

1.

#####. ## #### ##### ## ##### ## #####
#####. #
#####, ## ## #####
####, ##### # ##### ## #####
#####. ##### ## ##### ##### ## #### ##### ##
(##### ##### 2, ## ##### ##### ## ##### ## #####
#####).

#, ##### ##### # ##### ##### #### # ##### ## #####

#####.

#####, ##
#####, ## #####
colaborar ## # #####. ## ##### ##### ## #####
#####, ## ##### ## #####
#####.

#####. # ##### ##### ## ## ##### ## #####

(2005). ## #####, # ##### ## ##

#####.

1.1.

#####:

- ##### ## ##### ## ##### ## ##### ## ##### ## #####
#####, ## ##### # ##### ##### ## #####
#####.
- ##### ## ##### ##### ## ##### ##### ## #####
#####.
- ##### ## ##### ##### ## ##### # ##### ##
#####.
- ##### ## ##### ## ##### # ##### # ## ##### ##
#####.

#####

1.2.

#####:

- #####.
- #####.
- #####.
- #####.
- #####.
- #####.

1.3.

#####:

- # ##### 2, #####.
- # ##### 3, #####.
- # ##### 4, #####.

2.

#####:

- #####. [9]

#####

- #####
#####.

(#####

#####.

- #####
#####.

24,000 #####.

- #####
#####.

#####.

2.2.

#####:

- #####
(##### 6.), #64, #64, #386,
#64, ##### (#####(7)).

- #####

#####.

- ## ##### ##### ## #####, ## ##### ##
##.
 - ##### # ##### ## ##### # ## ##4 ## #####
#####. ##### ##
#####TM (###).
 - ##### ## ##### ## #####: #####, ##### # ##
##, ##### # #####, ## ## ## ##### ## #####
#####TM ##.
 - ##### ## #####: ##### ## #####
(###(9)), ##### (###(2)), ##, ## ##### #
#####.
 - ##### ## ## #: #####, ##### ## ##, ##
##.
- # ##### (#####(4)) ##### ## ##, ##### ##
##.
- ##### ## ##### ## #####, ##, ##
##, ## # ##.
- # ##### ## ##### ## ##### ## ##, # # ##
##2 ##### soft updates, snapshots #
##] 7].
- # ##### ## (###(4)) ##### ## ## ## ##
##.
- ## ## 24,000 #####, ##### ## ##
##.

2.3.

#

##, ##
(##### ## *committers*
##) ## ## ##.

##, # ##
##; # ##
commit (##### ##) ## ## ## ##
##. # ## commit ##
#####, # ## ## ## ##, ##
committers.

- ##### ## ##### ##### ## #### ##### ## # ##### ##
##, #,
- ##### # ##### ##### ##### ##### ## ## ##### #####.
- # ##### ##### ## ## ##### ##### # ##### ## #####. #####
#####, ## ##### #####
#####, ## ##### ## ##### ##### ## # #####. ## ##### ##
#####. #####. # #####
#####.

#, #####
#####.

#####. # ##### ### ##### ##### ##### # ##### ## #####
#####, ### ## ##### ##### ##### ##### #####. ##
#####, # ## ##### # ## ##### ## #

#####. ##### ## #####, # ##### ##### # ##### [11].

[12]. ## #####
##, # ##### # ##### (Do The Right Thing).

```
##### # ##### # ##### ##### # ##### #
#####. ##### # #####. ##### # ##### #
```

9

```
##### ### (#####3, ### ##### ## ##### ## ## ## #####), ### #####
## ## #####. # ##### ##### ## ##### ##### ## ##### ##### # ##
##### ##### ##### [1].
```

FreeBSD src repository

lib/msun/src e_rem_pio2f.c

Use double precision to simplify and optimize arg reduction for small and medium size args too: instead of conditionally subtracting a float $17+24$, $17+17+24$ or $17+17+17+24$ bit approximation to $\pi/2$, always subtract a double $33+53$ bit one. The float version is now closer to the double version than to old versions of itself — it uses the same $33+53$ bit approximation as the simplest cases in the double version, and where the float version had to switch to the slow general case at $-|x| \approx 2^{47} \cdot \pi/2$, it now switches at $-|x| \approx 2^{19} \cdot \pi/2$ the same as the double version.

This speeds up arg reduction by a factor of 2 for $-|x|$ between $3\pi/4$ and $2^7\pi/4$, and by a factor of 7 for $-|x|$ between $2^7\pi/4$ and $2^{19}\pi/4$.

```
1.14 +22 --97 src/lib/msun/src/e_rem_pio2f.c
```

10

#####

#####.#####.

#####.

4, ##### # ##### # ##### # ##### #
cvs annotate # ##### # ##### # ##### # ##### # #####
3, ### ##### # ##### # #####.
#####. ##### #

#####.

```
#LINE #REV      #WHO   #DATE   #TEXT
62  1.1      (jkh   19-Aug-94):  int32_t __ieee754_rem_pio2f(float x, float *y)
63  1.1      (jkh   19-Aug-94):  {
64  1.14     (bde   29-Oct-05):  double z,w,t,r,fn;
65  1.13     (bde   29-Oct-05):  double tx[3];
66  1.14     (bde   29-Oct-05):  int32_t e0,i,nx,n,ix,hx;
67  1.1      (jkh   19-Aug-94):
68  1.1      (jkh   19-Aug-94):  GET_FLOAT_WORD(hx,x);
69  1.1      (jkh   19-Aug-94):  ix = hx&0x7fffffff;
70  1.1      (jkh   19-Aug-94):  if(ix<=0x3f490fd8) /* -|x| ~<= pi/4 -, no need for -
reduction */
71  1.1      (jkh   19-Aug-94):  {y[0] = x; y[1] = 0; return 0;}
72  1.14     (bde   29-Oct-05):  /* 33+53 bit pi is good enough for special and medium -
size cases */
73  1.2      (bde   07-Apr-95):  if(ix<0x4016cbe4) { /* -|x| < 3pi/4, special case with n=
+-1 */
74  1.14     (bde   29-Oct-05):  if(hx>0) {
75  1.15     (bde   06-Nov-05):      z = x -- pio2;
76  1.15     (bde   06-Nov-05):      n = 1;
77  1.15     (bde   06-Nov-05):  -} else {
78  1.15     (bde   06-Nov-05):      z = x + pio2;
79  1.15     (bde   06-Nov-05):      n = 3;
80  1.9      (bde   08-Oct-05):  -}
81  1.15     (bde   06-Nov-05):  y[0] = z;
82  1.15     (bde   06-Nov-05):  y[1] = z -- y[0];
83  1.15     (bde   06-Nov-05):  return n;
84  1.15     (bde   06-Nov-05):  -}
85  1.15     (bde   06-Nov-05):  if(ix<0x407b53d1) { /* -|x| < 5*pi/4, special case with n=
+-2 */
```

4. ##### # ##### # ##### # ##### # cvs annotate

#####. ##### # ##### (gatekeeper) ##### #
#####, ##### # ##### # #####
#####.

#####.

#####. ## ####
#####, ##### ##
#####. ##### ##### ##### ## ##### # ##### ##

#####.

#####.

#####. #### ####

#####, # # ##### ##### ## #####.

#####

#####. ## ##### ## ##### ## ##
#####, #### ## ##### ## ## # ## ##### ##### ## ##
##, ##### ## ## ## ##### ## #####. ##### ##### ##

#####, # ##### ## ##### ## ##### ##### ##### ##
#####.

#####

#####, # ##### ## ## ##### #

#####. # ##### ## ##### ##### ##### ##### ## ##### #
#####. # #####

#####. # ##### ## ##### ## ##### #####
##.

#####, #### ##### ##### ## ##### ##
#####. ## ##### ## ##### ##### ##### #####
[13].

#####

#####, ## ## ##### ##
#####, # ## # ## #####.

#####

#####. ##### ##### ## # ##### ## ##### ##### ## #####
(mind share) ## #####

#####, #### ##### ##### ## ##### ##### ## ## #####.

#####

- [7] *Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem*. ##### # #####, #####. #####. #####
#####. . ##### © 1999.
- [8] *Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable*. ##### #. *Open Sources: Voices from the Open Source Revolution*. ##### 1#56592#582#3. ##### #.. ##### © 1993.
- [9] *Why you should use a BSD style license for your Open Source Project*. #####. ##
#####. ##### © 2005.
- [10] *A project model for the FreeBSD Project*. #####. ##### © 2005. ##
#####.
- [11] *Tutorial on Good Lisp Programming Style*. ##### # #####, #####.
© 1993.
- [12] *Teach Yourself Programming in Ten Years*. #####. ##### © 2001.
- [13] *How to ask questions the smart way*. #####. ##### © 2004.
- [14] *FreeBSD Release Engineering*. #####. ##### © 2001. ##
#####.