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: α and β are the coeffficients of the 4-point scheme (we consider the case when the coefficients are 9/16 and -1/16 respectively), $c = \cos\left(\frac{2\pi}{K}\right)$, and $\omega = \mathbf{e}^{\left(\frac{2\pi}{K}\right)}$.

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 te corresponding complex eigenvector should be a part of a regular quadrilateral grid in the complex plane.

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

$$Kobbelt := \begin{bmatrix}
\alpha + 4\beta d - \beta(1+2c\sim) & 4\frac{d\beta^2}{\alpha} - \frac{\beta^2(\overline{\omega}^2 + 2c\sim + 1)}{\alpha} & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4\beta\alpha d + \alpha^2(1+\omega) - (1+\omega)\alpha\beta & 4\beta^2 d - \beta^2(1+2c\sim) + 2\alpha\beta c\sim + \alpha^2 & (1+\omega)\alpha\beta & \beta^2\omega + \alpha\beta & \beta^2 & \beta^2\overline{\omega} + \alpha\beta & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha & \alpha + \beta\overline{\omega} & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha\omega & \alpha + \beta\omega & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha & 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 \\
\beta^2\overline{\omega} + \alpha^2 + \alpha\beta\omega & \alpha\beta\overline{\omega} + \alpha^2 & \beta^2\omega + \alpha^2 & \alpha^2 & \alpha\beta & \alpha\beta(\overline{\omega}+1) & \alpha\beta & \alpha\beta & \beta^2 & 0 & 0 & \beta^2\overline{\omega} \\
\beta\omega & \alpha & 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 \\
(1+\omega)\alpha\beta & \alpha^2 & (1+\omega)\alpha\beta & \alpha^2 & \alpha^2 & \alpha^2 & \beta^2(1+\omega) & \alpha\beta & \alpha\beta & \beta^2 & \alpha\beta & \alpha\beta \\
\beta & \alpha & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & \beta \\
\alpha\beta + \alpha^2\omega + \beta^2\omega^2 & \alpha^2 + \alpha\beta\omega & \beta^2 + \alpha^2\omega & (1+\omega)\alpha\beta & \alpha\beta & \alpha^2 & \alpha\beta\omega & \beta^2\omega & 0 & 0 & \beta^2 & \alpha\beta
\end{bmatrix}$$

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

$$Kobconst := \{\alpha = \frac{9}{16}, \beta = \frac{-1}{16}\}$$

```
> KobExpanded := subs(Kobconst , evalm(Kobbelt));
```

$$KobExpanded := \begin{bmatrix}
\frac{5}{8}-\frac{1}{4}d+\frac{1}{8}c\sim & \frac{1}{36}d-\frac{1}{144}\bar{\omega}^2-\frac{1}{72}c\sim-\frac{1}{144} & \frac{-1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{9}{64}d+\frac{45}{128}+\frac{45}{128}\omega & \frac{1}{64}d+\frac{5}{16}-\frac{5}{64}c\sim & -\frac{9}{256}-\frac{9}{256}\omega & \frac{1}{256}\omega-\frac{9}{256} & \frac{1}{256} & \frac{1}{256}\bar{\omega}-\frac{9}{256} & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{16} & \frac{9}{16}-\frac{1}{16}\bar{\omega} & 0 & 0 & 0 & \frac{-1}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{16}\omega & \frac{9}{16}-\frac{1}{16}\omega & 0 & \frac{-1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{16} & 0 & \frac{9}{16} & 0 & 0 & 0 & \frac{-1}{16} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{256}\bar{\omega}+\frac{81}{256}-\frac{9}{256}\omega & -\frac{9}{256}\bar{\omega}+\frac{81}{256} & \frac{1}{256}\omega+\frac{81}{256} & \frac{81}{256} & \frac{-9}{256} & -\frac{9}{256}\bar{\omega}-\frac{9}{256} & \frac{-9}{256} & \frac{-9}{256} & \frac{1}{256} & 0 & 0 & \frac{1}{256}\bar{\omega} \\
-\frac{1}{16}\omega & \frac{9}{16} & 0 & \frac{9}{16} & 0 & 0 & 0 & \frac{-1}{16} & 0 & 0 & 0 & 0 \\
-\frac{9}{256}-\frac{9}{256}\omega & \frac{81}{256} & -\frac{9}{256}-\frac{9}{256}\omega & \frac{81}{256} & \frac{81}{256} & \frac{81}{256} & \frac{1}{256}+\frac{1}{256}\omega & \frac{-9}{256} & \frac{-9}{256} & \frac{1}{256} & \frac{-9}{256} & \frac{-9}{256} \\
\frac{-1}{16} & \frac{9}{16} & 0 & 0 & 0 & \frac{9}{16} & 0 & 0 & 0 & 0 & 0 & \frac{-1}{16} \\
-\frac{9}{256}+\frac{81}{256}\omega+\frac{1}{256}\omega^2 & \frac{81}{256}-\frac{9}{256}\omega & \frac{1}{256}+\frac{81}{256}\omega & -\frac{9}{256}-\frac{9}{256}\omega & \frac{-9}{256} & \frac{81}{256} & -\frac{9}{256}\omega & \frac{1}{256}\omega & 0 & 0 & \frac{1}{256} & \frac{-9}{256}
\end{bmatrix}$$

Special case K = 4:

```
> KobRegular := map( unapply( subs( c = 0,x), x),  map( simplify, map( evalc, subs( { d = 0, omega = I, c = 0 },  evalm(KobExpanded)
  )))):
>
```

Manually computed matrix for the regular case

```
> KobRegularManual := 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,
  (-9/256)*(I+1), -9/256 +I/256, 1/256, -9/256 - (1/256)*I],
  [1,0,0,0,0,0], [9/16,9/16 +I/16, 0,0,0, -1/16],[0,1,0,0,0,0],[ 9*I/16, 9/16 - I/16, 0,-1/16,0, 0] ]);
```

$$KobRegularManual := \begin{bmatrix}
\frac{5}{8} & 0 & \frac{-1}{16} & 0 & 0 & 0 \\
\frac{45}{128}+\frac{45}{128}I & \frac{5}{16} & -\frac{9}{256}-\frac{9}{256}I & -\frac{9}{256}+\frac{1}{256}I & \frac{1}{256} & -\frac{9}{256}-\frac{1}{256}I \\
1 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{16} & \frac{9}{16}+\frac{1}{16}I & 0 & 0 & 0 & \frac{-1}{16} \\
0 & 1 & 0 & 0 & 0 & 0 \\
\frac{9}{16}I & \frac{9}{16}-\frac{1}{16}I & 0 & \frac{-1}{16} & 0 & 0
\end{bmatrix}$$

Check agreement with the regular case .

```
> norm(  evalm(submatrix( KobRegular, 1..6,1..6) - KobRegularManual));
```

$$0$$

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

```
> eigenvects( KobRegular );
```

$$\left[\frac{1}{256}, 1, \{[0,0,0,0,0,0,0,0,0,1,0,0]\}\right],\left[\frac{-1}{128}, 1, \{\left[0,0,0,0,0,0,0,\frac{1}{8},1,\frac{27}{8}-\frac{27}{8}I,-I,-\frac{1}{8}I\right]\}\right],\left[\frac{1}{2}, 1, \{[1,1+I,2,2+I,2+2I,1+2I,3,3+I,3+2I,3+3I,2+3I,1+3I]\}\right],$$
$$\left[\frac{-1}{32}, 1, \{\left[0,0,0,0,0,0,0,\frac{1}{2}I,I,\frac{3}{2}+\frac{3}{2}I,1,\frac{1}{2}\right]\}\right],\left[\frac{-1}{16}, 1, \{[0,0,0,0,0,0,0,1,1,1,1+I,I,I]\}\right],\left[\frac{-1}{64}, 2, \{[0,0,0,0,0,0,0,-I,-4I,9-9I,4,1]\}\right],$$
$$\left[\frac{1}{32}, 2, \{[0,1,0,6-2I,32,6+2I,0,18-9I,90-18I,243,90+18I,18+9I]\}\right],$$
$$\left[\frac{1}{8}, 3, \{\left[\frac{1}{8},0,1,\frac{3}{4}-\frac{3}{8}I,0,-\frac{3}{8}+\frac{3}{4}I,\frac{27}{8},3-I,\frac{15}{8}-\frac{5}{4}I,0,-\frac{5}{4}+\frac{15}{8}I,-1+3I\right],\left[0,\frac{1}{8},0,\frac{3}{8}+\frac{1}{8}I,1,\frac{3}{8}-\frac{1}{8}I,0,\frac{3}{4}+\frac{3}{8}I,\frac{15}{8}+\frac{3}{8}I,\frac{27}{8},\frac{15}{8}-\frac{3}{8}I,\frac{3}{4}-\frac{3}{8}I\right]\}\right]$$

Eigenvalues of the 0-th block for all valences:

```
> KobZeroBlock := map( unapply( subs( c = 1,'x'), 'x'),  map( simplify, map( evalc, subs( { d = 1, omega = 1, c = 1 },
  evalm(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.

```
> eigenvals(KobZeroBlock);
```

$$\frac{1}{256}, \frac{-1}{32}, \frac{-1}{128}, \frac{-1}{16}, \frac{-1}{64}, \frac{-1}{64}, \frac{1}{4}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}$$

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 $x = \cos\left(\frac{2\pi}{K}\right)$. 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 magnitudes of the roots. Of course, neither the plot nor the computed values cannot be used in the analysis without additional verification.

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

```
> KobCharpolynom := subs( { s^3 = s*(1-c^2),s^2 = 1 - c^2,s^4 = (1-c^2)^2}, factor( collect(charpoly( map( simplify, map( evalc, subs(
  s^2 = 1-c^2, map( simplify, subs( { d = 0, omega = c + I*s}, eval(KobExpanded))))))),lambda),lambda))):
> KobCharpolynom := factor( map( simplify, KobCharpolynom));
```

$$KobCharpolynom := -\frac{1}{72057594037927936}(32\lambda+1)(128\lambda+1)(1+64\lambda)^2$$
$$(-1+42880\lambda^3+90\lambda+576\lambda^2 c\sim+9216\lambda^4 c\sim+49152\lambda^5 c\sim-18\lambda c\sim-5376\lambda^3 c\sim+448\lambda^3 c\sim^2+983040\lambda^5-304128\lambda^4-2928\lambda^2-1048576\lambda^6)(256\lambda-1)(1+16\lambda)$$

```
> KobFactor6 := KobCharpolynom/ (32*lambda+1)*(128*lambda+1)*(1+64*lambda)^2*(256*lambda-1)*(16*lambda+1)):
> KobFactor6 := collect(KobFactor6/lcoeff(KobFactor6,lambda),lambda);
```

$$KobFactor6 := \lambda^6+\left(-\frac{3}{64}c\sim-\frac{15}{16}\right)\lambda^5+\left(-\frac{9}{1024}c\sim+\frac{297}{1024}\right)\lambda^4+\left(-\frac{335}{8192}-\frac{7}{16384}c\sim^2+\frac{21}{4096}c\sim\right)\lambda^3+\left(-\frac{9}{16384}c\sim+\frac{183}{65536}\right)\lambda^2+\left(-\frac{45}{524288}+\frac{9}{524288}c\sim\right)\lambda+\frac{1}{1048576}$$

Compute the eigenvalues; execution of this statement may take a while.

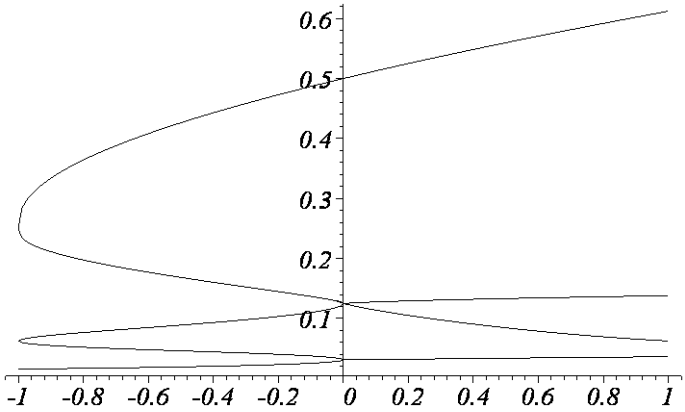
```
> EigenvalsList := seq([solve( subs( c = evalf( (n+1e-10)/100),KobFactor6))], n = -100..100):
```

Convert the lists of eigenvalues to the form suitable for plotting

```
>  EigenvalsPlotLists := seq( [ seq( [-1 + (i-1)/100, abs(op(j, op(i,[EigenvalsList])))], i = 1..201)] ,j=1..6):
```

Plot of the magnitudes of eigenvalues as functions of c; to see approximately the magnitudes of the eigenvalues for a block m of the subdivision matrix for valence K, draw a vertical line at $c = \cos\left(\frac{2\pi m}{K}\right)$ and find where it intersects the curves in the plot.

```
> display(seq(plot(op(i,[EigenvalsPlotLists]),color=black), i = 1..6),color=black, axesfont=[TIMES,ITALIC,10], labels=['','']);
```



Eigenvalues

The roots of the characteristic polynomial of Kobbelt's scheme in general cannot be found explicitly. However, we can obtain enough information about the eigenvalues to verify C1-continuity. We proove that for any $m, k,\ m = 1\ ..\ k-1$ the largest eigenvalue is real and unique, and that for $m \neq k-1$, 1 the largest eigenvalue is less than the largest eigenvalue of blocks 1 and $k-1$. 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 $k = 3$, eigenvalues are examined separately. The proof is performed in several steps:

(1). We show that for c < 0, all roots of the characteristic polynomial $P(c, \lambda)$ 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).

(2). We show that for any $c = 0\ ..\ 1$, there is a unique real root μ in the interval [0.5,0.613], and the function $\mu(c)$ is C1-continuous and increases.

(3). We "deflate" the characteristic polynomial (that is, divide by the monomial $\lambda - \mu$) in symbolic form, with μ and c as indeterminates. Next, we verify that for all $\mu = .5\ ..\ .613$, and corresponding $c(\mu)$, that 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 $\mu(c)$ for any $0 < c$. Using μ as the primary parameter is important, as c can be explicitly computed from μ, but not the other way.

As for $k > 4$, $.51 < \cos\left(\frac{2\pi}{k}\right)$, the largest eigenvalue cannot possibly correspond to a block m, for which $\cos\left(\frac{2m\pi}{k}\right) \leq 0$. From (3), it follows that the largest root has to be the real root $\mu(c)$ for some c.

As for any $1 < m, m < k-1$, $\cos\left(\frac{2m\pi}{k}\right) < \cos\left(\frac{2\pi}{k}\right)$, and we have shown (1) that $\mu(c)$, increases, and for any c $\mu(c)$ 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 $\frac{1}{a(z)}$, where $a(z)$ 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 aproporiate 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 calculaion 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 that the precision is at least $\varepsilon = .1\ 10^{-10}$: we use interval arithmetics to evaluate the polynomial at $\lambda_0 - \varepsilon$ and $\lambda_0 + \varepsilon$ and assert that the sign is guaranteed to change. There may be more than one root: we still have to prove that there is only a single root in the computed interval and that the rest of the roots are smaller. The maximal valence N is chosen in such a way that for N $\cos\left(\frac{2\pi}{N}\right)$ is "sufficiently close" to 1. This means that for all K > N corresponding eigenvalue differs from the limit value λ_∞ by no more than ε, where ε is small enough for us to establish, using interval arithmetics, that the Jacobian of the characteristic map is positive for eigenvectors computed using formulas derived below for all λ in the interval $[\lambda_\infty - \varepsilon, \lambda_\infty]$. The actual computation of the Jacobian and evaluation of the necessary contraction functions is performed in the C part of the code. We use maximal value 3000 here, just in case (below we see that 1450 is sufficient to require the interval for λ with the size of only $.1\ 10^{-5}$.

Marden-Jury test

`MardenJury(a,var,rootrad)` Compute Marden-Jury table for a polynomial **`p`** in variable **`var`**, with variable rescaled by rootrad.
Used to verify that all roots of a polynomial are inside the circle of radius r.

```
> MardenJury := proc(a::polynom, var::name, rootrad)
  local i, k, M, acol, tbl, restable;
      M := degree( a, var );
      for i from 0 to M do tbl[ 0, i ] := coeff( a, var, i )*rootrad^( i – M ) od;
      for i to M do for k from 0 to M – i do tbl[ i, k ] := tbl[ i – 1, 0 ]*tbl[ i – 1, k ] – tbl[ i – 1, M – k – i + 1 ]*tbl[ i – 1, M – i + 1 ] od od;
      for i to M do restable[ i ] := tbl[ i, 0 ] od;
      eval( restable )
  end
```

Interval version of Marden-Jury test

```
> IntervMardenJury := proc(a::polynom( interval ), var::name, rootrad::numeric)
  local i, k, M, acol, tbl, restable;
      M := degree( a, var );
      for i from 0 to M do tbl[ 0, i ] := Interval_times( coeff( a, var, i ), rootrad^( i – M ) ) od;
      for i to M do
          for k from 0 to M – i do tbl[ i, k ] := Interval_add( Interval_times( tbl[ i – 1, 0 ], tbl[ i – 1, k ] ), Interval_times( –1, Interval_times( tbl[ i – 1, M – k – i + 1 ], tbl[ i – 1, M – i + 1 ] ) ) ) od
      od;
      for i to M do restable[ i ] := tbl[ i, 0 ] od;
      eval( restable )
  end
>
```

Deflation

deflate(p,var,rootval) compute the coefficients of the polynomial $\frac{p(z)}{z-z_0}$; it is assumed that p is divisible by $z - z_0$. **var** is the name of the variable, **rootval** is the root.

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

Analysis of the eigenvalues

Now we perform steps 1-3 described above.

(1). We show that for c < 0, 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).

```
> MJtab := MardenJury(KobFactor6, lambda, 51/100):
> MJtabInterv := map(unapply('inapply'( dummy, c ),dummy), MJtab ):
> TestNegativeC := proc(cstart::numeric, cend::numeric, cstep::numeric)
  local MJ, cx;
  global MJtabInterv;
      for cx from cstart by cstep to cend do
          MJ := map( unapply( dummy( [ cx, cx + cstep ] ), dummy ), MJtabInterv );
          if 0 < op( 2, MJ[ 1 ] ) or op( 1, MJ[ 2 ] ) < 0 or op( 1, MJ[ 3 ] ) < 0 or op( 1, MJ[ 4 ] ) < 0 or op( 1, MJ[ 5 ] ) < 0 or op( 1, MJ[ 6 ] ) < 0 then
              ERROR( 'test failed for interval = ', [ cx, cx + cstep ] )
          fi
      od;
      print( 'All tests passed' )
  end
>
```

Do tests, adjusting the step in c. This is not really necessary -- we can simply take the smallest step, but to save time we use larger steps first.

```
> TestNegativeC(-1.0,-0.65,0.05);
```

All tests passed

```
> TestNegativeC(-0.6,-0.275,0.025);
```

All tests passed

```
> TestNegativeC(-0.25,-0.06,0.01);
```

All tests passed

```
> TestNegativeC(-0.05,0.,0.005);
```

All tests passed

(2). We show that for any $c = 0 .. 1$, there is a unique real root μ in the interval [0.5,0.613], and the function $\mu(c)$ is C1-continuous and increases.

Solve the characteritic polynoial for c

```
> csolutions := [solve( KobFactor6, c)]:
```

We are intersted in the first solution only; we will verify later that te second one is out of the range [-1..1] for relevant values of λ.

```
> clambda := csolutions[1];
```

$$clambda := \frac{1}{896}\,\frac{(-6144\,\lambda^3 - 1920\,\lambda^2 + 432\,\lambda - 18 + 2\sqrt{9437184\,\lambda^6 + 13238272\,\lambda^5 - 5451776\,\lambda^4 + 393216\,\lambda^3 + 30784\,\lambda^2 - 3440\,\lambda + 81}\,)\,(-1 + 8\,\lambda)}{\lambda^2}$$

```
>
```

Compute the derivative.

```
> clambdadiff := simplify( diff(clambda, lambda)):
```

The solution for $\lambda = \frac{1}{2}$ can be computed explicitly.

```
> simplify( subs( lambda = 1/2, clambda));
```

$$0$$

The solution for $\lambda = .613$ is outside the range.

```
> clambdaInterv := inapply( clambda, lambda): clambdaInterv(0.613);
```

$$[\,1.007236841,\ 1.007237163\,]$$

```
>
```

Show that the derivative is positive for λ in [0.5..0.613] (the upper bound is the upper estimate for $\lambda(1)$

```
> intervcdiff := inapply( clambdadiff, lambda):
> for xl from .5 by .004 to .613 do res := intervcdiff( [ xl, xl + .004 ] ); if res_1 < 0 then ERROR( test failed for interval, [ xl, xl + .004 ] ) fi od; print( all tests passed )
```

all tests passed

We conclude that $c_1(\lambda)$ increases from 0 to above 1 on [0.5..0.613]; therefore, the inverse increases from 0.5 to approx. 0.613 on [0..1].

The second solution is outside the range of c for this range of λ:

```
> inapply( csolutions[2], lambda)(0.5, 0.613);
```

$$[\,-28.12500027,\ -28.12499976\,]$$

We conclude that for c > 0 in the interval 0.5..0.613 there is a unique real solution.

(3). We "deflate" the characteristic polynomial (that is, divide by the monomial $\lambda - \mu$) in the symbolic form, with μ and c as the indeterminates. Next, we verify that for all $c = 0 .. 1$, and for all $\lambda = .5 .. .613$, 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 $\mu(c)$ for any $0 < c$.

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

```
> deflatedKobFactor6 := collect( expand( deflate(KobFactor6, lambda, mu)), lambda);
```

$$deflatedKobFactor6 := \lambda^5 + \left(-\frac{15}{16} - \frac{3}{64}c + \mu\right)\lambda^4 + \left(-\frac{9}{1024}c + \frac{297}{1024} - \frac{15}{16}\mu - \frac{3}{64}\mu\,c + \mu^2\right)\lambda^3 + \left(\frac{21}{4096}c - \frac{335}{8192} - \frac{7}{16384}c^2 - \frac{9}{1024}\mu\,c + \frac{297}{1024}\mu - \frac{15}{16}\mu^2 - \frac{3}{64}\mu^2\,c + \mu^3\right)\lambda^2$$
$$+ \left(\frac{183}{65536} - \frac{9}{16384}c + \frac{21}{4096}\mu\,c - \frac{335}{8192}\mu - \frac{7}{16384}\mu\,c^2 - \frac{9}{1024}\mu^2\,c + \frac{297}{1024}\mu^2 - \frac{15}{16}\mu^3 - \frac{3}{64}\mu^3\,c + \mu^4\right)\lambda + \frac{9}{524288}c - \frac{45}{524288} - \frac{335}{8192}\mu^2 - \frac{9}{16384}\mu\,c - \frac{9}{1024}\mu^3\,c + \frac{183}{65536}\mu$$
$$+ \frac{297}{1024}\mu^3 + \mu^5 + \frac{21}{4096}\mu^2\,c - \frac{15}{16}\mu^4 - \frac{3}{64}\mu^4\,c - \frac{7}{16384}\mu^2\,c^2$$

Verify that for all c in 0..1 and μ in 0.5.. 0.613 deflated polynomial has roots of magnitude < 0.5

```
> TestDeflated := proc(lstart::numeric, lend::numeric, lstep::numeric)
  local cf, i, MJ, lx, cinterv, deflatedinterv;
  global deflatedMJtabInterv, clambdaInterv;
      for i from 0 to 5 do cf[ i ] := inapply( coeff( deflatedKobFactor6, lambda, i ), c, mu ) od;
      for lx from lstart by lstep to lend do
          cinterv := clambdaInterv( [ lx, lx + lstep ] );
          deflatedinterv := 0;
          for i from 0 to 4 do deflatedinterv := deflatedinterv + eval( cf[ i ]( cinterv, [ lx, lx + lstep ] ) )*lambda^i od;
```

```
        deflatedinterv := deflatedinterv + [1.0, 1.0]*λ^5;
        MJ := IntervMardenJury( deflatedinterv, λ, .5);
        if 0 < op(2, MJ[1]) or op(1, MJ[2]) < 0 or op(1, MJ[3]) < 0 or op(1, MJ[4]) < 0 or op(1, MJ[5]) < 0 or op(1, MJ[6]) < 0 then
            ERROR( 'test failed for interval = ', [lx, lx + lstep])
        fi
    od;
    print( 'All tests passed')
  end
> TestDeflated(0.5, 0.613, 0.0005);
```

All tests passed

We conclude that in the range c = 0..1, all roots of the deflated polynomial have magnitudes less than 0.5

```
>
```

Special case: k = 3

Calculation of the largest eigenvalues with guaranteed precision

This function produces a table of approximate values of the eigenvalue with given precision for use with interval arithmetics in the C part of the analysis code; to avoid conversion problems, we write two integers: mantissa + exponent base 10. The last value is the limit value for infinity (computed with c set to 1 in the char. polynomial). The result is a C function written to a file; if the file name is 'default', then the output is written to the standard output.
The argument of the function is valence, the function returns the interval value for the largest eigenvalue. The body is just a large switch statement.
We assume that the exponents for eigenvalues are nonpositive, which is always the case for Kobbelt's scheme.

```
ComputeEigenvalues := proc(N::integer, eps::numeric, fname::string)
local K, intervKobFactor6, intervPi, intervc, expandedKobFactor6, approxEV, r, deflatedKobFactor6, i, marTable, cK;
global KobFactor6;
    Digits := 15;
    intervKobFactor6 := inapply( KobFactor6, λ, c);
    intervPi := Interval_times( [2.0, 2.0], Interval_arccos( [0, 0]));
    intervc := inapply( cos(2*intervPi / K), K);
    fprintf( fname, 'virtual Float Eigenvalue(int K) {\n');
    fprintf( fname, ' static INTEGER64 EV[] = {\n');
    expandedKobFactor6 := subs( c = 1, KobFactor6);
    approxEV := fsolve( expandedKobFactor6, λ, λ = .5 .. 1);
    if 0 < op( 2, intervKobFactor6( approxEV - eps, 1)) or op( 1, intervKobFactor6( approxEV + eps, 1)) < 0 then ERROR( 'fsolve precision failure for infinity') fi;
    fprintf( fname, 'CONST64(%d),CONST64(%d),CONST64(%d),\n', op( 1, approxEV - eps), op( 1, approxEV + eps), 10^(-op( 2, approxEV)));
    fprintf( fname, 'CONST64(0),CONST64(0),CONST64(0), CONST64(0),CONST64(0),CONST64(0),\n');
    for K from 3 to N do
        if (K - 3) mod 100 = 0 then print( K) fi;
        cK := intervc( K);
        expandedKobFactor6 := subs( c = cos( 2*π / K), KobFactor6);
        approxEV := fsolve( expandedKobFactor6, λ, λ = .25 .. 1);
        if 0 < op( 2, intervKobFactor6( approxEV - eps, cK)) or op( 1, intervKobFactor6( approxEV + eps, cK)) < 0 then ERROR( 'fsolve precision failure for K =', K) fi;
        fprintf( fname, 'CONST64(%d),CONST64(%d),CONST64(%d),\n', op( 1, eval( approxEV - eps)), op( 1, eval( approxEV + eps)), 10^(-op( 2, approxEV)))
    od;
    fprintf( fname, 'CONST64(0)};\n return Float(EV[3*K],EV[3*K+1])/Float(EV[3*K+2]) ;\n}\n\n');
    NULL
end
```

The derivative of the largest eigenvalue with respect to c at infinity.

To establish C1-continuity for all valences, we need to analyze behavior of the magnitude of the largest eigenvalue as the function of the valence, as the valence increases to infinity (c approaches 1). We estimate a constant B, such that $|\lambda - \lambda_\infty| < B|c - 1|$, sufficiently close to 1. This constant can be taken to be the maximum of $\left|\frac{\partial}{\partial c}\lambda\right|$, or, equivalently, as maximum of $\left|\frac{\partial}{\partial \lambda} c\right|^{(-1)}$; as the characteristic polynomial is quadratic in c, the latter is relatively easy to compute. Once B is known, we can estimate the size ε of the interval for λ near λ_∞, such that if the characteristic map is injective and regular for all these values, it is sufficient to establish C1-continuity for $K_0 < K$, where $\frac{\varepsilon}{B} < 1 - \cos\left(\frac{2\pi}{K_0}\right)$.

```
> intervcdiff := inapply( clambdadiff, lambda):
```

Evaluate for all lambda in the range 0.7..1; step 0.001 gives reasonable bounds; this may take some time.

```
> cdiffinterv := [];
  for i from 0 to 299 do
      cdiffinterv :=  Interval_union(cdiffinterv, intervcdiff([0.7+i*0.001, 0.7+(i+1)*0.001]));
  od: eval(cdiffinterv);
```

cdiffinterv := [ ]

[9.984080932, 15.63504120]

```
> B := op(2, Interval_reciprocal( op(1, cdiffinterv)));
```

B := .1001594446

For example, if we use the interval of size .1 10^{-5} for λ_∞, we have to consider all valences up to the valence for which $B\left|\cos\left(\frac{2\pi}{K}\right) - 1\right| < .1\ 10^{-5}$, which turns out to be approx. 1450.

```
> Interval_times( B, Interval_add( Interval_cos( Interval_times( Interval_times( 2, 2*Interval_arccos(0.0)),
  Interval_reciprocal(1450))),-1));
```

[-.9403469458 10^{-6}, -.9403269135 10^{-6}]

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 $B_{0,0}$ of the subdivision matrix.

First part of the eigenvector

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

$$KobBlock00 := \begin{bmatrix} \frac{5}{8}+\frac{1}{8}c & -\frac{1}{72}c^2-\frac{1}{72}c+\frac{1}{72}I\,c\,s & \frac{-1}{16} & 0 & 0 & 0 \\ \frac{45}{128}+\frac{45}{128}c+\frac{45}{128}I\,s & \frac{5}{16}-\frac{5}{64}c & -\frac{9}{256}-\frac{9}{256}c-\frac{9}{256}I\,s & \frac{1}{256}c+\frac{1}{256}I\,s-\frac{9}{256} & \frac{1}{256} & \frac{1}{256}c-\frac{9}{256}-\frac{1}{256}I\,s \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{9}{16} & \frac{9}{16}-\frac{1}{16}c+\frac{1}{16}I\,s & 0 & 0 & 0 & \frac{-1}{16} \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{9}{16}c+\frac{9}{16}I\,s & \frac{9}{16}-\frac{1}{16}c-\frac{1}{16}I\,s & 0 & \frac{-1}{16} & 0 & 0 \end{bmatrix}$$

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))), lambda) - KobFactor6;
```

$$0$$

```
> redBlock00 := submatrix ( evalm( KobBlock00 - lambda *  &*()), 2..6, 1..6):
> v0 := map( simplify, subs( { s^3 = (1-c^2)*s, s^2 = 1-c^2, _t[1] = 1}, linsolve( redBlock00, vector([0,0,0,0,0]) ) ));
```

$$v0 := \left[\lambda,\ 9\,\frac{\lambda\,(2\,c^2\,\lambda-1-c+384\,\lambda^2-I\,s+384\,c\,\lambda^2-2560\,c\,\lambda^3-2560\,\lambda^3+2\,I\,\lambda\,c\,s-2560\,I\,\lambda^3\,s+384\,I\,\lambda^2\,s+2\,c\,\lambda)}{80\,\lambda+4\,c^2\,\lambda-16\,c\,\lambda-2112\,\lambda^2-5120\,c\,\lambda^3-1+576\,c\,\lambda^2-65536\,\lambda^4+20480\,\lambda^3},\ 1,\right.$$
$$9\,\frac{I\,\lambda\,(-9\,s+9\,I\,c-2\,I\,c^2+2\,s\,c-228\,I\,\lambda\,c+1344\,I\,c\,\lambda^2+240\,s\,\lambda-144\,I\,\lambda+4096\,I\,\lambda^3+10\,I-1344\,\lambda^2\,s)}{80\,\lambda+4\,c^2\,\lambda-16\,c\,\lambda-2112\,\lambda^2-5120\,c\,\lambda^3-1+576\,c\,\lambda^2-65536\,\lambda^4+20480\,\lambda^3},$$
$$9\,\frac{2\,c^2\,\lambda-1-c+384\,\lambda^2-I\,s+384\,c\,\lambda^2-2560\,c\,\lambda^3-2560\,\lambda^3+2\,I\,\lambda\,c\,s-2560\,I\,\lambda^3\,s+384\,I\,\lambda^2\,s+2\,c\,\lambda}{80\,\lambda+4\,c^2\,\lambda-16\,c\,\lambda-2112\,\lambda^2-5120\,c\,\lambda^3-1+576\,c\,\lambda^2-65536\,\lambda^4+20480\,\lambda^3},$$
$$\left.-9\,\frac{I\,\lambda\,(-8\,I\,c-4096\,I\,c\,\lambda^3+10\,s-9\,I-144\,s\,\lambda+144\,I\,\lambda\,c+12\,s\,\lambda\,c+4096\,\lambda^3\,s+240\,I\,\lambda-1344\,I\,\lambda^2-12\,I\,\lambda\,c^2)}{80\,\lambda+4\,c^2\,\lambda-16\,c\,\lambda-2112\,\lambda^2-5120\,c\,\lambda^3-1+576\,c\,\lambda^2-65536\,\lambda^4+20480\,\lambda^3}\right]$$

Verify agreement with the regular case:

```
> subs( { s = 1, c = 0, lambda = 1/2}, eval(v0) );
```

$$\left[\frac{1}{2},\ \frac{1}{2}+\frac{1}{2}I,\ 1,\ 1+\frac{1}{2}I,\ 1+I,\ \frac{1}{2}+I\right]$$

Second part, separate real and imaginary parts

Now we compute the second part of the vector:

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

$$KobBlock10 := \begin{bmatrix} \frac{9}{16} & 0 & \frac{9}{16} & 0 & 0 & 0 \\ \frac{1}{256}\bar{\omega}+\frac{81}{256}-\frac{9}{256}\omega & -\frac{9}{256}\bar{\omega}+\frac{81}{256} & \frac{1}{256}\omega+\frac{81}{256} & \frac{81}{256} & \frac{-9}{256} & -\frac{9}{256}-\frac{9}{256}\bar{\omega} \\ -\frac{1}{16}\omega & \frac{9}{16} & 0 & \frac{9}{16} & 0 & 0 \\ -\frac{9}{256}-\frac{9}{256}\omega & \frac{81}{256} & -\frac{9}{256}-\frac{9}{256}\omega & \frac{81}{256} & \frac{81}{256} & \frac{81}{256} \\ \frac{-1}{16} & \frac{9}{16} & 0 & 0 & 0 & \frac{9}{16} \\ -\frac{9}{256}+\frac{81}{256}\omega+\frac{1}{256}\omega^2 & \frac{81}{256}-\frac{9}{256}\omega & \frac{1}{256}+\frac{81}{256}\omega & -\frac{9}{256}-\frac{9}{256}\omega & \frac{-9}{256} & \frac{81}{256} \end{bmatrix}$$

```
> KobBlock11lambda := submatrix ( evalm( KobExpanded - lambda *  &*()), 7..12, 7..12);
```

$$KobBlock11lambda := \begin{bmatrix} -\frac{1}{16}-\lambda & 0 & 0 & 0 & 0 & 0 \\ \frac{-9}{256} & -\frac{9}{256}-\lambda & \frac{1}{256} & 0 & 0 & \frac{1}{256}\bar{\omega} \\ 0 & \frac{-1}{16} & -\lambda & 0 & 0 & 0 \\ \frac{1}{256}+\frac{1}{256}\omega & \frac{-9}{256} & \frac{-9}{256} & \frac{1}{256}-\lambda & \frac{-9}{256} & \frac{-9}{256} \\ 0 & 0 & 0 & 0 & -\lambda & \frac{-1}{16} \\ -\frac{9}{256}\omega & \frac{1}{256}\omega & 0 & 0 & \frac{1}{256} & -\frac{9}{256}-\lambda \end{bmatrix}$$

```
> 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,
  evalm( - inverse(KobBlock11lambda) &* KobBlock10 &* v0))))));
```

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)) ) );
```

$$\left[\frac{1}{2},\ \frac{1}{2}+\frac{1}{2}I,\ 1,\ 1+\frac{1}{2}I,\ 1+I,\ \frac{1}{2}+I,\ \frac{3}{2},\ \frac{3}{2}+\frac{1}{2}I,\ \frac{3}{2}+I,\ \frac{3}{2}+\frac{3}{2}I,\ 1+\frac{3}{2}I,\ \frac{1}{2}+\frac{3}{2}I\right]$$

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)))):
```

Code generation