Nescoil SVD modification

Prashant Valanju, Neil Pomphrey, Art Brooks

NCSX Project Meeting Sept 23-26, 1998

• **Goals**:

1. Reduce “complexity” of coils without significantly reducing the accuracy of plasma reconstruction, i.e., get almost the same accuracy with “better” coils.
2. Find out the “dominant” basis functions for current representation for use in faster coil optimization.

• **Motivations**:

1. Neil has used this technique with some success using the “dipole method” for current potentials.
2. Implementation of SVD in Nescoil is similar. The only difference is the “fourier method” for current potentials used in Nescoil.
3. Increasing the accuracy requires using more fourier modes, but most of them are not essential. SVD may tell us which combinations are important.

Quick review of (small) Nescoil code:

1. Nescoil tries to minimize the normal component of the total B field created on the plasma surface (pls) by known currents (plasma and background coils) and an unknown current distribution on a fixed plasma-enclosing current-carrying surface (ccs).
2. The ccs current distribution is given via a potential: $\bar{j}(u,v) = \hat{n} \times \nabla\varphi(u,v)$, where φ has fourier coeffs

$$\varphi(u,v) = \sum_{m=0,n=-N}^{M,N} \varphi_{mn} \sin(2\pi[mu+nv]) - \frac{I_p}{n_f} v - I_t u$$

3. It uses the least-squares method, i.e., it minimizes

$$\chi^2 = \int_{pls} (\bar{B} \cdot \hat{n})^2 = \sum_{u,v}^{Nu,Nv} (\bar{B} \cdot \hat{n})^2 \text{ by varying } \varphi_{mn} .$$

4. $B_{total} = B_{ccs} + B_{ext}$ is given in terms of φ_{mn} by

$$B(u,v) = h(u,v) + \sum_{m,n} g_{mn}(u,v)\varphi_{mn}$$, where g and h

are available if you look carefully in Nescoil.

5. Since $B \alpha j$ (via Biot-Savart), and $j \alpha \varphi \alpha \varphi_{mn}$, this is a linear least-squares problem.

Solution by direct inversion using SVD:

1. This linear problem can be directly solved for φ_{mn} by trying to zero ($n \cdot B_{total}$) at the (Nu,Nv) array of plasma points, i.e., by directly inverting

   $$G\varphi \equiv \sum_{m,n} g_{mn}(u,v)\varphi_{mn} = B_{ext}(u,v) - h(u,v) \equiv b$$

2. G(Nuv,Nmn) is SVDecomposed into

   U(Nuv,Nmn) • w(Nmn,Nmn) •V(Nmn,Nmn)

3. The solution is $\varphi = \sum_{i=1}^{Nmn} w_i \left(U_{(i)}\, b\right) V_{(i)}$

4. If the number of Nuv points on pls where B•n is to be matched exceeds the number of Nmn variable fourier coeffs φ_{mn} , it has more equations than unknows, else it would be underdetermined.

5. Basis vectors $V_{(i)}$ with the largest SVD weights w_i contribute most to reducing the χ^2.

6. Dropping $V_{(i)}$ with small w_i <u>may</u> reduce the "probable error" in estimated coeffs while only slightly increasing χ^2. If any w_i become very small -> simply drop corresponding basis functions.

Plan of SVD attack on rat's nest for QA:

1. Implement SVD algorithm in small nescoil. Done.

2. Try it out on "rat's nest case" i_14_26

    a. Find the largest small number of m,n which gives acceptably smooth coils (5x5).
    b. Use large number of fourier harmonics mf, nf. (20x20). This will give very small error but very bad contours.
    c. Now start cutting off small weights, i.e., trade low error for smoother contours.
    d. If surface current contours look ok with error low enough for good reconstruction, try making coils and see if error still stays low.

SVD may let us have the cake and eat it too, i.e, may get

    a. the smoothness of small mf,nf, and

    b. small errors of large mf,nf.

Results for QA (attack on rat's nest i_14_26.d20):

nu=64, nv=64, Nuv=2112 points on plasma 1/2+ period.

nf=5,mf=5, Nmn=57; Ave err = 3.18%, max = 70% (bad)

nf=10,mf=10, Nmn=220; Run time 20 sec (SVD/STD)

nf=20,mf=20, Nmn=840; Run time 116 sec cpu.

nf=30,mf=30, Nmn=1860; Run time 10 min cpu.

Fig	**m, n**	**W's cut**	**Ave err%**	**Max err%**	**Comments**
1	5,5	0	3.18	70.0	Target contours, bad error
2	10,10	0	0.95	16.9	Target error, bad contours
3	20,20	0	0.086	3.56	Low error, worse contours
4	20,20	197	0.119	2.45	Still low err, bad contours
5	20,20	433	0.329	3.43	Contours getting better
6	20,20	571	0.662	7.45	Contours not too bad
7	20,20	721	9.71	30.9	Lost it. Error too big
8	10,10	35	1.15	10.0	Not enough modes to play
9	30,30	1199	1.07	15.1	Error a bit high

Max w = 671.5 , Min w = 2.947E-3 , Ratio = 227841.8

Conclusions so far:

1. SVD method has been implemented in Nescoil.

2. Results seem to agree with dipole method.

3. **SVD does better than cutting off fourier series**.

4. Trading error for smoothness works to some degree,

BUT

1. **Can't fully eradicate "rat's nest" and keep error low.**

2. Looking at the potential contours for the largest basis vectors does not give much insight for the QA case.

Rats still rule! But we can get them.

1. Increase mf,nf to 30x30. Takes longer (~10 min) to run.

2. Use non-uniform (u,v) grid to reduce run time.

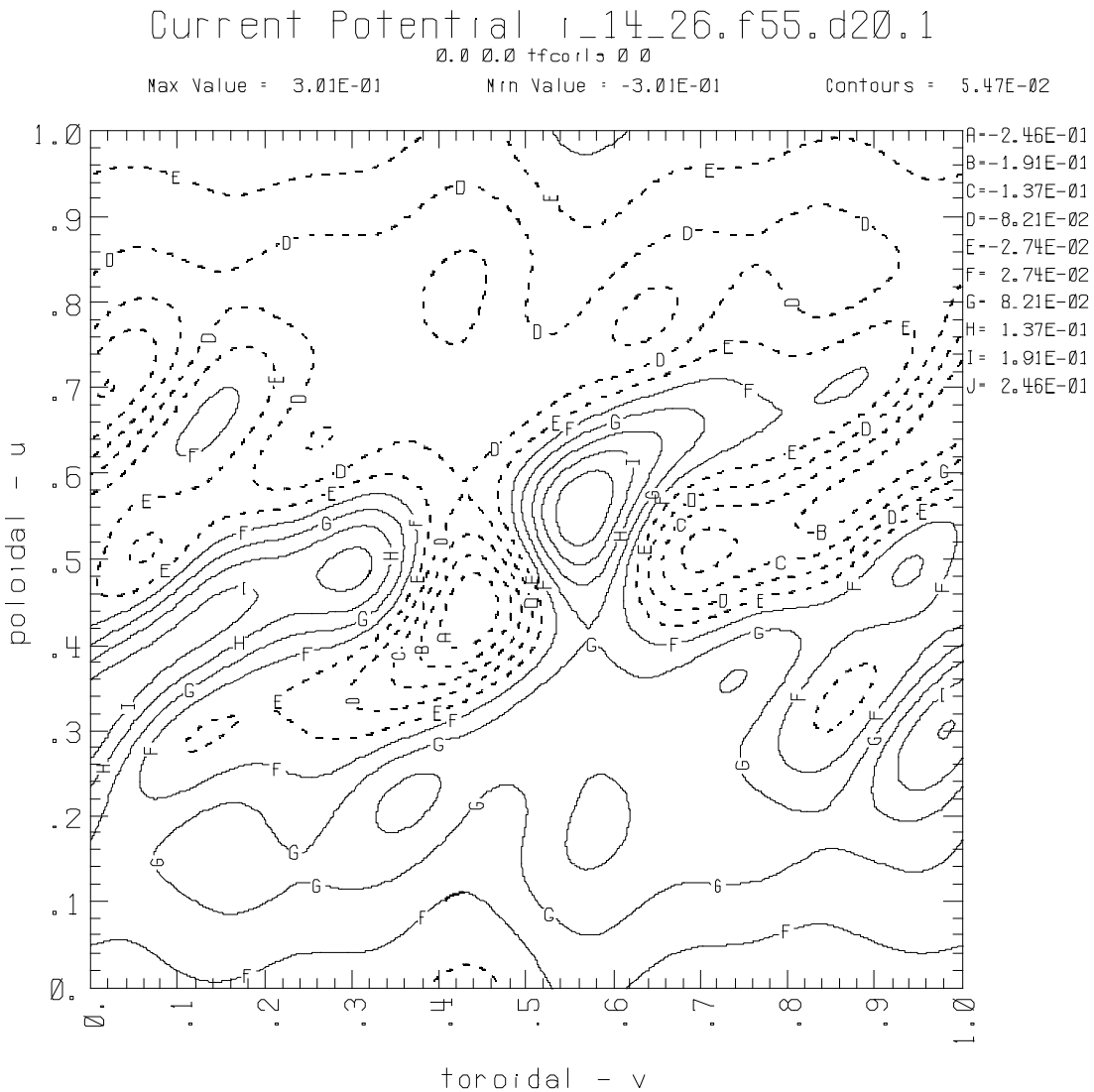

Fig. 1: Low m,n case: Ave err = 3.1812E-02, Max err = 7.001E-01 No SVD. Error too high, Contours ok?

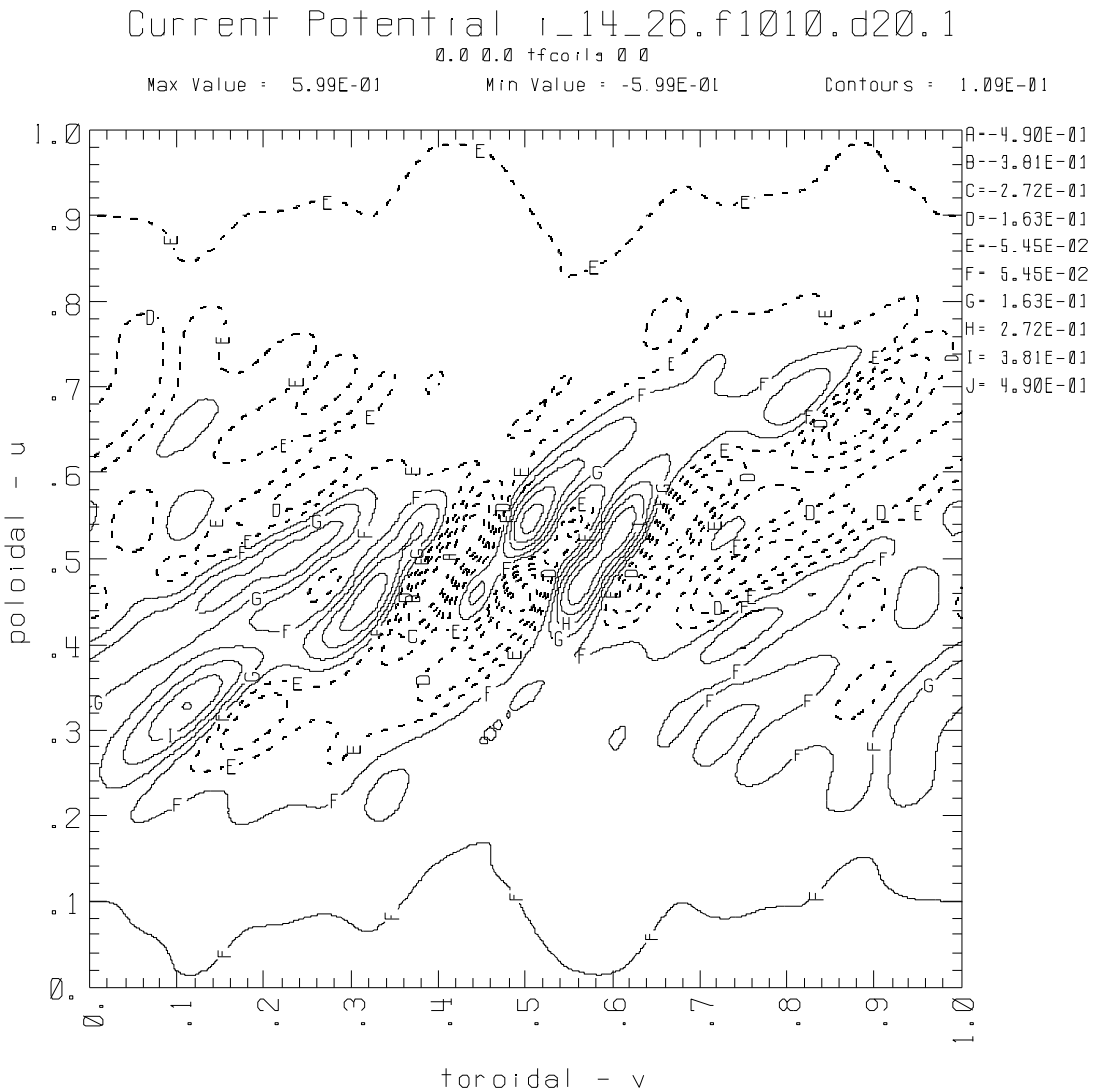

Fig. 2: Standard case: Ave err = 9.5135E-03, Max err = 1.689E-01. No SVD, right error target, bad contours.

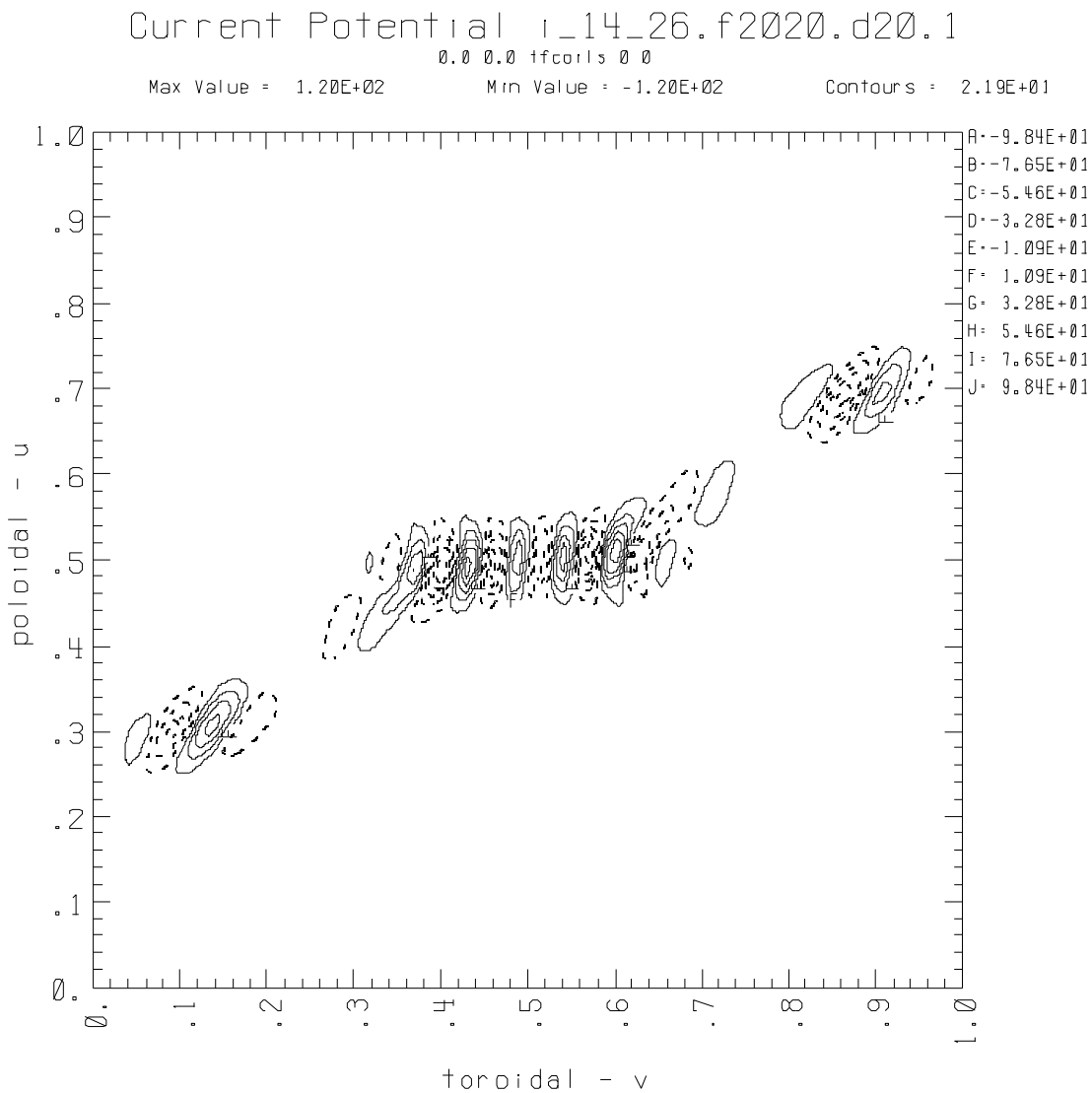

Fig. 3: High m,n, No SVD, Ave err = 8.6463E-04, Max err = 3.565E-02, Error very low, Contours very bad

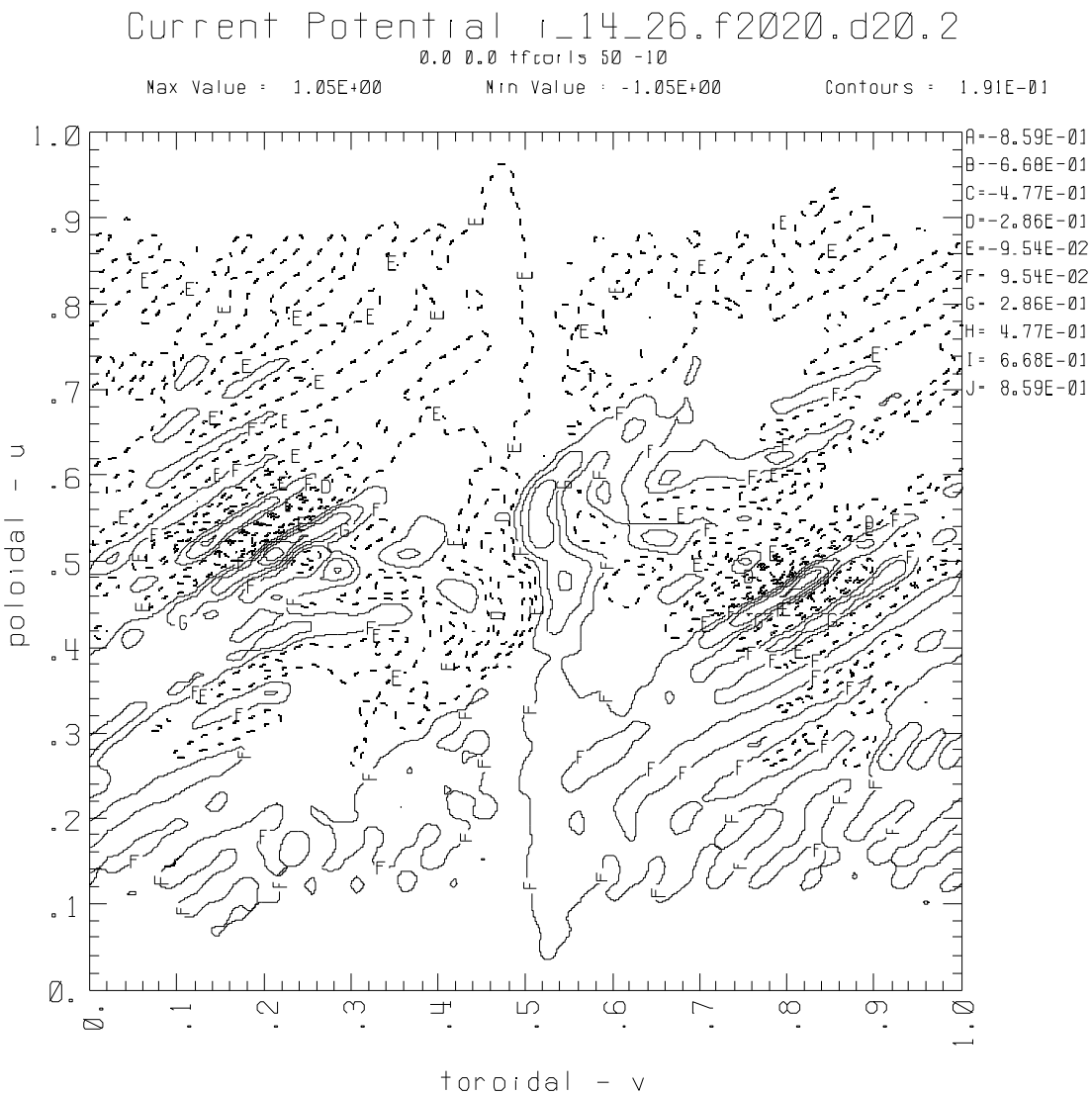

Fig. 4: SVD with 197 weights cutoff: Ave err = 1.1890E-03, Max err = 2.448E-02, error low, contours bad

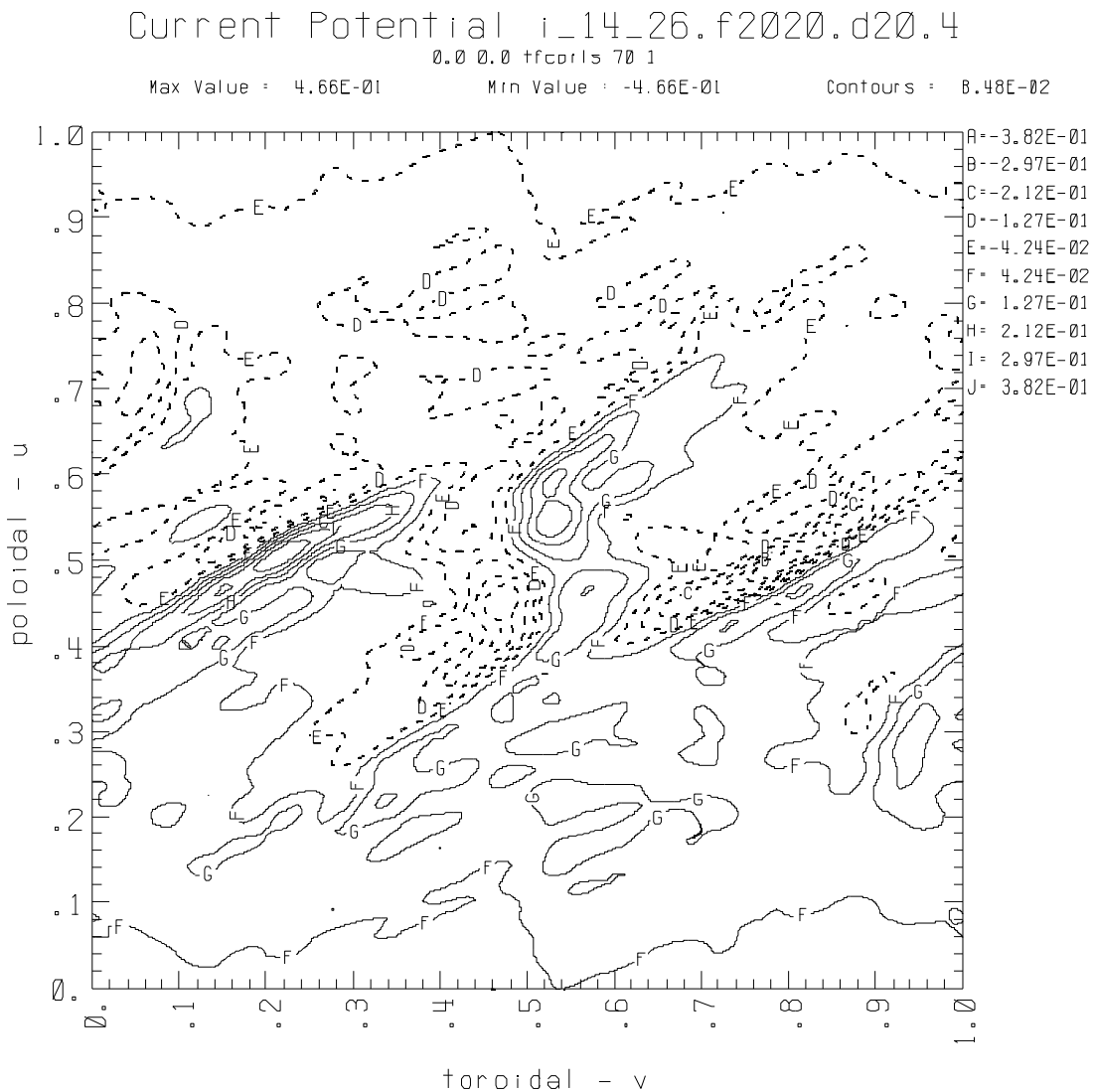

Fig. 5: SVD 433 weights cutoff, Ave err = 3.2902E-03, Max err = 3.430E-02, , error too low, contours still bad.

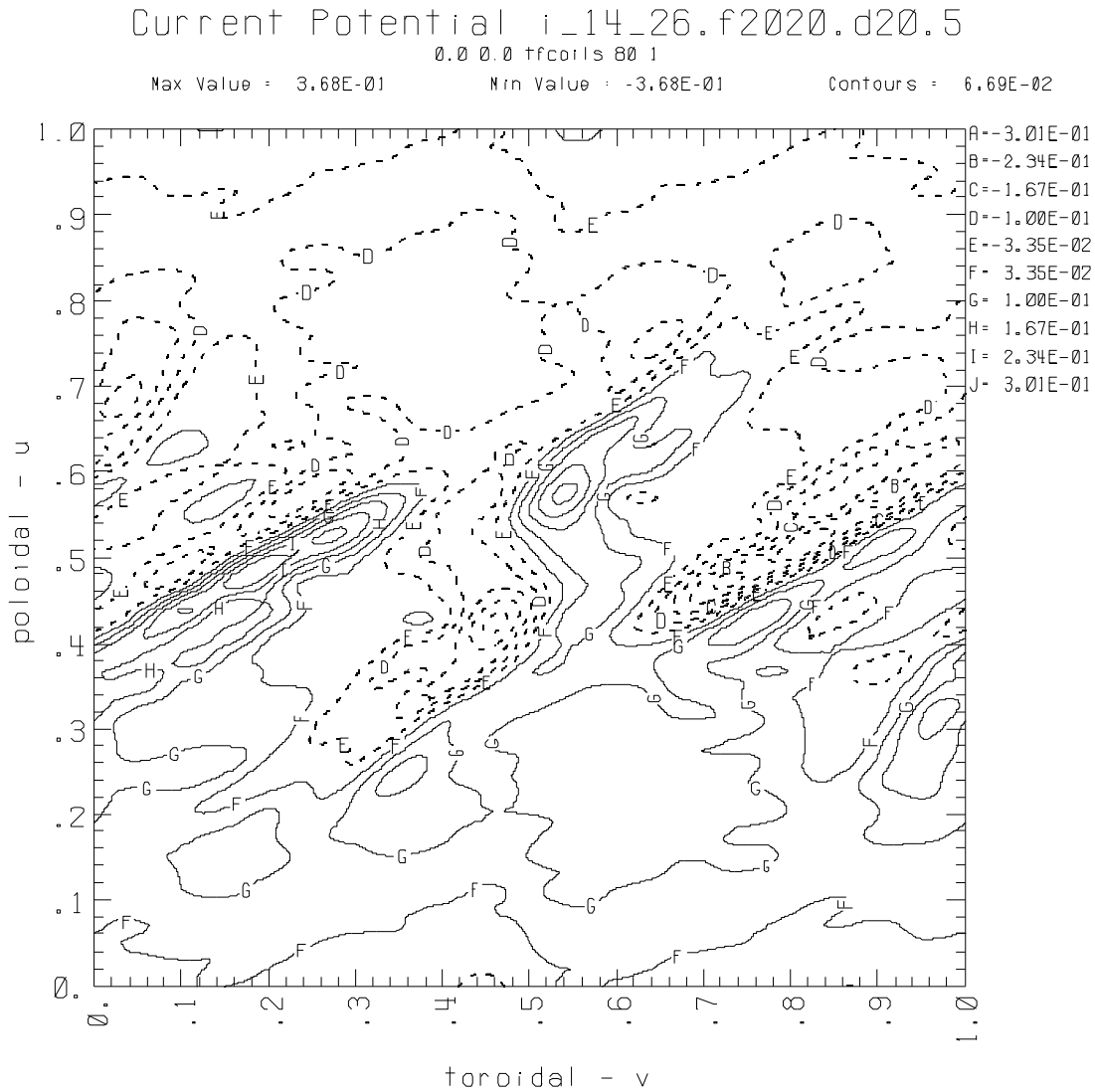

Fig. 6: SVD 571 weights cutoff, Ave err = 6.6186E-03, Max err = 7.453E-02, error low, contours still bad

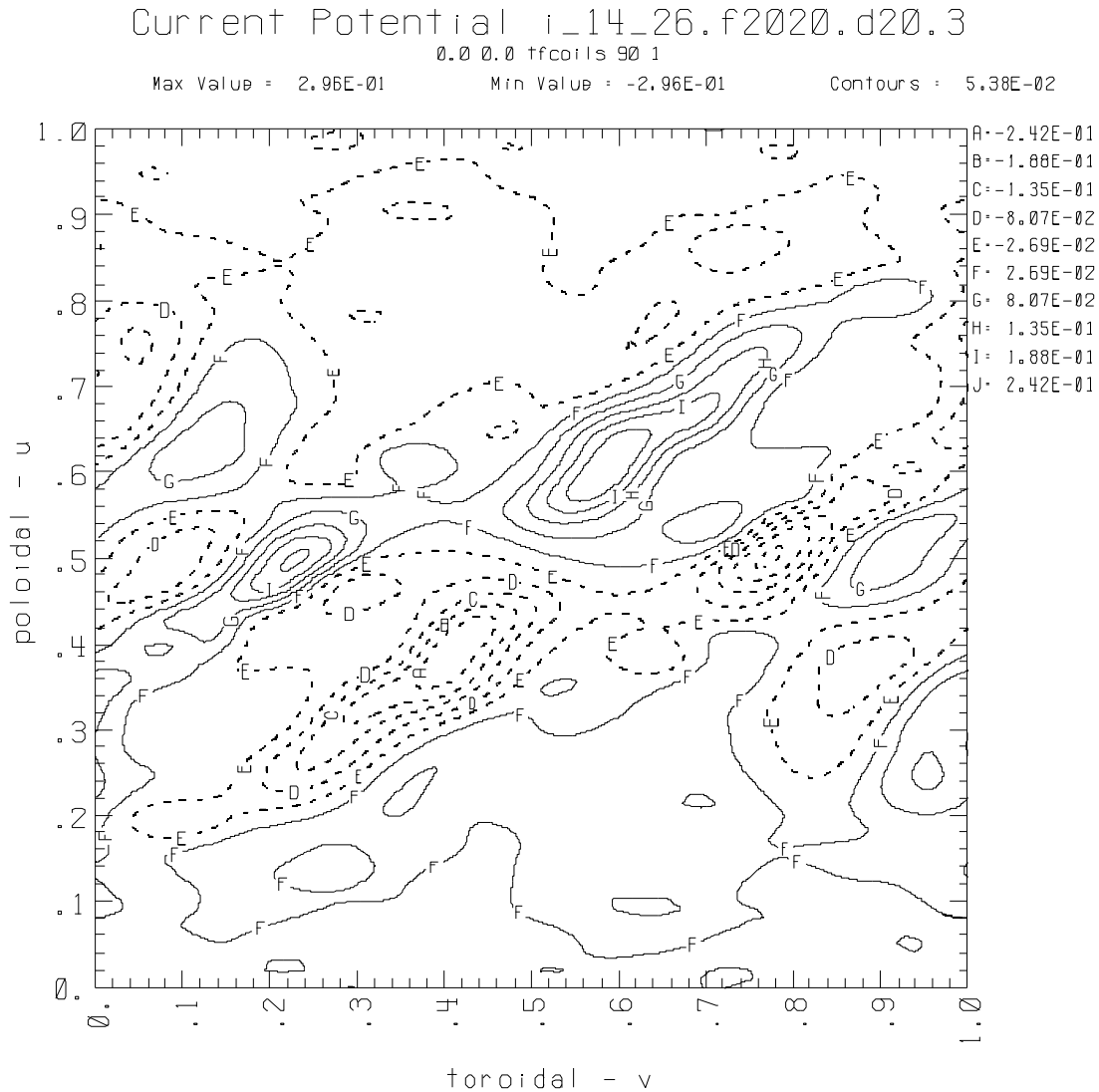

Fig 7: SVD 721 weights cutoff Ave err = 9.7117E-02, Max err = 3.090E-01, error high, contours ok?

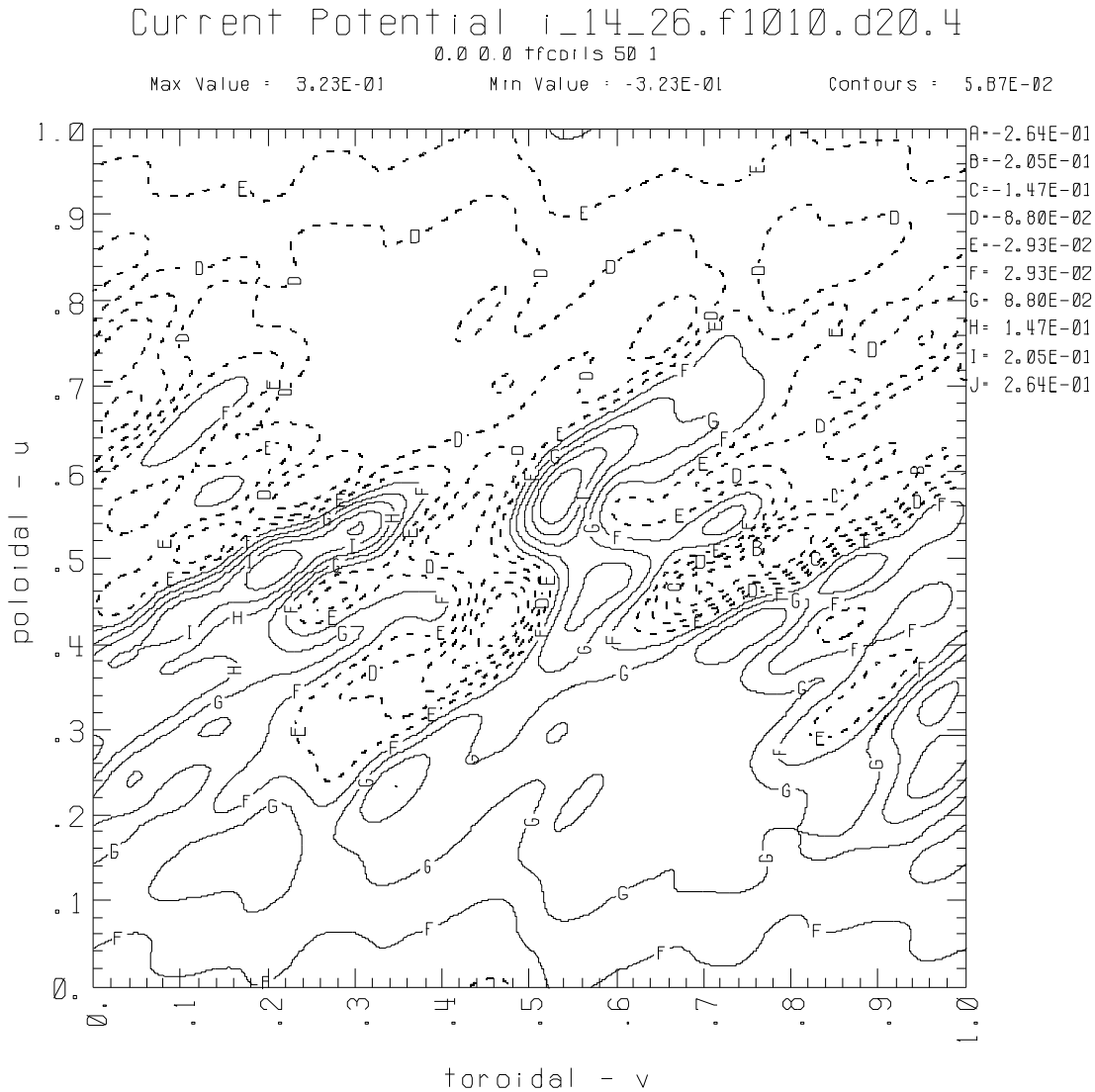

Fig 8 : SVD 35 weights cutoff , Ave err = 1.1552E-02, Max err = 1.002E-01 , error too big

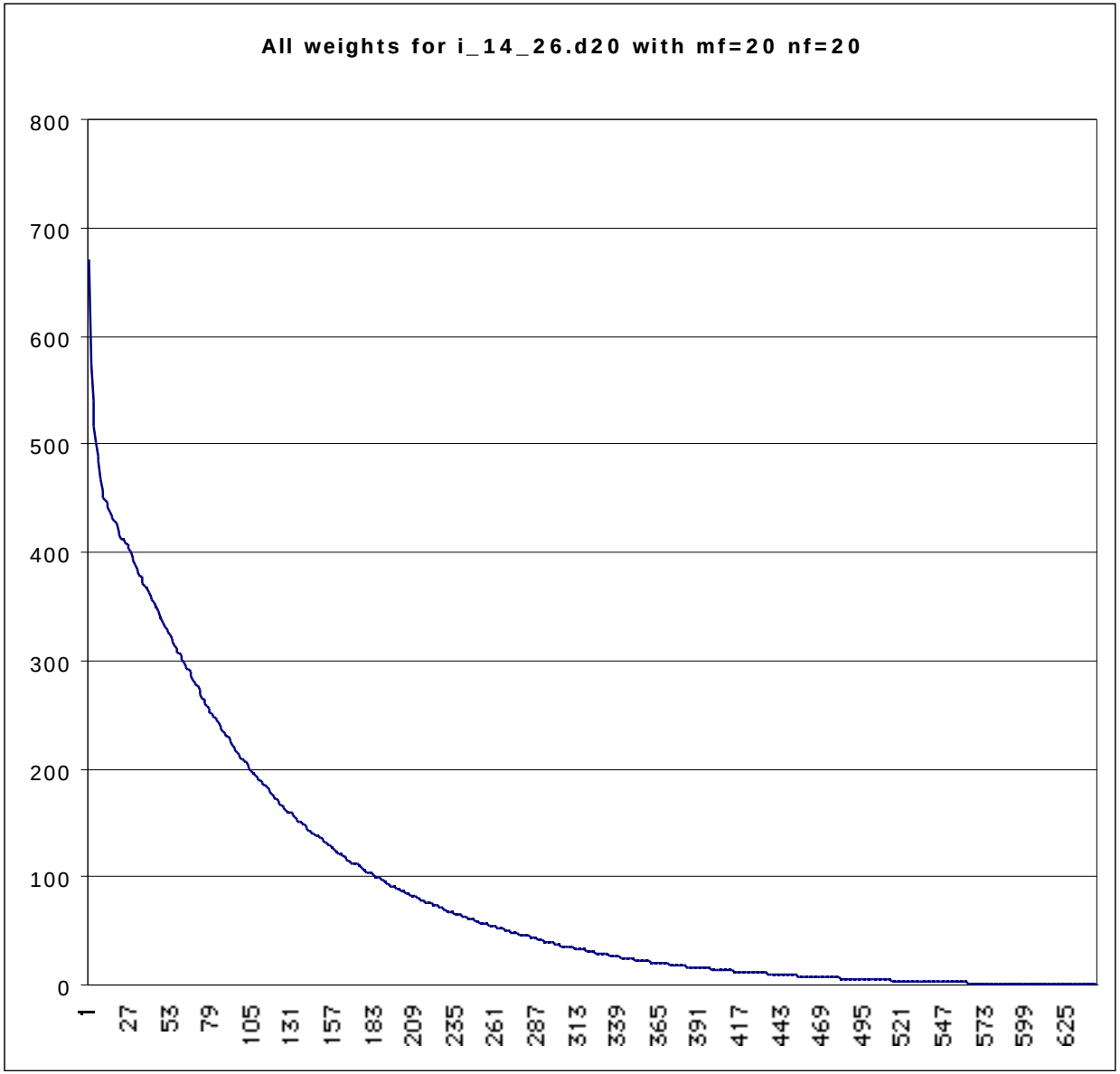

Fig. 9: Total number of weights = 840. Max = 671.5 , Min w = 2.9E-3 , Max/Min ratio R = 227841.8

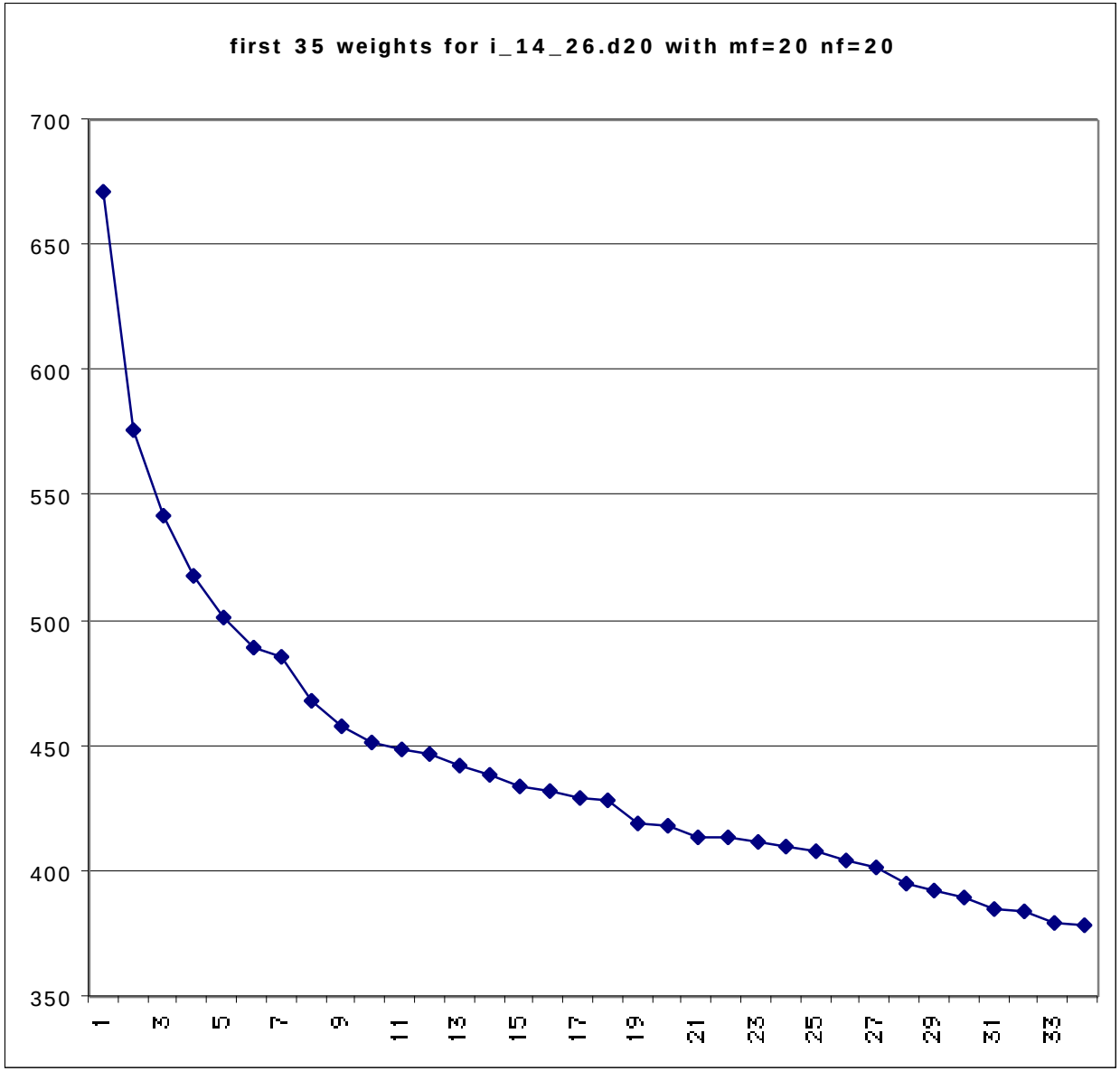

Fig 10: Largest 35 weights