BLIS CODE USER'S MANUAL

JOHN M. NEUBERGER, NÁNDOR SIEBEN, AND JAMES W. SWIFT

1. Background

This is a user's manual for the MATLAB code available at GitHub `https://github.com/jwswift/BLIS/`. It approximate solutions to the system of n equations

$$F(u,s)_i := f(u_i,s) - (Mu)_i = 0, \quad i \in \{1,2,\ldots,n\} \tag{1}$$

for the components of $u \in \mathbb{R}^n$, with $f : \mathbb{R}^2 \to \mathbb{R}$ and $F : \mathbb{R}^{2n} \times \mathbb{R} \to \mathbb{R}^{2n}$.

The notation is slightly different from that in the BLIS paper, which is

$$F(s,x)_i := f(s,x_i) - (Mx)_i = 0, \quad i \in \{1,2,\ldots,n\}$$

for the components of $x \in \mathbb{R}^n$, with $f : \mathbb{R}^2 \to \mathbb{R}$ and $F : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. Note that the parameter s is in a different place in the paper and the MATLAB code.

A description of the algorithm is in the BLIS paper. To run the code download `BLIS_2023b.m` (or the updated MATLAB file) and the directories `P3` and `diamond`. Put these all in the same directory, double click on the `.m` file, and say that yes, you want to change the current directory (if it asks).

If you find bugs in the code, please send a note to `Jim.Swift@nau.edu`. I make not guarantees that the code works. Please do not use this program for air traffic control or for running a nuclear power plant.

2. Code

Part of the MATLAB code on the gitHub repository is reproduced below. The parameters chosen here compute a bifurcation diagram very similar to Figure 5 in the paper.

This is the only part of the code that needs to be changed to produce different bifurcation diagrams. The line numbers do not agree with the code you will run. The actual MATLAB code has approximately 34 lines (subject to change) before the first line listed here.

```
% frequently changed parameters
graphName = ['diamond'];

SaveThisRun = false; % saves .txt of Command Window and .pdf of figure.
                     % Answer "y" in command window to really save files.

speed = .1;     % distance between computed points in bifurcation diagram

maxLevel = 14; % Follow daughters to this level. (First branch is level 0.)

plot_bifpoints = true;
plot_foldpoints = false;
plot_branch_as_line = true; % false useful for setting speed

smin = -4; smax = 4.5;        % bifurcation parameter window
ymax = 2;  %default 1000. Needed with asymptotically linear nonlinearity

maxNumBranches  = 50;  %  Default is 50

verbosity       = 1;       % output control: 0 is minimal, 1 gives more
```

2000 *Mathematics Subject Classification.* 34C15, 34C23, 34A34, 37C80.
Key words and phrases. coupled cell networks, synchrony, bifurcation.
August 16, 2023.

```
                               % info about daughters, 2 gives lots of info
print_plist      = false;    % print s y MI newton_its at end of branch

startWithTrivialBranch = true;
if startWithTrivialBranch
    sInit = smin;  % default
    %sInit = 70;  % un-comment this line to set sInit != smin
    usInit = [sInit]; vsInit = [1]; jInit = 1; % trivial branch
    % Needs sInit >= smin with vsInit = [1] (going right).
    % or sInit <= smax with vsInit = [-1] (going left).

else
    %usInit = [0.9003; -1.63; 2]; vsInit = [-1; -1; 3]; jInit = 6; %diamond
    usInit = [1.732; -3]; vsInit = [-1; 3]; jInit = 2; % diamond

    % must be used if f(0,s) != 0.  Start with any approx or exact solution
    % in the invariant subspace W_jInit with dimension d
    % usInit is the (approximate) initial solution [u1; ... ud; s]
    % vsInit is the (approximate) tangent vector [v1; ... vd; s] pointing
    % into the plotting region
end

% The following parameters are more spread out

% You can change the label for the schematic function, using LaTex.
% If you do, search for 'switch y' and change those y strings.
% The value of those y strings needs to agree with the y defined here.

% CHOOSE SCHEMATIC FUNCTION WITH THE NEXT LINE
switch 4 % change integer to get desired schematic function y vs. s
    % don't change the case statements unless you know what you're doing
    % Many schematice functions put two branches on top of each other.
    % Choice "switch 6" makes sure branches don't lie on top of each other.

    % Don't change the numbers below this line!
    case 1 % This defines y in case the line above is "switch 1"
        y = 'u_1';

    case 2
        y = 'u_N';

    case 3
        y = '\frac 1 N \|u\|_2^2';

    case 4
        y = '\frac 1 N \|u\|_1';

    case 5
        y = '\frac 1 {\sqrt N} \|u\|_2';

    case 6   % this one separates branches.
        y = '\displaystyle{\sum_{i=1}^N \sin(i) x_i}';

    case 7
        y = '\|u\|_\infty';
```

```
76
77      case 8
78          y = 'N u_N';
79
80
81  end
82
83  % the derivatives fp = D_1 f and dfds = D_2 f need to be input by hand.
84  % Usually f(u, s) = s*u + nonlinearity(u)
85
86  % CHOOSE NONLINEARITY WITH THE NEXT LINE
87  switch 1 % Change integer and possibly c5 or c2 to get desired nonlinearity
88
89      case 1   % f is odd, this is for "switch 1" in the previous line
90          c5 = 0; % coefficient of degree 5 term.
91          % Constant branch has fold at s = -1/4 * 1/c5 = -1/(4 c5)
92          f    = @(u,s) s*u + u.^3 - c5 * u.^5;
93          fp   = @(u,s) s + 3*u.^2 - 5 * c5 * u.^4;
94          dfds = @(u,s) u;
95
96      case 2  % f is not odd if c2 ~= 0
97          c2 = 0; % coef of degree 2 term.
98          % Constant branch has fold at s = -c2^2/4
99          f    = @(u,s) s*u - u.^3 + c2*u.^2;
100         fp   = @(u,s) s - 3*u.^2 + 2*c2*u;
101         dfds = @(u,s) u;
102
103     case 3   % f(0, s) ~= 0
104         f    = @(u,s) s*u + u.^3 + 1.;
105         fp   = @(u,s) s + 3*u.^2;
106         dfds = @(u,s) u;
107
108     case 4   % df/ds(0,s) ~= u
109         f    = @(u,s) (s*s-1)*u - u.^3;% - 1/2*u.^5;
110         fp   = @(u,s) (s*s-1) - 3*u.^2;%  - 5/2* u.^4;
111         dfds = @(u,s) 2*s*u;
112
113     case 5    % f us asymptotically linear
114         cs = -1;  % This gives vertical asymptote at s = lambda + cs
115         f    = @(u,s) s*u + cs*(tanh(u) - u);
116         fp   = @(u,s) s + cs*(sech(u).^2 -1);
117         dfds = @(u,s) u;
118
119     case 6    % gives non-BLIS bifurcation
120         f    = @(u,s) s*s - u.^2;
121         fp   = @(u,s) -2*u;
122         dfds = @(u,s) 2*s;
123 end
124
125
126 %%%% end of parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- Line 2 indicates the name of a directory (i.e. `diamond/`) that has the 4 files described in the next section. A few such directories are included at the gitHub repository.

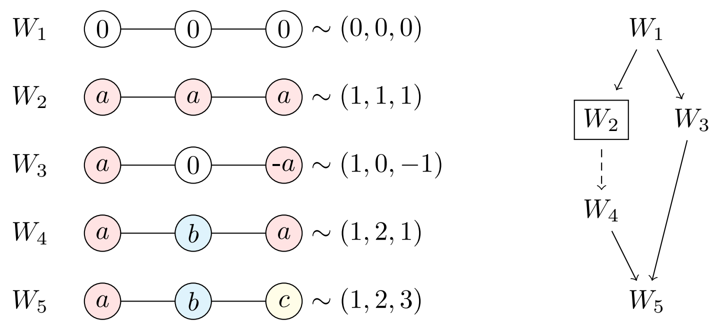

FIGURE 1. The L-invariant subspaces and the lattice of invariant subspaces for the Laplacian matrix of the graph P_3.

- Line 4 allows a txt file and a pdf of the run to be saved. They are saved in the graphName directory with the date and time to the minute. If you do 2 runs in the same minute the first will be overwritten. I usually save these files to a subdirectory called something like `diamond/savedRuns`.
- Line 7 has a big effect on the code. Different graphs work best with different speeds, and it is best to experiment with this. If the speed is too small the program will run slowly. If the speed is too big then the branches will have kinks, and the algorithm might not run correctly.
- Lines 9 through 22 are hopefully self-explanatory.
- Line 24 is usually true, in which case lines 25-41 are not needed.
- Lines 9 through 22 are hopefully self-explanatory.
- Line 40 chooses the schematic function, the y in the y vs. x bifurcation diagram. Note the the default is to label the axis with u and not x. Comments describe how to change that. You can add more schematic functions, as long as the new `y` variable is added in two case statements.
- Line 87 chooses the internal dynamics $f(u, s)$. More cases can be added.

3. Input Files

This section describes the files required as input to our MATLAB program These files are put in a subdirectory that must be in the same directory that contains the MATLAB code `BLIS2023b.m`. For example, the directories `diamond` and `P3` on gitHub each have 4 files, `wam.txt`, `aut.txt`, `inv.txt`, and `bif.txt`, that are automatically read into the program if line 2 in the code listing above is `graphName = ['P3'];` or `graphName = ['diamond'];`

There are various tools available to compute the information in `aut.txt`, `inv.txt`, and `bif.txt` for a given weighted adjacency matrix M, but these are not computed automatically by `BLIS2023b.m`.

We give two examples.

Example 3.1. For simplicity, we present the example of P_3, the path graph with 3 vertices. The graph Laplacian is

$$L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

We assume $M = L$ in Equation (1). The weighted adjacency matrix M, which need not be symmetric, is input using the file `wam.txt` in sparse format with rows of the form i, j, $M_{i,j}$. The automorphism group $\mathrm{Aut}(P_3)$ is the two-element group generated by the permutation (13). The `aut.txt` file lists the elements of $\mathrm{Aut}(P_3)$ in one-line form: $\{(1, 2, 3), (3, 2, 1)\}$. The invariant subspaces are input with the `inv.txt` file using coloring vectors with integer entries, as indicated in Figure 1. The ith row of `inv.txt` gives the coloring vector for W_i. The ordering of the invariant subspaces must satisfy $i \le j$ if $W_i \subseteq W_j$. The lattice of invariant subspaces is supplied to MATLAB with the `bif.txt` file. The ith row of `bif.txt` lists all the j such that $W_i \subseteq W_j$.

wam.txt
```
1 1 1
1 2 -1
2 1 -1
2 2 2
2 3 -1
3 2 -1
3 3 1
```

aut.txt
```
1 2 3
3 2 1
```

inv.txt
```
0 0 0
1 1 1
1 0 -1
1 2 1
1 2 3
```

bif.txt
```
1 2 3 4 5
2 4 5
3 5
4 5
5
```

Note that the trivial subspace is W_1 to accommodate MATLAB, not W_0 which makes more sense to humans. Note that the numbering of the `bif.txt` file in the next example is shifted from the numbering in Figure 5 of the BLIS paper.

Example 3.2. Consider the diamond graph example in the BLIS paper. The four files for the diamond graph follow.

wam.txt
```
1 1 2
1 2 -1
1 4 -1
2 1 -1
2 2 3
2 3 -1
2 4 -1
3 2 -1
3 3 2
3 4 -1
4 1 -1
4 2 -1
4 3 -1
4 4 3
```

aut.txt
```
1 2 3 4
1 4 3 2
3 2 1 4
3 4 1 2
```

inv.txt
```
0 0 0 0
1 1 1 1
1 -1 1 -1
0 1 0 -1
1 0 -1 0
1 2 1 1
1 2 1 2
1 2 -1 -2
1 2 1 3
1 2 3 2
1 2 3 4
```

bif.txt
```
1 2 3 4 5 6 7 8 9 10 11
2 6 7 9 10 11
3 7 9 10 11
4 8 9 11
5 8 10 11
6 9 11
7 9 10 11
8 11
9 11
10 11
11
```

4. Command Window Output

The output of the MATLAB program is a single pdf, with the bifurcation diagram, and the information written to the command window. If line 4 of the listed code says `SaveThisRun = true;`, then these two outputs are written to `datetime.pdf` and `datetime.txt`.

With the original BLIS2023b.m file from gitHub, the following is a fragment of the output printed to the command window, describing the blue branch seen in Figure 5 of the BLIS paper.

```
-------- branch 5 -------
Starting branch 5 at s = 4, y = 0, in invariant subspace W_4 (blue).
Array shows history of branch numbers, with invariant subspace below:
1 5
1 4

MI = 4 initially.
MI -> 3 between s = 3.08 and s = 3 with daughter in W_9.
Bifpoint is at s = 3.000000, y = 0.5
pitchfork bifurcation to W_9:  evec = (0.8165, -0.4082, -0.4082)
MI -> 2 between s = 2.05 and s = 1.96 with daughter in W_8.
Bifpoint is at s = 2.000000, y = 0.707107
pitchfork bifurcation to W_8:  evec = (1.0000, 0.0000)

Last point:  s = -4, y = 1.4142.  uj in W_j and u in R^N follow.
uj = -2.8284
u = (0, -2.8284, 0, 2.8284)
eigenvalues are -18.17 -16.00 6.00 6.17
```

The history of branch numbers means that branch 1 was in W_1, and this was the mother branch for branch 5 which is in W_4. (If you are looking at Figure 5 from the BLIS paper, remember that the numbering of the branches is shifted.)

The program and the output use the term MI for "Morse Index". That is the number of negative eigenvalues of the Jacobian matrix, evaluated at the solution, when M is symmetric. When M is not symmetric, the eigenvalues of the Jacobian might not be real, and MI lists the number of eigenvalues with negative real part.

When the MI changes between two points computed by the `tGNGA`, the new MI is indicated, and the (s, y) coordinates of the bifpoint, as shown on the bifurcation diagram, are computed. The `evec` listed is the critical eigenvector of the Jacobian restricted to W_j, which is the daughter subspace.

At the end of the branch, information about the last point on the branch is printed. The `uj` in `W_j` are the $y \in \mathbb{R}^d$ coordinates for W_j described in the BLIS paper. Then the full coordinates $u \in \mathbb{R}^N$ are printed.

E-mail address: `John.Neuberger@nau.edu, Nandor.Sieben@nau.edu, Jim.Swift@nau.edu`

Department of Mathematics and Statistics, Northern Arizona University PO Box 5717, Flagstaff, AZ 86011-5717, USA