

Review

Here is a review of my recent works on calculating the base flow for a specified separation bubble using SFD method. The non-convergence issue has been fully illustrated for cases using either the classic SFD method or the adaptive SFD method.

In particular, for the first part, which is focus on the classic SFD method, a brief discussion on effects made by different factors on the convergence is given and some related problems which I think can be critical to the non-convergence are raised at the end of the part; for the second part, a typical case using the adaptive SFD method is illustrated and its unreasonable results are analyzed, which seems to reveal some problems that I think, might limit the performance of the adaptive SFD method.

At the end of this review, a question list is summarized for your convenience.

Thank you very much for your patience and any response will be appreciated!

1. Classic SFD

My numerical experiments began from the classic SFD method in order to build an relatively clear awareness of the effects of different factors on the convergence of this method.

Firstly, in all cases here, the velocity components were non-dimensionalized with the free stream velocity at the inlet which is about 6.22m/s while the reference scale for length is chose to be 1m, the choosing of which is relatively arbitrary and with no physical meaning. Besides, the density here is about 1.1256 kg/m^3 and the dynamic viscosity is about $1.9157 \times 10^{-5} \text{ Pa} \cdot \text{s}$.

The **original settings** of the calculation is attached in the folder called **/classic_SFD/N20_order2/test0_1e5_fw02_ce5 in the package**, where the filterwidth $\Delta = 0.2$ and the control coefficient $\chi = 5$. These two parameters were specified simply using the method recommended by Espen Åkervik, where $\Delta = \frac{2}{\omega}$, $\chi \approx \frac{1}{\Delta}$ with ω being the dominant frequency, which was determined based on the spectral analysis of the experiment data. Apart from these, the time step is specified to be 1×10^{-5} and the polynomial expansion order is 1, i.e. NUMMODES=2.

1.1 Effects of the Time Step

First of all, I have found that with the original time step 1×10^{-5} , $\|q - \bar{q}\|_{\infty}$ would always go through a steep increase and even diverge, see FIG.1. This phenomenon, I think, was obviously caused by numerical instability and was then proved to be disappear after shorten the time step to 1×10^{-6} . (The related session files and field files for this case are

attached in the folder called `/classic_SFD/N20_order2/test2_1e6_fw02_ce5)`

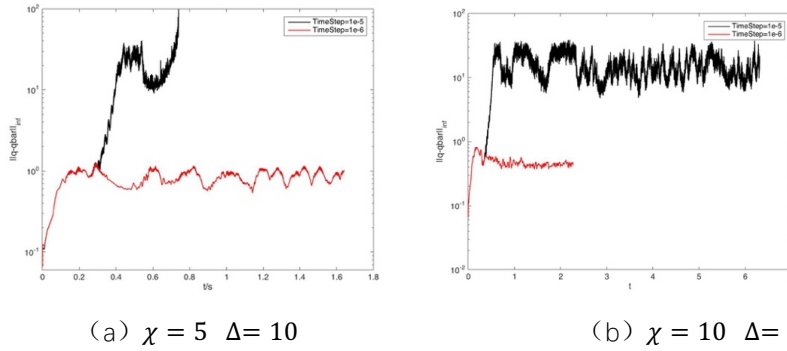


FIG. 1 effects of the time step on numerical stability of the scheme used by SFD method in Nektar++

Q1: I have found that the CFL number under the time step 1×10^{-5} was around $1e-5$, which I thought is theoretically small enough to ensure the numerical stability. But, from the above we can see that it is still not enough. So, I was wondering what is the limit on the CFL number to ensure the numerical stability for SFD calculation?

1.2 Effects of the Key Parameters in SFD Method

Although the numerical stability was guaranteed under the new time step, the calculation still cannot converge. Then I tried to respectively adjusted the filterwidth Δ and Control Coefficient χ . Despite improvements, all the tested cases came out failed!

1.2.1 FilterWidth Δ :

As I understand it, increasing filterwidth Δ is only to extend the frequency range of the disturbances which can be better restrained. That is to say, increasing filterwidth Δ can only benefit the low frequency disturbance while does few to the high frequency ones. As a result, for small- χ situations where high frequency disturbances take the critical part in non-convergence, increasing filterwidth Δ cannot make fundamental improvements, see FIG.2.

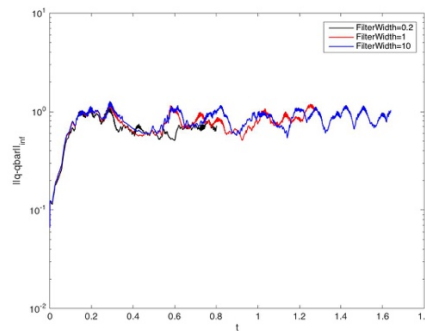


FIG. 2 effects of filterwidth Δ on the convergence of SFD method under time step= 1×10^{-6} , $\chi = 5$

On the contrary, if χ is relatively large and can better restrain the high frequency disturbance while some low frequency ones still obvious, increasing filterwidth Δ can bring about a dramatic improvement, see FIG.3.

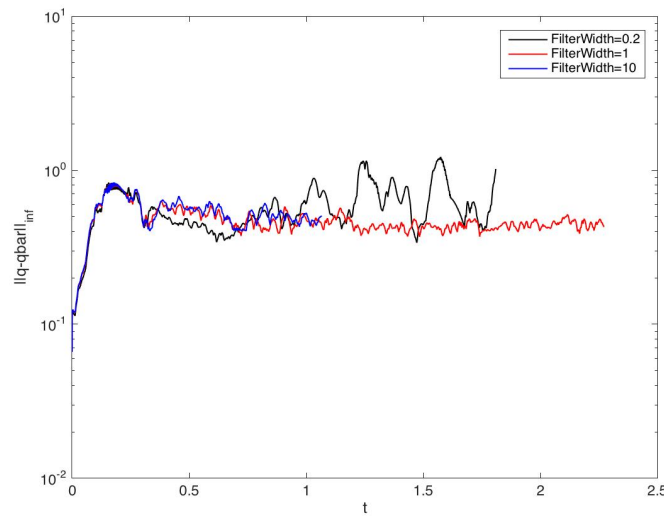


FIG. 3 effects of the filterwidth Δ on the convergence of SFD method

under time step= 1×10^{-6} , $\chi = 10$

1.2.2 Control Coefficient χ :

According to the results in 1.2.1, $\Delta = 1$ seems a better choice while some high frequency disturbances are still badly restrained, which means a larger χ is needed! Then I have increased χ to 20 and 50 which indeed can restrain the high frequency disturbance, but still cannot lead to a converged solution!

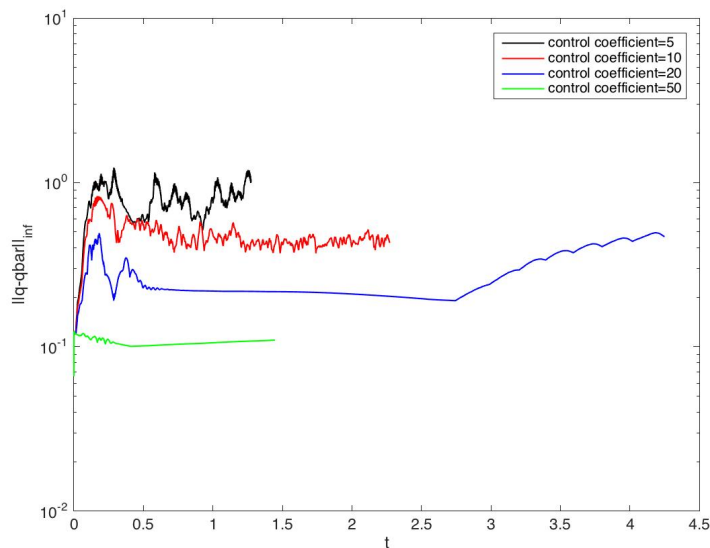


FIG. 4 effects of the control coefficient χ on the convergence of SFD method

under time step= 1×10^{-6} , $\Delta = 1$

1.3 Effects of the Polynomial Expansion Order

Finally, I have tested effects of the polynomial expansion order on the convergence of the classic SFD method. As shown in FIG.5, after enlarging the expansion order from 1 to 2, i.e. NUMMODES changed from 2 to 3, while keeping the other parameters unchanged, the disturbances can be better restrained, though still not thoroughly.

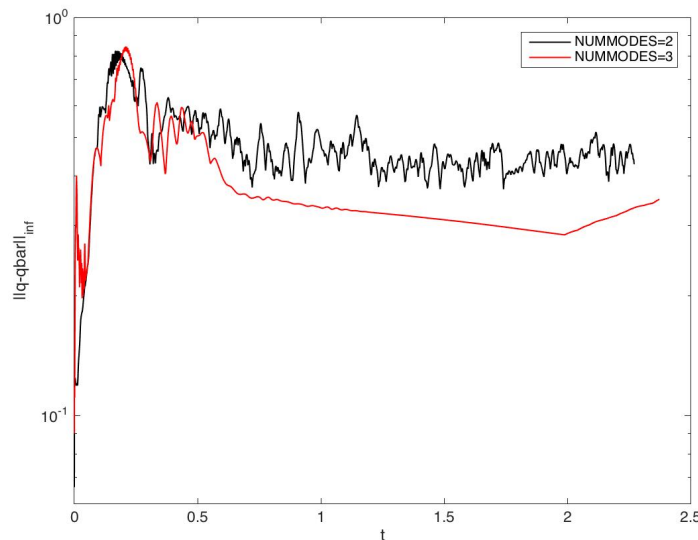


FIG. 5 effects of the expansion order on the convergence of SFD method

under time step= 1×10^{-6} , $\Delta = 1$, $\chi = 10$

Q2: Why a higher resolution of the spatial discretization seems can benefit the convergence of SFD calculation?

Q3: Is it necessary to make the resolution of the spatial discretization to be so high that can lead to a grid-independence solution?

From above, we can see that the polynomial expansion of order 1 (NUMMODES=2) is obviously not adequate for the SFD numerical calculation here to be independent of the resolution of the spatial discretization. However, **I was wondering if it is necessary to make the resolution of the spatial discretization to be that high, which, I think, is the resolution needed for DNS simulation?**

In my opinion, a spatial discretization whose resolution can meet the need of the base flow can be adequate for the SFD calculation.

Firstly, assuming that a SFD calculation can reach the converged solution, a spatial discretization whose resolution can meet the need of the base flow would be enough to make the numerical solution compatible with the analytical base flow; Then, **though the spatial discretization here might be too coarse to catch the analytical dynamic solution of the SFD system** for the scales of the flow structures in the base flow is much

larger than the ones in transient flow, I think it might not hinder the convergence of the SFD calculation as long as a pair of proper parameters χ and Δ is specified. That is to say, when considering the stability analysis, that coarse spatial discretization, as I understand it, would cause the dominant eigenvalue λ_D to deviate from its analytical value and as long as we optimize the parameters χ and Δ for this “numerical” λ_D , I think theoretically the SFD calculation could also converge, though to a numerical base flow (which is compatible to the real one like the above said).

Whereas, the test results seem not consistent with my thoughts:

FIG.6 shows part of a mesh with NUMMODES =2 (see (a)) whose resolution, I think, is already enough for the base flow. (The related cases are in folder called */classic_SFD/N20_order2* in the package) Besides, a finer one with NUMMODES =3 is also shown here as (c). (The related cases are in folder called */classic_SFD/N20_order3* in the package)

After testing several pairs of parameters χ and Δ , I have found that the latter seems more likely to converge, though not yet converge.

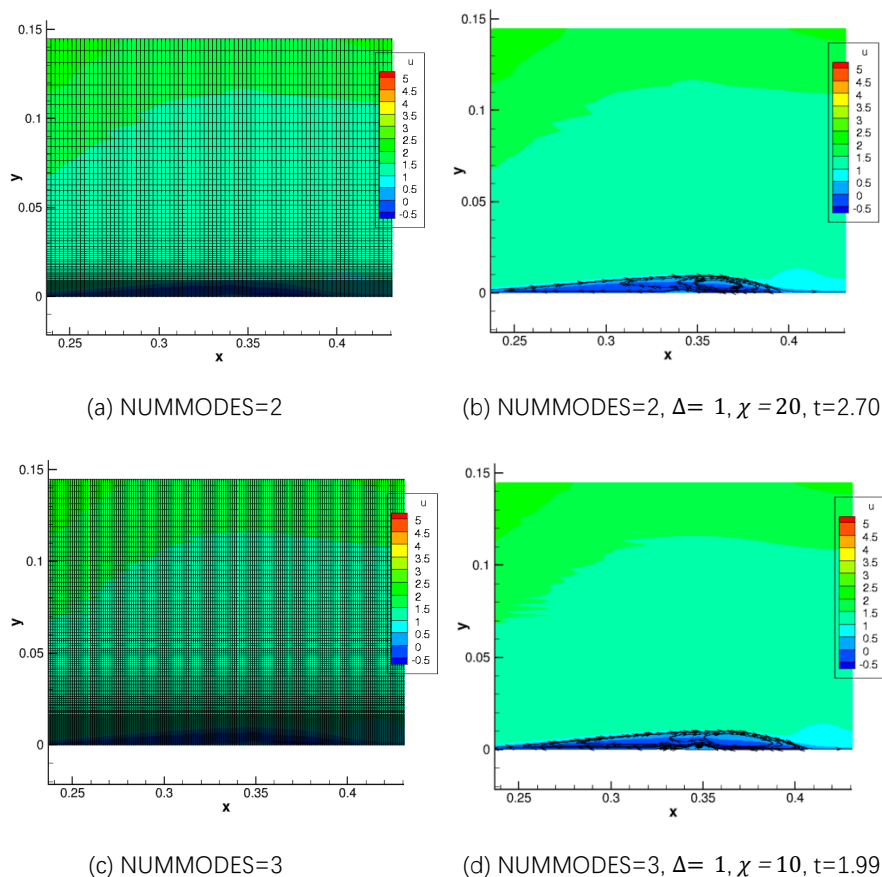


FIG. 6 meshes with different polynomial expansion order and the related transient field results

1.4 Critical Problems and Confusion

Apart from the above, I was wondering if the two problems below might cause my

calculations to fail from convergence:

Q4: Can a **NONLINEAR** disturbance be restrained by the SFD method?

In Jordi' s doctoral thesis, I have found that the discussion on stability properties of the SFD method seemed fundamentally based on a one-dimensional model $\dot{u} = \gamma u$, i.e. $u \propto e^{\gamma t}$. To **extend these properties to the Navier-Stokes system**, I think there seems to be a **necessary condition** that the disturbance should be **linear** and has the formation of a **normal mode, i.e. $u' = \hat{u}e^{\omega t}$** .

Besides, the similar discussion in Espen Åkervik' s paper is also based on linear disturbances, which is, making **a linear stability analysis on the SFD dynamic system** to find how a linear disturbance in original Navier-Stokes system is restrained by the SFD method.

But, in my point of view, since the initial conditions we specified might be far away from the base flow, **the initial disturbance might, as a result, be a nonlinear one**. At this situation, I was wondering if the SFD method can still have the ability to restrains the disturbance and reach a converged solution?

Q5: Can a **high-Reynolds-number** base flow be obtained by the SFD method?

In Jordi' s doctoral thesis, two examples on **incompressible flow past a cylinder** have been given with the Reynolds number being 100 and 300 respectively. But when I tried a similar one with rather high Reynolds number, i.e. **3900** (having tried a series of SFD parameters χ and Δ), the SFD calculation **seems could not converge**. So I was wondering if the SFD method can be used for the high-Reynolds-number flow? If not, I would love to know why.

2. Adaptive SFD

The case below is one of my test cases using adaptive SFD method and it is chosen for there are several typical problems in this case.

2.1 Settings and Results

The session files and other related field files are included in the folder called **/adaptive_SFD/N20_order2/test1_1e6_ce20** in the package. The related settings are listed below:

- [1] time step= 1×10^{-6} ;
- [2] expansion order NUMMODES=2 ;
- [3] initial filterwidth $\Delta = 1$;
- [4] initial control coefficient $\chi = 20$;
- [5] the tolerance for judging the 'partially converged' steady state
AdaptiveTOL=0.01 (default) ;
- [6] initial conditions for SFD : take the transient result at t=2.74 when the $\|q - \bar{q}\|_{\infty}$

decreased to its minimum during the classic SFD calculation under the NUMMODES=2, $\Delta=1$, $\chi=20$

(see `'/classic_SFD/N20_order2/test2_1e6_ce20/v1_2D_baseflow_1164_N20_order2_1e6_ce20_274.chk'`)

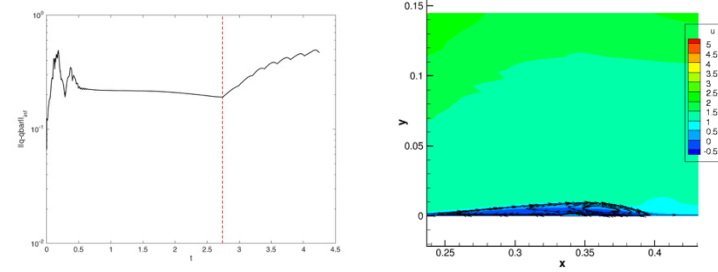


FIG. 7 the transient results of the classic SFD under time step= 1×10^{-6} , $\Delta=1$, $\chi=20$

which is used as the initial conditions for the adaptive SFD

Part of the output on the terminal is shown in FIG.8 below:

```
SFD - Step: 100; Time: 2.7401; CPU time = 14.0934 s; Tot time = 14.0934 s; X = 20; Delta = 1; |q-qBar|inf = 0.000423389

The SFD method is converging: we compute stability analysis using the 'partially converged' steady state as base flow:

growthEV = -5.6061
frequencyEV = 2783.4

We enter the Gradient Descent Method [...]
The Gradient Descent Method has converged!

The updated parameters are: X_tilde = 1.14859e+06 and Delta_tilde = -8.15983e+09
SFD - Step: 200; Time: 2.7402; CPU time = 2996.85 s; Tot time = 3010.95 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 300; Time: 2.7403; CPU time = 9.63641 s; Tot time = 3020.58 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 400; Time: 2.7404; CPU time = 9.60659 s; Tot time = 3030.19 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 500; Time: 2.7405; CPU time = 9.59951 s; Tot time = 3039.79 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 600; Time: 2.7406; CPU time = 9.56172 s; Tot time = 3049.35 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 700; Time: 2.7407; CPU time = 9.57577 s; Tot time = 3058.93 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 800; Time: 2.7408; CPU time = 9.58079 s; Tot time = 3068.51 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 900; Time: 2.7409; CPU time = 9.63045 s; Tot time = 3078.14 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1000; Time: 2.7411; CPU time = 9.60502 s; Tot time = 3087.74 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1100; Time: 2.7411; CPU time = 9.58763 s; Tot time = 3097.33 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1200; Time: 2.7412; CPU time = 9.55588 s; Tot time = 3106.89 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1300; Time: 2.7413; CPU time = 9.5827 s; Tot time = 3116.47 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1400; Time: 2.7414; CPU time = 9.61423 s; Tot time = 3126.08 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1500; Time: 2.7415; CPU time = 9.5801 s; Tot time = 3135.66 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1600; Time: 2.7416; CPU time = 9.58801 s; Tot time = 3145.25 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1700; Time: 2.7417; CPU time = 9.56725 s; Tot time = 3154.82 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1800; Time: 2.7418; CPU time = 9.54298 s; Tot time = 3164.36 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 1900; Time: 2.7419; CPU time = 10.0146 s; Tot time = 3174.38 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2000; Time: 2.742; CPU time = 9.76943 s; Tot time = 3184.15 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2100; Time: 2.7421; CPU time = 9.55144 s; Tot time = 3193.7 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2200; Time: 2.7422; CPU time = 10.006 s; Tot time = 3203.7 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2300; Time: 2.7423; CPU time = 9.66486 s; Tot time = 3213.37 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2400; Time: 2.7424; CPU time = 9.53177 s; Tot time = 3222.9 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2500; Time: 2.7425; CPU time = 9.82477 s; Tot time = 3232.73 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2600; Time: 2.7426; CPU time = 9.62208 s; Tot time = 3242.35 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2700; Time: 2.7427; CPU time = 9.58203 s; Tot time = 3251.93 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2800; Time: 2.7428; CPU time = 9.53904 s; Tot time = 3261.47 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 2900; Time: 2.7429; CPU time = 9.56894 s; Tot time = 3271.04 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3000; Time: 2.743; CPU time = 9.54413 s; Tot time = 3280.58 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3100; Time: 2.7431; CPU time = 9.59983 s; Tot time = 3290.18 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3200; Time: 2.7432; CPU time = 9.55784 s; Tot time = 3299.74 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3300; Time: 2.7433; CPU time = 9.59719 s; Tot time = 3309.34 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3400; Time: 2.7434; CPU time = 9.55487 s; Tot time = 3318.89 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3500; Time: 2.7435; CPU time = 9.57055 s; Tot time = 3328.46 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3600; Time: 2.7436; CPU time = 9.60148 s; Tot time = 3338.06 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3700; Time: 2.7437; CPU time = 9.54204 s; Tot time = 3347.61 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3800; Time: 2.7438; CPU time = 9.59991 s; Tot time = 3357.21 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 3900; Time: 2.7439; CPU time = 9.72969 s; Tot time = 3366.94 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
SFD - Step: 4000; Time: 2.744; CPU time = 9.84585 s; Tot time = 3376.78 s; X = 1.14859e+06; Delta = -8.15983e+09; q-qBar inf = 1.96589e-06
```

FIG. 8 part of the output on the terminal for the case above using adaptive SFD method

2.2 Typical Problems and Confusion

2.2.1 Unreasonable judgement on the 'partially converged' steady state

First, we can see that for the initial value of the $\|q - \bar{q}\|_\infty$ is much less than the AdaptiveTOL, the initial field is treated as the 'partially converged' steady state, see FIG.9.

SFD - Step: 100; Time: 2.7401; CPU time = 14.0934 s; Tot time = 14.0934 s; X = 20; Delta = 1; $|q - \bar{q}|_{\infty} = 0.000423389$

The SFD method is converging; we compute stability analysis using the 'partially converged' steady state as base flow:

FIG. 9 Judgement for the 'partially converged' steady state

However, as the transient result at $t=2.74$, the initial field has a $\|q - \bar{q}\|_{\infty}$ about 0.19

during the classic SFD calculation under the NUMMODES=2, $\Delta=1$, $\chi=20$, see FIG.7. So,

I think it is actually far from being a steady state and hence there was a mistake on judging the 'partially converged' steady state.

Q6 : For this phenomenon, I was wondering if it is the type of the judgement condition rather than the specified value of it that cause this mistake.

I have this point of view because I think a small $\|q - \bar{q}\|_{\infty}$ at the beginning of the SFD calculation might not mean a small $\frac{\partial q}{\partial t}$, thus neither a 'partially converged' steady state, while the judgement condition of default type 'threshold' does not take this situation into consideration.

To explain this, firstly I think the reason why the initial value of $\|q - \bar{q}\|_{\infty}$ here in the adaptive SFD calculation is so small, is that the initial value of $\bar{q}$ is set to be equal to the initial value of q (I found it just after a cursory read of the related program, so I am not sure if there was anything wrong with this understanding? If not, I would love to know what is the physical meaning of this treatment?). With this treatment, as I understand it, the $\|q - \bar{q}\|_{\infty}$ at the beginning of SFD calculation would no longer have its original physical meaning, i.e. the deviation from the steady state. Hence a small $\|q - \bar{q}\|_{\infty}$ at that situation might not mean a 'partially converged' steady state. Besides, this kind of small $\|q - \bar{q}\|_{\infty}$ might later quickly increase after calculated for a few steps, which has been confirmed to happen. That is to say, the $\frac{\partial q}{\partial t}$ here might not be small and the q here might not be a 'partially converged' steady state.

Q7: Considering this mistake can lead to a pair of unreasonable χ and Δ which might finally cause a non-convergence, is there any alternative judgement condition to solve this problem?

2.2.2 Unreasonable dominant eigenvalue and the resulted χ and Δ

After computing the linear stability analysis using that transient result which has been mistaken for a 'partially converged' steady state as base flow, the dominant eigenvalue and the related optimized parameters χ and Δ were obtained, see FIG.10.

```
growthEV = -5.6061
frequencyEV = 2783.4
```

We enter the Gradient Descent Method [...]
The Gradient Descent Method has converged!

The updated parameters are: $\tilde{X} = 1.14859e+06$ and $\tilde{\Delta} = -8.15983e+09$

FIG. 10 dominant eigenvalue and the related optimize parameters χ and Δ

Q8: When considering the convergence history of related classic SFD calculations, I was wondering if this dominant eigenvalue a reasonable one, which I think is not? If not, is it due to the aforementioned mistake on judging the 'partially converged' steady state or some other problems in computing stability analysis?

Q9: Apart from the dominant eigenvalue, the resulted optimal parameters χ and Δ , I think, are absolutely unreasonable, especially the negative value of Δ !

2.2.3 Hard to set the terminating condition

After optimizing the parameters χ and Δ , a classic SFD calculation started as follow, see FIG.11.

```
The updated parameters are: X_tilde = 1.14859e+06 and Delta_tilde = -8.15983e+09
SFD - Step: 200; Time: 2.7402; CPU time = 2996.85 s; Tot time = 3010.95 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 300; Time: 2.7403; CPU time = 9.63641 s; Tot time = 3020.58 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 400; Time: 2.7404; CPU time = 9.60659 s; Tot time = 3030.19 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 500; Time: 2.7405; CPU time = 9.59951 s; Tot time = 3039.79 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 600; Time: 2.7406; CPU time = 9.56172 s; Tot time = 3049.35 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 700; Time: 2.7407; CPU time = 9.57577 s; Tot time = 3058.93 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 800; Time: 2.7408; CPU time = 9.58079 s; Tot time = 3068.51 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 900; Time: 2.7409; CPU time = 9.63045 s; Tot time = 3078.14 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1000; Time: 2.741; CPU time = 9.60502 s; Tot time = 3087.74 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1100; Time: 2.7411; CPU time = 9.58763 s; Tot time = 3097.33 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1200; Time: 2.7412; CPU time = 9.55588 s; Tot time = 3106.89 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1300; Time: 2.7413; CPU time = 9.5827 s; Tot time = 3116.47 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1400; Time: 2.7414; CPU time = 9.61423 s; Tot time = 3126.08 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1500; Time: 2.7415; CPU time = 9.5801 s; Tot time = 3135.66 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1600; Time: 2.7416; CPU time = 9.58801 s; Tot time = 3145.25 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1700; Time: 2.7417; CPU time = 9.56725 s; Tot time = 3154.82 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1800; Time: 2.7418; CPU time = 9.54298 s; Tot time = 3164.36 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 1900; Time: 2.7419; CPU time = 10.0146 s; Tot time = 3174.38 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2000; Time: 2.742; CPU time = 9.76943 s; Tot time = 3184.15 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2100; Time: 2.7421; CPU time = 9.55144 s; Tot time = 3193.7 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2200; Time: 2.7422; CPU time = 10.006 s; Tot time = 3203.7 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2300; Time: 2.7423; CPU time = 9.66486 s; Tot time = 3213.37 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2400; Time: 2.7424; CPU time = 9.53177 s; Tot time = 3222.9 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2500; Time: 2.7425; CPU time = 9.82477 s; Tot time = 3232.73 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2600; Time: 2.7426; CPU time = 9.62208 s; Tot time = 3242.35 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2700; Time: 2.7427; CPU time = 9.58203 s; Tot time = 3251.93 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2800; Time: 2.7428; CPU time = 9.53904 s; Tot time = 3261.47 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 2900; Time: 2.7429; CPU time = 9.56894 s; Tot time = 3271.04 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3000; Time: 2.743; CPU time = 9.54413 s; Tot time = 3280.58 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3100; Time: 2.7431; CPU time = 9.59983 s; Tot time = 3290.18 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3200; Time: 2.7432; CPU time = 9.55784 s; Tot time = 3299.74 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3300; Time: 2.7433; CPU time = 9.59719 s; Tot time = 3309.34 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3400; Time: 2.7434; CPU time = 9.55487 s; Tot time = 3318.89 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3500; Time: 2.7435; CPU time = 9.57055 s; Tot time = 3328.46 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3600; Time: 2.7436; CPU time = 9.60148 s; Tot time = 3338.06 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3700; Time: 2.7437; CPU time = 9.54204 s; Tot time = 3347.61 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3800; Time: 2.7438; CPU time = 9.59991 s; Tot time = 3357.21 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 3900; Time: 2.7439; CPU time = 9.72969 s; Tot time = 3366.94 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
SFD - Step: 4000; Time: 2.744; CPU time = 9.84585 s; Tot time = 3376.78 s; X = 1.14859e+06; Delta = -8.15983e+09; q-GBAR inf = 1.96589e-06
```

FIG. 11 the resulted classic SFD calculation process

We can see that the $\|q-\bar{q}\|_{\infty}$ has been 1.96589e-6 for a long time, which, as I

understand it, means $\frac{\partial q}{\partial t}$ is small at that time. But in my opinion, it does not mean the almost unchanged q at that time is the steady state of the Navier-Stokes system because the χ is also very large which made the extra term $\chi(q-\bar{q})$ could not be neglected!

Q10: Hence, I think it seems necessary to take the value of χ into account as well in the terminating condition rather than just consider the value of $\|q-\bar{q}\|_{\infty}$?

3. Question list

3.1 Questions for Classic SFD

Q1: I have found that the CFL number under the time step 1×10^{-5} was around $1e-5$, which I thought is theoretically small enough to ensure the numerical stability. But, from the above we can see that it is still not enough. So, I was wondering what is the limit on the CFL number to ensure the numerical stability for SFD calculation?

Q2: Why a higher resolution of the spatial discretization seems can benefit the convergence of SFD calculation?

Q3: Is it necessary to make the resolution of the spatial discretization to be so high that can lead to a grid-independence solution?

Q4: Can a NONLINEAR disturbance be restrained by the SFD method?

Q5: Can a high-Reynolds-number base flow be obtained by the SFD method?

3.2 Questions for Adaptive SFD

Q6 : For this phenomenon, I was wondering if it is the type of the judgement condition rather than the specified value of it that cause this mistake.

Q7: Considering this mistake can lead to a pair of unreasonable χ and Δ which might finally cause a non-convergence, is there any alternative judgement condition to solve this problem?

Q8: When considering the convergence history of related classic SFD calculations, I was wondering if this dominant eigenvalue a reasonable one, which I think is not? If not, is it due to the aforementioned mistake on judging the 'partially converged' steady state or some other problems in computing stability analysis?

Q9: Apart from the dominant eigenvalue, the resulted optimal parameters χ and Δ , I think, are absolutely unreasonable, especially the negative value of Δ !

Q10: Hence, I think it seems necessary to take the value of χ into account as well in the terminating condition rather than just consider the value of $\|q-\bar{q}\|_{\infty}$?

