z_0) + (\lambda - \lambda^{(0)})(\lambda^{(0)} - \lambda_0) = 0.$$

Applying Newton's method to the above system we get the scheme:

$$\begin{pmatrix} I - H & -Z^T F_\lambda \\ z^{(0)} - z_0 & \lambda^{(0)} - \lambda_0 \end{pmatrix} \begin{pmatrix} z^{(k+1)} - z^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} z^{(k)} - Z^T F \\ N(z^{(k)}, \lambda^{(k)}). \end{pmatrix} \qquad (20)$$

In the actual computing we do not know accurate values of the flow φ. That is why we employ a standard Runge–Kutta integrator with automatic step control to approximate these values:

$$\varphi(\Delta t; u, \lambda) \approx RK(\Delta t, u, \lambda, tol).$$

The last parameter sent to the Runge–Kutta procedure is the local discretization error tolerance. As shown in the tests it plays an important role for the RPM algorithm. In fact we call the Runge–Kutta procedure with two various choices of the last parameter:

1. With $tol = tolDerF$ we evaluate values of the flow more accurately for the purposes of computing differences

   $$\frac{F(u + \varepsilon v, \lambda) - F(u, \lambda)}{\varepsilon},$$

   which for small values of the scalar ε approximate Gateaux derivative of F in the direction v.

2. With $tol = tolF$ we evaluate values of the flow less accurately.

Let us give a sketch of the RPM algorithm we have implemented:

```
do
    cs := cs+1;
    SecantPredictor $(u_{-1}, \lambda_{-1}, u_0, \lambda_0, u, \lambda)$;
    $F := F(u, \lambda)$;
    $z := Z^T u; \quad q := u - Zz$;
    $H := Z^T F_u Z$
    **if** there are eigenvalues of $H$ in $K_\delta$ **then** DecreaseBasisSize();
    iter:=0;
    **while** $(\|u - F\| > tolRes)$ **do**
        iter := iter+1;
        $q := q - ZZ^T q$;
        $\begin{pmatrix} z \\ \lambda \end{pmatrix} := \begin{pmatrix} z \\ \lambda \end{pmatrix} - \begin{pmatrix} H & -Z^T F_\lambda \\ N_z & N_\lambda \end{pmatrix}^{-1} \begin{pmatrix} Z^T F \\ N(z, \lambda) \end{pmatrix}$;
        **if** iter > iterMax **then** IncreaseBasisSize();
        $u := Zz + q$;
        $F := F(u, \lambda)$;
    **endwhile**;
**until** end of continuation;
```

The following table lists parameters used in the RPM algorithm:

$tolRes$	—	used as a stopping criterion in the RPM corrector.
$tolF$, $tolDerF$	—	see the text above.
ε	—	used for approximating derivatives by differences.
Δt	—	time step in the dynamical simulation. We have used $\Delta t = 0.1$ in the experiments.
δ	—	important for the procedure *DecreaseBasisSize()*. We have used $\delta = 0.5$ in the experiments.
$iterMax$	—	the number of RPM Corrector iterations after which the procedure *IncreaseBasisSize* is involved. We have used $iterMax = 8$ in the experiments.
cs	—	counter of continuation steps
$iter$	—	RPM corrector steps

4 Model problems

For numerical tests we chose two examples mentioned in [6]. The first one is the so called 1-D Bratu problem.

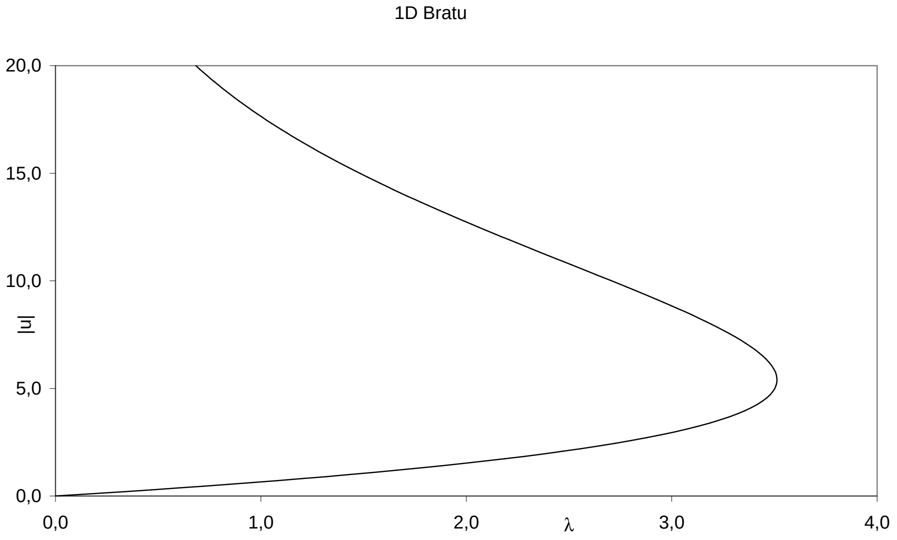

Figure 1: Bifurcation diagram for 1D Bratu problem

4.1 Bratu problem

It is described by partial differential equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + \lambda e^u & (21) \\ u(0) &= u(1) = 0. & (22) \end{aligned}$$

We can see that Bratu problem is perturbed heat conduction problem. We discretized this equation by means of finite differences and obtained mapping $G(u, \lambda)$ the i-th component of which was

$$G_i(u, \lambda) = \frac{1}{h^2}(u_{i-1} - 2u_i + u_{i+1}) + \lambda e^{u_i}.$$

Here h denotes the spatial mesh size

$$h = \frac{1}{n+1}$$

where n is number of mesh points. Figure 1 shows bifurcation diagram for Bratu problem computed by Newton's method with $n = 40$ and stopping criterion

$$\|G(u, \lambda)\| < 10^{-7}.$$

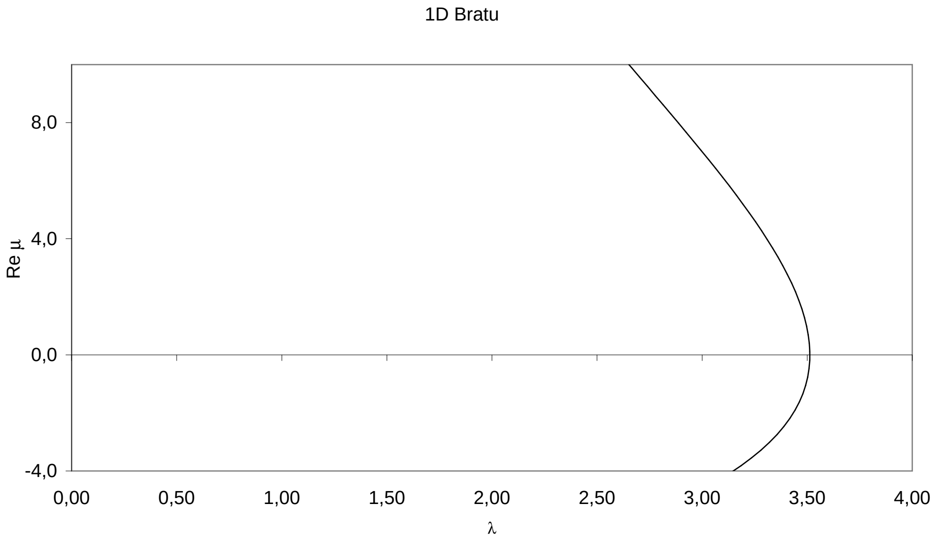

Figure 2: Dependence of Re μ on λ

test nr.	tol	tolF	tolderF	eps
I.	10^{-4}	10^{-5}	10^{-5}	10^{-3}
II.	10^{-3}	10^{-3}	10^{-6}	10^{-3}
III.	10^{-4}	10^{-6}	10^{-12}	10^{-6}

Table 1: Parameter settings for numerical experiments

We can see one fold point for $\lambda \approx 3.51$. Due to the symmetry of G_u, there are no Hopf bifurcation points on Γ. For the reference, we have computed spectrum of G_u at each detected point of Γ. As expected, for $\lambda \approx 3.51$ one real eigenvalue crossed the imaginary axis in the complex plain from left to the right. Consequently steady states lost their stability. This fact is documented in the Figure 2 which shows dependence of real part of the dominant eigenvalue Re μ on the bifurcation parameter λ. The performance of RPM was tested for three different settings of basic parameters, see Table 1. As far as the computation of bifurcation diagram is concerned, RPM produced the same solution set as is depicted in Figure 2. We were interested how the fold point was indicated. As expected, the value of m (i.e. dimension of the invariant subspace $\mathcal{P}$) has changed. On Figure 3, the dependence of m on the position on Γ is shown. Note that the actual point number is related to the actual arclength. This graph was obtained for the particular parameter

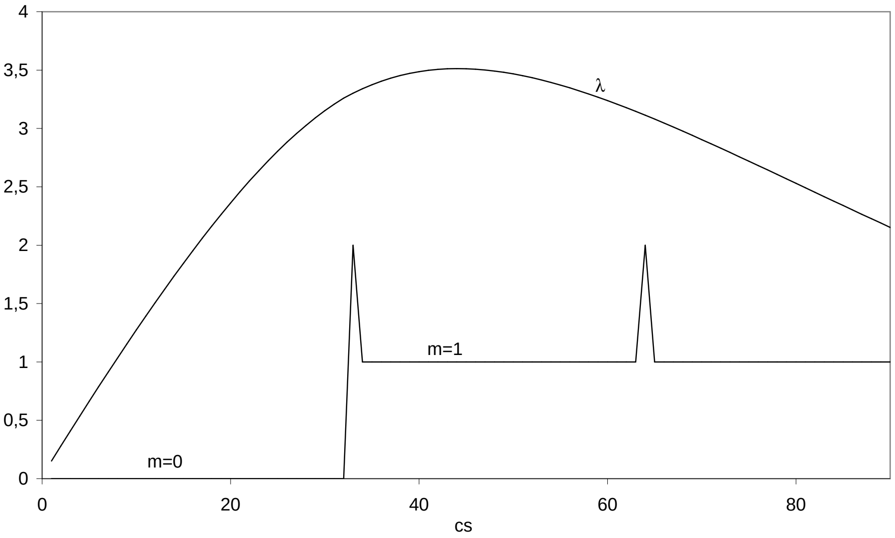

Figure 3: Bratu problem. Continuation via RPM with parameter setting II

setting II. We can see that dimension m of subspace $\mathcal{P}$ is increased to 2 and immediately after that decreased to 1. It is caused by inaccurate estimate of dominant eigenspace in window procedure.

4.2 Bratu problem 2

As the second example, we considered couple of PDEs:

$$\frac{\partial v}{\partial t} = \frac{1}{5}\frac{\partial^2 v}{\partial x_1^2} + \lambda(v - w) + \frac{1}{5}\lambda^2 e^v \tag{23}$$

$$\frac{\partial w}{\partial t} = \frac{1}{5}\frac{\partial^2 w}{\partial x_1^2} + \lambda(v + w) + \frac{1}{5}\lambda^2 e^v \tag{24}$$

with

$$v(0) = v(1) = 0, \qquad w(0) = w(1) = 0.$$

We will call the resulting system Bratu problem 2. Again we discretized the right–hand side by means of finite differences. We ordered the unknowns so that

$$u_{2i} = v_i, \qquad u_{2i+1} = w_i, \qquad i = 1, \ldots, n.$$

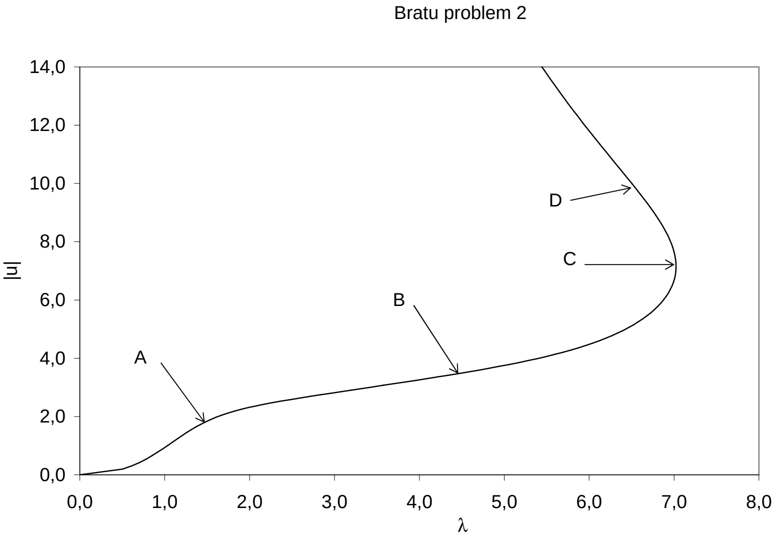

Figure 4: Bifurcation diagram for 1D Bratu problem 2

Figure 4 shows bifurcation diagram for Bratu problem 2 computed by Newton's method with $n = 20$ and stopping criterion $\|G(u, \lambda)\| < 10^{-7}$. We can see one fold point for $\lambda \approx 7.02$. However the behaviour of this model is much more complex than in case of Bratu problem. Due to the fact that this system is unsymmetric we obtain several Hopf bifurcation points on the path. In the figure 4 we marked 4 points of interest. They correspond to different scenarios on the complex plain. Points A and B on Γ are Hopf bifurcation points where two complex conjugate eigenvalues cross imaginary axis from left to the right. Point C is a fold where single real eigenvalue crosses imaginary axis from left to the right. At the point D on the curve a transverse bifurcation occurs where single real eigenvalue crosses imaginary axis from right to the left. Figure 5 shows the changes of the number of eigenvalues to the right of the imaginary axis during the continuation. Again we performed three tests with RPM with various settings of parameters, see Table 1. These various settings appeared to have great influence on the continuation. Especially they affect dimension m of the unstable subspace $\mathcal{P}$ and consequently the possibility to detect bifurcation points by testing the spectrum of the small matrix H. Having a look at the Figure 5 we could expect the following behaviour of the RPM algorithm:

- At points A and B two complex conjugate eigenvalues leaving the unit

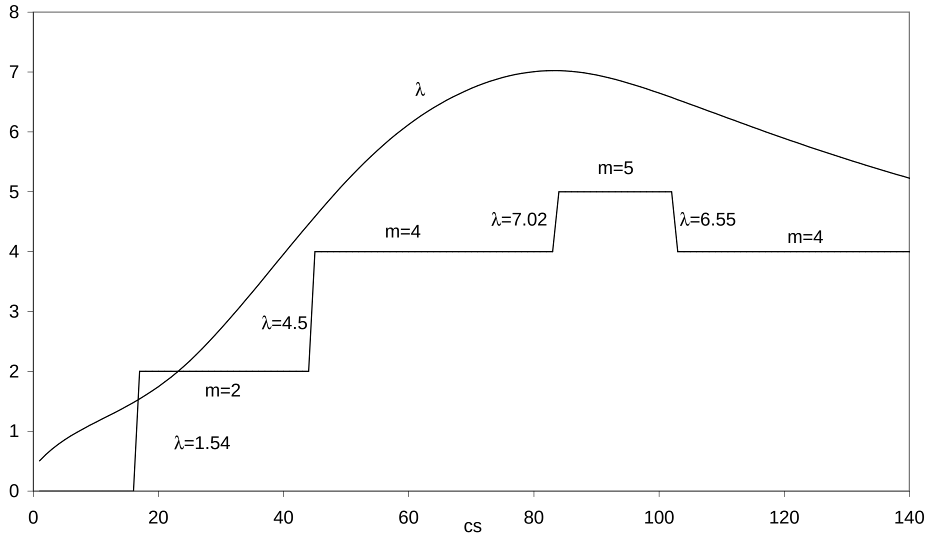

Figure 5: Bratu 2. Changes of number of unstable modes during continuation

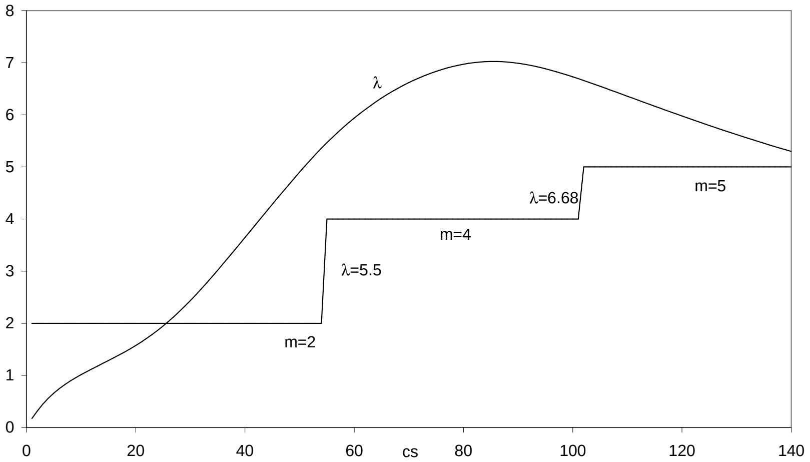

Figure 6: Bratu 2. Changes of m during the continuation via RPM with parameter setting I

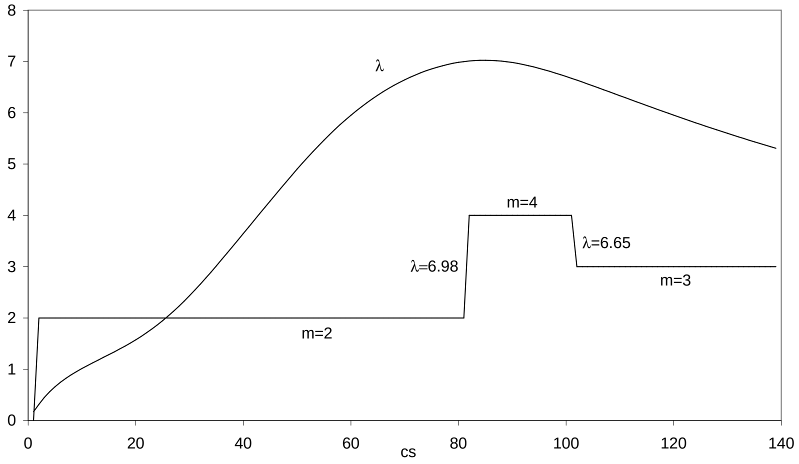

Figure 7: Bratu 2. Changes of m during the continuation via RPM with parameter setting II

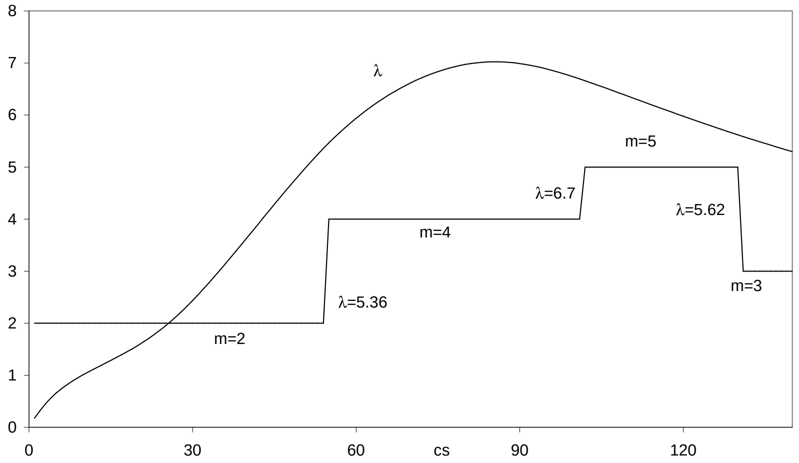

Figure 8: Bratu 2. Changes of m during the continuation via RPM with parameter setting III

circle causing an increase of the number of unstable modes to 2 or 4, resp.

- At the point C one real eigenvalue should leave the unit circle causing an increase of the number of unstable modes to 5.
- At the point D one real eigenvalue should enter the unit circle causing a decrease of the number of unstable modes to 4.

However the performance of the RPM suggested the following scenario. We observed that two complex conjugate eigenvalues left the unit circle at the point A and one real eigenvalue left the unit circle at the point C. The presence of bifurcation points B and D was not indicated.

5 Conclusions

The experiments verified that continuation via the RPM is satisfactory. Turning points were detected within the prescribed tolerance too. We encountered problems with detecting of Hopf bifurcations. Their detection appears to be sensitive on tuning the parameters. We envisage the following remedies of RPM. At first, we should be able to change the arithmetics in the sensitive part of RPM. We also prepare a version of RPM where the original equation is solved. Note that the present RPM is an iterative algorithm of fixed point type. Finally, a better approximation of the invariant subspace is badly needed. To this end we intend to explore [2].

References

[1] E. L. Allgower, K. Georg *Numerical continuation methods*, Springer Verlag, New York, 1990

[2] J.W.Demmel, L. Dieci, M. J. Friedman *Computing connecting orbits via an improved algorithm for continuing invariant subspaces*, to appear in STSC

[3] W. Govaerts, *Numerical methods for bifurcation of dynamical equilibria*, Philadelphia,

[4] W. Govaerts, *Computation of singularities in large nonlinear systems*, SIAM J. Numer. Anal., 34(1997), pp. 867-880

[5] H. B. Keller *Numerical methods in bifurcation problems*, Springer Verlag, New York, 1987

[6] G. M. Shroff, H. B. Keller, *Stabilization of unstable procedures: the Recursive Projection Method*, SIAM J. Numer. Anal., 30(4)(1993), pp. 1099–1120

[7] K. Lust, *Numerical bifurcation analysis of periodic solutions of partial differential equations*, Kathoelieke Universiteit Lueven, Ph.D. thesis, 1997

[8] K. Lust, D. Roose *Computation and bifurcation analysis of periodic solutions of large–scale systems*, IMA Preprint Series #1536, Feb 1998, IMA, University of Minnesota

[9] R. Neubert *Predictor corrector techniques for detecting Hopf points*, Ind. J. Bifurcation and Chaos 3 (1993), pp. 1311–1318

[10] W. C. Rheinboldt *Numerical analysis for large eigenvalue problems*, Manchester University Press, 1992

[11] Y. Saad *Numerical methods for large eigenvalue problems*, Manchester Press, Manchester, 1992

[12] B. Werner *Computation of Hopf bifurcation with bordered matrices*, SIAM J. Numer. Anal. 33 (1996) pp. 388–399

Projected version of the Recursive Projection Method algorithm

Vladimír Janovský[1], Ondřej Liberda[1]

Abstract: A modified version of the Recursive Projection Method called Projected RPM is proposed. Both the original and the Projected RPM are compared. The latter underlines the stabilization effect. Both methods suffer from the same drawback, namely a poor update of the unstable invariant subspace.

Keywords: steady states, pathfollowing, stability exchange, unstable invariant subspace

1 Introduction

In the following we consider a parameter dependent dynamical system

$$\dot{u} = G(u, \lambda) \tag{1}$$

We assume that $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is sufficiently smooth vector field.

In this paper, we shall concentrate on two issues:

1. Continuation of steady states; we have in mind large dynamical systems especially those arising from a discretisation of PDEs.

2. An indication of stability exchange in the course of pathfollowing.

As far as the first problem is concerned, the pathfollowing techniques are well known, see e.g. [14], [12], [1]. For the latest development, see [7]: Mixed block elimination techniques could exploit black box iterative solvers for very large matrices, [18].

The second problem is quite hard. There were some prospective ideas in this direction: The "global" test function for Hopf bifurcation points (based on the bialternate-product of matrices), see e.g. [9] or [7], would indicate the

[1] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

stability exchange. Unfortunately, this kind of test function could be reasonably computed for small scaled problems. Alternative techniques monitoring the "traffic" through the imaginary axis, see e.g. [8], suffer from the same shortcoming.

Continuation of the rightmost eigenvalue, see [17], can answer the posed question. This computational technique is based on properties of Cayley trasform, see [6]. Direct continuation of an invariant subspace, [5] and [2] might be an expensive but a promissing technique.

The Recursive Projection Method (RPM), [13], is designed to perform both of the above assigned jobs. The idea is to compute the steady states $G(u,\lambda)=0$ as parameter dependent fixed points of a mapping F namely,

$$u = F(u,\lambda)\,. \tag{2}$$

At each continuation step, the state space $\mathbb{R}^n$ is splitted as

$$\mathbb{R}^n = \mathbb{P} \oplus \mathbb{Q}\,, \quad \mathbb{Q} = \mathbb{P}^\perp$$

where $\mathbb{P}$ is the invariant subspace of $F_u(u,\lambda)$ containing all unstable modes. Denoting by P and Q the orthogonal projectors on $\mathbb{P}$ and $\mathbb{Q}$ respectively, the problem (2) can be reformulated as follows: Find $p \in \mathbb{P}$ and $q \in \mathbb{Q}$ such that

$$\begin{aligned} p &= PF(p+q,\lambda) & (3)\\ q &= QF(p+q,\lambda), & (4) \end{aligned}$$

In [13] there is shown that QF is contractive. Therefore, fixed points of (4) can be computed via Picard iterations. In order to find p, Newton-like methods are suggested. Note that the choice of F also includes the possibility to use any black-box ODE solver in the Picard iteration step, see [13]. This step can be interpreted as a dynamical simulation.

Just to resume, the iteration step consists in *dynamical simulation* applied to (4) and a low-dimensional Newton's correction applied on (3).

The preconditioned version of RPM, [4], tries to fight the inevitable stiffness of ODE's. The technique is justified for semilinear PDE. The version of RPM designed for continuation of limit cycles, [16], has been very successful.

An attempt to improve RPM was made in [3]. Numerical tests reported failure, [15]. The present paper is another attempt to reformulate RPM in the spirit of [3]. The resulting algorithm is called *Projected* RPM.

The outline of this paper is as follows: In the section 2.1 the Projected RPM algorithm is formulated. Theorem justifying the convergence is quoted. For details see [11]. In the section 2.2 a pseudocode of the algorithm is given. Section 3 contains results of the numerical test. The last section are the conclusions.

2 Projected RPM

In this section we will outline the key idea behind the Projected RPM. Let us consider an autonomous system of ODEs (1). In practical computations G is often a spatial discretization of a differential operator. Let $\Gamma \subset \mathbb{R}^n \times \mathbb{R}$ be the path of steady states of (1), i.e. the solution set to

$$G(u, \lambda) = 0. \tag{5}$$

2.1 Coupled iterations

Let $(u^*, \lambda^*) \in \Gamma$ be a steady state of (1). We will keep this point fixed in the whole section. Let us denote

$$\mu_k = \rho_k + i\tau_k, \qquad k = 1 \ldots, n \tag{6}$$

the eigenvalues of the Jacobian $A \equiv G_u(u^*, \lambda^*)$ and arrange them so that

$$\rho_1 \geq \ldots \geq \rho_m > 0 > \rho_{m-1} \geq \ldots \geq \rho_n.$$

Generally $m \ll n$, i.e. only a few modes cause the instability. Let $\mathbb{P}$ denote the maximal invariant subspace of A corresponding to the eigenvalues $\mu_1, \ldots \mu_m$. That means $A(\mathbb{P}) \subset \mathbb{P}$, $\sigma(A_{|\mathbb{P}}) = \{\mu_1, \ldots, \mu_m\}$ and $\mathbb{P}$ is maximal subspace with the above properties. Moreover let us define

$$\mathbb{Q} = \mathbb{P}^{\perp}.$$

Clearly $\mathbb{R}^n = \mathbb{P} \oplus \mathbb{Q}$. The above decomposition defines orthogonal projectors

$$\begin{aligned} P : \mathbb{R}^n &\rightarrow \mathbb{P}, \qquad \text{with } Ker(P) = \mathbb{Q} & (7) \\ Q : \mathbb{R}^n &\rightarrow \mathbb{Q}, \qquad \text{with } Ker(Q) = \mathbb{P}. & (8) \end{aligned}$$

From the definition it is evident that

$$I = P + Q,$$

where I denotes the identity operator. Now the solution to (5) is equivalent to the solution of the coupled system

$$\begin{aligned} G^P(p, q, \lambda) &\equiv PG(p + q, \lambda) = 0 & (9) \\ G^Q(p, q, \lambda) &\equiv QG(p + q, \lambda) = 0, & (10) \end{aligned}$$

where

$$G^P : \mathbb{P} \times \mathbb{Q} \times \mathbb{R} \rightarrow \mathbb{P} \qquad \text{and} \qquad G^Q : \mathbb{P} \times \mathbb{Q} \times \mathbb{R} \rightarrow \mathbb{Q}.$$

Let

$$p^* = Pu^*, \qquad q^* = Qu^*$$

be the projections of the steady state u^* on $\mathbb{P}$ and $\mathbb{Q}$, resp. Let us investigate equation (10). Partial differential $D_q G^Q(p^* + q^*, \lambda^*)$ is a linear mapping

$$QA : \mathbb{Q} \to \mathbb{Q}. \tag{11}$$

An analogy to the Lemma 2.10 from [13] can be proved:

Lemma: *The $n - m$ eigenvalues of the linear mapping (11) are exactly the stable modes $\mu_{m+1}, \ldots, \mu_n$ of A.*

The Lemma above gives us a possibility to solve the system (10) by dynamical simulation. Let us consider a dynamical system

$$\begin{aligned} \dot{q}(t) &= G^Q(p, q(t), \lambda) & (12)\\ q(0) &= q \in \mathbb{Q}. & (13) \end{aligned}$$

We will denote its flow by

$$\psi(t, q; p, \lambda).$$

Here p and λ are parameters while q is a state variable and t denotes time. Let us take a fixed time step $\Delta t > 0$ and define a mapping

$$\begin{aligned} F^Q : \mathbb{P} \times \mathbb{Q} \times \mathbb{R} &\to \mathbb{Q} & (14)\\ F^Q(p, q, \lambda) &= \psi(\Delta t, q; p, \lambda). & (15) \end{aligned}$$

An important consequence of the Lemma is a local convergence of Picard iterations via mapping F^Q in a neighborhood of (u^*, λ^*) for sufficiently large Δt. It is stated in the following theorem:

Theorem: *Let $G \in \mathcal{C}^1(U_{\delta_1}(u^*, \lambda^*))$ where $\delta_1 > 0$, $U_{\delta_1}(u^*, \lambda^*)$ is a δ_1-neighborhood of (u^*, λ^*) and F^Q be defined as above. Then there exist positive numbers γ, K, α and δ with*

$$K\gamma < \alpha, \qquad \delta < \delta_1$$

such that

$$\|F^Q(p^*, q, \lambda) - q^*\| \leq K\|q - q^*\| e^{-(\alpha - K\gamma)\Delta t}$$

whenever

$$\|q - q^*\| < \delta.$$

Moreover the sequence

$$q^{(k+1)} = F^Q(p^*, q^{(k)}, \lambda)$$

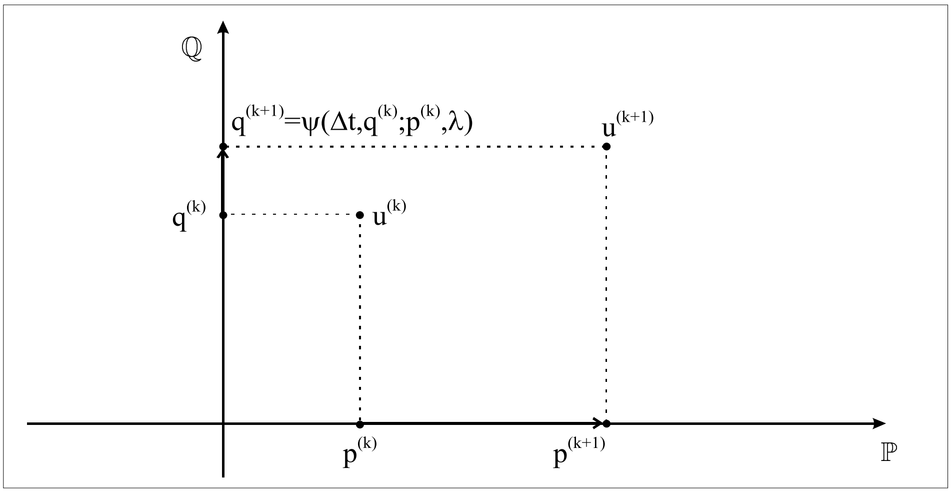

Figure 1: One step of the Projected RPM

converges to the fixed point $q^* = F^Q(p^*, q^*)$.

The system (9) on the unstable subspace $\mathbb{P}$ cannot be solved in the same manner as the system (10): the eigenvalues $\mu_1, \ldots, \mu_m$ of the linearization of G^P are to the right of the imaginary axis. Therefore we use Newton's method on $\mathbb{P}$. One Newton's step can be written in the form

$$p^{(k+1)} = F^P(p^{(k)}, q, \lambda)$$

with $F^P : \mathbb{P} \times \mathbb{Q} \times \mathbb{R} \to \mathbb{P}$,

$$F^P(p, q, \lambda) = p - \left(\frac{\partial G^P(p, q, \lambda)}{\partial p}\right)^{-1} G^P(p, q, \lambda). \tag{16}$$

Combining the two types of iterations on subspaces $\mathbb{P}$ and $\mathbb{Q}$ together we obtain the following iteration scheme:

$$p^{(k+1)} = F^P(p^{(k)}, q^{(k)}, \lambda) \tag{17}$$
$$q^{(k+1)} = F^Q(p^{(k)}, q^{(k)}, \lambda) \tag{18}$$

It can be shown that the above scheme is locally convergent. For details see [11]. We would like to emphasize that the analysis of local convergence is made under the assumptions that the decomposition $\mathbb{R}^n = \mathbb{P} \oplus \mathbb{Q}$ is known and the exact values of the flow ψ are available.

One step of the Projected RPM is sketched in the Fig.1. For comparison we introduce also a sketch of one iteration step of the original RPM method

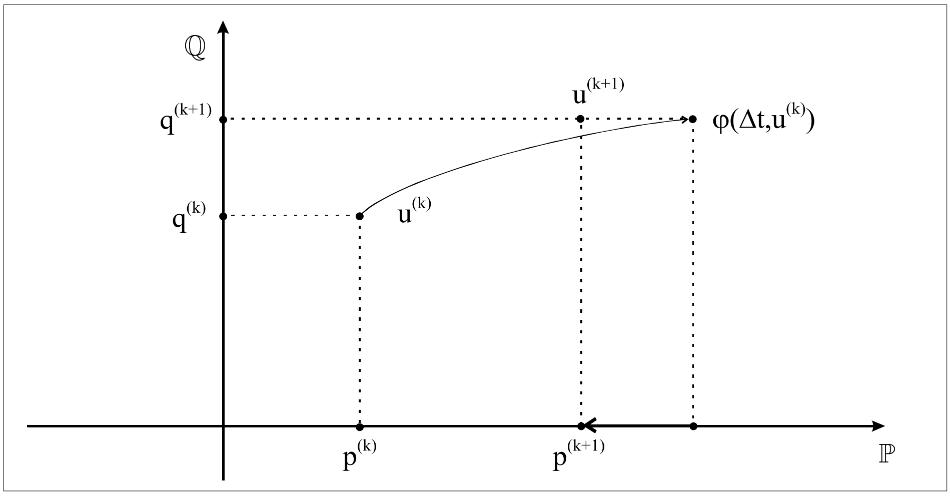

Figure 2: One step of the original RPM

starting from the same point $u^{(k)}$, see Fig.2. Let us comment the differences between the two schemes. For simplicity we dropped the dependence on the bifurcation parameter λ.

In order to understand Fig.2 we introduce the flow $\varphi(t, u^{(k)})$ of the dynamical system

$$\begin{aligned} \dot{u} &= G(u) && (19) \\ u(0) &= u^{(k)}. && (20) \end{aligned}$$

In the original RPM, the dynamical simulation is applied to compute the flow $\varphi(\Delta t, u^{(k)})$. Then $q^{(k+1)}$ is the $\mathbb{Q}$-coordinate of $\varphi(\Delta t, u^{(k)})$, while $p^{(k+1)}$ is the $\mathbb{P}$-coordinate of $\varphi(\Delta t, u^{(k)})$ corrected by one Newton step.

On the other hand, see Fig.1, in the projected RPM $q^{(k+1)}$ is obtained by solving *projected* dynamical system (12), (13) and $p^{(k+1)}$ is computed by Newton's step.

2.2 Numerical implementation of the Projected RPM

In this section we provide a code of the Projected RPM. We drop details about the predictor and concentrate on the corrector routine.

Let us assume that a point (u_0, λ_0) lies on the solution curve Γ within the prescribed tolerance. A starting point $(u^{(0)}, \lambda^{(0)})$ is obtained via standard predictor (we used secant method in the computations). Afterwards coupled iterations (17), (18) are applied until a new point on Γ is reached. In order to benefit from the fact that Newton's method is necessary only in low-dimensional space $\mathbb{P}$ we introduce new coordinate system which is represented by an orthonormal basis $Z \in \mathbb{R}^{n \times m}$. New coordinate $z \in \mathbb{R}^m$ is then defined by

$$z \equiv Z^T p = Z^T u.$$

Conversely

$$u = q + Zz.$$

Furthermore let

$$H(u, \lambda) = Z^T G_u(u, \lambda) Z$$

denote $m \times m$ matrix which is the restriction of Jacobian $G_u(u, \lambda)$ on the subspace $\mathbb{P}$. To make the continuation past turning points possible we use one-dimensional bordering of the system (9). An augmented Newton's step in new coordinates has the form

$$\begin{pmatrix} H & Z^T G_\lambda \\ \dot{z} & \dot{\lambda} \end{pmatrix} \begin{pmatrix} z^{(k+1)} - z^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} Z^T G \\ 0. \end{pmatrix} \tag{21}$$

where $\dot{z}$, $\dot{\lambda}$ are approximations of the tangent vectors to the solution path in the "old" point (u_0, λ_0) projected on $(\mathbb{P}, \lambda)$ plane. In the actual computation we have to approximate values of the flow ψ. We employed a standard Runge–Kutta integrator with automatic step control:

$$\psi(\Delta t, q; p, \lambda) \approx RK(\Delta t, p, q, \lambda, tol).$$

The last parameter sent to the Runge–Kutta procedure is the local discretization error tolerance *tol*. The Projected RPM corrector routine in terms of pseudocode looks as follows:

```
procedure ProjectedRPM ($u^{(0)}, \lambda^{(0)}$)
    $G := G(u^{(0)}, \lambda^{(0)})$;
    $z^{(0)} := Z^T u^{(0)}$;   $q^{(0)} := u^{(0)} - Zz^{(0)}$;
    $H := Z^T G_u(z^{(0)}, \lambda^{(0)}) Z$;
    $k := 0$;
    while ($\|G\| > tolRes$) do
        $q^{(k+1)} := RK(\Delta t, p^{(k)}, q^{(k)}, \lambda^{(k)}, tol)$;
        $\begin{pmatrix} z^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} := \begin{pmatrix} z^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \begin{pmatrix} H & Z^T G_\lambda \\ \dot{z} & \dot{\lambda} \end{pmatrix}^{-1} \begin{pmatrix} Z^T G \\ 0 \end{pmatrix}$;
        if ($k > iterMax$) then IncreaseBasisSize();
        $\mathtt{u}^{(\mathtt{k+1})} := \mathtt{Zz}^{(\mathtt{k+1})} + \mathtt{q}^{(\mathtt{k+1})}$;
        $\mathtt{G} := \mathtt{G}(\mathtt{u}^{(\mathtt{k+1})}, \lambda^{(\mathtt{k+1})})$;
        $\mathtt{k} := \mathtt{k} + 1$;
    **endwhile**;
    SubspaceIteration(Z);
**end**
```

In the algorithm above two procedure calls `IncreaseBasisSize` and `SubspaceIteration` are involved. The former is called in order to augment basis Z whenever coupled iterations fail to converge. The latter controls the accuracy of the invariant subspace $\mathbb{P}$. Both procedures are essentially based on the routines `increase_basis_size` and `power_iteration_step` described in sections 4.1 and 4.2 in [13].

The following table lists parameters used in the above algorithm together with values we used in the computations:

$\mathtt{tolRes} = 10^{-5}$	— used as a stopping criterion in the Projected RPM corrector.
$\Delta\mathtt{t} = 0.1$	— time step in the dynamical simulation
$\mathtt{iterMax} = 15$	— number of iterations after which the procedure *IncreaseBasisSize* is involved

3 Model problems

We compared performance of the Projected RPM with original RPM on two model problems. We provide three types of outputs:

- CPU times and number of function calls
- solution curves
- eigenvalue movies concerning unstable modes (paths of complex conjugate pairs are depicted in bold)

All computations were made on PC with processor Celeron on 433MHz and 384MB RAM under OS Linux. All methods were coded in C++ language and compiled by GNU gcc compiler.

	Original RPM		Projected RPM	
	$nx = 30$	$nx = 100$	$nx = 30$	$nx = 100$
CPU time $[s]$	400	1689	120	901
nGCall	6 499 101	8 803 972	1 705 799	3 677 995

Table 1: CPU times for nonsymmetric system in 1-D

3.1 Nonsymmetric system in dimension 1-D

We consider a nonsymmetric system of PDEs:

$$\begin{aligned}\frac{\partial v}{\partial t} &= \frac{1}{5}\frac{\partial^2 v}{\partial x^2} + \lambda(v-w) + \frac{1}{5}\lambda^2 e^v \\ \frac{\partial w}{\partial t} &= \frac{1}{5}\frac{\partial^2 w}{\partial x^2} + \lambda(v+w) + \frac{1}{5}\lambda^2 e^w\end{aligned}$$

with

$$v(0) = v(a) = 0, \qquad w(0) = w(a) = 0.$$

In all computations we set $a = 1$. The above system is discretized by finite differences. A number of mesh points is denoted by nx. Table 1 shows CPU times and number of evaluation of the right-hand side G (nGCall) for problems with $nx = 30$ and $nx = 100$. Notice that the dimension of the resulting problem is 60 or 200. Fig. 3 depicts the solution curve computed by Projected RPM for $nx = 100$. Fig. 4 shows the dependence of real parts of the eigenvalues $\mu_1, \ldots, \mu_m$ on the number of continuation steps cs. For comparison we computed the whole spectrum by Newton's method (QR method applied on Jacobian). Fig. 5 depicts the actual eigenvalue movie of the unstable modes. We observed that Hopf bifurcation point H2 was not detected by the Projected RPM.

3.2 Nonsymmetric system in 2-D

Let us consider nonsymmetric system described in section 3.1 on a rectangle

$$(0,a) \times (0,b) \subset \mathbb{R}^2,$$

i.e. the following system of PDEs:

$$\begin{aligned}\frac{\partial v}{\partial t} &= \frac{1}{5}\Delta v + \lambda(v-w) + \frac{1}{5}\lambda^2 e^v \\ \frac{\partial w}{\partial t} &= \frac{1}{5}\Delta w + \lambda(v+w) + \frac{1}{5}\lambda^2 e^w\end{aligned}$$

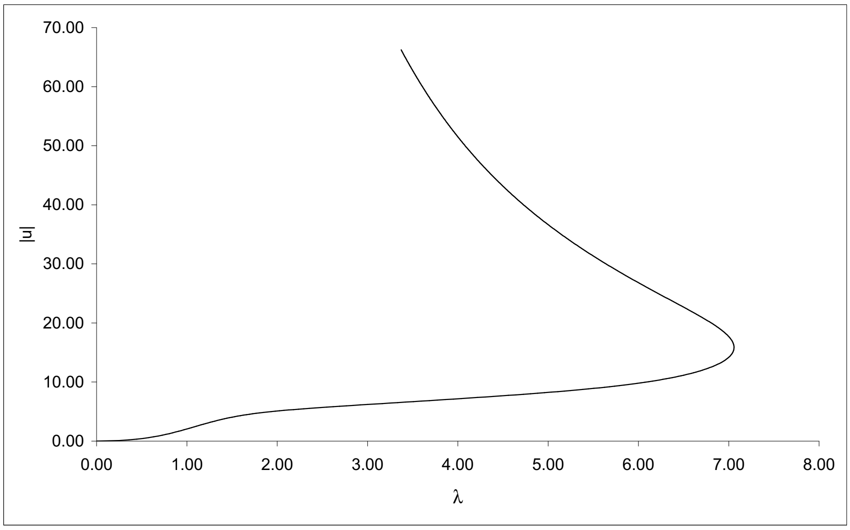

Figure 3: Solution path of nonsymmetric system in 1D for nx=100 computed by the Projected RPM

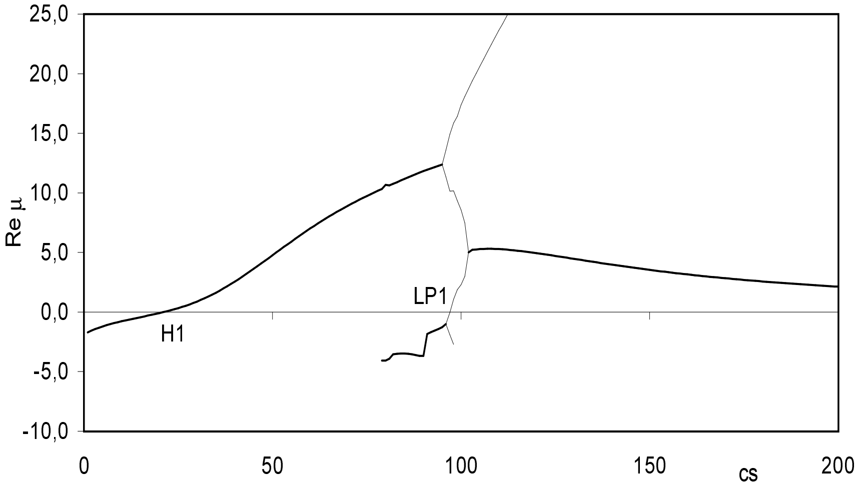

Figure 4: Eigenvalue movie of the unstable modes computed by Projected RPM

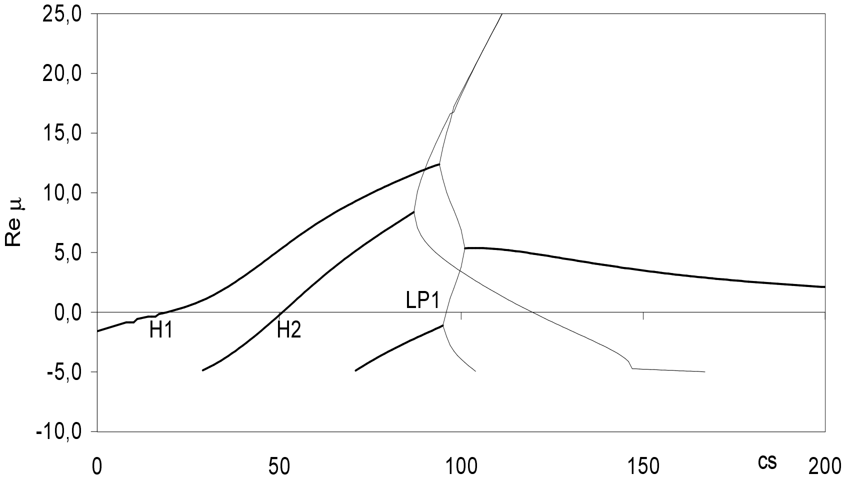

Figure 5: Eigenvalue movie of the unstable modes computed by Newton's method

with Dirichlet boundary conditions

$$v(0,y) = v(a,y) = 0, \qquad w(0,y) = w(a,y) = 0, \qquad y \in (0,b),$$

$$v(x,0) = v(x,b) = 0, \qquad w(x,0) = w(x,b) = 0, \qquad x \in (0,a).$$

In all computations we set $a = 1$ and $b = 2$. The above system is discretized by finite differences with number of mesh points along x and y axis denoted by nx and ny. Table 2 shows CPU times and number of evaluations of the right-hand side G (nGCall) for problems with $nx = ny = 5$ and $nx = ny = 10$. Notice that the dimension of the resulting problem is 50 or 200, resp. Mind the paradoxical ratio of the CPU times for the cases $nx = ny = 5$ and $nx = ny = 10$. It is caused by the fact that the portions of the computed curves were not the same. The reason is the failure of both algorithms for $nx = ny = 10$. Fig. 6 depicts a solution curve computed by the Projected RPM for $nx = ny = 10$. Fig. 7 shows dependence of real part of the unstable modes $\mu_1, \ldots, \mu_m$ on the number of continuation steps cs.

4 Conclusions

Numerical tests indicate that the Projected RPM is superior to the original version of the RPM. In our opinion it is due to the four following reasons:

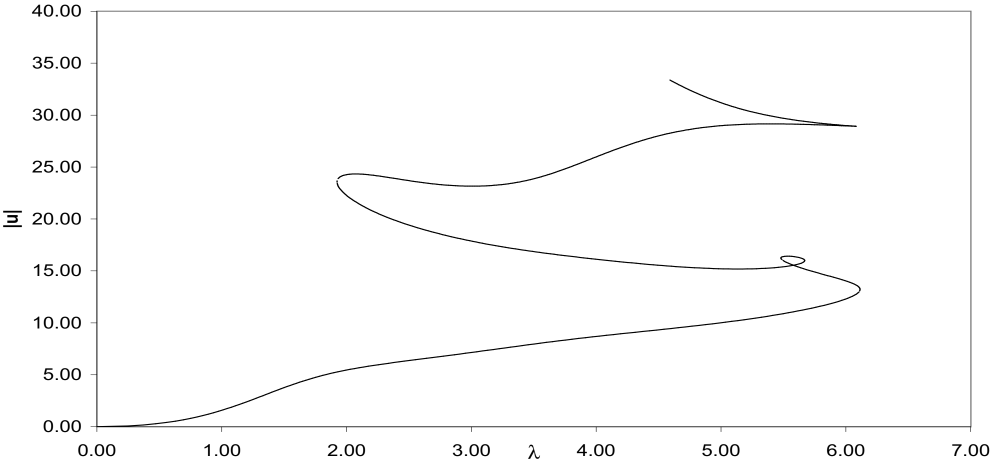

Figure 6: Solution path of nonsymmetric system in 2-D for nx=ny=10 computed by the Projected RPM

	Original RPM		Projected RPM	
	$nx = 5$	$nx = 10$	$nx = 5$	$nx = 10$
CPU time $[s]$	639	547	173	189
nGCall	11 150 076	2 420 610	2 417 740	651 889

Table 2: CPU times for nonsymmetric system in 2-D

- Solution of the dynamical system (12), (13) is bounded due to the Lemma. On the other hand the solution of the dynamical system (19), (20) may be unbounded whenever any unstable mode is present.

- In the Projected RPM the original system (5) is solved, especially Newton's step (16) is clearly defined. On the other hand the original version of the RPM relies on the fixed-point formulation which is not natural. In particular, the smallness of $\|u - F(u, \lambda)\|$ does not imply the smallness of $\|G(u, \lambda)\|$.

- Relation between the spectra $\sigma(G_u)$ and $\sigma(F_u)$ is not clear when a black-box solver is used. It is crucial for detection of Hopf bifurcation points.

- In usual cases when G arises from the discretization of PDEs the differential G_u is available explicitly. It is not true for F_u due to the

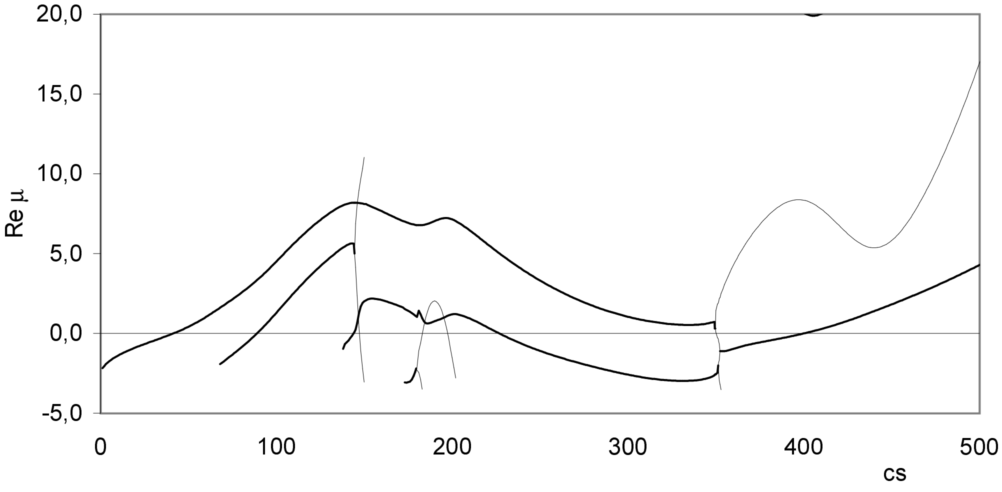

Figure 7: Eigenvalue movie of the unstable modes for nonsymmetric system in 2-D with nx=ny=10 computed by the Projected RPM

machinery behind the fixed-point formulation.

However both methods suffer from the same drawback: They seem to be sensitive to the accuracy with which we compute the basis Z. Unfortunately procedures maintaining the basis in both original and Projected RPM are essentially the same. We conclude that for any future effort to improve RPM this will be the key problem.

References

[1] E. L. Allgower, K. Georg *Numerical continuation methods*, Springer Verlag, New York, 1990

[2] W.-J. Beyn, W. Kless, V. Thümmer *Continuation framework for invariant subspaces and its application to travelling waves*, In: F. Keil, W. Mackens, H. Voss and J. Werther eds., Scientific Computing in Chemical Engineering II, pp 144–151, Springer, 1999

[3] J. Bošek, V. Janovský *A note on the recursive projection method* Proceedings of GAMM96, ZAMM 1997, Berlin, pp.437-440

[4] B. D. Davidson *Large-scale continuation and numerical bifurcation for partial differential equations*, SIAM J. Numer. Anal., 34(5)(1997), pp. 2001–2027

[5] J. W. Demmel, L. Dieci, M. J. Friedman *Computing connecting orbits via an improved algorithm for continuing invariant subspaces*, to appear

[6] T. J. Garratt, G. Moore, A. Spence *A generalised Cayley transform for the numerical detection of Hopf bifurcation points in large systems*, In Contributions in numerical mathematics, pp. 177–195, World Sci. Publ. 1993

[7] W. Govaerts, *Numerical methods for bifurcation of dynamical equilibria*, SIAM, Philadelphia, 2000

[8] W. Govaerts, A. Spence *Detection of Hopf points by counting sectors in the complex plane*, Numer. Math. 75(1996), pp. 43–58

[9] J. Guckenheimer, M. Myers, B. Sturmfels *Computing Hops bifurcation I*, SIAM J. Numer. Anal 34(1997), pp. 1–21

[10] V. Janovský, O. Liberda *Recursive Projection Method for detecting bifurcation points*, Proceedings SANM'99, pp 121–134, Union of Czech Mathematicians and Physicists, 1999

[11] V. Janovský, O. Liberda *Continuation of invariant subspaces via RPM*, in preparation

[12] H. B. Keller *Numerical methods in bifurcation problems*, Springer Verlag, New York, 1987

[13] G. M. Shroff, H. B. Keller, *Stabilization of unstable procedures: the Recursive Projection Method*, SIAM J. Numer. Anal., 30(4)(1993), pp. 1099–1120

[14] M. Kubíček, M. Marek *Computational Methods in Bifurcation Theory and Dissipative Structures*, Springer(1983)

[15] Š. Kubíček *Praktické implementace metod RPM1 a RPM2*, Diplomová práce, MFF UK Praha, 1997

[16] K. Lust, D. Roose *Computation and bifurcation analysis of periodic solutions of large–scale systems*, IMA Preprint Series #1536, Feb 1998, IMA, University of Minnesota

[17] R. Neubert *Predictor corrector techniques for detecting Hopf points*, Int. J. Bifurcation and Chaos 3 (1993), pp. 1311–1318

[18] Y. Saad *Iterative methods for sparse linear systems*, PWS Publishing Company, Boston, 1996

Acknoledgments: The authors gratefully acknowledge the support of grant MSM 113200007.

Continuation of invariant subspaces via the Recursive Projection Method

Vladimír Janovský[1], Ondřej Liberda[1]

Abstract: The Recursive Projection Method is a technique for continuation of both steady states and dominant invariant subspaces. In this paper a modified version of the RPM called projected RPM is proposed. The modification underlines the stabilization effect. In order to improve the poor update of the unstable invariant subspace we applied subspace iterations preconditioned by Cayley transform. A statement concerning the local convergence of the resulting method is proved. Results of numerical tests are presented.

Keywords: steady states, pathfollowing, stability exchange, unstable invariant subspace

AMS Classification: 65H17, 47H17

1 Introduction

We consider a parameter dependent dynamical system

$$\dot{u} = G(u, \lambda), \tag{1}$$

where $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is sufficiently smooth vector field. In order to find the steady states of (1), i.e. to solve the system of nonlinear equations

$$G(u, \lambda) = 0 \tag{2}$$

we can use a standard predictor-corrector continuation, see e.g. [7], [1]. To indicate the stability exchange in the course of pathfollowing is much harder. The Recursive Projection Method (RPM), see [10], could serve to this purpose.

[1] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

The classical RPM computes the steady states of (1) as parameter dependent fixed points of a mapping F, namely

$$F(u,\lambda) = u. \tag{3}$$

At each continuation step, the state space $\mathbb{R}^n$ is split as

$$\mathbb{R}^n = \mathbb{P} \oplus \mathbb{Q}, \quad \mathbb{Q} = \mathbb{P}^\perp$$

where $\mathbb{P}$ is the invariant subspace of $F_u(u,\lambda)$ containing all unstable modes. Denoting by P and Q the orthogonal projectors on $\mathbb{P}$ and $\mathbb{Q}$ respectively, the problem (3) can be reformulated as follows: Find $p \in \mathbb{P}$ and $q \in \mathbb{Q}$ such that

$$p = PF(p+q,\lambda) \tag{4}$$

$$q = QF(p+q,\lambda), \tag{5}$$

Under certain assumptions, see [10], QF is contractive. Therefore, fixed points of (5) can be computed via Picard iterations. In order to find p, Newton-like methods are suggested. The choice of F also includes the possibility to use any black-box ODE solver in the Picard iteration step, see [10]. This step can be interpreted as a dynamical simulation. As far as the stability exchange is concerned, the path of $\mathbb{P}$'s is supposed to be one of the outputs of RPM. Therefore, the unstable modes can be extracted from a small dimensional $\mathbb{P}$ at a reasonable cost.

The presence of modes with large negative real parts (which are irrelevant for the bifurcation analysis) can slow down RPM algorithms rapidly. An attempt to eliminate the influence of these modes by a preconditioning was made in [3].

Note that the technique for a continuation of limit cycles, [9], originated from [10].

According to our own experience with RPM, [5], the detection of Hopf bifurcation points is not reliable. From a theoretical point of view, the relationship between the spectra $\sigma(G_u)$ and $\sigma(F_u)$ is not clear (except for a very simple black-box solver, certainly without an adaptive time-stepping); this is crucial for a detection of Hopf bifurcation points. In general, the weak point of RPM is a poor approximation of $\mathbb{P}$.

In [6] we introduced a modification of RPM which we refer to as *Projected RPM*; note that we were hinted by [2]. Projected RPM attempts to solve (2) directly without a fixed-point reformulation (3). Let us point out the advantages:

- In the dynamical simulation step, a suitably projected vector field is integrated. It underlines the stabilisation effects.

- Differential G_u is often explicitly available. It is not true for F_u due to the machinery behind the fixed-point formulation.

Concerning an update of $\mathbb{P}$: In the original RPM, subspace iterations were suggested. However, subspace iterations tend to span an invariant subspace corresponding to the eigenvalues with large moduli. Unfortunately these need not be the rightmost eigenvalues.

Therefore we propose to use subspace iterations preconditioned by Cayley transform, see [4]. The effect is that the rightmost modes are mapped outside the unit circle.

The paper is organized as follows: In section 2 the Projected RPM is formulated; a pseudocode is given. Important parts of the algorithm such as a correction of the basis of $\mathbb{P}$ and changing the dimension of $\mathbb{P}$ are discussed in detail. In section 3 we state and prove a theorem about the local convergence of the Projected RPM. The last section contains results of the numerical experiment.

2 Projected RPM

First we introduce some notation. Let $\Gamma \subset \mathbb{R}^n \times \mathbb{R}$ be the solution set to (2) and let $(u^*, \lambda^*) \in \Gamma$ be an arbitrary point. By A we denote the Jacobian matrix

$$A = D_u G(u^*, \lambda^*).$$

Stability properties of the steady state u^* are determined by the eigenvalues $\mu_1, \ldots, \mu_n$ of A. Let us arrange them decreasingly with respect to their real parts,

$$\Re\mu_1 \geq \ldots \geq \Re\mu_m \geq 0 > \Re\mu_{m-1} \geq \ldots \geq \Re\mu_n.$$

Let $\mathbb{P}$ be the invariant subspace corresponding to the first m eigenvalues and let Z_1 be an orthonormal basis in $\mathbb{P}$. Thus $Z_1 \in \mathbb{R}^{n\times m}$ satisfies the following equations:

$$\begin{aligned} AZ_1 &= Z_1\Lambda_1 \\ Z_1^T Z_1 &= I_m, \end{aligned}$$

where $I_m \in \mathbb{R}^{m\times m}$ is an identity matrix and $\Lambda_1 \in \mathbb{R}^{m\times m}$ is a square matrix with eigenvalues $\mu_1, \ldots, \mu_m$. Further let us define the orthogonal complement

$$\mathbb{Q} = \mathbb{P}^\perp,$$

and let Z_2 be an orthonormal basis in $\mathbb{Q}$, i.e.

$$Z_2^T Z_2 = I_{n-m}.$$

From the definition of $\mathbb{Q}$ it follows that $Z_1^T Z_2 = 0 \in \mathbb{R}^{m\times(n-m)}$ and $Z_2^T Z_1 = 0 \in \mathbb{R}^{(n-m)\times m}$. For each vector $u \in \mathbb{R}^n$ there exist unique vectors $p \in \mathbb{R}^m$ and $q \in \mathbb{R}^{n-m}$ such that $u = Z_1 p + Z_2 q$. The assertion is proved by setting $p = Z_1^T u$ and $q = Z_2^T u$. Especially $u^* = Z_1 p^* + Z_2 q^*$.

2.1 Coupled iterations

Let $u^{(k)}$ be the k-th approximation to u^*, $u^{(k)} = Z_1 p^{(k)} + Z_2 q^{(k)}$. To define $u^{(k+1)}$ we perform one step of coupled iteration: System (2) can be split into two systems

$$\begin{aligned} G^P(p,q,\lambda) &\equiv Z_1^T G(Z_1 p + Z_2 q, \lambda) = 0 && (6)\\ G^Q(p,q,\lambda) &\equiv Z_2^T G(Z_1 p + Z_2 q, \lambda) = 0, && (7)\end{aligned}$$

where

$$G^P : \mathbb{R}^m \times \mathbb{R}^{n-m} \times \mathbb{R} \longrightarrow \mathbb{R}^m \qquad \text{and} \qquad G^Q : \mathbb{R}^m \times \mathbb{R}^{n-m} \times \mathbb{R} \longrightarrow \mathbb{R}^{n-m}$$

are projections of $G(u,\lambda)$ onto $\mathbb{R}^m$ and $\mathbb{R}^{n-m}$, resp. Equation (6) is solved with respect to the unknown p by Newton's method. Namely, we set

$$p^{(k+1)} = p^{(k)} - \left(D_p G^P(p^{(k)}, q^{(k)}, \lambda)\right)^{-1} G^P(p^{(k)}, q^{(k)}, \lambda). \qquad (8)$$

Equation(7) is solved by means of dynamical simulation — for fixed time step Δt we set

$$q^{(k+1)} = \psi(\Delta t, q^{(k)}; p^{(k)}, \lambda), \qquad (9)$$

where $\psi(t, q; p, \lambda)$ denotes the flow of parameter dependent dynamical system

$$\begin{aligned} \dot{q}(t) &= G^Q(p, q(t), \lambda) && (10)\\ q(0) &= q. && (11)\end{aligned}$$

It is important for the numerical stability that the projected vector field G^Q doesn't contain unstable modes. In practice we approximate the flow by means of a blackbox solver for ODEs. One step of coupled iteration can be written in the form

$$\begin{aligned} p^{(k+1)} &= F^P(p^{(k)}, q^{(k)}, \lambda) && (12)\\ q^{(k+1)} &= F^Q(p^{(k)}, q^{(k)}, \lambda) && (13)\\ u^{(k+1)} &= Z_1 p^{(k+1)} + Z_2 q^{(k+1)}, && (14)\end{aligned}$$

where

$$F^P(p,q,\lambda) = p - \left(D_p G^P(p,q,\lambda)\right)^{-1} G^P(p,q,\lambda)$$

and

$$F^Q(p,q,\lambda) = \psi(\Delta t, q; p, \lambda).$$

In fact, basis Z_2 need not be computed at all. Indeed, according to the equation (14) we only need to evaluate the quantity $Z_2 q$. To this end we set $v(t) = Z_2 q(t)$. After differentiating v with respect to t and inserting into (10), (11) we arrive at the dynamical system

$$\begin{aligned} \dot{v}(t) &= Z_2 Z_2^T G(Z_1 p + v(t), \lambda) = (I - Z_1 Z_1^T) G(Z_1 p + v(t), \lambda) \\ v(0) &= Z_2 q = v. \end{aligned}$$

We use the coupled iterations (12), (13), (14) in the context of the standard pathfollowing of $(u(s), \lambda(s))$; we assume the arclength parametrization. In particular, let (u_0, λ_0) be a point on Γ. Let $(\dot{u}, \dot{\lambda})$ be the unit tangent vector (or its approximation) at (u_0, λ_0). Let $u^{(0)}, \lambda^{(0)}$ be the relevant predictor; in the actual computation we used secant method. Then

$$\frac{1}{n}\left(u^{(k)} - u_0\right)^T \dot{u} + \left(\lambda^{(k)} - \lambda_0\right)\dot{\lambda} = (\delta s)^2 ,$$

generates the sequence $(u^{(k)}, \lambda^{(k)})$ of correctors; $u^{(k)}$ is given by means of (14).

2.2 Approximation of the basis Z_1

To obtain a reasonable method we have to reflect the fact that invariant subspace $\mathbb{P}$ changes in the course the continuation. Also the dimension of $\mathbb{P}$ may change due to the eigenvalue(s) crossing the imaginary axis. Moreover, for a detection of bifurcation points along the solution path it is necessary that $\mathbb{P}$ also contains the eigenvectors corresponding to the modes with $\Re\mu > \delta$ for some $\delta < 0$. In other words, when a single eigenvalue or a pair of complex conjugate eigenvalues travels through the imaginary axis from the left to the right we should add the corresponding eigenvectors to Z_1 in advance. Therefore it is important to correct the basis Z_1 after each continuation step and to ensure that it spans the invariant subspace corresponding to the eigenvalues with $\Re\mu > \delta$.

In [10] it is suggested to use subspace iterations with Jacobian A on basis Z_1. It is not suitable for our case since the rightmost eigenvalues need not be the dominant ones. The remedy is to use the so called *Cayley transform* of A. The resulting algorithm is referred to as subspace iterations preconditioned by Cayley transform in the literature. In the following we will outline the basic idea behind it, for details see [4].

Cayley transform of matrix A is defined by

$$\mathcal{C}(A) = (A - \alpha_1 I)^{-1} (A - \alpha_2 I) ,$$

for $\alpha_2 < \alpha_1$, $\alpha_1 \notin \sigma(A)$. The most important property is that it maps the eigenvalues with $\Re\mu < \frac{1}{2}(\alpha_1 + \alpha_2)$ inside the unit circle and eigenvalues with $\Re\mu > \frac{1}{2}(\alpha_1 + \alpha_2)$ outside the unit circle. It follows that well chosen α_1 and α_2 enable us to check the δ-boundary in the complex plain. Moreover, vectors in Z_1 will converge more rapidly to the eigenvectors corresponding to the rightmost eigenvalues. The resulting algorithm looks as follows:

```
procedure CorrectBasisViaCayley($\delta, \alpha_1, \alpha_2$)
    Add 6 randomly initialized vectors to $\mathtt{Z}_1$;
    $\mathtt{Q}^{(0)} = \mathtt{orth}(\mathtt{Z}_1)$;
    $\mathcal{C}(\mathtt{A}) = (\mathtt{A} - \alpha_1 \mathtt{I})^{-1}(\mathtt{A} - \alpha_2 \mathtt{I})$
    for $\mathtt{k} = 1, \dots, \mathtt{k}_{\mathtt{max}}$ do
        $\mathtt{V}^{(\mathtt{k})} = \mathcal{C}(\mathtt{A})\mathtt{Q}^{(\mathtt{k}-1)}$;
        $\mathtt{Q}^{(\mathtt{k})} = \mathtt{orth}(\mathtt{V}^{(\mathtt{k})})$;
    end for
    $\mathtt{H} = (\mathtt{Q}^{(\mathtt{k}_{\mathtt{max}})})^{\mathtt{T}}\mathtt{A}\mathtt{Q}^{(\mathtt{k}_{\mathtt{max}})}$;
    Compute $\sigma(\mathtt{H}) = \{\mu_1, \dots, \mu_{\mathtt{m}+6}\}$;
    Solve $\mathtt{A}\mathtt{z}_\mathtt{k} = \mu_\mathtt{k}\mathtt{z}_\mathtt{k}$ for $\mathtt{k} = 1, \dots, \mathtt{m}+6$
    $\mathtt{m}_{\mathtt{new}} = 0$;
    for $\mathtt{k} = 1, \dots, \mathtt{m}+6$ do
        if $\Re\mu_\mathtt{k} > \delta$ then
            Add $\mathtt{Q}^{(\mathtt{k}_{\mathtt{max}})}\mathtt{z}_\mathtt{k}$ to $\mathtt{Z}_1$;
            $\mathtt{m}_{\mathtt{new}} = \mathtt{m}_{\mathtt{new}} + 1$;
        end if;
    end for;
    $\mathtt{m} = \mathtt{m}_{\mathtt{new}}$;
```

In the code above orth(V) means the orthonormalization of V by Gramm-Schmidt proccess. The spectrum $\sigma(H)$ can be easily computed by QR-method since the dimension $m+6$ is small. We have chosen to add 6 randomly initialized vectors to the old basis Z_1 before the actual subspace iterations are performed. It enables us to increase the basis size in case when an eigenvalue (or a pair of complex conjugate eigenvalues) passes the δ-boundary.

2.3 Code of the Projected RPM

Combining all the things together, i.e. coupled iterations, pseudo-arclength condition and `CorrectBasisViaCayley` procedure, we arrive at the following code:

```
procedure ProjectedRPM ($\mathtt{u}^{(0)}, \lambda^{(0)}$)
    $\mathtt{G} = \mathtt{G}(\mathtt{u}^{(0)}, \lambda^{(0)})$;
    $\mathtt{p}^{(0)} = \mathtt{Z}_1^{\mathtt{T}}\mathtt{u}^{(0)}$;    $\mathtt{v}^{(0)} = \mathtt{u}^{(0)} - \mathtt{Z}_1\mathtt{p}^{(0)}$;
    $\mathtt{H} = \mathtt{Z}_1^{\mathtt{T}}\mathtt{G}_{\mathtt{u}}(\mathtt{u}^{(0)}, \lambda^{(0)})\mathtt{Z}_1$;
    $\mathtt{k} := 0$;
    while ($\|\mathtt{G}\| > \mathtt{tolRes}$) do
        $\mathtt{v}^{(\mathtt{k}+1)} = \mathtt{RK}(\Delta\mathtt{t}, \mathtt{p}^{(\mathtt{k})}, \mathtt{v}^{(\mathtt{k})}, \lambda^{(\mathtt{k})}, \mathtt{tol})$;
        $\begin{pmatrix} p^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} = \begin{pmatrix} p^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \begin{pmatrix} H & Z_1^T G_\lambda \\ \dot{p} & \dot{\lambda} \end{pmatrix}^{-1} \begin{pmatrix} Z_1^T G \\ 0 \end{pmatrix}$;
        $\mathtt{u}^{(\mathtt{k}+1)} = \mathtt{Z}_1\mathtt{p}^{(\mathtt{k}+1)} + \mathtt{v}^{(\mathtt{k}+1)}$;
        $\mathtt{G} = \mathtt{G}(\mathtt{u}^{(\mathtt{k}+1)}, \lambda^{(\mathtt{k}+1)})$;
        $\mathtt{H} = \mathtt{Z}_1^{\mathtt{T}}\mathtt{G}_{\mathtt{u}}(\mathtt{u}^{(\mathtt{k}+1)}, \lambda^{(\mathtt{k}+1)})\mathtt{Z}_1$;
        $\mathtt{k} = \mathtt{k} + 1$;
    endwhile;
    CorrectBasisViaCayley($\delta, \alpha_1, \alpha_2$);
end
```

In order to approximate the flow ψ we have employed a standard Runge–Kutta integrator with automatic step control:

$$\psi(\Delta t, q; p, \lambda) \approx RK(\Delta t, p, q, \lambda, tol).$$

The last parameter sent to the Runge–Kutta procedure is the local discretization error tolerance *tol*. The following table lists parameters used in the above algorithm together with values we used in the computations:

$\mathtt{tolRes} = 10^{-4}$	— used as a stopping criterion corrector.
$\Delta\mathtt{t} = 0.1$	— time step in the dynamical simulation
$\delta = -5.0$	— line in the complex plain to be tested for transitions of eigenvalues
$\alpha_1 = \delta + 1.4$	— see section 2.2
$\alpha_2 = \delta - 1.4$	— see section 2.2

3 Convergence Analysis

In this section we will provide a proof of local convergence of the iterations defined in section 2. As a preparation we investigate properties of matrix $A^Q \in \mathbb{R}^{(n-m)\times(n-m)}$ defined as a projection of A on $\mathbb{R}^{n-m}$,

$$A^Q = Z_2^T A Z_2.$$

An analogy to the lemma from [10] is stated and proved.

Lemma 1. *Matrix A^Q has $n-m$ eigenvalues $\mu_{m+1}, \dots, \mu_n$.*

Proof. Let the columns of matrix $\tilde{Z}_2 \in \mathbb{R}^{n\times(n-m)}$ form a basis in invariant subspace of A corresponding to the last $n-m$ eigenvalues, i.e. $A\tilde{Z}_2 = \tilde{Z}_2\Lambda_2$ with $\sigma(\Lambda_2) = \{\mu_{m+1}, \dots, \mu_n\}$. Note that columns of both Z_1 and $\tilde{Z}_2$ together form a basis of $\mathbb{R}^n$ and hence

$$\mathbb{R}^n = \mathbb{P} \oplus \tilde{Z}_2(\mathbb{R}^{n-m}) \tag{15}$$

As a next step we will show that columns of the matrix

$$V_2 = Z_2^T \tilde{Z}_2 \in \mathbb{R}^{(n-m)\times(n-m)}$$

span an invariant subspace of A^Q. Indeed,

$$A^Q V_2 = Z_2^T A Z_2 Z_2^T \tilde{Z}_2 = Z_2^T A \left(Z_1 Z_1^T + Z_2 Z_2^T\right) \tilde{Z}_2 = Z_2^T A \tilde{Z}_2 = V_2 \Lambda_2$$

Here we used the fact that $Z_2^T A Z_1 = Z_2^T Z_1 \Lambda_1 = 0$ and $Z_1 Z_1^T + Z_2 Z_2^T = I$. The latter identity is a simple consequence of the decomposition $\mathbb{R}^n = Z_1(\mathbb{R}^m) \oplus Z_2(\mathbb{R}^{n-m})$. To finish the proof it suffices to show that columns of V_2 are linearly independent. Let us suppose that there exists a vector $0 \neq q \in \mathbb{R}^{n-m}$ such that $V_2 q = Z_2^T \tilde{Z}_2 q = 0$. Hence $\tilde{Z}_2 q \perp \mathbb{Q}$ and thus $\tilde{Z}_2 q \in \mathbb{P}$. At the same time $\tilde{Z}_2 q \in \tilde{Z}_2(\mathbb{R}^{n-m})$ which implies $\tilde{Z}_2 q = 0$ due to (15. On the other hand $\tilde{Z}_2 q \neq 0$ since $\tilde{Z}_2$ is a basis. This is a contradiction. ♢

Before we prove the main result of this section we will need to evaluate three differentials. This is done in the following three lemmas.

Lemma 2. *Let G be from the class $\mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$. Then*

$$D_p F^P(p^*, q^*, \lambda^*) = 0 \in \mathbb{R}^{m\times m}.$$

Proof. It follows immediately from the definition of F^P and from the fact that $G^P(p^*, q^*, \lambda^*) = 0$. Really, differentiating F^Q with respect to p we obtain

$$D_p F^Q = I - D_p[(D_p G^P)^{-1}]G^P - (D_p G^P)^{-1} D_p G^P = -D_p[(D_p G^P)^{-1}]G^P.$$

$\diamondsuit$

Lemma 3. *Let G be from the class $\mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$. Then*

$$D_p F^Q(p^*, q^*, \lambda^*) = 0 \in \mathbb{R}^{(n-m)\times m}.$$

Proof. According to the definition of the mapping F^Q we shall compute derivative of the flow ψ with respect to the parameter p. Under the conditions on the smoothness of the mapping G the existence of the differential with respect to p is ensured, see [8], Theorem 14.2.1. Moreover, let $\delta p \in \mathbb{R}^m$ be an arbitrary vector and let $v(t)$ denote the directional derivative

$$v(t) = D_p F^Q(p, q(t), \lambda) \cdot \delta p.$$

Then according to [8], eq. (2.8), (2.9) on page 245, $v(t)$ satisfies the following differential equation

$$\begin{aligned} \dot{v}(t) &= D_q G^Q(p, q(t), \lambda) v(t) + D_p G^Q(p, q(t), \lambda) \cdot \delta p \\ v(0) &= 0. \end{aligned}$$

When we make all the computations in the point (p^*, q^*, λ^*) we obtain $q(t) \equiv q^*$ and $v(t)$ must satisfy differential equation

$$\begin{aligned} \dot{v}(t) &= Z_2^T A Z_2 v(t) + Z_2^T A Z_1 \delta p = Z_2^T A Z_2 v(t) && (16) \\ v(0) &= 0. && (17) \end{aligned}$$

Clearly the only solution of the above system is $v(t) = 0$. $\diamondsuit$

Lemma 4. *Let G be from the class $\mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$. Then*

$$D_q F^Q(p^*, q^*, \lambda^*) = e^{Z_2^T A Z_2 \Delta t}$$

Proof. According to the definition of the mapping F^Q we shall compute derivative of the flow ψ with respect to the initial data. Its existence is assured by the conditions on the smoothness of G, see [8], Theorem 13.1.4. Let δq be any vector from $\mathbb{R}^{n-m}$ and let $z(t)$ denote the directional derivative

$$z(t) = D_q F^Q(p, q(t), \lambda) \cdot \delta q.$$

The so called *equation in variations* can be assembled according to [8], eq. (1.11) on page 232. It is a differential equation for $z(t)$:

$$\begin{aligned} \dot{z}(t) &= D_q F^Q(p, q(t), \lambda) z(t) \\ z(0) &= \delta q. \end{aligned}$$

When we make all the computations in the point (p^*, q^*, λ^*) we obtain $q(t) \equiv q^*$ and

$$\begin{aligned} \dot{z}(t) &= Z_2^T A Z_2 z(t) && (18) \\ z(0) &= \delta q. && (19) \end{aligned}$$

Hence we can deduce $z(t) = e^{Z_2^T A Z_2 t}\delta q.$ ♢

Theorem. *Let G be from the class $\mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$. Then iterations (12), (13) are locally convergent with*

$$\lim_{k\to\infty} p^{(k)} = p^*, \qquad \lim_{k\to\infty} q^{(k)} = q^*.$$

Proof. Let us consider a composite mapping $H : \mathbb{R}^m \times \mathbb{R}^{n-m} \longrightarrow \mathbb{R}^m \times \mathbb{R}^{n-m}$ defined as

$$H(p, q) = \begin{pmatrix} F^P(p, q) \\ F^Q(p, q). \end{pmatrix}$$

To prove the convergence of the iteration process it is sufficient to show that the spectral radius satisfies the condition

$$\rho(D_{p,q}H(p^*, q^*)) < 1. \qquad (20)$$

For the differential $D_{p,q}H(p^*, q^*)$ we obtain

$$D_{p,q}H(p^*, q^*) = \begin{pmatrix} 0 & D_q F^P(p^*, q^*) \\ 0 & e^{Z_2^T A Z_2 \Delta t} \end{pmatrix}$$

due to lemmas 2, 3 and 4. Lemma 1 yields that the eigenvalues of the Jacobian matrix of H are $0, \ldots, 0, e^{\mu_{m+1}}, \ldots, e^{\mu_n}$. There are m zeroes at the beginning of the list. Since

$$\Re\mu_i < 0, \qquad i = m+1, \ldots, n$$

we get $e^{\mu_i} < 1$ and thus (20) holds. ♢

4 Numerical experiment

In order to compare the performance of both the original RPM and the Projected RPM we have chosen the same example as in [10], i.e. the non-symmetric system of PDEs

$$\begin{aligned} \frac{\partial v}{\partial t} &= \frac{1}{5}\frac{\partial^2 v}{\partial x^2} + \lambda(v - w) + \frac{1}{5}\lambda^2 e^v \\ \frac{\partial w}{\partial t} &= \frac{1}{5}\frac{\partial^2 w}{\partial x^2} + \lambda(v + w) + \frac{1}{5}\lambda^2 e^w \end{aligned}$$

λ	$\|u\|$	$\hat{\mu}$	$\lvert\frac{\mu-\hat{\mu}}{\mu}\rvert$
1.3918	1.6824	$-0.1566 + 1.3861$i	$1.5{\cdot}10^{-9}$
1.5456	1.903	0.0767+1.5363i	$2.5{\cdot}10^{-9}$
4.4477	3.4694	-0.2733+4.3172i	$1.4{\cdot}10^{-7}$
4.5973	3.5427	0.0821+4.4481i	$4.8{\cdot}10^{-7}$
7.0235	7.1372	-0.2391	$1.1{\cdot}10^{-4}$
7.0230	7.2211	0.8011	$1.5{\cdot}10^{-4}$
6.6935	9.2843	0.6832	$1.2{\cdot}10^{-4}$
6.5597	9.7910	-0.0401	$4{\cdot}10^{-5}$

Table 1: Accuracy of the computed eigenvalues near bifurcation points

with

$$v(0) = v(1) = 0, \qquad w(0) = w(1) = 0.$$

The above system is discretized by finite differences. A number of mesh points is denoted by nx. Notice that the dimension of the resulting problem is $n = 2{*}nx$. Fig. 1 depicts the solution curve computed by Projected RPM for $nx = 40$. All four bifurcation points were detected along the path — two Hopf bifurcation points, one turning point and one symmetry breaking bifurcation point. Table 1 shows numerical values of computed eigenvalues $\hat{\mu}$ together with relative errors (for comparison we computed the whole spectrum by QR method). From the table we can conclude that all the eigenvalues needed for the detection of bifurcation points are computed within satisfactory accuracy. Note that the original RPM missed to detect the Hopf bifurcation point for $\lambda \approx 4.5$.

Finally, Fig. 2 shows eigenvalue movie, i.e. dependence of real parts of rightmost eigenvalues on the pseudo-arclength parameter. Lines in bold correspond to the complex conjugate pairs of eigenvalues.

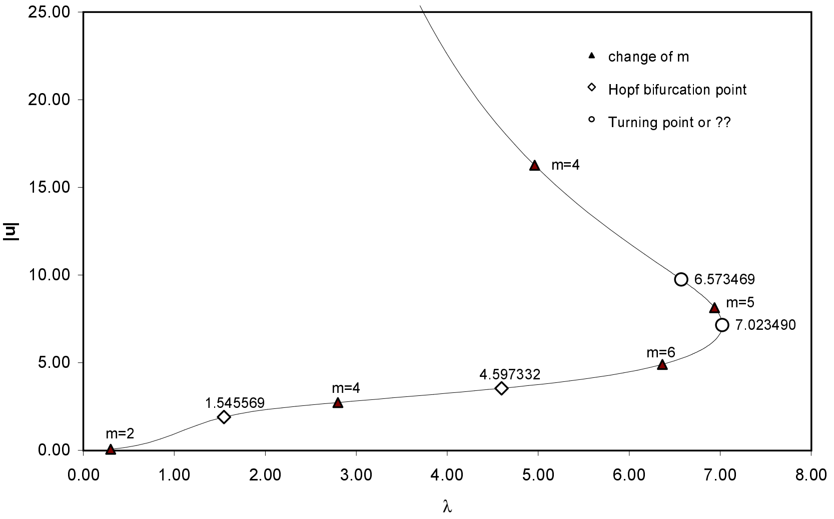

Figure 1: Solution path of nonsymmetric system in for nx=40 computed by the Projected RPM

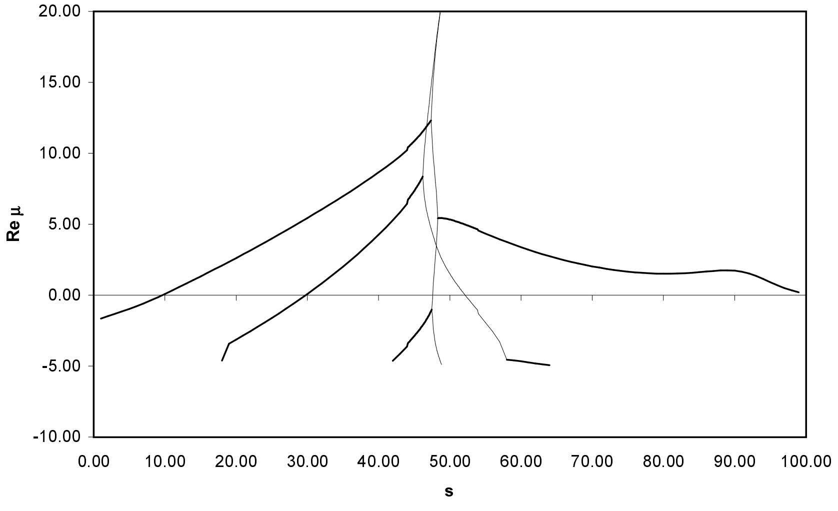

Figure 2: Eigenvalue movie

References

[1] Allgower E. L.; Georg K.: *Numerical continuation methods.* Springer Verlag, New York 1990.

[2] Bošek J.; Janovský V.: *A note on the recursive projection method.* Proceedings of GAMM96, ZAMM 1997, Berlin, 437-440.

[3] Davidson B. D.: *Large-scale continuation and numerical bifurcation for partial differential equations.* SIAM J. Numer. Anal., 34(5)(1997), 2001–2027.

[4] Garratt T. J.; Moore G.; Spence A.: *A generalised Cayley transform for the numerical detection of Hopf bifurcation points in large systems.* Contributions in numerical mathematics, World Sci. Publ. (1993), 177–195.

[5] Janovský V.; Liberda O.: *Recursive Projection Method for detecting bifurcation points* . Proceedings SANM'99, Union of Czech Mathematicians and Physicists (1999), 121–124.

[6] Janovský V.; Liberda O.: *Projected version of the Recursive Projection Method algorithm.* Proceedings of 3^{rd} Scientific Colloqium, Institute of Chemical Technology, Prague, (2001), 89–100.

[7] Kubíček M.; Marek M.: *Computational Methods in Bifurcation Theory and Dissipative Structures.* Springer 1983.

[8] Kurzweil J.: *Ordinary Differential Equations.* Elsevier 1986.

[9] Lust K.; Roose D.: *Computation and bifurcation analysis of periodic solutions of large–scale systems.* IMA Preprint Series #1536, Feb 1998, IMA, University of Minnesota.

[10] Shroff G. M.; Keller H. B.: *Stabilization of unstable procedures: the Recursive Projection Method.* SIAM J. Numer. Anal., 30(4)(1993), 1099–1120.

Acknoledgments: The authors gratefully acknowledge the support of grant MSM 113200007.

Address: Vladimír Janovský, Ondřej Liberda
Faculty of Mathematics and Physics, Charles University
Sokolovská 83
186 00 Prague 8
Czech Republic
janovsky@karlin.mff.cuni.cz
liberda@karlin.mff.cuni.cz

Indication of a stability loss in the Continuation of Invariant Subspaces

Ondřej Liberda, Vladimír Janovský

Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Abstract

Bifurcation analysis of dynamical systems is of importance in many applications. It consists in testing the spectrum of Jacobian matrices. For large dimensional systems, however, the computation of the whole spectrum is not possible. Therefore continuation of the rightmost eigenspaces comes into question. Computation of an arbitrary eigenspace can be carried out with the help of Continuation of Invariant Subspaces (CIS) algorithm. A single eigenvalue or a complex conjugate pair can become unstable during the continuation and it should be added to the eigenspace being continuated. Subspace iterations performed on the Cayley transform of the Jacobian matrix serve as an indicator of this situation.

Key words: bifurcation analysis, pathfollowing, continuation of invariant subspaces

1 Introduction

We consider a system of equations

$$G(u,\lambda) = 0 \tag{1}$$

where $G(u,\lambda) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a nonlinear mapping depending on a real parameter. In practice n can be fairly large since $G(u,\lambda)$ often arises from the spatial dicretization of PDE. The aim is to find the path of solutions and to detect the bifurcation points.

Email address: liberda@karlin.mff.cuni.cz (Ondřej Liberda, Vladimír Janovský).

Preprint submitted to Elsevier Science *8 June 2006*

Standard predictor–corrector methods are widely used for continuation of steady states. For reference see [1]. As far as the bifurcation analysis is concerned all the information is available from the spectrum of the Jacobian matrix

$$A(u, \lambda) = D_u G(u, \lambda).$$

Namely transitions of the eigenvalues through the imaginary axis indicate the stability exchange.

Since the evaluation of the whole spectrum of $A(u, \lambda)$ is not possible for large scale problems we would like to compute the right-most part of spectrum which contains all the unstable and central modes. In [2] an algorithm for continuation of an m-dimensional eigenspace was proposed. It is called Continuation of Invariant Subspaces and we will use its abbreviation CIS in the following text. CIS is based on the application of predictor-corrector techniques to the pair of matrices $\Phi(u, \lambda) \in \mathbb{R}^{n \times m}$ and $\Lambda(u, \lambda) \in \mathbb{R}^{m \times m}$ which appear in the definition of invariant subspace:

$$A(u, \lambda)\Phi(u, \lambda) = \Phi(u, \lambda)\Lambda(u, \lambda). \tag{2}$$

As an alternative to CIS let us mention the Recursive Projection Method (RPM) introduced in [7]. Various improvements of the RPM are developed in [5], [6].

However, neither of the algorithms above gives us guarantee that $\Phi(u, \lambda)$ contains the unstable and central invariant subspace. Moreover we actually need to capture the eigenvalues $\mu \in \sigma(A)$ with $\Re\mu > -\delta$ for a $\delta > 0$ too. This is because these eigenvalues are candidates for transitions through the imaginary axis and that's why they have to be detected in advance. The aim of this paper is therefore to *update* the dimension of the basis $\Phi(u, \lambda)$. To this end we employ subspace iterations applied on $\mathcal{C}(A)$ which is the so called Cayley transform of matrix A. This algorithm described in [3] computes rightmost eigenvalues and corresponding eigenvectors of A. In order to incorporate this method to CIS we have to modify it slightly. The modification consists in suitable projection of $A(u, \lambda)$. The motivation is to exclude the known modes $\mu_1, \ldots, \mu_m$ from the computation. For entirety let us mention that we tested an analogical combination of RPM and subspace iterations with Cayley transform. It is discussed in [6].

The outline of the paper is as follows. Section 2 describes CIS algorithm. At first predictor and corrector steps are defined. They both lead to the Sylvester equation bordered by additional constrints. The so called Bartels-Stewart algorithm transforms it to the sequence of linear systems. At the end of section

2 methods for effective solving of the linear systems are suggested. They take into account the sparsity of matrix A.

Computation of the rightmost eigenvalues of A is the subject of section 3. First subspace iterations and Cayley transform are recalled. Description of the suitable projection which cuts off the known eigenvalues follows. A block scheme of the resulting algorithm encloses this section.

Finally the end of the paper is dedicated to the numerical experiments.

2 Continuation of invariant subspaces

Let us suppose that a parametrized system of square matrices $A(s) \in \mathbb{R}^{n\times n}$ is given. Typically, s is an arc-length parameter and $A(s)$ are obtained as Jacobian matrices $G_u(u(s), \lambda(s))$. Let $\Phi(s) \in \mathbb{R}^{n\times m}$ be a basis in an invariant subspace of $A(s)$ depending smoothly on s. The dimension m is supposed to be a relatively small integer (we did not get over the value 10 in the experiments). We conclude from the definition of the invariant subspace that there exists a smoothly parametrized system of matrices $\Lambda(s) \in \mathbb{R}^{m\times m}$ satisfying

$$A(s)\Phi(s) = \Phi(s)\Lambda(s). \tag{3}$$

Eigenvalues of matrix $\Lambda(s)$ are exactly the eigenvalues of $A(s)$ corresponding to the eigenvectors from $\Phi(s)$. Since m is small we can use methods such as QR-algorithm that compute the whole spectrum of $\Lambda(s)$. Thus the bifurcation analysis is easy once the class of matrices $\Lambda(s)$ is computed. Moreover, if a bifurcation point is detected then $\Phi(s)$ provides information about the direction in which a new branch of steady states emanates.

It is easily seen that (3) represents mn equations for $mn + mm$ unknowns. Therefore mm extra equations have to be added. A natural choice is to demand on an orthonormal basis $\Phi(s)$, i.e. to consider the equation $\Phi^T(s)\Phi(s) = I_m$ where I_m denotes the $m \times m$ identity matrix. However, from the reason for simplicity it is preferable to consider a constraint

$$\hat{\Phi}^T\Phi(s) = I_m \tag{4}$$

with $\hat{\Phi}^T \in \mathbb{R}^{n\times m}$.

2.1 Predictor-Corrector method

In [2], predictor-corrector techniques are applied to the equations (3), (4). Let us suppose that

$$\Phi_0 = \Phi(s_0), \qquad \Lambda_0 = \Lambda(s_0)$$

have already been evaluated for some s_0. By differentiating (3), (4) in s_0 we obtain equations for the tangents $\Phi'(s_0)$, $\Lambda'(s_0)$:

$$\begin{pmatrix} A(s_0)\Phi'(s_0) - \Phi'(s_0)\Lambda_0 - \Phi_0\Lambda'(s_0) \\ \hat{\Phi}^T\Phi'(s_0) \end{pmatrix} = \begin{pmatrix} -A'(s_0)\Phi_0 \\ 0 \end{pmatrix}. \tag{5}$$

The predictor step consists in solving (5) and in setting

$$\begin{aligned} s &= s_0 + \delta s \\ \Phi^{(0)} &= \Phi_0 + \delta s\, \Phi'(s_0) \\ \Lambda^{(0)} &= \Lambda_0 + \delta s\, \Lambda'(s_0), \end{aligned}$$

where δs is a step-length.

Corrector steps are based on the standard Newton's method applied to equations (3), (4). The $(k+1)$-th approximation $(\Phi^{(k)}, \Lambda^{(k)})$is obtained from $(\Phi^{(k)}, \Lambda^{(k)})$ by solving

$$\begin{pmatrix} A(s)\Phi^{(k+1)} - \Phi^{(k+1)}\Lambda^{(k)} - \Phi^{(k)}\Lambda^{(k+1)} \\ \hat{\Phi}^T\Phi^{(k+1)} \end{pmatrix} = \begin{pmatrix} -\Phi^{(k)}\Lambda^{(k)} \\ I_m \end{pmatrix}. \tag{6}$$

2.2 Bartels-Stewart algorithm

Both systems (5) and (6) are of the same form:

$$AH - H\Lambda - \Phi\Delta = B \tag{7}$$
$$\hat{\Phi}^T H = C. \tag{8}$$

which is the so called bordered Sylvester equation for unknowns $H \in \mathbb{R}^{n\times m}$, $\Delta \in \mathbb{R}^{m\times m}$. Effective method called Bartels-Stewart algorithm is available for solving this matrix equation. It consists in orthogonal transformation of Λ to the upper triangular matrix $\tilde{\Lambda}$,

$$\tilde{\Lambda} = Q^H \Lambda Q, \qquad Q^H Q = I_m.$$

It can be done cheaply (in terms of CPU time) since Λ is a small square matrix. Let us write in the similar fashion

$$\tilde{B} = BQ, \qquad \tilde{C} = CQ, \qquad \tilde{H} = HQ, \qquad \tilde{\Delta} = \Delta Q.$$

Finally, let $\lambda_j = \tilde{\Lambda}_{jj}$ denote the j-th diagonal entry of $\tilde{\Lambda}$, $\tilde{H}_j$ the j-th column of $\tilde{H}$ and $\tilde{\Delta}_j$ the j-th column of $\tilde{\Delta}$. Using the notation above we can assemble linear systems

$$\begin{pmatrix} A - \lambda_j I_m & -\Phi \\ \hat{\Phi}^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{H}_j \\ \tilde{\Delta}_j \end{pmatrix} = \begin{pmatrix} \tilde{B}_j + \sum_{k=1}^{j-1} \tilde{\Lambda}_{kj} \tilde{H}_k \\ \tilde{C}_j \end{pmatrix} \tag{9}$$

with unknowns $\tilde{H}_j$, $\tilde{\Delta}_j$. These can be solved sequentially for $j = 1, \dots m$. According to the standard conventions the sum $\sum k = 10$ is supposed to be zero.

2.3 Block elimination methods

Linear systems (9) have the form

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \tag{10}$$

where M_{11} is almost singular and therefore bordering by M_{12}, M_{21} and M_{22} is essential. On the other hand matrix M_{11} is of the form $A - \lambda I$ which has the same structure as the matrix A. Thus methods for solving (10) that use the sparsity of A are welcome. In [4] the so called Block Elimination Mixed for Wider-bordered systems (BEMW) algorithm is introduced. It consists in converting system (10) to series of systems with one-dimensional bordering

$$\begin{pmatrix} M_{11} & b \\ c^T & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{11}$$

Each of the systems (11) is then solved with the help of one-dimensional Block Elimination Method (BEM) which in turn involves 2 linear systems with matrix A, one linear system with A^T and somewhat vector linear algebra and manipulation. For details see [4]. An important feature of BEMW method is its backward stability provided solvers for systems with matrices A and A^T are backward stable as well. Fortunately it holds for majority of solvers used in practice. Backward stability of BEMW together with well conditioned matrix

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

ensures that the relative error of the computed solution will be small.

3 Updating the dimension

For the purposes of bifurcation analysis we have to ensure that continuated subspace $(\Phi(s), \Lambda(s))$ contains all eigenvalues that cross the imaginery axis. To this aim we employed methods that check for presence of eigenvalues $\mu \in \sigma(A(s))$ with

$$\Re\mu > -\delta \tag{12}$$

for some $\delta > 0$. In the actual computations we used $\delta = 5.0$ in order to detect candidates for passing the imaginary axis in advance.

3.1 Subspace iterations and Cayley transform

Subspace iterations is an algorithm that computes eiganspace corresponding to the eigenvalues with largest moduli. Let $r \ll n$ be a number of eigenvalues we want to compute. If eigenvalues $\mu_k \in \sigma(A)$ satisfy the property

$$\|\mu_1\| \geq \cdots \geq \|\mu_r\| > \|\mu_{r+1}\| \geq \cdots \geq \|\mu_n\|$$

(note the sharp inequality between μ_r and μ_{r+1}) then the scheme

```
do
    $\mathtt{V}^{(\mathtt{k+1})} = \mathtt{AV}^{(\mathtt{k})}$;
    Orthonormalize $\mathtt{V}^{(\mathtt{k+1})}$;
    $\mathtt{k} = \mathtt{k} + \mathtt{1}$;
until convergence;
```

produces sequence of $V^{(k)} \in \mathbb{R}^{n\times r}$. These matrices converge to the eigenspace corresponding to $\mu_1, \dots, \mu_r$ provided initial matrix $V^{(0)}$ contains components in this eigenspace. It is automatically satisfied in computer arithmetic due to round-off errors. Therefore we initialize $V^{(0)}$ randomly.

To find the eigenspace corresponding to the *rightmost* eigenvalues we need to find a transformation of A which maps eigenvalues with largest moduli to the rightmost eigenvalues. The so called Cayely transform possesses such a property. Let $\alpha_1 > \alpha_2$ be chosen so that $A - \alpha_1 I$ is regular. Then we define the Cayley transform $\mathcal{C}(A)$ by

$$\mathcal{C}(A) = (A - \alpha_1 I)^{-1}(A - \alpha_2 I).$$

This rational transform maps eigenvalues $\mu \in \sigma(A)$ to $\Theta \in \sigma(\mathcal{C}(A))$ in a similar way

$$\Theta = \mathcal{C}(\mu) = \frac{\mu - \alpha_2}{\mu - \alpha_1}.$$

In [3] it is shown that the half plane $\Re\mu > \frac{1}{2}(\alpha_1 + \alpha_2)$ ($\Re\mu < \frac{1}{2}(\alpha_1 + \alpha_2)$) is mapped outside (inside) the unit circle, resp. It follows that well chosen α_1, α_2 enable us to detect the presence of eigenvalues with $\Re\mu > -\delta$.

3.2 Projected Cayley transform

Subspace iterations applied on the Cayley transform $\mathcal{C}(A)$ compute r rightmost eigenvalues of A. Not all of them fullfill the condition (12) neccesarily if r is overestimated. But even if we detect an eigenvalue satisfying (12) it is likely that it has already been included in $(\Phi(s), \Lambda(s))$ and therefore there is no need for increasing the dimension m. To overcome this technical difficulty and to improve the numerical properties of the method we suggested to perform a projection in each step of subspace iterations. The projector has the form $I_m - \Phi(s)\Phi^T(s)$ and it cuts off the components of $V^{(k)}$ that lie in the subspace *span*$(\Phi(s))$. The resulting procedure then looks as follows:

```
Initialize $\mathtt{V}^{(0)}$ randomly;
do
    Solve $(\mathtt{A} - \alpha_1 \mathtt{I_m})\mathtt{V}^{(k+1)} = (\mathtt{A} - \alpha_2 \mathtt{I_m})\mathtt{V}^{(k)}$;
    $\mathtt{V}^{(k+1)} = (\mathtt{I_m} - \Phi\Phi^{\mathtt{T}})\mathtt{V}^{(k+1)}$;
    Orthonormalize $\mathtt{V}^{(k+1)}$;
    $\mathtt{k} = \mathtt{k} + 1$;
until convergence;
```

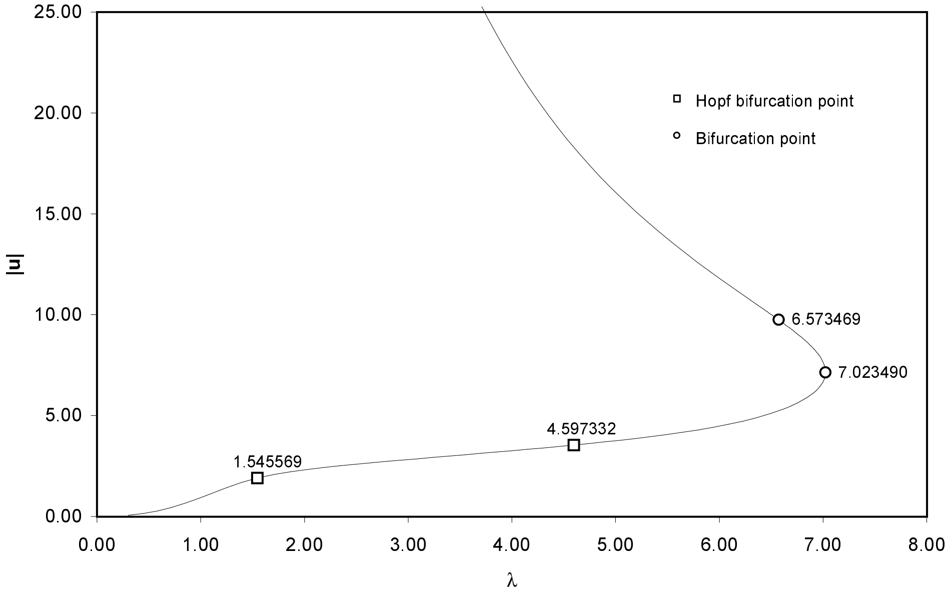

Fig. 1. Solution path

4 Numerical Experiments

In this section we demonstrate the performance of CIS and subspace iterations preconditioned by Cayley transform. We consider a nonsymmetric system of PDEs:

$$\frac{\partial v}{\partial t} = \frac{1}{5}\frac{\partial^2 v}{\partial x^2} + \lambda(v-w) + \frac{1}{5}\lambda^2 e^v$$
$$\frac{\partial w}{\partial t} = \frac{1}{5}\frac{\partial^2 w}{\partial x^2} + \lambda(v+w) + \frac{1}{5}\lambda^2 e^w$$

with

$$v(0) = v(1) = 0, \qquad w(0) = w(1) = 0.$$

The above system is discretized by finite differences. A number of mesh points is denoted by n_x. Fig. 1 depicts the solution curve computed for $n_x = 1000$. Notice that the resulting dimension of the problem was $n = 2000$. Continuation was started with

$$u = (0, \ldots, 0)^T, \qquad \lambda = 0.$$

Initial values for Φ and Λ were derived from the analytical values of eigenvalues

$$\mu_{2k-1} = \mu_{2k} = -\frac{2}{5}\left((n_x+1)\sin\frac{k\pi}{2(n_x+1)}\right)^2, \qquad k = 1,\dots,n_x.$$

for $\lambda = 0$. We chose to set $m = 2$ and to incorporate eigenvalues μ_1, μ_2 into Λ and corresponding eigenvectors (which are available analytically too) into Φ. The next two eigenvalue movies illustrate the effect of employing subspace iterations. Both Fig. 2 and Fig. 3 shows the dependence of real parts of the eigenvalues $\mu_1,\dots,\mu_m$ on the arc-length parameter s. Thick lines are used when a pair of complex conjugate eigenvalues is detected while thin line style is reserved for single real eigenvalues. Eigenvalue movie in Fig. 2 was obtained by CIS in conjunction with modified subspace iterations. We set $\alpha_1 = -3.8$, $\alpha_2 = -6.2$ to check the line $\Re z = -5.0$ in the complex plane. We can see that dimension was increased twice during continuation which enabled us to detect Hopf bifurcation points H1, H2, turning point LP1 and bifurcation point B1. On the contrary Fig. 3 contains eigenvalue movie computed by CIS without checking the line $\Re z = -5.0$. Only one Hopf bifurcation point H1 was detected. Moreover the program crashed when two real eigenvalues collided and gave rise to the complex conjugate pair.

References

[1] E. L. Allgower, K. Georg *Numerical continuation methods*, Springer Verlag, New York, 1990

[2] W. J. Beyn, W. Kless, V. Thümmler *Continuation of low-dimensional invariant subspaces in dynamical systems of large dimension*

[3] T. J. Garratt, G. Moore, A. Spence *A generalised Cayley transform for the numerical detection of Hopf bifurcation points in large systems*, In Contributions in numerical mathematics, pp. 177–195, World Sci. Publ. 1993

[4] W. Govaerts, *Numerical methods for bifurcation of dynamical equilibria*, SIAM, Philadelphia, 2000

[5] V. Janovský, O. Liberda *Projected version of the Recursive Projection Method algorithm*, Proceedings of 3^{rd} Scientific Colloqium, Institute of Chemical Technology, Prague (2001), 89–100

[6] V. Janovský, O. Liberda *Continuation of invariant subspaces via RPM*, in preparation

[7] G. M. Shroff, H. B. Keller, *Stabilization of unstable procedures: the Recursive Projection Method*, SIAM J. Numer. Anal., 30(4)(1993), pp. 1099–1120

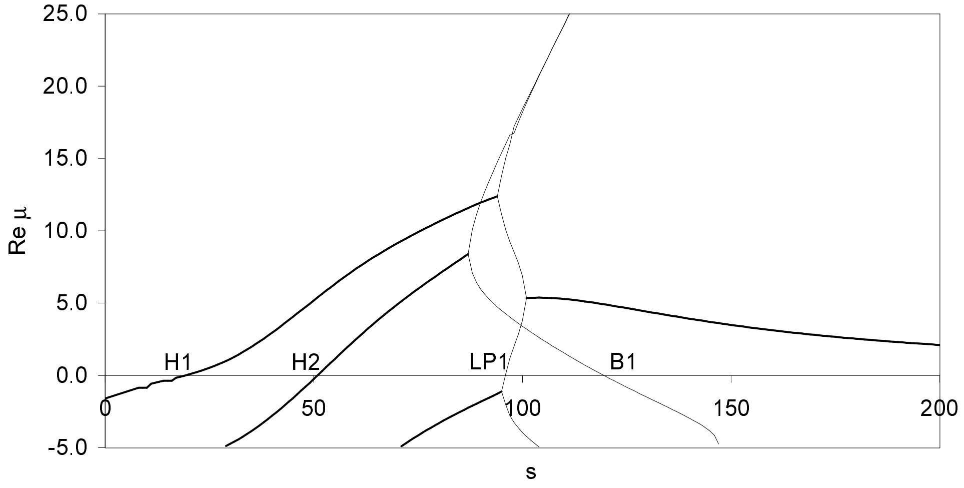

Fig. 2. Eigenvalue movie computed with checking the line $\Re z = -5.0$.

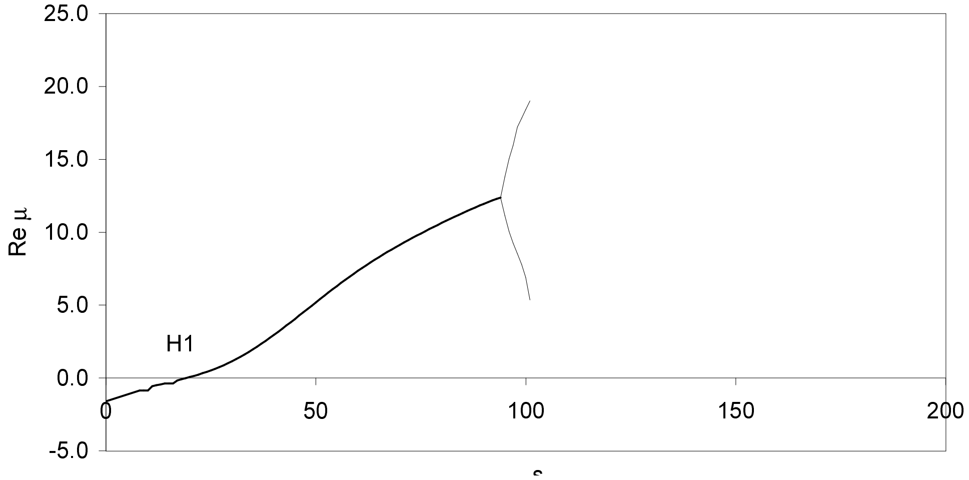

Fig. 3. Eigenvalue movie computed without checking the line $\Re z = -5.0$