Eigenstructure of the Kobbelt Scheme

Denis Zorin, Stanford University, February 1998

In this worksheet, we explore the eigenstructure of the subdivision matrix for Kobbelt’s subdivision scheme. We compute guaranteed intervals for the magnitude of the largest eigenvalue, and formulas for computing corresponding eigenvector. The result is a file with three interval arithemtics C functions computing the magnitude of the largest eigenvalue for a large range of valences, and two functions to compute eigenvectors for the eigenvalue with the largest magnitude. In addition, we estimate the range of eigenvalues for large valences, which allows us to analyze C1-continuity for all valences.

Utilities

Subdivision matrix of the Kobbelt scheme

Define the blocks of the DFT-transformed subdivision matrix; perform some tests to check if the matrix was defined correctly. We use the following parameters:

\[ \alpha \] and \[ \beta \] are the coefficients of the 4-point scheme (we consider the case when the coefficients are \(9/16\) and \(-1/16\) respectively), \[ \alpha = \cos \left( \frac{2\pi}{K} \right) \] and \[ \omega = e^{\frac{2\pi i}{K}} \].

We test the correctness of the matrix in two ways: first, we compute a submatrix explicitly for the case \(K = 4\), and check if the matrices agree; second, compute the eigenvectors and eigenvalues for \(K = 4\); in this case, the matrix has to have eigenvalue 1/2, and the corresponding complex eigenvector should be a part of a regular quadrilateral grid in the complex plane.

```plaintext
Kobbelt := matrix([[
4*alpha+4*beta*d-beta*(1+2*c), 4*beta*d*beta^2/alpha-beta^2*(conjugate(omega)^2+2*c+1)/alpha, beta, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[4*beta*d*alpha*d+alpha^2*(1+omega)-(1+omega)*alpha*beta, 4*beta^2*d-beta^2*(1+2*c)+2*alpha*beta*c+alpha^2, (1+omega)*alpha*beta, beta^2*omega+alpha*beta, beta^2, beta^2*conjugate(omega)+alpha*beta, 0, 0, 0, 0, 0, 0],
[alpha, alpha*w, 0, 0, 0, 0, beta, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[alpha*omega, alpha+beta*omega, 0, beta, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[alpha, 0, alpha, 0, 0, 0, beta, 0, 0, 0, 0, 0, 0],
[beta^2*conjugate(omega)+alpha^2+alpha*beta*omega, alpha*beta*conjugate(omega)+alpha^2, beta^2*omega+alpha^2, alpha^2, alpha*beta, alpha*beta*(1+conjugate(omega)), alpha*beta, alpha*beta, beta^2, 0, 0, beta^2*conjugate(omega)]]);
```

Kobconst := \{ alpha = 9/16, beta = -1/16 \};

KobExpanded := subs(Kobconst , evalm(Kobbelt));
roots, which do not depend on c, and a factor of degree 6. For illustrative purposes, and to guide us in the subsequent derivations, we compute all the roots of the polynomial numerically, and plot the

Compute and factor the characteristic polynomial of the blocks of DFT-transformed subdivision matrix. The resulting polynomial is parameterized by

We have already computed eigenvalues for the 0th block, and we can assume that $d = 0$

\[
\text{KobZeroBlock} := \text{map}( \text{unapply}( \text{subs}( c = 1, 'x'), 'x'), \text{map}( \text{simplify}, \text{map}( \text{evalc}, \text{subs}( \{ d = 0, \omega = 1, c = 0 \}, \text{eval(KobExpanded)}))));
\]

Check agreement with the regular case.

\[
\text{KobRegularManual} := \text{matrix}( [ [ 9/16 - I^2/16, 0, -1/16, 0, 0, 0 ], [ (81/256)*(I+1) - (9/256)*(I^2 - I), 81/256 + (1/256)*I^2, \ldots ] ]);
\]

\[
\text{KobRegularManual} := \begin{bmatrix}
\frac{9}{16} & 0 & -\frac{1}{16} & 0 & 0 & 0 \\
\frac{9}{256} & \frac{81}{256} + \frac{1}{256} & \ldots & \ldots & \ldots & \ldots \\
\frac{9}{256} & \frac{81}{256} & \ldots & \ldots & \ldots & \ldots \\
\frac{81}{256} & \frac{81}{256} & \ldots & \ldots & \ldots & \ldots \\
\end{bmatrix}
\]

Check if the eigenvector for the eigenvalue 1/2 is a regular grid:

\[
\text{eigenvectors( KobRegular )};
\]

\[
\begin{bmatrix}
\frac{5}{8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{81}{1024} & \frac{81}{1024} & \frac{81}{1024} & \frac{81}{1024} & \frac{81}{1024} & \frac{81}{1024} \\
\end{bmatrix}
\]

Eigenvectors of the 0-th block for all values:

\[
\text{KobZeroBlock} := \text{map}( \text{unapply}( \text{subs}( \{ c = 1, 'x', 'x' \}), \text{map}( \text{simplify}, \text{map}( \text{evalc}, \text{subs}( \{ d = 0, \omega = 1, c = 1 \}, \text{eval(KobExpanded)})))));
\]

The largest eigenvalue is 1/4; we will see that the largest eigenvalue of one of the other blocks is always greater than 1/4, and the dominant eigenvalue is never in the 0-th block.

\[
\text{eigenvectors(KobZeroBlock)};
\]

\[
\begin{bmatrix}
\frac{1}{256} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1] \\
\frac{1}{128} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \\
\frac{1}{64} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
\frac{1}{16} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
\frac{1}{4} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
\frac{1}{2} & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
\end{bmatrix}
\]

Characteristic polynomial of the subdivision matrix

Compute and factor the characteristic polynomial of the blocks of DFT-transformed subdivision matrix. The resulting polynomial is parameterized by $\cos \left( \frac{2\pi}{d} \right)$. The polynomial has a number of small roots, which do not depend on $c$, and a factor of degree 6. For illustrative purposes, and to guide us in the subsequent derivations, we compute all the roots of the polynomial numerically, and plot the

\[
\text{KobCharPoly} := \text{subs}( \{ a^3 = a^{(c-1)^2}, a^2 = 1 - c^2, a^4 = (1-c^2)2, \text{factor}( \text{collect}(\text{charpoly}( \text{map}( \text{simplify}, \text{map}( \text{evalc}, \text{subs}( \{ d = 0, \omega = c + I*a \}, \text{eval(KobExpanded))})), \lambda)), \lambda)) \};
\]

\[
\text{KobCharPoly} := \left( \frac{1}{2} + 16 \lambda + 90 \lambda^3 + 576 \lambda^5 + 9126 \lambda^7 + 49152 \lambda^9 + 5376 \lambda^{11} + 448 \lambda^{13} + 983040 \lambda^{15} + 304128 \lambda^{17} - 2928 \lambda^{19} + 1048576 \lambda^{21} \right)(256 \lambda - 1)(1 + 16 \lambda)
\]

\[
\lambda^6 + \lambda^5 + \lambda^4 + \lambda^3 + \lambda^2 + \lambda + 1
\]

\[
\begin{bmatrix}
\frac{255}{1024} & \frac{3}{16} & \frac{1}{256} & 0 & 0 & 0 \\
8192 & 16384 & 2048 & 2048 & 2048 & 2048 \\
\frac{355}{1024} & \frac{297}{16} & \frac{1024}{1024} & \frac{1024}{1024} & \frac{1024}{1024} & \frac{1024}{1024} \\
\frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} \\
\frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} \\
\frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} & \frac{9}{1024} \\
\end{bmatrix}
\]
The roots of the characteristic polynomial of Kobbel’s scheme in general cannot be found explicitly. However, we can obtain enough information about the eigenvalues to verify C1-continuity. We prove that for any m, k, m = 1, k = 1 the largest eigenvalue is real and unique. And that for m = k = 1, the largest eigenvalue is less than the largest eigenvalue of blocks 1 and 1 = 1. We also show that the unique largest eigenvalue is a single eigenvalue in the interval [0.5, 0.613]. For k = 4.

For the value k = 1, eigenvalues are examined separately. The proof is performed in several steps:

1. We show that for c < 0, all roots of the characteristic polynomial P(c, K) are less than 0.51 (actually, they are less than 0.5, but due to numerical nature of our calculations, we have to relax the upper boundary).

2. We show that for any c = 0, there is a unique real root μ in the interval [0.5, 0.613], and the function μ(c) is C1-continuous and increases.

3. We define the characteristic polynomial (that is, divide by the monomial m!) in symbolic form, with μ and ε as indeterminates. Next, we verify that for all m = 5, 6, 7, 8, and corresponding c(μ), all roots of the deflated polynomial are inside the circle of radius 0.5 centered at 0 in the complex plane, that is, have magnitudes less than 0.5 for any 0 < c. Using μ as the primary parameter is important, as μ can be explicitly computed from μ, but not the other way.

As for k > 4, k < 5 < cos(2π/k) the largest eigenvalue cannot possibly correspond to a block, for which cos(2π/m) < 0. From (3), it follows that the largest root has to be the real root μ(c) for some c.

As for any 1 < m, m < K = 1, cos(2π/m) < cos(2π/k) and we have shown (1) that μ(c) increases, and for any c μ(c) is the largest root, we conclude that the largest eigenvalue always corresponds to m = 1, is real, and is the unique eigenvalue in the range [0.5, 0.613].

On steps 1 and 3 we have to show that roots of a polynomial are inside a circle of radius r in the complex plane. This task is similar to the task of establishing stability of a filter with the transfer function 1/(z−c), where z(c) is a polynomial. Such filter is stable, if all roots of the polynomial are inside the unit circle.

A variety of tests exist for this condition; for our purposes, the algebraic Marden-Jury test is convenient. With appropriate rescaling of the variable it can be used to prove that all roots of a polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

Finally, we compute the largest root of the characteristic polynomial numerically for all valences up to some maximum. For each computed root, we verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.

We use maximal value 3000 here, just in case below we see that 1450 is sufficient to require the interval for each computed root. We verify that the precision is at least 1 × 10^−5. We use interval arithmetics to evaluate the polynomial at all roots of the polynomial are inside the circle of any given radius r. As the test requires only a simple algebraic calculation on the coefficients of the polynomial, it can be easily performed for symbolic and interval coefficients.
Deflation
deflate(p, var, rootval) compute the coefficients of the polynomial \( p(x) \) as
it is assumed that \( p \) is divisible by \( x - \zeta \), \( \zeta \) is the name of the
to variable; \( rootval \) is the root.

[deflate := ]
proc(p::polynom, var::name, rootval) local i, dp, r, dp0 := 0; r := lcoeff(p, var); for i from degree(p, var) by -1 to 0 do dp := dp + r*var\(^i\); r := coeff(p, var, i) + rootval\(^i\) od; dp end proc;

Analysis of the eigenvalues

Now we perform steps 1-3 described above.

(1). We show that for \( c < 0.1 \), all roots of the characteristic polynomial \( p(c, \lambda) \) are less than \( 0.51 \) (actually, they are less than \( 0.5 \), but due to numerical nature of our calculations, we have to relax the
upper boundary).

\[
\text{proc}() := \text{MardenJury}(\text{KobFactor6}, \lambda, 51/100):\]
\[
\text{clambda} := \text{inapply}(\text{clambda}, \lambda): \text{clambda}(0.613);\]
\[
\text{simplify}() := \text{subs}(\lambda = 1/2, \text{clambda});\]
\[
\text{csolutions} := \text{[solve}(\text{KobFactor6}, c)\text{:}];\]
\[
\text{all tests passed}\]

(2). We show that for any \( c = 0 \ldots 1 \), there is a unique real root \( \mu \) in the interval \([0.5,0.613]\), and the function \( p(c) \) is \( C^1 \)-continuous and increases.

\[
\text{proc}() := \text{solve}(\text{KobFactor6}, \lambda)\]
\[
\text{0};\]
\[
\text{all tests passed}\]

(3). We "deflate" the characteristic polynomial (that is, divide by the monomial \( \mu \)) in the symbolic form, with \( \mu \) and \( c \) as the indeterminates. Next, we verify that for \( \lambda = 0 \ldots 1 \), and for all
\( \lambda = 0 \ldots 0.613 \), all roots of the deflated polynomial are inside the circle of radius 0.5 centered at 0 in the complex plane, that is, have magnitudes less than \( 0.51 \) for any \( c < 0 \).

Symbolic definition; if we substitute a pair, \( p(c) \) we get the deflated polynomial for a specific value of \( c \).

\[
\text{deflatedKobFactor6} := \text{collect}(\text{expand}(\text{deflate}(\text{KobFactor6}, \lambda, \mu)), \lambda)\]
\[
\text{deflatedKobFactor6};\]
\[
\text{all tests passed}\]

Verify that for all \( c \) in \([0, 0.1]\) and in \([0.5, 0.613]\) deflated polynomial has roots of magnitude < 0.5

\[
\text{TestDeflated} := \text{proc}(\text{start}, \text{tend}, \text{numeric}, \text{local}, \text{symp}\text{ics})\]
\[
l$ocal\text{c}\text{f}, \text{amu}, c, \text{citer, deflatedinterv};\]
\[
\text{global deflatedMultithread, clambdanull};\]
\[
\text{for i from 0 to 5 do}\text{cf}(i) := \text{inapply}(\text{coeff}(\text{deflatedKobFactor6}(\lambda, c, \mu))\text{od};\]
\[
\text{for i from 0 to 5 do} \text{cf}(i) := \text{evalf}(\text{cf}(i)\text{citer, \{amu, c, \text{tend}\})})\text{\text{\text{old}};\]

All tests passed.

Page 4
Finally, we derive the expressions for the complex eigenvector of the largest eigenvalue of the subdivision matrix. We use the fact that the largest eigenvalue has multiplicity 1 and is a real eigenvalue of the first subblock $B_0$ of the subdivision matrix.

**Eigenvectors**

Finally, we derive the expressions for the complex eigenvector of the largest eigenvalue of the subdivision matrix. We use the fact that the largest eigenvalue has multiplicity 1 and is a real eigenvalue of the first subblock $B_0$ of the subdivision matrix.

**First part of the eigenvector**

```maple
> KobBlock00 := subs( s^2 = 1-c^2, map( evalc, subs( [d = 0, omega = c + I*s], submatrix( KobExpanded, 1..6,1..6))));
```
Code generation

In addition, scale imaginary part by by $1/s$

Separate real and complex parts

> subs( { s = 1, c = 0, lambda = 1/2}, evalm( eval(KobEigenvect)));

Verify agreement with the regular case:

> KobEigenvect := vector( [seq( v0[i], i = 1..6), seq( v1[i], i = 1..6)]):

Put together the vector:

> v1 := map(simplify, subs( { s^4 = (1-c^2)^2, s^2 = 1-c^2, s^3 = (1-c^2)*s}, map( simplify, map( evalc, subs( omega = I*s + c, map( simplify, map( evalc, subs( lambda = 1/2, KobExpanded - lambda * &*(), 7..12, 7..12))))) )

The characteristic polynomial of this submatrix is exactly the degree 6 factor of the characteristic polynomial of the whole matrix:

> collect( simplify( subs( { s^3 = s*(1-c^2), s^2 = 1 - c^2}, charpoly(KobBlock00,lambda) - KobFactor6));

Verify agreement with the regular case:

> redBlock00 := submatrix( evalm( KobBlock00 - lambda * &*()), 2..6, 1..6):

Second part, separate real and imaginary parts

Now we compute the second part of the vector:

> KobBlock10 := submatrix( KobExpanded, 7..12, 1..6);

> KobBlock10lambda := submatrix( evalm( KobExpanded - lambda * &*()), 7..12, 7..12);

> v1 := map(simplify, subs( { s^4 = (1-c^2)^2, s^2 = 1-c^2, s^3 = (1-c^2)*s}, map( simplify, map( evalc, subs( omega = I*s + c, map( simplify, map( evalc, subs( lambda = 1/2, KobExpanded - lambda * &*(), 7..12, 7..12))))) )

Put together the vector:

> KobEigenvect := vector( [seq( v0[i], i = 1..6), seq( v1[i], i = 1..6)]):

Verify agreement with the regular case:

> subs( { s = 1, c = 0, lambda = 1/2}, map( simplify, evalm( eval(KobEigenvect)) ));

Separate real and complex parts

> KobEigenvectRe := map( evalc, map( Re, KobEigenvect));

In addition, scale imaginary part by by $1/s$

> KobEigenvectIm := map( simplify, evalm( (1/s) * map( evalc, map( Im, KobEigenvect)))):