condition number

May 20, 2020

1 condition number

- we compute the condition number of the Jacobian of the function which maps the coefficients of a polynomial to the zeros
- this is the same as the condition number of the Jacobian of the mapping taking the zeros to the coefficients
- the condition number mesures how much the rounding errors are amplified during the numerical computation

M. Hegland, May 2020

[1]:
```
restart
loadPackage "NumericalAlgebraicGeometry";
```

```
--loading configuration for package "FourTiTwo" from file
/home/hegland/.Macaulay2/init-FourTiTwo.m2
--loading configuration for package "Topcom" from file
/home/hegland/.Macaulay2/init-Topcom.m2

--loading configuration for package "NumericalAlgebraicGeometry" from file
/home/hegland/.Macaulay2/init-NumericalAlgebraicGeometry.m2
--loading configuration for package "PHCpack" from file
/home/hegland/.Macaulay2/init-PHCpack.m2
--loading configuration for package "Bertini" from file
/home/hegland/.Macaulay2/init-Bertini.m2
```

[135]:
```
n = 10  -- degree of polynomial
k = CC
R = k[x_1..x_n]  -- ring for polynomial coefficients
P = R[z]         -- our univariate polynomial

-- generate polynomial from zeros
p = product(gens R, xi-> z-xi)

-- the coefficients (basically +/- elem. symm. polys)
C = take(flatten entries (last coefficients p), {1,n})

--    and the Jacobian  of the map zeros -> coeffs
```

```
J = jacobian matrix{apply(C, ci -> lift(ci,R))};
```

```
                10       10
o135 : Matrix R   <--- R
```

[109]:

```
-- evaluate Jacobian for specific zeros x_i

-- example 1: x_i = i
F = map(k, R, toList(1..n)) -- substitute the xi by actual values
J1 = matrix apply(entries J, rowi -> apply(rowi, Jij -> F(Jij)))
(S,U,V) = SVD J1  -- singular value decomposition
<< "condition number " << S#0/S#(n-1) << endl;
```

```
condition number 2.92405e11
```

[100]:

```
-- example 2: random x_i
F = map(k, R, apply(n, i -> random(1.0)))
J2 = matrix apply(entries J, rowi -> apply(rowi, Jij -> F(Jij)))
(S,U,V) = SVD J2
<< "condition number " << S#0/S#(n-1) << endl;
```

```
condition number 32454500000

singular values {34.799     }
                {1.76424    }
                {.148912    }
                {.0141099   }
                {.000863034 }
                {.000164683 }
                {.0000200284}
                {.000002154 }
                {7.05257e-8 }
                {1.07224e-9 }
```

[90]:

```
-- example 3: Chebyshev points
F = map(k, R, apply(n, i -> cos(2*pi*i/n)))
J3 = matrix apply(entries J, rowi -> apply(rowi, Jij -> F(Jij)))
(S,U,V) = SVD J3
<< "condition number " << S#0/S#(n-1) << endl;
<< "'numerical' condition number " << S#0/S#(n//2) << endl;
<< "singular values " << S << endl;
-- for the numerical condition number we have removed singularities
-- effectively, this only leaves us with half the points
```

```
condition number 2.10032e18

'numerical' condition number 71.2922
```

```
singular values {8.4545     }
                {5.04066    }
                {1.17694    }
                {.573757    }
                {.137873    }
                {.118589    }
                {6.27761e-16}
                {2.29904e-16}
                {6.25754e-17}
                {4.02535e-18}
```

[95]:

```
-- example 4: Chebyshev points without multiple points
F = map(k, R, apply(n, i -> cos(pi*i/n)))
J3 = matrix apply(entries J, rowi -> apply(rowi, Jij -> F(Jij)))
(S,U,V) = SVD J3
<< "condition number " << S#0/S#(n-1) << endl;
<< "singular values " << S << endl;
```

```
condition number 3074.28

singular values {8.82138  }
                {5.21686  }
                {1.2554   }
                {.576363  }
                {.213411  }
                {.105434  }
                {.0486911 }
                {.0220663 }
                {.00880429}
                {.00286942}
```

[128]:

```
-- example 5: random coefficients

RC = CC[w]
-- from earlier notebook ...
c = apply(n+1, j -> random(1.0))
F6 = {sum(n+1, i -> c#i * w^i)}
zs = solveSystem(F6)
xsol = sort(apply(zs, zi->(coordinates zi)#0))
-- end from earlier ...

F = map(k, R, xsol)
J4 = matrix apply(entries J, rowi -> apply(rowi, Jij -> F(Jij)))
(S,U,V) = SVD J4
<< "condition number " << S#0/S#(n-1) << endl;
```

```
condition number 15.5139
```