# SNflag1

## Code

```
self.SNflag1 = (1-epost <= Apre*(1-epre)/Apost) & (Apre*(1+epre)/Apost <= 1+epost)
```

## Source

Willems et al. 2005 equation 21 `https://iopscience.iop.org/article/10.1086/429557`
Flannery & van den Heuvel 1975 equation 10 `http://adsabs.harvard.edu/full/1975A%26A....39...61F`

## Principles

SNflag1 is checking to make sure the post-SN orbit passes through the position of the two stars at the time of the SN explosion.

## Equations

This is the incorrect equation that is in the paper.

$$1 - e_{postSN} \leq \frac{A_{preSN}}{A_{postSN}} \leq 1 - e_{postSN} \tag{1}$$

This is the corrected equation.

$$A_{post}(1 - e_{post}) \leq A_{pre} \leq A_{post}(1 + e_{post}) \tag{2}$$

This equation is what is in the script.

$$1 - e_{postSN} \leq \frac{r}{A_{postSN}} \leq 1 + e_{epostSN} \tag{3}$$

## Notes

This part has changed with $e_{preSN}$ implemented, and we are confident in the change.

# SNflag2

## Code

```
self.SNflag2 = (Apre/Apost < 2-((Mtot_pre/Mtot_post)*((Vkick/Vr)-1)**2))
             & (Apre/Apost > 2-((Mtot_pre/Mtot_post)*((Vkick/Vr)+1)**2))
```

## Source

Willems et al. 2005 equations 22 and 23 https://iopscience.iop.org/article/10.1086/429557
Kalogera & Lorimer 2000 equations 6 and 7 https://iopscience.iop.org/article/10.1086/308417

## Principles

The last two inequalities correspond to lower and upper limits on the amount of orbital contraction or expansion that can take place for a given amount of mass loss and a given magnitude of the kick velocity.

## Equations

$$\frac{A_{preSN}}{A_{postSN}} < 2 - \frac{M_{He} + M_2}{M_{BH} + M_2}\left(\frac{V_k}{V_{He,preSN}} - 1\right)^2 \tag{4}$$

$$\frac{A_{preSN}}{A_{postSN}} < 2 - \frac{M_{He} + M_2}{M_{BH} + M_2}\left(\frac{V_k}{V_{He,preSN}} + 1\right)^2 \tag{5}$$

## Notes

For flag 2 we are unsure whether it needs to be change, and more research will be done with it.

# SNflag3

## Code

```
self.SNflag3 = (Vkick/Vr < 1 + np.sqrt(2*Mtot_post/Mtot_pre))
             & ((Mtot_post/Mtot_pre > 0.5)
             | (Vkick/Vr>1 - np.sqrt(2*Mtot_post/Mtot_pre)))
```

## Source

Willems et al. 2005 equations 24 and 25 https://iopscience.iop.org/article/10.1086/429557

## Principles

The magnitude of the kick velocity imparted to the BH at birth is restricted to the range determined by the equations. The first inequality expresses the requirement that the binary must remain bound after the SN explosion, while the second inequality yields the minimum kick velocity required to keep the system bound if more than half of the total system mass is lost in the explosion.

## Equations

$$\frac{V_k}{V_{He,preSN}} < 1 + \left(2\frac{M_{BH}+M_2}{M_{He}+M_2}\right)^{\frac{1}{2}} \tag{6}$$

$$\frac{V_k}{V_{He,preSN}} < 1 - \left(2\frac{M_{BH}+M_2}{M_{He}+M_2}\right)^{\frac{1}{2}} \tag{7}$$

## Notes

We are also unsure whether flag3 needs to be updated, and more reserach is being done to find out.

# SNflag4

## Code

```
self.oldSNflag4 = np.full(self.SNflag3.shape,0)
ieGT1 = np.where((epost>1)|(epost<0))[0]
ieLT1 = np.where(epost<=1)[0]
self.SNflag4[ieGT1] = False
kvar=2*(Apost[ieLT1]/Apre[ieLT1])-(((Vkick[ieLT1]**2)*Apost[ieLT1]
    /(G*Mtot_post[ieLT1]))+1)

tmp1 = kvar**2 * Mtot_post[ieLT1] * (Apre[ieLT1]/Apost[ieLT1])
tmp2 = 2 * (Apost[ieLT1]/Apre[ieLT1])**2 * (1-epost[ieLT1]**2) - kvar
tmp3 = - 2 * (Apost[ieLT1]/Apre[ieLT1]) * np.sqrt(1-epost[ieLT1]**2)
     * np.sqrt((Apost[ieLT1]/Apre[ieLT1])**2 * (1-epost[ieLT1]**2) - kvar)
prgmax = -Mcomp[ieLT1] + tmp1 / (tmp2 + tmp3)

self.oldSNflag4[ieLT1] = Mhe[ieLT1] <= prgmax

self.SNflag4 = (Mhe <= -Mcomp + kvar**2 * (Mtot_post) * Apre/Apost *
1/(2*Apost/Apre*np.sqrt(1-epost**2) * np.sqrt((Apost/Apre)**2 * (1-epost**2) - kvar))
```

## Source

Willems et al. 2005 equation 26 `https://iopscience.iop.org/article/10.1086/429557`
Kalogera & Fryer 1997 equation 12 `https://iopscience.iop.org/article/10.1086/304772/`

## Principles

An upper limit on the mass of the BH progenitor can be derived from the condition that the azimuthal direction of the kick is real.

## Derivation

$$
\begin{aligned}
M_{He} \leq & -M_2 + k^2(M_2 + M_{BH})\frac{A_{preSN}}{A_{postSN}} \\
& \times \left\{ 2\left(\frac{A_{postSN}}{A_{preSN}}\right)^2 \left(1 - e_{postSN}^2\right) \right. \\
& - 2\frac{A_{postSN}}{A_{preSN}}\left(1 - e^2\right)^{\frac{1}{2}} \\
& \left. \times \left[ \left(\frac{A_{postSN}}{A_{preSN}}\right)^2 \left(1 - e_{postSN}^2\right) - k \right]^{\frac{1}{2}} \right\}^{-1}
\end{aligned}
$$

## Notes

Unsure if check is necessary, because all kicks sampled have to be real. When checked, many events that passed all other flags failed this flag, which is unlike what is expected. In addition, this flag most likely has to change with epre implemented, but we are unsure how it will be changed.

# SNflag5

## Code

```
self.SNflag5 = (1/epost)*(1.-rpre/Apost)<=1.
```

## Notes

This check looks the same as the updated SNcheck1, and all systems have the same truth value for this check and check1.

# SNflag6

## Code

```
self.SNflag6 = zams.rzams(Mcomp)<zams.roche_lobe(Mcomp,Mns)*Apost
```

## Notes

This check always evaluates to the same value as check1 and check5.

# SNflag7

## Code

```
self.SNflag7 = zams.rhe(Mhe)<zams.roche_lobe(Mhe,Mcomp)*Apost
```

## Notes

This flag is not documented in the code, so I am currently looking to it.