0: sin

1: summSin

2: summSinNechet

massPQ[n-1] + Math.sin((2\*n-1)\*fi);

3: summSinDiv

massPQ[n-1] - (Math.sin(n\*fi)/constanta); massPQ[n-1] + (Math.sin(n\*fi)/constanta);

4: sinProd

5 \* Math.sin((n+1)\*fi) \* Math.sin(n\*fi);

5: summCos

massPQ[n-1] + Math.cos(n\*fi);

6: summSinZnakoPeremen

massPQ[n-1] - Math.sin(n\*fi); massPQ[n-1] + Math.sin(n\*fi);

7: summCosZnakoPeremen

massPQ[n-1] - Math.cos(n\*fi); massPQ[n-1] + Math.cos(n\*fi);

8: cosRaznost

2 \* (Math.cos(fi) - Math.cos((2\*n+1)\*fi) );

9: sinDiv

Math.sin((n+1)\*fi) / Math.sin(n\*fi);

10: summSinNechetZnakoPeremen

massPQ[n-1] - Math.sin((2\*n-1)\*fi); massPQ[n-1] + Math.sin((2\*n-1)\*fi);

11: summCosNechetZnakoPeremen

massPQ[n-1] - Math.cos((2\*n-1)\*fi); massPQ[n-1] + Math.cos((2\*n-1)\*fi);

14: summSinChetnZnakoPeremen

massPQ[n] = massPQ[n-1] - Math.sin((2\*n)\*fi); else massPQ[n-1] + Math.sin((2\*n)\*fi);

15: summCosChetnZnakoPeremen

massPQ[n] = massPQ[n-1] - Math.cos((2\*n)\*fi); else massPQ[n-1] + Math.cos((2\*n)\*fi);

16: summSin4n\_1

massPQ[n] = massPQ[n-1] + Math.sin((4\*n-3)\*fi);

17: summCosDivSin

massPQ[n] = (massPQ2[n-1] + Math.cos((2\*n-1)\*fi)) / (this.massPQ3[n-1] + Math.sin((2\*n-1)\*fi))